SIAM J. OPTIMIZATION
Vol. 3, No. 1, pp. 1-24, February 1993

001

A THEORETICAL AND EXPERIMENTAL STUDY OF THE SYMMETRIC RANK-ONE UPDATE*

H. FAYEZ KHALFAN†, R. H. BYRD‡, AND R. B. SCHNABEL‡

Abstract. This paper first discusses computational experience using the SR1 update in conventional line search and trust region algorithms for unconstrained optimization. The experiments show that the SR1 is very competitive with the widely used BFGS method. They also indicate two interesting features: the final Hessian approximations produced by the SR1 method are not generally appreciably better than those produced by the BFGS, and the sequences of steps produced by the SR1 do not usually seem to have the "uniform linear independence" property that is assumed in recent convergence analysis. This paper presents a new analysis that shows that the SR1 method with a line search is $(n+1)$-step q-superlinearly convergent without the assumption of linearly independent iterates. This analysis assumes that the Hessian approximations are positive definite and bounded asymptotically, which, from computational experience, are reasonable assumptions.

Key words. quasi-Newton method, symmetric rank-one update, superlinear convergence

AMS(MOS) subject classifications. 65, 49

1. Introduction.

This paper is concerned with secant (quasi-Newton) methods for finding a local minimum of the unconstrained optimization problem

$$\min_{x \in R^n} f(x). \tag{1.1}$$

We assume that $f(x)$ is at least twice continuously differentiable, and that the number of variables n is sufficiently small to permit storage of an $n \times n$ matrix, and $O(n^2)$ or possibly $O(n^3)$ arithmetic operations per iteration.

Algorithms for solving (1.1) are iterative, and the basic framework of an iteration of a secant method is:

> Given the current iterate $x_c, f(x_c), \nabla f(x_c)$, or finite difference approximation, and $B_c \in R^{n \times n}$ symmetric (a secant approximation to $\nabla^2 f(x_c)$):
>
> > Select the new iterate x_+ by a line search or trust region method based on the quadratic model $m(x_c+d) = f(x_c) + \nabla f(x_c)^T d + \frac{1}{2} d^T B_c d$.
> >
> > Update B_c to B_+ such that B_+ is symmetric and satisfies the secant equation $B_+ s_c = y_c$, where $s_c = x_+ - x_c$ and $y_c = \nabla f(x_+) - \nabla f(x_c)$.

In this paper, we consider the symmetric rank-one (SR1) update for the Hessian approximation

$$B_+ = B_c + \frac{(y_c - B_c s_c)(y_c - B_c s_c)^T}{s_c^T (y_c - B_c s_c)} \tag{1.2}$$

and, for purpose of comparison, the BFGS update

$$B_+ = B_c + \frac{y_c y_c^T}{y_c^T y_c} + \frac{B_c s_c s_c^T B_c}{s_c^T y_c}. \tag{1.3}$$

* Received by the editors March 13, 1991; accepted for publication (in revised form) December 2, 1991. This research was supported by National Science Foundation grants CCR-8702403 and CCR-8920519, Air Force Office of Scientific Research grant AFOSR-90-0109, and Army Research Office contract DAAL 03-88-K-0086.

† Department of Mathematics, United Arab Emirates University, Al-Ain, United Arab Emirates.

‡ Department of Computer Science, Campus Box 430, University of Colorado, Boulder, Colorado 80309.

For background on these updates and others, see Fletcher (1980), Gill, Murray, and Wright (1981), and Dennis and Schnabel (1983).

The BFGS update has been the most commonly used secant update for many years. It makes a symmetric, rank-two change to the previous Hessian approximation B_c, and if B_c is positive definite and $s_c^T y_c > 0$, then B_+ is positive definite.

The BFGS method has been shown by Broyden, Dennis, and Moré (1973) to be locally q-superlinearly convergent provided that the initial Hessian approximation is sufficiently accurate. Powell (1976) proved a global superlinear convergence result for the BFGS method when applied to strictly convex functions and used in conjunction with line searches that satisfy Wolfe conditions. The BFGS update has been used successfully in many production codes for unconstrained optimization.

The SR1 formula, on the other hand, makes a symmetric rank-one change to the previous Hessian approximation B_c. Compared with other secant updates, the SR1 update is simpler and may require less computation per iteration when unfactored forms of updates are used. (Factored updates are those in which a decomposition of B_c is updated at each iteration.) A basic disadvantage of the SR1 update, however, is the fact that its denominator may be zero or nearly zero, which causes numerical instability. A simple remedy to this problem is to set $B_+ = B_c$ whenever this difficulty arises, but this may prevent fast convergence. Another problem is that the SR1 update may not preserve positive definiteness even if this is possible, i.e., when B_c is positive definite and $s_c^T y_c > 0$.

Fiacco and McCormick (1968) showed that if the SR1 update is applied to a positive definite quadratic function in a line search method, then, provided that the updates are all well defined, the solution is reached in at most $n+1$ iterations. Furthermore, if $n+1$ iterations are required, then the final Hessian approximation is the actual Hessian at the solution. This result is not generally true for the BFGS update or other members of the Broyden family, unless exact line searches are used.

For nonquadratic functions, however, convergence of the SR1 is not as well understood as convergence of the BFGS method. In fact, Broyden, Dennis, and Moré (1973) have shown that under their assumptions the SR1 update can be undefined, and thus their convergence analysis cannot be applied in this case. Also, no global convergence result similar to that for the BFGS method given by Powell (1976) exists, so far, for the SR1 method when applied to a nonquadratic function.

Recent work by Conn, Gould, and Toint (1988a, 1988b, 1991) has sparked renewed interest in the SR1 update. Conn, Gould, and Toint (1991) proved that the sequence of matrices generated by the SR1 formula converges to the actual Hessian at the solution $\nabla^2 f(x_*)$, provided that the steps taken are uniformly linearly independent, that the SR1 update denominator is always sufficiently different from zero, and that the iterates converge to a finite limit. (Using this result it is simple to prove that the rate of convergence is q-superlinear.) On the other hand, for the BFGS method Ge and Powell (1983) proved, under a different set of assumptions, that the sequence of generated matrices converges, but not necessarily to $\nabla^2 f(x_*)$.

The numerical experiments of Conn, Gould, and Toint (1988b) indicate that minimization algorithms based on the SR1 update may be competitive computationally with methods using the BFGS formula. The algorithm used by Conn, Gould, and Toint (1988b) is designed to solve problems with simple bound constraints, i.e., $l_i \leqq x_i \leqq u_i$, $i = 1, 2, \ldots, n$. The bound constraints are incorporated into a box constrained trust region strategy for calculating global steps, in which an inexact Newton's method oriented towards large-scale problems is used. This method uses a conjugate gradient method to approximately solve the trust region problem at each iteration, and also incorporates a new procedure that allows the set of active bound constraints to change substantially at each iteration. In this context, Conn, Gould, and Toint (1988b) conclude that the SR1 performance is generally somewhat better than the BFGS in terms of iterations and function evaluations on their test problems. They point out that the use of a trust region removes a main disadvantage of SR1 methods by allowing a meaningful step to be taken even when the approximation is indefinite. They also point out that the skipping technique used when the SR1 denominator is nearly zero was almost never used in their tests. They attribute part of the success of the SR1 to the possible convergence of the updates to the true second derivatives, as discussed above. Conn, Gould, and Toint (1991) tested this convergence using random search directions. These tests showed that, in comparison with other updates such as the BFGS and the DFP, the SR1 generates more accurate Hessian approximations, and that, although the PSB has the potential to give accurate Hessian approximations, the convergence is sometimes so slow as to be almost unobservable.

The purpose of this paper is to better understand the computational and theoretical properties of the SR1 update in the context of basic line search and trust region methods for unconstrained optimization. In the next section, we present computational results we obtained for the SR1 and the BFGS methods using standard line search and trust region algorithms for small to medium sized unconstrained optimization problems. We also report on tests of the convergence of the sequence of Hessian approximation matrices $\{B_k\}$, generated by the SR1 and BFGS formulas, and on tests of the condition of uniform linear independence of the sequence of steps which is required by the theory of Conn, Gould, and Toint (1991). These results indicate that this assumption may not be satisfied for many problems. Therefore, in § 3, we prove a new convergence result without the assumption of uniform linear independence of steps. Instead, it requires the assumption of boundedness and positive definiteness of the Hessian approximation. In § 4, we present computational results regarding the positive definiteness of the SR1 update and an interesting example. Finally, in § 5 we make some brief conclusions and comments regarding future research.

2. Computational results and algorithms. In this section, we present and discuss some numerical experiments that were conducted in order to test the performance of secant methods for unconstrained optimization using the SR1 formula against those using the BFGS update.

The algorithms we used are from the UNCMIN unconstrained optimization software package (Schnabel, Koontz, and Weiss (1985)), which provides the options of both line search and trust region strategies for calculating global steps. The line search is based on backtracking, using a quadratic or cubic modeling of $f(x)$ in the direction of search, and the trust region step is determined using the "hook step" method to approximately minimize the quadratic model within the trust region. The frameworks of these algorithms are given below.

ALGORITHM 2.1. Quasi-Newton method (line search).

Step 0. Given an initial point x_0, an initial positive definite matrix B_0, and $\alpha = 10^{-4}$, set k (iteration number) $= 0$.

Step 1. If a convergence criterion is achieved, then stop.

Step 2. Compute a quasi-Newton direction

$$p_k = -(B_k + \mu_k I)^{-1}\nabla f(x_k),$$

where μ_k is a nonnegative scalar such that $\mu_k = 0$ if B_k is safely positive definite, else $\mu_k > 0$ is such that $B_k + \mu_k I$ is safely positive definite.

```
Step 3. {Using a backtracking line search, find an acceptable steplength.}
    (3.1) Set $\lambda_k = 1$.
    (3.2) If $f(x_{k+1}) \leqq f(x_k) + \alpha\lambda_k \nabla f(x_k)^T p_k$, then go to Step 4.
    (3.3) If first backtrack, then select the new $\lambda_k$ such that $x_{k+1}(\lambda_k)$ is the
          local minimizer of the one-dimensional quadratic interpolating
          $f(x_k)$, $\nabla f(x_k)^T p_k$, and $f(x_k + p_k)$, but constrain the new $\lambda_k$ to be
          $\geqq 0.1$, else select the new $\lambda_k$ such that $x_{k+1}(\lambda_k)$ is the local minimizer
          of the one-dimensional cubic interpolating $f(x_k)$, $\nabla f(x_k)^T p_k$,
          $f(x_{k+1}(\lambda_{\text{prev}}))$, and $f(x_{k+1}(\lambda_{\text{2prev}}))$ but constrain the new $\lambda_k$ to be in
          $[0.1\lambda_{\text{prev}}, 0.5\lambda_{\text{prev}}]$.
          ($x_{k+1}(\lambda) = x_k + \lambda p_k$ and $\lambda_{\text{prev}}$, $\lambda_{\text{2prev}}$ = previous two steplengths.)
    (3.4) Go to (3.2).
Step 4. Set $x_{k+1} = x_k + \lambda_k p_k$.
Step 5. Compute the next Hessian approximation $B_{k+1}$.
Step 6. Set $k = k+1$, and go to Step 1.
```

ALGORITHM 2.2. Quasi-Newton method (trust region).

```
Step 0. Given an initial point $x_0$, an initial positive definite matrix $B_0$, an initial
        trust region radius $\Delta_0$, $\eta_1 \in (0, 1)$, and $\eta_2 \geqq 1$, set $k = 0$.
Step 1. If a convergence criterion is achieved, then stop.
Step 2. If $B_k$ is not positive definite, set $\hat{B}_k = B_k + \mu_k I$ where $\mu_k$ is such that
        $\hat{B}_k = B_k + \mu_k I$ is safely positive definite, else set $\hat{B}_k = B_k$.
Step 3. {Compute trust region step by hook step approximation.}
        Find an approximate solution to

            $\min_{s \in R^n} \nabla f(x_k)^T s + \frac{1}{2} s^T \hat{B}_k s \quad \text{subject to } \|s\| \leqq \Delta_k$

        by selecting

            $s_k = -(\hat{B}_k + \nu_k I)^{-1} \nabla f(x_k), \qquad \nu_k \geqq 0$

        such that $\|s_k\| \in [0.75\Delta_k, 1.5\Delta_k]$, or

            $s_k = -\hat{B}_k^{-1} \nabla f(x_k),$

        if $\|\hat{B}_k^{-1} \nabla f(x_k)\| \leqq 1.5\Delta_k$.
Step 4. Set $\text{ared}_k = f(x_k + s_k) - f(x_k)$.
Step 5. If $\text{ared}_k \leqq 10^{-4} \nabla f(x_k)^T s_k$, then
    (5.1) Set $x_{k+1} = x_k + s_k$;
    (5.2) Calculate $\text{pred}_k = \nabla f(x_k)^T s_k + \frac{1}{2} s_k^T B_k s_k$;
    (5.3) If $(\text{ared}_k / \text{pred}_k) < 0.1$, then set $\Delta_{k+1} = \Delta_k / 2$, else if $(\text{ared}_k / \text{pred}_k) >$
          $0.75$, then set $\Delta_{k+1} = 2\Delta_k$, otherwise $\Delta_{k+1} = \Delta_k$;
    (5.4) Go to Step 7;
Step 6. Else
    (6.1) If the relative steplength is too small, then stop; else calculate the
          $\lambda_k$ for which $x_k + \lambda_k s_k$ is the minimizer of the one-dimensional
          quadratic interpolating $f(x_k)$, $f(x_k + s_k)$, and $\nabla f(x_k)^T s_k$; set the new
          $\Delta_k = \lambda_k \|s_k\|$, but constrain the new $\Delta_k$ to be between 0.1 and 0.5
          times the current $\Delta_k$.
    (6.2) Go to Step 3.
Step 7. Compute the next Hessian approximation, $B_{k+1}$.
Step 8. Set $k = k+1$, and go to Step 1.
```

Procedures for updating λ_k in Step 3 of Algorithm 2.1 are found in Algorithm A6.3.1 of Dennis and Schnabel (1983). While a steplength $\lambda_k > 1$ is not considered in the reported results, in our experience permitting $\lambda_k > 1$ makes very little difference on these test problems. Procedures for finding ν_k in Step 3 of the trust region algorithm are found in Algorithm A6.4.2 of Dennis and Schnabel (1983), and are based on Hebden (1973) and Moré (1977). In both algorithms, the procedure for selecting μ_k in Step 2 is found in Gill, Murray, and Wright (1981). (They give an algorithm for finding a diagonal matrix D, such that $B_k + D$ is safely positive definite. If $D = 0$, then μ_k is set to 0, else an upper bound b_1 on μ_k is calculated using the Gerschgorin circle theorem, and μ_k is set to $\min\{b_1, b_2\}$ where $b_2 = \max\{[D]_{ii}, 1 \leqq i \leqq n\}$.) In our experience, when B_k is indefinite, μ_k is quite close to the most negative eigenvalue of B_k, so that the algorithm usually finds an approximate minimizer of the quadratic model subject to the trust region constraint.

Both algorithms terminate if one of the following stopping criteria is met.

(1) The number of iterations exceeds a given upper limit.

(2) The relative gradient,

$$\max_{1 \leqq i \leqq n} \left\{ |[\nabla f(x_k)]_i| \frac{\max\{|[x_{k+1}]_i|, 1\}}{\max\{|f(x_{k+1})|, 1\}} \right\},$$

is less than a given gradient tolerance.

(3) The relative step,

$$\max_{1 \leqq i \leqq n} \left\{ \frac{\max\{|[x_{k+1}]_i - [x_k]_i|\}}{\max\{|[x_{k+1}]_i|, 1\}} \right\},$$

is less than a given step tolerance.

All the algorithms used $B_0 = I$.

2.1. Comparison of the SR1 and the BFGS methods. Using the above-outlined algorithms, we tested the SR1 method and the BFGS method on a variety of test problems selected from Moré, Garbow, and Hillstrom (1981) and from Conn, Gould, and Toint (1988b) (see Table A1 in the Appendix). First derivatives were approximated using finite differences. The gradient stopping tolerance used was 10^{-5}, and the step tolerance was (machine epsilon)$^{1/2}$. The upper bound used on the number of iterations was 500. As was done in Conn, Gould, and Toint (1988b), we skipped the SR1 update if either

$$|s_k^T(y_k - B_k s_k)| < r \|s_k\| \, \|y_k - B_k s_k\|,$$

where $r = 10^{-8}$, or $\|B_{k+1} - B_k\| > 10^8$. The BFGS update was skipped if $s_k^T y_k <$ (machine epsilon)$^{1/2} \|s_k\| \, \|y_k\|$. All experiments were run using double precision arithmetic on a Pyramid P90 computer that has a machine epsilon of order 10^{-16}.

For each test function, Tables A2 and A3 in the Appendix report the performance of the SR1 and BFGS methods using the line search and trust region algorithm, respectively. The tables contain the number of the function as given in the original source (see Table A1), the dimension of the problem (n), the number of iterations required to solve the problem (itrn.), the number of function evaluations (f-eval.) required to solve the problem (which includes n for each finite difference gradient evaluation), and the relative gradient at the solution (rgx). The last column (sp) indicates whether the starting point used is x_0, $10x_0$, or $100x_0$, where x_0 is the standard starting point.

In order to compare the performance of the two methods with respect to the number of iterations and the number of function evaluations required to solve these problems, we consider problems solved by both methods and calculate the ratio of the mean of the number of iterations (or function evaluations) required to solve these problems by the SR1 method to the corresponding mean for the BFGS method. Table 1 below reports the ratios of these means, using both arithmetic mean and geometric mean. These numbers indicate that on the set of test problems we used, the SR1 is 10 percent to 15 percent faster and cheaper than the BFGS method.

TABLE 1
Ratio of SR1 *cost to* BFGS *cost.*

	Line search		Trust region	
Mean	Itrn.	Function evaluations	Itrn.	Function evaluations
Arithmetic	0.82	0.83	0.84	0.88
Geometric	0.83	0.85	0.84	0.92

Table 2 gives the number of problems where the SR1 method requires at least 5, 10, 20, 30, 40, and 50 iterations less than the BFGS method, and vice versa. This table, which is based on numbers from Table A2, also indicates the superiority of the SR1 on these problems.

TABLE 2
Comparisons of iterations.

	Line search						Trust region					
Iterations different	5	10	20	30	40	50	5	10	20	30	40	50
SR1 better	26	20	13	10	7	3	27	20	11	9	5	1
BFGS better	7	5	2	2	1	1	8	6	3	1	1	1

2.2. Error in the Hessian approximation and uniform linear independence. In an attempt to understand the difference between the SR1 and the BFGS, we tested how closely the final Hessian approximations produced by the line search and trust region SR1 and BFGS algorithms come to the exact Hessians at the final iterates. Recall that the Hessian error for the SR1 is analyzed by Conn, Gould, and Toint (1991) under the assumption of uniform linear independence which we redefine here.

DEFINITION. A sequence of vectors $\{s_k\}$ in R^n is said to be uniformly linearly independent if there exist $\zeta > 0$, k_0, and $m \geqq n$ such that, for each $k \geqq k_0$, one can choose n distinct indices $k \leqq k_1 < \cdots < k_n \leqq k + m$ such that the minimum singular value of the matrix $S_k = [s_{k_1}/\|s_{k_1}\|, \ldots, s_{k_n}/\|s_{k_n}\|]$ is $\geqq \zeta$.

Using this definition, Theorem 2 of Conn, Gould, and Toint (1991) proves the following.

THEOREM 2.1 (Conn, Gould, and Toint (1991)). *Suppose that $f(x)$ is twice continuously differentiable everywhere, and that $\nabla^2 f(x)$ is bounded and Lipschitz continuous, that is, there exist constants $M > 0$ and $\gamma > 0$ such that for all x, $y \in R^n$,*

$$\|\nabla^2 f(x)\| \leqq M \quad \textit{and} \quad \|\nabla^2 f(x) - \nabla^2 f(y)\| \leqq \gamma \|x - y\|.$$

Let $x_{k+1} = x_k + s_k$, where $\{s_k\}$ is a uniformly linearly independent sequence of steps, and *suppose that* $\lim_{k\to\infty}\{x_k\} = x_*$ *for some* $x_* \in R^n$. *Let* $\{B_k\}$ *be generated by the* SR1 *formula*

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{s_k^T(y_k - B_k s_k)},$$

where B_0 *is symmetric, and suppose that for all* $k \geqq 0$, y_k *and* s_k *satisfy*

$$|s_k^T(y_k - B_k s_k)| \geqq r\|s_k\|\,\|y_k - B_k s_k\|, \tag{2.1}$$

for some fixed $r \in (0, 1)$. *Then* $\lim_{k\to\infty}\|B_k - \nabla^2 f(x_*)\| = 0$.

We now present some computational tests to determine to what extent such Hessian convergence occurs in practice. For these tests we used analytic gradients and a gradient stopping tolerance of 10^{-10} and computed the quantity

$$\|B_l - \nabla^2 f(x_l)\| / \|\nabla^2 f(x_l)\|,$$

where x_l is the solution obtained by the algorithm, and B_l is the Hessian approximation at x_l. These results are reported in Tables A4 and A5 in the Appendix and summarized in Tables 3 and 4. Tables 3 and 4 list, for each method, the number of problems for which $\|B_l - \nabla^2 f(x_l)\| / \|\nabla^2 f(x_l)\|$ lies in a given range.

While the SR1 seems to produce slightly better final approximations than the BFGS, there is no evidence from Tables 3 and 4 that it significantly outperforms the BFGS with respect to convergence to the actual Hessian at the solution. Also, in a good number of cases, neither method comes close to the correct Hessian.

TABLE 3
Number of problems with $\|B_l - \nabla^2 f(x_l)\| / \|\nabla^2 f(x_l)\|$ *in indicated range (line search methods).*

	$\leqq 10^{-4}$	$[10^{-4}, 10^{-3})$	$[10^{-3}, 10^{-2})$	$[10^{-2}, 10^{-1})$	$[10^{-1}, 1)$	$\geqq 1$
SR1	4	3	2	8	3	8
BFGS	3	0	1	10	6	8

The lack of convergence of the SR1 Hessian approximations to the correct value in many of these problems may appear to conflict with the analysis of Conn, Gould, and Toint (1991) given in Theorem 2.1. In fact, there are two possible explanations for this apparent conflict: either the algorithm has not converged closely enough for the final convergence of the matrices to be apparent (this is hard to test in finite precision arithmetic) or an assumption of Theorem 2.1 must be violated. The two assumptions of Theorem 2.1 that could possibly be invalid are (1) that the denominator of the SR1 is not too small (2.1), and (2) the uniform linear independence condition. In our experiments, (2.1) was violated at most once for each test problem, and so this assumption does not appear to be a problem in the SR1 method. Thus we decided to test whether the uniform linear independence condition is satisfied for these problems.

Since the uniform linear independence condition would be very hard to test due to the freedom to choose m and ζ in the definition of uniform linear independence,

TABLE 4
Number of problems with $\|B_l - \nabla^2 f(x_l)\| / \|\nabla^2 f(x_l)\|$ *in indicated range (trust region methods).*

	$\leqq 10^{-4}$	$[10^{-4}, 10^{-3})$	$[10^{-3}, 10^{-2})$	$[10^{-2}, 10^{-1})$	$[10^{-1}, 1)$	$\geqq 1$
SR1	5	0	4	5	4	10
BFGS	0	0	5	7	7	9

we have tested a weaker condition. For each value $\tau = 10^{-i}$, $i = 1, 2, \ldots, 8$, we computed the number of steps (say m) required so that the smallest singular value of the matrix, $\hat{S}_m$, composed of the final normalized m steps of the algorithm, is greater than τ ($\hat{S}_m = [s_l/\|s_l\|, s_{l-1}/\|s_{l-1}\|, \ldots, s_{l-(m-1)}/\|s_{l-(m-1)}\|]$, where $m \geqq n$). Tables A6 and A7 contain the results of these experiments, which are summarized in Tables 5 and 6. A "*" entry in Tables A6 and A7 means that the smallest singular value is less than τ even if all the iterates are used.

These results indicate that the uniform linear independence assumption does not seem to hold for all problems, especially those with large dimensions. Therefore, in the next section we develop a convergence result for the SR1 method that does not make this assumption.

TABLE 5

Number of problems where $\sigma_{\min}(\hat{S}_m) > \tau$ for m/n in indicated range (line search SR1 *method).*

	m/n in					
τ	$[1, 2)$	$[2, 3)$	$[3-4)$	$[4-5)$	$[5-10)$	Never
10^{-1}	7	1	3	3	6	8
10^{-2}	12	1	0	3	5	7
10^{-8}	12	1	0	4	4	7

TABLE 6

Number of problems where $\sigma_{\min}(\hat{S}_m) > \tau$ for m/n in indicated range (trust region SR1 *method).*

	m/n in					
τ	$[1, 2)$	$[2, 3)$	$[3-4)$	$[4-5)$	$[5-10)$	Never
10^{-1}	4	3	0	3	6	12
10^{-2}	12	1	0	3	5	7
10^{-8}	13	0	0	3	5	7

3. Convergence rate of the SR1 without uniform linear independence. As was indicated at the end of the previous section, the condition of uniform linear independence of the sequence $\{s_k\}$ under which Conn, Gould, and Toint (1991) analyze the performance of the SR1 method may be too strong in practice. Therefore, in this section we consider the convergence rate of the SR1 method without this condition. We will show that if we drop the condition of uniform linear independence of $\{s_k\}$ but add instead the assumption that the sequence $\{B_k\}$ remains positive definite and bounded, then the line search algorithm, Algorithm 2.1, generates at least p q-superlinear steps out of every $n + p$ steps. This will enable us to prove that convergence is $2n$-step q-quadratic.

The basic idea behind our proof is that, if any step falls close enough to a subspace spanned by $m \leqq n$ recent steps, then the Hessian approximation must be quite accurate in this subspace. Thus, if in addition the step is the full secant step $-B_k^{-1}\nabla f(x_k)$, it should be a superlinear step. But in a line search method, for the step to be the full secant step, B_k must be positive definite, which accounts for the new assumption of positive definiteness of B_k at the good steps. In §4 we will show that empirically this assumption seems very sound, although counterexamples are possible.

Throughout this section the following assumptions will frequently be made.

ASSUMPTION 3.1. The function f has a local minimizer at a point x_* such that $\nabla^2 f(x_*)$ is positive definite, and its Hessian $\nabla^2 f(x)$ is Lipschitz continuous near x_*, that is, there exists a constant $\gamma > 0$ such that for all x, y in some neighborhood of x_*,

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leqq \gamma \|x - y\|.$$

ASSUMPTION 3.2. The sequence $\{x_k\}$ converges to the local minimizer x_*.

We first state the following result, due to Conn, Gould, and Toint (1991), which does not assume linear independence of the step directions and which will be used in the proof of the next lemma.

LEMMA 3.1. *Let* $\{x_k\}$ *be a sequence of iterates defined by* $x_{k+1} = x_k + s_k$. *Suppose that Assumptions* 3.1 *and* 3.2 *hold, that the sequence of matrices* $\{B_k\}$ *is generated from* $\{x_k\}$ *by the* SR1 *update, and that for each iteration*

$$|s_k^T(y_k - B_k s_k)| \geqq r\|s_k\| \, \|y_k - B_k s_k\|, \tag{3.1}$$

where r *is a constant* $\in (0, 1)$. *Then, for each* j, $\|y_j - B_{j+1}s_j\| = 0$, *and*

$$\|y_j - B_i s_j\| \leqq \frac{\gamma}{r}\left(\frac{2}{r}+1\right)^{i-j-2} \eta_{i,j} \|s_j\| \tag{3.2}$$

for all $i \geqq j+2$, *where* $\eta_{i,j} = \max\{\|x_p - x_s\| \,|\, j \leqq s \leqq p \leqq i\}$, *and* γ *is the Lipschitz constant from Assumption* 3.1.

Actually, it is apparent from the proof of Lemma 3.1 by Conn, Gould, and Toint (1991), that if the update is skipped whenever (3.1) is violated, then (3.2) still holds for all j for which (3.1) is true.

In the lemma below, we show that if the sequence of steps generated by an iterative process using the SR1 update satisfies (3.1), and the sequence of matrices is bounded, then out of any set of $n+1$ steps, at least one is very good. As in the previous lemma, condition (3.1) actually must only hold at this set of $n+1$ steps, as long as the update is not made when that condition fails.

LEMMA 3.2. *Suppose the assumptions of Lemma* 3.1 *are satisfied for the sequences* $\{x_k\}$ *and* $\{B_k\}$ *and that in addition there exists an* M *for which* $\|B_k\| \leqq M$ *for all* k. *Then there exists a* $K \geqq 0$ *such that for any set of* $n+1$ *steps,* $\mathcal{S} = \{s_{k_j}: K \leqq k_1 \leqq \cdots \leqq k_{n+1}\}$, *there exists an index* k_m *with* $m \in \{2, 3, \ldots, n+1\}$ *such that*

$$\frac{\|(B_{k_m} - \nabla^2 f(x_*))s_{k_m}\|}{\|s_{k_m}\|} < \bar{c}\varepsilon_{\mathcal{S}}^{1/n},$$

where

$$\varepsilon_{\mathcal{S}} = \max_{1 \leqq j \leqq n+1} \{\|x_{k_j} - x_*\|\}$$

and

$$\bar{c} = 4\left[\gamma + \sqrt{n}\,\frac{\gamma}{2}\left(\frac{2}{r}+1\right)^{k_{n+1}-k_1-2} + M + \|\nabla^2 f(x_*)\|\right].$$

Proof. Given $\mathcal{S}$, for $j = 1, 2, \ldots, n+1$ define

$$S_j = \left[\frac{s_{k_1}}{\|s_{k_1}\|}, \frac{s_{k_2}}{\|s_{k_2}\|}, \ldots, \frac{s_{k_j}}{\|s_{k_j}\|}\right].$$

We will first show that there exists $m \in [2, n+1]$ such that $s_{k_m}/\|s_{k_m}\| = S_{m-1}u - w$, S_{m-1} has full column rank and is well conditioned, and $\|w\|$ is very small. (In essence, either

$m = n+1$, S_{m-1} spans n-space well, and $w = 0$, or $m < n+1$, S_{m-1} has full rank and is well conditioned, and s_{k_m} is nearly in the space spanned by S_{m-1}.) Then, using the fact that $(B_{k_m} - \nabla^2 f(x_*))S_{m-1}$ is small due to the Hessian approximating properties of the SR1 update given in Lemma 3.1 above, and that w is small by this construction, we will have the desired result.

For $j \in \{1, \dots, n\}$, let τ_j be the smallest singular value of S_j and define $\tau_{n+1} = 0$. Note that

$$1 = \tau_1 \geqq \tau_2 \cdots \geqq \tau_{n+1} = 0.$$

Let m be the smallest integer for which

$$\frac{\tau_m}{\tau_{m-1}} < \varepsilon_{\mathcal{S}}^{1/n}. \tag{3.3}$$

Then since $m \leqq n+1$ and $\tau_1 = 1$,

$$\tau_{m-1} = \tau_1 \left(\frac{\tau_2}{\tau_1}\right) \cdots \left(\frac{\tau_{m-1}}{\tau_{m-2}}\right) > \varepsilon_{\mathcal{S}}^{(m-2)/n} > \varepsilon_{\mathcal{S}}^{(n-1)/n}. \tag{3.4}$$

Since $x_k \to x_*$, we may assume without loss of generality that $\varepsilon_{\mathcal{S}} \in (0, (\frac{1}{4})^{'n})$ for all k. Now we choose $z \in R^m$ such that

$$\|S_m z\| = \tau_m \|z\| \tag{3.5}$$

and

$$z = \begin{bmatrix} u \\ -1 \end{bmatrix},$$

where $u \in R^{m-1}$. (The last component of z is nonzero due to (3.3).) Let $w = S_m z$. Then, from the definition of S_m and z,

$$\frac{s_{k_m}}{\|s_{k_m}\|} = S_{m-1}u - w. \tag{3.6}$$

Since τ_{m-1} is the smallest singular value of S_{m-1} we have that

$$\|u\| \leqq \frac{1}{\tau_{m-1}} \|S_{m-1}u\| = \frac{1}{\tau_{m-1}} \left\| w + \frac{s_{k_m}}{\|s_{k_m}\|} \right\| \leqq \frac{\|w\|+1}{\tau_{m-1}}. \tag{3.7}$$

By (3.4) this implies that

$$\|u\| < \frac{\|w\|+1}{\varepsilon_{\mathcal{S}}^{(n-1)/n}}. \tag{3.8}$$

Also, using (3.5) and (3.7), we have that

$$\|w\|^2 = \|S_m z\|^2 = \tau_m^2 \|z\|^2 = \tau_m^2(1 + \|u\|^2) \leqq \tau_m^2 + \left(\frac{\tau_m}{\tau_{m-1}}\right)^2 (\|w\|+1)^2.$$

Therefore, since (3.3) implies that $\tau_m < \varepsilon_{\mathcal{S}}^{1/n}$, using (3.3),

$$\|w\|^2 < \varepsilon_{\mathcal{S}}^{2/n} + \varepsilon_{\mathcal{S}}^{2/n}(\|w\|+1)^2 < 4\varepsilon_{\mathcal{S}}^{2/n}(\|w\|+1)^2. \tag{3.9}$$

This implies that

$$\|w\|(1 - 2\varepsilon_{\mathcal{S}}^{1/n}) < 2\varepsilon_{\mathcal{S}}^{1/n},$$

and hence $\|w\|<1$, since $\varepsilon_{\mathcal{S}}<(\frac{1}{4})^n$. Therefore, (3.8) and (3.9) imply that

$$\|u\|<\frac{2}{\varepsilon_{\mathcal{S}}^{(n-1)/n}}, \tag{3.10}$$

$$\|w\|<4\varepsilon_{\mathcal{S}}^{1/n}. \tag{3.11}$$

This gives the desired result that w is small, as well as a necessary bound on $\|u\|$.

Now we show that $\|(B_{k_j}-\nabla^2 f(x_*))S_{j-1}\|$, $j\in[2,n+1]$, is small. Note that this result is independent of the choice of j. By Lemma 3.1 we have that

$$\begin{aligned}\|y_i-B_{k_j}s_i\| &\leqq \frac{\gamma}{r}\left(\frac{2}{r}+1\right)^{k_j-i-2}\eta_{k_j,i}\|s_i\| \\ &\leqq 2\frac{\gamma}{r}\left(\frac{2}{r}+1\right)^{k_{n+1}-k_1-2}\varepsilon_{\mathcal{S}}\|s_i\|\end{aligned} \tag{3.12}$$

for all $i\in\{k_1,k_2,\dots,k_{j-1}\}$. Also, letting

$$G_i=\int_0^1\nabla^2 f(x_i+ts_i)\,dt,$$

we have

$$G_is_i=\int_0^1\nabla^2 f(x_i+ts_i)s_i\,dt=\nabla f(x_{i+1})-\nabla f(x_i)=y_i,$$

and by the Lipschitz continuity of $\nabla^2 f(x)$,

$$\begin{aligned}\|y_i-\nabla^2 f(x_*)s_i\| &= \|(G_i-\nabla^2 f(x_*))s_i\| \\ &= \left\|\int_0^1(\nabla^2 f(x_i+ts_i)-\nabla^2 f(x_*))s_i\,dt\right\| \\ &\leqq \|s_i\|\int_0^1\|\nabla^2 f(x_i+ts_i)-\nabla^2 f(x_*)\|\,dt \\ &\leqq \gamma\|s_i\|\int_0^1\|x_i+ts_i-x_*\|\,dt \\ &\leqq \gamma\|s_i\|\varepsilon_{\mathcal{S}},\end{aligned} \tag{3.13}$$

where γ is the Lipschitz constant. Therefore, using the triangle inequality and (3.12) and (3.13), we have

$$\begin{aligned}\left\|(B_{k_j}-\nabla^2 f(x_*))\frac{s_i}{\|s_i\|}\right\| &\leqq \left\|(y_i-B_{k_j})\frac{s_i}{\|s_i\|}\right\|+\left\|(y_i-\nabla f(x_*))\frac{s_i}{\|s_i\|}\right\| \\ &\leqq (2c+\gamma)\varepsilon_{\mathcal{S}},\end{aligned}$$

where

$$c=\frac{\gamma}{2}\left(\frac{2}{r}+1\right)^{k_{n+1}-k_1-2},$$

and hence for any $j\in[2,n+1]$,

$$\|(B_{k_j}-\nabla^2 f(x_*))S_{j-1}\|\leqq\sqrt{n}(2c+\gamma)\varepsilon_{\mathcal{S}}. \tag{3.14}$$

Finally, using (3.6) and (3.14) with $j=m$, (3.11), and (3.10) we have that

$$\begin{aligned}\frac{\|(B_{k_m}-\nabla^2 f(x_*))s_{k_m}\|}{\|s_{k_m}\|} &= \|(B_{k_m}-\nabla^2 f(x_*))(S_{m-1}u-w)\| \\ &\leqq \|(B_{k_m}-\nabla^2 f(x_*))S_{m-1}\|\,\|u\|+\|B_{k_m}-\nabla^2 f(x_*)\|\,\|w\| \\ &\leqq \|u\|\sqrt{n}(2c+\gamma)\varepsilon_{\mathcal{S}}+\|w\|(\|B_{k_m}\|+\|\nabla^2 f(x_*)\|) \\ &< \left(\frac{2}{\varepsilon_{\mathcal{S}}^{(n-1)/n}}\right)\sqrt{n}(2c+\gamma)\varepsilon_{\mathcal{S}}+4\varepsilon_{\mathcal{S}}^{1/n}(M+\|\nabla^2 f(x_*)\|) \\ &< 4[\sqrt{n}(c+\gamma)+M+\|\nabla^2 f(x_*)\|]\varepsilon_{\mathcal{S}}^{1/n} \\ &= \bar{c}\varepsilon_{\mathcal{S}}^{1/n}. \qquad \square\end{aligned}$$

In order to use this lemma to establish a rate of convergence we need the following result which is closely related to the well-known superlinear convergence characterization of Dennis and Moré (1974).

LEMMA 3.3. *Suppose the function f satisfies Assumption* 3.1. *If the quantities*

$$e_k=\|x_k-x_*\| \quad \textit{and} \quad \frac{\|(B_k-\nabla^2 f(x_*))s_k\|}{\|s_k\|}$$

are sufficiently small, and if $B_k s_k=-\nabla f(x_k)$, then

$$\|x_k+s_k-x_*\|\leqq\|\nabla^2 f(x_*)^{-1}\|\left[2\frac{\|(B_k-\nabla^2 f(x_*))s_k\|}{\|s_k\|}e_k+\frac{\gamma}{2}e_k^2\right].$$

Proof. By the definition of s_k,

$$\nabla^2 f(x_*)s_k=(\nabla^2 f(x_*)-B_k)s_k-\nabla f(x_k),$$

so that

$$s_k=-(x_k-x_*)+\nabla^2 f(x_*)^{-1}[(\nabla^2 f(x_*)-B_k)s_k-\nabla f(x_k)+\nabla^2 f(x_*)(x_k-x_*)]. \tag{3.15}$$

Therefore, using Taylor's theorem and Assumption 3.1,

$$\|x_k-x_*+s_k\|\leqq\|\nabla^2 f(x_*)^{-1}\|\left[\|(\nabla^2 f(x_*)-B_k)s_k\|+\frac{\gamma}{2}e_k^2\right]. \tag{3.16}$$

Now it follows from (3.15) that if $\|\nabla^2 f(x_*)^{-1}\|\,\|(B_k-\nabla^2 f(x_*))s_k\|/\|s_k\|\leqq\frac{1}{3}$, then by Taylor's theorem,

$$\|s_k\|\leqq\frac{3}{2}\left[\|x_k-x_*\|+\|\nabla^2 f(x_*)^{-1}\|\frac{\gamma}{2}\|x_k-x_*\|^2\right]\leqq 2\|x_k-x_*\|,$$

if e_k is sufficiently small. Using this inequality together with (3.16) gives the result. □

Using these two lemmas one can show that for any $p>n$, Algorithm 2.1 will generate at least $p-n$ superlinear steps every p iterations, provided that B_k is safely positive definite, which implies that B_k is not perturbed in Step 2 and $\mu_k=0$. In the following theorem, this is proved and used to establish a rate of convergence for Algorithm 2.1 under the assumption that the sequence $\{B_k\}$ becomes, and stays, positive definite. In a corollary we show that this implies that the rate of convergence for Algorithm 2.1 is $2n$-step q-quadratic. As we will see in the next section, our test results show that the positive definiteness condition is generally satisfied in practice. We are assuming here that if B_k is positive definite, then it is not perturbed in Step 2, i.e., we are assuming that "safely positive definite" just means positive definite.

THEOREM 3.1. *Consider Algorithm* 2.1 *and suppose that Assumptions* 3.1 *and* 3.2 *hold. Assume also that for all* $k \geqq 0$,

$$|s_k^T(y_k - B_k s_k)| \geqq r\|s_k\|\,\|y_k - B_k s_k\|$$

for a fixed $r \in (0, 1)$, *and that there exists* M *for which* $\|B_k\| \leqq M$ *for all* k. *Then, if there exists a* K_0 *such that* B_k *is positive definite for all* $k \geqq K_0$, *then for any* $p \geqq n+1$ *there exists a* K_1 *such that for all* $k \geqq K_1$,

$$e_{k+p} \leqq \alpha e_k^{p/n}, \tag{3.17}$$

where α *is a constant and* e_j *is defined as* $\|x_j - x_*\|$.

Proof. Since $\nabla^2 f(x_*)$ is positive definite, there exists a K_1, $\beta_1 > 0$ and $\beta_2 > 0$ such that

$$\beta_1[f(x_k) - f(x_*)]^{1/2} \leqq \|x_k - x_*\| \leqq \beta_2[f(x_k) - f(x_*)]^{1/2} \tag{3.18}$$

for all $k \geqq K_1$. Therefore, since we have a descent method, for all $l > k > K_1$,

$$\|x_l - x_*\| \leqq \frac{\beta_2}{\beta_1}\|x_k - x_*\|.$$

Now, given $k > K_1$ we apply Lemma 3.2 to the set $\{s_k, s_{k+1}, \dots, s_{k+n}\}$. Thus there exists $l_1 \in \{k+1, \dots, k+n\}$ such that

$$\frac{\|(B_{l_1} - \nabla^2 f(x_*))s_{l_1}\|}{\|s_{l_1}\|} < \bar{c}\left(\frac{\beta_2}{\beta_1}e_k\right)^{1/n}. \tag{3.19}$$

(If there is more than one such index l_1, we choose the smallest.) Equation (3.19) implies that for $\|x_{l_1} - x_*\|$ sufficiently small, by Theorem 6.4 of Dennis and Moré (1977), Algorithm 2.1 will choose $\lambda_{l_1} = 1$ so that $x_{l_1+1} = x_{l_1} + s_{l_1}$. This fact, together with Lemma 3.3 and (3.19), implies that if e_k is sufficiently small, then

$$e_{l_1+1} \leqq \hat{\alpha} e_k^{1/n} e_{l_1} \tag{3.20}$$

for some constant $\hat{\alpha}$. Now we can apply Lemma 3.2 to the set

$$\{s_k, s_{k+1}, \dots, s_{k+n}, s_{k+n+1}\} - \{s_{l_1}\}$$

to get l_2. Repeating this $n-p$ times we get a set of integers $l_1 < l_2 < \cdots < l_{p-n}$, with $l_1 > k$ and $l_{p-n} < k+p$ such that

$$e_{l_i+1} \leqq \hat{\alpha} e_k^{1/n} e_{l_i} \tag{3.21}$$

for each l_i. Now letting $h_j = [f(x_j) - f(x_*)]^{1/2}$, since we have a descent method,

$$h_{j+1} \leqq h_j, \tag{3.22}$$

and using (3.18) we have that for our arbitrary $k \geqq K_1$,

$$h_{l_i+1} \leqq \frac{1}{\beta_1} e_{l_i+1} \leqq \frac{\hat{\alpha}}{\beta_1} e_k^{1/n} e_{l_i} \leqq \frac{\hat{\alpha}\beta_2}{\beta_1} e_k^{1/n} h_{l_i} \tag{3.23}$$

for $i = 1, 2, \dots, p-n$. Therefore, using (3.22) and (3.23) we have that

$$h_{k+p} \leqq \left(\frac{\hat{\alpha}\beta_2}{\beta_1} e_k^{1/n}\right)^{p-n} h_k,$$

which, by (3.18), implies that

$$e_{k+p} \leqq \frac{\beta_2}{\beta_1}\left(\frac{\hat{\alpha}\beta_2}{\beta_1} e_k^{1/n}\right)^{p-n} e_k.$$

Therefore,

$$e_{k+p} \leqq \hat{\alpha}^{p-n}\left(\frac{\beta_2}{\beta_1}\right)^{p-n+1} e_k^{p/n},$$

and 3.17 follows. □

COROLLARY 3.1. *Under the assumptions of Theorem* 3.1 *the sequence* $\{x_k\}$ *generated by Algorithm* 2.1 *is* $n+1$*-step q-superlinear, i.e.*,

$$\frac{e_{k+n+1}}{e_k} \to 0,$$

and is $2n$*-step q-quadratic, i.e.*,

$$\limsup_{k\to\infty} \frac{e_{k+2n}}{e_k^2} \leqq \infty.$$

Proof. Let $p=n+1$ and $p=2n$ in Theorem 3.1. □

Note that a $2n$-step q-quadratically convergent sequence has an r-order of $(\sqrt{2})^{1/n}$. Since the integer p in the theorem is arbitrary, an interesting, purely theoretical question is what value of p will prove the highest r-convergence order for the sequence. It is not hard to show that, by choosing p to be an integer close to en, the r-order approaches $e^{1/en} \approx 1.44^{1/n}$ for n sufficiently large, and that this value is optimal for this technique of analysis.

4. Positive definiteness of the SR1 update. One of the requirements in Theorem 3.1 for the rate of convergence to be p-step q-superlinear is that the sequence $\{B_k\}$ generated by the SR1 method be positive definite. Actually, the proof of Theorem 3.1 only requires positive definiteness of B_k at the $p-n$ out of p "good iterations." In this section, we present computational results to confirm that, in practice, the SR1 method generally satisfies this requirement.

In Table A8 in the Appendix, in the fourth column, we report for each iteration whether B_k is positive definite or not. The fifth column reports the percentage of iterates at which the SR1 update is positive definite, and the sixth column contains the largest number j for which all of $B_{l-(j-1)}, \ldots, B_l$ are positive definite, where B_l is the Hessian approximation at the final iterate. The results of Table A8 are summarized in Table 7, which indicates that the SR1 formula was positive definite at least 70 percent of the time on every one of our test problems. In light of this, and since Theorem 3.1 really only requires positive definiteness at the "good steps" (at other steps all that is needed is that f be reduced), the chances that superlinear steps will be taken at least every n steps by the algorithm seem good. Another way of viewing this is the following. We know from Theorem 3.1 that out of every $2n$ steps at least n will be "good steps" as long as B_k is positive definite at these iterations. Thus if, for example, B_k is positive

TABLE 7
Percentage of iterations with B_k positive definite.

Percentage	≦70	[70, 90)	[80, 90)	[90, 100)	100
Problems	0	5	12	6	5

definite at 80 percent of these $2n$ steps, at least 30 percent of the $2n$ iterates must be "good steps."

We also tested the denominator condition that

$$|s_k^T(y_k - B_k s_k)| \geqq r\|s_k\| \, \|y_k - B_k s_k\| \tag{4.1}$$

where $r = 10^{-8}$ using standard initial points. The last column in Table A8, which reports the number of times this condition was violated, indicates that this condition is rarely violated in practice. This finding is consistent with the results of Conn, Gould, and Toint (1988b).

Finally we present an example that shows that it is possible for a line search SR1 algorithm to fail to have B_k positive definite at all iterations, and to converge linearly to the minimizer x_*. This shows that the assumptions of Theorem 3.1 cannot be guaranteed to hold. We then consider the same example in a trust region SR1 algorithm, and show that it does not suffer from the same problems. This leads us to feel that it may not be necessary to assume $\{B_k\}$ positive definite in order to prove superlinear convergence for a trust region SR1 method.

Example 4.1. Let

$$f(x) = \frac{1}{2}x^T x, \quad x_0 = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}, \quad \text{and} \quad B_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sigma \end{bmatrix},$$

where $\sigma < 0$. At the first iteration, the algorithm will compute

$$x_1 = x_0 - \begin{bmatrix} 1+\delta_0 & 0 \\ 0 & \sigma+\delta_0 \end{bmatrix}^{-1} \nabla f(x_0) = \frac{\delta_0}{1+\delta_0} x_0$$

for some $\delta_0 > -\sigma$, and accept this point as the next iterate. The SR1 update will produce $y_0 - B_0 s_0 = 0$, so that $B_1 = B_0$. The remaining iterates proceed analogously, so that for each k, $B_k = B_0$ and

$$x_{k+1} = \frac{\delta_k}{1+\delta_k} x_k$$

for some $\delta_k > -\sigma$, meaning that the rate of convergence is not better than linear with constant $|\sigma|/(1+|\sigma|)$.

It is interesting to consider the behavior on the same problem of a trust region SR1 algorithm that exactly solves the problem

$$\min_{s \in R^n} \nabla f(x_k)^T s + \tfrac{1}{2} s^T B_k s \quad \text{subject to } \|s\| \leqq \Delta_k \tag{4.2}$$

at each iteration. If there exists μ_0 such that $B_0 + \mu_0 I$ is positive definite and $\|(B_0 + \mu_0 I)^{-1} \nabla f(x_0)\| = \Delta_0$, then as in the line search method,

$$x_1 = \frac{\mu_0}{1+\mu_0} x_0 \quad \text{and} \quad B_1 = B_0.$$

Since $\text{ared}_0 = \text{pred}_0$, the trust region radius is not decreased. Thus eventually at some iterate k, we must have $\|(B_k + \mu_k I)^{-1} \nabla f(x_k)\| < \Delta_k$ for all $\mu_k > -\lambda_k$, where $\lambda_k < 0$ is the smallest eigenvalue of B_k. In this case the solution to (4.2) is the step

$$\begin{aligned} x_{k+1} &= x_k - (B_k - \lambda_k I)^+ \nabla f(x_k) - \nu e_2 \\ &= x_k - \left(\frac{1}{1-\sigma}\right) x_k - \nu e_2 \end{aligned}$$

for a $\nu \neq 0$ that makes $\|s_k\| = \Delta_k$. (Here $e_2 = (0, 1)^T$ is the eigenvector of B_k corresponding to the negative eigenvalue.) It is then straightforward to verify that $y_k - B_k s_k = \nu(\sigma - 1)e_2$, $B_{k+1} = I = \nabla^2 f(x)$, and $x_{k+2} = x_*$.

A practical trust region algorithm will not solve (4.2) exactly, but any algorithm that deals with the "hard case" (when $\|(B_k - \lambda_k I)^+ \nabla f(x_k)\| < \Delta_k$) well, such as algorithms of Moré and Sorenson (1983), will have the same effect. That is, at some point it will set

$$x_{k+1} = x_k - (B_k + \mu_k I)^{-1} \nabla f(x_k) - v_k,$$

where v_k is a negative curvature direction for B_k. This implies that $v_k^T e_2 \neq 0$, which in turn leads to $B_{k+1} = I$ and $x_{k+2} = x_*$. Thus the trust region method has the ability, for this example, to correct negative eigenvalues in the Hessian approximation. This indicates that it may be possible to establish superlinear convergence of a trust region SR1 algorithm without assuming a priori either strong linear independence of the iterates or positive definiteness of $\{B_k\}$. This issue is currently under investigation.

5. Conclusions and future research. In this paper, we have attempted to investigate theoretical and numerical aspects of quasi-Newton methods that are based on the SR1 formula for the Hessian approximation. We considered both line search and trust region algorithms.

We tested the SR1 method on a fairly large number of standard test problems from Moré, Garbow, and Hillstrom (1981), and Conn, Gould, and Toint (1988b). Our test results show that on the set of problems we tried, the SR1 method, on the average, requires somewhat fewer iterations and function evaluations than the BFGS method in both line search and trust region algorithms. Although there is no result for the BFGS method concerning the convergence of the sequence of approximating matrices to the correct Hessian like the one given by Conn, Gould, and Toint (1991) for the SR1, numerical tests do not show that the SR1 method is more accurate than the BFGS method in this regard. One reason for this, as indicated by our numerical experiments, is that the requirement of uniform linear independence that is needed by the theory of Conn, Gould, and Toint (1991) often fails to be satisfied in practice.

Under conditions that do not assume uniform linear independence of the generated steps, but do assume positive definiteness and boundedness of the Hessian approximations, we were able to prove $n+1$-step q-superlinear convergence, and $2n$-step quadratic convergence, of a line search SR1 method. We also gave numerical evidence that the SR1 update is positive definite most of the time, and that one of the potential problems of the formula, that of the denominator being zero, is rarely encountered in practice.

An interesting topic for future research that was mentioned in §4 is the convergence analysis of a trust region SR1 method, again without the assumption of uniform linear independence of steps. It is possible that the assumption of the positive definiteness of the Hessian approximations, which we showed is necessary and sufficient to prove superlinear convergence in the line search SR1 method, may not be necessary to prove superlinear convergence for a properly chosen trust region SR1 algorithm.

Appendix.

TABLE A1
List of test functions, numbers, and names.

Number	Dimension	Name
MGH05	2	Beale function
MGH07	2	Helical valley function
MGH09	3	Gaussian function
MGH12	3	Box three-dimensional function
MGH14	3	Wood function
MGH16	4	Brown and Dennis function
MGH18	4	Biggs Exp6 function
MGH20	6	Watson function
MGH21	9	Extended Rosenbrock function
MGH22	10	Extended Powell singular function
MGH23	10	Penalty function I
MGH24	10	Penalty function II
MGH25	10	Variably dimensioned function
MGH26	10	Trigonometric function
MGH35	9	Chebyquad function
CGT01	8	Generalized Rosenbrock function
CGT02	25	Chained Rosenbrock function
CGT04	20	Generalized singular function
CGT05	20	Chained singular function
CGT07	8	Generalized Wood function
CGT08	8	Chained Wood function
CGT10	30	A generalized Broyden tridiagonal function
CGT11	30	Another generalized Broyden tridiagonal function
CGT12	30	Generalized Broyden banded function
CGT14	30	Toint's seven-diagonal generalization of Broyden tridiagonal function
CGT16	30	Trigonometric function
CGT17	8	A generalized Cragg and Levy function
CGT21	30	A generalized Brown function

MGH: problems from Moré, Garbow, and Hillstrom (1981).
CGT: problems from Conn, Gould, and Toint (1988b).

TABLE A2
Iterations and function evaluations—line search.

		BFGS			SR1			
Function	n	itrn.	f-eval	rgx	itrn.	f-eval	rgx	sp
MGH05	2	16	58	0.7E−06	14	52	0.1E−05	1
MGH07	3	26	141	0.4E−05	30	142	0.4E−06	1
MGH09	3	5	34	0.3E−05	3	26	0.2E−07	1
MGH12	3	35	157	0.5E−06	21	99	0.6E−06	1
MGH14	4	32	186	0.7E−05	26	160	0.5E−05	1
MǴH16	4	31	183	0.1E−05	21	133	0.3E−07	1
MGH18	6	43	336	0.2E−05	37	302	0.6E−06	1
MGH20	9	95	1020	0.2E−05	46	532	0.8E−05	1
MGH21	10	34	461	0.9E−05	34	462	0.3E−05	1
MGH22	8	45	464	0.7E−05	36	382	0.4E−05	1
MGH23	10	135	1604	0.9E−05	204	2377	0.6E−05	1
MGH24	10	25	358	0.7E−05	25	362	0.8E−05	1

TABLE A2 (*continued*).

		BFGS			SR1			
Function	*n*	itrn.	f-eval	rgx	itrn.	f-eval	rgx	sp
MGH25	10	16	259	0.7E−06	16	259	0.7E−06	1
MGH26	10	27	374	0.3E−05	27	375	0.2E−05	1
MGH35	9	25	320	0.2E−05	25	320	0.3E−06	1
MGH05	2	47	154	0.3E−07	41	139	0.1E−06	10
MGH07	3	29	136	0.6E−06	38	175	0.4E−07	10
MGH09	3	20	98	0.1E−05	17	102	0.3E−06	10
MGH12	3	66	286	0.5E−05	55	259	0.5E−05	10
MGH14	4	58	316	0.6E−05	69	379	0.1E−06	10
MGH16	4	59	322	0.3E−05	37	212	0.1E−05	10
MGH18	6	45	361	0.3E−05	46	369	0.1E−05	10
MGH20	9	95	1020	0.2E−05	46	532	0.8E−05	10
MGH21	10	57	775	0.3E−05	60	813	0.4E−07	10
MGH22	8	88	977	0.9E−05	67	793	0.3E−05	10
MGH23	10	177	2080	0.9E−05	192	2235	0.9E−05	10
MGH25	10	41	535	0.3E−05	23	337	0.3E−05	10
MGH26	10	72	876	0.7E−05	43	560	0.9E−06	10
MGH07	3	31	174	0.4E−06	23	113	0.6E−07	100
MGH14	4	118	625	0.5E−06	104	567	0.5E−05	100
MGH16	4	89	472	0.2E−05	55	303	0.3E−06	100
MGH20	9	95	1020	0.2E−05	46	532	0.8E−05	100
MGH21	10	158	2185	0.8E−05	154	1906	0.5E−06	100
MGH22	8	129	1227	0.4E−05	90	875	0.9E−05	100
MGH25	10	472	5276	0.1E−04	335	3769	0.1E−04	100
CGT01	8	71	707	0.5E−05	81	843	0.4E−06	1
CGT02	25	36	1315	0.7E−05	43	1505	0.6E−05	1
CGT04	20	85	2049	0.9E−05	49	1291	0.5E−05	1
CGT05	20	311	6797	0.8E−05	180	4055	0.9E−05	1
CGT07	8	129	1273	0.3E−05	116	1132	0.4E−06	1
CGT08	8	141	1348	0.5E−05	140	1347	0.1E−05	1
CGT10	30	58	2328	0.9E−05	40	1770	0.7E−05	1
CGT11	30	37	1686	0.3E−05	32	1526	0.8E−05	1
CGT12	30	264	8734	0.6E−05	199	6734	0.5E−05	1
CGT14	30	70	2699	0.5E−05	100	3640	0.9E−05	1
CGT16	10	11	203	0.4E−05	11	204	0.2E−05	1
CGT17	8	134	1269	0.8E−05	92	892	0.3E−05	1
CGT21	20	12	504	0.2E−05	11	483	0.3E−09	1

TABLE A3

Iterations and function evaluations—trust region.

		BFGS			SR1			
Function	*n*	itrn.	f-eval	rgx	itrn.	f-eval	rgx	sp
MGH05	2	15	57	0.3E−06	16	68	0.5E−05	1
MGH07	3	27	133	0.1E−05	29	150	0.4E−06	1
MGH09	3	5	38	0.3E−05	3	31	0.2E−07	1
MGH12	3	32	150	0.3E−05	26	146	0.8E−05	1
MGH14	4	46	265	0.4E−07	34	247	0.5E−05	1
MGH16	4	33	188	0.1E−05	20	138	0.7E−05	1
MGH18	6	43	341	0.9E−05	40	344	0.8E−05	1
MGH20	9	88	957	0.3E−05	46	584	0.3E−05	1
MGH21	10	42	555	0.2E−05	49	671	0.2E−06	1

TABLE A3 (*continued*).

		BFGS			SR1			
Function	n	itrn.	f-eval	rgx	itrn.	f-eval	rgx	sp
MGH22	8	41	428	0.6E−05	26	294	0.8E−05	1
MGH24	10	24	344	0.2E−05	24	357	0.8E−05	1
MGH25	10	14	236	0.6E−05	14	236	0.6E−05	1
MGH26	10	27	373	0.2E−05	24	349	0.1E−05	1
MGH35	9	24	308	0.4E−05	21	285	0.3E−05	1
MGH05	2	45	160	0.9E−05	36	147	0.9E−06	10
MGH07	3	29	141	0.1E−05	33	171	0.4E−05	10
MGH09	3	21	112	0.8E−05	15	84	0.9E−05	10
MGH12	3	62	292	0.9E−06	19	122	0.7E−05	10
MGH14	4	82	443	0.6E−06	74	467	0.8E−06	10
MGH16	4	59	324	0.5E−06	35	222	0.8E−07	10
MGH18	6	39	323	0.5E−05	51	437	0.6E−07	10
MGH20	9	88	957	0.3E−05	46	584	0.3E−05	10
MGH21	10	63	788	0.3E−05	58	800	0.2E−05	10
MGH22	8	94	913	0.5E−05	56	575	0.8E−05	10
MGH23	10	22	337	0.4E−05	113	1335	0.8E−05	10
MGH24	10	224	2609	0.1E−04	253	3140	0.1E−04	10
MGH25	10	36	488	0.7E−05	25	371	0.3E−05	10
MGH26	10	87	1040	0.7E−05	48	650	0.1E−05	10
MGH07	3	34	158	0.2E−05	22	118	0.2E−05	100
MGH14	4	85	471	0.1E−05	69	426	0.3E−05	100
MGH16	4	89	472	0.4E−06	52	311	0.1E−04	100
MGH20	9	88	957	0.3E−05	46	584	0.3E−05	100
MGH21	10	165	1941	0.2E−05	149	2139	0.3E−06	100
MGH22	8	116	1127	0.8E−05	80	840	0.2E−05	100
CGT01	8	58	584	0.7E−05	80	848	0.8E−05	1
CGT02	25	45	1550	0.4E−05	46	1597	0.2E−05	1
CGT04	20	110	2579	0.3E−05	89	2195	0.5E−05	1
CGT05	20	323	7048	0.5E−05	156	3645	0.8E−05	1
CGT07	8	123	1190	0.4E−05	139	1429	0.3E−06	1
CGT08	8	130	1255	0.9E−05	146	1524	0.5E−05	1
CGT10	30	58	2326	0.9E−05	42	1832	0.7E−05	1
CGT11	30	35	1619	0.3E−05	31	1493	0.5E−05	1
CGT12	30	62	2454	0.8E−05	44	1916	0.5E−05	1
CGT14	30	34	1582	0.8E−05	29	1452	0.5E−05	1
CGT16	10	11	204	0.4E−05	11	206	0.3E−05	1
CGT17	8	83	818	0.9E−05	74	802	0.8E−05	1
CGT21	20	12	504	0.2E−05	11	485	0.3E−09	1

TABLE A4
Testing convergence of $\{B_k\}$ *to* $\nabla^2 f(x_*)$*—line search.*

		BFGS		SR1	
Function	n	itr	$\|H_l - B_l\|/\|H_l\|$	itr	$\|H_l - B_l\|/\|H_l\|$
MGH05	2	19	0.458E−04	16	0.686E−05
MGH07	3	28	0.274E−04	33	0.175E−06
MGH09	3	9	0.918E+00	4	0.918E+00
MGH12	3	38	0.545E−04	24	0.147E−03
MGH14	4	35	0.830E−02	29	0.154E−04
MGH16	4	34	0.928E−01	23	0.348E−04
MGH18	6	47	0.234E+01	40	0.234E+01

TABLE A4 (*continued*).

		BFGS		SR1	
Function	n	itr	$\|H_l - B_l\|/\|H_l\|$	itr	$\|H_l - B_l\|/\|H_l\|$
MGH20	9	175	0.105E+00	100	0.264E−02
MGH21	10	35	0.804E−01	34	0.645E−01
MGH22	8	74	0.161E+01	49	0.160E+01
MGH23	10	178	0.167E+04	215	0.167E+04
MGH24	10	348	0.177E−01	330	0.140E−03
MGH25	10	16	0.748E+04	16	0.748E+04
MGH26	10	31	0.689E−01	31	0.468E−01
MGH35	9	28	0.834E+00	26	0.833E+00
CGT01	8	73	0.393E−01	83	0.144E−01
CGT02	25	43	0.570E−01	50	0.317E−01
CGT04	20	500	0.133E+04	500	0.133E+04
CGT05	20	500	0.582E+03	500	0.503E+03
CGT07	8	138	0.691E−01	124	0.111E−01
CGT08	8	147	0.425E−01	146	0.492E−02
CGT10	30	150	0.134E+03	84	0.185E+03
CGT11	30	44	0.781E−01	37	0.448E−01
CGT12	30	273	0.384E+00	210	0.691E−01
CGT14	30	86	0.279E+00	107	0.303E+00
CGT16	10	18	0.466E−01	16	0.385E−03
CGT17	8	216	0.462E+00	125	0.566E−01
CGT21	20	16	0.124E+01	12	0.120E+01

TABLE A5
Testing convergence of $\{B_k\}$ *to* $\nabla^2 f(x_*)$*—trust region.*

		BFGS		SR1	
Function	n	itr	$\|H_l - B_l\|/\|H_l\|$	itr	$\|H_l - B_l\|/\|H_l\|$
MGH05	2	17	0.235E−02	18	0.102E−05
MGH07	3	30	0.400E−02	31	0.172E−05
MGH09	3	9	0.918E+00	4	0.918E+00
MGH12	3	36	0.396E−02	30	0.473E−02
MGH14	4	47	0.216E−02	41	0.290E−05
MGH16	4	36	0.809E−01	22	0.369E−04
MGH18	6	47	0.234E+01	40	0.234E+01
MGH20	9	157	0.261E−01	99	0.176E−02
MGH21	10	47	0.999E+00	51	0.999E+00
MGH22	8	77	0.277E+01	43	0.276E+01
MGH23	10	500	0.154E+04	149	0.218E+04
MGH24	10	287	0.391E−02	202	0.173E+02
MGH25	10	15	0.103E+05	15	0.103E+05
MGH26	10	31	0.906E−01	28	0.234E−01
MGH35	9	28	0.880E+00	23	0.880E+00
CGT01	8	61	0.110E+00	81	0.275E−01
CGT02	25	51	0.228E+00	50	0.107E+00
CGT04	20	500	0.314E+04	500	0.248E+04
CGT05	20	500	0.104E+04	500	0.671E+03
CGT07	8	122	0.354E−01	138	0.579E−02
CGT08	8	138	0.532E−01	139	0.405E−04
CGT10	30	115	0.109E+03	82	0.112E+03
CGT11	30	40	0.982E−01	34	0.690E−01
CGT12	30	97	0.770E+03	66	0.756E+03

TABLE A5 (*continued*).

		BFGS		SR1	
Function	n	itr	$\|H_l - B_l\|/\|H_l\|$	itr	$\|H_l - B_l\|/\|H_l\|$
CGT14	30	46	0.220E+00	40	0.160E−01
CGT16	10	16	0.523E−01	15	0.298E−02
CGT17	8	200	0.250E+00	123	0.117E−01
CGT21	20	16	0.124E+01	12	0.120E+01

TABLE A6
Testing uniform linear independence of $\{s_k\}$*—line search.*

			No. of steps so that $\sigma_{\min}(\hat{S}_m)* >$							
$f(x)$	n	itr	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
MGH05	2	16	3	2	2	2	2	2	2	2
MGH07	3	33	4	3	3	3	3	3	3	3
MGH09	3	4	*	*	*	*	*	*	*	*
MGH12	3	24	14	5	3	3	3	3	3	3
MGH14	4	29	10	5	5	4	4	4	4	4
MGH16	4	23	6	4	4	4	4	4	4	4
MGH18	6	40	*	*	*	*	*	*	*	*
MGH20	9	100	74	70	67	64	63	62	61	60
MGH21	10	34	*	*	*	*	*	*	*	*
MGH22	8	49	*	*	*	*	*	*	*	*
MGH23	10	215	77	77	77	77	77	77	77	77
MGH24	10	330	79	79	79	79	79	79	79	79
MGH25	10	16	*	*	*	*	*	*	*	*
MGH26	10	31	30	16	10	10	10	10	10	10
MGH35	9	26	*	*	*	*	*	*	*	*
CGT01	8	83	26	15	13	13	13	13	13	13
CGT02	25	50	47	28	25	25	25	25	25	25
CGT04	20	500	87	87	87	87	87	87	87	87
CGT05	20	500	87	87	87	87	87	87	87	87
CGT07	8	124	76	76	76	42	34	34	34	34
CGT08	8	146	45	45	45	45	45	45	45	45
CGT10	30	84	*	*	60	34	30	30	30	30
CGT11	30	37	35	33	30	30	30	30	30	30
CGT12	30	210	98	98	88	88	88	88	88	88
CGT14	30	107	59	36	36	36	36	36	36	36
CGT16	10	16	11	10	10	10	10	10	10	10
CGT17	8	125	67	45	42	34	34	34	34	34
CGT21	20	12	*	*	*	*	*	*	*	*

$* \ \hat{S}_m = [s_l/\|s_l\|, s_{l-1}/\|s_{l-1}\|, \ldots, s_{l-m}/\|s_{l-m}\|]$, where $m \geqq n$.

TABLE A7
Testing uniform linear independence of $\{s_k\}$*—trust region.*

			No. of steps so that $\sigma_{\min}(\hat{S}_m)*>$							
$f(x)$	n	itr	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
MGH05	2	18	3	2	2	2	2	2	2	2
MGH07	3	31	5	3	3	3	3	3	3	3
MGH09	3	4	*	*	*	*	*	*	*	*

TABLE A7 (*continued*).

$f(x)$	n	itr	No. of steps so that $\sigma_{\min}(\hat{S}_m)^*>$							
			10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
MGH12	3	30	7	6	5	3	3	3	3	3
MGH14	4	41	8	5	4	4	4	4	4	4
MGH16	4	22	5	4	4	4	4	4	4	4
MGH18	6	40	*	*	*	*	*	*	*	*
MGH20	9	99	75	64	63	62	62	61	61	61
MGH21	10	51	*	*	*	*	*	*	*	*
MGH22	8	43	*	*	*	*	*	*	*	*
MGH23	10	149	77	77	77	77	77	77	77	77
MGH24	10	202	79	79	79	74	74	74	74	74
MGH25	10	15	*	*	*	*	*	*	*	*
MGH26	10	28	26	18	10	10	10	10	10	10
MGH35	9	23	*	*	*	*	*	*	*	*
CGT01	8	81	32	17	13	12	12	12	12	12
CGT02	25	50	*	29	26	25	25	25	25	25
CGT04	20	500	88	88	88	88	88	88	88	88
CGT05	20	500	88	87	87	87	87	87	87	87
CGT07	8	138	76	76	50	43	41	41	41	41
CGT08	8	139	41	41	41	41	41	41	41	41
CGT10	30	82	*	*	59	36	32	30	30	30
CGT11	30	34	*	31	30	30	30	30	30	30
CGT12	30	66	*	*	*	60	40	31	30	30
CGT14	30	40	*	33	30	30	30	30	30	30
CGT16	10	15	12	10	10	10	10	10	10	10
CGT17	8	123	73	49	39	34	33	33	33	33
CGT21	20	12	*	*	*	*	*	*	*	*

* $\hat{S}_m=[s_l/\|s_l\|,\ s_{l-1}/\|s_{l-1}\|, \ldots, s_{l-m}/\|s_{l-m}\|]$, where $m \geqq n$.

TABLE A8
Testing positive definiteness—line search.

$f(x)$	n	itr	0: Indefinite; 1: Positive definite	%pd	1*	2*
MGH05	2	14	1111111111111	1.00	13	1
MGH07	3	30	11111101111011110111111111111	0.90	12	1
MGH09	3	3	11	1.00	2	1
MGH12	3	21	11111111111111111111	1.00	20	1
MGH14	4	26	1111111101111110111110111	0.88	3	1
MGH16	4	21	10111111111111111111	0.95	18	1
MGH18	6	37	111111100111111111111111011111111111	0.92	11	1
MGH20	9	46	1111011111111111011111011101101111110 11111011	0.84	2	1
MGH21	10	34	111011111110111101001111111111111	0.85	13	1
MGH22	8	36	11111101011111111111111110111111111	0.91	9	1
MGH23	10	204	111111111111111111101111111111101111 111011101101101001101001111011110111 111111011010001111100111111101110011 111101011111101111010100110101111110 111101101111111010011011101111011001 11111011111101111110111	0.77	3	0
MGH24	10	25	111111101110111110111111	0.88	6	1
MGH25	10	16	111111111111111	1.00	15	0
MGH26	10	27	11101110111011101101110111	0.77	3	1

TABLE A8 (*continued*).

$f(x)$	n	itr	0: Indefinite; 1: Positive definite	%pd	1*	2*
MGH35	9	25	1111101101111110111111111	0.88	9	1
CGT01	8	81	111111110011010011110101101111110100 110111111011011101100110111011111011 11111111	0.75	10	1
CGT02	25	43	111111110011111110011011011011011111 111111	0.81	11	1
CGT04	20	49	111111111011111110111111101111111111 11111111111	0.94	22	1
CGT05	20	180	1111111110111110111111111111101110111 111111111111110101111011111111110111 111111110111011010001110111111101111 111111110101111110110111111001110111 1111111111110111111111111111111111111	0.87	21	1
CGT07	8	116	11111111111111110111111101000011011 010010011111101011010011011101111011 011111111111111101111011110110111111 1111111	0.78	13	1
CGT08	8	140	111111110110111110111011111101101101 111110011011111101101110011011110100 110110000000011110111111001110100111 11101100110100110111110101111111	0.70	6	1
CGT10	30	40	111111111111111111111111101111111111 001	0.92	1	1
CGT11	30	32	11110111011111111101110111111111	0.87	8	1
CGT12	30	199	111111111101111111101101111011111111 111110110111110111011101110111110111 011111111110111011111101100111111010 1100111111111111010101101111111101011 101111110011111011111110110011011111 11010101110111101	0.80	1	1
CTG14	30	100	111010111110111011101110011110110111 111111101110111101101111111010101111 111111111111111011111111111111	0.83	12	1
CGT16	10	11	1111111111	1.00	10	1
CGT17	8	92	111110111111111110111111011110110111 011111100111111111101111101111111101 11111111101111111111	0.87	9	1
CGT21	20	11	1110101111	0.80	4	1

1*: Number of consecutive iterations where B_k was positive definite immediately prior to the termination of the algorithm.

2*: Number of iterations where the SR1 update is skipped because condition (4.1) was violated.

REFERENCES

C. G. BROYDEN, J. E. DENNIS, JR., AND J. J. MORÉ (1973), *On the local and superlinear convergence of quasi-Newton methods*, J. Inst. Math. Appl., 12, pp. 223-246.

A. R. CONN, N. I. M. GOULD, AND PH. TOINT (1988a), *Global convergence of a class of trust region algorithms for optimization with simple bounds*, SIAM J. Numer. Anal., 25, pp. 433-460. (Also in SIAM J. Numer. Anal., 26 (1989), pp. 764-767.)

——— (1988b), *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comp., 50, pp. 399-430.

——— (1991), *Convergence of quasi-Newton matrices generated by the symmetric rank one update*, Math. Programming, 50, pp. 177-195.

J. E. DENNIS, JR. AND J. J. MORÉ (1974), *A characterization of superlinear convergence and its application to quasi-Newton methods*, Math. Comp., 28, pp. 549-560.

——— (1977), *Quasi-Newton methods, motivation and theory*, SIAM Rev., 19, pp. 46-89.

J. E. DENNIS, JR. AND R. B. SCHNABEL (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ.

A. V. FIACCO AND G. P. MCCORMICK (1968), *Nonlinear Programming*, John Wiley and Sons, New York.

R. FLETCHER (1980), *Practical Methods of Optimization*, John Wiley and Sons, New York.

R.-P. GE AND M. J. D. POWELL (1983), *The convergence of variable metric matrices in unconstrained optimization*, Math. Programming, 27, pp. 123-143.

P. E. GILL, W. MURRAY, AND M. H. WRIGHT (1981), *Practical Optimization*, Academic Press, London.

D. GOLDFARB (1970), *A family of variable metric methods derived by variational means*, Math. Comp., 24, pp. 23-26.

M. D. HEBDEN (1973), *An algorithm for minimization using exact second derivatives*, Report TP515, A.E.R.E., Harwell, England.

J. J. MORÉ (1977), *The Levenberg-Marquardt algorithm: Implementation and theory*, in Numerical Analysis, G. A. Watson, ed., Lecture Notes in Math. 630, Springer-Verlag, Berlin, pp. 105-116.

J. J. MORÉ, B. S. GARBOW, AND K. E. HILLSTROM (1981), *Testing unconstrained optimization software*, ACM Trans. Math. Software, 7, pp. 17-41.

J. J. MORÉ AND D. C. SORENSEN (1983), *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4, pp. 553-572.

M J. D. POWELL (1976), *Some global convergence properties of a variable metric algorithm for minimization without exact line searches*, in Nonlinear Programming, SIAM-AMS Proceedings, Vol. IX, R. W. Cottle and C. E. Lemke, eds., Society for Industrial and Applied Mathematics, Philadelphia.

R. B. SCHNABEL, J. E. KOONTZ, AND B. E. WEISS (1982), *A modular system of algorithms for unconstrained minimization*, ACM Trans. Math. Software, 11, pp. 419-440.

SIAM J. OPTIMIZATION
Vol. 3, No. 1, pp. 25–42, February 1993

A CUTTING PLANE APPROACH TO THE SEQUENTIAL ORDERING PROBLEM (WITH APPLICATIONS TO JOB SCHEDULING IN MANUFACTURING)*

N. ASCHEUER†, L. F. ESCUDERO‡, M. GRÖTSCHEL†, AND M. STOER†

Abstract. The sequential ordering problem (SOP) finds a minimum cost Hamiltonian path subject to certain precedence constraints. The SOP has a number of practical applications and arises, for instance, in production planning for flexible manufacturing systems. This paper presents several 0–1 models of the SOP and reports the authors' computational experience in finding lower bounds of the optimal solution value of several real-life instances of SOP. One of the most successful approaches is a cutting plane procedure that is based on polynomial time separation algorithms for large classes of valid inequalities for the associated polyhedron.

Key words. traveling salesman problem, sequential ordering problem, linear ordering problem, precedence constraints, cutting plane algorithm, separation algorithm, polyhedral combinatorics

AMS(MOS) subject classifications. 90C10, 90C35

1. Introduction and problem definition. Problems for the flexible manufacturing systems we are considering (see, e.g., [6]) can be phrased in graph theoretical terminology in the following way. We are given a directed or undirected graph where an arc or edge represents the possibility of performing two tasks consecutively and where a (e.g., transportation or set-up) cost is incurred by changing from one task to another. In addition, some precedence relations are given that specify that some tasks have to be executed before certain others. The problem is to schedule all jobs at minimum cost, i.e., to find a feasible Hamiltonian path, say $\mathcal{H}$ of minimum cost, where $\mathcal{H}$ is called *feasible* if it does not violate the precedence constraints.

In this paper (and in the real application that motivated this work) the given graph is the complete directed graph $D_n = (V, A_n)$ on n nodes. (An application, where the given graph is undirected, can be found in [22].) We denote an arc going from some node i to another node j by (i, j) and the associated cost by c_{ij}. The precedence constraints are given by a digraph $P = (V, R)$, on the same node set V as D_n, where an arc $(i, j) \in R$ means that task i has to be performed before task j. Clearly, this *precedence digraph* P has to be acyclic (i.e., may not contain a directed cycle). Moreover, if (i, j), $(j, k) \in R$ then k cannot be performed before i; in other words, we can also assume that P is transitively closed.

So the precedence constraints are given by an acyclic and transitively closed digraph $P = (V, R)$. Using this notation we call a Hamiltonian path in D_n *feasible* if $(j, i) \notin R$ holds for all $i \prec j$, where $i \prec j$ means that there is a directed path from node i to node j in the Hamiltonian path.

Now we can state the *sequential ordering problem* (SOP) formally. Given a complete digraph $D_n = (V, A_n)$ with costs c_{ij} for all $(i, j) \in A_n$ and a transitively closed acyclic digraph $P = (V, R)$, find a feasible Hamiltonian path $\mathcal{H}$ in D_n that has minimum cost.

If the precedence digraph $P = (V, R)$ has empty arc set, the SOP reduces to finding a minimum cost Hamiltonian path in D_n. This is an NP-hard problem and so is the SOP. Our main concern here, though, is not an algorithm for the "pure" Hamiltonian

* Received by the editors December 26, 1990; accepted for publication (in revised form) November 13, 1991.

† Institut für Mathematik, Universität Augsburg, Germany.

‡ Facultad de Matematicas, Universidad Complutense de Madrid, Spain.

path problem (or, equivalently, the asymmetric traveling salesman problem, ATSP) but a method that deals with precedences.

This paper is organized as follows. In § 2 we present three different 0–1 models of the SOP, in particular, some classes of inequalities valid for the associated polyhedra. Polynomial time *separation algorithms* for some of these classes are described in § 3. Further classes of valid inequalities are discussed in § 4. In § 5 we present some preprocessing procedures for our cutting plane algorithm that help reduce the instance sizes. The implementations of the cutting plane algorithm are outlined in § 6; our computational results are reported in § 7.

2. 0–1 Models. The SOP, in the form stated here, seems to have been formulated for the first time in [4]. The aim of [4] and the subsequent paper [5] was the design of a heuristic that performs well in practice with respect to running time and solution quality. It was decided, however, to analyze the quality performance of the heuristic before using its implementation in a production planning system.

Before describing the 0–1 model of the SOP introduced in [4], we introduce the following notation.

Let $D_n = (V, A_n)$ be the complete digraph of n nodes and let $P = (V, R)$ be a transitively closed, acyclic subdigraph of D_n. We set

$$\overleftarrow{R} := \{(j, i) \in V \times V \mid (i, j) \in R\}, \tag{2.1a}$$

$$\overrightarrow{R} := \{(i, k) \in V \times V \mid \exists j \text{ with } (i, j), (j, k) \in R\}, \tag{2.1b}$$

$$A := A_n \backslash (\overrightarrow{R} \cup \overleftarrow{R}). \tag{2.1c}$$

Note that a feasible Hamiltonian path can contain neither an arc from $\overleftarrow{R}$ nor an arc from $\overrightarrow{R}$, while for each arc in A there is some feasible Hamiltonian path containing this arc. We thus call A the *feasible arc set* and $D = (V, A)$ the *feasible subdigraph* of D_n. Furthermore, set

$$\alpha_k := 1 + |\{i \mid (i, k) \in R\}|, \qquad k \in V, \tag{2.2a}$$

$$\beta_k := n - |\{j \mid (k, j) \in R\}|, \qquad k \in V. \tag{2.2b}$$

It is clear that $\alpha_k - 1$ (respectively, $n - \beta_k$) is the minimum number of predecessor (respectively, successor) nodes for node k in any feasible Hamiltonian path.

Let us introduce the following two types of variables. For each arc $(i, j) \in A$, x_{ij} is a 0–1 variable that indicates whether (i, j) is in the Hamiltonian path (i.e., $x_{ij} = 1$) or not. (We do not need variables for the arcs from $A_n \backslash A$.) The second type are 0–1 variables ξ_{kh} for $k, h \in V$, which are auxiliary variables that help to model the precedence constraints, such that $\xi_{kh} = 1$ means that node k is to be sequenced at level h for $\alpha_k \leqq h \leqq \beta_k$ and, otherwise, zero.

To obtain a compact formulation we introduce further terminology. If F is a subset of A we abbreviate the sum $\sum_{(i,j) \in F} x_{ij}$ by $x(F)$. If W is a subset of V then $A(W) = \{(i, j) \in A \mid i, j \in W\}$. If $j \in V$ then $\delta^+(j) = \{(j, k) \in A\}$ and $\delta^-(j) = \{(i, j) \in A\}$, and if, moreover, $W \subseteq V \backslash \{j\}$ then $(j : W) = \{(j, k) \in A \mid k \in W\}$ and $(W : j) = \{(i, j) \in A \mid i \in W\}$. Let us now assume that, in addition to $D_n = (V, A_n)$ and $P = (V, R)$, costs $c_{ij} \in \mathbb{R}$ for all $(i, j) \in A$ are given.

The model introduced in [4] is as follows.

$$\lambda^* = \min c^t x \quad \text{subject to} \tag{2.3}$$

(1) $\quad x(A) = n - 1,$

(2) $\quad x(\delta^-(j)) \leqq 1 \quad \text{for all } j \in V,$

(3) $x(\delta^+(j)) \leqq 1$ for all $j \in V$,

(4) $x_{ij} \geqq 0$ for all $(i,j) \in A$,

(5) $x(A(W)) \leqq |W| - 1$ for all $W \subset V$, $2 \leqq |W| \leqq n-1$,

(6) $x_{ij} \in \{0; 1\}$ for all $(i,j) \in A$,

(7) $\sum_{k|\alpha_k \leqq h \leqq \beta_k} \xi_{kh} = 1$ for all $h \in V$,

(8) $\sum_{\alpha_k \leqq h \leqq \beta_k} \xi_{kh} = 1$ for all $k \in V$,

(9) $\sum_{\alpha_i \leqq h_i \leqq \beta_i} h_i \xi_{ih_i} + 1 \leqq \sum_{\alpha_j \leqq h_j \leqq \beta_j} h_j \xi_{jh_j}$ for all $(i,j) \in R \setminus \vec{R}$,

(10) $\xi_{ih} + \xi_{jh+1} \leqq 1$ for all $(i,j) \in A_n \setminus A$, $\max\{\alpha_i, \alpha_j - 1\} \leqq h \leqq \min\{\beta_i, \beta_j - 1\}$,

(11) $\xi_{kh} \in \{0; 1\}$ for all $k \in V$, $\alpha_k \leqq h \leqq \beta_k$,

(12) $\xi_{ih} + \xi_{jh+1} \leqq 1 + x_{ij}$ for all $(i,j) \in A$, $\max\{\alpha_i, \alpha_j - 1\} \leqq h \leqq \min\{\beta_i, \beta_j - 1\}$.

We briefly indicate the logic of the model. In analogy to the well-known 0–1 model of the ATSP, constraints (1)–(6) provide an IP-formulation of the Hamiltonian path problem in $D = (V, A)$. Inequalities (5) are called, as usual, *subtour elimination constraints* (SECs).

Constraints (7)–(12) ensure that the given precedence constraints are observed. Constraints (7) (respectively, constraints (8)) force one node (respectively, level) per level (respectively, node). Constraints (9) prevent reverse sequencing for pairs of nodes that are linked by *direct* precedence relationships. Constraints (10) prevent illegal immediate sequencings. Finally, constraints (12) are the so-called *linking constraints* that integrate submodels (1)–(6) and (7)–(11).

Clearly, there are a number of model improvements possible, e.g., turning some of the inequalities into equalities, etc., but we state here only the basic model.

The computational experience with this model reported in [4] and [5] was unsatisfactory with respect to the integrality gap, i.e., in a number of cases the relative deviation $(\lambda^H - \lambda_{LR})/\lambda_{LR}$ was rather large, where λ^H gives the cost of the solution found by the heuristic algorithm, and λ_{LR} is the lower bound of the optimal value λ^* obtained from the (restricted) Lagrangian relaxation of model (2.3) used in [4]. Such a gap has one of the following causes. Either λ^H or λ_{LR}, or both, are far away from λ^*. The belief was that λ^H was good and λ_{LR} poor. This belief motivated the introduction and investigation of further 0–1 models of the SOP, which is the subject of the rest of the paper.

The first new model requires two types of variables. The first type are 0–1 variables x_{ij} with the same meaning as before. The second type are real variables y_{ij} for all $(i,j) \in A_n$, which are auxiliary variables that help to model the precedence constraints.

Our first new model of the SOP is as follows.

(2.4) $\lambda^* = \min c^t x$ subject to x satisfies (2.3) (1)–(6) and

(7) $y_{ij} = 1$ for all $(i,j) \in R$,

(8) $y_{ij} + y_{ji} = 1$ for all $(i,j) \in A_n$,

(9) $y_{ij} + y_{jk} + y_{ki} \leqq 2$ for all $i, j, k \in V$, $i \neq j \neq k$,

(10) $y_{ij} \geqq 0$ for all $(i,j) \in A_n$,

(11) $x_{ij} - y_{ij} \leqq 0$ for all $(i,j) \in A$.

If we add integrality constraints to (8)–(10), we obtain a well-known 0–1 formulation of the linear ordering problem; see [10]. In our case, integrality stipulations for the y_{ij}'s are not needed, since integrality of the x_{ij}'s implies integrality of the y_{ij}'s via (11). Constraints (7)–(11) ensure that the given precedence constraints are observed. Clearly, there are a number of model improvements possible, e.g., we can also skip some of the variables y_{ij}'s, turn some of the inequalities to equalities (see § 5), etc. These obvious modifications have been done in our implementation. We state here only the basic model for notational ease.

A nice feature of model (2.4) is that it combines two well-known combinatorial optimization problems in a natural way. Looking at this model we can say that the SOP is the Hamiltonian path problem plus the linear ordering problem integrated through the *linking constraints* (11). An obvious disadvantage of this model is the use of the auxiliary variables y_{ij}'s. In fact, we can get rid of these by replacing (7)–(11) by a new class of constraints of size exponential in n.

Our second new model of the SOP is as follows.

$$(2.5) \qquad \lambda^* = \min c^t x \quad \text{subject to} \quad x \text{ satisfies (1)–(6) and}$$

$$(12) \qquad x((j: W)) + x(A(W)) + x((W: i)) \leqq |W|$$

$$\text{for all } (i, j) \in R \quad \text{and all } \emptyset \neq W \subseteq V \setminus \{i, j\}.$$

We call the inequalities (12) *precedence forcing constraints* (PFCs). It is obvious that every feasible solution of (1)–(6) and (12) is the incidence vector of a feasible Hamiltonian path and vice versa.

Although model (2.4) provides a nice interpretation of the SOP as a combination of two other well-known problems, our computational experience (see below) shows that model (2.5) is a more natural setting for the SOP, given the type of separation algorithms that we propose.

Both models give rise to polyhedra associated with the SOP. We only introduce here the one arising from (2.5). Let $D_n = (V, A_n)$ be the complete digraph on n nodes, let $P = (V, R)$ be a transitively closed acyclic subdigraph of D_n, $A := A_n \setminus (\vec{R} \cup \overleftarrow{R})$, and set

$$(2.6) \qquad \text{SOP}(n, P) := \text{conv}\{x \in \mathbb{R}^A \mid x \text{ satisfies (1)–(6), (12)}\}.$$

SOP (n, P) is called the *sequential ordering polytope* associated with D_n and P, since every point that satisfies (1)–(6) and (12) is an incidence vector of a feasible Hamiltonian path, i.e., a feasible solution of the SOP. The study of the structure of this polytope (dimensions, facets, etc.) is of course of particular interest for the solution of the SOP. It is clearly closely related to the study of the ATSP polytope; see [16]. The scope of the present paper is, however, computational and there is no space here to discuss even some of the basic polyhedral facts about SOP (n, P).

3. Separation algorithms. The cutting plane algorithms we are going to describe follow the standard scheme described in, e.g., [3], [8]–[10], [13], [16], [20], and [21]. One of the main ingredients of such an algorithm are routines that check whether a given point (usually the optimum solution of the last LP relaxation solved) satisfies all inequalities of some given class of constraints and, if not, output at least one inequality of this class violated by the given point.

Such procedures are called *separation algorithms*; see [11] for some theory behind this approach. Of course, we are interested in separation algorithms that run in polynomial time.

In this section we describe polynomial time separation algorithms for the subtour elimination constraints (SECs) (2.3) (5) and the precedence forcing constraint (PFCs) (2.5) (12); see also [1]. Note that both classes contain a number of inequalities that is exponential in n. (Note also that constraints (1)-(6) of model (2.3) are inherited by models (2.4) and (2.5).)

We begin with the SECs. The input of our separation algorithm is a point $z \in \mathbb{Q}^A$. We assume that $z_{ij} \geqq 0$ for all $(i, j) \in A$; we do not require that z satisfies constraints (1)-(3) of (2.3), i.e., our algorithm will handle more general situations than those arising in models (2.3), (2.4), and (2.5). The output of the algorithm provides either the statement that z satisfies all inequalities

$$(3.1) \qquad x(A(W)) \leqq |W| - 1 \quad \text{for all } W \subseteq V, \quad 2 \leqq |W| \leqq n,$$

or it provides a node set $W \subseteq V$, $2 \leqq |W| \leqq n$ such that $z(A(W)) > |W| - 1$. In fact (this will be clear from the description of the algorithm), we can even find a node set W such that $z(A(W)) - |W| + 1$ is as large as possible, i.e., a *most violated* SEC can be identified.

For this purpose we construct a (first) auxiliary digraph $D_0 = (V_0, A_0)$ as follows.

$$(3.2a) \qquad V_0 = V \cup \{0\}, \quad \text{where 0 is a new node,}$$

$$(3.2b) \qquad A^z := \{(i, j) \in A \mid z_{ij} > 0\},$$

$$(3.2c) \qquad A_0 := A^z \cup \{(0, v) \mid v \in V\} \cup \{(j, i) \mid (i, j) \in A^z \text{ and } (j, i) \notin A^z\}.$$

In other words, we make $D^z = (V, A^z)$ symmetric by reversing arcs and add a source zero that is linked to all nodes in D^z. We solve the separation problem for the SECs (5) by reducing it to a sequence of min-cut problems. To do this we introduce (auxiliary) capacities c^0_{ij} for the arcs of D_0 in the following way. First, we set

$$(3.3) \qquad \zeta_j = z(\delta^-(j)) + z(\delta^+(j)) \quad \text{for all } j \in V$$

and we define the capacities c^0_{ij} by

$$(3.4) \qquad c^0_{0j} := 1 - \tfrac{1}{2}\zeta_j + M \quad \text{for all } j \in V,$$

where M is a positive number chosen such that $c^0_{0j} \geqq 0$ for all $j \in V$. Furthermore, we set

$$(3.5) \qquad c^0_{ij} := c^0_{ji} := \tfrac{1}{2}(z_{ij} + z_{ji}) \quad \text{for all } (i, j) \in A^z.$$

(In case $(j, i) \notin A^z$ for some $(i, j) \in A^z$ we assume z_{ij} to have value zero.)

Now we introduce n further auxiliary digraphs that are slight modifications of D_0 as follows. For every $k \in V$ we define a digraph $D_k = (V_k, A_k)$ with capacities c^k_{ij} by setting

$$(3.6a) \qquad V_k := V_0,$$

$$(3.6b) \qquad A_k := A_0 \dot{\cup} B_k \quad \text{where } B_k = \{(v, k) \mid v \in V \setminus \{k\}\},$$

$$(3.6c) \qquad c^k_{ij} := c^0_{ij} \quad \text{for all } (i, j) \in A_0,$$

$$(3.6d) \qquad c^k_{vk} := M \quad \text{for all } (v, k) \in B_k.$$

(In (b) above $\dot{\cup}$ means disjoint union, i.e., if A_0 contains an arc from B_k, we add a parallel one.)

(3.7) SEPARATION ALGORITHM FOR THE SUBTOUR ELIMINATION CONSTRAINTS.
Input. A point $z \in \mathbb{Q}^A$ satisfying $z_{ij} \geqq 0$.

Output. At least one node set of cardinality between 2 and n, such that the corresponding SEC is violated by z, or the information that no such node set exists.

```
For each $k \in V$ do:
  1. Construct the digraph $D_k = (V_k, A_k)$ with capacities $c_{ij}^k$ as outlined before; see (3.6).
  2. Use a max-flow algorithm to determine a $(0, k)$-cut $\delta^-(W_k)$ in $D_k$ (i.e., a cut separating 0 and $k$ such that $k \in W_k$, $0 \notin W_k$), so that its capacity $c^k(\delta^-(W_k))$ is as small as possible.
  3. If $c^k(\delta^-(W_k)) < nM + 1$ then $x(A(W_k)) \leqq |W_k| - 1$ is a SEC violated by $z$.
End For
```

If the above procedure does not output a violated constraint then z satisfies all SECs.

LEMMA 3.8. *If, for all $k \in V$, the minimum capacity of a $(0, k)$-cut in D_k is not smaller than $nM + 1$, then z satisfies all inequalities $z(A(W)) \leqq |W| - 1$, $W \subseteq V$, $2 \leqq |W| \leqq n$. If, for some $k \in V$, there is a $(0, k)$-cut $\delta^-(W_k)$, $W_k \subseteq V$, $2 \leqq |W_k| \leqq n$ with $c^k(\delta^-(W_k)) < nM + 1$, then $z(A(W_k)) > |W_k| - 1$.*

Proof. The capacity $c^k(\delta^-(W_k))$ of any cut $\delta^-(W_k)$ in D_k with $0 \notin W_k$, $k \in W_k$ is nothing but $|W_k| - z(A(W_k)) + nM$. This can be seen as follows.

$$
\begin{aligned}
c^k(\delta^-(W_k)) &= \sum_{w \in W_k} c_{0w}^k + \sum_{v \in V \setminus W_k} \; \sum_{w \in W_k | (v,w) \in A_0} c_{vw}^k + \sum_{v \in V \setminus W_k | (v,k) \in B_k} c_{vk}^k \\
&= \sum_{w \in W_k} c_{0w}^0 + \sum_{v \in V \setminus W_k} \; \sum_{w \in W_k | (v,w) \in A_0} c_{vw}^0 + \sum_{v \in V \setminus W_k | (v,k) \in B_k} c_{vk}^k \\
&= \left(|W_k| - \frac{1}{2} \sum_{w \in W_k} \zeta_w + M |W_k| \right) + \frac{1}{2} \sum_{(v,w) \in \delta^-(W_k)} (z_{vw} + z_{wv}) + |V \setminus W_k| M \\
&= nM + |W_k| - \frac{1}{2} \left(\sum_{(v,w) \in \delta^-(W_k)} (z_{vw} + z_{wv}) + 2 \sum_{(v,w) \in A(W_k)} (z_{vw} + z_{wv}) \right) \\
&\quad + \frac{1}{2} \sum_{(v,w) \in \delta^-(W_k)} (z_{vw} + z_{wv}) \\
&= |W_k| - z(A(W_k)) + nM.
\end{aligned}
$$

Therefore, $z(A(W_k)) \leqq |W_k| - 1$ holds if and only if $c^k(\delta^-(W_k)) \geqq 1 + nM$ holds.

If there is a cut $\delta^-(W_k)$ with $c^k(\delta^-(W_k)) < 1 + nM$ we still have to show that $|W_k| \geqq 2$. But this is obvious. Since $k \in W_k$, $|W_k| \geqq 1$. If $W_k = \{k\}$ then $c^k(\delta^-(k)) = 1 + nM$. And therefore, $c^k(\delta^-(W_k)) < 1 + nM$ implies $|W_k| \geqq 2$. Finally, note that by construction $W_k = V$ is a possible solution. □

Remark 3.9. The separation algorithm for the SEC (3.7) (plus nonnegativity constraints) can be solved by calling n times a max-flow algorithm and is thus solvable in polynomial time.

For the best running time of max-flow algorithms currently known, consult the survey article [2].

Algorithm (3.7) handles a more general situation than we need in the present application. If we assume that the given vector $z \in \mathbb{Q}^A$ satisfies the cardinality constraint (2.3) (1) in addition to the nonnegativity constraints (2.3) (4) then the node set whose related SEC is violated is such that $|W_k| \leqq n - 1$. To see this, note that $c^k(\delta^-(V)) = |V| - z(A(V)) + nM = 1 + nM$ and so $W_k \subset V$ from Lemma 3.8.

Conversely, if we assume that the given vector $z \in \mathbb{Q}^A$ satisfies the star constraints (2.3) (2) and (3) we can reduce the separation problem to a min-cut problem in an undirected graph (see [1]) and therefore apply the Gomory-Hu algorithm or any other efficient algorithm to compute a minimum capacity cut. We outline this further reduction briefly. Note that if z satisfies (2.3) (2) and (3), then $\zeta_j \leqq 2$ (see (3.3)) for all $j \in V$ and thus the number M needed in (3.4) can be chosen as zero. This implies that the arc sets B_k introduced in (3.6) and thus the auxiliary graphs D_k, $k \in V$, are not needed. Moreover, we can symmetrize D_0 to a digraph $\tilde{D} = (V_0, \tilde{A})$ with capacities $\tilde{c}_{ij}$ by setting

(3.10a) $\tilde{A} := A_0 \cup \{(v, 0) \mid v \in V\}$,

(3.10b) $\tilde{c}_{ij} := \tilde{c}_{ji} := c^0_{ij} = c^0_{ji} = \frac{1}{2}(z_{ij} + z_{ji})$ for all $(i, j) \in A_0$,

(3.10c) $\tilde{c}_{v0} := \tilde{c}_{0v}$ for all $v \in V$.

Let $G = (V_0, E)$ be the undirected graph underlying $\tilde{D} = (V_0, \tilde{A})$ with capacities $\hat{c}_{ij}$ defined by

(3.11a) $E = \{ij \mid (i, j) \in \tilde{A}\}$,

(3.11b) $\hat{c}_{ij} := \tilde{c}_{ij} = \tilde{c}_{ji}$ for all $ij \in E$.

G has the property that $\hat{c}(\delta(W)) = \tilde{c}(\delta^-(W)) = \tilde{c}(\delta^+(W))$ for any $W \subseteq V$, where $\delta^+(W) = \delta^-(V \setminus W)$. Thus a cut $\delta(W)$ in G with capacity $\hat{c}(\delta(W))$ as small as possible corresponds to a minimum capacity cut $\delta^-(W)$ in $\tilde{D}$ and vice versa.

This construction shows that the separation problem for the SECs—under the assumption that the given point satisfies (2.3) (2)-(4)—can be solved by any algorithm that determines a minimum capacity cut in an undirected graph.

Although the worst case complexity of algorithm (3.7) and the method outlined above are about the same, the latter approach works much better in practice, at least if one uses the method described in [18], as we did.

Let us now turn our attention to the *precedence forcing constraints*, the PFCs,

(3.12) $x((j : W)) + x(A(W)) + x((W : i)) \leqq |W|$

for all $(i, j) \in R$ and all $\emptyset \neq W \subseteq V \setminus \{i, j\}$.

As above, we reduce the separation problem for (3.12) to a series of min-cut problems.

We assume that a point $z \in \mathbb{Q}^A$ satisfying $z_{ij} \geqq 0$ for all $(i, j) \in A$ is given and we want to find an inequality of (3.12) that is violated by z, if one exists. We do this by constructing, for each arc $(i, j) \in R$, a min-cut problem that proves whether or not (3.12) is satisfied for all $\emptyset \neq W \subseteq V \setminus \{i, j\}$.

For every arc $(i, j) \in R$ of the precedence digraph $P = (V, R)$, we introduce a new digraph $D_{ij} = (V_{ij}, A_{ij})$ with capacities d^{ij} as follows.

(3.13a) $V_{ij} := (V \setminus \{i, j\}) \cup \{v_{ij}\}$ where v_{ij} is a new node,

(3.13b) $A^z := \{(i, j) \in A \mid z_{ij} > 0\}$,

(3.13c) $A_{ij} = \{(k, l) \mid (k, l) \in A^z, k, l \notin \{i, j\}\} \cup \{(v_{ij}, l) \mid (j, l) \in A^z, l \notin \{i, j\}\}$

$\cup \{(k, v_{ij}) \mid (k, i) \in A^z, k \notin \{i, j\}\}$,

(3.13d) $d^{ij}_{kl} := z_{kl}$ for all $(k, l) \in A_{ij} \cap A^z$,

(3.13e) $d^{ij}_{v_{ij}l} := z_{jl}$ for all $(j, l) \in A^z$,

(3.13f) $d^{ij}_{kv_{ij}} := z_{ki}$ for all $(k, i) \in A^z$.

See Fig. 1 for an illustration. Observe that D_{ij} is obtained from $D^z = (V, A^z)$ by deleting all arcs directed into j, all arcs leaving i, all arcs between i and j, and by identifying the nodes i and j. The capacities are just the values of the (positive) components of z.

The PFCs concerning $(i, j) \in R$ and D,

(3.14) $x((j : W)) + x(A(W)) + x((W : i)) \leqq |W|$,

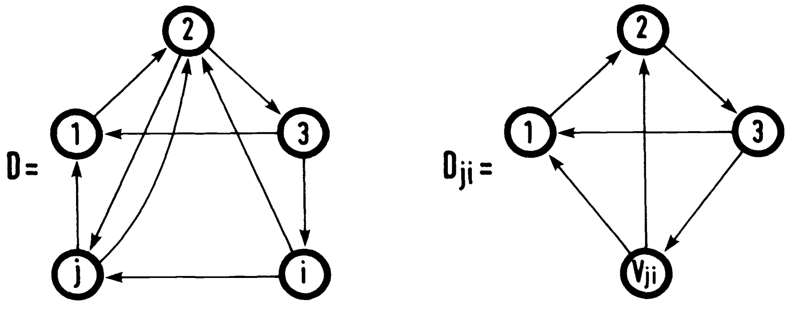


FIG. 1. *Original and shrinking graphs of precedence relationships.*

can be written using this transformation in the form

$$x(A(W\cup\{v_{ij}\}))\leqq|W|=|W\cup\{v_{ij}\}|-1 \tag{3.15}$$

with respect to the digraph D_{ij}. In other words, to check the PFCs concerning $(i,j)\in R$, we have to determine whether the SECs

$$x(A(\tilde{W}))\leqq|\tilde{W}|-1 \quad \text{for all } \tilde{W}\subseteq V_{ij},\ v_{ij}\in\tilde{W} \quad \text{and} \quad 2\leqq|\tilde{W}|\leqq n-1 \tag{3.16}$$

for D_{ij} are satisfied by z. (Recall $n=|V|$ and then $n-1=|V_{ij}|$.) If we can determine a node set $\tilde{W}\subseteq V_{ij}$ with $v_{ij}\in\tilde{W}$, $2\leqq|\tilde{W}|\leqq n-1$, such that $z(A(\tilde{W}))>|\tilde{W}|-1$ then, for $W:=\tilde{W}\backslash\{v_{ij}\}$, $z((j:W))+z(A(W))+z((W:i))>|W|$ obviously holds. If no such $\tilde{W}$ exists, all inequalities (3.14) concerning $(i,j)\in R$ are satisfied.

By repeating this procedure for all $(i,j)\in R$ we can solve the separation problem for (3.12).

Our task now is to solve the separation problem for (3.15). This can be done by a simplified version of algorithm (3.7). Let $\sigma\equiv v_{ij}$. Normally, we only have to construct in step 1 of (3.7) the auxiliary digraph $D_\sigma=(V_\sigma,A_\sigma)$ with capacities c^σ (associated with the shrunk node σ) from digraph $D_{ij}=(V_{ij},A_{ij})$ with capacities d^{ij}, and perform steps 2 and 3 for this case. If in step c of (3.7) a node set $\tilde{W}\subseteq V_\sigma$ with $\sigma\in\tilde{W}$, $0\notin\tilde{W}$ is identified with $c^\sigma(\delta^-(\tilde{W}))<1+(n-1)M$ then

$$z(A(\tilde{W}))>|\tilde{W}|-1 \tag{3.17}$$

holds and thus the associated precedence forcing constraint is violated. Otherwise, these constraints are satisfied by z. We still have to check whether $|\tilde{W}|\geqq 2$ holds. But this is obvious since $\sigma\in\tilde{W}$ and $c^\sigma(\delta^-(\sigma))=1+(n-1)M$. This shows that the separation problem for precedence forcing constraints can be solved in polynomial time for any $z\in\mathbb{Q}^A$ (with $z\geqq 0$). (Note that $\tilde{W}=V_{ij}$ is allowed in (3.16) and, then, $W=V\backslash(i,j)$ is also allowed.)

If we require that the given $z\in\mathbb{Q}^A$ satisfies (2.3) (2) and (3) in addition—as is the case in our application—we can set the number M equal to zero. This, in fact, simplifies the algorithm a little.

The overall running time of our separation routine for the precedence forcing constraints is at most $O(|R|t)$, where t is the running time for the max-flow algorithm used in step 2 of (3.7). We use the algorithm given in [7].

4. Further inequalities. We note that the sequential ordering problem is closely related to the ATSP and that any inequality valid for the ATSP polytope P_T^n can be brought into a form that is valid for the SOP polytope SOP(n, P) and valid for the set of solutions of (2.3), (2.4), or (2.5).

The classes of valid and facet-defining inequalities for P^n_T (known by 1985) have been surveyed in [16]. In recent years further classes of valid and facet-defining inequalities for P^n_T have been discovered by Balas, Chopra, Fischetti, and Rinaldi, among others. Surveying these achievements here is beyond the scope of this paper. We simply mention those (few) classes of inequalities that we considered in our computations.

The first class consists of the so-called *T_k-inequalities* (most of them facet defining for P^n_T) introduced in [8]. They are defined as follows. Let $k \geqq 2$ and let $W \subseteq V$ be a node set such that $|W| = k$, $w \in W$, and $i, j \in V \setminus W$. Then, the inequality

$$x_{ij} + x_{iw} + x_{wj} + x(A(W)) \leqq |W| \tag{4.1}$$

is called a T_k-inequality. Using algorithm (3.7) for the separation problem of the SECs one can easily design a polynomial time algorithm for T_k-inequalities for all k. We did not implement this procedure but used a heuristic to check this type of inequality for $k = 2, 3, 4$.

Other classes of inequalities, facet defining for P^n_T, can be derived by lifting cycle constraints (see [8], [12], and [16]); we use two of these. They are as follows.

For any ordered set of nodes $\{i_1, i_2, \dots, i_k\} \subset V$, $3 \leqq k \leqq n-1$,

$$\sum_{g=1}^{k-1} x_{i_g i_{g+1}} + x_{i_k i_1} + 2 \sum_{g=2}^{k-1} x_{i_g i_1} + \sum_{g=3}^{k-1} \sum_{h=2}^{g-1} x_{i_g i_h} \leqq k-1 \tag{4.2}$$

is called a *D^+_k-inequality* and

$$\sum_{g=1}^{k-1} x_{i_g i_{g+1}} + x_{i_k, i_1} + 2 \sum_{g=3}^{k} x_{i_1 i_g} + \sum_{g=4}^{k} \sum_{h=3}^{g-1} x_{i_g i_h} \leqq k-1 \tag{4.3}$$

is called a *D^-_k-inequality.* All D^+_k- and D^-_k-inequalities are valid with respect to P^n_T.

We do not know how to solve the separation problem for D^+_k- or D^-_k-inequalities in polynomial time unless we fix k and enumerate. This is a ridiculous procedure for large k, but we implemented it for $k = \{3, 4\}$.

There are two more liftings of four-cycle inequalities that are facet defining for P^n_T and that we checked by enumeration. These inequalities are of the following types. Again let i_1, i_2, i_3, i_4 be four nodes of V; then the inequalities

$$\sum_{g=1}^{3} x_{i_g i_{g+1}} + x_{i_4 i_1} + 2x_{i_2 i_1} + x_{i_2 i_4} + x_{i_3 i_1} + x_{i_4 i_3} \leqq 3, \tag{4.4}$$

$$\sum_{g=1}^{3} x_{i_g i_{g+1}} + x_{i_4 i_1} + 2x_{i_1 i_3} + 2x_{i_3 i_1} \leqq 3 \tag{4.5}$$

are valid for SOP (n, P). Clearly, they are only useful if all arcs used in (4.4) or (4.5) occur in A.

We are aware of the fact that there are more valid and facet-defining inequalities for P^n_T that might be of interest for solving the sequential ordering problem. For instance, the class of *two-matching* inequalities (in their asymmetric version) could also be considered, in particular, since a polynomial time separation routine is available that is a straightforward adaptation of the method of Padberg and Rao [17] designed for the symmetric case. Moreover, *comb* and *clique tree* inequalities could be used in their asymmetric form since they turned out to be very useful for solving the symmetric TSP in practice (see [9], [19], and [20]). In our case, however, the scope was more limited towards finding good lower bounds for not too large problem instances and, due to the requirements from practice, no attempt was made to solve the given problems to optimality. Clearly, if one intends to attack truly large scale SOP instances all these classes of inequalities have to be considered.

We should also mention that the idea to separate by enumeration the "small inequalities" listed above was motivated by studying fractional solutions that could not be cut off by SECs or PFCs. The "small inequalities" frequently did the job.

Let us remark, moreover, that most of the inequalities for P_T^n can be extended to take care of precedences in the same way as the SECs were extended to PFCs. To give an example, take the inequality (facet defining for P_T^n)

$$x_{i_1 i_2} + x_{i_2 i_3} + x_{i_3 i_1} + 2x_{i_2 i_1} \leqq 2. \tag{4.6}$$

Assume that (i, j) belongs to R and that the node v_{ij} obtained by identifying nodes i and j (see (3.13)) is the node i_1; then the inequality

$$x_{ji_2} + x_{i_2 i_3} + x_{i_3 i} + 2x_{i_2 i} \leqq 2 \tag{4.7}$$

is valid for SOP (n, P). This type of SOP extension can be made in various ways. We have implemented separation routines for a few of them but do not want to discuss the simple but rather technical details.

5. Preprocessing. A (usually important) part of a cutting plane procedure consists of analyzing the given problem instance in order to discover some structure that helps to decompose the instance, to reduce its size, or to tighten the IP-formulation by turning some inequalities into equations, fixing certain variables, etc.

We do not want to elaborate on all preprocessing routines that we have implemented; we simply list a few of the straightforward cases. We concentrate here on the IP-formulation (2.5) of the SOP. Suppose the complete digraph $D_n = (V, A_n)$ with cost c_{ij} for all $(i, j) \in A$, and the acyclic and transitively closed precedence digraph $P = (V, R)$ are given. In a first step we determine the node sets V^- and V^+ as follows:

$$V^- = \{v \in V \mid \exists (i, v) \in R, i \neq v\}, \tag{5.1a}$$

$$V^+ = \{v \in V \mid \exists (v, j) \in R, j \neq v\}, \tag{5.1b}$$

i.e., V^- is the set of nodes that have predecessors in P, and V^+ is the set of nodes that have successors in P. It is obvious that the inequalities (2) and (3) of (2.3) can be transformed into

$$\begin{array}{lll} (2') & x(\delta^-(j)) = 1 & \text{for all } j \in V^-, \\ (2) & x(\delta^-(j)) \leqq 1 & \text{for all } j \in V \backslash V^-, \\ (3') & x(\delta^+(j)) = 1 & \text{for all } j \in V^+, \\ (3) & x(\delta^+(j)) \leqq 1 & \text{for all } j \in V \backslash V^+. \end{array}$$

Since we drop all variables corresponding to arcs in $A_n \backslash A$ it may happen that by logical implication some of the inequalities (2) or (3) can also be turned into equations. This type of analysis is made not only in the preprocessing phase but also in all later steps when certain variables can be fixed to zero or 1 due to reduced cost criteria. Again, we do not want to discuss the obvious and well-known details of this technique.

Another preprocessing step that is based on an analysis of the precedence digraph P and the cost values c_{ij} turned out to be quite useful in solving some of our cases, due to their special cost matrix structure. It sometimes happens that, for two nodes i and j that are unrelated for the given precedences, an "artificial" precedence, say (i, j), can be introduced (by analyzing the cost matrix) in such a way that the optimum value of the SOP before and after introducing the relation (i, j) is the same. In such a case we can repeat the other preprocessing steps such as variable fixing and inequality tightening, and start the whole process anew.

To show how an "artificial" precedence can be created we consider a small example on seven nodes. The cost matrix is shown in Table 1. The precedence digraph $P = (V, R)$ is given by $R = \{(1, j) \mid j = 4, 5, 6, 7\} \cup \{(i, 5) \mid i = 1, 4, 6, 7\}$. Nodes 2 and 3 are not related to any other node.

Consider the two unrelated nodes 2 and 7. We observe that $c_{27} = c_{72} = 0$. Moreover, $c_{2k} = c_{7k}$ and $c_{k7} = c_{k2}$ hold for all $k \in V \backslash \{2, 7\}$. In addition, we can observe that $c_{uv} \leqq c_{u2} + c_{2v}$ for all $u, v \in V \backslash \{2, 7\}$, $u \neq v$. Since node 2 is not related to any other node in P we can either add the arc $(2, 7)$ or the arc $(7, 2)$ to P, and we can also set $x_{27} = 1$ or $x_{72} = 1$, without changing the objective function (cost) value of the optimal solution.

It is easy to see how to generalize this observation. If there are two nodes $i, j \in V$ such that

(5.2a) $(i, j), (j, i) \notin R$,

(5.2b) $c_{ij} = c_{ji} = 0$,

(5.2c) $c_{ik} = c_{jk}$ and $c_{ki} = c_{kj}$ for all $k \in V \backslash \{i, j\}$,

(5.2d) $c_{uv} \leqq c_{uj} + c_{jv}$ for all $u, v \in V \backslash \{i, j\}$, $u \neq v$,

(5.2e) $j \notin (V^- \cup V^+)$ (i.e., j has neither a predecessor nor a successor in P),

then either (i, j) or (j, i) can be added to R, and either x_{ij} or x_{ji} can be set to 1, such that at least one optimum solution of the original SOP instance is still optimum for the new case.

It turned out that this "precedence addition rule" helped in some cases to substantially reduce the problem size and to increase the lower bound from the LP relaxation.

6. Outline of the implementations. We have made three new implementations of cutting plane algorithms that compute lower bounds for the SOP. Two algorithms use model (2.5) and one uses model (2.4). Moreover, we compared this with the algorithm for the LP relaxation of model (2.3) described in [4].

Implementation A is based on model (2.4) and implementation B is based on model (2.5). Both were coded in FORTRAN, used Marsten's simplex-based LP solver XMP (see [15]), and were implemented and executed on a SIEMENS PC MX-2 (a 0.7 MIPS personal computer with UNIX operating system).

TABLE 1
Cost coefficients.

j i	1	2	3	4	5	6	7
1	—	1.00	2.00	0.75	0.00	3.00	1.00
2	4.00	—	5.00	3.25	4.00	6.00	0.00
3	7.00	8.00	—	5.50	7.00	9.00	8.00
4	2.75	2.50	2.25	—	2.75	5.25	2.50
5	0.00	1.00	2.00	0.75	—	3.00	1.00
6	10.00	11.00	12.00	10.75	10.00	—	11.00
7	4.00	0.00	5.00	3.25	4.00	6.00	—

Implementation C is based on model (2.5). It was coded in PL/I version 1.5, used the algorithmic tools of the LP-solver MPSX (see [14]), and was implemented and executed on an IBM 4381 (a 7.7 MIPS computer with VM/CMS operating system).

Implementations A and B were meant to determine which of the two models (2.4) and (2.5) are superior from a computational point of view. We also wanted to see whether or not SOP instances of the size coming up in practice (up to about 100 nodes and 280 precedence relationships) can be solved in reasonable time on a PC.

We now briefly outline the basics of our cutting plane approach. We concentrate mainly on the codes B and C that solve the LP relaxations of model (2.5).

The algorithm receives an $n \times n$ cost matrix and an acyclic digraph of precedences as input. We may assume that all cost coefficients are integral (for expository purposes). In a first step we compute the transitive hull of this digraph to obtain the initial precedence subdigraph. In a second step we try to add precedence relations by analyzing the cost matrix as described in § 5. If we add a precedence, we recompute the transitive hull and repeat until no further precedence can be added. We denote the final precedence subdigraph by $P = (V, R)$.

Then we compute the arc set $A = A_n \setminus (\overleftarrow{R} \cup \vec{R})$ (see (2.1)) and we try to find out whether further arcs can be deleted from A (or fixed) by analyzing logical implications.

Now we set up the initial LP consisting of (2.3) (1)-(4), taking care that (as outlined in § 5) some of the inequalities can be turned into equations.

To solve model (2.4) we also set up (7), (8), (10), and (11). In this case we project away half of the variables y_{ij}'s using (8) and we fix some of the variables y_{ij}'s appropriately according to the previous fixing of variables x_{ij}'s.

We now run the heuristic described in [4] and [5] to find a "good" feasible Hamiltonian path. Let λ^H denote its cost. We use it to set up an initial basis for the LP-solver.

We solve the present LP and obtain an optimum solution z with value λ_{LP}. If z is the incidence vector of a feasible Hamiltonian path we are done. We are also done if $\lambda^H - \lambda_{LP} < 1$. In this case the heuristically found feasible Hamiltonian path is optimal.

Otherwise we enter the separation process. We first check whether z satisfies the SECs and then the PFCs using the separation algorithms described in § 3. We add all inequalities found this way to the current LP. If z satisfies all SECs and all PFCs then we call the separation algorithms for the further inequalities mentioned in § 4. Again we add all inequalities found to the current LP.

If the second stage of separation routines fails, we finish the cutting plane algorithm reporting the lower bound λ_{LP}.

Otherwise we continue, but before resolving the augmented LP, we do the well-known reduced cost fixing of variables. If some of the variables can be fixed, we determine the logical implications in order to fix further variables. Moreover, we call the preprocessing routines to tighten the current LP further. In addition, we delete redundant constraints. After these preparations we call the LP solver using the modified LP and the old (dually feasible) basis.

When solving model (2.4), we additionally check the triangle inequalities (9) by enumeration and add all inequalities found to the present LP.

This finishes the outline of our implementations. There are many technical details that we think are important, but it is impossible to report all of them here. The codes B and C, although following the same ideas, do not always produce the same value λ_{LP}, since they were written by different people, and some differences in the order of performing certain steps, setting tolerances, etc., caused variations in the running times and the LP values. In particular, implementation C does not use the T_k-inequalities

(4.1), nor the mechanism for generating "artificial" precedences based on the cost matrix structure (see § 5). On the other hand, the D_k^+- and D_k^--inequalities (4.2) and (4.3) as well as the other further inequalities were only used at selected LP problems, where a given analysis of the point z may suggest a potential violation of the constraints; of course, there is no guarantee that all violations were investigated. Additionally, implementation C temporarily declares "neutral" certain currently nonactive inequalities based on counting the number of previous consecutive LPs where they have been nonactive.

For illustrative purposes let us consider the case described in § 5; see also Table 1. The heuristic gives the feasible solution $1 \rightarrow 4 \rightarrow 2 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 3$. The total cost is $\lambda^H = 2125$. The optimal value of the LP model (2.3) (1)–(4) is $\lambda_{LP} = 1800$; it gives the solution $1 \rightarrow 4 \rightarrow 6 \rightarrow 5 \rightarrow 3$ and $2 \rightarrow 7 \rightarrow 2$. By using our separation algorithm for model (2.5) but without considering the cost matrix structure, the optimal solution of the augmented LP is $\lambda_{LP} = 2075$ (then the gap is 2.40 percent). By exploiting the cost structure as described in § 5 we force the precedence $(2, 7) \in R$ and then update $A := A \backslash \{(7, 2), (5, 2), (2, 5)\}$. It turns out that the optimal value of the new LP is precisely $\lambda_{LP} = 2125$.

7. Computational results. We now report some computational experiences with the three implementations of the cutting plane algorithms outlined in § 6 and compare these with the heuristic described in [4] and [5] and the lower bounding algorithm described in [4].

The report covers 16 instances of the SOP where the number n of tasks ranges from 7 to 98 and the number $|R|$ of precedence relationships from 0 to 283. Six of these cases are real-life and came up in a scheduling system for manufacturing. Four further cases ($P1$, $P1A$, $P4$, and $P9$) were created (artificially) to test certain aspects of the cut generation, mainly the performance of exploiting the cost structure. The remaining six cases are obtained from the real-life cases by dropping all precedence relationships. So these are, in fact, "pure" Hamiltonian path problems.

The artificial cases were constructed as follows. Cases $P4$ and $P9$ are created by replicating case $P1$ $\rho = 2$ and 14 times, respectively. (Case $P1$ is described in § 5.) Node i in $P1$ has the counterparts $nj = i + 7(j-1)$ for $i = 1, 2, \dots, 7$ in cases $P4$ and $P9$, $j = 1, 2$ in $P4$ and $j = 1, 2, \dots, 14$ in $P9$. The sets of nodes $\{i = 1, 2, \dots, 7\}$ and $\{nj = 8, 9, \dots, 14\}$ in case $P4$ have the same internal precedence relationships as the set of nodes in $P1$ have; on the other hand, none of the nodes from one set has precedence relationships with the nodes from the other set. The cost matrix has the following structure for $P4$: $c_{p,q+7} = c_{p+7,q} = c_{p+7,q+7} = c_{p,q}$, $c_{p,q}$ for $p, q = 1, 2, \dots, 7, p \neq q$ as in $P1$, and the other elements are zero. (A similar construction is used for $P9$.) The optimal solution is $N1 \rightarrow N4 \rightarrow N2 \rightarrow N7 \rightarrow N6 \rightarrow N5 \rightarrow N3$ with $\lambda^* = 2125$, where, e.g., $N6$ denotes any sequencing of the node set $\{6, 13, \dots, k\}$ for $k = 6 + 7(\rho - 1)$. Note that, naturally, the optimal value of the LP relaxation (2.3) (1)–(4) is zero for $P4$ and $P9$. By exploiting the cost matrix structure as in § 5 (see (5.2)), implementation B gets the optimal solution without adding any further cut. Additionally, we also analyze the performance of our separation algorithm when the cost matrix structure is not exploited.

Table 2 reports some results on the performance of our algorithm. It gives information about the objective function value and the gap between the best-known upper bound (frequently, the optimal solution value) and the lower bounds obtained by our implementations. The headings are as follows. H refers to the heuristic described in [4] and [5]. E refers to the algorithm described in [4] for obtaining a lower bound of the optimal solution value λ^* in model (2.3). Finally, A, B, and C refer to our three

TABLE 2
Performance of our cutting separation implementations.

			Objective function value					Gap in objective function			
Case	n	$\|R\|$	H	E	A	B	C	E	A	B	C
$P1$	7	7	2125	1950	2125	2125	2075	8.97	0.00	0.00	2.40
$P1A$	7	0	550	450	550	550	550	22.22	0.00	0.00	0.00
$P2$	11	5	2075	2021	2075	2075	2075	2.70	0.00	0.00	0.00
$P2A$	11	0	1866	1763	1866	1866	1843	5.80	0.00	0.00	1.25
$P3$	12	11	1675	1417	1598	1597	1535	18.20	4.82	4.88	9.12
$P3A$	12	0	1472	1386	1472	1472	1459	6.20	0.00	0.00	0.89
$P4$	14	14	2125	1525	2125	2125	2075	39.34	0.00	0.00	2.40
$P5$	25	11	1684	1518	1588	1577	1584	10.90	6.05	6.79	6.31
$P5A$	25	0	1145	1041	1134	1141	1118	10.00	0.97	0.35	2.42
$P6$	47	32	1288	1199	1219	1218	1219	7.40	5.66	5.75	5.66
$P6A$	47	0	915	856	872	872	871	6.09	4.93	4.93	5.05
$P7$	63	233	63	63	62	62	63	0.00	1.61	1.61	0.00
$P7A$	63	0	45	45	45	45	45	0.00	0.00	0.00	0.00
$P8$	78	283	18480	18205	18205	18205	18205	1.51	1.51	1.51	1.51
$P8A$	78	0	1845	1410	1305	1845	1712	30.85	41.37	0.00	7.76
$P9$	98	98	2125	1525	2125	2125	2075	39.34	0.00	0.00	2.40

implementations A, B, and C (see § 6). The first part of Table 2 reports the cost, say λ^H, of the heuristic solution (except for $P8A$; see below) as well as the lower bound, say λ_a, obtained by implementation a for $a =$ A, B, C, and E. It is worth noting that the heuristic gives 2325 as the cost value for $P8A$, but one of our implementations found the (optimal) value 1845. The gap as reported in Table 2 is $100(\lambda^H - \lambda_a)/\lambda_a$. Note that frequently the gap is zero (i.e., implementations A, B, and C prove the optimality of the heuristic solution). We should mention that the gap for $P1$, $P4$, and

TABLE 3
CPU *time of our cutting separation implementations.*

			H	E	A	B	C
Case	n	$\|R\|$	(sec.)	(sec.)	(min.)	(min.)	(sec.)
$P1$	7	7	0.08	0.10	0.14	0.10	0.05
$P1A$	7	0	0.13	0.24	0.18	0.10	0.05
$P2$	11	5	0.26	0.55	0.17	0.20	0.04
$P2A$	11	0	0.22	0.55	1.05	0.15	0.09
$P3$	12	11	0.16	0.36	9.59	1.27	0.89
$P3A$	12	0	0.25	0.56	1.06	0.18	0.21
$P4$	14	14	0.31	1.10	0.17	0.10	0.71
$P5$	25	11	1.17	1.19	23.34	0.43	0.55
$P5A$	25	0	0.35	1.28	13.25	0.46	0.45
$P6$	47	32	3.05	4.78	87.39	4.51	1.14
$P6A$	47	0	1.98	3.85	100.10	1.20	1.08
$P7$	63	233	4.35	3.38	340.43	27.27	5.63
$P7A$	63	0	0.06	3.47	109.37	1.42	4.08
$P8$	78	283	27.62	12.93	443.13	153.01	12.05
$P8A$	78	0	8.93	6.94	250.52	9.36	18.41
$P9$	98	98	28.47	25.01	0.23	0.20	6.20

$P9$ is 2.40 percent when the cost matrix structure is not exploited (as with implementation C) and it is zero when it is exploited.

We should remark at this point that implementation A (using the largest number of variables) was not able to finish all runs due to space limitations on the PC. (We could not store all inequalities found.) In this case we report in Table 2 the lower bound obtained before termination. (Further cutting plane steps might have led to better lower bounds.)

Table 3 reports the CPU time required by the implementations. Implementations H, E, and C were run on an IBM 4381 and the time is given in seconds. Implementations A and B were run on a SIEMENS PC MX-2 and the time is given in minutes. All times reported include input-output operations. Note that the PC-implementation B solves the cases in less than $2\frac{1}{2}$ CPU hours. The mainframe version does this in a few seconds.

Tables 4–6 report the dimensions of the instances and number of cuts that have been generated by each of the three implementations. The headings are as follows. F01 indicates the number of variables x_{ij} that are (permanently) fixed by reduced cost fixing and logical implications. (Note that $|A|$ gives the set of variables x_{ij}'s in the model and, then, $|A| - \text{F01}$ is the number of x_{ij}'s in the last LP problem.) NAP is the number of constraints (2.3) (1)–(3) (i.e., number of constraints in the initial LP relaxation) in implementations B and C; NAP is the number of constraints (2.4) (1)–(3), (7), and (8) in implementation A; NLP is the number of cutting plane separation steps (i.e., number of LP problems); NSEC is the number of subtour elimination constraints (2.3) (5) that have been generated; NYSC is the number of y-related constraints (2.4) (9) that have been generated in implementation A; NPFC is the number of precedence forcing constraints (2.5) (12) that have been generated in implementations B and C; NLC is the number of further cuts generated from the class of inequalities described in § 4; NC is the total number of cuts that have been generated. One can observe that the total number of constraints in any LP is rather small. By comparing NLP and NC we can see the average number of cuts that are appended to the LP model at each iteration.

TABLE 4
Problem dimensions and cut generation. Implementation A.

Case	n	$\|R\|$	$\vec{R}$	$\|A\|$	F01	NAP	NLP	NSEC	NYSC	NLC	NC
$P1$	7	7	1	35	10	39	2	1	10	0	11
$P1A$	7	0	0	63	26	57	1	0	0	0	0
$P2$	11	5	2	153	80	123	4	2	56	6	64
$P2A$	11	0	0	165	89	133	3	2	49	0	51
$P3$	12	11	4	172	53	135	7	4	64	29	97
$P3A$	12	0	0	198	98	157	2	2	41	0	43
$P4$	14	14	2	35	10	39	2	1	10	0	11
$P5$	25	11	2	876	496	629	8	4	667	11	682
$P5A$	25	0	0	900	0	651	6	3	434	0	437
$P6$	47	32	22	3157	1890	2199	15	2	1502	0	1504
$P6A$	47	0	0	3243	1949	2257	5	5	1800	0	1805
$P7$	63	233	138	5255	2134	3567	4	10	1800	13	1823
$P7A$	63	0	0	5859	2957	4033	1	0	0	0	0
$P8$	78	283	206	8237	2079	5597	3	12	600	8	612
$P8A$	78	0	0	9009	0	6163	1	1	200	8	209
$P9$	98	98	14	35	12	39	2	1	10	0	11

TABLE 5
Problem dimensions and cut generation. Implementation B.

Case	n	$\lvert R\rvert$	$\vec{R}$	$\lvert A\rvert$	F01	NAP	NLP	NSEC	NPFC	NLC	NC
$P1$	7	7	1	23	10	15	2	1	1	0	2
$P1A$	7	0	0	42	28	15	2	1	0	0	1
$P2$	11	5	2	103	80	23	4	4	9	0	13
$P2A$	11	0	0	110	89	23	4	3	0	0	3
$P3$	12	11	4	117	53	25	12	9	6	22	37
$P3A$	12	0	0	132	98	25	3	10	0	0	10
$P4$	14	14	2	166	10	29	2	1	1	0	2
$P5$	25	11	2	587	498	51	3	5	12	0	17
$P5A$	25	0	0	600	550	51	4	5	0	0	5
$P6$	47	32	22	2108	1870	95	4	7	7	0	14
$P6A$	47	0	0	2162	1956	95	4	8	0	0	8
$P7$	63	233	138	3535	2802	127	7	8	63	65	136
$P7A$	63	0	0	3906	2957	127	1	0	0	0	0
$P8$	78	283	206	5517	2079	157	15	11	65	66	142
$P8A$	78	0	0	6006	1547	157	18	31	0	16	47
$P9$	98	98	14	9492	—	197	2	1	1	0	2

TABLE 6
Problem dimensions and cut generation. Implementation C.

Case	n	$\lvert R\rvert$	$\vec{R}$	$\lvert A\rvert$	F01	NAP	NLP	NSEC	NPFC	NLC	NC
$P1$	7	7	1	23	14	15	2	2	2	3	7
$P1A$	7	0	0	42	31	15	2	2	0	0	2
$P2$	11	5	2	103	78	23	5	4	12	6	22
$P2A$	11	0	0	110	83	23	5	3	0	0	3
$P3$	12	11	4	117	25	25	11	9	6	12	27
$P3A$	12	0	0	132	74	25	3	9	0	0	9
$P4$	14	14	2	166	20	29	2	3	6	2	11
$P5$	25	11	2	587	469	51	3	5	9	0	14
$P5A$	25	0	0	600	505	51	4	4	0	0	4
$P6$	47	32	22	2108	1842	95	7	7	6	0	13
$P6A$	47	0	0	2162	1925	95	4	7	0	0	7
$P7$	63	233	138	3535	3227	127	7	9	0	0	9
$P7A$	63	0	0	3906	2603	127	3	4	0	0	4
$P8$	78	283	206	5517	4748	147	17	15	72	16	103
$P8A$	78	0	0	6006	2079	157	9	28	0	13	41
$P9$	98	98	14	9492	4723	197	2	3	11	3	17

Table 7 reports the gap reduction on the objective function value obtained by our implementations. The headings are as follows. H is the best known upper bound of λ^*. ALB is the objective function value of the LP relaxation (2.3) (1)-(3). A-GAP = (H − ALB)/ALB percent. KLB is our best-known lower bound on λ^* (i.e., the objective function value of the last LP). K-GAP = (H − KLB)/KLB percent and RK = (KLB − ALB)/(H − ALB) percent.

The first analysis that we can draw from the results shown in Table 7 is the observance of a big discrepancy in the value of A-GAP between the results reported in the literature for randomly generated cases and our experience with real-life cases. It is reported for ATSP cases that the value of ALB was found on the average to be 99.5 percent of the optimal value. We have obtained the optimal value in more than

TABLE 7
Gap reduction on the objective function value.

Case	n	$\|R\|$	H	ALB	A-GAP	KLB	K-GAP	RK
$P1$	7	7	2125	1800	18.06	2125	0.00	100.00
$P1A$	7	0	550	225	100.00	550	0.00	100.00
$P2$	11	5	2075	1946	6.63	2075	0.00	100.00
$P2A$	11	0	1866	1763	5.84	1866	0.00	100.00
$P3$	12	11	1675	1293	29.54	1598	4.88	79.58
$P3A$	12	0	1472	1240	18.71	1472	0.00	100.00
$P4$	14	14	2125	0	*	2125	0.00	100.00
$P5$	25	11	1684	1518	10.94	1588	6.05	42.16
$P5A$	25	0	1145	1041	9.99	1141	0.35	96.15
$P6$	47	32	1288	1199	7.42	1219	5.66	22.47
$P6A$	47	0	915	856	6.89	872	4.93	27.11
$P7$	63	233	63	62	1.61	63	0.00	100.00
$P7A$	63	0	45	45	0.00	45	0.00	*
$P8$	78	283	18480	18204	1.52	18205	1.51	0.36
$P8A$	78	0	1845	1305	41.37	1845	0.00	100.00
$P9$	98	98	2125	0	*	2175	0.00	100.00

50 percent of the cases and we have at hand the lower bound KLB for the other subset; we have to report a big difference between H and ALB and even KLB and ALB.

We should point out the effectiveness of the separation algorithm for identifying subtour elimination constraints that are violated by the current LP solution. See column RK in Table 7 for the ATSP cases (i.e., cases with $|R|=0$). It gives the gap reduction obtained by appending violated SECs to the current LP model. On the other hand, we may observe the performance of the preprocessing procedure based on (2.1) ($|A_n \setminus A|$ variables x_{ij}'s are fixed to zero) for tightening the lower bound ALB for the cases with precedence relationships (i.e., cases with $|R|>0$). Note also how effective the reduced cost fixing can be whenever ALB and H are close enough. Finally, see that ALB is zero for $P4$ and $P9$ in implementation C (i.e., the cost matrix structure is not exploited).

The column headed KLB in Table 7 gives our tightest lower bound on the SOP optimal solution. It is the optimal value of the LP relaxation (2.3) (1)-(4) enlarged by appending the cuts that our separation algorithm identifies as violated cuts. By comparing the columns headed A-GAP and K-GAP and, in particular, analyzing the column headed RK, we can see the effectiveness of appending violated cuts. Notice that the optimality of the solution provided by the heuristic has been proved for 9 out of 16 cases. On the other hand, the largest gap is only 6.05 percent. Branch-and-bound has not been used, since our only objective was to create (hopefully) good lower bounds for the heuristic given in [4] and [5].

8. Conclusions. In this work we have presented two new 0–1 models for the sequential ordering problem. Both are stronger than the model introduced in [4]. We have also introduced polynomial time separation algorithms for subtour elimination constraints and precedence forcing constraints. We have outlined the LP framework of three implementations for tightening the lower bound of the optimal solution and reported our computational results. More theoretical work is required mainly for identifying (in reasonable time) violated further inequalities mentioned in § 5. In any case, our computational experience indicates that this LP-based approach is a quite promising way to analyze the quality of a feasible solution and eventually to obtain an optimal one.

Acknowledgment. We appreciate Manfred Padberg's and Giovanni Rinaldi's comments.

REFERENCES

[1] N. Ascheuer, L. F. Escudero, M. Grötschel, and M. Stoer, *On identifying in polynomial time violated subtour elimination and precedence forcing constraints for the sequential ordering problem*, in Integer Programming and Combinatorial Optimization, R. Kannan and W. R. Pulleyblank, eds., University of Waterloo, Waterloo, Ontario, Canada, 1990, pp. 19-28.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows*, in Optimization, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, eds., North-Holland, Amsterdam, 1989, pp. 211-369.

[3] H. Crowder and M. Padberg, *Solving large-scale symmetric traveling salesman problems to optimality*, Management Sci., 26 (1980), pp. 495-509.

[4] L. F. Escudero, *An inexact algorithm for the sequential ordering problem*, European J. Oper. Res., 37 (1988), pp. 236-253.

[5] ———, *On the implementation of an algorithm for improving a solution to the sequential ordering problem*, Trabajos de Investigación-Operativa, 3 (1988), pp. 117-140.

[6] ———, *A production planning problem in* FMS, Ann. Oper. Res., 17 (1989), pp. 69-104.

[7] A. V. Goldberg and R. E. Tarjan, *A new approach to the maximum flow problem*, Assoc. Comput. Mach., 35 (1988), pp. 921-940.

[8] M. Grötschel, *Polyedrische Charakterisierungen kombinatorischer Optimierungsprobleme*, Hain, Meisenheim am Glan, Germany, 1977.

[9] M. Grötschel and O. Holland, *Solution of large-scale symmetric travelling salesman problems*, Math. Programming, 51 (1991), pp. 191-202.

[10] M. Grötschel, M. Jünger, and G. Reinelt, *A cutting plane algorithm for the linear ordering problem*, Oper. Res., 34 (1984), pp. 1195-1220.

[11] M. Grötschel, L. Lovasz, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, 1988.

[12] M. Grötschel and M. Padberg, *Lineare Charakterisierungen von Travelling Salesman Problemen*, Z. Oper. Res., 21 (1977), pp. 33-64.

[13] K. Hoffman and M. Padberg, LP-*based combinatorial problem solving*, Ann. Oper. Res., 5 (1986), pp. 145-194.

[14] IBM, *Mathematical Programming System Extended* (MPSX/370) *Version* 2, Program reference manual, SH19-6553, 1988.

[15] R. E. Marsten, *The design of the* XMP *linear programming library*, ACM Trans. Math. Programming Software, 7 (1981), pp. 481-497.

[16] M. Padberg and M. Grötschel, *Polyhedral computations*, in The Traveling Salesman Problem, A Guided Tour of Combinatorial Optimization, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy-Kan, and D. B. Shmoys, eds., John Wiley, New York, 1985, pp. 251-360.

[17] M. Padberg and M. R. Rao, *Odd minimum cut-sets and b-matchings*, Math. Oper. Res., 7 (1982), pp. 67-80.

[18] M. Padberg and G. Rinaldi, *An efficient algorithm for the minimum capacity cut problem*, Math. Programming, 47 (1990), pp. 19-36.

[19] ———, *Facet identification for the symmetric travelling salesman polytope*, Math. Programming, 47 (1990), pp. 219-258.

[20] ———, *Optimization of a* 532-*city symmetric traveling salesman problem*, Oper. Res. Lett., 6 (1987), pp. 1-7.

[21] ———, *A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems*, SIAM Rev., 33 (1991), pp. 60-100.

[22] W. Pulleyblank and M. Fiala-Timlin, *Precedence constrained routing*, ORSA/TIMS Joint National Meeting, New York, 1989.

SIAM J. OPTIMIZATION
Vol. 3, No. 1, pp. 43–59, February 1993

003

ERROR BOUND AND REDUCED-GRADIENT PROJECTION ALGORITHMS FOR CONVEX MINIMIZATION OVER A POLYHEDRAL SET*

ZHI-QUAN LUO† AND PAUL TSENG‡

Abstract. Consider the problem of minimizing, over a polyhedral set, the composition of an affine mapping with a strongly convex differentiable function. The polyhedral set is expressed as the intersection of an affine set with a (simpler) polyhedral set and a new local error bound for this problem, based on projecting the reduced gradient associated with the affine set onto the simpler polyhedral set, is studied. A class of reduced-gradient projection algorithms for solving the case where the simpler polyhedral set is a box is proposed and this bound is used to show that algorithms in this class attain a linear rate of convergence. Included in this class are the gradient projection algorithm of Goldstein and Levitin and Poljak, and an algorithm of Bertsekas. A new algorithm in this class, reminiscent of active set algorithms, is also proposed. Some of the results presented here extend to problems where the objective function is extended real valued and to variational inequality problems.

Key words. local error bound, convex minimization, linear convergence, reduced-gradient projection algorithms

AMS(MOS) subject classifications. 49, 90

1. Introduction.

We consider the convex program

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{X}, \end{array} \tag{1.1}$$

where $\mathcal{X}$ is a polyhedral set in the n-dimensional Euclidean space $\Re^n$ and f is a real-valued function defined on $\Re^n$. We assume that f is of the special form

$$f(x) = g(Ex) + \langle q, x\rangle, \tag{1.2}$$

where E is some $m \times n$ matrix, q is some vector in $\Re^n$, and g is a continuously differentiable function in $\Re^m$ with ∇g Lipschitz continuous and strongly monotone in the sense that there exist positive scalars $\rho > 0$ and $\sigma > 0$ such that

$$\|\nabla g(z) - \nabla g(w)\| \le \rho \|z - w\| \quad \forall z, \quad \forall w, \tag{1.3}$$

and

$$\langle \nabla g(z) - \nabla g(w), z - w\rangle \ge \sigma \|z - w\|^2 \quad \forall z, \quad \forall w. \tag{1.4}$$

We also assume that the optimal solution set for (1.1), denoted by $\mathcal{X}^*$, is nonempty and denote by v^* the value of f on $\mathcal{X}^*$. In our notation, all vectors are column vectors, superscript T denotes matrix transpose, $\langle\cdot,\cdot\rangle$ denotes the usual Euclidean inner product, and $\|\cdot\|$ denotes the Euclidean norm induced by $\langle\cdot,\cdot\rangle$.

* Received by the editors January 3, 1991; accepted for publication (in revised form) November 18, 1991.

† Communications Research Laboratory, Room 225, McMaster University, Hamilton, Ontario, L8S 4K1, Canada (luozq@sscvax.cis.mcmaster.ca). The research of this author is supported by Natural Sciences and Engineering Research Council of Canada grant OPG0090391.

‡ Department of Mathematics, GN-50, University of Washington, Seattle, Washington 98195 (tseng@math.washington.edu).

There are many optimization problems that satisfy the above assumptions, including convex quadratic programs and a certain routing problem in data networks (see [BeG87]). We remark that the assumption that g be real valued is made only to simplify the analysis and can be relaxed so as to allow, for example, certain entropy optimization problems and their dual to be captured by the problem framework. (See §6 for detailed discussions.)

A classical method for solving (1.1) is the *gradient projection* algorithm of Goldstein [Gol64] and Levitin and Poljak [LeP65], which follows each gradient step by a projection onto the feasible set $\mathcal{X}$:

$$x := [x - \alpha \nabla f(x)]^+_{\mathcal{X}},$$

where $[\cdot]^+_{\mathcal{X}}$ denotes the orthogonal projection onto $\mathcal{X}$ and α is some suitably chosen positive stepsize. This method has been well studied and, when combined with second-order scaling, has been successful in solving large quadratic programs with box constraints (see, e.g., [Ber76], [Ber82], [GaB84], and [Mor89]). However, when $\mathcal{X}$ is not a box, the projection $[\cdot]^+_{\mathcal{X}}$ cannot be easily computed and this method can suffer from poor performance.

For the special case where $\mathcal{X}$ is the Cartesian product of simplices, Bertsekas proposed a modification of the gradient projection algorithm which avoids the relatively expensive operation of projecting onto the simplices (see [Ber80], [Ber82], [BeG83], and [BeG87]). (A simplex in $\Re^n$ is a set of the form $\{x \in \Re^n \mid \sum_i x_i = c,\ x \geq 0\}$ for some $c > 0$.) Instead, the algorithm of Bertsekas moves an iterate opposite the direction of a certain *reduced* gradient associated with the knapsack constraints and follows this step with a projection onto the nonnegative orthant. This algorithm has been successfully applied to solving a certain routing problem in data networks (see [BeG83], [BeG87], and [BeT89]) and can even be implemented in a distributed asynchronous manner (see [Tsa89] and [TsB86]).

A key question concerns the convergence and the rate of convergence of the above algorithms. For the gradient projection algorithm this question is largely resolved. It was shown by Bertsekas and Gafni [BeG82], in the more general context of variational inequality problems, and rediscovered by Luo and Tseng [LuT92b], that the gradient projection algorithm for solving (1.1) attains a linear rate of convergence, provided that the stepsize α is suitably chosen. Similar results were obtained by Dunn [Dun81], [Dun87] and Gawande and Dunn [GaD88] for the general problem of minimizing a differentiable function over a closed convex set, but under an additional assumption that all local minimizers are isolated and that the objective function satisfies a certain local growth condition. Central to their analysis is a certain local *error bound* for estimating the distance from a point $x \in \mathcal{X}$ to $\mathcal{X}^*$, defined as

$$\phi(x) = \min_{x^* \in \mathcal{X}^*} \|x - x^*\|. \tag{1.5}$$

In particular, it was shown in [LuT92b] that $\phi(x)$ can be bounded above by some constant times

$$\|x - [x - \nabla f(x)]^+_{\mathcal{X}}\|,$$

the norm of the "natural residual" at x, provided that the latter quantity is small. The same local error bound also extends to affine variational inequality problems (see [Rob81] and [LuT92c]) and holds globally if f is strongly convex [Pan87]. For the Bertsekas algorithm, however, no comparable result was known. We remark that bounds for ϕ have been studied quite extensively, although the focus has been on global bounds and on using the bounds to terminate iterative algorithms and to extract sensitivity/stability information near the optimal solution set (see [MaS87], [MaD88], [Pan87], and [Rob82]).

The goals of this paper are twofold. First, we propose a generalization of the above error bound based on a certain decomposition of the polyhedral set $\mathcal{X}$. More specifically, let us express $\mathcal{X}$ as

$$\mathcal{X} = \mathcal{C} \cap \{x \in \Re^n \mid Bx = c\}, \tag{1.6}$$

for some (simpler) polyhedral set $\mathcal{C} \in \Re^n$, some $l \times n$ matrix B, and some vector c in $\Re^l$. We will show that $\phi(x)$ can be bounded above by some constant times

$$\|x - [x - \nabla f(x) + B^T p]^+_{\mathcal{C}}\| + \|Bx - c\|, \tag{1.7}$$

for any $x \in \mathcal{C}$ and any $p \in \Re^l$ for which the above quantity is "sufficiently" small. Here $[\cdot]^+_{\mathcal{C}}$ denotes the orthogonal projection onto $\mathcal{C}$. Some obvious advantages of this new local error bound, relative to the earlier one, are (i) x is only required to be in $\mathcal{C}$, not $\mathcal{X}$, and (ii) instead of projecting onto $\mathcal{X}$, we project onto the simpler set $\mathcal{C}$.

Second, we propose a class of feasible descent algorithms for solving the special case of (1.1) where $\mathcal{C}$ is a box. At each iteration of these algorithms, we compute a z according to the projection step

$$z := [x - \alpha(\nabla f(x) - B^T p)]^+_{\mathcal{C}},$$

for some stepsize $\alpha > 0$ and some multiplier vector p, and then adjust a subset of the coordinates of z to obtain a new iterate in $\mathcal{X}$. Both the gradient projection algorithm and the algorithm of Bertsekas described earlier can be shown to belong to this class. By using the new local error bound, we show that the iterates generated by any algorithm in this class converge at least linearly to an optimal solution. (Here and throughout, by linear convergence we mean R-linear convergence in the sense of Ortega and Rheinboldt [OrR70].) We also propose a new algorithm in this class reminiscent of active set algorithms.

The remainder of this paper is organized as follows. In §2 we prove some technical facts concerning the problem (1.1); in §3 we use these facts to establish the new local error bound. In §4, we describe the class of feasible descent algorithms mentioned above and relate them to the gradient projection algorithm and to the algorithm of Bertsekas. In §5, we use the error bound of §3 to show that any algorithm in this class which uses an Armijo-like stepsize rule is linearly convergent. In §6, we give our conclusion and discuss extensions.

Throughout this paper, we adhere to the following notations. For any vector x in $\Re^k$, we denote by x_j the jth component of x and, for any subset $J \subseteq \{1, \ldots, k\}$, we denote by x_J the vector with components x_j, $j \in J$. For any matrix A, we denote by $\|A\|$ the matrix norm of A induced by the vector Euclidean norm $\|\cdot\|$, i.e., $\|A\| = \max_{\|x\|=1} \|Ax\|$.

2. Technical preliminaries. In this section we will prove a number of interesting facts concerning the solution set $\mathcal{X}^*$ and the level sets of f over certain subsets of $\mathcal{C}$. These facts will be used in the analysis of subsequent sections.

First, by using the strict convexity of g (cf. (1.4)) and the special structure of f (cf. (1.2)), we have the following simple lemma which says that the linear mapping $x \mapsto Ex$ is invariant over the solution set $\mathcal{X}^*$ (also see [LuT92a] and [Tse91]).

LEMMA 2.1. *There exists a $t^* \in \Re^m$ such that $Ex^* = t^*$ for all $x^* \in \mathcal{X}^*$.*

From (1.2) and the chain rule for differentiation, we have

$$\nabla f(x) = E^T \nabla g(Ex) + q, \quad \forall x. \tag{2.1}$$

Then, (1.3) yields that ∇f is Lipschitz continuous with Lipschitz constant $\rho \|E^T\| \|E\|$, that is,

$$\|\nabla f(x) - \nabla f(y)\| \le \rho \|E^T\| \|E\| \|x - y\|, \quad \forall x, \quad \forall y, \tag{2.2}$$

and Lemma 2.1 yields that ∇f is invariant over $\mathcal{X}^*$ or, more precisely,

$$\nabla f(x^*) = d^*, \quad \forall x^* \in \mathcal{X}^*, \tag{2.3}$$

where we let $d^* = E^T \nabla g(t^*) + q$.

The optimality conditions for (1.1), together with (2.3), imply that $\mathcal{X}^*$ is equivalently the solution set of the linear program $\min_{x \in \mathcal{X}} \langle d^*, x \rangle$. Then, as we shall see in the next section, the question of finding a local error bound for (1.1) translates into a perturbation analysis on the solution set to this linear program. To perform this analysis, we will need the following result, due originally to Hoffman [Hof52] (see also [Rob73] and [MaS87]), on the Lipschitzian continuity of the solution set to a linear system as a multifunction of the right-hand side. This result will be used in the proofs of Lemma 3.1 and Theorem 3.2 which follow.

LEMMA 2.2. *Let C and D be any $r \times k$ and $s \times k$ matrices. Then, there exists a constant $\theta > 0$ depending on C and D only such that, for any $\bar{x} \in \Re^k$ and any $(d, e) \in \Re^r \times \Re^s$ such that the linear system $Cy = d$, $Dy \ge e$ is consistent, there is a point $\bar{y}$ satisfying $C\bar{y} = d$, $D\bar{y} \ge e$ with*

$$\|\bar{x} - \bar{y}\| \ \le \ \theta(\|C\bar{x} - d\| + \|D\bar{x} - e\|).$$

For each $v \ge v^*$ and $\delta \ge 0$, define the level set

$$\mathcal{F}_\delta^v = \{x \in \mathcal{C} \mid \|Bx - c\| \le \delta, \ f(x) \le v\}.$$

(Note that $\mathcal{F}_0^{v^*} = \mathcal{X}^*$ and $\mathcal{F}_{\delta'}^{v'} \subseteq \mathcal{F}_\delta^v$ whenever $v' \le v, \delta' \le \delta$.) By using the polyhedral structure of $\mathcal{X}$ (cf. (1.6)) together with the strict convexity of g (cf. (1.4)), we can show the following boundedness property of $E\mathcal{F}_\delta^v$. This property will be used in the proofs of Lemma 3.1 and Theorem 5.3. Its proof is patterned after that of Fact 4.1 in [Tse91] and is based on the observation that a strictly convex function has bounded level sets whenever its infimum is attained at some point.

LEMMA 2.3. *For any $v \ge v^*$ and any $\delta \ge 0$, the set $E\mathcal{F}_\delta^v$ is nonempty and bounded.*

Proof. Fix any $v \ge v^*$ and any $\delta \ge 0$. The set $E\mathcal{F}_\delta^v$ is clearly nonempty since $\mathcal{F}_\delta^v$ is nonempty. If $E\mathcal{F}_\delta^v$ were not bounded, then the closed convex set

$$\mathcal{L} = \{(t, x, \zeta) \in \Re^{m+n+1} \mid t = Ex, \ x \in \mathcal{C}, \ \|Bx - c\| \le \delta, \ f(x) \le \zeta\}$$

would have a direction of recession $(v, u, 0)$ with $v \ne 0$ (see [Roc70]). Let x^* be any element of $\mathcal{X}^*$. Then, by Lemma 2.1, (t^*, x^*, v^*) is a point in $\mathcal{L}$, so $(t^*, x^*, v^*) + \theta(v, u, 0)$ is also in $\mathcal{L}$ for all $\theta \ge 0$. This implies $x^* + \theta u \in \mathcal{C}$ and $f(x^* + \theta u) \le v^*$ for all $\theta \ge 0$. Moreover, we see from the structure of $\mathcal{L}$ that $Bu = 0$ and $Eu = v$. The former implies $B(x^* + \theta u) = Bx^* = c$ for all $\theta \ge 0$, so $x^* + \theta u \in \mathcal{X}^*$ for all $\theta \ge 0$. On the other hand, the latter, together with $v \ne 0$, implies that $E(x^* + \theta u)$ is not constant for $\theta \ge 0$, a contradiction of Lemma 2.1. □

3. A new local error bound. In this section we show that the distance from a point x in $\mathcal{C}$ to $\mathcal{X}^*$ can be bounded above by the quantity (1.7) when the latter quantity is small and $f(x)$ is bounded. The proof of this is analogous to an argument used in [LuT92b] and is based on a certain property of (1.7) for identifying (locally) those constraints which are "active" at some optimal solution. By treating these active constraints as equalities, we then apply Hoffman's result (Lemma 2.2), together with the Lipschitz continuity and strong monotonicity properties of ∇g (cf. (1.3) and (1.4)), to establish the desired bound.

First, since $\mathcal{C}$ is a polyhedral set, we can express it as

$$\mathcal{C} = \{x \in \Re^n \mid Ax \geq b\}, \tag{3.1}$$

for some $k \times n$ matrix A and some $b \in \Re^k$. For convenience, we denote by A_i the ith row of A and, for any subset $I \subseteq \{1, \ldots, k\}$, by A_I the submatrix of A obtained by removing all rows i of A with $i \notin I$. Then, for any $(x, p) \in \mathcal{C} \times \Re^l$, the vector $z = [x - \nabla f(x) + B^T p]^+_{\mathcal{C}}$ satisfies, together with some multiplier vector $\lambda \in \Re^k$, the following Kuhn–Tucker conditions:

$$x - z + B^T p + A^T \lambda = \nabla f(x), \quad \lambda_i = 0, \quad \forall i \notin I, \quad A_i z = b_i, \quad \forall i \in I, \tag{3.2}$$

$$Az \geq b, \qquad \lambda \geq 0, \tag{3.3}$$

where I is some (possibly empty) subset of $\{1, \ldots, k\}$. We say that an $I \subseteq \{1, \ldots, k\}$ is *identifiably basic* at a vector $(x, p) \in \mathcal{C} \times \Re^l$ if (x, p), together with $z = [x - \nabla f(x) + B^T p]^+_{\mathcal{C}}$ and some $\lambda \in [0, \infty)^k$, satisfies (3.2).

By using Lemmas 2.1, 2.2, and 2.3, we show the following lemma which roughly says that if $x \in \mathcal{C}$ is sufficiently close to $\mathcal{X}^*$, then those indices which are identifiably basic at (x, p) for some p are also identifiably basic at some element of $\mathcal{X}^* \times \Re^l$.

LEMMA 3.1. *Fix any $v \geq v^*$. There exists an $\epsilon > 0$ such that, for any $(x, p) \in \mathcal{F}^v_\epsilon \times \Re^l$ with $\|x - [x - \nabla f(x) + B^T p]^+_{\mathcal{C}}\| \leq \epsilon$ and any $I \subseteq \{1, \ldots, k\}$ that is identifiably basic at (x, p), there is some $(x^*, p^*) \in \mathcal{X}^* \times \Re^l$ at which I is identifiably basic.*

Proof. We argue by contradiction. If the claim does not hold, then there would exist an $I \subseteq \{1, \ldots, k\}$ and a sequence of vectors $\{(x^r, p^r)\}_{r=1,2,\ldots}$ in $\mathcal{F}^v_1 \times \Re^l$ with I identifiably basic at (x^r, p^r) for all r and

$$x^r - z^r \to 0, \qquad Bx^r \to c, \tag{3.4}$$

where we let

$$z^r = [x^r - \nabla f(x^r) + B^T p^r]^+_{\mathcal{C}}, \quad \forall r, \tag{3.5}$$

and yet there is no $(x^*, p^*) \in \mathcal{X}^* \times \Re^l$ at which I is identifiably basic.

Since $x^r \in \mathcal{F}^v_1$ for all r, it follows from Lemma 2.3 that $\{Ex^r\}$ is bounded. Let t^∞ be any cluster point of $\{Ex^r\}$ and let R be a subsequence of $\{1, 2, \ldots\}$ such that

$$\{Ex^r\}_R \to t^\infty. \tag{3.6}$$

We show below that t^∞ is equal to t^*.

Since ∇g is continuous everywhere, then we obtain from (3.6) (and using the fact $\nabla f(x^r) = E^T \nabla g(Ex^r) + q$ for all r) that

$$\{\nabla f(x^r)\}_R \to E^T \nabla g(t^\infty) + q. \tag{3.7}$$

For each $r \in R$, consider the following linear system in x, p, z, and λ:

$$\begin{aligned}&B^T p + A^T \lambda = \nabla f(x^r) + z^r - x^r, \quad Az \geq b, \quad \lambda \geq 0,\\ &\lambda_i = 0, \quad \forall i \notin I, \quad A_i z = b_i, \quad \forall i \in I,\\ &Ex = Ex^r, \quad z - x = z^r - x^r, \quad Bx = Bx^r.\end{aligned}$$

The above system is consistent since, by I being identifiably basic at (x^r, p^r) and by (3.2)–(3.3), (x^r, p^r, z^r), together with some $\lambda^r \in \Re^k$, is a solution of it. Then, by Lemma 2.2, it has a solution $(\hat{x}^r, \hat{p}^r, \hat{z}^r, \hat{\lambda}^r)$ whose size is bounded by some constant (depending on A, B, and E only) times the size of the right-hand side. Since the right-hand side of the above system is clearly bounded as $r \to \infty$, $r \in R$ (cf. (3.4), (3.6), and (3.7)), we have that $\{(\hat{x}^r, \hat{p}^r, \hat{z}^r, \hat{\lambda}^r)\}_R$ is bounded. Moreover, every one of its cluster points, say $(x^\infty, p^\infty, z^\infty, \lambda^\infty)$, satisfies (cf. (3.4), (3.6), and (3.7))

$$\begin{aligned}&B^T p^\infty + A^T \lambda^\infty = E^T \nabla g(t^\infty) + q, \quad Az^\infty \geq b, \quad \lambda^\infty \geq 0,\\ &\lambda_i^\infty = 0, \quad \forall i \notin I, \quad A_i z^\infty = b_i, \quad \forall i \in I,\\ &Ex^\infty = t^\infty, \quad z^\infty - x^\infty = 0, \quad Bx^\infty = c.\end{aligned}$$

Upon using (cf. (2.1)) $E^T \nabla g(Ex^\infty) + q = \nabla f(x^\infty)$, we can simplify the above relations to

$$\begin{aligned}&B^T p^\infty + A^T \lambda^\infty = \nabla f(x^\infty), \quad Ax^\infty \geq b, \quad \lambda^\infty \geq 0,\\ &\lambda_i^\infty = 0, \quad \forall i \notin I, \quad A_i x^\infty = b_i, \quad \forall i \in I, \quad Bx^\infty = c.\end{aligned}$$

This shows that $x^\infty \in \mathcal{X}$ and that $\langle \nabla f(x^\infty), x - x^\infty \rangle \geq 0$ for all $x \in \mathcal{X}$ (cf. (1.6) and (3.1)). Thus $x^\infty \in \mathcal{X}^*$ and, by Lemma 2.1, $t^\infty = t^*$. Moreover, I is identifiably basic at (x^∞, p^∞) (cf. (3.2)), so a contradiction is established. □

Lemmas 2.1, 2.2, and 3.1 together yield the main result of this section.

THEOREM 3.2 (local error bound). *Fix any $v \geq v^*$. There exist scalars $\epsilon > 0$ and $\kappa > 0$ (depending on v and the problem data only) such that*

$$\phi(x) \leq \kappa(\|x - [x - \nabla f(x) + B^T p]_{\mathcal{C}}^+\| + \|Bx - c\|)$$

for any $(x, p) \in \mathcal{F}_\epsilon^v \times \Re^l$ with $\|x - [x - \nabla f(x) + B^T p]_{\mathcal{C}}^+\| \leq \epsilon$.

Proof. Let ϵ be the scalar in Lemma 3.1 corresponding to v. Consider any $(x, p) \in \mathcal{F}_\epsilon^v \times \Re^l$ satisfying the hypothesis of the theorem and let I be any subset of $\{1, \ldots, k\}$ that is identifiably basic at (x, p) and let $z = [x - \nabla f(x) + B^T p]_{\mathcal{C}}^+$. By (3.2) and (3.3), there exists some $\lambda \in \Re^k$ satisfying, together with x, p, and z,

$$\begin{aligned}&B^T p + A^T \lambda = z - x + \nabla f(x), \quad Ax \geq b + A(x - z), \quad \lambda \geq 0,\\ &\lambda_i = 0, \quad \forall i \notin I, \quad A_i x = b_i + A_i(x - z), \quad \forall i \in I.\end{aligned}$$

By Lemma 3.1, there exists an $(x^*, p^*) \in \mathcal{X}^* \times \Re^l$ such that I is identifiably basic at (x^*, p^*), so the following linear system in x^*, p^*, and λ^*:

$$\begin{aligned}&B^T p^* + A^T \lambda^* = d^*, \quad Ax^* \geq b, \quad \lambda^* \geq 0,\\ &\lambda_i^* = 0, \quad \forall i \notin I, \quad A_i x^* = b_i, \quad \forall i \in I, \quad Ex^* = t^*, \quad Bx^* = c\end{aligned}$$

is consistent (cf. (2.3), (3.2)–(3.3), and Lemma 2.1). Conversely, it can be seen that every solution (x^*, p^*, λ^*) to this linear system satisfies $x^* \in \mathcal{X}^*$. Upon comparing the above two systems, we see that, by Lemma 2.2, there exists a solution (x^*, p^*, λ^*) to the second system such that

$$\|(x,p,\lambda)-(x^*,p^*,\lambda^*)\| \le \theta(\|z-x+\nabla f(x)-d^*\|+\|A(x-z)\|+\|Ex-t^*\|+\|Bx-c\|),$$

where θ is some scalar constant depending on A, B, and E only. By (2.1), the definition of d^*, and the Lipschitz condition (1.3), we also have $\|\nabla f(x) - d^*\| = \|E^T\nabla g(Ex) - E^T\nabla g(t^*)\| \le \rho\|E^T\|\|Ex - t^*\|$, so the above relation yields

$$\|(x,p,\lambda) - (x^*,p^*,\lambda^*)\| \le \theta((\|A\|+1)\|x-z\| + (\rho\|E^T\|+1)\|Ex-t^*\| + \|Bx-c\|).$$

Upon rewriting some of the above relations and by using the fact $d^* = \nabla f(x^*)$ (cf. (2.3)), we have

$$x - z + B^Tp + A_I^T\lambda_I = \nabla f(x), \quad B^Tp^* + A_I^T\lambda_I^* = \nabla f(x^*), \tag{3.8}$$

$$A_Iz = b_I, \quad A_Ix^* = b_I, \quad Bx^* = c, \tag{3.9}$$

and

$$\|(x,p,\lambda) - (x^*,p^*,\lambda^*)\| \le O(\|Ex-t^*\| + \gamma), \tag{3.10}$$

where we let $\gamma = \|x-z\| + \|Bx-c\|$ and, for convenience, use the notation $\alpha \le O(\beta)$ to indicate that $\alpha \le \omega\beta$ for some scalar $\omega > 0$ depending on v and the problem data only. In addition, I is identifiably basic at (x^*, p^*) and (cf. (1.4))

$$\sigma\|Ex - t^*\|^2 \le \langle Ex - t^*, \nabla g(Ex) - \nabla g(t^*)\rangle. \tag{3.11}$$

We will use (3.8)–(3.11) to show that $\|x-x^*\| \le O(\gamma)$, which would then complete the proof. Since $Ex^* = t^*$ (cf. Lemma 2.1) and $\nabla f(x) - \nabla f(x^*) = E^T\nabla g(Ex) - E^T\nabla g(Ex^*)$ (cf. (2.1)), then (3.11), together with (3.8)–(3.9), yields

$$\begin{aligned}
\sigma\|Ex - t^*\|^2 &\le \langle Ex - Ex^*, \nabla g(Ex) - \nabla g(Ex^*)\rangle \\
&= \langle x - x^*, \nabla f(x) - \nabla f(x^*)\rangle \\
&= \langle x - x^*, B^Tp + A_I^T\lambda_I + x - z - B^Tp^* - A_I^T\lambda_I^*\rangle \\
&= \langle B(x - x^*), p - p^*\rangle + \langle A_I(x - x^*), \lambda_I - \lambda_I^*\rangle + \langle x - x^*, x - z\rangle \\
&= \langle Bx - c, p - p^*\rangle + \langle A_I(x - z), \lambda_I - \lambda_I^*\rangle + \langle x - x^*, x - z\rangle \\
&\le \|Bx - c\|\|p - p^*\| + \|x - z\|(\|A\|\|\lambda - \lambda^*\| + \|x - x^*\|) \\
&\le \|A\|(\|p - p^*\| + \|\lambda - \lambda^*\| + \|x - x^*\|)\gamma,
\end{aligned}$$

where the last inequality follows from the definition of γ. Applying the above relation once and (3.10) twice then gives

$$\begin{aligned}
\|x - x^*\|^2 &\le O((\|Ex - t^*\| + \gamma)^2) \\
&\le O(\|Ex - t^*\|^2 + \gamma^2) \\
&\le O((\|p - p^*\| + \|\lambda - \lambda^*\| + \|x - x^*\|)\gamma + \gamma^2) \\
&\le O((\|Ex - t^*\| + \gamma)\gamma + \gamma^2).
\end{aligned}$$

Since $\|Ex - t^*\| \le \|E\|\|x - x^*\|$, the above relation implies that there exists a scalar constant $\omega > 0$ (depending on v and the problem data only) such that

$$\|Ex - t^*\|^2 \le \omega(\|Ex - t^*\|\gamma + \gamma^2).$$

This is a quadratic inequality of the form $a^2 \le \omega(a\gamma + \gamma^2)$, which implies $a \le \frac{1}{2}(\omega + \sqrt{\omega^2 + 4\omega})\gamma$ and therefore

$$\|Ex - t^*\| \le \tfrac{1}{2}(\omega + \sqrt{\omega^2 + 4\omega})\gamma.$$

Combine this bound with (3.10) and we obtain $\|(x,p,\lambda) - (x^*,p^*,\lambda^*)\| \le O(\gamma)$. □

We note that the proof of Theorem 3.2 in fact yields the stronger result that, for any $(x,p) \in \mathcal{F}_\epsilon^v \times \Re^l$ satisfying $\|x - [x - \nabla f(x) + B^T p]_C^+\| \le \epsilon$ and any $I \subseteq \{1,\ldots,k\}$ that is identifiably basic at (x,p), there exists an $(x^*,p^*) \in \mathcal{X}^* \times \Re^l$ such that I is identifiably basic at (x^*,p^*) and $\|x - x^*\| \le \kappa(\|x - [x - \nabla f(x) + B^T p]_C^+\| + \|Bx - c\|)$, for some scalar κ depending on v and the problem data only. Roughly speaking, we can bound ϕ and identify the active constraints at the same time. Finally, we remark that, at the price of forgoing this stronger result, the proof of Theorem 3.2 can be simplified further by appealing to a result of Robinson [Rob81] on the local upper Lipschitzian nature of polyhedral multifunctions.

4. RGP algorithms. In this section, we introduce a general class of feasible descent algorithms for solving the special case of (1.1) where C is the nonnegative orthant in $\Re^n$, i.e.,

$$C = [0,\infty)^n. \tag{4.1}$$

An algorithm in this class updates an iterate by first moving it opposite a certain reduced-gradient direction, then projecting it onto C, and finally adjusting a subset of the coordinates with zero reduced gradient, so that the new iterate remains in $\mathcal{X}$. We will show that both the gradient projection algorithm and the algorithm of Bertsekas mentioned in §1 belong to this class. We also propose a new algorithm in this class reminiscent of active set algorithms and, in particular, of a projected Newton method of Bertsekas [Ber82]. Unlike most active set algorithms, this algorithm can add/drop many constraints from its active set at each iteration. We remark that the above class of algorithms readily extends to the case where C is a *box* in $\Re^n$, i.e., the Cartesian product of closed intervals, but, for simplicity, we will not consider this more general case here.

In what follows, we denote by B_j the jth column of B and, for each $J \subseteq \{1,\ldots,n\}$, by B_J the matrix obtained by removing all columns B_j, $j \notin J$, from B. We define $\nabla_j f$ and $\nabla_J f$ analogously. We also denote by $\bar{J}$ the complement of J with respect to $\{1,\ldots,n\}$.

To motivate our algorithms, consider an iteration of the gradient projection algorithm: $x' = [x - \alpha\nabla f(x)]_{\mathcal{X}}^+$, where x is the current iterate, α is the stepsize, and x' is the new iterate. Let $[\cdot]_+$ denote the orthogonal projection onto $[0,\infty)^n$. By using the structure of $\mathcal{X}$ given by (1.6) and (4.1), we can rewrite this iteration as $x' \in \mathcal{X}$ and, for some $p \in \Re^l$,

$$x' = [x - \alpha(\nabla f(x) - B^T p)]_+. \tag{4.2}$$

(It can be seen that p is in fact an optimal Lagrange multiplier vector associated with the constraints $Bx = c$ in the problem of projecting $x/\alpha - \nabla f(x)$ onto $\mathcal{X}$.) Thus, the above iteration is equivalent to the problem of finding a $p \in \Re^l$ so that x' given by (4.2) is in $\mathcal{X}$. Can the restriction (4.2) be relaxed so it would be relatively easy to find such a p?

To answer this question, suppose that, in addition to (4.1), we have $B = [1\ 1\ \cdots\ 1]$ and $c = 1$ (so $\mathcal{X}$ is the unit simplex). Consider the algorithm of Bertsekas mentioned in §1 for solving this special case of (1.1), which operates as follows: Given an iterate $x \in \mathcal{X}$, it chooses an index $j \in \{1, \ldots, n\}$ for which

$$\nabla_j f(x) = \min_k \nabla_k f(x), \tag{4.3}$$

and computes a new iterate $x' \in \mathcal{X}$ according to

$$x'_k = [x_k - \alpha(\nabla_k f(x) - \nabla_j f(x))]_+ \quad \forall k \neq j, \tag{4.4}$$

$$x'_j = 1 - \sum_{k \neq j} x'_k, \tag{4.5}$$

where α is some positive stepsize. (The fact that $x'_j \geq 0$ follows from the observation that $x'_k \leq x_k$ for all $k \neq j$, so the fact $\sum_k x_k = 1 = \sum_k x'_k$ yields $x'_j \geq x_j$.) A moment of reflection shows that the iteration (4.4) is simply the following relaxed version of (4.2):

$$x'_k = [x_k - \alpha(\nabla_k f(x) - B_k^T p)]_+, \quad \forall k \neq j, \tag{4.6}$$

with $p = \nabla_j f(x)$. Moreover, by combining (4.4) with (4.5), we see that

$$\|x' - x\| \leq \sqrt{n}\|x - [x - \alpha(\nabla f(x) - B^T p)]_+\|. \tag{4.7}$$

We remark that, for simplicity, we considered only the unscaled version of the Bertsekas algorithm. See [BeG87, §5.7] for a description of the full algorithm; see [Ber82, §3] and [BeG83] for a related algorithm in which j is chosen by the maximum component rule: $j = \arg\max_k x_k$. This latter algorithm is closely linked to the active-set-type algorithm to be described below.

The formulas (4.6) and (4.7) suggest the following generalization of the gradient projection algorithm and the Bertsekas algorithm for solving (1.1) (under the condition (4.1)) whereby, given an iterate $x \in \mathcal{X}$, we choose a positive stepsize α and we compute a new iterate x' which, together with some $p \in \Re^l$, satisfies

$$x'_k = [x_k - \alpha(\nabla_k f(x) - B_k^T p)]_+, \quad \forall k \quad \text{with } \nabla_k f(x) \neq B_k^T p, \tag{4.8}$$

and

$$\|x' - x\| \leq \tau_1 \|x - [x - \alpha(\nabla f(x) - B^T p)]_+\|, \tag{4.9}$$

with τ_1 some scalar constant. In order to maintain feasibility, we assume that the new iterate x' has the property that

$$x' \in \mathcal{X} \quad \text{whenever } \alpha < \frac{\tau_2}{\|\nabla f(x)\|}, \tag{4.10}$$

with τ_2 some scalar constant (possibly $\tau_2 = \infty$). Thus x' is feasible whenever α is chosen to be sufficiently small.

We will call any iteration a *reduced-gradient* projection (RGP) iteration if it generates, for a given iterate $x \in \mathcal{X}$ and a stepsize $\alpha > 0$, a new iterate x' satisfying (together with some $p \in \Re^l$) the relations (4.8)–(4.10). Roughly speaking, at each RGP iteration we take a step opposite the *reduced-gradient* direction $\nabla f(x) - B^T p$, project onto $[0, \infty)^n$, and then adjust those coordinates with zero reduced gradient so as to remain in $\mathcal{X}$. Any algorithm that generates iterates in $\mathcal{X}$ by successive applications of RGP iterations will be called an RGP algorithm. We now describe three example RGP algorithms, the first two of which we have encountered earlier. The issue of stepsize rules will be addressed in the next section.

Example 4.1. Gradient projection algorithm. By (4.2), the gradient projection algorithm is an RGP algorithm with $\tau_1 = 1$, $\tau_2 = \infty$, and p an optimal multiplier vector associated with $Bx = c$ in the problem of projecting $x/\alpha - \nabla f(x)$ onto $\mathcal{X}$.

Example 4.2. Bertsekas algorithm. By (4.6) and (4.7), the Bertsekas algorithm (4.3)–(4.5) is an RGP algorithm with $\tau_1 = \sqrt{n}$, $\tau_2 = \infty$, and $p = \min_k \nabla_k f(x)$.

Example 4.3. An active-set-type algorithm. Consider the following algorithm for solving (1.1), under the condition (4.1): Fix any $\gamma > 0$. Given an iterate $x \in \mathcal{X}$, we choose a positive stepsize α and a (possibly empty) subset $J \subseteq \{\, j \in \{1, \ldots, n\} \mid x_j \geq \gamma \,\}$ with B_J having full column rank, and we compute a new iterate x' as the (unique) solution of a convex quadratic program, given by

$$(4.11)\quad x' = \arg \min_{\substack{\xi \text{ with } B\xi = c \\ \xi_k \geq 0 \ \forall k \notin J}} \sum_{k \in J} \nabla_k f(x)(\xi_k - x_k) + \frac{1}{2\alpha} \sum_{k \notin J} |\xi_k - (x_k - \alpha \nabla_k f(x))|^2.$$

We will show that the iteration (4.11) is well defined and the x' thus generated, together with some p, satifies (4.8)–(4.10) for some scalar constants τ_1 and τ_2.

The above algorithm may be viewed as a generalization of the gradient projection algorithm in which projection is omitted for coordinates that are far from the boundary. In particular, if we take J to be the empty set, then we recover the gradient projection algorithm (see Example 4.1). A key advantage of the algorithm is its flexibility. For example, we can choose the set J so that the work in solving (4.11) is less than that for performing the full projection (see discussions to follow). The parameter γ, however, needs to be chosen with care. If γ is too large, the choices for J would be restricted; if γ is small, then, as we shall see, α may need to be small (cf. (4.14)), in which case the algorithm would take small steps. Finally, we note that γ need not be fixed but can be adjusted dynamically, provided that it remains bounded away from zero.

We now show that the iteration (4.11) is a well-defined RGP iteration. If J is the empty set, then (4.11) reduces to a gradient projection iteration, so it is well defined and the x' generated by it, together with some p, satisfies (4.8)–(4.10) with $\tau_1 = 1$ and $\tau_2 = \infty$ (cf. Example 4.1). Thus, it remains to prove the above assertion for the case where J is nonempty. First, notice that the feasible set for the minimization in (4.11) is nonempty (since it contains $\mathcal{X}$) and bounded (since the objective function is strongly convex in $\xi_{\bar{J}}$ and, by virtue of B_J having full column rank, ξ_J is determined uniquely by $\xi_{\bar{J}}$ on the feasible set). Thus, the minimization in (4.11) has an optimal solution. It is easily seen that this optimal solution is unique, so (4.11) is well defined. From the optimality conditions for the minimization in (4.11) we have that $Bx' = c$ and

$$(4.12)\qquad x'_{\bar{J}} = [x_{\bar{J}} - \alpha(\nabla_{\bar{J}} f(x) - B_{\bar{J}}^T p)]_+, \qquad \nabla_J f(x) = B_J^T p,$$

where p is any optimal Lagrange multiplier vector associated with the constraints $B\xi = c$ in (4.11). The former, together with the fact $Bx = c$, implies $0 = B(x' - x) = B_J(x'_J - x_J) + B_{\bar{J}}(x'_{\bar{J}} - x_{\bar{J}})$ so, multiplying both sides by B_J^T and using the fact that B_J has full column rank, we can solve for $x'_J - x_J$ to obtain

$$x'_J - x_J = -(B_J^T B_J)^{-1} B_J^T B_{\bar{J}}(x'_{\bar{J}} - x_{\bar{J}}),$$

implying

$$(4.13)\qquad \|x'_J - x_J\| \leq \|(B_J^T B_J)^{-1} B_J^T B_{\bar{J}}\| \|x'_{\bar{J}} - x_{\bar{J}}\|.$$

Relations (4.12) and (4.13) show that x', together with p, satisfies (4.8) and (4.9) with $\tau_1 = 1 + \|(B_J^T B_J)^{-1} B_J^T B_{\bar{J}}\|$.

It only remains to show that x' satisfies (4.10) for some scalar constant τ_2. For any subset I of $\{1, \ldots, m\}$ and any subset J of $\{1, \ldots, n\}$, let B_{IJ} denote the matrix obtained by removing from B_J all rows i with $i \notin I$. We show below that $x' \in \mathcal{X}$ whenever

$$\alpha \leq \frac{\min_{k \in J}\{x_k\}}{\|(B_J^T B_J)^{-1} B_J^T B_{\bar{J}}\| \|\nabla_{\bar{J}} f(x) - B_{I\bar{J}}^T (B_{IJ}^T)^{-1} \nabla_J f(x)\|}, \tag{4.14}$$

where I is any subset of $\{1, \ldots, m\}$ such that B_{IJ} is invertible. This, together with the fact $x_j \geq \gamma$ for all $j \in J$, would then complete the proof. First, we observe that the constraints $B\xi = c$ can be rewritten as $B_{IJ}\xi_J + B_{I\bar{J}}\xi_{\bar{J}} = c_I$ and $B_{\tilde{I}J}\xi_J + B_{\tilde{I}\bar{J}}\xi_{\bar{J}} = c_{\tilde{I}}$, where $\tilde{I}$ is the complement of I relative to $\{1, \ldots, m\}$. Using the first set of constraints to eliminate ξ_J from the second set and from the objective function in (4.11), we reduce the minimization in (4.11) to the following problem:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2\alpha} \sum_{k \notin J} |\xi_k - (x_k - \alpha \nabla_k f(x))|^2 - \langle \nabla_J f(x), (B_{IJ})^{-1} B_{I\bar{J}} \xi_{\bar{J}} \rangle \\ &\text{subject to} \quad \left(B_{\tilde{I}\bar{J}} - B_{\tilde{I}J}(B_{IJ})^{-1} B_{I\bar{J}}\right) \xi_{\bar{J}} = c_{\tilde{I}} - B_{\tilde{I}J}(B_{IJ})^{-1} c_I, \qquad \xi_{\bar{J}} \geq 0, \end{aligned}$$

to which $x'_{\bar{J}}$ is an optimal solution. Then, $x'_{\bar{J}}$ satisfies the optimality conditions:

$$x'_{\bar{J}} = [x_{\bar{J}} - \alpha(\nabla_{\bar{J}} f(x) - B_{I\bar{J}}^T (B_{IJ}^T)^{-1} \nabla_J f(x))]_{\mathcal{D}}^+,$$

where $\mathcal{D}$ denotes the feasible set for the reduced problem. This, combined with the observation that $x_{\bar{J}} \in \mathcal{D}$ (cf. $x \in \mathcal{X}$), implies

$$\begin{aligned} \|x'_{\bar{J}} - x_{\bar{J}}\| &= \|[x_{\bar{J}} - \alpha(\nabla_{\bar{J}} f(x) - B_{I\bar{J}}^T (B_{IJ}^T)^{-1} \nabla_J f(x))]_{\mathcal{D}}^+ - [x_{\bar{J}}]_{\mathcal{D}}^+\| \\ &\leq \alpha \|\nabla_{\bar{J}} f(x) - B_{I\bar{J}}^T (B_{IJ}^T)^{-1} \nabla_J f(x)\|, \end{aligned}$$

where the last inequality follows from the nonexpansive property of the projection mapping $[\cdot]_{\mathcal{D}}^+$. Combining this with (4.13) gives

$$\|x'_J - x_J\| \leq \alpha \|(B_J^T B_J)^{-1} B_J^T B_{\bar{J}}\| \|\nabla_{\bar{J}} f(x) - B_{I\bar{J}}^T (B_{IJ}^T)^{-1} \nabla_J f(x)\|,$$

and it follows that $x'_J \geq 0$ whenever α satisfies (4.14). Since $Bx' = c$ and (cf. (4.12)) $x'_{\bar{J}} \geq 0$, this shows that $x' \in \mathcal{X}$ (cf. (1.6) and (4.1)) whenever α satisfies (4.14).

The iteration (4.11) admits an interesting interpretation as an active-set-type iteration. To see this, let us assume for simplicity that the matrix B_J therein is invertible. Then, since $\nabla_J f(x) = B_J^T p$ (cf. (4.12)), we can eliminate p from the first expression in (4.12) to obtain

$$x'_{\bar{J}} = [x_{\bar{J}} - \alpha(\nabla_{\bar{J}} f(x) - B_{\bar{J}}^T (B_J^T)^{-1} \nabla_J f(x))]_+.$$

Also, since $Bx' = c$, we can solve for x'_J to obtain

$$x'_J = (B_J)^{-1}(c - B_{\bar{J}} x'_{\bar{J}}).$$

Thus we may interpret (4.11) as an iteration in which we first take a reduced-gradient projection step, and then we adjust those coordinates for which the reduced gradient is zero so that the new iterate x' satisfies $Bx' = c$. This philosophy of taking a descent step with respect to those coordinates "active" at their respective bounds (i.e., $x_{\bar{J}}$) is reminiscent of active set schemes for solving problems with simple bounds. In fact, it can be seen that the above iteration is very similar to an unscaled version of a projected Newton method studied by Bertsekas [Ber82, §3] and Bertsekas and Gafni [BeG83]. In contrast to conventional active set schemes, the above scheme has the advantage that it can add and drop many elements from its currently active set $\bar{J}$ at each iteration.

5. Convergence of RGP algorithms. In this section we show, by using the local error bound of §3, that every RGP algorithm with the stepsizes chosen according to an Armijo-like rule is linearly convergent. The proof of this is analogous to a proof given in [LuT92b].

First, we describe the rule for choosing the stepsizes α. This rule is based on the efficient Armijo-like rule proposed by Bertsekas for the gradient projection algorithm [Ber76]. Let τ_1 and τ_2 be the parameters of a given RGP iteration (cf. (4.9) and (4.10)). We fix two parameters $\beta \in (0,1)$ and $\tau_3 > 0$ and we let

$$\tau_4 = \tfrac{1}{2}\|E^T\|\|E\|\rho(\tau_1)^2 + \tau_3.$$

Given an iterate $x \in \mathcal{X}$, we choose a number α_0 with $\alpha_0 \geq \min\{1/\tau_4, \tau_2/\|\nabla f(x)\|\}$ and we set

$$\alpha = \alpha_0\beta^k, \tag{5.1}$$

where k is the first nonnegative integer for which an x' and a p generated by the RGP iteration with α given as above (i.e., x' and p together satisfy (4.8)–(4.10)) satisfies $x' \in \mathcal{X}$ and the sufficient descent condition

$$f(x) - f(x') \geq \tau_3\alpha\|x - [x - \nabla f(x) + B^Tp]_+\|^2. \tag{5.2}$$

We remark that, instead of the Armijo-like rule given above, we can also use a stepsize rule analogous to one proposed by Goldstein [Gol74] and the analysis can be adapted accordingly.

We next show that the stepsize rule (5.1)–(5.2) is well defined and that the stepsize generated is sufficiently large.

LEMMA 5.1. *The stepsize rule* (5.1)–(5.2) *is well defined. Moreover, the stepsize* α *generated by this rule is bounded below by* $\beta\min\{1/\tau_4, \tau_2/\|\nabla f(x)\|\}$.

Proof. First, we show that, for a given $x \in \mathcal{X}$ and a positive number α strictly less than $\min\{1/\tau_4, \tau_2/\|\nabla f(x)\|\}$, any x' and any $p \in \Re^l$ that together satisfy (4.8)–(4.10) also satisfy $x' \in \mathcal{X}$ and (5.2). Since ∇f is Lipschitz continuous with Lipschitz constant $\|E^T\|\|E\|\rho$ (cf. (2.2)), we have

$$f(x) - f(x') \geq \langle \nabla f(x), x - x'\rangle - \frac{\|E^T\|\|E\|\rho}{2}\|x' - x\|^2. \tag{5.3}$$

Let $J = \{j \in \{1,\ldots,n\} \mid B_j^Tp = \nabla_j f(x)\}$. Then, by (4.8), $x'_{\bar{J}}$ is the orthogonal projection of $x_{\bar{J}} - \alpha(\nabla_{\bar{J}} f(x) - B_{\bar{J}}^Tp)$ onto the nonnegative orthant. Since $x \geq 0$, this implies

$$\langle x'_{\bar{J}} - x_{\bar{J}} + \alpha(\nabla_{\bar{J}} f(x) - B_{\bar{J}}^Tp), x_{\bar{J}} - x'_{\bar{J}}\rangle \geq 0.$$

Since $Bx = Bx'$ (cf. $x \in \mathcal{X}$ and $x' \in \mathcal{X}$), we have from the definition of J and the above relation that

$$
\begin{aligned}
\langle \nabla f(x), x - x' \rangle &= \langle \nabla f(x) - B^T p, x - x' \rangle \\
&= \langle \nabla_{\bar{J}} f(x) - B_{\bar{J}}^T p, x_{\bar{J}} - x'_{\bar{J}} \rangle \\
&\geq \frac{1}{\alpha} \|x_{\bar{J}} - x'_{\bar{J}}\|^2.
\end{aligned} \tag{5.4}
$$

Upon combining (5.3) with (5.4), we obtain

$$f(x) - f(x') \geq \frac{1}{\alpha}\|x_{\bar{J}} - x'_{\bar{J}}\|^2 - \frac{\|E^T\|\|E\|\rho}{2}\|x - x'\|^2,$$

so (4.8), (4.9) together with the definitions of J and τ_4 yield

$$f(x) - f(x') \geq \left(\frac{1}{\alpha} - \tau_4 + \tau_3\right)\|x - [x - \alpha(\nabla f(x) - B^T p)]_+\|^2.$$

Since $\|x - [x - \alpha d]_+\| \geq \alpha \|x - [x - d]_+\|$ for any $d \in \Re^n$ (see, for example, Lemma 1 in [GaB84]), this shows

$$f(x) - f(x') \geq (1 - \tau_4\alpha + \tau_3\alpha)\|x - [x - \nabla f(x) + B^T p]_+\|^2.$$

Thus x' together with p satisfies (5.2) whenever α is less than $1/\tau_4$. Since x' satisfies (4.10), we also have that $x' \in \mathcal{X}$ whenever α is less than $\tau_2/\|\nabla f(x)\|$.

The above result implies that, for a given $x \in \mathcal{X}$, if the integer k is sufficiently large, then any x' and p satisfying (4.8)–(4.10), with α given by (5.1), also satisfies $x' \in \mathcal{X}$ and (5.2). There must be a first k for which this occurs, so the stepsize rule (5.1)–(5.2) is well defined. Now we prove the second claim. Let $\bar{\alpha}$ be the stepsize given by this rule. Then, either $\bar{\alpha} = \alpha_0$ or $\bar{\alpha} < \alpha_0$. In the former case the second claim holds trivially (by choice of α_0). In the latter case, there must exist some x' and p satisfying (4.8)–(4.10), with α set to $\bar{\alpha}/\beta$, such that either $x' \notin \mathcal{X}$ or (5.2) fails to hold. By the result proven above, this means that $\bar{\alpha}/\beta$ must be greater than or equal to $\min\{1/\tau_4, \tau_2/\|\nabla f(x)\|\}$ or, equivalently, $\bar{\alpha}$ is greater than or equal to β times the latter quantity. The second claim then follows. □

Our final lemma bounds the cost difference $f(x') - v^*$ in terms of the *inexact* residual $x - [x - \nabla f(x) + B^T p]^+$. This bound is analogous to the cost bounds used in the convergence analysis of gradient projection methods (see [Dun87, eq. (23)], [GaD88, Lemmas 2 and 3], and [LuT92b, Thms. 2.1 and 3.1]).

LEMMA 5.2. *Fix any $v \geq v^*$ and let ϵ be the corresponding scalar given in Theorem* 3.2. *For any $x \in \mathcal{X}$, any $p \in \Re^l$, and any $x' \in \mathcal{X}$ satisfying $f(x) \leq v$, $\|x - [x - \nabla f(x) + B^T p]_+\| \leq \epsilon$ and* (4.8)–(4.9), *we have*

$$f(x') - v^* \leq \tau_5\left(1 + \frac{1}{\alpha}\right)\|x - [x - \nabla f(x) + B^T p]_+\|^2,$$

where $\tau_5 > 0$ is some scalar constant depending on v and the problem data only.

Proof. Fix any x, x', and p satisfying the hypothesis of the lemma. Let $z = [x - \nabla f(x) + B^T p]^+$. Then, $(x,p) \in \mathcal{F}_0^v \times \Re^l$ and $\|x - z\| \leq \epsilon$, so (x,p) satisfies the hypothesis of Theorem 3.2. Upon invoking Theorem 3.2, we have that there exists some $x^* \in \mathcal{X}^*$ such that

$$\|x - x^*\| \leq \kappa\|x - z\|, \tag{5.5}$$

where κ is the scalar in Theorem 3.2.

Since $Bx' = Bx^*$, then

$$\begin{aligned}\langle \nabla f(x), x' - x^* \rangle &= \langle \nabla f(x) - B^T p, x' - x^* \rangle \\ &= \langle \nabla_J f(x) - B_J^T p, x'_J - x^*_J \rangle,\end{aligned}$$

where we let $J = \{j \in \{1, \ldots, n\} \mid B_j^T p = \nabla_j f(x)\}$. Since $x'_{\bar{J}}$ is the orthogonal projection of $x_{\bar{J}} - \alpha(\nabla_{\bar{J}} f(x) - B_{\bar{J}}^T p)$ onto the nonnegative orthant (cf. (4.8)) and $x^*_{\bar{J}} \geq 0$, we also have

$$\langle x'_{\bar{J}} - x_{\bar{J}} + \alpha(\nabla_{\bar{J}} f(x) - B_{\bar{J}}^T p), x'_{\bar{J}} - x^*_{\bar{J}} \rangle \leq 0,$$

which, when combined with the previous relation, yields

$$\langle \nabla f(x), x' - x^* \rangle \leq \frac{1}{\alpha} \langle x_{\bar{J}} - x'_{\bar{J}}, x'_{\bar{J}} - x^*_{\bar{J}} \rangle.$$

Also, by the Mean Value Theorem, there exists some ζ lying on the line segment joining x' with x^* such that

$$f(x') - f(x^*) = \langle \nabla f(\zeta), x' - x^* \rangle.$$

Summing the above two relations and rearranging terms give

$$\begin{aligned}f(x') - f(x^*) &\leq \langle \nabla f(\zeta) - \nabla f(x), x' - x^* \rangle + \frac{1}{\alpha} \langle x_{\bar{J}} - x'_{\bar{J}}, x'_{\bar{J}} - x^*_{\bar{J}} \rangle \\ &\leq \left(\|\nabla f(\zeta) - \nabla f(x)\| + \frac{1}{\alpha}\|x - x'\| \right) \|x' - x^*\| \\ &\leq \left(\rho \|E^T\| \|E\| \|\zeta - x\| + \frac{1}{\alpha}\|x - x'\| \right) \|x' - x^*\| \\ &\leq \left(\rho \|E^T\| \|E\| \|x^* - x\| + \frac{1}{\alpha}\|x - x'\| \right) \|x' - x^*\|,\end{aligned}$$

where the third inequality follows from the Lipschitz continuity property of ∇f (cf. (2.2)). Using (5.5) and the fact $\|x - x'\| \leq \tau_1 \|x - z\|$ (cf. (4.9)) to bound the right-hand side of the above relation completes our proof. □

Upon using Lemmas 5.1 and 5.2, we can now establish the linear rate of convergence for RGP algorithms employing the Armijo-like stepsize rule.

THEOREM 5.3 (linear convergence). *Let* $\{x^0, x^1, \ldots\}$ *be a sequence in* $\mathcal{X}$ *generated by a* RGP *algorithm* (*cf.* (4.8)–(4.10)) *using the Armijo-like stepsize rule* (*cf.* (5.1)–(5.2)). *Then,* $\{x^r\}$ *converges at least linearly to an element of* $\mathcal{X}^*$ *and* $\{f(x^r)\}$ *converges at least linearly to* v^*.

Proof. For each index $r \geq 0$, let α^r and p^r denote, respectively, the stepsize and the multiplier vector associated with the generation of x^{r+1} by the RGP algorithm using the Armijo-like stepsize rule. In other words, the conditions (4.8)–(4.9) and (5.1)–(5.2), as well as $x' \in \mathcal{X}$, are satisfied by $x = x^r$, $x' = x^{r+1}$, $\alpha = \alpha^r$, and $p = p^r$ for every r. By (5.2), we have

$$f(x^r) - f(x^{r+1}) \geq \tau_3 \alpha^r \|x^r - [x^r - \nabla f(x^r) + B^T p^r]_+\|^2, \quad \forall r, \tag{5.6}$$

and, by Lemma 5.1, we have

$$\alpha^r \geq \beta \min\{1/\tau_4, \tau_2/\|\nabla f(x^r)\|\}, \quad \forall r. \tag{5.7}$$

Relation (5.6) implies $f(x^r) \leq f(x^0)$ for all r. Since in addition $x^r \in \mathcal{X}$ for all r, we obtain from (1.6) that $x^r \in \mathcal{F}_0^v$ for all r where we let $v = f(x^0)$. Then, Lemma 2.3 implies that the sequence $\{Ex^r\}$ is bounded. Since ∇g is continuous, this in turn implies that $\{\nabla g(Ex^r)\}$ is bounded, so that (cf. (2.1)) $\{\nabla f(x^r)\}$ is bounded. Combining this with (5.7), we see that $\{\alpha^r\}$ is bounded below by some positive scalar constant.

Since $\{\alpha^r\}$ is bounded away from zero and f is bounded below on $\mathcal{X}$, the relation (5.6) implies $x^r - [x^r - \nabla f(x^r) + B^T p^r]_+ \to 0$. Then, by Lemma 5.2, there exist a scalar constant $\tau_5 > 0$ and an index $\bar{r}$ such that

$$\|x^r - [x^r - \nabla f(x^r) + B^T p^r]_+\|^2 \geq \frac{\alpha^r}{\tau_5(1+\alpha^r)}(f(x^{r+1}) - v^*), \quad \forall r \geq \bar{r},$$

which, when combined with (5.6), yields

$$f(x^r) - f(x^{r+1}) \geq \frac{\tau_3(\alpha^r)^2}{\tau_5(1+\alpha^r)}(f(x^{r+1}) - v^*), \quad \forall r \geq \bar{r}.$$

Upon rearranging terms in the above relation, we obtain

$$f(x^{r+1}) - v^* \leq \frac{\tau_5(1+\alpha^r)}{\tau_5(1+\alpha^r) + \tau_3(\alpha^r)^2}\left(f(x^r) - v^*\right), \quad \forall r \geq \bar{r}.$$

Since $\{\alpha^r\}$ is bounded away from zero, this shows that $f(x^r) \to v^*$ at least linearly, which, together with (5.6), shows that $\|x^r - [x^r - \nabla f(x^r) + B^T p^r]_+\| \to 0$ at least linearly. Since $\|x^{r+1} - x^r\| \leq \tau_1 \|x^r - [x^r - \nabla f(x^r) + B^T p^r]_+\|$ (cf. (4.9)), it follows that $\|x^{r+1} - x^r\| \to 0$ at least linearly, so $\{x^r\}$ converges. Since $f(x^r) \to v^*$, the limit point of $\{x^r\}$ is in $\mathcal{X}^*$. □

We have just shown that any RGP algorithm using the Armijo-like stepsize rule attains a linear rate of convergence. Upon applying Theorem 5.3 to the algorithm of Bertsekas and to the active-set-type algorithm of §4, we immediately obtain the following new convergence results.

COROLLARY 5.4. *Suppose that* $C = [0, \infty)^n$, $B = [1\ 1\ \cdots\ 1]$, *and* $c = 1$. *Then, any sequence of iterates generated by the Bertsekas algorithm* (*cf.* (4.3)–(4.5)), *with stepsizes determined by the Armijo-like rule* (*cf.* (5.1)–(5.2)), *converges at least linearly to an element of* $\mathcal{X}^*$.

COROLLARY 5.5. *Suppose that* $C = [0, \infty)^n$. *Then, any sequence of iterates generated by the active-set-type algorithm* (*cf.* (4.11)), *with stepsizes determined by the Armijo-like rule* (*cf.* (5.1)–(5.2)), *converges at least linearly to an element of* $\mathcal{X}^*$.

6. Concluding remarks. In this paper, we studied a (new) local error bound for certain convex minimization problems over a polyhedral set. We then used this error bound to prove linear convergence for a class of reduced-gradient projection algorithms.

There are several directions in which our results may be generalized. We briefly describe two main ones below.

1. Problems with extended-real-valued cost function. In many situations, g is defined only on some open subset $\mathcal{G}$ of $\Re^m$ and ∇g is Lipschitz continuous and strongly monotone on any compact subset of $\mathcal{G}$. All of our results can be extended to this situation provided that, for some $\bar{v} > v^*$, the level set $\mathcal{F} = \{x \in \mathcal{X} \mid f(x) \leq \bar{v}\}$ satisfies

$$E\mathcal{F} \subseteq \mathcal{G}.$$

(Notice that the above condition holds automatically if dom g is open and g tends to ∞ at the boundary of dom g.) In particular, Theorem 3.2 still holds provided that v therein does not exceed $\bar{v}$. The proof of this is based on an interesting fact that, for $\delta > 0$ sufficiently small, $E\mathcal{F}_{\delta}^{\bar{v}}$ is a compact subset of $\mathcal{G}$, where $\mathcal{F}_{\delta}^{\bar{v}}$ is defined as in §2. (The proof of this is similar to that of Lemma 9.1 in [Tse91].) By using this fact in place of Lemma 2.3, we can verify that all the steps in the proof of Theorem 3.2 go through, provided that we take $v \leq \bar{v}$. Linear convergence of the algorithms described in §4 also holds, provided that the stepsize α is taken sufficiently small so as to ensure that each new iterate remains within $\mathcal{F}$. (The proof of the latter uses the boundedness of ∇f on $\mathcal{F}$ and the strict inclusion of $E\mathcal{F}$ by $\mathcal{G}$.)

2. Variational inequality problems. The error bound in §3 readily extends to the following variational inequality problem, first studied by Bertsekas and Gafni [BeG82], of finding an x^* satisfying

$$x^* = [x^* - F(x^*)]^+_{\mathcal{X}},$$

where $F(x) = E^TG(Ex) + q$ and $G : \Re^m \mapsto \Re^m$ is a Lipschitz continuous strongly monotone function. However, it is unclear whether the bound would help in the development of algorithms for solving such a problem. The error bound also readily extends to *affine* variational inequality problems (where F in the above problem is any affine mapping). This follows from a result of Robinson [Rob81] on certain Lipschitz continuity properties of polyhedral multifunctions.

There remain many open questions which we plan to investigate. Specifically, can the local error bound described in §3 be extended to problems with general convex constraints? Can the linear convergence result of Corollary 5.4 be extended to an asynchronous version of the Bertsekas algorithm proposed by Tsitsiklis and Bertsekas [TsB86]? Some progress along this latter direction has already been made (see [LuT91]). Are there other reduced-gradient projection algorithms, different from those described here, to which our convergence analysis can be fruitfully applied?

It was pointed out to us by one of the referees that, although RGP algorithms typically require less work per iteration than the gradient projection algorithm, their rate of convergence may be slower, thus offsetting any saving in the per iteration workload. In particular, a careful examination of the convergence analysis in §5 shows that, in the worst case, the rate of convergence of an RGP algorithm may depend on n, whereas the gradient projection algorithm does not. Does this dependence exist in practice and, if yes, what are its effects on the performance of an RGP algorithm? This is yet another question that we hope to address in the future.

Acknowledgment. We thank Professor D. P. Bertsekas and an anonymous referee for their many helpful comments, which led to a number of improvements in the presentation.

REFERENCES

[Ber76] D. P. Bertsekas, *On the Goldstein–Levitin–Poljak gradient projection method*, IEEE Trans. Automat. Control, AC21 (1976), pp. 174–184.

[Ber80] ———, *A class of routing algorithms for communication networks*, in Proc. Fifth Internat. Conf. Comput. Commun., Atlanta, GA, 1980, pp. 71–76.

[Ber82] ———, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.

[BeG82] D. P. Bertsekas and E. M. Gafni, *Projection methods for variational inequalities with application to the traffic assignment problem*, in Math. Programming Stud., D. C. Sorensen and R. J.-B. Wets, eds., 17 (1982), pp. 139–159.

[BeG83] D. P. Bertsekas and E. M. Gafni, *Projected Newton methods and optimization of multicommodity flows,* IEEE Trans. Automat. Control, AC28 (1983), pp. 1090–1096.

[BeG87] D. P. Bertsekas and R. Gallager, *Data Networks*, Prentice–Hall, Englewood Cliffs, NJ, 1987.

[BeT89] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice–Hall, Englewood Cliffs, NJ, 1989.

[Dun81] 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.

[Dun87] ———, *On the convergence of projected gradient processes to singular critical points*, J. Optim. Theory Appl., 55 (1987), pp. 203–216.

[GaB84] E. M. Gafni and D. P. Bertsekas, *Two-metric projection methods for constrained optimization,* SIAM J. Control Optim., 22 (1984), pp. 936–964.

[GaD88] M. Gawande and J. C. Dunn, *Variable metric gradient projection processes in convex feasible sets defined by nonlinear inequalities,* Appl. Math. Optim., 17 (1988), pp. 103–119.

[Gol64] A. A. Goldstein, *Convex programming in Hilbert space*, Bull. Amer. Math. Soc., 70 (1964), pp. 709–710.

[Gol74] ———, *On gradient projection*, in Proc. 12th Allerton Conference Circuits and System Theory, Univ. of Illinois, Allerton Park, IL, 1974, pp. 38–40.

[Hof52] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 263–265.

[LeP65] E. S. Levitin and B. T. Poljak, *Constrained minimization methods,* Z. Vychisl. Mat. i Mat. Fiz., 6 (1965), pp. 787–823. (In Russian.) Translation in USSR Comput. Math. and Math. Phys., 6 (1965), pp. 1–50.

[LuT91] Z.-Q. Luo and P. Tseng, *On the rate of convergence of a class of distributed asynchronous routing algorithms*, Tech. Rep., Dept. of Electrical and Computer Engineering, McMaster Univ., Hamilton, Ontario and Dept. of Mathematics, Univ. of Washington, Seattle, WA, May 1991.

[LuT92a] ———, *On the convergence of the coordinate descent method for convex differentiable minimization*, J. Optim. Theory Appl., 72 (1992), pp. 7–35.

[LuT92b] ———, *On the linear convergence of descent methods for convex essentially smooth minimization,* SIAM J. Control Optim., 30 (1992), pp. 408–425.

[LuT92c] ———, *Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem*, SIAM J. Optimization, 2 (1992), pp. 43–54.

[MaD88] 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.

[MaS87] 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.

[Mor89] J. J. Moré, *Gradient projection techniques for large-scale optimization problems,* in Proc. 28th Conf. Decision and Control, Tampa, FL, December 1989.

[OrR70] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[Pan87] J.-S. Pang, *A posteriori error bounds for the linearly-constrained variational inequality problem*, Math. Oper. Res., 12 (1987), pp. 474–484.

[Rob73] S. M. Robinson, *Bounds for error in the solution set of a perturbed linear program*, Linear Algebra Appl., 6 (1973), pp. 69–81.

[Rob81] ———, *Some continuity properties of polyhedral multifunctions*, Math. Programming Stud., 14 (1981), pp. 206–214.

[Rob82] ———, *Generalized equations and their solutions, part* II: *Applications to nonlinear programming*, Math. Programming Stud., 14 (1982), pp. 200–221.

[Roc70] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[Tsa89] W. K. Tsai, *Convergence of gradient projection routing methods in an asynchronous stochastic quasi-static virtual circuit network*, IEEE Trans. Automat. Control, AC34 (1989), pp. 20–33.

[Tse91] P. Tseng, *Descent methods for convex essentially smooth minimization,* J. Optim. Theory Appl., 71 (1991), pp. 425–463.

[TsB86] J. N. Tsitsiklis and D. P. Bertsekas, *Distributed asynchronous optimal routing in data networks*, IEEE Trans. Automat. Control, AC31 (1986), pp. 325–332.

SIAM J. OPTIMIZATION
Vol. 3, No. 1, pp. 60–80, February 1993

BLACK-BOX COMPLEXITY OF LOCAL MINIMIZATION*

STEPHEN A. VAVASIS†

Abstract. The complexity of local minimization in the black-box model, that is, the model in which the objective function and its gradient are available as external subroutines, is studied. The black-box model is used, for example, in all the optimization algorithms in Dennis and Schnabel [*Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983]. The first main result is that the complexity grows polynomially with the number of variables n, in contrast to other related black-box problems (global minimization and Brouwer fixed points) for which the worst-case complexity is exponential in n.

The second contribution is the construction of a family of functions that are bad cases for all possible black-box local optimization algorithms.

Key words. optimization, local optimality, black-box model, information-based, complexity

AMS(MOS) subject classifications. 90C60, 65K10, 65Y20, 68Q25

1. Black-box model.

Numerical optimization refers to the problem of minimizing a continuous function $f : D \rightarrow \mathbf{R}$ where D is a subset of $\mathbf{R}^n$. For nonconvex problems, most optimization algorithms will not return global minima; instead, they will return (at best) local minima.

It is therefore natural to inquire about the complexity of local minimization for general nonconvex objective functions. In order to make general statements about local optimization, it is necessary to have definitions of valid objective functions and of "approximate" local minima. These definitions will be the subject of most of this introduction. To our knowledge, this paper is the first attempt to define approximate local minimization.

The remainder of the paper is organized as follows. In §2 we present the first main result of this paper, that is, a simple algorithm to find an approximate local minimum. Its running time is polynomial in n (the number of variables) and M/ϵ (see below for an explanation). In §3 we present the second main result, a family of functions that constitutes a bad case for minimization algorithms. These functions lead to a lower bound that is polynomial in M/ϵ. In §5 we give an algorithm with a better bound for some values of the parameters. In §6 we compare our bounds to the bounds known for global minimization and Brouwer fixed points (a closely related problem).

The model of computation will be a real-number model. We assume that the algorithm can store and compute exact real numbers. We assume that the objective function f is provided by the user via a subroutine. This subroutine takes as input a vector $\mathbf{x} \in \mathbf{R}^n$ and returns a real number $f(\mathbf{x})$. We assume for this work that f is continuously differentiable. We assume that the gradient ∇f is also available as a subroutine (see further remarks on this in §2). Some of the algorithms for unconstrained problems that fall into this category are the steepest descent method, the Powell-symmetric-Broyden method, the Broyden–Fletcher–Goldfarb–Shanno method, and the line-search and trust-region modifications of these algorithms. See Dennis and Schnabel [1] for more information.

* Received by the editors August 7, 1990; accepted for publication (in revised form) October 23, 1991. This work was supported in part by the Applied Mathematical Sciences Program (KC-04-02) of the Office of Energy Research, U. S. Department of Energy, under grant DE-FG02-86ER25013.A000.

† Department of Computer Science, Cornell University, Ithaca, New York 14853.

This model of computation is known as a "black-box" model, a "function-evaluation" model, or an "oracle" model. The key limiting feature is that global information about f is not available to the minimization algorithm (unlike, for instance, the special case of quadratic programming).

Because our focus is on the objective function rather than the constraints of the problem, we will assume the simple case that the domain of f is the n-dimensional unit cube denoted by I^n (the n-fold Cartesian product of the interval $I = [0, 1]$). It would perhaps be easier to assume simply that f is unconstrained (i.e., the domain is $\mathbf{R}^n$), but this leads to difficulties of scale as well as to the problem that local minima might not exist. Since I^n is compact, there is always a global (and hence a local) minimum.

An algorithm to find a local minimum takes as input a function f and its gradient ∇f as black-box subroutines. It must repeatedly evaluate f and ∇f at points in I^n until it has found a local minimum. It is easy to see that in the real-number function-evaluation model, there will always be some uncertainty about the exact position of the local minimum. Accordingly it is useful to define approximate local minima.

Recall that $\mathbf{x}^* \in I^n$ is said to be a *global* minimum of f if $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x} \in I^n$. The point $\mathbf{x}^*$ is said to be a *local* minimum of f if there exists an open set N containing $\mathbf{x}^*$ such that $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x} \in N \cap I^n$.

DEFINITION. A point $\mathbf{x}^* \in I^n$ is said to be an *ϵ-approximate local minimum* of a continuous function $f : I^n \to \mathbf{R}$ if there exists an open set N containing $\mathbf{x}^*$ such that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) + \epsilon\|\mathbf{x} - \mathbf{x}^*\|$$

for all $\mathbf{x} \in N \cap I^n$.

Below we give an alternate characterization of this definition. First, we explain this definition and also point out its shortcomings. The motivation for this definition is that while $\mathbf{x}^*$ may not have the smallest function value in the neighborhood N, the value of f decreases slowly (at a rate no faster than ϵ) as one moves away from $\mathbf{x}^*$.

The most obvious shortcoming of this definition is that an interior local maximum or interior saddle point would also qualify as an ϵ-approximate local minimum under this definition. We do not feel that this property is a severe flaw in the definition, however. For example, examining the local minimization algorithms of Dennis and Schnabel, we see that it is possible for these algorithms to converge to saddle points. Indeed, distinguishing local minima from other kinds of stationary points in general is a computationally difficult problem; see, for example, Murty and Kabadi [3].

We observe that it is required to select a norm in the above definition. For this paper we will assume that the one-norm is used in that definition. The norms in this paper have been selected to make the analysis simple.

We now give an alternative characterization of an approximate local minima. We will say that $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$ is an *ϵ-KKT point* of $f : I^n \to \mathbf{R}$ if

1. For all i such that $x_i^* > 0$, $\partial f/\partial x_i(\mathbf{x}^*) \leq \epsilon$.
2. For all i such that $x_i^* < 1$, $\partial f/\partial x_i(\mathbf{x}^*) \geq -\epsilon$.

(Note that if $\epsilon = 0$ these conditions are the KKT (Karush–Kuhn–Tucker) necessary conditions for local optimality.) If $\mathbf{x}^*$ is interior, these conditions are equivalent to the requirement that $\|\nabla f(\mathbf{x}^*)\|_\infty \leq \epsilon$.

If f is continuously differentiable, then there is a close connection between its ϵ-KKT points and its ϵ-local minima, as proved by the following lemma.

LEMMA 1.1. *Suppose $f : I^n \to \mathbf{R}$ is C^1. If $\mathbf{x}^* \in I^n$ is an ϵ-approximate local minimum of f, then it is an ϵ-*KKT *point. Conversely, if $\mathbf{x}^*$ is an ϵ-*KKT *point, then it is an ϵ'-local minimum for all $\epsilon' > \epsilon$.*

Proof. We start with the first claim made by the lemma. We verify condition 1 in the definition of ϵ-KKT point for a particular index i (condition 2 is similar). Assuming $x_i^* > 0$, the point $\mathbf{x}^* - t\mathbf{e}_i$ is feasible for small enough $t > 0$, where $\mathbf{e}_i$ is the ith column of the identity matrix. For small enough t, this point is contained in $N \cap I$, hence $f(\mathbf{x}^*) - f(\mathbf{x}^* - t\mathbf{e}_i) \leq t\epsilon$ by definition of approximate local minimum. Since this holds for all t small enough, by definition of the partial derivative this implies $\partial f/\partial x_i(\mathbf{x}^*) \leq \epsilon$.

To prove the second statement of the lemma, recall that the definition of a derivative is that for all $\mathbf{d}$,

$$f(\mathbf{x}^* + \mathbf{d}) = f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T\mathbf{d} + o(\|\mathbf{d}\|).$$

Suppose that $x_i^* > 0$ for some i; then we know

$$f(\mathbf{x}^* - t\mathbf{e}_i) = f(\mathbf{x}^*) - t\frac{\partial f}{\partial x_i}(\mathbf{x}^*) + o(t)$$

so

$$f(\mathbf{x}^* - t\mathbf{e}_i) \geq f(\mathbf{x}^*) - t\epsilon + o(t)$$

so

$$f(\mathbf{x}^* - t\mathbf{e}_i) \geq f(\mathbf{x}^*) - t\epsilon'$$

for all t small enough. This inequality holds not only for $\mathbf{x}^*$ but for every $\mathbf{x}$ in a neighborhood of $\mathbf{x}^*$ since we are assuming that f is continuously differentiable. Then we see that we can get a lower bound on $f(\mathbf{x}^* + \mathbf{d})$ for an arbitrary $\mathbf{d}$ that is small enough by expressing $\mathbf{d}$ as a sum of small steps of the form $t_i\mathbf{e}_i$. □

We next ask the question: Given a continuously differentiable function $f : I^n \rightarrow \mathbf{R}$ and given a number $\epsilon > 0$, what is the complexity of finding an ϵ-approximate local minimum? It turns out that the number of steps required is infinite. In particular, for any finite sequence of test points $x_1, \ldots, x_k$, there exists a continuously differentiable function $f : [0,1] \rightarrow \mathbf{R}$ such that $f(x_i) = 0$ and $f'(x_i) = 1$ at all test points (except if $x_i = 0$ then $f'(0) = -1$). Moreover, $-1 \leq f'(x) \leq 1$ for all $x \in [0,1]$. Figure 1 illustrates an example of a sequence of test points and the bad-case function for these points. To construct this function, put the x_i's into increasing order, and then let f be a correctly chosen cubic Hermite function on each interval.

An algorithm trying to find approximate local minima for this family of functions will always completely fail (i.e., it will discover that $f(x) = 0$ and $f'(x) = 1$ at all of its test points) for at least one function in the family after any finite number of steps.

The problem with this family of functions is that the first derivatives can vary too much over short intervals, so that no algorithm can get a bound on the first derivative of the function.

Accordingly, we place additional restrictions on the function. In particular, we require that the first derivative satisfy a *Lipschitz condition*, that is, there exists a constant M such that

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_\infty \leq M\|\mathbf{x} - \mathbf{y}\|_\infty$$

for all $\mathbf{x}, \mathbf{y} \in I^n$.

We now ask the question: What is the complexity of finding an ϵ-approximate local minimum for a function in this class? Clearly the answer depends on ϵ, M, and

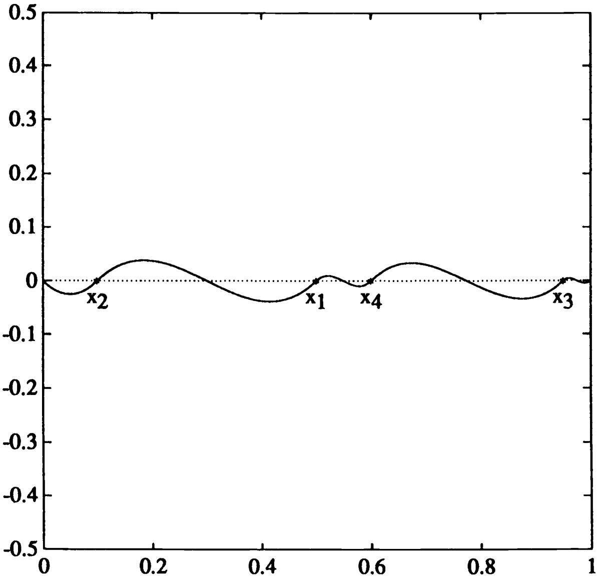


FIG. 1. *An impossible case for local minimization.*

n. In the next section, we give an algorithm for this problem, which yields an upper bound on the complexity.

We remark that none of our complexity bounds depend on M or ϵ individually; instead, they all depend on the ratio M/ϵ. This is expected because the problem of finding an ϵ-approximate local minimum for f is the same problem as finding a $c\epsilon$-approximate local minimum for cf (where $c > 0$). Therefore, we would expect the complexity to be unchanged if M and ϵ are scaled by the same amount.

2. An algorithm for local minimization. In this section we propose an algorithm for approximate local minimization, along with a complexity analysis. We call this algorithm LOCAL1. We assume that ϵ, M, and n are given. We assume also that M/ϵ is an integer. We are given a starting point $\mathbf{x}^{(0)} \in I^n$, which is assumed to have each coordinate equal to an integer multiple of ϵ/M. If no starting point is given, the origin can be used.

Given a function $f(\mathbf{x})$, we define the vector-valued function $\mathbf{g}(\mathbf{x})$ as follows. The ith entry of $\mathbf{g}(\mathbf{x})$ is defined by

$$g_i(\mathbf{x}) = \begin{cases} \min\left(0, \dfrac{\partial f}{\partial x_i}(\mathbf{x})\right) & \text{if } x_i = 0, \\ \dfrac{\partial f}{\partial x_i}(\mathbf{x}) & \text{if } 0 < x_i < 1, \text{ or} \\ \max\left(0, \dfrac{\partial f}{\partial x_i}(\mathbf{x})\right) & \text{if } x_i = 1. \end{cases}$$

Notice that if $\mathbf{x}$ is interior to I^n, then $\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x})$. This function $\mathbf{g}(\mathbf{x})$ could be called the "projected gradient," although this terminology is not standard.

Algorithm LOCAL1 begins by testing whether $\|\mathbf{g}(\mathbf{x}^{(0)})\|_\infty > M$. If this inequality holds, then for some i we know $|\partial f/\partial x_i(\mathbf{x}^{(0)})| > M$. Take the case in which $\partial f/\partial x_i(\mathbf{x}^{(0)}) > M$ (the negative case is similar). We claim that $\partial f/\partial x_i(\mathbf{x}) > 0$ for all $\mathbf{x} \in I^n$. This follows from the Lipschitz bound on ∇f.

This means in particular that any local minimum of f must occur on the face $T = \{\mathbf{x} \in I^n : x_i = 0\}$ of I^n. Moreover, if $f_0 : T \to \mathbf{R}$ denotes the restriction of f to T, then it suffices to find an ϵ-approximate local minimum of f_0. Therefore, we can project $\mathbf{x}^{(0)}$ onto T and work on the restricted problem. The restriction operation has the effect of deleting the ith entry from the vector $\mathbf{g}(\mathbf{x})$.

Accordingly, we can continue to reduce the dimensionality of the problem coordinate by coordinate. Therefore, without loss of generality, we can assume that our starting point satisfies $\|\mathbf{g}(\mathbf{x}^{(0)})\|_\infty \leq M$.

Let $\mathbf{x}^*$ be a global minimum of f. We can use the upper bound on $\mathbf{g}$ to derive an upper bound on the difference $f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*)$. Let $s = \|\mathbf{x}^* - \mathbf{x}^{(0)}\|_1$. Then we can construct a path made up of segments parallel to the coordinate axes from $\mathbf{x}^*$ to $\mathbf{x}^{(0)}$; the length of this path will be exactly s. Assume that the path is made up of n segments $P_1, \ldots, P_n$ such that P_i is parallel to $\mathbf{e}_i$.

Then we can write a line integral for the change in function values:

$$\begin{aligned} f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*) &= \sum_{i=1}^{n} \int_{P_i} \nabla f \cdot dl \\ &= \sum_{i=1}^{n} \int_{P_i} \sigma_i \frac{\partial f}{\partial x_i}\, dx_i, \end{aligned} \tag{1}$$

where $\sigma_i = \pm 1$ depending on the orientation of P_i with respect to $\mathbf{e}_i$. We now derive an upper bound on each integral in (1). There are two cases. In the first case, $g_i(\mathbf{x}^{(0)}) = \partial f/\partial x_i(\mathbf{x}^{(0)})$. In this case, we can apply the Lipschitz bound directly. We know that $|g_i(\mathbf{x}^{(0)})| = |\partial f/\partial x_i(\mathbf{x}^{(0)})| \leq M$. Since the distance from $\mathbf{x}^{(0)}$ to any point of P_i is at most 1 in the ∞-norm, we know that the magnitude of $\partial f/\partial x_i$ along P_i is at most $2M$. Therefore, the above integral has magnitude at most $2M$.

In the second case, $g_i(\mathbf{x}^{(0)}) \neq \partial f/\partial x_i(\mathbf{x}^{(0)})$. Examining the definition of $\mathbf{g}$, we see that there are two possible subcases: either $x_i^{(0)} = 0$ and $\partial f/\partial x_i(\mathbf{x}^{(0)}) > 0$, or $x_i^{(0)} = 1$ and $\partial f/\partial x_i(\mathbf{x}^{(0)}) < 0$. We treat the first subcase since the second subcase is analogous. Since $x_i^{(0)} = 0$ and $\partial f/\partial x_i(\mathbf{x}^{(0)}) > 0$, we know that $\partial f/\partial x_i$ cannot drop below $-M$ at any point on P_i. Moreover, we know that P_i is oriented in the negative direction with respect to $\mathbf{e}_i$, because $x_i^{(0)} = 0$. Therefore, the ith integral in the above summation is at most M (this argument does not give a lower bound, but only an upper bound is needed).

We conclude that all the integrals in (1) are at most $2M$, and hence

$$f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*) \leq 2Mn.$$

This gives an upper bound on how much the objective function can decrease.

We now return to the main part of LOCAL1 under the assumption that $\|\mathbf{g}(\mathbf{x}^{(0)})\|_\infty$ is at most M. The algorithm operates on an imaginary grid of nodes spaced ϵ/M apart in each dimension of I^n and aligned with the coordinate axes. By our earlier assumptions, there is an integer number of mesh cells in every dimension, and the initial point $\mathbf{x}^{(0)}$ is one of the mesh points.

We now use the following iteration. Assume that the current iterate is $\mathbf{x}^{(k)}$. We compute $\mathbf{g}(\mathbf{x}^{(k)})$. If $\|\mathbf{g}(\mathbf{x}^{(k)})\|_\infty < \epsilon$, then we halt. The justification for halting is as follows. If $\|\mathbf{g}(\mathbf{x}^{(k)})\|_\infty = \epsilon'$ and $\epsilon' < \epsilon$, then it is easy to verify from the definition of $\mathbf{g}$ that $\mathbf{x}^{(k)}$ is an ϵ'-KKT point and is therefore an ϵ-approximate local minimum.

Otherwise, suppose $\|\mathbf{g}(\mathbf{x}^{(k)})\|_\infty \geq \epsilon$. Then we identify a component, say $g_i(\mathbf{x}^{(k)})$, whose absolute value is at least ϵ. Say, for example, that $g_i(\mathbf{x}^{(k)}) \geq \epsilon$ (the negative case is similar). This means by definition that $\partial f/\partial x_i(\mathbf{x}^{(k)}) \geq \epsilon$ and that $x_i^{(k)} > 0$. Then we set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\epsilon/M)\mathbf{e}_i$. If $g_i(\mathbf{x}^{(k)})$ had been negative then we would have instead added $(\epsilon/M)\mathbf{e}_i$. Notice that, under this definition, $\mathbf{x}^{(k+1)}$ will be a mesh point lying in I^n.

With this formula for $\mathbf{x}^{(k+1)}$, we claim that $f(\mathbf{x}^{(k+1)}) \leq f(\mathbf{x}^{(k)}) - 0.5\epsilon^2/M$. To see this, observe that in the case in which $\partial f/\partial x_i(\mathbf{x}^{(k)})$ is positive,

$$\begin{aligned} f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k+1)}) &= \int_{-\epsilon/M}^{0} \frac{\partial f}{\partial x_i}(\mathbf{x}^{(k)} + t\mathbf{e}_i)\,dt \\ &\geq \int_{-\epsilon/M}^{0} (\epsilon + Mt)\,dt \\ &\geq 0.5\epsilon^2/M. \end{aligned}$$

To derive the second line we used the fact that $\partial f/\partial x_i(\mathbf{x}^{(k)}) \geq \epsilon$ as well as the Lipschitz bound on $\partial f/\partial x_i$.

We conclude that the objective function decreases by at least $0.5\epsilon^2/M$ per iteration. As noted earlier, the most that the objective function can decrease is $2Mn$. Therefore, the maximum number of iterations is $4n(M/\epsilon)^2$. Let us state this as a theorem.

THEOREM 2.1. *Let $f : I^n \to \mathbf{R}$ be a C^1 function whose gradient satisfies a Lipschitz condition with bound M. Then an ϵ-approximate local minimum can be found with at most $4n(M/\epsilon)^2$ function and gradient evaluations.*

We remark that if gradient values are not available, the first part of the algorithm (the restrictions to subproblems) can be carried out by estimating the gradient via finite differences. A bound can be derived on the accuracy of finite difference approximations to the gradient using the hypothesis that the gradient is Lipschitz bounded.

If gradient values are not available, then the main local-search step of the algorithm can be replaced with a comparison of the objective value at $\mathbf{x}^{(k)}$ to the objective values at the neighboring grid points. This requires $2n$ function evaluations per local search step.

3. A lower bound for local minimization. In the last section we saw polynomial dependence on both n and M/ϵ. The polynomial dependence on n is to be expected in general (since f depends on n variables, it presumably takes at least n operations merely to evaluate f). The polynomial dependence on M/ϵ is clearly unavoidable with that algorithm since the step size is ϵ/M.

It is natural to inquire whether the polynomial dependence on M/ϵ is actually necessary for all algorithms. Indeed, for the case $n = 1$ there is a simple bisection approach solving the problem in $O(\log(M/\epsilon))$ steps. Could an algorithm with large steps (say, steepest descent combined with line search) achieve better complexity for $n > 1$?

The purpose of this section is to give a lower bound in the $n = 2$ case showing that polynomial dependence on M/ϵ is inherent in the problem of black-box local minimization. The lower bound applies to all algorithms based on the function evaluation model (not merely to the algorithm of the last section). The lower bound is based on a family of functions that could fool any algorithm until it has made at least $\Omega(\sqrt{M/\epsilon})$ function and gradient evaluations. Here the notation $\Omega(\sqrt{M/\epsilon})$ means that the worst-case running time is bounded below by a constant multiple of $\sqrt{M/\epsilon}$ for some sequence of values of M/ϵ tending to infinity. The construction of this family has two parts: an algebraic/geometric part and a combinatorial part. Notice that because we are trying to provide a "bad case" (lower bound) for all possible information-based algorithms, we need a whole family of bad-case functions rather than a single function.

These functions are bad cases in the sense that an algorithm for local minimization will require many steps. There are other senses in which a local optimization example could be bad (for instance, it may be that local minima are easily found but have large objective function values with respect to the global minimum).

We focus on the $n = 2$ case since the interest here is the dependence on M/ϵ. This lower bound is based on the same ideas of a lower bound for Brouwer fixed points in two dimensions due to Hirsch, Papadimitriou, and Vavasis [2]. We assume that M and ϵ are given. In this section we work with $\|\cdot\|_2$ norms because we use two rotated coordinate systems (other norms could lead to confusion).

The lower bound is based on how much information any algorithm could get about f. We argue informally in this section about what the algorithm "knows" from its function evaluations, but the information model can be cast into formal terms. See, for example, Traub, Wasilkowski, and Woźniakowski [6].

We divide the unit square I^2 into $K \times K$ subsquares, where K is an integer on the order of $\sqrt{M/\epsilon}$ (the exact value will be selected below). Besides K, we also have the parameters δ and δ', which are both on the order of ϵ (the exact formulas are below). Number the subsquares with ordered pairs $\langle u, v\rangle$, $u, v = 0, \ldots, K-1$. Two subsquares are said to be *adjacent* if they have a common edge.

We will embed I^2 in the plane diagonally, i.e., with corners at $(0,0)$, $(\sqrt{2}/2, \pm\sqrt{2}/2)$, and $(\sqrt{2}, 0)$. The relationship between the subsquare numbering and coordinate system is as follows. The vertex of subsquare $\langle u, v\rangle$ with minimum x coordinate is at

$$\frac{1}{J}(u+v, v-u),$$

where $J = K\sqrt{2}$. The embedding along with some numbered subsquares is indicated in Fig. 2.

A *southeast track* is a sequence of adjacent subsquares with increasing first coordinates, and a *northeast track* is a sequence of adjacent subsquares with increasing second coordinates.

The *west subsquare* is subsquare $\langle 0, 0\rangle$. Define a *riverbed* to be a sequence of adjacent subsquares starting at the west subsquare, proceeding along a northeast track, and then following a sequence of alternating southeast and northeast tracks, and ending somewhere inside the square. This terminology is used because the mesh plot of function f based on this construction resembles the top view of a riverbed on a hillside. An example of a riverbed is indicated in Fig. 3. Note that the riverbed will have at most $2K - 1$ subsquares. The last (closest to the east) subsquare of the riverbed is called the *sink*. The subsquares of I^2 not in the riverbed are called *hillside* subsquares.

Our functions will be defined based on K, δ, and δ' (i.e., based on M and ϵ) and on a particular choice of riverbed. The function f will be constructed below so that all ϵ-approximate local minima lie in the sink subsquare.

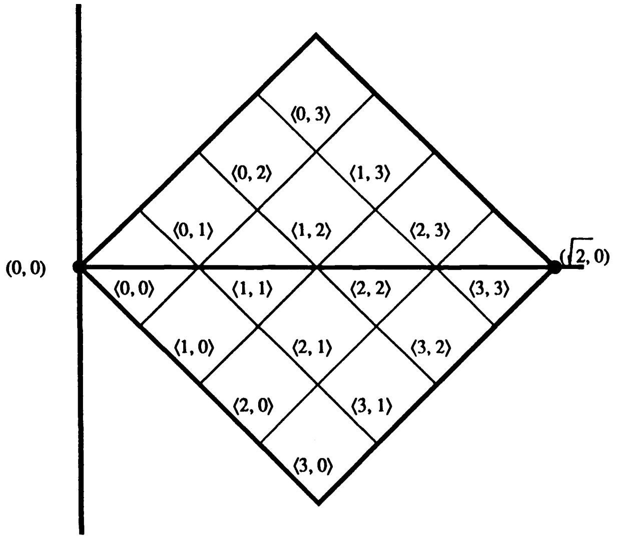


FIG. 2. *Embedding I^2 in the plane with subsquares indicated.*

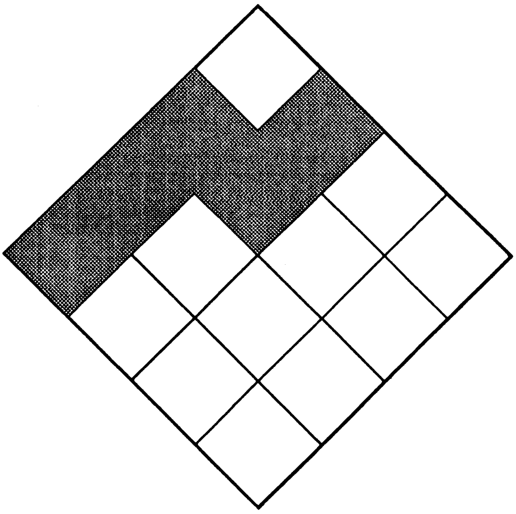


FIG. 3. *An example of a riverbed for $K = 4$.*

Notice that there is a large but finite set of possible riverbeds for each particular value of K. The functions on I^2 in our family will be in correspondence with choices of riverbeds. The particular riverbed to choose will depend on the algorithm at hand—this is the combinatorial part of the construction described below.

For now, we assume that a particular riverbed is selected, and we proceed with the construction of f. All the properties that f should have are stated in the lemmas below. The reader uninterested in the geometric details can skip ahead to Fig. 7 and read the lemmas.

The first part of the construction is the function $s(x)$ that traces the shape of the riverbed. The path defined by $(x, s(x))$ as x varies from 0 to x_e passes through all the subsquares of the riverbed (x_e is defined below). It enters and leaves each subsquare through the midpoint of the edge between adjacent subsquares of the riverbed.

In particular, $s(x)$ is defined piecewise on intervals of the form $[(i+0.5)/J, (i+1.5)/J]$ where i is an integer. If x is the endpoint of one of these intervals, $s'(x) = \pm 1$. The pieces are matched so that $s(x)$ is continuously differentiable. The formulas for $s(x)$ are as follows. In the west subsquare, for x between 0 and $1.5/J$, we define

$$s(x) = \frac{4}{27J}(Jx)^3.$$

We notice that this function leaves the west square through the point $(3/(2J), 1/(2J))$, i.e., the midpoint of the edge between subsquares $\langle 0,0\rangle$ and $\langle 0,1\rangle$. This means that the subsquare after $\langle 0,0\rangle$ in the riverbed will always be $\langle 0,1\rangle$ (as mentioned above, a riverbed is defined to start with a northeast track). Also, we can check that $s'(3/(2J)) = 1$.

In a subsquare $\langle u,v\rangle$ that is interior to a northeast track, $s(x)$ is a linear function with slope 1. Specifically, $(x, s(x))$ linearly joins the point

$$\frac{1}{J}(u+v+0.5, v-u-0.5)$$

to the point

$$\frac{1}{J}(u+v+1.5, v-u+0.5).$$

Similarly, in a subsquare interior to a southeast track, $s(x)$ is linear with slope -1.

In a subsquare $\langle u,v\rangle$ in which the riverbed makes a turn, say, from northeast to southeast, the formula is

$$s(x) = \frac{1}{J}(Jx-u-v-0.5)(u+v+1.5-Jx)+v-u-0.5.$$

This function starts at

$$\frac{1}{J}(u+v+0.5, v-u-0.5)$$

and ends at

$$\frac{1}{J}(u+v+1.5, v-u-0.5).$$

Function $s(x)$ has slope $+1$ at the first point and -1 at the second. A turn from southeast to northeast is analogous.

In the sink subsquare, $s(x)$ is defined by the linear function of slope 1 if the sink subsquare is the terminal of a northeast track, otherwise $s(x)$ is a linear function of slope -1.

The end x_e of the domain of definition of $s(x)$ is the x-coordinate of the midpoint of the edge of the sink square where the riverbed terminates. If $\langle u_e, v_e\rangle$ is the sink subsquare, this coordinate is

$$x_e = \frac{1}{J}(u_e+v_e+1.5).$$

An example of this construction with $K=4$ is plotted in Fig. 4. Here, the riverbed is given by $\langle 0,0\rangle$, $\langle 0,1\rangle$, $\langle 0,2\rangle$, $\langle 1,2\rangle$, $\langle 1,3\rangle$.

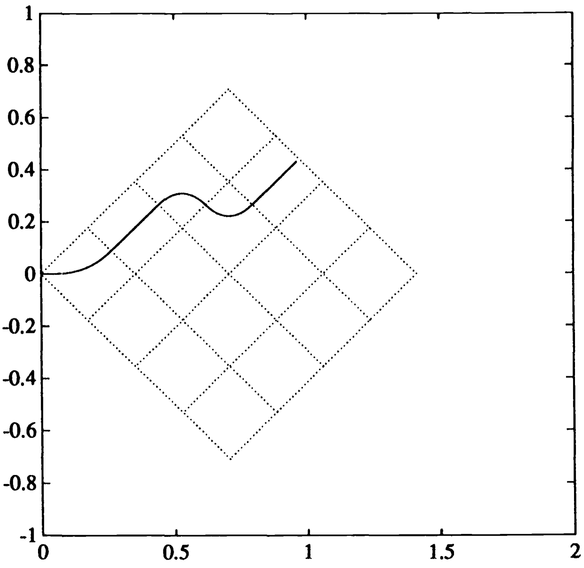


FIG. 4. *An example of* $s(x)$.

We observe that $s(x)$ has the following properties. It is C^1 and piecewise C^2. The maximum value of $|s'(x)|$ is 1, and the maximum value of $|s''(x)|$ (where defined) is $2J$.

We have now defined a function to specify the shape of the riverbed. The next step is to define the two functions controlling the value of function f on the riverbed portion of I^2. The first function indicates how f varies in the direction across the riverbed, and the second indicates how f varies parallel to the riverbed. The first function is defined by

$$c(w) = \begin{cases} 0 & \text{for } w \le -1, \\ -w^4 + 2w^2 - 1 & \text{for } w \in [-1, 1], \\ 0 & \text{for } w \ge 1, \end{cases}$$

which is plotted in Fig. 5.

It is easily checked that this function has the following properties: c is C^1, $c(-1) = c(1) = 0$, $c(0) = -1$, and $c'(-1) = c'(0) = c'(1) = 0$. Also, the maximum value of $|c(w)|$ is 1, of $|c'(w)|$ is approximately 1.54, and of $|c''(w)|$ (which is undefined at ± 1) is 8.

Next we define the function $p(x)$, which determines how f varies as the riverbed is followed. The value of $p(x)$ depends on the position of the sink. Specifically, suppose $\langle u_e, v_e \rangle$ is the sink subsquare. Then the formulas for $p(x)$ are as follows. Let $x_b = (u_e + v_e + 0.5)/J$, that is, the x-coordinate of the point where path $(x, s(x))$ enters the sink square. Let $x_c = (u_e + v_e + 0.75)/J$ and $x_d = (u_e + v_e + 1.0)/J$. Let $b_0 = \delta + \delta x_b$. Then

$$p(x) = \begin{cases} \delta + \delta x & \text{for } x \in [0, x_b], \\ -2\delta J(x - x_b)^2 + \delta(x - x_b) + b_0 & \text{for } x \in [x_b, x_c], \\ (b_0 + \delta/(8J)) \Big[2\,(4J(x - x_c))^3 & \\ \qquad - 3\,(4J(x - x_c))^2 + 1 \Big] & \text{for } x \in [x_c, x_d], \text{ and} \\ 0 & \text{for } x \ge x_d. \end{cases}$$

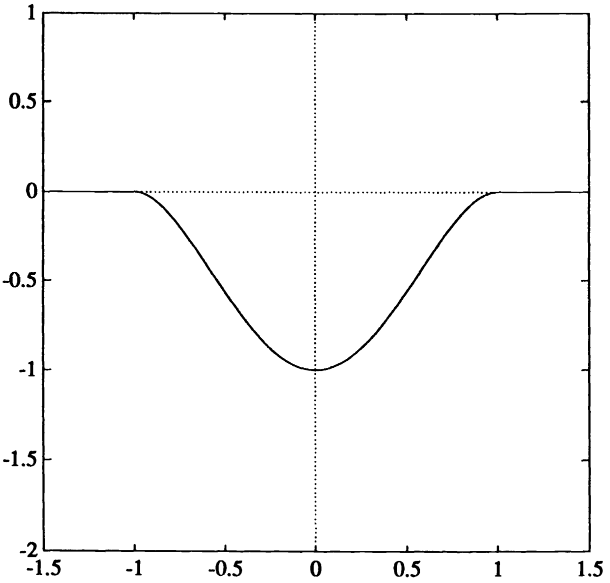


FIG. 5. *The graph of* $c(x)$.

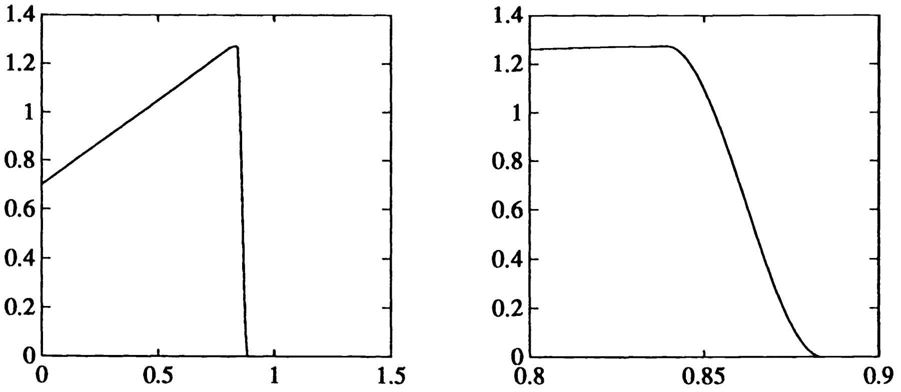


FIG. 6. *An example of* $p(x)$; *the right plot shows a detail.*

It can be checked that $p(x)$ is C^1. In particular, $p(x_b) = b_0$, $p(x_c) = b_0 + \delta/(8J)$, and $p(x_d) = 0$. Also, $p'(x_b) = \delta$ and $p'(x_c) = p'(x_d) = 0$. The maximum value of $|p(x)|$ is $b_0 + \delta/(8J)$, which is at most 3δ. Also, it can be checked that $|p'(x)|$ is at most $18\delta J$, and $|p''(x)|$ (where defined) is at most $288\delta J^2$. An example of $p(x)$ is plotted in Fig. 6.

From $c(w)$, $p(x)$, and $s(x)$ we now assemble the function $f(x,y)$,which is defined as follows:

$$f(x,y) = p(x) \cdot c(2K(y - s(x))) + \delta' x.$$

A MATLAB™ mesh plot of $f(x,y)$ is illustrated in Fig. 7. MATLAB, an interactive package for numerical computation, is a trademark of The Mathworks, Inc. Many of the other figures in this paper were also produced using MATLAB.

We now establish some properties of this function.

LEMMA 3.1. *If* (x,y) *lies in the hillside, then* $f(x,y) = \delta' x$.

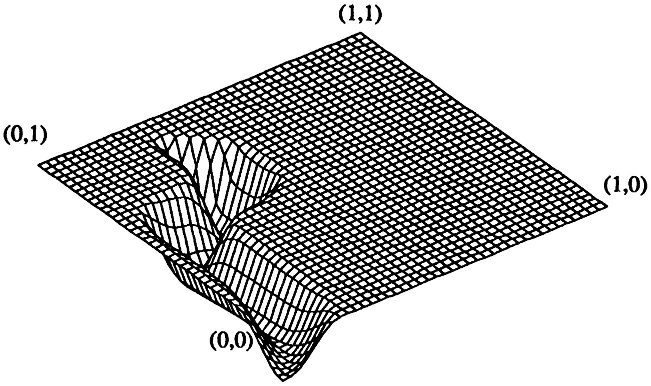


FIG. 7. *An example of $f(x,y)$.*

Proof. We must show that the first term vanishes outside the riverbed. If (x,y) is not in the riverbed, either $x > x_d$ or $|y - s(x)| \geq 3/(4J)$. This latter inequality arises from the fact that the distance from $(x, s(x))$ in the y-direction to the boundary of the riverbed is always at least $3/(4J)$ by definition of $s(x)$. If $x > x_d$ then $p(x) = 0$, so the claim is true. Similarly, if $|y - s(x)| \geq 3/(4J)$ then $2K|y - s(x)| \geq (3K)/(2J) \geq 1$, hence $c(2K(y - s(x))) = 0$. □

At this point, we choose an ϵ' to be slightly larger than ϵ, and we let

$$\delta = 32\sqrt{2}\epsilon'$$

and

$$\delta' = 19\sqrt{2}\epsilon'.$$

These choices are made so that we can prove the following lemma.

LEMMA 3.2. *Let (x,y) be a point not in the sink square. Then $\|\nabla f(x,y)\|_2 \geq \sqrt{2}\epsilon'$.*

Proof. Let w denote $2K(y - s(x))$. We compute

$$\nabla f(x,y) = (p'(x)c(w) - 2Kp(x)c'(w)s'(x) + \delta', 2Kp(x)c'(w)).$$

We now take cases to prove a lower bound on the size of $\nabla f(x,y)$. The first case is that we are not in the riverbed, which was handled by the previous lemma and by the fact that $\delta' \geq \sqrt{2}\epsilon'$. This is the case in which $|w| \geq 1$ or $x \geq x_d$. For the other cases we assume that $|w| \leq 1$ and $x \leq x_b$. (We can assume that $x \leq x_b$ since (x,y) is not in the sink.) We now take subcases. The first subcase is that $|w| \in [1 - 1/(64K), 1]$. In this case, $|c(w)|$ and $|c'(w)|$ are at most $1/(8K)$. Then, we observe that for x not in the sink, $|p(x)| \leq 2\delta$, $|p'(x)| \leq \delta$, and $|s'(x)| \leq 1$. Thus the term $|p'(x)c(w)|$ above is at most $\delta/16$, and the term $|2Kp(x)c'(w)s'(x)|$ is at most $\delta/2$. The third term of the first entry of ∇f is exactly δ'; therefore, the first entry of the derivative has magnitude at least $\delta' - (9/16)\delta$. Using the above formulas for δ, δ', this is a magnitude of at least $\sqrt{2}\epsilon'$.

In the second subcase, $|w| \in [1/(64K), 1 - 1/(64K)]$. In this case, we observe that $|c'(w)| \geq 1/(10K)$. This means that the second entry $|2Kp(x)c'(w)|$ of $\nabla f(x,y)$ is at least $|p(x)/5|$, i.e., at least $\delta/5$. This quantity is greater than $\sqrt{2}\epsilon'$.

In the third subcase, $|w| \in [0, 1/(64K)]$. In this case, $|c(w)| \geq 7/8$, whereas $|c'(w)| \leq 1/(16K)$. Therefore, the first term $|p'(x)c(w)|$ is at least $(7/8)\delta$. The second term $|2Kp(x)c'(w)s'(x)|$ is at most $(1/4)\delta$. The last term is exactly δ'. Thus the first entry has magnitude at least $(7/8)\delta - (1/4)\delta - \delta'$, which is $\sqrt{2}\epsilon'$. □

LEMMA 3.3. *Let (x, y) be a point not in the sink square. Then (x, y) is not an ϵ-approximate local minimum of f.*

Proof. This follows from the previous lemma, with special attention paid to the boundaries. Region I^2 has four boundaries and four corners. We must check whether, if (x, y) is on a boundary, the projection of $\nabla f(x, y)$ onto the boundary will be at least ϵ'. The argument is as follows. Along a boundary we know that $\nabla f(x, y) = (\delta', 0)$ from the construction, except in the west subsquare. The gradient $(\delta', 0)$ has magnitude at least ϵ when projected on every boundary except the point $(0, 0)$. This takes care of the whole boundary outside the west square.

Therefore, we only have to examine the exterior boundary of the west subsquare. A calculation shows that every point has a projected gradient of size at least ϵ'. □

Note that we have not established the existence of an ϵ-approximate local minimum in the sink square. We know by compactness, however, that such a point exists, and the previous lemma forbids its existence anywhere else.

It is now time to select the value of K, which will be

$$K = \left\lfloor \frac{1}{310}\sqrt{\frac{M}{\epsilon'}} \right\rfloor .$$

The reason for this value of K is to establish the following lemma.

LEMMA 3.4. *The gradient $\nabla f(x, y)$ for f defined above is continuous and has Lipschitz constant at most M.*

Proof. The gradient exists everywhere and is continuous because f is assembled from C^1 functions of one variable. Because f is continuously differentiable and piecewise C^2, then the following inequality holds:

$$\|\nabla f(x_1, y_1) - \nabla f(x_2, y_2)\|_2 \leq \int_P \|D^2 f(x, y)\|_2 \cdot \|(x_2, y_2) - (x_1, y_1)\|_2 \, dP,$$

where P is a straight-line path from (x_1, y_1) to (x_2, y_2), and D^2 denotes the second derivative. This inequality holds for almost all pairs of points (the only exception being the case when P intersects a continuum of points where $D^2 f$ fails to exist). Thus, to get a Lipschitz bound on $\nabla f(x, y)$ it suffices to establish an upper bound on the two-norm of the second derivative wherever it is defined. Since the two-norm of a matrix is hard to work with, we instead put an upper bound on the infinity norm, and then multiply it by $\sqrt{2}$.

We compute the second derivative entry by entry:

$$\begin{aligned}
\frac{\partial^2 f(x, y)}{\partial x^2} &= p''(x)c(w) - 4Kp'(x)c'(w)s'(x) \\
&\quad + 4K^2 p(x)c''(w)s'(x)^2 - 2Kp(x)c'(w)s''(x), \\
\frac{\partial^2 f(x, y)}{\partial x \partial y} &= 2Kp'(x)c'(w) - 4K^2 p(x)c''(w)s'(x), \\
\frac{\partial^2 f(x, y)}{\partial y^2} &= 4K^2 p(x)c''(w).
\end{aligned}$$

We can go through each term and use the crude estimates made earlier to get an upper bounds of $1116K^2\delta$ on $\partial^2 f/\partial x^2$, $312K^2\delta$ on $\partial^2 f/\partial x \partial y$, and $96K^2\delta$ on $\partial^2 f/\partial y^2$.

Estimating $\delta \leq 45.3\epsilon'$ gives an upper bound of about $6.5 \cdot 10^4 K^2 \epsilon'$ for the ∞-norm of the second derivative, which translates to an upper bound of about $9.2 \cdot 10^4 K^2 \epsilon'$ for the two-norm. Therefore, with the above choice of K we are guaranteed to have a Lipschitz bound of at most M. Note that the true Lipschitz bound for our construction grows proportionally to $K^2\epsilon$ but with a much smaller constant. □

LEMMA 3.5. *Suppose that (x,y) lies in subsquare $\langle u,v\rangle$. From $f(x,y)$ and $\nabla f(x,y)$ it is not possible to determine any information about the riverbed except possibly whether or not $\langle u,v\rangle$ lies in the riverbed and, if so, what the positions of the two neighboring riverbed squares are.*

Proof. This follows from the definition of $f(x,y)$. If $x \geq x_d$ or $|y-s(x)| \geq 1/(2K)$ then we cannot determine anything about the riverbed except that $\langle u,v\rangle$ is not in the riverbed. If $|y - s(x)| \leq 1/(2K)$ and $x \leq x_d$ then we might be able to determine the values of $s(x)$, $s'(x)$, and $p(x)$ from $f(x,y)$ and $\nabla f(x,y)$. This means we can determine that the particular square is in the riverbed, and we can determine what kind of turn the riverbed makes. Nothing else can be determined. □

We now prove a general lower bound for finding approximate local minima for this family of functions. We imagine an algorithm A that makes function and gradient evaluations. We want to find a pair of functions $f(x,y)$, $f'(x,y)$ in our family with disjoint sets of approximate local minimum such that algorithm A cannot distinguish them until K function/gradient evaluations (i.e., $\Omega(\sqrt{M/\epsilon})$) have been made. Notice that the only approximate local minima for functions in our family occur in the sink square, and therefore f and f' will have different sinks. We start by assuming that the algorithm knows M and ϵ (and therefore K).

The combinatorial argument that constitutes the remainder of this section is identical to the argument of [2], but we present it again here for the sake of completeness.

To construct f and f' we need to specify riverbeds R, R'. The riverbeds for these two functions will be almost identical. The riverbeds are constructed "adaptively." In particular, we fix more and more of R, R' as we observe the test points made by A. The idea is that f and f' will agree at all test points, so R and R' will agree almost until the end.

We assume that A makes a deterministic sequence of test points, and that at each test point it evaluates f and ∇f. The sequence of test points is denoted (x_i, y_i). Each one may depend on previous test points in any way possible. Thus A has unlimited computational power. Note that there is no advantage for A to make a test point exactly on a boundary of a subsquare for our family (i.e., no more information about the riverbed can be gleaned from a boundary than from a nearby interior point), so we assume that all test points lie in a unique subsquare. Once a test point has been made in a subsquare, we assume that A has complete information about the subsquare (i.e., all the values of $f(x,y)$ are known to A for (x,y) lying in the subsquare).

The riverbeds R, R' are constructed as a sequence of tracks alternating northeast and southeast. They are built up as the limit of the sequence $R_0, R_1, R_2, \ldots$, where each $R_i \subset R_{i+1}$, and R_i denotes the partial riverbed that is determined after i test points from A (note that $R_0 = \{\langle 0,0\rangle\}$ since this subsquare is in every riverbed).

We know that R_i starts from $\langle 0,0\rangle$; denote its last square as $\langle u_i, v_i\rangle$. In our construction R and R' will agree with R_i all the way up to subsquare $\langle u_i, v_i\rangle$, and moreover, that R, R' will make a bend in subsquare $\langle u_i, v_i\rangle$. We let T_i denote the track starting from $\langle u_i, v_i\rangle$ following the direction of the bend and proceeding to the border of I^2. The invariant property of the upcoming construction is that no test points have been made in T_i on iterations 1 up to i. Note that T_0 is the northeast

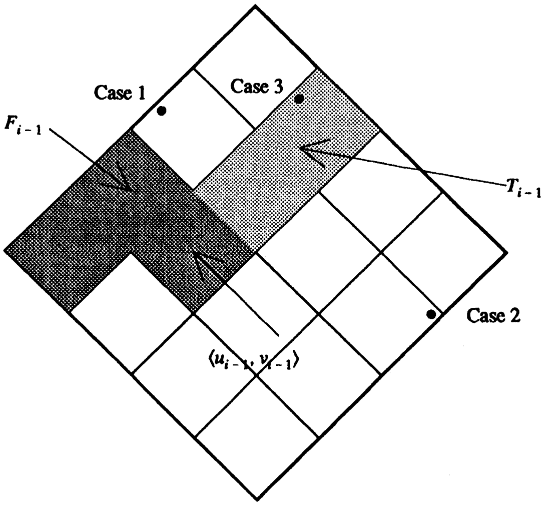


FIG. 8. *The three cases for test points.*

track of subsquares with first coordinates equal to 0.

We now give the rules for extending R_{i-1} to R_i. There are three cases for test point i. In the first case, (x_i, y_i) lies in the part of I^2 that is already determined. To be specific, suppose, for example, that R_{i-1} ends at subsquare $\langle u_{i-1}, v_{i-1}\rangle$ and that T_{i-1} is a northeast track emerging from this subsquare. Suppose that (x_i, y_i) lies in subsquare $\langle u', v'\rangle$. If $u' < u_{i-1}$ or $v' \le v_{i-1}$ then the behavior of R is entirely known in $\langle u', v'\rangle$, and hence f, f' are determined already. In this case, we set $R_i = R_{i-1}$, $T_i = T_{i-1}$, and $\langle u_i, v_i\rangle = \langle u_{i-1}, v_{i-1}\rangle$.

The second case is that (x_i, y_i) lies in a subsquare of I^2, not in T_{i-1}, but through which R or R' might eventually pass. This is the case in which $u' > u_{i-1}$ and $v' > v_{i-1}$. In this case, we mark all subsquares with first coordinate equal to u' as "forbidden" and the same with subsquares with second coordinate equal to v'. In this case we again set $R_i = R_{i-1}$, $T_i = T_{i-1}$, and $\langle u_i, v_i\rangle = \langle u_{i-1}, v_{i-1}\rangle$. We also set $f(x, y) = \delta' x$ in this subsquare (i.e., we "tell" the algorithm that the riverbed does not pass through this subsquare).

In the third case, (x_i, y_i) lies in T_{i-1}. In this case (assuming as above that T_{i-1} is a northeast track), we let v_i be the smallest integer coordinate greater than v_{i-1} that has not yet been forbidden in the construction procedure described above. Then we let R_i be the union of R_{i-1} and the portion of T_{i-1} connecting $\langle u_{i-1}, v_{i-1}\rangle$ to $\langle u_{i-1}, v_i\rangle$. We let $u_i = u_{i-1}$ and T_i be the southeast track starting at $\langle u_i, v_i\rangle$ and including subsquares with increasing first coordinates. Finally, we assign values to $f(x, y)$ based on R_i in the subsquare that contained the test point. If T_{i-1} had been a southeast track, then T_i would have been a northeast track.

Figure 8 shows the three possible locations for a test point. Notice that the rule for forbidding northeast and southeast tracks keeps the whole procedure consistent, i.e., each R_i is a valid riverbed that is consistent with all the test points up to (x_i, y_i).

How long can this construction proceed? We notice that if R_i terminates at subsquare $\langle u_i, v_i\rangle$, then $u_i \le i$ and $v_i \le i$ because we never pass to a higher value of u unless all lower integer values of u had test points associated with them. The same holds for the second coordinate.

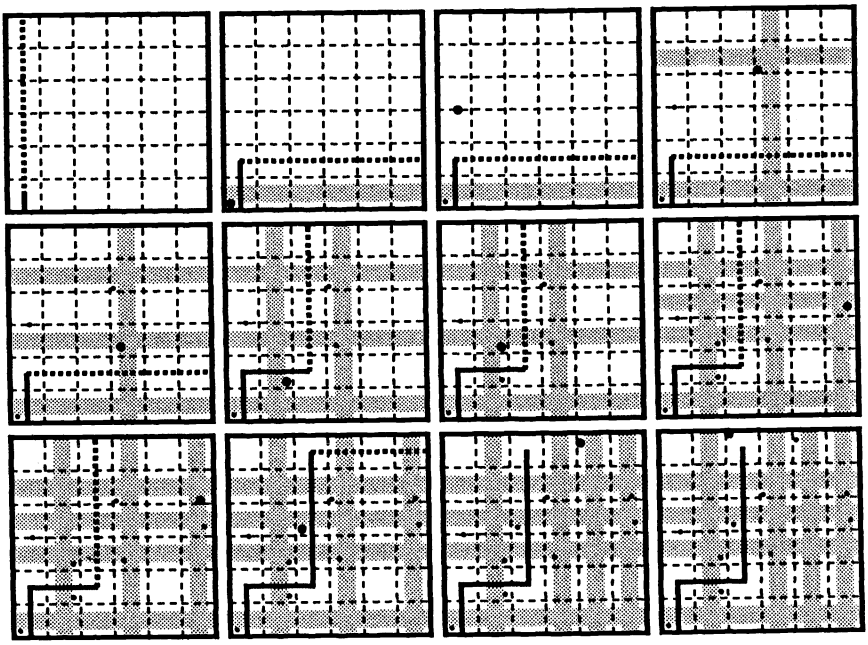


FIG. 9. *The construction of the riverbed for* 12 *test points.*

Therefore, we can continue extending R_i until K test points have been made. Until the $K-1$st test point, there are at least two subsquares in T_i, and therefore, there are at least two subsquares in which the riverbed could end. Therefore, we let R be R_{K-1} terminated with one of these sinks, and R' be R_{K-1} terminated with the other. Then the algorithm cannot distinguish f from f' until $K-1$ test points have been made.

We state this as a theorem.

THEOREM 3.6. *Let A be any deterministic algorithm to find ϵ-approximate local minima of functions $f : I^2 \to \mathsf{R}$ whose gradients satisfy Lipschitz conditions with constant M. Assume the algorithm is limited to using function and gradient evaluations. Then, in the worst case, algorithm A requires $\Omega(\sqrt{M/\epsilon})$ function and gradient evaluations.*

As a further example of the construction, we give a series of 12 test points in a 6×6 grid, illustrated in Fig. 9. The last test point is the sink square. The forbidden rows and columns are shaded. The most recent test point in each figure (i.e., the test point not in the preceding picture) is shown enlarged. The dashed line indicates T_i in the preceding construction. A test point in T_i causes the riverbed to be extended. Notice that the riverbed never reaches a row or column until all previous rows and columns have had test points. In the twelfth plot, the riverbed can no longer be extended, so the sink square is finally fixed in subsquare $\langle 2, 5\rangle$.

4. Tests with an actual optimization algorithm. We implemented an optimization algorithm that is based on algorithms common in the literature. In particular, our algorithm uses a second-order model of the objective function. The quadratic term in the model is either the exact Hessian $f''(x)$ in the case when the Hessian is positive definite, or a matrix of the form $f''(x) + \lambda I$ for some choice of $\lambda > 0$. We remark that, outside the riverbed, our function is linear, so that $f''(x) = 0$. This means

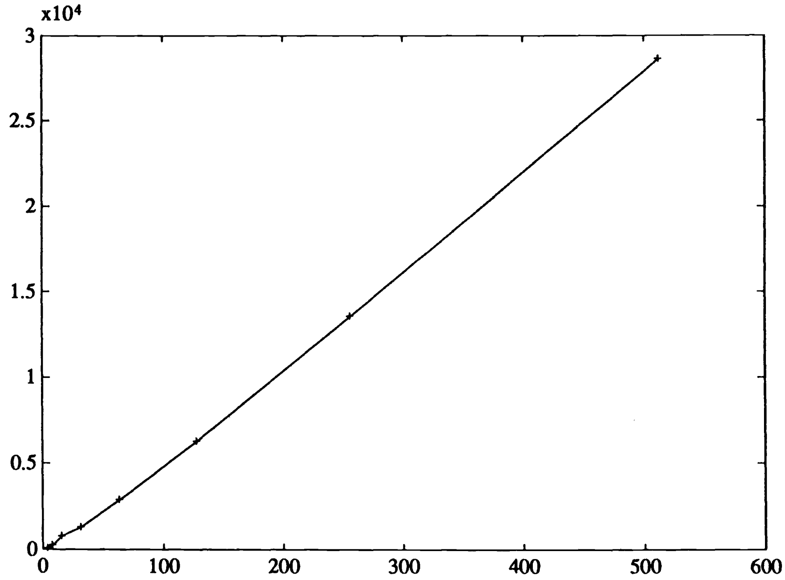


FIG. 10. *Growth of the number of function evaluations* (y *axis*) *as a function of* K.

that the second-order step (after λI is added) becomes simply a scaled gradient step (steepest descent).

For our class of functions, the second derivative is not even defined at all points. This means that, in principle, there is no reason to believe that second-order information would speed up global convergence. Nonetheless, we found that second-order information sped up convergence by a factor of about 20.

We use an Armijo-type line search once a search direction is identified. Finally, we take special action to project the search direction when the test point happens to be on the boundary. See [1] for a description of Armijo line searches and for minimization with second-order models. Since our interest is on the lower bound and not on the particular optimization algorithm used, we omit the details of our algorithm.

The function f is the same function described in the previous section, and we use the adaptive riverbed construction technique used to prove Theorem 3.6. The whole procedure was implemented in MATLAB. The number of function evaluations for $K = 4, 8, 16, 32, 64, 128, 256, 512$ is plotted in Fig. 10. Theorem 3.6 mandates that the number of function/gradient evaluations be at least K. The table suggests that for this algorithm, the number of evaluations is linear in K, about $55K$. We did not tabulate gradient and Hessian evaluations.

5. An improved algorithm when n is small. We notice that the upper bound on Algorithm LOCAL1 in §2 grows like $(M/\epsilon)^2$, whereas the lower bound grows only like $\sqrt{M/\epsilon}$. Is it possible to bring these bounds in closer agreement? In this section we propose an improvement on the algorithm of §2 in the case when n is very small with respect to M/ϵ. The new algorithm will be called LOCAL2.

The main point of LOCAL2 is to pick the initial point $\mathbf{x}^{(0)}$ for LOCAL1 in an intelligent manner. We make a mesh with points spaced $1/k$ in every dimension (the "coarse" grid), where k is an integer determined below. Then we evaluate f at every one of these $(k+1)^n$ mesh points. We let $\mathbf{x}^{(0)}$ be the coarse grid point with the minimum value of f. We begin the local improvement algorithm (on the "fine mesh," that is, the mesh with spacing ϵ/M) from this $\mathbf{x}^{(0)}$. Assume that k is an integer divisor of M/ϵ, so that all the coarse grid points are also fine grid points.

Now we re-analyze the number of steps to find a local minimum. Let $\mathbf{x}^{(0)}$ be as in the previous paragraph. Let $\mathbf{x}^*$ be a global minimum of f. In §2 we established an upper bound on $f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*)$ without any special knowledge about $\mathbf{x}^{(0)}$. In this section we want a better bound on this difference.

To establish this bound, let $\mathbf{x}'$ be the coarse grid point closest to $\mathbf{x}^*$. First, we establish the claim that $\|\mathbf{g}(\mathbf{x}')\|_\infty \leq M/(2k)$. Suppose not; suppose, e.g., that $g_i(\mathbf{x}') > M/(2k)$. This means that $\partial f/\partial x_i(\mathbf{x}') > M/(2k)$ and $x'_i > 0$. Since $\mathbf{x}'$ is the closest coarse grid point to $\mathbf{x}^*$, the difference between x'_i and x^*_i is at most $1/(2k)$. In particular, $x^*_i > 0$ since $x'_i \geq 1/k$. We have the bound $\|\mathbf{x}' - \mathbf{x}^*\| \leq 1/(2k)$, so the Lipschitz bound implies that $\partial f/\partial x_i(\mathbf{x}^*) > 0$. This, combined with the fact that $x^*_i > 0$, contradicts the minimality of $\mathbf{x}^*$.

Thus, $\|\mathbf{g}(\mathbf{x}')\|_\infty \leq M/(2k)$. Now we put an upper bound on the difference $f(\mathbf{x}') - f(\mathbf{x}^*)$. We use the same reasoning as in §2, namely, we form a path with n segments between the two points and express $f(\mathbf{x}') - f(\mathbf{x}^*)$ as the integral of partial derivatives along the path. Each path segment has length at most $1/(2k)$, and each integrand is bounded above by $2M/(2k)$. Therefore, the total difference is at most $nM/(2k^2)$. This gives an upper bound on $f(\mathbf{x}') - f(\mathbf{x}^*)$. Since $f(\mathbf{x}^{(0)}) \leq f(\mathbf{x}')$ (because $\mathbf{x}^{(0)}$ is the coarse grid point with the smallest value of the objective function), we conclude that

$$f(\mathbf{x}^{(0)}) - f(\mathbf{x}^*) \leq \frac{nM}{2k^2}.$$

Starting from $\mathbf{x}^{(0)}$, we apply the same local search algorithm, LOCAL1, as was used in §2. We now get a new bound on the number of steps. Since each step decreases the objective function by at least $0.5\epsilon^2/M$, and since the maximum possible decrease is given above, we get a bound of $nM^2/(\epsilon^2 k^2)$ on the number of search iterations.

Thus the algorithm requires a total of

$$(k+1)^n + \frac{nM^2}{\epsilon^2 k^2}$$

function and gradient evaluations. We want to choose k to be an integer that minimizes this total. A good choice is to choose k between

$$\left(\frac{nM^2}{\epsilon^2}\right)^{\frac{1}{n+2}} - 2 \leq k \leq \left(\frac{nM^2}{\epsilon^2}\right)^{\frac{1}{n+2}} - 1.$$

With these choices, we can estimate

$$(k+1)^n \leq \left(\frac{nM^2}{\epsilon^2}\right)^{\frac{n}{n+2}}.$$

To analyze the other term, we assume that $M^2/\epsilon^2 \geq 4^{n+2}/n$ (recall that the method of this section is meant to be applied when n is small with respect to M/ϵ). If this holds, then $(nM^2/\epsilon^2)^{1/(n+2)} \geq 4$. Thus,

$$\frac{nM^2}{\epsilon^2 k^2} \leq \frac{nM^2}{\epsilon^2}\left[\left(\frac{nM^2}{\epsilon^2}\right)^{\frac{1}{n+2}} - 2\right]^{-2}$$

$$\leq \frac{nM^2}{\epsilon^2}\left[\frac{1}{2}\left(\frac{nM^2}{\epsilon^2}\right)^{\frac{1}{n+2}}\right]^{-2}$$

$$\leq 4\left(\frac{nM^2}{\epsilon^2}\right)^{\frac{n}{n+2}}.$$

Thus we see that the total time for LOCAL2 is at most

$$O\left(\left(\frac{nM^2}{\epsilon^2}\right)^{\frac{n}{n+2}}\right).$$

In the special case of $n = 2$ (the case covered in the previous section), this gives a bound of $O(M/\epsilon)$, which is closer but still not equal to the lower bound.

The above choice of k will generally not be an integer divisor of M/ϵ. This can be addressed with further analysis, which we omit.

6. Local minima compared to global minima and fixed points. In §2 we came up with a bound of $O(nM^2/\epsilon^2)$ for finding ϵ-approximate local minima. The purpose of this section is to compare this result to information bounds for global minima and Brouwer fixed points. As we will see, these two other problems both depend on n exponentially.

Define an *ϵ-approximate global minimum* of a function $f : I^n \rightarrow \mathbf{R}$ to be a point $\mathbf{x}$ such that, if $\mathbf{x}^*$ is a global minimum, then $f(\mathbf{x}) - f(\mathbf{x}^*) \leq \epsilon$. It turns out that the reasonable assumption to make for this problem is that f has Lipschitz bound L (rather than assuming a Lipschitz bound on ∇f). It is fairly straightforward to prove upper and lower bounds on this problem of the form $(cL/\epsilon)^n$, where c is a constant. This result is implicit in work by Sikorski [5] and appears in other places in the literature. Thus we see an exponential instead of polynomial dependence on n.

Local minima are more closely connected to Brouwer fixed points than to global minima. In fact, as we will show, local minima may be regarded as a special case of Brouwer fixed points. Let $\mathbf{u} : I^n \rightarrow I^n$ be a continuous function. Then Brouwer's fixed point theorem states that there exists an $\mathbf{x} \in I^n$ such that $\mathbf{u}(\mathbf{x}) = \mathbf{x}$. Such an $\mathbf{x}$ is called a *fixed point.*

Define an *ϵ-approximate fixed point* to be a vector $\mathbf{x} \in I^n$ such that $\|\mathbf{u}(\mathbf{x}) - \mathbf{x}\|_\infty \leq \epsilon$. It turns out that the reasonable assumption to make is that function $\mathbf{u}(\mathbf{x}) - \mathbf{x}$ has Lipschitz bound K. In this case, [2] showed that the worst case for Brouwer fixed points in the information model behaves roughly like $(cK/\epsilon)^n$ (again, exponential in n).

We claim that local minima can in fact be phrased as Brouwer fixed point problems.

In particular, given a continuously differentiable function $f : I^n \rightarrow \mathbf{R}$ with a Lipschitz bound of M on the gradient, we define a vector-valued function $\mathbf{u}(\mathbf{x})$ as follows. For the purpose of this discussion, it is convenient to assume that $I^n = [-1/2, 1/2]^n$ so that the origin is the center of the domain. For $c > 0$ let $p_c(x)$ be the function that projects onto the interval $[-c, c]$ (i.e., $p_c(x) = \text{median}\{-c, x, c\}$), and let $\mathbf{p}_c$ be the coordinate-wise projection onto $[-c, c]^n$, i.e., $\mathbf{p}_c(\mathbf{x}) = (p_c(x_1), \ldots, p_c(x_n))$.

Let ϵ' be slightly larger than ϵ. We define a new domain U to be $[-1/2 - \epsilon', 1/2 + \epsilon']^n$. Notice that $\mathbf{p}_{1/2}$ maps U onto I^n. Then we define $\mathbf{u}(\mathbf{x})$ on U as follows:

$$\mathbf{u}(\mathbf{x}) = \mathbf{p}_{1/2}(\mathbf{x}) - \mathbf{p}_{\epsilon'}(\nabla f(\mathbf{p}_{1/2}(\mathbf{x}))).$$

The image of $\mathbf{u}$ lies in U (because the first term has ∞-norm at most $1/2$, and the second term at most ϵ'), hence $\mathbf{u}(\mathbf{x})$ satisfies the conditions of Brouwer's theorem.

The first claim is that $\mathbf{u}(\mathbf{x})-\mathbf{x}$ has Lipschitz constant equal to M. If $\mathbf{x} \in I^n$ then $\mathbf{u}(\mathbf{x}) = \mathbf{x} - \mathbf{p}_{\epsilon'}(\nabla f(\mathbf{x}))$; hence $\mathbf{u}(\mathbf{x}) - \mathbf{x} = -\mathbf{p}_{\epsilon'}(\nabla f(\mathbf{x}))$. This right-hand side has a Lipschitz constant of M. The other case is handled in a similar manner.

Now, suppose $\mathbf{x} \in U$ is an ϵ-approximate fixed point of $\mathbf{u}$. Let $\mathbf{y} = \mathbf{p}_{1/2}(\mathbf{x})$; we claim that $\mathbf{y}$ is an ϵ'-approximate local minimum of f. Let $\mathbf{d} = \mathbf{y} - \mathbf{x}$. For each i, if $d_i > 0$ then $y_i = -1/2$, and if $d_i < 0$ then $y_i = 1/2$. With this notation,

$$\mathbf{u}(\mathbf{x}) - \mathbf{x} = \mathbf{d} - \mathbf{p}_{\epsilon'}(\nabla f(\mathbf{y})). \tag{2}$$

The left-hand side of (2) is assumed to have ∞-norm at most ϵ. Consider an index i such that $y_i > -1/2$. In this case, $d_i \leq 0$, so (2) implies that the ith entry of $\mathbf{p}_{\epsilon'}(\nabla f(\mathbf{y}))$ is at most ϵ. This means that $\partial f/\partial x_i(\mathbf{y}) \leq \epsilon$. Analogous reasoning applies to the case when $y_i < 1/2$. This shows that $\mathbf{y}$ satisfies the conditions for being an ϵ-KKT point, and hence an ϵ'-approximate local minimum.

Conversely, suppose that $\mathbf{y}$ is an ϵ-approximate local minimum of f. For each i such that $y_i = 1/2$, define $d_i = \min(0, p_{\epsilon'}(\partial f/\partial x_i(\mathbf{y})))$. For each i such that $y_i = -1/2$, define $d_i = \max(0, p_{\epsilon'}(\partial f/\partial x_i(\mathbf{y})))$. For other i, let $d_i = 0$. Then it can be checked that the point $\mathbf{y} - \mathbf{d}$ will be an ϵ-approximate fixed point of $\mathbf{u}$.

Notice that the size of U is slightly larger than the size of I^n. The size of U can be brought to 1 in every dimension by scaling. This would have an effect on the value of M.

The construction of $\mathbf{u}$ from f introduced in the last few paragraphs was rather intricate. We remark that simpler definitions for $\mathbf{u}$ that might seem plausible do not give true Brouwer functions. For example, if we simply defined $\mathbf{u}(\mathbf{x}) = \mathbf{x} - \nabla f(\mathbf{x})$, then the image of $\mathbf{u}$ would not necessarily be contained in I^n. Similarly, if we defined $\mathbf{u}(\mathbf{x})$ to be $\mathbf{x} - \mathbf{g}(\mathbf{x})$, where $\mathbf{g}$ is the "projected gradient" of §2, we would find that $\mathbf{u}$ is discontinuous. Either way, $\mathbf{u}$ would not be covered by Brouwer's theorem.

The earlier construction shows that approximate local minimization can be expressed as a special case of Brouwer fixed points. Finally, we remark that approximate local minimization is related to complexity classes PLS and PPAD designed for combinatorial problems, the first having to do with local minima and the second with Brouwer fixed points. See Papadimitriou [4] for more information.

7. Conclusion. We have presented a simple local search algorithm whose running time is polynomial in the dimension of the problem. We have also presented a family of problems for which finding a local minimum would be time-consuming for any information-based algorithm.

There are many questions left unanswered by this work. What happens when more complicated domains than I^n are used? How can the gap between the lower bound of §3 and the upper bound of §5 be closed?

We have assumed that the functions under consideration are C^1. What if we assumed that they are C^2 with a Lipschitz bound on the second derivative? This would open up the possibility of using Newton-type methods. Would these Newton-type methods be provably more efficient than gradient-based methods?

Our algorithms LOCAL1 and LOCAL2 were designed mainly with ease of analysis in mind. Can more practical algorithms be placed in the context of this paper and analyzed? In particular, would the algorithm of §4 (or some similar algorithm) always converge within a number of steps comparable to the time bounds of LOCAL1 and LOCAL2?

Finally, is there a good explanation for the fact that approximate local minima can be found in time polynomial in n but not in Brouwer fixed points?

Acknowledgment. The author thanks Michael Todd of Cornell for pointing out a key improvement to an earlier version of algorithms LOCAL1 and LOCAL2.

REFERENCES

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

[2] M. D. Hirsch, C. H. Papadimitriou, and S. A. Vavasis, *Exponential lower bounds for finding Brouwer fixed points,* J. Complexity, 5 (1989), pp. 379–416.

[3] K. G. Murty and S. N. Kabadi, *Some NP-complete problems in quadratic and nonlinear programming,* Math. Programming, 39 (1987), pp. 117–129.

[4] C. H. Papadimitriou, *On graph-theoretic lemmata and complexity classes,* preprint, 1990.

[5] K. Sikorski, *Optimal solution of nonlinear equations satisfying a Lipschitz condition,* Numer. Math., 43 (1984), pp. 225–240.

[6] J. F. Traub, G. W. Wasilkowski, and H. Woźniakowski, *Information-Based Complexity,* Academic Press, Boston, 1988.

SIAM J. OPTIMIZATION
Vol. 3, No. 1, pp. 81–117, February 1993

MESH INDEPENDENCE FOR NONLINEAR LEAST SQUARES PROBLEMS WITH NORM CONSTRAINTS*

MATTHIAS HEINKENSCHLOSS†

Abstract. If one solves an infinite-dimensional optimization problem by introducing discretizations and applying a solution method to the resulting finite-dimensional problem, one often observes the very stable behavior of this method with respect to varying discretizations. The most striking observation is the constancy of the number of iterations needed to satisfy a given stopping criterion. In this paper an analysis of these phenomena is given and the so-called mesh independence for nonlinear least squares problems with norm constraints (NCNLLS) is proved. A Gauss–Newton method for the solution of NCNLLS is discussed and its convergence properties are analyzed. The mesh independence is proven in its sharpest formulation. Sufficient conditions for the mesh independence to hold are related to conditions guaranteeing convergence of the Gauss–Newton method. The results are demonstrated on a two-point boundary value problem.

Key words. nonlinear least squares, Gauss–Newton method, mesh independence, parameter identification

AMS(MOS) subject classifications. 65K10, 35R30, 65L50, 65M50

1. Introduction.

This paper is concerned with the analysis of Gauss–Newton methods applied to (Galerkin) discretizations of infinite-dimensional nonlinear least squares problems of the following type:

$$\begin{array}{ll} \min & ||F(x)||_Y^2 \\ \text{s.t.} & ||x||_X \leq R, \end{array} \tag{1.1}$$

where F is a sufficiently smooth, weakly continuous function, which acts between the two Hilbert spaces X and Y. Problems of this kind frequently arise in parameter identification (see, e.g., [5], [20], [26], and [28]). The constraint $||x||_X \leq R$ reflects a priori information on the sought parameter and guarantees the solvability of (1.1).

If residual and nonlinearity of F are of moderate size, a Gauss–Newton-like method is an appropriate technique for solving (1.1). For the solution of the constrained problem (1.1) we propose a Gauss–Newton method in which the function F is linearized around a given approximation x_k of the solution, whereas the constraint is retained. The approximation is improved by solving the resulting constrained linear least squares problem. This yields the following algorithm (here and in the subsequent sections $B_r(x)$ will be the open ball around x with radius r).

ALGORITHM 1.1.

(0) Given an initial point $x_0 \in \overline{B_R(0)}$, set $k = 0$.

(1) Compute the solution x_{k+1} of the linearized problem (let μ_{k+1} denote the corresponding Lagrange multiplier)

$$\begin{array}{ll} \min & ||F(x_k) + F'(x_k)(x - x_k)||_Y^2 \\ \text{s.t.} & ||x||_X \leq R. \end{array} \tag{1.2}$$

(2) Test for convergence. If the test succeeds, take x_{k+1} as an approximation of the solution. Else

[*] Received by the editors August 10, 1990; accepted for publication (in revised form) October 21, 1991.

[†] Universität Trier, FB IV—Mathematik, Postfach 3825, D-5500 Trier, Germany (na.heinken@na-net.ornl.gov). This research was partially supported by the Gottlieb-Daimler und Karl-Benz-Stiftung, Ladenburg, and National Science Foundation, Cooperative Agreement CCR-8809615.

(3) Set $k = k + 1$ and goto (1)

Reviewing the convergence theorems for Gauss–Newton methods for unconstrained problems (see, e.g., [11], [12], and [14]), one expects a linear convergence rate for this algorithm if the starting point is sufficiently close to the solution of (1.1). Moreover, the speed of convergence should depend on the nonlinearity and size of the residual of F. A detailed convergence analysis confirming these considerations is given in §2.

Subproblems of the type (1.2) also arise in trust region methods for unconstrained optimization. Here, however, R is fixed and is not the variable trust region radius. Nevertheless, we may use efficient methods established for the solution of trust region subproblems to obtain x_{k+1}. Such methods are discussed, for example, in [12] and [25]. Hence, if a good initial point is available, problem (1.1) can theoretically be solved with the Gauss–Newton method as the outer iteration and an inner iteration scheme, e.g., the Newton or Hebden–Reinsch–Moré iteration [24, Algorithm 5.5], [25, p. 273], for the solution of (1.2).

For a globalization of the convergence one can add a line search or trust region strategy. The latter leads to minimization problems with two norm constraints instead of (1.2). Utilizing the special structure of this subproblem, it can be solved using efficient methods designed for the solution of minimization problems with quadratic objective and one simple norm constraint as in (1.2). However, in this paper we are only concerned with the local analysis and assume that a good estimation for the solution is available.

For the numerical solution one has to approximate the infinite-dimensional problem by introducing discretizations for the parameter space X and the output space Y.

It is to be expected that the underlying infinite-dimensional problem influences the behavior of the Gauss–Newton method applied to the discretized problem. Therefore, it is important to study the relation between the solution method applied to the infinite-dimensional problem and its application to the discretized problem as well as to give an analysis of the method under varying discretizations. If all quantities of the method, such as iterates, Lagrange multipliers, and convergence constants, depend continuously on the discretization, we say that the method is mesh independent. Mesh independence in its sharpest form is developed in [2] for Newton's method, where estimates are given which are uniform with respect to the iteration count. The influence of discretizations on Broyden's method is studied in [18]. There a weaker mesh independence property is proven, which does not guarantee uniform bounds on the error between infinite- and finite-dimensional iterates; the bounds depend on the iteration count.

Mesh independence is important for two reasons. First, it allows us to predict the convergence of the method applied to the discretized problem when the method has been analyzed for the infinite-dimensional problem. Second, it can be used to improve the performance of the method. Since we are interested in the solution of an infinite-dimensional problem, it is usually necessary to choose reasonably fine discretizations. This leads to a large number of variables in the discretized minimization problem and therefore to a large amount of work per iteration. If the method is fixed, the only possibility for reducing the total amount of work consists in the improvement of the starting value. For these problems it is obvious that we must use information from the coarse discretizations to obtain good starting points for the finer ones. This leads to mesh refinement strategies. Mesh independence is a theoretical justification for mesh refinement strategies and, moreover, can be used to design the refinement process and to predict the overall performance of the method.

The second point is not addressed in this paper, so we refer the interested reader to the literature, where several applications of refinement strategies can be found. Such strategies are presented in [1] and [17] for Newton's method; in [19] for quasi-Newton methods, and in [16] for the Gauss–Newton method.

In this paper we extend the mesh independence results of [2] to the norm constrained Gauss–Newton method, but we use a somewhat different discretization scheme based on Galerkin approximations. We will assume that X_M and Y_N are finite-dimensional linear subspaces of X and Y, respectively, and that $F_N : X \to Y_N$ is a suitable approximation for F.

Although F_N is defined on the whole space X, it is evaluated only for some $x_M \in X_M$ during the numerical calculation. The discretized problem is then given as

$$\begin{array}{ll} \min & ||F_N(x^M)||_Y^2 \\ \text{s.t.} & ||x^M||_X \le R, \quad x^M \in X_M, \end{array} \tag{1.3}$$

and in the kth iteration of the Gauss–Newton method the current iteration point $x_k^{MN} \in \overline{B_R(0)} \cap X_M$ is given and we must solve

$$\begin{array}{ll} \min & ||F_N(x_k^{MN}) + F_N'(x_k^{MN})(x^M - x_k^{MN})||_Y^2 \\ \text{s.t.} & ||x^M||_X \le R, \quad x^M \in X_M \end{array} \tag{1.4}$$

instead of (1.2). Throughout the paper we will denote the iterates of the Gauss–Newton method applied to (1.3) by x_k^{MN} and the corresponding Lagrange multipliers by μ_k^{MN}. For the solution of (1.4) we have to compute the adjoints of $F_N'(x_k^{MN})$. Since we are working in the finite-dimensional spaces, we define the adjoint $F_N'(x)^* \in L(Y, X_M)$ through

$$\langle F_N'(\bar{x})^* y^N, x^M \rangle_X = \langle y^N, F_N'(\bar{x}) x^M \rangle_Y \quad \forall\, x^M \in X_M, y^N \in Y_N \,.$$

$F_N'(x)^*$ can be any extension of the $(X_M, ||\cdot||_X), (Y_N, ||\cdot||_Y)$ adjoint of $F_N'(x)$ onto Y. We need the extensions of F_N, $F_N'(x)$, and $F_N'(x)^*$ to apply these operators to points that are not contained in the finite-dimensional subspaces. This allows us to compare infinite- and finite-dimensional terms without prolongation or restriction operators. For finite element discretizations these extensions are given in a natural way (see also §4).

It is important to note that $F_N'(x)^*$ is an extension of the $(X_M, ||\cdot||_X), (Y_N, ||\cdot||_Y)$ adjoint onto Y, but not the adjoint for the pair $(X, ||\cdot||_X), (Y, ||\cdot||_Y)$ since in general we do not have

$$\langle F_N'(\bar{x})^* y, x \rangle_X = \langle y, F_N'(\bar{x}) x \rangle_Y \quad \forall\, x \in X, \quad y \in Y \,.$$

A consequence of this fact is that

$$||F_N'(\bar{x})^* - F'(\bar{x})^*||_{L(Y,X)} \neq ||F_N'(\bar{x}) - F'(\bar{x})||_{L(X,Y)},$$

and therefore we have to impose different approximation properties on the function and its derivative, on one hand, and on the adjoint of its derivative on the other. Since F_N is defined on X, it is evident that the approximation properties of F_N and F_N' are affected only by the discretization of Y, whereas the quality of approximation of ${F_N'}^*$ is also influenced by the discretization of X. We now list the assumptions we impose on X_M, Y_N and on the function F and its discretizations.

ASSUMPTIONS.

(A1) $F \in C^1(B_R(0))$.

(A2) $||F^{(i)}(x) - F^{(i)}(y)|| \leq L_i||x-y||$ for all $x, y \in B_R(0)$, $i = 0, 1$.

(A3) $F_N \in C^1(B_R(0))$.

(A4) There exist uniformly bounded Lipschitz constants L_i^N, $i = 0, 1$ such that $||F_N^{(i)}(x) - F_N^{(i)}(y)|| \leq L_i^N||x-y||$, $i = 0, 1$, for all $x, y \in B_R(0)$ and for all N. Without loss of generality we assume that $L_i^N \leq L_i, i = 0, 1$, for all $N \in I\!N$.

(A5) There exists a bounded function $\rho_Y : [0,1] \rightarrow I\!R^+$ which is continuous at 0 with $\rho_Y(0) = 0$ and satisfies $||F^{(i)}(x) - F_N^{(i)}(x)|| \leq \rho_Y(1/N)$, $i = 0, 1$, for all $x, y \in B_R(0)$ and for all N.

(A6) For every x and $\delta > 0$ there exists $M_{\delta,x}$, such that for all $M \geq M_{\delta,x}$ there exists $x_M \in X_M$ with $||x - x_M|| \leq \delta$.

(A7) There exists a bounded function $\rho_X : [0,1] \longrightarrow I\!R^+$ which is continuous at 0 with $\rho_X(0) = 0$, such that the adjoints of the original and discretized Fréchet derivatives obey $||F'(x)^* - F_N'(x)^*|| \leq \rho_Y(1/N) + \rho_X(1/M)$ for all $x \in B_R(0)$.

This setting is suitable for finite element discretizations and, as already pointed out, allows us to compare the discretized and infinite-dimensional terms without the incorporation of prolongation and restriction operators. Another more important gain is that we obtain uniform bounds for $||x_k - x_k^{MN}||$, which we would not obtain with the method of [2], where for Newton's method the finite-dimensional iterates x_k^{MN} are compared with projections of the infinite-dimensional ones, $\Pi_M x_k$, and Π_M denotes the projection of X onto X_M. These uniform bounds enable us to deduce estimates for the error between the solution of (1.1) and the solutions of the discretized problems, which improve estimates derived from perturbation theory for infinite-dimensional optimization problems. In this sense the Gauss–Newton method can be viewed as a tool for the analysis of (1.1) and its discretizations.

The sufficient conditions for mesh independence are strongly related to the conditions that are sufficient for the convergence of the Gauss–Newton method and throughout the paper we will use these conditions to formulate our mesh independent results.

Throughout the paper, we let x_* be a (local) solution of (1.1) with corresponding Lagrange multiplier μ_*. $\{x_k\}_{I\!N}$ always denotes the sequence generated by Algorithm 1.1 and $\{x_k^{MN}\}_{I\!N}$ denotes its discrete analogue (see (1.3) and (1.4)).

The outline of this paper is as follows: In §2 we present a convergence analysis for the algorithm stated above. In addition to the convergence theorem, we will give a result concerning the perturbation of solutions of (1.1) in the presence of discretization. This result is based on perturbation theory for infinite-dimensional optimization problems. In §3 we will develop the mesh independence principle for the Gauss–Newton method and in §4 we will discuss its application to a boundary value problem and present some numerical results.

2. Local convergence. The Gauss–Newton method for unconstrained problems has been intensively studied, and convergence results for this case can be found, for example, in [11], [12], [13], [14], and [27]. Algorithms for the solution of nonlinear least squares problems with equality constraints based on Gauss–Newton sequential quadratic programming (SQP)–like approaches are described and analyzed, for example, in [6] and [29]. Our approach is different in that we keep the original constraint and solve in each iteration a subproblem with quadratic objective function and quadratic constraint.

In this section we present a convergence theory for our algorithm, which generalizes Theorem 10.2.1 in [12] and partly generalizes the results in [14]. In [28], Vogel also uses Algorithm 1.1 and gives a convergence theorem. He uses second-order information to formulate and prove his results. In the proof of his result he distinguishes between whether the constraint is active at the solution x_* or not. If the constraint is inactive, he uses the results in [12]; if the constraint is active at x_*, he applies techniques similar to those used in the convergence proofs of (Newton) SQP methods. This may give an imprecise description of the algorithm when the constraint is active and $\mu_* = 0$. In this case, in the proof given in [28], it is assumed that all iterates are also locally active, which may not hold.

In the analysis presented here we only use first-order information and we incorporate the special structure of the problem completely. This leads to stronger convergence results and yields estimates for the iterates and the Lagrange multipliers.

It is well known that the solutions of (1.2) can be characterized as solutions of the system of Kuhn–Tucker conditions.

$$\begin{aligned} (F'(x_k)^*F'(x_k) + \mu_{k+1}I)x_{k+1} &= -F'(x_k)^*(F(x_k) - F'(x_k)x_k), \\ \mu_{k+1}(||x_{k+1}||_X^2 - R^2) = 0, & \\ \mu_{k+1} \geq 0, \qquad ||x_{k+1}||_X^2 - R^2 \leq 0 \,. & \end{aligned} \tag{2.1}$$

The Kuhn–Tucker conditions for (1.1) at x_* are given by (2.1) with x_k, x_{k+1}, μ_{k+1} replaced by x_*, x_*, μ_*, respectively. It should be noted that

$$(F'(x_*)^*F'(x_*) + \mu_*I)x_* = -F'(x_*)^*(F(x_*) - F'(x_*)x_*)$$

is equivalent to the commonly used condition

$$F'(x_*)^*F(x_*) + \mu_* x_* = 0 \,. \tag{2.2}$$

For $\mu > 0$ let $x_k(\mu)$ be defined as the unique solution of

$$(F'(x_k)^*F'(x_k) + \mu I)x = -F'(x_k)^*(F(x_k) - F'(x_k)x_k), \tag{2.3}$$

and let $x_k(0)$ denote the minimum norm solution of (2.3) with $\mu = 0$. If $||x_k(0)||_X > R$, the problem of finding a solution of the Kuhn–Tucker system is equivalent to the computation of a positive root of

$$g_k(\mu) \equiv ||x_k(\mu)||_X^2 - R^2 \,. \tag{2.4}$$

On $[0, \infty)$, g_k is a convex and monotonically decreasing function with $g_k(\mu) \longrightarrow -R^2$ as $\mu \longrightarrow \infty$. Therefore, the root is uniquely determined. Furthermore, g_k is continuously differentiable on $(0, \infty)$ with derivative given by

$$g_k'(\mu) = -2\langle x_k(\mu), (F'(x_k)^*F'(x_k) + \mu I)^{-1} x_k(\mu)\rangle_X \,. \tag{2.5}$$

We will use the relation between Lagrange multiplier and iterate given through (2.1) as well as the special structure of g_k to prove the following convergence results.

To simplify notation we define

$$H(x, \mu) \equiv F'(x)^*F'(x) + \mu I.$$

LEMMA 2.1. *Let F satisfy* (A1) *and* (A2). *Assume further that there exist $\epsilon, \omega > 0$ and $\kappa \in (0,1)$ such that for all $x \in B_\epsilon(x_*) \cap \overline{B_R(0)}$, $\mu \in B_\epsilon(\mu_*) \cap I\!R^+$, and $t \in [0,1]$ the following conditions hold:*

$$||H(x,\mu)^{-1}(F'(x)^* - F'(x_*)^*)F(x_*)|| \leq \kappa ||x - x_*|| \,, \tag{2.6}$$

$$||H(x,\mu)^{-1}F'(x)^*(F'(x_* + t(x - x_*)) - F'(x))(x - x_*)|| \leq \omega t ||x - x_*||^2. \tag{2.7}$$

Then there exist $\bar{\epsilon} \in (0, \epsilon]$, $\tau > 0$, and $\theta > 0$ such that for $x_k \in B_{\bar{\epsilon}}(x_*) \cap \overline{B_R(0)}$ and $\mu_k \in B_{\bar{\epsilon}}(\mu_*) \cap I\!R^+$ the following inequalities are valid

$$|\mu_* - \mu_{k+1}| \leq \theta \left(\kappa ||x_* - x_k|| + \frac{\omega}{2} ||x_* - x_k||^2 \right) + \theta ||x_* - x_{k+1}||, \tag{2.8}$$

$$|\mu_* - \mu_{k+1}| \leq \tau ||x_* - x_k|| \,. \tag{2.9}$$

Proof. First, we will collect a few technical details and definitions. Define

$$s \equiv \sup_{x \in \overline{B_R(0)}} ||F'(x)^*(F(x) - F'(x)x)|| \quad \text{and} \quad \Lambda^2 \equiv \sup_{x \in \overline{B_R(0)}} ||F'(x)^* F'(x)||.$$

For $A \in L(X, Y)$, $b \in X$, and $\mu > 0$ it holds that

$$||(A^*A + \mu I)^{-1} b|| \leq \frac{1}{\mu} ||b||. \tag{2.10}$$

If we set $A = F'(x_k)$ and $b = F'(x_k)^*(F(x_k) - F'(x_k)x_k)$, and if we assume that $\mu_{k+1} > 0$, then the complementarity condition $R = ||x_k(\mu_{k+1})|| = ||x_{k+1}||$, (2.3), and (2.10) yield

$$\mu_{k+1} \leq \frac{||F'(x_k)^*(F(x_k) - F'(x_k)x_k)||}{R} \leq \frac{s}{R}. \tag{2.11}$$

Inequality (2.11) is clearly also valid for $\mu_{k+1} = 0$. Similarly,

$$\mu_* \leq \frac{||F'(x_*)^*(F(x_*) - F'(x_*)x_*)||}{R} \leq \frac{s}{R}. \tag{2.12}$$

Let

$$\xi \equiv \min \left(R^2, \frac{1}{4(\Lambda^2 + s/R)^2} ||F'(x_*)^*(F(x_*) - F'(x_*)x_*)||^2 \right).$$

We will show that (2.8) holds with

$$\theta \equiv \frac{(\Lambda^2 + s/R)(c + R)}{2\xi}, \tag{2.13}$$

where c is a constant such that $||x_k(\mu_*)|| \leq c$ for all k (the existence of such a constant will be established below).

From the definition of $x_k(\mu)$, and from (2.3), we can conclude that

$$\begin{aligned} ||x_k(\mu_*)|| &\geq \frac{1}{\Lambda^2 + \mu_*} ||F'(x_k)^*(F(x_k) - F'(x_k)x_k)|| \\ &\geq \frac{1}{\Lambda^2 + \mu_*} (||F'(x_*)^*(F(x_*) - F'(x_*)x_*)|| - L||x_* - x_k||) \end{aligned} \tag{2.14}$$

(here, L is a Lipschitz constant depending on the Lipschitz constants of F, F' as well as $\Lambda^2, \sup_{x \in \overline{B_R(0)}} ||F(x)||,$ and R). Moreover, the definitions of $x_k(\mu)$ and (2.2) yield

$$\begin{aligned} x_* - x_k(\mu_*) = &- (F'(x_k)^* F'(x_k) + \mu_* I)^{-1} [(F'(x_k)^* F'(x_k) + \mu_* I)(x_k - x_*) \\ &- (F'(x_k)^* F(x_k) + \mu_* x_k \\ &- F'(x_k)^* F(x_*) - \mu_* x_*) \\ &+ (F'(x_*) - F'(x_k))^* F(x_*)]. \end{aligned}$$

With (2.6) and (2.7) the latter inequality implies

$$||x_* - x_k(\mu_*)|| \le \kappa||x_* - x_k|| + \frac{\omega}{2}||x_* - x_k||^2, \tag{2.15}$$

provided that $||x_* - x_k|| \le \epsilon$.

Inequality (2.15) guarantees the existence of c such that $||x_k(\mu_*)|| \le c$ independently of k.

Note that μ_{k+1} and μ_* are either 0 or they are the roots of g_k and g_*, respectively. In either case the special structure of these functions will be used to derive bounds for the Lagrange multipliers and estimates for the error $|\mu_* - \mu_{k+1}|$. If $\mu_* = \mu_{k+1}$, then (2.8) and (2.9) trivially hold. For the derivation of the estimates in the nontrivial case we consider the cases $\mu_* > \mu_{k+1}$ and $\mu_* < \mu_{k+1}$.

First, we consider the case $\mu_* > \mu_{k+1}$. Since g_k is convex, we obtain that

$$0 \ge g_k(\mu_{k+1}) \ge g_k(\mu_*) + g_k'(\mu_*)(\mu_{k+1} - \mu_*).$$

Using this equation, (2.5), and the complementarity condition $||x_*|| = R$, we find that

$$\begin{aligned}
\mu_* - \mu_{k+1} &\le \frac{g_k(\mu_*)}{g_k'(\mu_*)} \\
&= -\frac{||x_k(\mu_*)||^2 - R^2}{2\langle x_k(\mu_*), (F'(x_k)^*F'(x_k) + \mu_* I)^{-1} x_k(\mu_*)\rangle} \\
&\le \frac{\Lambda^2 + \mu_*}{2} \frac{R^2 - ||x_k(\mu_*)||^2}{||x_k(\mu_*)||^2} \\
&\le \frac{\Lambda^2 + \mu_*}{2} \frac{||x_*|| + ||x_k(\mu_*)||}{||x_k(\mu_*)||^2} ||x_* - x_k(\mu_*)|| \\
&\le \frac{\Lambda^2 + \mu_*}{2} \frac{R + c}{||x_k(\mu_*)||^2} ||x_* - x_k(\mu_*)|| .
\end{aligned} \tag{2.16}$$

If we choose

$$\bar{\epsilon} \equiv \min\left\{\epsilon, \frac{1}{2L}||F'(x_*)^*(F(x_*) - F'(x_*)x_*)||\right\},$$

we obtain with (2.12), (2.13), (2.14), (2.15), and (2.16) that, for $||x_* - x_k|| \le \bar{\epsilon}$,

$$\mu_* - \mu_{k+1} < \theta\left(\kappa||x_* - x_k|| + \frac{\omega}{2}||x_* - x_k||^2\right). \tag{2.17}$$

When $\mu_* < \mu_{k+1}$ we again use the convexity of g_k and $g_k(\mu_{k+1}) = 0$ to conclude that

$$\begin{aligned}
g_k(\mu_*) \ge g_k(\mu_{k+1}) + g_k'(\mu_{k+1})(\mu_* - \mu_{k+1}) &= |g_k'(\mu_{k+1})|(\mu_{k+1} - \mu_*) \\
&\ge \frac{2}{\Lambda^2 + \mu_{k+1}} ||x_k(\mu_{k+1})||^2 (\mu_{k+1} - \mu_*) .
\end{aligned}$$

With $||x_{k+1}|| = ||x_k(\mu_{k+1})|| = R$ and (2.15) this implies

$$\begin{aligned}
\mu_{k+1} - \mu_* &\le \frac{\Lambda^2 + \mu_{k+1}}{2R^2}(||x_k(\mu_*)||^2 - ||x_{k+1}||^2) \\
&\le \frac{\Lambda^2 + \mu_{k+1}}{2R^2}(c + R)(||x_k(\mu_*) - x_*|| + ||x_* - x_{k+1}||) \\
&\le \theta\left(\kappa||x_* - x_k|| + \frac{\omega}{2}||x_* - x_k||^2\right) + \theta||x_* - x_{k+1}|| .
\end{aligned} \tag{2.18}$$

Equations (2.17) and (2.18) yield the estimate (2.8).

From $||x_k|| \le R = ||x_{k+1}||$ we find, analogously to the derivation of (2.18), that

$$\mu_{k+1} - \mu_* \le \theta \left(\kappa ||x_* - x_k|| + \frac{\omega}{2} ||x_* - x_k||^2 \right) + \theta ||x_* - x_k|| \,. \tag{2.19}$$

Setting $\tau \equiv \theta(\kappa + \bar{\epsilon}\omega/2 + 1)$, (2.9) follows from (2.17) and (2.19). □

To guarantee the convergence of the iterates we have to replace (2.6) and (2.7) by stronger conditions. The following theorem is a generalization of Theorem 10.2.1 in [12].

THEOREM 2.2. *Let F satisfy* (A1) *and* (A2). *Assume further that for $\epsilon, \gamma_*, \sigma \ge 0$, and that for all $x, y \in \overline{B_R(0)} \cap B_\epsilon(x_*)$, $h \in \{h \in X |\, x_* + h \in \overline{B_R(0)}\,\}$,*

$$||F'(x_*)h||^2 \ge \gamma_* ||h||^2, \tag{2.20}$$

and

$$||(F'(x)^* - F'(x_*)^*)F(x_*)|| \le \sigma ||x - x_*|| \,. \tag{2.21}$$

Define $\Lambda = \sup_{x \in \overline{B_R(0)}} ||F'(x)||$. If $\sigma < \gamma_* + \mu_*$, then for all $\alpha \in (1, (\gamma_* + \mu_*)/\sigma)$ there exists $\epsilon_* = \epsilon_*(\alpha)$, $\epsilon_* > 0$ such that the solution x_{k+1} of (1.2) *obeys*

$$||x_{k+1} - x_*|| \le \frac{\alpha\sigma}{\gamma_* + \mu_*} ||x_k - x_*|| + \frac{\alpha L_1 \Lambda}{2(\gamma_* + \mu_*)} ||x_k - x_*||^2 \tag{2.22}$$

and

$$||x_{k+1} - x_*|| \le \frac{\gamma_* + \mu_* + \alpha\sigma}{2(\gamma_* + \mu_*)} ||x_k - x_*|| < ||x_k - x_*|| \,, \tag{2.23}$$

provided that $x_k \in B_{\epsilon_*}(x_*) \cap \overline{B_R(0)}$.

Proof. Let $\alpha \in (1, (\gamma_* + \mu_*)/\sigma)$ be an arbitrary constant. Since F' is continuous, we obtain from (2.20) the existence of $\epsilon_1 \in (0, \epsilon)$, such that for all $x \in \overline{B_R(0)} \cap B_{\epsilon_1}(x_*)$, $\mu \in B_{\epsilon_1}(\mu_*) \cap \mathbb{R}^+$, and $h \in \{h \in X |\, x_* + h \in \overline{B_R(0)}\,\}$, the following inequality holds:

$$\langle H(x, \mu)h, h \rangle \ge \frac{\gamma_* + \mu_*}{\alpha} ||h||^2. \tag{2.24}$$

Since the assertions (2.20), (2.21) imply the assertions in Lemma 2.1 with $\kappa = \alpha\sigma/(\gamma_* + \mu_*)$ and $\omega = \alpha L_1 \Lambda/(\gamma_* + \mu_*)$, there exists $\epsilon_2 \in (0, \epsilon_1)$ such that $||x_k - x_*|| \le \epsilon_2$ implies $\mu \in B_{\epsilon_1}(\mu_*) \cap \mathbb{R}^+$.

From the necessary optimality conditions we obtain the identities

$$\begin{aligned} (F'(x_k)^* F'(x_k) + \mu_{k+1} I) x_{k+1} &= -F'(x_k)^* (F(x_k) - F'(x_k) x_k) \,, \\ F'(x_*)^* F(x_*) + \mu_* x_* &= 0. \end{aligned}$$

These yield

$$\begin{aligned} H(x_k, \mu_{k+1})(x_{k+1} - x_*) = {} & F'(x_k)^* (F(x_*) - F(x_k) - F'(x_k)(x_* - x_k)) \\ & + (F'(x_*)^* - F'(x_k)^*) F(x_*) + (\mu_* - \mu_{k+1}) x_* \end{aligned} \tag{2.25}$$

and

$$\begin{aligned} H(x_k, \mu_*)(x_{k+1} - x_*) = {} & F'(x_k)^* (F(x_*) - F(x_k) - F'(x_k)(x_* - x_k)) \\ & + (F'(x_*)^* - F'(x_k)^*) F(x_*) + (\mu_* - \mu_{k+1}) x_{k+1} . \end{aligned} \tag{2.26}$$

If $\mu_* > \mu_{k+1}$, then $R = ||x_*|| \geq ||x_{k+1}||$, and thus

$$\langle x_*, x_{k+1} - x_* \rangle \leq ||x_*|| \, ||x_{k+1}|| - ||x_*||^2 \leq 0.$$

Combining this inequality with (2.25), we obtain that for $||x_k - x_*|| \leq \epsilon_2$ the following inequality holds:

$$\begin{aligned} &\langle H(x_k, \mu_{k+1})(x_{k+1} - x_*), x_{k+1} - x_* \rangle \\ &\leq \langle F'(x_k)^*(F(x_*) - F(x_k) - F'(x_k)(x_* - x_k)), x_{k+1} - x_* \rangle \\ &\quad + \langle (F'(x_*)^* - F'(x_k)^*)F(x_*), x_{k+1} - x_* \rangle. \end{aligned}$$

Together with (2.24) and (2.21), this yields

$$\frac{\gamma_* + \mu_*}{\alpha} ||x_{k+1} - x_*||^2 \leq \left(\frac{L_1 \Lambda}{2} ||x_k - x_*||^2 + \sigma ||x_k - x_*|| \right) ||x_{k+1} - x_*||. \tag{2.27}$$

If $\mu_* \leq \mu_{k+1}$, then we can proceed analogously and obtain

$$\langle x_{k+1}, x_* - x_{k+1} \rangle \leq 0,$$

$$\begin{aligned} &\langle H(x_k, \mu_*)(x_{k+1} - x_*), x_{k+1} - x_* \rangle \\ &\leq \langle F'(x_k)^*(F(x_*) - F(x_k) - F'(x_k)(x_* - x_k)), x_{k+1} - x_* \rangle \\ &\quad + \langle (F'(x_*)^* - F'(x_k)^*)F(x_*), x_{k+1} - x_* \rangle. \end{aligned}$$

Hence for $||x_k - x_*|| \leq \epsilon_2$,

$$\frac{\gamma_* + \mu_*}{\alpha} ||x_{k+1} - x_*||^2 \leq \left(\frac{L_1 \Lambda}{2} ||x_k - x_*||^2 + \sigma ||x_k - x_*|| \right) ||x_{k+1} - x_*||. \tag{2.28}$$

The estimates (2.27) and (2.28) yield (2.22). The local q-linear convergence follows from (2.22) if we set

$$\epsilon_*(\alpha) = \min \left\{ \epsilon_2, \frac{\gamma_* + \mu_* - \alpha\sigma}{\alpha L_1 \Lambda} \right\}.$$

For $||x_k - x_*|| \leq \epsilon_*(\alpha)$ we obtain

$$\begin{aligned} ||x_{k+1} - x_*|| &\leq \frac{\alpha\sigma}{\gamma_* + \mu_*} ||x_k - x_*|| + \frac{\alpha L_1 \Lambda}{2(\gamma_* + \mu_*)} ||x_k - x_*||^2 \\ &\leq \left(\frac{\alpha\sigma}{\gamma_* + \mu_*} + \frac{\alpha L_1 \Lambda}{2(\gamma_* + \mu_*)} \frac{\gamma_* + \mu_* - \alpha\sigma}{\alpha L_1 \Lambda} \right) ||x_k - x_*|| \\ &\leq \frac{\gamma_* + \mu_* + \alpha\sigma}{2(\gamma_* + \mu_*)} ||x_k - x_*|| < ||x_k - x_*||. \qquad \square \end{aligned}$$

Remark 2.3. (i) For unconstrained problems local q-linear convergence can be guaranteed under the weaker conditions (2.6) and (2.7). This follows, since

$$\begin{aligned} x_* - x_{k+1} = &- (F'(x_k)^* F'(x_k))^{-1} [F'(x_k)^* \{F'(x_k)(x_k - x_*) - F(x_k) + F(x_*)\} \\ &+ (F'(x_*) - F'(x_k))^* F(x_*)] \end{aligned}$$

implies

$$||x_* - x_{k+1}|| \le \kappa ||x_* - x_k|| + \frac{\omega}{2}||x_* - x_k||^2$$

(compare with the proof of Lemma 2.1). Conditions of the type (2.6), (2.7) are also used in [13] and [14] for the analysis of the Gauss–Newton method for unconstrained problems.

(ii) Under the conditions of Theorem 2.2 the results of Lemma 2.1 can be strengthened. With $\kappa = \alpha\sigma/(\gamma_* + \mu_*)$ and $\omega = \alpha L_1 \Lambda/(\gamma_* + \mu_*)$ we obtain from (2.8) and (2.22) that

$$|\mu_* - \mu_{k+1}| \le 2\theta \left(\kappa ||x_* - x_k|| + \frac{\omega}{2}||x_* - x_k||^2\right).$$

We conclude this section with an analysis of the assumptions made in Lemma 2.1 and Theorem 2.2. A similar analysis is given in [6].

LEMMA 2.4. *Let $F \in C^1(\overline{B_R(0)})$. Moreover, assume that $F''(x_*)$ exists and that $H(x_*, \mu_*)$ is continuously invertible.*

(i) *If*

$$||(H(x_*, \mu_*))^{-1}(F'(x)^* - F'(x_*)^*)F(x_*)|| \le \kappa ||x - x_*|| \quad \forall x \in B_\epsilon(x_*) \cap \overline{B_R(0)},$$

then

$$||(H(x_*, \mu_*))^{-1}(F''(x_*)(\cdot, h))^* F(x_*)|| \le \kappa ||h|| \quad \forall h \in X.$$

(ii) *If*

$$||(H(x_*, \mu_*))^{-1}(F''(x_*)(\cdot, h))^* F(x_*)|| \le \hat{\kappa} ||h|| \quad \forall h \in X,$$

then for each $\kappa > \hat{\kappa}$ there exists $\epsilon > 0$ such that for all $x \in B_\epsilon(x_*) \cap \overline{B_R(0)}$,

$$||(H(x_*, \mu_*))^{-1}(F'(x)^* - F'(x_*)^*)F(x_*)|| \le \kappa ||x - x_*||.$$

Proof. (i) Define $Z(x_*) = \{h \in X \,|\, x_* + h \in \overline{B_R(0)}\,\}$ and assume that

$$||(H(x_*, \mu_*))^{-1}(F'(x)^* - F'(x_*)^*)F(x_*)|| \le \kappa ||x - x_*||$$

for all $x \in B_\epsilon(x_*) \cap \overline{B_R(0)}$. From the differentiability we obtain that

$$||(H(x_*, \mu_*))^{-1}(F''(x_*)(\cdot, h))^* F(x_*)|| \le (\kappa + \phi(||h||))||h|| \quad \forall h \in Z(x_*),$$

where ϕ is continuous at the origin and fulfills $\phi(0) = 0$. Since $(F''(x_*)(\cdot, h))^*$ is linear in h, the inequality remains valid if we multiply h on both sides by a positive constant. By the continuity of ϕ for each $n \in \mathbb{N}$ there exists $\delta_n > 0$ such that $\phi(||h||) < 1/n$ for all $h \in B_{\delta_n}(0)$. This yields

$$\left\|(H(x_*, \mu_*))^{-1}\left(F''(x_*)\left(\cdot, \frac{\delta_n}{||h||}h\right)\right)^* F(x_*)\right\| \le \left(\kappa + \frac{1}{n}\right)\delta_n \quad \forall h \in Z(x_*)\backslash\{0\}.$$

Multiplying both sides by $||h||/\delta_n$ and taking the limit $n \to \infty$ gives

$$||(H(x_*, \mu_*))^{-1}(F''(x_*)(\cdot, h))^* F(x_*)|| \le \kappa ||h|| \quad \forall h \in Z(x_*). \tag{2.29}$$

Since $h \to (F''(x_*)(\cdot,h))^*F(x_*)$ is linear, the set $Z(x_*)$ in (2.29) can be replaced by

$$\{h \in X \,|\, \exists t : x_* + th \in \overline{B_R(0)}\} \supset \{h \in X \,|\, \langle x_*, h\rangle \neq 0\}.$$

(Both sets are equal if $||x_*|| = R$.) Finally, the continuity of $F''(x_*)(\cdot,\cdot)$ implies that

$$||(H(x_*,\mu_*))^{-1}(F''(x_*)(\cdot,h))^*F(x_*)|| \leq \kappa||h|| \quad \forall h \in \overline{\{h|\, \langle x_*, h\rangle \neq 0\}} = X.$$

(ii) The second assertion can be proved in a similar way. □

LEMMA 2.5. *Let $F \in C^2(\overline{B_R(0)})$. If $H(x_*,\mu_*)$ is continuously invertible the following statements are equivalent:*

(i) *There exists $\gamma > 0$ with*

$$\langle H(x_*,\mu_*)h,h\rangle - |\langle (F''(x_*)(\cdot,h))^*F(x_*),h\rangle| \geq \gamma||h||^2 \quad \forall h \in X. \tag{2.30}$$

(ii) *There exists $\kappa < 1$ with*

$$||(H(x_*,\mu_*))^{-1}(F''(x_*)(\cdot,h))^*F(x_*)|| \leq \kappa||h|| \quad \forall h \in X. \tag{2.31}$$

Proof. First we prove that (i) implies (ii). The operator $h \to F''(x_*)(\cdot,h)^*F(x_*)$ is self-adjoint, since $F \in C^2(\overline{B_R(0)})$. Using the existence of the square root of $H(x_*,\mu_*)$, e.g., [9, Thm. 4.6.2], (2.30) with the variable transformation $h \to H(x_*,\mu_*)^{-\frac{1}{2}}h$ yields that for all $h \in X$,

$$|\langle H(x_*,\mu_*)^{-\frac{1}{2}}(F''(x_*)(\cdot,H(x_*,\mu_*)^{-\frac{1}{2}}h))^*F(x_*),h\rangle| \leq \left(1 - \frac{\gamma}{||H(x_*,\mu_*)||}\right)||h||^2.$$

The latter inequality implies [9, Thm. 4.4.5] that

$$||H(x_*,\mu_*)^{-\frac{1}{2}}(F''(x_*)(\cdot,H(x_*,\mu_*)^{-\frac{1}{2}}\cdot))^*F(x_*)|| \leq 1 - \frac{\gamma}{||H(x_*,\mu_*)||}.$$

Since

$$||H(x_*,\mu_*)^{-\frac{1}{2}}(F''(x_*)(\cdot,H(x_*,\mu_*)^{-\frac{1}{2}}\cdot))^*F(x_*)|| = ||H(x_*,\mu_*)^{-1}(F''(x_*)(\cdot,\cdot))^*F(x_*)||,$$

(2.31) holds with $\kappa = 1 - \gamma/||H(x_*,\mu_*)||$.

The implication (ii) ⇒ (i) follows by similar arguments. Here, one obtains $\gamma = (1-\kappa)/||H(x_*,\mu_*)^{-1}||$. □

Lemma 2.5 shows that in the situation of Theorem 2.2 the second-order sufficient optimality criteria is satisfied at x_*. In particular, we obtain that x_* is an isolated minimizer and that the objective in (1.1) possesses local quadratic growth [23, Thm. 5.6]. This requirement seems to be inappropriate, since parameter identification problems are often rank deficient and ill posed. But in the presence of ill-posedness one has to employ regularization techniques to stabilize the problem, i.e., to guarantee continuous dependence of solutions of (1.1) upon input data. Such a technique may be the Tikhonov regularization, where a regularization term of the form $\alpha||x||^2$ is added to the objective. Similarly, a regularization may be obtained by reducing R. Hence, under suitable assumptions on F and on the regularization, the regularized parameter identification problem may fit the requirements of Theorem 2.2. In fact, in [7] and [8] it is shown that the output least squares formulation of certain elliptic parameter identification problems exhibit quadratic growth for properly chosen regularization (see also §4).

The quadratic growth of the objective function can also be used to derive an estimate for the error between the solution of the infinite-dimensional problem and the solutions of the discretized ones.

THEOREM 2.6. *Let* (A1)–(A6) *be valid and assume that* F *and* F_N *are weakly continuous functions. If there exist* $\alpha > 0$ *and* $\epsilon > 0$ *such that for all* $x \in \overline{B_R(0)} \cap B_\epsilon(x_*)$,

$$||F(x)||^2 \geq ||F(x_*)||^2 + \alpha||x - x_*||^2 \tag{2.32}$$

holds, and if there exists a continuous function g *with* $g(0) = 0$ *and* $g(t) \geq t$ *for all* $t \in [0,1]$ *such that* $d(h_1, h_2) \equiv g(\rho_Y(|h_1 - h_2|))$ *defines a metric on* $[0,1]$, *then for all* $\delta > 0$ *there exists* M_δ *and* N_δ *such that for all* $M \geq M_\delta$, $N \geq N_\delta$ *the discretized problem*

$$\begin{array}{ll} \min & ||F_N(x^M)||^2 \\ s.t. & ||x^M|| \leq R, \quad x^M \in X_M \end{array} \tag{2.33}$$

has a solution x_*^{MN} *satisfying*

$$||x_* - x_*^{MN}|| \leq \delta\,.$$

Proof. For brevity we set $d_N = d(0, 1/N)$ and $\rho = \rho_Y(1/N)$. By (A2) and (A5) there exists $c > 0$ such that for all N sufficiently large and $x_1, x_2 \in \overline{B_R(0)}$,

$$||F(x_1)||^2 - ||F_N(x_2)||^2 \leq c(\rho + ||x_1 - x_2||) \leq c(d_N + ||x_1 - x_2||)\,.$$

This shows that the discretization of F defines a Lipschitzian perturbation. The results of Alt [3, Thms. 4 and 6] yield the existence of $\tilde{N}$, such that for each $N \geq \tilde{N}$ there exists a solution x_*^N of

$$\begin{array}{ll} \min & ||F_N(x)||^2 \\ \text{s.t.} & ||x|| \leq R \end{array}$$

with

$$||x_* - x_*^N|| \leq \tilde{c}\sqrt{d_N}, \tag{2.34}$$

where $\tilde{c}$ is independent of N.

In the next step we will analyze the behavior of the discretized objective near x_*^N. We will show that a perturbation of the growth function for the original objective describes the growth of the discretized one. This result will be used to prove the assertion of the theorem.

For $x \in B_\epsilon(x_*)$ we deduce from (2.32), (2.34), and assumption (A5) that

$$\begin{aligned} \rho^2 + 2\rho||F_N(x)|| + ||F_N(x)||^2 &\geq ||F(x)||^2 \\ &\geq ||F(x_*)||^2 + \alpha||x - x_*||^2 \\ &\geq ||F(x_*^N)||^2 - 2L_0||x_* - x_*^N||\,||F(x_*^N)|| \\ &\quad + \alpha||x - x_*^N||^2 - 2\alpha||x - x_*^N||\,||x_* - x_*^N|| \\ &\geq ||F_N(x_*^N)||^2 + \alpha||x - x_*^N||^2 - 2\rho||F_N(x_*^N)|| \\ &\quad - (2L_0||F(x_*^N)|| + 4R\alpha)\tilde{c}\sqrt{d_N}\,. \end{aligned}$$

Let $\xi > 0$ be chosen with

$$\xi < \min\left\{\epsilon, \frac{\delta}{2}\right\}. \tag{2.35}$$

If we choose $N_\delta \geq \tilde{N}$ such that

$$\sqrt{d_N} \leq \xi/\tilde{c} \tag{2.36}$$

and

$$\rho^2 + 2\rho(||F_N(x)|| + ||F_N(x_*^N)||) + (2L_0||F(x_*^N)|| + 4R\alpha)\tilde{c}\sqrt{d_N} \leq \frac{\alpha\xi^2}{2}$$

for all $N \geq N_\delta$, then we obtain with (2.32), (2.34), and assumption (A2) the following growth condition for the finite-dimensional objective function:

$$||F_N(x)||^2 \geq ||F_N(x_*^N)||^2 + \alpha||x - x_*^N||^2 - \frac{\alpha\xi^2}{2} \quad \forall x \in \overline{B_\xi(x_*^N)}. \tag{2.37}$$

By (A6) there exists M_δ such that for all $M \geq M_\delta$ there exists $x^M \in X_M$ with

$$||x_*^N - x^M|| < \min\left\{\xi, -\frac{||F_N(x_*^N)||}{L_0} + \sqrt{\frac{||F_N(x_*^N)||^2}{L_0^2} + \frac{\alpha\xi^2}{4L_0^2}}\right\}. \tag{2.38}$$

Let x_*^{MN} denote a solution of

$$\begin{array}{ll} \min & ||F_N(x^M)||^2 \\ \text{s.t.} & ||x^M|| \leq R, \quad x^M \in \overline{B_\xi(x_*^N)} \cap X_M. \end{array}$$

In the next step we will show that x_*^{MN} is a local solution of (2.33), which will be proven if we show that $||x_*^N - x_*^{MN}|| < \xi$. Assume that $||x_*^N - x_*^{MN}|| = \xi$. Then (2.37) yields

$$||F_N(x_*^{MN})||^2 \geq ||F_N(x_*^N)||^2 + \alpha\xi^2 - \frac{\alpha\xi^2}{2}. \tag{2.39}$$

On the other hand, each $x^M \in \overline{B_R(0)}$ that obeys (2.38) satisfies

$$||F_N(x_*^{MN})||^2 \leq ||F_N(x^M)||^2$$

and

$$\begin{aligned} ||F_N(x^M)||^2 &\leq ||F_N(x_*^N)||^2 + 2L_0||F_N(x_*^N)||\,||x_*^N - x^M|| + L_0^2||x_*^N - x^M||^2 \\ &\leq ||F_N(x_*^N)||^2 + \frac{\alpha\xi^2}{4}. \end{aligned}$$

Hence, by (2.39),

$$||F_N(x_*^{MN})||^2 \leq ||F_N(x^M)||^2 \leq ||F_N(x_*^{MN})||^2 - \frac{\alpha\xi^2}{4},$$

a contradiction.

This gives the assertion, since each local minimizer x_*^{MN} of (2.33) fulfills

$$||x_*^{MN} - x_*|| \leq ||x_*^{MN} - x_*^N|| + ||x_*^N - x_*|| < \frac{\delta}{2} + \tilde{c}\sqrt{d_N} \leq \delta$$

(see (2.35), (2.36)). □

If we have $\rho_Y(h) = ch^p$ with $p \geq 1$, which is usually the case for finite element discretizations, we can choose $g(t) = t^{\frac{1}{p}}$. In view of Lemmas 2.4 and 2.5, (2.32) is clearly satisfied if $F \in C^2(\overline{B_R(0)})$ and if the assertions of Theorem 2.2 are valid.

Theorem 2.6 gives a qualitative result on the perturbations of solutions, but does not give error estimates for the difference between x_* and x_*^{MN}, although the derivation of the theorem indicates that $||x_*^M - x_*||$ is dominated by $\sqrt{d(0, 1/N)}$ and $||x_*^{MN} - x_*^N||$ by $\operatorname{dist}(X, X_M) = \sup_{x\in X} \inf_{x^M \in X_M} ||x^M - x||$. But note that since $M_{\delta,x}$ in (A6) depends on δ and x, the distance $\operatorname{dist}(X, X_M)$ may be infinite for fixed M. A detailed analysis of the Gauss–Newton method, which will be presented in the next section, will enable us to improve this theorem. We will derive error estimates related to the approximation properties of the discretization as well as uniqueness results for the minimizers of the discretized problems.

3. Mesh independence. In this section we will investigate the behavior of the Gauss–Newton method for the discretized problem. Our goal is to develop estimates for the difference between the Gauss–Newton iterates of the infinite- and finite-dimensional problem.

In what follows we will use some basic estimates, which are collected in the following lemma.

LEMMA 3.1. *Assume that* (A1), (A2), (A3), (A5), *and* (A7) *are valid. Define* $\rho = \rho_X(1/M) + \rho_Y(1/N)$. *Then there exist constants* $\tilde{c}_1$, c_2, *and* c_3, *independent of* M *and* N, *such that for all* $x, x^M, y \in \overline{B_R(0)}$, *and* $N \in I\!N$ *the following inequalities hold:*

$$||F_N'(x^M)^* F_N'(x^M) - F'(x)^* F'(x)|| \leq \tilde{c}_1(\rho + ||x - x^M||), \tag{3.1}$$

$$\begin{aligned} &||F'(x)^*(F(x) - F'(x)x) - F_N'(x^M)^*(F_N(x^M) - F_N'(x^M)x^M)|| \\ &\qquad \leq c_2(\rho + ||x - x^M||), \end{aligned} \tag{3.2}$$

$$||F'(x)^* F'(x) - F'(y)^* F'(y)|| \leq c_3 ||x - y||. \tag{3.3}$$

Proof. The proof is a straightforward application of (A1)–(A3), (A5), and (A7), and is therefore omitted. □

Before we derive the fundamental estimates for the iterates and Lagrange multipliers, we introduce some notation. Define

$$c_4 = \max\left\{ \sup_N \sup_{x\in\overline{B_R(0)}} ||F_N(x)||, \sup_{x\in\overline{B_R(0)}} ||F(x)|| \right\}$$

and

$$c_5 = \max\left\{ \sup_N \sup_{x\in\overline{B_R(0)}} ||F_N'(x)||, \sup_N \sup_{x\in\overline{B_R(0)}} ||F_N'(x)^*||, \sup_{x\in\overline{B_R(0)}} ||F'(x)|| \right\}.$$

Note that $c_4, c_5 < \infty$ by (A2), (A4), and (A5).

In the following proofs we will use the special representation of the iterates x_{k+1}, x_{k+1}^{MN}. We recall that

$$\begin{aligned} H(x,\mu) &\equiv F'(x)^*F'(x)+\mu I, \\ x_k(\mu) &\equiv -H(x_k,\mu)^{-1}F'(x_k)^*(F(x_k)-F'(x_k)x_k) \\ &= x_k - H(x_k,\mu)^{-1}(F'(x_k)^*F(x_k)+\mu x_k)\,, \end{aligned} \tag{3.4}$$

and we define $x_k^{MN}(\mu)$ to be the discrete analogue of $x_k(\mu)$ with F, F', x_k replaced by F_N, F_N', x_k^{MN}, and we define $H_N(x,\mu)$ to be the operator corresponding to $H(x,\mu)$ with F' replaced by F_N'.

With these abbreviations we obtain that $x_k(\mu_{k+1}) = x_{k+1}$ and $x_k^{MN}(\mu_{k+1}^{MN}) = x_{k+1}^{MN}$.

Again, we will use the convexity of the functions

$$\begin{aligned} g_k(\mu) &\equiv \|x_k(\mu)\|^2 - R^2, \\ g_k^{MN}(\mu) &\equiv \|x_k^{MN}(\mu)\|^2 - R^2 \end{aligned}$$

to derive the estimate for the Lagrange multipliers μ_{k+1}, μ_{k+1}^{MN}, which are given as the roots of g_k, g_k^{MN}, respectively. Due to the special structure of g_k, μ_{k+1} is bounded by

$$\frac{\|F'(x_k)^*(F(x_k)-F'(x_k)x_k)\|}{R},$$

as established in the proof of Lemma 2.1. Since F, F' are Lipschitz continuous, μ_{k+1} is uniformly bounded. The same is true for μ_{k+1}^{MN} by Lemma 3.1.

Finally, we set

$$c_1 = \max\{\tilde{c}_1, B\tilde{c}_1\}, \tag{3.5}$$

where B is an upper bound for $\|H(x_k,\mu)^{-1}\|$, $\|H(x_k,\mu_{k+1})^{-1}\|$.

LEMMA 3.2. *Assume that* (A1)–(A5) *and* (A7) *are valid and let* $\kappa \in (0,1)$, $B > 0$ *be such that*

$$\begin{aligned} \|H(x_k,\mu)^{-1}(F'(x_k)^* - F'(x_k^{MN})^*)F(x_k)\| &\le \kappa\|x_k^{MN} - x_k\|, \\ \|H(x_k,\mu)^{-1}\| &\le B. \end{aligned}$$

Define $e_k \equiv \|x_k^{MN} - x_k\|$ *and* $\rho \equiv \rho_X(\frac{1}{M}) + \rho_Y(\frac{1}{N})$. *If* $c_1(\rho + e_k) < 1$, *then there exists* $c_6 > 0$, *independent of* M, N, *and* k, *such that*

$$\|x_k^{MN}(\mu) - x_k(\mu)\| \le \frac{c_6e_k^2 + c_1(\rho+e_k)\|x_k(\mu)-x_k\| + \kappa e_k + c_6\rho}{1 - c_1(\rho+e_k)}. \tag{3.6}$$

Proof. From the definition of $x_k(\mu)$ and $x_k^{MN}(\mu)$ (see (3.4)), we obtain

$$\begin{aligned} x_k^{MN}(\mu) - x_k(\mu) \le H_N(x_k^{MN},\mu)^{-1}H(x_k,\mu) & \\ (H(x_k,\mu)^{-1}\{(F_N'(x_k^{MN})^*F_N'(x_k^{MN}) + \mu I)(x_k^{MN} - x_k) & \\ - (F_N'(x_k^{MN})^*F_N(x_k^{MN}) + \mu x_k^{MN} & \\ - F_N'(x_k^{MN})^*F_N(x_k) - \mu x_k)\} & \\ + H(x_k,\mu)^{-1}\{[F_N'(x_k^{MN})^*F_N'(x_k^{MN}) + \mu I & \\ - (F'(x_k)^*F'(x_k) + \mu I)] & \\ (F'(x_k)^*F'(x_k) + \mu I)^{-1}(F'(x_k)^*F(x_k) + \mu x_k)\} & \\ + H(x_k,\mu)^{-1}\{(F'(x_k)^*F(x_k) + \mu x_k) & \\ - (F_N'(x_k^{MN})^*F_N(x_k) + \mu x_k)\})\,. & \end{aligned} \tag{3.7}$$

For operators $A, C \in L(X, X)$ with continuous inverse we have the following equality:

$$A^{-1}C = A^{-1}CC^{-1}(C - A) + I.$$

From this it can be seen that if $||C^{-1}||\,||A - C|| < 1$

$$||A^{-1}C|| \leq \frac{1}{1 - ||C^{-1}||\,||A - C||}.$$

The application of the latter inequality to $H_N(x_k^{MN}, \mu)^{-1}H(x_k, \mu)$ together with (3.1) and (3.5) yields

$$||H_N(x_k^{MN}, \mu)^{-1}H(x_k, \mu)|| \leq \frac{1}{1 - c_1(\rho + e_k)}.$$

Using the basic estimates of Lemma 3.1, the terms in (3.7) can be estimated as follows:

$$\begin{aligned} &||F_N'(x_k^{MN})^* F_N'(x_k^{MN})(x_k^{MN} - x_k) - (F_N'(x_k^{MN})^* F_N(x_k^{MN}) - F_N'(x_k^{MN})^* F_N(x_k))|| \\ &\quad \leq ||F_N'(x_k^{MN})^*|| \quad ||F_N'(x_k^{MN})(x_k^{MN} - x_k) - F_N(x_k^{MN}) - F_N(x_k)|| \\ &\quad \leq c_5(L_1/2)e_k^2, \end{aligned}$$

$$\begin{aligned} &||(F_N'(x_k^{MN})^* F_N'(x_k^{MN}) - F'(x_k)^* F'(x_k))H(x_k, \mu)^{-1}(F'(x_k)^* F(x_k) + \mu x_k)|| \\ &\quad \leq \tilde{c}_1(\rho + e_k)||x_k(\mu) - x_k||, \end{aligned}$$

$$\begin{aligned} &||H(x_k, \mu)^{-1}(F'(x_k)^* F(x_k) - F_N'(x_k^{MN})^* F_N(x_k))|| \\ &\quad \leq ||H(x_k, \mu)^{-1}(F'(x_k)^* - F'(x_k^{MN})^*)F(x_k)|| \\ &\qquad + B||(F'(x_k^{MN})^* - F_N'(x_k^{MN})^*)F(x_k)|| \\ &\qquad + B||F_N'(x_k^{MN})^*||\,||F_N(x_k) - F(x_k)|| \\ &\quad \leq \kappa e_k + Bc_4\rho + Bc_5\rho_Y(1/N). \end{aligned}$$

Inserting these bounds into (3.7), we obtain the desired result by setting

$$c_6 \equiv \max\{Bc_5L_1/2, B(c_5 + c_4)\}. \qquad \square$$

For the derivation of the estimate for the Lagrange multipliers we will use the convexity of $||x_k(\mu)||^2 - R^2$ and its discretized analogue.

LEMMA 3.3. *Assume that* (A1), (A2), (A3), (A5), *and* (A7) *are valid. If* $||x_k^{MN}(\mu_{k+1}) - x_{k+1}|| < R$, *then there exists* c_7 *independent of* M *and* N *such that*

$$(3.8) \quad |\mu_{k+1}^{MN} - \mu_{k+1}| \leq \frac{c_7(1 + ||x_k^{MN}(\mu_{k+1})||)}{(1 - ||x_k^{MN}(\mu_{k+1}) - x_{k+1}||/R)^2}\,||x_k^{MN}(\mu_{k+1}) - x_{k+1}||.$$

Proof. For brevity we set

$$e_{k+1} \equiv ||x_k^{MN}(\mu_{k+1}) - x_{k+1}||.$$

If $\mu_{k+1} = \mu_{k+1}^{MN}$ the assertion follows immediately. Therefore let us assume that $\mu_{k+1} \neq \mu_{k+1}^{MN}$. From the definition of g_k, g_k^{MN} we obtain

$$(3.9) \quad \begin{aligned} |g_k(\mu) - g_k^{MN}(\mu)| &= (\,||x_k(\mu)|| + ||x_k^{MN}(\mu)||\,) \quad |\,||x_k(\mu)|| - ||x_k^{MN}(\mu)||\,| \\ &\leq 2R||x_k(\mu) - x_k^{MN}(\mu)||, \end{aligned}$$

provided $\mu \geq \max\{\mu_{k+1}, \mu_{k+1}^{MN}\}$, and

$$|g_k^{MN\prime}(\mu)| = 2\langle x_k^{MN}(\mu), (F_N'(x_k^{MN})^* F_N'(x_k^{MN}) + \mu I)^{-1} x_k^{MN}(\mu)\rangle.$$

Since $F_N'(x_k^{MN})^* F_N'(x_k^{MN})$ is self-adjoint on $(X_M, \langle\cdot,\cdot\rangle_X)$, it holds that for all $h_M \in X_M$

$$\langle (F_N'(x_k^{MN})^* F_N'(x_k^{MN}) + \mu I)^{-1} h_M, h_M\rangle \geq \frac{1}{||F_N'(x_k^{MN})||^2 + \mu} ||h_M||^2 .$$

Hence

$$|g_k^{MN\prime}(\mu)| \geq \frac{2}{c_5^2 + \mu} ||x_k^{MN}(\mu)||^2 . \tag{3.10}$$

Now we will combine the estimates above to develop the estimate for the error in the Lagrange multipliers. First let us consider the case $\mu_{k+1} < \mu_{k+1}^{MN}$. For $\mu \in [\mu_{k+1}, \mu_{k+1}^{MN}]$ we obtain, as in (3.9), that

$$g_k^{MN}(\mu) - (R + ||x_k^{MN}(\mu)||) ||x_k(\mu) - x_k^{MN}(\mu)|| \leq g_k(\mu) \leq 0 . \tag{3.11}$$

Since g_k^{MN} is convex and $g_k^{MN}(\mu_{k+1}^{MN}) = 0$ (keep in mind that $\mu_{k+1}^{MN} > 0$), we conclude that

$$g_k^{MN}(\mu) \geq |g_k^{MN\prime}(\mu_{k+1}^{MN})| \, |\mu - \mu_{k+1}^{MN}| .$$

With (3.10) and $||x_k^{MN}(\mu_{k+1}^{MN})|| = R$ this gives

$$g_k^{MN}(\mu) \geq \frac{2R^2}{c_5^2 + \mu_{k+1}^{MN}} |\mu - \mu_{k+1}^{MN}| . \tag{3.12}$$

Inserting (3.12) into (3.11) yields

$$\begin{aligned} 0 &\geq g_k^{MN}(\mu_{k+1}) - (R + ||x_k^{MN}(\mu_{k+1})||) \, e_{k+1} \\ &\geq \frac{2R^2}{c_5^2 + \mu_{k+1}^{MN}} |\mu_{k+1}^{MN} - \mu_{k+1}| - (R + ||x_k^{MN}(\mu_{k+1})||) \, e_{k+1} , \end{aligned}$$

respectively,

$$|\mu_{k+1}^{MN} - \mu_{k+1}| \leq \frac{c_5^2 + \mu_{k+1}^{MN}}{2R^2} (R + ||x_k^{MN}(\mu_{k+1})||) \, e_{k+1} . \tag{3.13}$$

In the case $\mu_{k+1} > \mu_{k+1}^{MN}$ we can proceed as follows. From the convexity of g_k^{MN} we obtain

$$0 \geq g_k^{MN}(\mu_{k+1}^{MN}) \geq g_k^{MN}(\mu_{k+1}) + g_k^{MN\prime}(\mu_{k+1})(\mu_{k+1}^{MN} - \mu_{k+1}).$$

This, together with the fact that $\mu_{k+1} > 0$ is the root of g_k yields

$$\mu_{k+1} - \mu_{k+1}^{MN} \leq \frac{g_k(\mu_{k+1}) - g_k^{MN}(\mu_{k+1})}{|g_k^{MN\prime}(\mu_{k+1})|} .$$

With the estimates (3.9), (3.10), and $||x_k(\mu_{k+1})|| = R$ we finally obtain

$$|\mu_{k+1}^{MN} - \mu_{k+1}| \leq \frac{R(c_5^2 + \mu_{k+1})}{||x_k^{MN}(\mu_{k+1})||^2} \, e_{k+1} \leq \frac{(c_5^2 + \mu_{k+1})/R}{(1 - e_{k+1}/R)^2} \, e_{k+1} . \tag{3.14}$$

Since the Lagrange multipliers μ_{k+1}, μ_{k+1}^{MN} are uniformly bounded the assertion follows from (3.13), (3.14), and $x_k(\mu_{k+1}) = x_{k+1}$. □

After providing these technical lemmas, we are able to prove our main result.

The assumptions required in the following theorem are closely related to the assumptions needed to guarantee local convergence of the Gauss–Newton method, but convergence of the method is not explicitly used. It should be noted that the requirement on the Lagrange multipliers in (3.15) is implied by condition (3.15) for the iterates if the assumptions of Lemma 2.1 hold. If the conditions of the convergence theorem, Theorem 2.2, are valid, then (3.15) is satisfied. For unconstrained problems (3.15) is implied by (3.16) and (A2), which was shown in Remark 2.3 (ii).

THEOREM 3.4. *Assume that* (A1)–(A5) *and* (A7) *are valid and that there exist* $\bar{\epsilon} > 0, \epsilon_* \in (0, \bar{\epsilon})$, *such that for all* $\epsilon \in (0, \epsilon_*)$ *the implications*

$$\begin{aligned} ||x_k - x_*|| < \epsilon \quad &\Longrightarrow \quad ||x_{k+1} - x_*|| < \epsilon, \\ ||x_k - x_*|| < \epsilon \quad &\Longrightarrow \quad |\mu_{k+1} - \mu_*| < \bar{\epsilon} \end{aligned} \tag{3.15}$$

hold. Moreover, let $\kappa \in (0,1)$, $B > 0$ *be such that for all* $x, y \in B_{\bar{\epsilon}}(x_*) \cap \overline{B_R(0)}$ *and* $\mu \in B_{\bar{\epsilon}}(\mu_*) \cap I\!R^+$ *the following conditions are valid:*

$$\begin{aligned} &||H(x,\mu)^{-1}(F'(x)^* - F'(y)^*)F(x)|| \le \kappa ||x - y||, \\ &||H(x,\mu)^{-1}|| \le B. \end{aligned} \tag{3.16}$$

Then there exists $\epsilon \in (0, \epsilon_*), c > 0$ (*both independent of* M, N), M_ϵ, N_ϵ, *and a function* $\tau : I\!N^2 \to I\!R^+$, *such that for all* $x_0 \in \overline{B_R(0)} \cap B_\epsilon(x_*)$, $M \ge M_\epsilon$, *and* $N \ge N_\epsilon$ *the condition* $||x_0 - x_0^{MN}|| \le \tau(M, N)$ *implies*

$$||x_k - x_k^{MN}|| \le c\left(\rho_X\left(\frac{1}{M}\right) + \rho_Y\left(\frac{1}{N}\right)\right) \quad \forall k \qquad \textit{and} \tag{3.17}$$

$$|\mu_k - \mu_k^{MN}| \le c\left(\rho_X\left(\frac{1}{M}\right) + \rho_Y\left(\frac{1}{N}\right)\right) \quad \forall k. \tag{3.18}$$

Proof. For brevity we define

$$\rho \equiv \rho_X\left(\frac{1}{M}\right) + \rho_Y\left(\frac{1}{N}\right) \quad \text{and} \quad e_k \equiv ||x_k - x_k^{MN}||.$$

The assertion will be proven by induction. However, the proof is quite technical, because we have to bound $|\mu_{k+1} - \mu_{k+1}^{MN}|$ and $||x_k(\mu_{k+1}) - x_k^{MN}(\mu_{k+1})||$ simultaneously.

In order to give a better idea why we have to choose the parameters in the ways to be specified, we will introduce the choices step by step.

In the first part we will derive bounds for $||x_k(\mu_{k+1}) - x_k^{MN}(\mu_{k+1})||$ and $|\mu_{k+1} - \mu_{k+1}^{MN}|$ based on the Lemmas 3.2 and 3.3. We then combine these results with a stability estimate for the solution of linear systems to obtain the desired inequalities.

Choose

$$\tilde{\epsilon} < \min\left\{\epsilon_*, \frac{1-\kappa}{8c_1}\right\}$$

and set $c_8 = \max\{1, 2\tilde{\epsilon}\}$.

Define

$$c_9 \equiv \frac{8(c_1 + c_6)}{3(1-\kappa)} c_8 \tag{3.19}$$

and let $\tilde{M}, \tilde{N}$ be such that

$$\rho < \min\left\{ \frac{1}{c_1 + c_1 c_9}, \frac{1-\kappa}{4c_1}, \frac{3(1-\kappa)^2}{64 c_8 (c_6 + c_1)^2}, \frac{R}{2c_9}, \frac{\epsilon_* - \tilde{\epsilon}}{c_9} \right\}$$

for all $M \geq \tilde{M}, N \geq \tilde{N}$.

Moreover, define

$$\Gamma(M, N, a, b, \epsilon) \equiv \left(\frac{1-\kappa}{4(a+b)} + \sqrt{\frac{(1-\kappa)^2}{16(a+b)^2} - \frac{2a\epsilon + b}{a+b}\rho} \right)^{-1} \frac{2a\epsilon + b}{a+b}\rho \tag{3.20}$$

and set

$$\tau_1(M, N) = \Gamma(M, N, c_1, c_6, \tilde{\epsilon}).$$

From $\rho < 3(1-\kappa)^2/(64c_8(c_6+c_1)^2)$ we obtain

$$\left(\frac{1-\kappa}{4(c_1+c_6)} + \sqrt{\frac{(1-\kappa)^2}{16(c_1+c_6)^2} - \frac{2c_1\tilde{\epsilon} + c_6}{c_1+c_6}\rho} \right) \geq \frac{3(1-\kappa)}{8(c_6+c_1)}.$$

With (3.19) and (3.20) this yields

$$\tau_1(M, N) \leq c_9\, \rho. \tag{3.21}$$

Now, assume that $M \geq \tilde{M}, N \geq \tilde{N}$, and $x_k \in B_{\tilde{\epsilon}}(x_*) \cap \overline{B_R(0)}$, and that x_k^{MN} is given such that

$$||x_k - x_k^{MN}|| \leq \tau_1(M, N).$$

For brevity we set $\tau_1 \equiv \tau_1(M, N)$. Then (3.21) and $\rho < (\epsilon_* - \tilde{\epsilon})/c_9$ imply

$$||x_* - x_k^{MN}|| \leq ||x_* - x_k|| + ||x_k - x_k^{MN}|| < \epsilon_*.$$

Thus, the inequalities (3.15) and (3.16) are satisfied for the triple $(x_k, x_k^{MN}, \mu_{k+1})$.

Lemma 3.2 and (3.15) and (3.16) yield

$$\begin{aligned} ||x_k(\mu_{k+1}) - x_k^{MN}(\mu_{k+1})|| &\leq \frac{c_6 e_k^2 + c_1(\rho + e_k)||x_k - x_{k+1}|| + \kappa e_k + c_6\rho}{1 - c_1\rho - c_1 e_k} \\ &\leq \frac{c_6 e_k^2 + 2c_1(\rho + e_k)\tilde{\epsilon} + \kappa e_k + c_6\rho}{1 - c_1\rho - c_1 e_k}. \end{aligned} \tag{3.22}$$

Since ρ is chosen less than $1/(c_1 + c_1 c_9)$ we obtain with (3.19) and (3.21) that the denominator of (3.22) is greater than 0. Therefore, (3.22) is well defined.

From $\tilde{\epsilon} \leq (1-\kappa)/(8c_1)$ and $\rho < (1-\kappa)/(4c_1)$ we find that

$$\kappa - 1 + 2c_1\tilde{\epsilon} + c_1\rho < (\kappa - 1)/2.$$

Therefore, we can continue to estimate (3.22) by

$$\begin{aligned} &||x_k(\mu_{k+1}) - x_k^{MN}(\mu_{k+1})|| \\ &\leq \frac{c_6\tau_1^2 + (\kappa - 1)\tau_1/2 + 2c_1\rho\tilde{\epsilon} + c_6\rho + (1 - c_1\rho)\tau_1}{1 - c_1\rho - c_1\tau_1} \\ &= \tau_1 , \end{aligned} \tag{3.23}$$

where for the last equation we used the fact that τ_1 is the smallest root of

$$(c_1 + c_6)\tau^2 + \frac{\kappa - 1}{2}\tau + 2c_1\rho\tilde{\epsilon} + c_6\rho = 0 .$$

Since $||x_k(\mu_{k+1})|| \leq R$ we obtain from (3.23) that $||x_k^{MN}(\mu_{k+1})||$ is bounded. Therefore, there exists $\tilde{c}_7$, independent of M, N, such that $||x_k^{MN}(\mu_{k+1})|| \leq \tilde{c}_7$ and $c_7(1 + ||x_k^{MN}(\mu_{k+1})||) \leq \tilde{c}_7$, where c_7 is defined in Lemma 3.3. From

$$\rho \leq \frac{R}{2c_9} ,$$

(3.21), and (3.23), we obtain

$$\frac{\tilde{c}_7}{(1 - ||x_k(\mu_{k+1}) - x_k^{MN}(\mu_{k+1})||/R)^2} \leq 4\tilde{c}_7 . \tag{3.24}$$

Since, up to the constant

$$\tilde{c}_7/(1 - ||x_k(\mu_{k+1}) - x_k^{MN}(\mu_{k+1})||/R)^2,$$

$|\mu_{k+1} - \mu_{k+1}^{MN}|$ is bounded by the same term as $||x_k(\mu_{k+1}) - x_k^{MN}(\mu_{k+1})||$, we obtain from (3.8), (3.24), and (3.23) that $||x_k - x_k^{MN}|| \leq \tau_1(M, N)$ implies

$$|\mu_{k+1} - \mu_{k+1}^{MN}| \leq 4\tilde{c}_7\tau_1(M, N) . \tag{3.25}$$

Together with (3.21), this gives the desired estimate for the Lagrange multipliers.

To prove the estimate for the iterates, we have to combine the previous results. Lemma 3.2 yields

$$\begin{aligned} &||x_{k+1} - x_{k+1}^{MN}|| \\ &= ||x_k(\mu_{k+1}) - x_k^{MN}(\mu_{k+1}^{MN})|| \\ &\leq ||x_k(\mu_{k+1}) - x_k^{MN}(\mu_{k+1})|| + ||x_k^{MN}(\mu_{k+1}) - x_k^{MN}(\mu_{k+1}^{MN})|| \\ &\leq \frac{c_6 e_k^2 + c_1(\rho + e_k)||x_{k+1} - x_k|| + \kappa e_k + c_6\rho}{1 - c_1\rho - c_1 e_k} \\ &\quad + ||x_k^{MN}(\mu_{k+1}) - x_k^{MN}(\mu_{k+1}^{MN})|| . \end{aligned} \tag{3.26}$$

If $A, \tilde{A} \in L(X, X)$ are continuously invertible with $||A^{-1}|| \, ||A - \tilde{A}|| < 1$, then

$$||A^{-1}b - \tilde{A}^{-1}b|| \leq \frac{||A^{-1}|| \, ||A - \tilde{A}||}{1 - ||A^{-1}|| \, ||A - \tilde{A}||} ||A^{-1}b||. \tag{3.27}$$

With Lemma 3.1 we get

$$||(F_N'(x_k^{MN})^* F_N'(x_k^{MN}) + \mu_{k+1} I)^{-1}|| \leq \frac{B}{1 - c_1(\rho + ||x_k - x_k^{MN}||)} . \tag{3.28}$$

Now (3.27) and (3.28) yield

$$\begin{aligned}
&||x_k^{MN}(\mu_{k+1}) - x_k^{MN}(\mu_{k+1}^{MN})|| \\
&\leq \frac{||(F_N'(x_k^{MN})^* F_N'(x_k^{MN}) + \mu_{k+1} I)^{-1}||\, |\mu_{k+1} - \mu_{k+1}^{MN}|\, ||x_k^{MN}(\mu_{k+1})||}{1 - ||(F_N'(x_k^{MN})^* F_N'(x_k^{MN}) + \mu_{k+1} I)^{-1}||\, |\mu_{k+1} - \mu_{k+1}^{MN}|} \\
&\leq \frac{|\mu_{k+1} - \mu_{k+1}^{MN}|\, B\tilde{c}_7}{1 - c_1\rho - c_1||x_k - x_k^{MN}|| - B|\mu_{k+1} - \mu_{k+1}^{MN}|}.
\end{aligned}$$

Define $c_{10} = c_1 + 4B\tilde{c}_7 c_9$ and $c_{11} = 4B\tilde{c}_7^2 c_9$. Then we conclude with (3.21) and (3.25) that

$$(3.29) \qquad ||x_k^{MN}(\mu_{k+1}) - x_k^{MN}(\mu_{k+1}^{MN})|| \leq \frac{c_{11}\rho}{1 - c_{10}\rho - c_{10}e_k},$$

provided $M \geq \tilde{M}, N \geq \tilde{N}$ and $e_k \leq \tau_1(M, N)$.

If we insert (3.29) into (3.26), we observe that $||x_{k+1} - x_{k+1}^{MN}||$ is bounded by a term which has the same structure as the bound in (3.22) (replacing c_1 by c_{10} and c_6 by $c_{12} \equiv c_6 + c_{11}$). Therefore, with the choices $M_\epsilon \geq \tilde{M}$, $N_\epsilon \geq \tilde{N}$ such that

$$\rho \leq \min\left\{\frac{3(1-\kappa)^2}{64c_8(c_{10}+c_{12})^2}, \frac{1-\kappa}{4c_{10}}\right\} \quad \forall M \geq M_\epsilon, \quad N \geq N_\epsilon,$$

$$\epsilon < \min\left\{\tilde{\epsilon}, \frac{1-\kappa}{8c_{10}}\right\},$$

and

$$\tau(M,N) \equiv \min\left\{\tau_1(M,N), \Gamma(M,N,c_{10},c_{12},\epsilon)\right\},$$

we finally obtain that $||x_k - x_k^{MN}|| \leq \tau(M,N)$ implies $||x_{k+1} - x_{k+1}^{MN}|| \leq \tau(M,N)$. This gives the assertion, since

$$\tau(M,N) \leq \frac{8(c_{10}+c_{12})}{3(1-\kappa)} c_8\rho. \qquad \square$$

To guarantee that the error between x_k and x_k^{MN} can be bounded by $\rho_X(1/M) + \rho_Y(1/N)$, we have to ensure that the starting point x_0^{MN} satisfies a certain approximation property, which is essentially $||x_0 - x_0^{MN}|| \leq O(\rho_X(1/M) + \rho_Y(1/N))$. However, if the starting point for the infinite-dimensional problem satisfies $x_0 \in X_M$ for all M, we can choose $x_0^{MN} = x_0$ for all M (and N). In this case we always have $||x_0 - x_0^{MN}|| \leq \tau(M,N)$. Such situations occur, for example, if $X_M = \operatorname{span}\{\phi_1, \ldots, \phi_M\}$, where ϕ_i are splines and x_0 is a constant function.

The advantage of our approach is that we obtain uniform bounds between the infinite-dimensional iterates x_k and the corresponding finite-dimensional ones x_k^{MN}, whereas using the approach of [2], we would obtain uniform bounds between the restriction of the infinite-dimensional iterates onto the finite-dimensional space, $\Pi_M x_k$, and the iterates x_k^{MN}. In the case of finite element discretizations, with $X = H^\ell$, H^ℓ some Sobolev space, and Π_M the spline interpolant, this would lead to estimates of the form (see, e.g., [4, p. 217])

$$\begin{aligned}
||x_k - x_k^{MN}||_{H^\ell} &\leq ||x_k - \Pi_M x_k||_{H^\ell} + ||\Pi_M x_k - x_k^{MN}||_{H^\ell} \\
&\leq c\frac{1}{M^p}||x_k||_{H^{\ell+p+1}} + c(\rho_X(1/M) + \rho_Y(1/N)).
\end{aligned}$$

This bound involves the $H^{\ell+p+1}$-norm of x_k and therefore only leads to a pointwise estimate since $||x_k||_{H^{\ell+p+1}}$ may not be bounded.

For fine discretizations, subproblem (1.4) is a large-scale problem, and most of the computing time for the determination of a solution of (1.3) is spent solving the subproblems (1.4). In practice, it is therefore often useful to reduce the amount of work by solving the subproblems only partially. This leads to so-called inexact methods. It is usually possible to retain good convergence properties of the inexact method when the accuracy with which the subproblems are solved is sufficiently improved when the iterates approach the solution. The question of how the quality of the computed solutions of the subproblems affect the convergence speed of the method has been analyzed in [10] for Newton's method and in [22] for the Gauss–Newton method for unconstrained problems. If x_{k+1} denotes the exact solution of (1.2), $\tilde{x}_{k+1}$ the computed, inexact solution, and if we know that for the exact method with some $\kappa \in (0,1)$,

$$||x_{k+1} - x_*|| \leq \kappa ||x_k - x_*||,$$

then linear convergence can be retained if the computed solution satisfies

$$||x_{k+1} - \tilde{x}_{k+1}|| \leq \delta_k ||x_k - x_*|| \tag{3.30}$$

with some $\delta_k \leq \delta \in (0, 1-\kappa)$:

$$||\tilde{x}_{k+1} - x_*|| \leq ||x_{k+1} - \tilde{x}_{k+1}|| + ||x_{k+1} - x_*|| \leq (\delta_k + \kappa)||x_k - x_*||.$$

In practice, one has to replace $||x_{k+1} - \tilde{x}_{k+1}||$ and $||x_k - x_*||$ by cheaply computable terms. In case of unconstrained least squares problems with full rank Jacobians we can replace (3.30), for example, by

$$||F'(x_k)^* F'(x_k)\tilde{s}_k + F'(x_k)^* F(x_k)|| \leq \tilde{\delta}_k ||F'(x_k)^* F(x_k)||, \tag{3.31}$$

where $\tilde{\delta}_k < \delta_k$ must be chosen sufficiently small; see [22]. (In the unconstrained case we solve for the step $\tilde{s}_k$ rather than for the new iterate. The computed new iterate is then given by $\tilde{x}_{k+1} = x_k + \tilde{s}_k$. In the next iteration (3.31) is solved for $\tilde{s}_{k+1}$ with x_k replaced by $\tilde{x}_{k+1}$.)

Instead of analyzing local convergence properties of the inexact methods for the discretized problem, we will focus on the question of under which conditions the inexact iterates exhibit a mesh independent behavior. We will show that, instead of forcing the error between the exact solution x_{k+1} and the computed inexact solution $\tilde{x}_{k+1}$ to be less than a small constant times the error between the current iterate and solution, $||x_k - x_*||$ (which is sufficient to guarantee local convergence), we must adjust the quality of computed solutions onto the discretization error in order to obtain a mesh independent behavior. Thus, for mesh independence it is sufficient to impose a quality measure that is fixed and that is not strengthened if the iteration progresses. However, in practice one will enforce stronger criteria on the computed iterates in order to obtain good local convergence behavior.

To distinguish between exact and inexact methods, we denote the inexact iterates by $\tilde{x}_k$, $\tilde{x}_k^{MN}$. Then $x_{k+1} \equiv \tilde{x}_k(\mu_{k+1})$, $x_{k+1}^{MN} \equiv \tilde{x}_k^{MN}(\mu_{k+1}^{MN})$ will denote the exact solution of the subproblems (1.2), (1.4) with x_k, x_k^{MN} replaced by $\tilde{x}_k$, $\tilde{x}_k^{MN}$, respectively.

THEOREM 3.5. *Let the assumptions of Theorem* 3.4 *be valid and let* η, η^{MN} *be constants such that with some* $\vartheta > 0$,

$$\eta, \eta^{MN} \leq \vartheta \left(\rho_X \left(\frac{1}{M} \right) + \rho_Y \left(\frac{1}{N} \right) \right).$$

If the inexact iterates $\tilde{x}_k$, $\tilde{x}_k^{MN}$ are computed such that for all $k \in \mathbb{N}$

$$(3.32) \qquad ||\tilde{x}_k - x_k|| \le \eta, \qquad ||\tilde{x}_k^{MN} - x_k^{MN}|| \le \eta^{MN},$$

then for ϵ given by Theorem 3.4 there exists $c > 0$ (<u>independent</u> of M, N), $\tilde{M}_\epsilon$, $\tilde{N}_\epsilon$, and a function $\tau : \mathbb{N}^2 \rightarrow \mathbb{R}^+$, such that for all $\tilde{x}_0 \in \overline{B_R(0)} \cap B_\epsilon(x_*)$, $M \ge \tilde{M}_\epsilon$, $N \ge \tilde{N}_\epsilon$, and the condition $||\tilde{x}_0 - \tilde{x}_0^{MN}|| \le \tau(M, N)$ implies

$$(3.33) \qquad ||\tilde{x}_k - \tilde{x}_k^{MN}|| \le c\left(\rho_X\left(\frac{1}{M}\right) + \rho_Y\left(\frac{1}{N}\right)\right).$$

Proof. The proof of this theorem closely follows that of Theorem 3.4. We let ϵ, M_ϵ, and N_ϵ be the values given by Theorem 3.4, and let $c_1, \ldots, c_{12}$ be the constants defined in the proof of Theorem 3.4.

Define

$$c_{13} \equiv c_{12} + 2\vartheta \quad \text{and} \quad c_{14} \equiv \frac{8(c_{10} + c_{12})}{3(1-\kappa)} c_8$$

and let $\tilde{M}_\epsilon \ge M_\epsilon$, $\tilde{N}_\epsilon \ge N_\epsilon$ be such that for all $M \ge \tilde{M}_\epsilon, N \ge \tilde{N}_\epsilon$,

$$\rho < \min\left\{\frac{1}{c_{10} + c_{10}c_{14}}, \frac{1-\kappa}{4c_{10}}, \frac{3(1-\kappa)^2}{64c_8(c_{10} + c_{12})^2}\right\}.$$

Moreover, define

$$\tau(M, N) \equiv \Gamma(M, N, c_{10}, c_{13}, \epsilon)$$

(compare (3.20)). We have

$$\begin{aligned}||\tilde{x}_{k+1} - \tilde{x}_{k+1}^{MN}|| \le{}& ||\tilde{x}_k(\mu_{k+1}) - \tilde{x}_k^{MN}(\mu_{k+1})|| + ||\tilde{x}_k^{MN}(\mu_{k+1}) - \tilde{x}_k^{MN}(\mu_{k+1}^{MN})|| \\ &+ ||\tilde{x}_{k+1} - \tilde{x}_k(\mu_{k+1})|| + ||\tilde{x}_{k+1}^{MN} - \tilde{x}_k^{MN}(\mu_{k+1}^{MN})||.\end{aligned}$$

Using the estimates (3.22), (3.29), and (3.32) we obtain that

$$\begin{aligned}&(3.34)\\ &||\tilde{x}_{k+1} - \tilde{x}_{k+1}^{MN}|| \\ &\quad\le \frac{c_6 e_k^2 + 2c_1(\rho + e_k)\tilde{\epsilon} + \kappa||\tilde{x}_k - \tilde{x}_k^{MN}|| + c_6\rho}{1 - c_1\rho - c_1 e_k} + \frac{c_{11}\rho}{1 - c_{10}\rho - c_{10}e_k} + \eta + \eta^{MN} \\ &\quad\le \frac{c_{12}e_k^2 + 2c_{10}(\rho + ||\tilde{x}_k - \tilde{x}_k^{MN}||)\tilde{\epsilon} + \eta + \eta^{MN} + \kappa||\tilde{x}_k - \tilde{x}_k^{MN}|| + c_{12}\rho}{1 - c_{10}\rho - c_{10}e_k} \\ &\quad< \frac{c_{13}e_k^2 + 2c_{10}(\rho + ||\tilde{x}_k - \tilde{x}_k^{MN}||)\tilde{\epsilon} + \kappa||\tilde{x}_k - \tilde{x}_k^{MN}|| + c_{13}\rho}{1 - c_{10}\rho - c_{10}e_k}.\end{aligned}$$

(Recall that $c_{10} = c_1 + 4B\tilde{c}_7 c_9 > c_1$, and $c_{13} > c_6$.)

Using the same arguments as in the proof of Theorem 3.4 and the abbreviation $\tau \equiv \tau(M, N)$, we can conclude from (3.34) that if $M \ge M_{\tilde{\epsilon}}$, $M \ge M_{\tilde{\epsilon}}$, and $||\tilde{x}_k - \tilde{x}_k^{MN}|| \le \tau(M, N)$ then

$$\begin{aligned}||\tilde{x}_{k+1} - \tilde{x}_{k+1}^{MN}|| &\le \frac{c_{14}e_k^2 + 2c_{10}\rho\tilde{\epsilon} + (\kappa - 1)\tau/2 + c_{14}\rho + (1 - c_{10}\rho)\tau}{1 - c_{10}\rho - c_{10}\tau} \\ &= \tau(M, N).\end{aligned}$$

The assertion follows from an induction argument and $\tau(M,N) \leq c_{14}\rho$ (compare the estimate (3.21) in the proof of Theorem 3.4). □

An immediate consequence of this mesh independent behavior is the fact that independent of the mesh size an (almost) constant number of iterations is needed to satisfy an appropriate stopping criterion. Appropriate stopping criteria for the restricted Gauss–Newton method are either

$$||x_k - P(x_k - F'(x_k)^* F(x_k))|| < \text{TOL} \quad \text{or} \quad ||F'(x_k)^* F(x_k) + \mu_k x_k|| < \text{TOL}\,,$$

where TOL is a given bound and P denotes the projection onto the feasible set. In our case,

$$P(y) = \begin{cases} \dfrac{R}{||y||}\, y, & \text{if } ||y|| > R, \\ y, & \text{otherwise.} \end{cases}$$

If the iteration point x_k is an interior point and the gradient is sufficiently small, both criteria reduce to $||F'(x_k)^* F(x_k)|| < \text{TOL}$. We will use the abbreviation

$$t_k \equiv ||x_k - P(x_k - F'(x_k)^* F(x_k))|| \tag{3.35}$$

or

$$t_k \equiv ||F'(x_k)^* F(x_k) + \mu_k x_k||\,, \tag{3.36}$$

depending on which criteria is used. With t_k^{MN} we will denote the corresponding discretized values. We use the same notation for both terms, since we have the same type of estimates for $|t_k - t_k^{MN}|$ regardless of whether (3.35) or (3.36) is used. $k(\text{TOL})$ and $k^{MN}(\text{TOL})$ will be defined as the smallest iteration counts for which the termination criteria is satisfied, i.e.,

$$\begin{aligned} k(\text{TOL}) &\equiv \min\{k \mid t_k < \text{TOL}\}, \\ k^{MN}(\text{TOL}) &\equiv \min\{k \mid t_k^{MN} < \text{TOL}\}. \end{aligned}$$

Now, the uniform estimate derived in Theorem 3.4, yields the following result.

COROLLARY 3.6. *Let the assumptions of Theorem 3.4 hold. Moreover, let x_0 and x_0^{MN} be given such that $x_0 \in B_{\epsilon_1}(x_*)$ and $||x_0 - x_0^{MN}|| \leq \tau(M,N)$, for ϵ_1 and $\tau(M,N)$ defined as in Theorem 3.4. Then for every* TOL > 0 *and $\delta > 0$, there exist M_1, N_1 such that*

$$k(\text{TOL} + \delta) \leq k^{MN}(\text{TOL}) \leq k(\text{TOL}) \quad \forall M \geq M_1, \quad N \geq N_1\,.$$

If $t_{k(\text{TOL})-1} >$ TOL then

$$k^{MN}(\text{TOL}) = k(\text{TOL}) \quad \forall M \geq M_1, \quad N \geq N_1\,.$$

Proof. In the proof of Theorem 3.4 it was shown under the assumptions listed above that

$$||x_k - x_k^{MN}|| \leq c(\rho_X(1/M) + \rho_Y(1/N)), \qquad |\mu_k - \mu_k^{MN}| \leq c(\rho_X(1/M) + \rho_Y(1/N))$$

for all k and $M \geq M_\epsilon, N \geq N_\epsilon$. This yields the existence of $\tilde{c}$, independent of M, N such that

$$|t_k - t_k^{MN}| \leq \tilde{c}(\rho_X(1/M) + \rho_Y(1/N)) \quad \forall M \geq M_\epsilon, \quad N \geq N_\epsilon\,.$$

This estimate can be derived using the Lipschitz continuity of F and F' and the estimates in Lemma 3.1. If t_k is defined through (3.36) one also has to incorporate the fact that the Lagrange multipliers are uniformly bounded, and if t_k is defined through (3.35) one has to incorporate the contraction property of projections, i.e., $||Px - Py|| \le ||x - y||$.

If we choose M_1, N_1 such that

$$|t_{k(\mathrm{TOL})} - t^{MN}_{k(\mathrm{TOL})}| < \mathrm{TOL} - t_{k(\mathrm{TOL})} \quad \forall M \ge M_1, \quad N \ge N_1$$

and

$$|t_k - t_k^{MN}| < \delta \quad \forall k, \quad M \ge M_1, \quad N \ge N_1,$$

then we obtain for all $M \ge M_1, N \ge N_1$ that

$$t^{MN}_{k(\mathrm{TOL})} \le t_{k(\mathrm{TOL})} + |t_{k(\mathrm{TOL})} - t^{MN}_{k(\mathrm{TOL})}| < \mathrm{TOL}$$

and

$$t_{k^{MN}(\mathrm{TOL})} \le t^{MN}_{k^{MN}(\mathrm{TOL})} + |t_{k^{MN}(\mathrm{TOL})} - t^{MN}_{k^{MN}(\mathrm{TOL})}| < \mathrm{TOL} + \delta\,.$$

If $t_{k(\mathrm{TOL})-1} > \mathrm{TOL}$ we can choose $\delta = (t_{k(\mathrm{TOL})-1} - \mathrm{TOL})/2$. This yields $k(\mathrm{TOL}+\delta) = k(\mathrm{TOL})$. Hence the assumption is proven. □

We conclude this section with results on the convergence rate of the Gauss–Newton method for the discretized problem and on perturbations of solutions and Lagrange multipliers. In addition to the assumptions (A1)–(A7) we need an assumption on the curvature of F and F_N:

(A8) There exists a sequence $\{\xi_{MN}\}$ with $\lim_{M,N\to\infty} \xi_{MN} = 0$, such that for all $x, y \in \overline{B_R(0)} \cap X_M$

$$||((F_N'(x)^* - F_N'(y)^*) - (F'(x)^* - F'(y)^*))F_N(y)|| \le \xi_{MN}||x - y||.$$

If F and F_N are twice Fréchet differentiable, a sufficient condition for (A8) to hold is that

$$||F_N''(y)^* F_N(x) - F''(y)^* F_N(x)||_{L(X,X)} \le c\left(\rho_X\left(\frac{1}{M}\right) + \rho_Y\left(\frac{1}{N}\right)\right) \quad \forall x, y \in \overline{B_R(0)}\,.$$

Since $F_N''(y)^*$ is applied to an element of Y_N, it is the ordinary Y_N, X_M adjoint, $F_N''(y)^* \in L(Y_N, X_M \otimes X_M)$. In this case we obtain $\xi_{MN} = O(\rho_X(1/M) + \rho_Y(1/N))$.

In the following theorem we will use the notation of Theorem 2.2 and its proof.

THEOREM 3.7. *Let* (A1)–(A8) *and the assumptions of Theorems* 2.2 *and* 3.4 *hold. Then for all* $\alpha \in (1, (\gamma_* + \mu_*)/\sigma)$ *and all* $\epsilon \in (0, \epsilon_*(\alpha))$ *there exist* M_ϵ *and* N_ϵ *such that for all* $M \ge M_\epsilon$, $N \ge N_\epsilon$, *and* $x_0^{MN} \in \overline{B_R(0)} \cap B_\epsilon(x_*)$, *the Gauss–Newton method for the discretized problem with starting point* x_0^{MN} *converges to a solution* x_*^{MN} *of* (1.3). *Moreover,* x_*^{MN} *is the unique minimizer of* (1.3) *in* $B_\epsilon(x_*)$, *and the convergence rate is given by*

$$\begin{aligned} ||x_{k+1}^{MN} - x_*^{MN}|| &\le \frac{\alpha\sigma^{MN}}{\gamma_*^{MN} + \mu_*^{MN}}||x_k^{MN} - x_*^{MN}|| + \frac{\alpha L_1^{MN}\Lambda^{MN}}{2(\gamma_*^{MN} + \mu_*^{MN})}||x_k^{MN} - x_*^{MN}||^2 \\ &< ||x_k^{MN} - x_*^{MN}||, \end{aligned}$$

where $\Lambda^{MN} \equiv \sup_{x\in\overline{B_R(0)}} ||F_N'(x)||$ *and* $\{\gamma_*^{MN}\}_{\mathbb{N}}$, $\{\sigma^{MN}\}_{\mathbb{N}}$ *are sequences with*

$$|\gamma_*^{MN} - \gamma_*| = O\left(\rho_X\left(\frac{1}{M}\right) + \rho_Y\left(\frac{1}{N}\right)\right),$$
$$|\sigma^{MN} - \sigma| = O\left(\xi_{MN} + \rho_X\left(\frac{1}{M}\right) + \rho_Y\left(\frac{1}{N}\right)\right).$$

The errors between x_* *and* x_*^{MN} *and between the Lagrange multipliers can be estimated by*

$$\begin{aligned} ||x_* - x_*^{MN}|| &\le c\left(\rho_X\left(\frac{1}{M}\right) + \rho_Y\left(\frac{1}{N}\right)\right), \\ |\mu_* - \mu_*^{MN}| &\le c\left(\rho_X\left(\frac{1}{M}\right) + \rho_Y\left(\frac{1}{N}\right)\right), \end{aligned} \tag{3.37}$$

where $c > 0$ *denotes a generic constant.*

Proof. We only give a sketch of the proof. Theorem 2.6 yields the existence of a sequence $\{x_*^{MN}\}_{\mathbb{N}}$ of minimizers of (1.3) such that $x_*^{MN} \to x_*$ $(M, N \to \infty)$.

From (2.21), (A5), and (A8), we obtain that for all $x, y \in \overline{B_R(0)}$

$$\begin{aligned} &||(F_N'(x)^* - F_N'(y)^*)F_N(y)|| \\ &\quad \le ||(F'(x)^* - F'(y)^*)F(y)|| + ||(F'(x)^* - F'(y)^*)(F(y) - F_N(y))|| \\ &\qquad + ||((F_N'(x)^* - F_N'(y)^*) - (F'(x)^* - F'(y)^*))F_N(y)|| \\ &\quad \le \sigma||x - y|| + \rho_Y(1/N)L_1||x - y|| + \xi_{MN}\,||x - y||. \end{aligned}$$

Hence there exist $\sigma^{MN} \ge \sigma$ such that $|\sigma^{MN} - \sigma| = O(\xi_{MN} + \rho_X(1/M) + \rho_Y(1/N))$ and

$$||(F_N'(x)^* - F_N'(y)^*)F_N(y)|| \le \sigma^{MN}||x - y||.$$

Assumption (A5) and (2.20) and (3.38) yield

$$\begin{aligned} ||F_N'(x_*^{MN})h||^2 &\ge (||F'(x_*)h|| - ||F'(x_*^{MN})h - F'(x_*)h|| \\ &\qquad - ||F_N'(x_*^{MN})h - F'(x_*^{MN})h||)^2 \\ &\ge \left(\sqrt{\gamma_*} - L_1||x_* - x_*^{MN}|| - \rho_Y\left(\frac{1}{N}\right)\right)^2 ||h||^2. \end{aligned}$$

Hence, there exists a sequence $\{\gamma_*^{MN}\}_{\mathbb{N}}$, $\gamma_*^{MN} = \gamma_* + O(||x_* - x_*^{MN}|| + \rho_Y(1/N))$ such that

$$||F_N'(x_*^{MN})h_M||^2 \ge \gamma_*^{MN}||h_M||^2 \quad \forall h_M \in X_M.$$

If we denote the Lagrange multiplier corresponding to x_*^{MN} by μ_*^{MN}, one can show, as in the proofs of Lemmas 3.2 and 3.3 (note that $x_*(\mu_*) = x_*$ and use (3.37)) that for sufficiently large M, N there exists c independent of M, N such that

$$|\mu_* - \mu_*^{MN}| \le c(\rho_X(1/M) + \rho_Y(1/N)).$$

These preliminaries show that we can choose $\bar{M}, \bar{N}$ such that

$$\alpha \in \left(1, \frac{\gamma_*^{MN} + \mu_*^{MN}}{\sigma^{MN}}\right) \quad \forall M \ge \bar{M}, \quad N \ge \bar{N}.$$

If we apply Theorem 2.2 to x_*^{MN}, we obtain the existence of ϵ_*^{MN} such that the Gauss–Newton method for the discretized problem with arbitrary starting point $x_0^{MN} \in \overline{B_R(0)} \cap B_{\epsilon_*^{MN}}(x_*^{MN})$ converges to x_*^{MN}:

$$\begin{aligned}||x_{k+1}^{MN} - x_*^{MN}|| &\le \frac{\alpha\sigma^{MN}}{\gamma_*^{MN} + \mu_*^{MN}}||x_k^{MN} - x_*^{MN}|| + \frac{\alpha L_1^{MN}\Lambda^{MN}}{2(\gamma_*^{MN} + \mu_*^{MN})}||x_k^{MN} - x_*^{MN}||^2 \\ &< ||x_k^{MN} - x_*^{MN}||.\end{aligned}$$

Moreover, the proof of Theorem 2.2 shows that $\epsilon_*^{MN} \longrightarrow \epsilon_*$.

The uniqueness of the solution x_*^{MN} follows from the fact that the Gauss–Newton method with arbitrary starting point $x_0 \in B_\epsilon(x_*)$ converges towards x_*^{MN}.

Theorem 3.4 yields the error estimate

$$(3.38) \qquad ||x_* - x_*^{MN}|| \le 2c\left(\rho_X\left(\frac{1}{M}\right) + \rho_Y\left(\frac{1}{N}\right)\right),$$

since (3.17) holds for all k. □

4. Examples. In this section we will demonstrate how the analysis of the previous sections can be applied to a parameter identification problem. Although we are considering the one-dimensional problem, it should be mentioned that our analysis can be extended to the multidimensional case. The parameter identification problem for the two-point boundary value problem can be stated as follows.

For a given observation $z \in L^2(0,1)$ or $H_0^1(0,1)$ find $q \in H^1(0,1)$ with $||q||_{H^1} \le R$ and $q(x) \ge \gamma > 0$ almost everywhere on $(0,1)$, such that

$$u(q) \approx z.$$

Here $u(q) \in H_0^1(0,1)$ is defined to be the weak solution of the state equation

$$-(qu')' = f \quad \text{in } (0,1), \quad u(0) = u(1) = 0$$

with $f \in L^2(0,1)$, i.e., $u(q)$ is defined through

$$(4.1) \qquad \langle qu', v'\rangle = \langle f, v\rangle \quad \forall v \in H_0^1(0,1).$$

(For the rest of the section we will drop the notation of the space $(0,1)$ and we will always use the notation $\langle\cdot,\cdot\rangle$ for the L^2-scalar product.) It is well known that (4.1) always possesses a solution $u(q) \in H_0^1$ and since $q \in H^1$, $f \in L^2$ one can even show that $u(q) \in H_0^1 \cap H^2$ with

$$(4.2) \qquad ||u(q)||_{H^2} \le c||f||_{L^2},$$

where c is a constant depending on γ and R (see, e.g., [5, p. 223]). In what follows, we will denote by $u(q)$ the solution of (4.1).

In the following we use the output least squares formulation to solve the parameter identification problem; i.e., we determine q such that $u(q)$ is close to z in the norm of the observation space $\mathcal{Z}$, and we use $\mathcal{Z} = L^2(0,1)$ or $H_0^1(0,1)$. It is well known that this problem may be ill posed in the sense that small perturbations in the observation z may lead to large errors in the solution q. In order to get a stable problem for which it is possible to estimate the error between the computed solution of the output least squares problem with perturbed data z and the true, but unknown solution corresponding to the unperturbed data, one has to modify the problem. A possible approach to removing this difficulty is the Tikhonov regularization. Here one adds a regularization term to the objective, so that we have to solve

$$\begin{array}{ll} \min & ||u(q)-z||^2_{\mathcal{Z}} + \alpha ||q||^2_{H^1} \\ \text{s.t.} & ||q||_{H^1} \le R, \quad q(x) \ge \gamma \quad \text{a.e. on } (0,1). \end{array} \tag{4.3}$$

Another strongly related regularization technique might consist of reducing the size of R. The Tikhonov regularization for nonlinear problems is studied by many authors (see, e.g., [5], [7], [8], [15], and [20]). In the following we assume that q_* is a solution of (4.3), which satisfies $q_*(x) > \gamma$ almost everywhere on $(0,1)$. Since $||\cdot||_{H^1}$ dominates the infinity norm and since we are concerned with a local analysis, we may drop the pointwise constraint on q. In the sequel it will always implicitly be assumed that the considered parameter functions q $(,q_1,q_2,\ldots)$ satisfy this constraint. In this case (4.3) fits into our framework if we set

$$X = H^1, \quad Y = \mathcal{Z} \times H^1 \quad \text{(endowed with the product topology)}$$

and

$$F(q) = \left(\begin{array}{c} u(q) - z \\ \sqrt{\alpha} q \end{array} \right) .$$

(In this section we follow the conventional notation in parameter identification and denote the sought variable by q, whereas $x \in (0,1)$ denotes the space variable!) It can be shown that F is infinitely often Fréchet differentiable. The first Fréchet derivative is given by

$$F'(q)h = \left(\begin{array}{c} \eta \\ \sqrt{\alpha} h \end{array} \right),$$

where $\eta = u_q(q)(h)$ is the solution of

$$\langle q\eta', v'\rangle = -\langle hu', v'\rangle \quad \forall v \in H^1_0 . \tag{4.4}$$

The variational equation

$$\langle q\xi', v'\rangle = -\langle h_1\eta_2', v'\rangle - \langle h_2\eta_1', v'\rangle \quad \forall v \in H^1_0 , \tag{4.5}$$

where η_i is the solution of (4.4) with h_i instead of h, characterizes the second Fréchet derivative of F, which is given as

$$F''(q)(h_1,h_2) = \left(\begin{array}{c} \xi \\ 0 \end{array} \right) .$$

From the structure of $F'(q)$ it can be seen that for arbitrary q and h

$$||F'(q)h||^2 \ge \alpha ||h||^2 .$$

This inequality shows that the Tikhonov regularization shifts the spectrum of the first Fréchet derivative of F. On the other hand, the regularization causes an increase of the residual $||u(q) - z||$ and therefore an increase of the weight of the second-order term $\langle F''(q)(h,h), F(q)\rangle$ in the Hessian of $||F(q)||^2$. In [8], Colonius and Kunisch show that for small α the effect of shifting the spectrum is stronger than the increase of the residual. They show that if the residual of the unregularized problem ($\alpha = 0$) is sufficiently small there exist parameters α such that the inequality (2.30) holds (with $\mu_* = 0$). Although this only guarantees the validity of the conditions for convergence of the Gauss–Newton method and for its mesh independence in the case where regularization is only performed by using Tikhonov regularization, we also obtained extremely good results in the cases where the problem is regularized by the norm constraint.

For the numerical solution of (4.3) we have to discretize the problem. We choose piecewise linear splines. Let φ_i^M, ψ_j^N be the hat functions with $\varphi_i^M(i/M) = 1$, $\psi_j^N(j/N) = 1$, and $\varphi_i^M(x) = 0$ for $x \notin (\frac{i-1}{M}, \frac{i+1}{M})$, and $\psi_j^N(x) = 0$ for $x \notin (\frac{j-1}{N}, \frac{j+1}{N})$.

We set $X_M := \operatorname{span}\{\varphi_0^M, \dots, \varphi_M^M\}$, $V_N := \operatorname{span}\{\psi_1^N, \dots, \psi_{N-1}^N\}$, and $Y_N := V_N \times X$.

The discretized solution of the state equation is given as the uniquely determined element $u^N = u^N(q)$, which satisfies

$$\langle q u^{N'}, v^{N'} \rangle = \langle f, v^N \rangle \quad \forall v^N \in V_N . \tag{4.6}$$

Now we choose the discretization of F as follows:

$$F_N(q) = \begin{pmatrix} u^N(q) - z^N \\ \sqrt{\alpha} q \end{pmatrix},$$

where z^N is a discretization of z, for example, the spline interpolant.

From (4.6) it can be seen that although in the computations, F_N has to be evaluated only at points $q = q^M \in X_M$, the functions are defined on the whole infinite-dimensional space X. The same is true for the Fréchet derivative and its adjoint, since they are defined through variational equalities similar to (4.6).

The Fréchet derivative of $u^N(q)$, $\eta^N := u_q^N(q)(h)$, is given as the unique solution of

$$\langle q \eta^{N'}, v^{N'} \rangle = -\langle h u^{N'}, v^{N'} \rangle \quad \forall v^N \in V_N . \tag{4.7}$$

The second Fréchet derivative is given analogously to (4.5). This especially proves the validity of (A1) and (A3).

In the following we will denote by $u(q)$ the solution of (4.1) and by $u^N(q)$ its discretization, i.e., the solution of (4.6), for a given parameter function q. And we will use a similar notation for the Fréchet derivatives.

We will now verify that F and its discretization satisfy the assumptions (A2) and (A4). In the following we will use c as a generic constant to reduce the notational complexity.

Since $u(q_1) - u(q_2)$ satisfies the variational equation

$$\langle q_1 (u(q_1) - u(q_2))', v' \rangle = \langle (q_2 - q_1) u(q_2)', v' \rangle \quad \forall v \in H_0^1$$

we obtain with (4.2) and

$$||v||_{L^\infty} \leq c ||v||_{H^1} \tag{4.8}$$

that

$$||u(q_1) - u(q_2)||_{H^1} \leq c ||(q_1 - q_2) u(q_2)'||_{L^2} \leq c ||f||_{L^2} ||q_1 - q_2||_{H^1} . \tag{4.9}$$

From the error analysis of finite element methods we get (see, e.g., [4, pp. 152, 217])

$$\|u(q) - u^N(q)\|_{H^1} \le c\|u(q)\|_{H^2}\frac{1}{N} \le c\|f\|_{L^2}\frac{1}{N}. \tag{4.10}$$

Using the Aubin–Nitsche trick (see, e.g., [4, p. 229]), this estimate can be improved for the L^2-norm to

$$\|u(q) - u^N(q)\|_{L^2} \le \frac{c}{N^2}\,.$$

The Fréchet derivatives $u_q(q)$ and $u_q^N(q)$ are defined through the same kind of elliptic differential equation. Therefore, we can apply a similar analysis to derive continuity results for the derivatives. If we use the corresponding estimates to (4.2), (4.9), and inequality (4.8), we obtain

$$\begin{aligned}&\|u_q(q_1)(h) - u_q(q_2)(h)\|_{H^1}\\ &\le c(\|q_1 - q_2\|_{H^1}\|u_q(q_2)(h)\|_{H^1} + \|h\|_{H^1}\|u(q_1) - u(q_2)\|_{H^1})\\ &\le c\|f\|_{L^2}\|q_1 - q_2\|_{H^1}\|h\|_{H^1}.\end{aligned}$$

Let $\zeta \in H_0^1$ denote the solution of

$$\langle q\zeta', v'\rangle = \langle hu^{N'}, v'\rangle \quad \forall v \in H_0^1$$

and let η, η^N be the solutions of (4.4) and (4.7), respectively. Then the error between the discretized and infinite-dimensional Fréchet derivative can be estimated through

$$\begin{aligned}\|\eta - \eta^N\|_{H^1} &\le \|\eta - \zeta\|_{H^1} + \|\zeta - \eta^N\|_{H^1}\\ &\le c\|h\|_{H^1}\|u(q) - u^N(q)\|_{H^1} + c\|h\|_{H^1}\|u^N(q)\|_{H^1}\,\frac{1}{N}\\ &\le c\|f\|_{L^2}\|h\|_{H^1}\,\frac{1}{N}\,.\end{aligned}$$

The above techniques can obviously be applied to the second and even higher Fréchet derivatives. The calculations above show that F satisfies the assumptions (A2) and (A4) for $\mathcal{Z} = H_0^1$, or $\mathcal{Z} = L^2$ with $\rho_Y(1/N) = c_Y/N$, provided that $\|z-z^N\|_{H_0^1} \le c/N$, or $\|z - z^N\|_{L^2} \le c/N$.

Now, we will investigate the computation of the adjoint of F'. From the structure of F' it is obvious that it is sufficient to study the calculation of $(u_q(q))^*$. The adjoint of $u_q(q)$ applied to $g \in L^2$ can be computed in two steps:

(1) Solve the adjoint equation for given q and g:

$$\langle qw', v'\rangle = \langle g, v\rangle_{\mathcal{Z}} \quad \forall v \in H_0^1. \tag{4.11}$$

(2) Move from the L^2 to the H^1 topology

$$\langle p, \varphi\rangle + \langle p', \varphi'\rangle = -\langle u(q)'w', \varphi\rangle \quad \forall \varphi \in H^1. \tag{4.12}$$

(In our example, the adjoint equation is just the state equation, since the differential operator $D_x q(D_x\cdot)$ is formally self-adjoint.) If we solve the two equations, we obtain $p = (u_q(q))^*(g)$, which can be seen if we set $v = u_q(q)(\varphi)$ in (4.11):

$$\begin{aligned}\langle g, u_q(q)(\varphi)\rangle_{\mathcal{Z}} &= \langle qw', (u_q(q)(\varphi))'\rangle\\ &= \langle w', q\,(u_q(q)(\varphi))'\rangle\\ &= -\langle \varphi u(q)', w'\rangle = \langle p, \varphi\rangle_{H^1}\end{aligned}$$

(for the third equality we used the definition of the Fréchet derivative; see (4.4) with v replaced by w). The variational equation (4.12) is the weak formulation of the elliptic problem

$$-p'' + p = -u(q)'w' \quad \text{in } (0,1)$$

with homogeneous Neumann boundary conditions. Equation (4.12) yields

$$-\langle p', \varphi' \rangle = \langle p, \varphi \rangle + \langle u(q)'w', \varphi \rangle \quad \forall \varphi \in C_0^\infty ,$$

which shows that p'' exists and equals $p + u(q)'w'$. In particular, we obtain $p'' \in L^2$ and

$$||p''||_{L^2} \le ||p||_{L^2} + c||u(q)||_{H^2}\,||w||_{H^1} .$$

The Lax–Milgram theorem and (4.8) yield

$$||p||_{L^2} \le ||p||_{H^1} \le ||u(q)'w'||_{L^2} \le ||u(q)'||_{L^\infty}\,||w||_{H^1} \le c||u(q)||_{H^2}\,||w||_{H^1} .$$

Hence we obtain that the weak solution of the Neumann problem obeys the regularity property $p \in H^2$ and

$$||p||_{H^2} \le c||u(q)||_{H^2}\,||w||_{H^1} \le c||f||_{L^2}||g||_{L^2} . \tag{4.13}$$

This bound together with the techniques already applied to prove (A2) and (A4) can now be used to derive an estimate of type (A5). If we discretize the Neumann problem (4.12) and solve

$$\langle \hat{p}^M, \varphi^M \rangle + \langle \hat{p}^{M'}, \varphi^{M'} \rangle = \langle u(q)'w', \varphi^M \rangle \quad \forall \varphi^M \in X_M, \tag{4.14}$$

the error between the solutions of (4.12) and (4.14) can be estimated by

$$||p - \hat{p}^M||_{H^1} \le c||f||_{L^2}||g||_{L^2}\frac{1}{M} \tag{4.15}$$

(see (4.13) and [4, pp. 152, 217]). The adjoint of the discretized Fréchet derivative $u_q^N(q)$ is given through:

(1) Solve the adjoint equation

$$\langle qw^{N'}, v^{N'} \rangle_{\mathcal{Z}} = \langle g, v^N \rangle \quad \forall v^N \in V_N. \tag{4.16}$$

(2) Move from the L^2 to the H^1 topology

$$\langle p^M, \varphi^M \rangle + \langle p^{M'}, \varphi^{M'} \rangle = -\langle u^N(q)' w^{N'}, \varphi^M \rangle \quad \forall \varphi^M \in X_M. \tag{4.17}$$

At the end we obtain $p^M = (u_q^N(q))^*(g)$. The error between the infinite-dimensional and discretized adjoints can be estimated by (see (4.2), (4.8), (4.10), and (4.15))

$$\begin{aligned}
||p - p^M||_{H^1} &\le ||p - \hat{p}^M||_{H^1} + ||\hat{p}^M - p^M||_{H^1} \\
&\le c||f||_{L^2}||g||_{L^2}\frac{1}{M} + \sup_{||\varphi||_{H^1}=1} \langle u^N(q)' w^{N'} - u(q)'w', \varphi \rangle \\
&\le c||f||_{L^2}||g||_{L^2}\frac{1}{M} + \sup_{||\varphi||_{H^1}=1} \langle w^{N'} - w', u(q)'\varphi \rangle \\
&\quad + \langle u^N(q)' - u(q)', w^{N'} \varphi \rangle \\
&\le c||f||_{L^2}||g||_{L^2}\frac{1}{M} \\
&\quad + c||w^N - w||_{H^1}||u(q)||_{H^1} + c||u^N(q) - u(q)||_{H^1}||w^{N'}||_{H^1} \\
&\le c||f||_{L^2}||g||_{L^2}\frac{1}{M} + c||f||_{L^2}||g||_{L^2}\frac{1}{N} .
\end{aligned}$$

TABLE 1

Number of iterations $q_0 \equiv 0.2, \quad z = u(q_*)$												
	Example 1											
	TOL $= 10^{-8}$						TOL $= 10^{-6}$					
M	6	12	24	48	96	192	6	12	24	48	96	192
α N	12	24	48	96	192	384	12	24	48	96	192	384
0	7	7	7	7	7	7	7	7	7	7	7	7
10^{-8}	7	7	7	7	7	7	7	7	7	7	7	7
10^{-6}	7	7	7	7	7	7	7	7	7	7	7	7
10^{-4}	8	8	8	8	8	8	7	7	7	7	7	7
10^{-2}	10	10	10	10	10	10	8	8	8	8	8	8
	Example 2											
0	11	15*	8	6	7	7	10	15*	8	7	7	7
10^{-8}	8	7	7	7	7	7	7	7	7	7	7	7
10^{-6}	7	7	7	7	7	7	6	6	6	6	6	6
10^{-4}	9	9	9	9	9	9	7	7	7	7	7	7
10^{-2}	9	9	9	9	9	9	7	7	7	7	7	7
	Example 3											
0	8	8	8	8	8	8	7	7	7	7	7	7
10^{-8}	8	8	8	8	8	8	7	7	7	7	7	7
10^{-6}	8	8	8	8	8	8	7	7	7	7	7	7
10^{-4}	8	8	8	8	8	8	7	7	7	7	7	7
10^{-2}	10	10	10	10	10	10	8	8	8	8	8	8

The last inequality proves that (A7) is also valid with $\rho_X(1/M) = c_X/M$.

Since (A6) is a standard result in finite element error analysis, (A1)–(A7) are valid for this example.

We ran several test examples from the set of test problems in [21]. The test functions for the results we present below are given in the following examples.

Example 1.

$$u(q_*) = \sin(\pi x), \quad q_* = \tfrac{1}{2} + \cos(x), \quad ||q_*||^2_{H^1} = \tfrac{5}{4} + \sin(1).$$

Example 2.

$$u(q_*) = \begin{cases} -9x^2 + 6x, & x \in [0, \frac{1}{3}], \\ 1, & x \in (\frac{1}{3}, \frac{2}{3}], \\ -9x^2 + 12x - 3, & x \in (\frac{2}{3}, 1]. \end{cases}$$

$$q_* = \frac{1}{2} + \sin(\pi x), \quad ||q_*||^2_{H^1} = \frac{3}{4} + \frac{2}{\pi} + \frac{\pi^2}{2}.$$

Example 3.

$$u(q_*) = \sin(\pi x), \quad q_* = 1 + x, \quad ||q_*||^2_{H^1} = \tfrac{10}{3}.$$

The Gauss–Newton method is implemented using the Hebden–Reinsch method for the computation of μ^{MN}_{k+1} as the inner iteration. In all test runs we chose z^N to be the spline interpolant of z. The iterations were terminated if $t^{MN}_k \leq$ TOL or $k > 15$. For all test runs we took $q_0 \equiv 0.2$ and incorporated either the Tikhonov

TABLE 2

Number of iterations $q_0 \equiv 0.2, \quad z = u(q_*)$						
TOL = 10^{-6}						
	Example 1					
M	6	12	24	48	96	192
R N	12	24	48	96	192	384
1.3	7(7)	7(7)	7(7)	7(7)	7(7)	7(7)
1.0	8(8)	8(8)	8(8)	8(8)	8(8)	8(8)
0.8	6(7)	6(7)	6(7)	6(7)	6(7)	6(7)
	Example 2					
2.5	6(6)	7(7)	7(7)	7(7)	7(7)	7(7)
2.0	6(6)	6(6)	6(6)	6(6)	6(6)	6(6)
1.5	6(6)	6(6)	6(6)	6(6)	6(6)	6(6)
1.2	7(7)	6(6)	6(6)	6(6)	6(6)	6(6)
	Example 3					
1.8	7(7)	7(7)	7(7)	7(7)	7(7)	7(7)
1.3	8(8)	8(8)	8(8)	8(8)	8(8)	8(8)
1.0	8(8)	8(8)	8(8)	8(8)	8(8)	8(8)
0.8	7(7)	7(7)	7(7)	7(7)	7(7)	7(7)
TOL = 10^{-8}						
	Example 1					
1.3	8(8)	8(8)	8(8)	8(8)	8(8)	8(8)
1.0	10(10)	10(10)	10(10)	10(10)	10(10)	10(10)
0.8	8(9)	8(9)	8(9)	8(9)	8(9)	8(9)
	Example 2					
2.5	7(7)	7(7)	7(7)	7(7)	7(7)	7(7)
2.0	7(7)	7(7)	7(7)	7(7)	7(7)	7(7)
1.5	8(8)	8(8)	8(8)	8(8)	8(8)	7(7)
1.2	8(8)	8(8)	8(8)	8(8)	7(7)	7(7)
	Example 3					
1.8	8(8)	8(8)	8(8)	8(8)	8(8)	8(8)
1.3	10(10)	10(10)	10(10)	10(10)	10(10)	10(10)
1.0	10(10)	10(10)	10(10)	10(10)	10(10)	10(10)
0.8	9(9)	9(9)	9(9)	9(9)	9(9)	9(9)

regularization or the regularization by norm constraint. All computations were done on a SUN Sparcstation1 in double precision FORTRAN.

Tables 1 and 2 show the results in case of unperturbed observations for Tikhonov regularization and regularization by constraints, respectively. For small regularization parameter α the discretized problems have almost zero residual at the solution and the Gauss–Newton method converges almost quadratically. Therefore, there is no difference in the number of iterations for small α, except for Example 2, where regularization is needed to observe mesh independence.

In Table 2 the first numbers of each column show the number of iterations needed for the termination criteria $||q_k^{MN} - P(q_k^{MN} - F_N'(q_k^{MN})^* F_N(q_k^{MN}))|| < \text{TOL}$; the numbers in parentheses show the number of iterations needed to satisfy the termination criteria $||F_N'(q_k^{MN})^* F_N(q_k^{MN}) + \mu_k^{MN} q_k^{MN}|| < \text{TOL}$.

TABLE 3

Number of Iterations

$q_0 \equiv 0.2, \quad z = u(q_*) + 0.05 \sin(10\pi x - 0.5\pi)$

	Example 1											
	TOL $= 10^{-8}$						TOL $= 10^{-6}$					
M	6	12	24	48	96	192	6	12	24	48	96	192
α N	12	24	48	96	192	384	12	24	48	96	192	384
0	15*	8	10	15*	15*	15*	11	9	8	15*	15*	15*
10^{-6}	10	10	11	11	11	11	7	7	8	8	8	8
10^{-4}	8	8	8	8	8	8	7	7	7	7	7	7
10^{-2}	10	10	10	10	10	10	8	8	8	8	8	8
10^{-1}	15*	15*	15*	15*	15*	15*	12	12	12	12	12	12
	Example 2											
0	12	15*	15*	15*	15*	15*	10	15*	13	15*	15*	15*
10^{-6}	8	10	15*	15*	10	10	7	7	15*	15*	7	7
10^{-4}	8	8	8	8	7	9	7	7	7	7	7	7
10^{-2}	9	9	9	9	9	9	7	7	7	7	7	7
10^{-1}	11	11	11	11	11	11	9	9	9	9	9	9
	Example 3											
0	15*	9	10	15*	15*	15*	11	8	9	15*	15*	15*
10^{-6}	11	11	11	13	13	14	7	8	8	9	9	9
10^{-4}	8	8	8	8	8	8	7	7	7	7	7	7
10^{-2}	10	10	10	10	10	10	8	8	8	8	8	8
10^{-1}	15*	15*	15*	15*	15*	15*	13	13	13	13	13	13

The notation 15* in the tables means that the iteration was terminated because the maximum number of iterations, 15, was exceeded.

In the norm constraint case, we obtain similar results, except for Example 2. Here we recognize unstable behavior for $R = 1.5, 1.2$ and TOL $= 10^{-8}$. This might be due to the fact that the Lagrange multipliers are computed approximately. If the constraint is active, we stop the inner iteration for the computation of μ_k^{MN} if

$$| \, \|q_{k-1}^{MN}(\mu_k^{MN})\|_{H^1} - R| / R < 10^{-4} .$$

Therefore, the projection is computed in the following way:

$$P(\xi_k^{MN}) = \begin{cases} \|q_k^{MN}\|_{H^1} / \|\xi_k^{MN}\|_{H^1} & \text{if } \dfrac{| \, \|q_k^{MN}\|_{H^1} - R|}{R} < 10^{-4}, \\ R / \|\xi_k^{MN}\|_{H^1} & \text{otherwise,} \end{cases}$$

where $\xi_k^{MN} = q_k^{MN} - F_N'(q_k^{MN})^* F_N(q_k^{MN})$.

Tables 3 and 4 show the results for perturbed observations. In the case of Tikhonov regularization mesh independence can be observed only for sufficiently large regularization parameter α. This behavior is theoretically justified through the analysis presented in §§2 and 3. Our results indicate that $\kappa < 1$ for small but sufficiently large α. If α is further increased, the residual and therefore the second-order part in the Hessian, which is neglected in the Gauss–Newton method, increases. For regularization parameters $\alpha \geq 1$ the method did not converge (a result that is not reported in our tables). For Examples 1 and 3, $\alpha = 0.1$, the criteria $k > 15$ is satisfied before the gradient reaches TOL, although the method converges.

TABLE 4

Number of Iterations						
$q_0 \equiv 0.2, \quad z = u(q_*) + 0.5\sin(10\pi x - 0.5\pi)$						
TOL $= 10^{-6}$						
	Example 1					
M	6	12	24	48	96	192
R N	12	24	48	96	192	384
1.3	7(7)	7(7)	7(7)	7(7)	7(7)	7(7)
1.0	7(7)	7(7)	7(7)	7(7)	7(7)	7(7)
0.8	6(6)	6(6)	6(6)	6(6)	6(6)	6(6)
	Example 2					
2.5	8(8)	7(7)	8(8)	8(8)	8(8)	8(8)
2.0	8(8)	7(7)	7(7)	7(7)	7(7)	7(7)
1.5	8(8)	8(8)	7(7)	7(7)	7(7)	7(7)
1.2	8(8)	7(7)	7(7)	7(7)	7(7)	7(7)
	Example 3					
1.8	8(8)	8(8)	8(8)	8(8)	8(8)	8(8)
1.3	8(8)	8(8)	8(8)	8(8)	8(8)	8(8)
1.0	8(8)	8(8)	8(8)	8(8)	8(8)	8(8)
0.8	7(7)	7(7)	7(7)	7(7)	7(7)	7(7)
TOL $= 10^{-8}$						
	Example 1					
1.3	9(9)	8(8)	9(9)	9(9)	9(9)	9(9)
1.0	9(9)	9(10)	9(10)	9(10)	9(10)	9(10)
0.8	8(8)	8(8)	8(8)	8(8)	8(8)	8(8)
	Example 2					
2.5	11(11)	9(9)	10(10)	10(10)	11(11)	11(11)
2.0	11(11)	10(10)	10(10)	9(9)	9(9)	9(9)
1.5	11(11)	10(10)	10(10)	10(10)	9(9)	9(9)
1.2	11(11)	10(10)	10(10)	9(9)	9(9)	9(9)
	Example 3					
1.8	9(9)	9(9)	9(9)	9(9)	9(9)	9(9)
1.3	10(10)	10(10)	10(10)	10(10)	10(10)	10(10)
1.0	10(10)	10(10)	10(10)	10(10)	10(10)	10(10)
0.8	9(9)	9(9)	9(9)	9(9)	9(9)	9(9)

In the case of regularization by restriction, we chose a stronger perturbation since the given constraints force a strong regularization. For the perturbation $0.05\sin(10\pi x - 0.5\pi)$ we obtained almost the same results as in Table 2. As in Table 2, the first numbers of each column in Table 4 show the number of iterations needed for the termination criteria $||q_k^{MN} - P(q_k^{MN} - F_N'(q_k^{MN})^* F_N(q_k^{MN}))|| <$ TOL; the numbers in parentheses show the number of iterations needed to satisfy the termination criteria $||F_N'(q_k^{MN})^* F_N(q_k^{MN}) + \mu_k^{MN} q_k^{MN}|| <$ TOL.

Acknowledgments. Most of this work was done while the author was visiting Rice University, Houston. The author would like to thank the Gottlieb-Daimler and Karl-Benz-Stiftung for their financial support; the members of the Department of Mathematical Sciences and the Center for Research on Parallel Computation (CRPC) for making the stay at Rice University a very pleasant one; and especially Prof. J. E.

Dennis, Jr., of Rice University, and Prof. E. W. Sachs, of Universität Trier, for their encouraging support. Finally, the careful reading and constructive comments of two referees are gratefully acknowledged.

REFERENCES

[1] E. L. Allgower and K. Böhmer, *Application of the mesh independence principle to mesh refinement strategies*, SIAM J. Numer. Anal., 24 (1987), pp. 1335–1351.

[2] E. L. Allgower, K. Böhmer, F. A. Potra, and W. C. Rheinboldt, *A mesh-independence principle for operator equations and their discretizations*, SIAM J. Numer. Anal., 23 (1986), pp. 160–169.

[3] W. Alt, *Lipschitzian perturbations of infinite optimization problems*, in Mathematical Programming with Data Perturbations II, A. Fiacco, ed., Marcel Dekker, New York, Basel, 1983, pp. 7–21.

[4] O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems*, Academic Press, New York, 1984.

[5] H. T. Banks and K. Kunisch, *Estimation Techniques for Distributed Parameter Systems*, in Systems & Control: Foundations & Applications, Birkhäuser-Verlag, Basel, Switzerland, 1989.

[6] H. G. Bock, *Randwertprobleme zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, Preprint Nr. 442, Institut für Angewandte Mathematik, Universität Heidelberg, Heidelberg, Germany, 1988.

[7] F. Colonius and K. Kunisch, *Stability for parameter estimation in two point boundary value problems*, J. Reine Angew. Math., 370 (1986), pp. 1–29.

[8] ———, *Output least squares stability in elliptic systems*, Appl. Math. Optim., 19 (1989), pp. 33–63.

[9] L. Debnath and P. Mikusinski, *Introduction to Hilbert Spaces with Applications*, Academic Press, New York, 1990.

[10] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[11] J. E. Dennis, Jr., *Nonlinear least squares and equations*, in The State of The Art in Numerical Analysis, D. Jacobs, ed., Academic Press, London, 1977, pp. 269–312.

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

[13] P. Deuflhard and V. Apostulescu, *A study of the Gauss–Newton algorithm for the solution of nonlinear least squares problems*, in Special Topics of Applied Mathematics, J. Frehse, D. Pallaschke, and U. Trottenberg, eds., North Holland, Amsterdam, New York, 1980, pp. 129–150.

[14] P. Deuflhard and G. Heindl, *Affine invariant convergence theorems for Newton's method and extensions to related methods*, SIAM J. Numer. Anal., 16 (1979), pp. 1–10.

[15] H. W. Engl, K. Kunisch, and A. Neubauer, *Convergence rates for Tikhonov regularization of non-linear ill-posed problems*, Inverse Problems, 5 (1989), pp. 523–540.

[16] M. Heinkenschloss, M. Laumen, and E. W. Sachs, *Gauss-Newton methods with grid refinement*, in Optimal Control of Partial Differential Equations, F. Kappel and K. Kunisch, eds., Birkhäuser-Verlag, Basel, Boston, Berlin, 1991, pp. 161–174.

[17] C. T. Kelley, *Operator prolongation methods for nonlinear equations*, in Computational Solution of Nonlinear Systems of Equations, E. Allgower and G. Georg, eds., Lectures in Appl. Math., 26, Amer. Math. Soc., Providence, RI, 1990, pp. 359–388.

[18] C. T. Kelley and E. W. Sachs, *Broyden's method for approximate solution of nonlinear integral equations*, J. Integral Equations, 9 (1985), pp. 25–44.

[19] ———, *Approximate quasi-Newton methods*, Math. Programming, 48 (1990), pp. 41–70.

[20] C. Kravaris and J. H. Seinfeld, *Identification of parameters in distributed parameter systems by regularization*, SIAM J. Control Optim., 23 (1985), pp. 217–241.

[21] M. Kroller and K. Kunisch, *A numerical study of an augmented Lagrangian method for the estimation of parameters in a two point boundary value problem*, Tech. Rep. 87, Technical Univ. of Graz, Austria, 1987.

[22] J. Martinez, *An algorithm for solving sparse nonlinear least squares problems*, Computing, 39 (1987), pp. 307–325.

[23] H. Maurer and J. Zowe, *First- and second-order necessary and sufficient optimality conditions for infinite-dimensional programming problems*, Math. Programming, 16 (1979), pp. 98–110.

[24] J. J. MORÉ, *The Levenberg–Marquardt algorithm: Implementation and theory*, in Numerical Analysis, Proceedings, Biennial Conference, Dundee 1977, G. A. Watson, ed., Springer-Verlag, Berlin, New York, 1977, pp. 105–116.

[25] ———, *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, New York, 1983, pp. 258–287.

[26] S. OMATU AND J. H. SEINFELD, *Distributed Parameter Systems. Theory and Applications*, Oxford University Press, Oxford, 1989.

[27] R. SCHABACK, *Convergence of the general Gauss–Newton algorithm*, Numer. Math., 46 (1985), pp. 281–309.

[28] C. R. VOGEL, *A constrained least squares regularization method for nonlinear ill-posed problems*, SIAM J. Control Optim., 28 (1990), pp. 34–49.

[29] P.-Å. WEDIN AND P. LINDSTRÖM, *Methods and software for nonlinear least squares problems*, Report UMINF–133.87, Institute of Information Processing, University of Umeå, Umeå, Sweden, 1987.

SIAM J. OPTIMIZATION
Vol. 3, No. 1, pp. 118–133, February 1993

A SUPERLINEARLY CONVERGENT POLYNOMIAL PRIMAL-DUAL INTERIOR-POINT ALGORITHM FOR LINEAR PROGRAMMING[*]

YIN ZHANG[†] AND RICHARD A. TAPIA[‡]

Abstract. The choices of the centering parameter and the step-length parameter are the fundamental issues in primal-dual interior-point algorithms for linear programming. Various choices for these two parameters that lead to polynomial algorithms have been proposed. Recently, Zhang, Tapia, and Dennis derived conditions for the choices of the two parameters that were sufficient for superlinear or quadratic convergence. However, prior to this work it had not been shown that these conditions for fast convergence are compatible with the choices that lead to polynomiality; none of the existing polynomial primal-dual interior-point algorithms satisfy these fast convergence requirements. This paper gives an affirmative answer to the question: Can a primal-dual algorithm be both polynomial and superlinearly convergent for general problems? A "large step" algorithm that possesses both polynomiality and, under the assumption of the convergence of the iteration sequence, Q-superlinear convergence, is constructed and analyzed. For nondegenerate problems, the convergence is actually Q-quadratic.

Key words. linear programming, primal-dual interior-point algorithms, quadratic and superlinear convergence, polynomiality

AMS(MOS) subject classifications. 90C05, 65K05

1. Introduction.

We consider linear programs in the standard form:

$$\text{(1)} \qquad \begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b, \quad x \geq 0, \end{array}$$

where $c, x \in \mathbf{R}^n$, $b \in \mathbf{R}^m$, $A \in \mathbf{R}^{m \times n} (m < n)$, and A is assumed to have full rank m.

The first-order optimality conditions for (1) can be written

$$\text{(2)} \qquad \begin{pmatrix} Ax - b \\ A^T\lambda + y - c \\ XYe \end{pmatrix} = 0, \qquad (x, y) \geq 0,$$

where λ and y are dual variables, $X = \text{diag}(x)$, $Y = \text{diag}(y)$, and e has all components equal to one. To facilitate our presentation, we will eliminate the dual variable λ from the above system (although such an elimination may not be advisable from a practical point of view). Let $B \in \mathbf{R}^{(n-m)\times n}$ be any matrix such that the columns of B^T form a basis for the null space of A. Premultiply the second equation by the nonsingular matrix $[A^T \; B^T]^T$. Notice that $BA^T = 0$, so

$$0 = \begin{bmatrix} A \\ B \end{bmatrix} (A^T\lambda + y - c) = \begin{pmatrix} AA^T\lambda + A(y - c) \\ By - Bc \end{pmatrix}.$$

[*] Received by the editors May 10, 1991; accepted for publication (in revised form) December 3, 1991.

[†] Department of Mathematics and Statistics, University of Maryland, Baltimore, Maryland 21228-5398; visiting member of the Center for Research in Parallel Computation, Rice University, Houston, Texas 77251-1892. This research was supported in part by National Science Foundation Cooperative Agreement CCR-8809615, National Science Foundation grant DMS-9102761, and Department of Energy grant DE-FG05-91ER25100.

[‡] Department of Mathematical Sciences and the Center for Research in Parallel Computation, Rice University, Houston, Texas 77251-1892. This research was supported in part by National Science Foundation Cooperative Agreement CCR-8809615, Air Force Office of Scientific Research grant 89-0363, Department of Energy grant DEFG05-86ER25017, and Army Research Office grant 9DAAL03-90-G-0093.

Since AA^T is nonsingular, λ is uniquely determined once y is known. Removing the equation for λ, we arrive at the following $2n \times 2n$ nonlinear system with nonnegativity constraints on the variables:

$$F(x,y) = \begin{pmatrix} Ax - b \\ By - Bc \\ XYe \end{pmatrix} = 0, \qquad (x,y) \geq 0. \tag{3}$$

By the feasibility set of problem (3) we mean:

$$\Omega = \{(x,y) : x, y \in \mathbf{R}^n, Ax = b, By = Bc, (x,y) \geq 0\}.$$

A feasible pair $(x,y) \in \Omega$ is said to be strictly feasible if it is positive. In this work we tacitly assume that strictly feasible points exist.

It is easy to see that for $(x,y) \in \Omega$, $\|F(x,y)\|_1 = x^T y$, which can be shown to be the duality gap for problem (1); we will use the duality gap as the merit function for our algorithm, i.e., the criterion that tells us when one feasible point should be preferred to another.

Mathematically speaking, the concepts of polynomiality and rate of convergence are incompatible. Polynomiality is meaningful only for algorithms that terminate in a finite number of steps, while rate of convergence is defined only for algorithms that take an infinite number of steps to converge. When we say that an interior-point algorithm is polynomial, we have in mind integral (or rational) data and finite termination. On the other hand, when we say the same algorithm is linearly convergent, for example, we do so in the traditional numerical analysis sense. With this understanding, we can discuss both polynomiality and rate of convergence of an algorithm at the same time.

It is clearly desirable to develop algorithms that possess both polynomiality and fast asymptotic convergence, or, in other words, both good global behavior and good local behavior. To our knowledge, the only prior work in this direction is Yamashita [10]. Using the multiplicative penalty function of Iri and Imai [2], Yamashita constructed a polynomial primal algorithm and demonstrated its quadratic convergence under the following two assumptions: (i) the optimal objective value is known, and (ii) the iteration sequence converges to a nondegenerate optimal vertex. The first assumption is not realistic in general. The second assumption is very restrictive because most practical problems are degenerate.

The objective of this work is to construct a primal-dual interior-point algorithm for problem (1) that possesses both polynomiality and fast convergence under more realistic and less restrictive assumptions. We construct such an algorithm and show that it takes at most $O(nL)$ iterations to reduce the duality gap to 2^{-L}. Moreover, we demonstrate that this algorithm gives quadratic convergence for nondegenerate problems and gives Q-superlinear convergence for degenerate problems.

Subscripts will be used to distinguish values of quantities at a particular iteration and superscripts will indicate components of vectors. We also use the notation

$$\min(v) = \min_{1 \leq i \leq n} v^i \quad \text{and} \quad \max(v) = \max_{1 \leq i \leq n} v^i$$

for a vector $v \in \mathbf{R}^n$. The symbol $\|\cdot\|$ denotes the ℓ_2 norm unless otherwise stated. We will use the standard big-O notation in this paper; in particular, for a sequence $\{v_k\} \subset \mathbf{R}^n$ and a positive sequence $\{\alpha_k\} \subset \mathbf{R}$, $v_k = O(\alpha_k)$ implies the existence of positive constants β and k_0 such that $\|v_k\| \leq \beta \alpha_k$ for all $k > k_0$.

The paper is organized as follows. In §2, we describe a general interior-point algorithmic framework for problem (1) based on the nonlinear system (3) and give a brief survey of existing results for algorithms that fall into this framework. In §§3 and 4, we specify our procedures for determining the step length and for choosing the centering parameter. We state our algorithm in §5. Global linear convergence (and polynomiality) are established in §6. Quadratic convergence for nondegenerate problems is established in §7, and superlinear convergence for all problems is established in §8. Concluding remarks are given in §9.

2. General algorithm.

We now present a general framework for the primal-dual interior-point algorithms.

ALGORITHM 1 (General Algorithm). Given a strictly feasible pair (x_0, y_0). For $k = 0, 1, 2, \ldots$, do:

Step 1. Compute the Newton step

$$\begin{pmatrix} \Delta x_k^N \\ \Delta y_k^N \end{pmatrix} = -[F'(x_k, y_k)]^{-1} F(x_k, y_k)$$

and the centering step

$$\begin{pmatrix} \Delta x_k^C \\ \Delta y_k^C \end{pmatrix} = \frac{1}{n} x_k^T y_k [F'(x_k, y_k)]^{-1} \begin{pmatrix} 0 \\ e \end{pmatrix}.$$

Step 2. Choose $\sigma_k \in (0,1)$ and form the combined step

$$\begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} = \begin{pmatrix} \Delta x_k^N \\ \Delta y_k^N \end{pmatrix} + \sigma_k \begin{pmatrix} \Delta x_k^C \\ \Delta y_k^C \end{pmatrix}.$$

Step 3. Choose $\tau_k \in (0,1)$ and set $\alpha_k = \tau_k \hat{\alpha}_k$, where

$$\hat{\alpha}_k = \frac{-1}{\min(X_k^{-1} \Delta x_k, Y_k^{-1} \Delta y_k)}.$$

Step 4. Compute the new iterate

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} + \alpha_k \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix}.$$

We will now briefly comment on this general algorithmic framework. From a direct calculation, we have

$$F'(x, y) = \begin{bmatrix} A & 0 \\ 0 & B \\ Y & X \end{bmatrix}. \tag{4}$$

Since we assumed that A has full rank, it is a straightforward matter to verify that $F'(x, y)$ is nonsingular for any positive pair (x, y). In addition, relation (8) below guarantees that $\hat{\alpha}_k > 0$. Hence the iterates produced by Algorithm 1 are well defined. Notice that the restriction $\alpha_k < \hat{\alpha}_k$ guarantees that the iterates remain strictly feasible. Moreover, we have the following useful relationships:

$$Y_k \Delta x_k^N + X_k \Delta y_k^N = -X_k Y_k e, \tag{5}$$

$$Y_k \Delta x_k^C + X_k \Delta y_k^C = \tfrac{1}{n} x_k^T y_k e, \tag{6}$$

$$Y_k \Delta x_k + X_k \Delta y_k = -X_k Y_k e + \sigma_k \tfrac{1}{n} x_k^T y_k e, \tag{7}$$

$$\Delta x_k^T \Delta y_k = 0, \tag{8}$$

$$x_{k+1}^T y_{k+1} = x_k^T y_k (1 - (1 - \sigma_k)\alpha_k). \tag{9}$$

We have stated Algorithm 1 in this form for notational convenience. It is not difficult to verify that identical iterates $\{(x_k, y_k)\}$ can be generated using (2) instead of (3). For this case, there is no need to introduce the matrix B (see [11], for example).

From (9) we see that Algorithm 1 is a descent algorithm for the duality gap $\|F(x,y)\|_1 = x^T y$. Moreover, the duality gap is reduced at iteration k by a factor $1 - \alpha_k(1 - \sigma_k) < 1$; thus, linear convergence will be obtained if $\{\alpha_k\}$ is bounded away from zero and if $\{\sigma_k\}$ is bounded away from one. In addition, Q-superlinear convergence will be obtained if $\alpha_k(1 - \sigma_k) \to 1$. Observe that we have direct control over the choice of σ_k. However, we do not have the freedom of choosing α_k uniformly bounded away from zero, since we must enforce the requirement $\alpha_k < \hat{\alpha}_k$ and $\hat{\alpha}_k$ is not directly under our control.

A number of existing primal-dual algorithms fit into the above general algorithmic framework with different choices for the parameters σ_k and α_k. For example, in the primal-dual algorithm of Kojima, Mizuno, and Yoshise [3], σ_k is a constant and α_k is a particular function of σ_k. They showed that their algorithm requires at most $O(nL)$ iterations to reduce the duality gap by a factor of 2^{-L}. Other examples include the Todd and Ye [9] primal-dual potential-reduction algorithm and the Monteiro and Adler [7] path-following primal-dual algorithm. Todd and Ye's algorithm uses the choice

$$\sigma_k = \frac{\sqrt{n}}{\sqrt{n} + \nu},$$

where ν is a constant. In Monteiro and Adler's algorithm,

$$\sigma_k = 1 - \frac{\delta}{\sqrt{n}},$$

where δ is a constant (Monteiro and Adler actually used $\delta = 0.35$ in their analysis). In both algorithms, a rather short step length α_k is required. Furthermore, both of these algorithms require at most $O(\sqrt{n}L)$ iterations to reduce the duality gap to 2^{-L}. This is the best complexity bound obtained for linear programming so far. Observe that all three algorithms use constant σ_k. In each of the three cases, if σ denotes the constant value of σ_k, then Q-superlinear convergence is possible (see (9)) only if

$$\alpha_k \to \frac{1}{1 - \sigma},$$

which seems extremely unlikely.

In analyzing the convergence of Algorithm 1, a central quantity is

$$\eta_k = \frac{x_k^T y_k / n}{\min(X_k Y_k e)}. \tag{10}$$

Since $\frac{1}{n} x_k^T y_k$ is the average value of the components of $X_k Y_k e$, it is clear that $\eta_k \geq 1$. In all the above-mentioned polynomial algorithms, it is essential that the sequence $\{\eta_k\}$ be bounded.

Recently, Zhang, Tapia, and Dennis [11] showed that under appropriate assumptions, Algorithm 1 has fast convergence. The following two theorems summarize their main results. By a nondegenerate vertex of (1), we mean a feasible point of (1) that has exactly m positive components and the corresponding m columns of A are linearly independent.

THEOREM 2.1 (see [11]). *Let (x_*, y_*) be a solution of problem* (3) *and let* $\{(x_k, y_k)\}$ *be generated by Algorithm* 1. *Assume that*

(i) *strict complementarity holds at* (x_*, y_*);
(ii) x_* *is a nondegenerate vertex of* (1);
(iii) $\sigma_k = O(x_k^T y_k)$ *and* $\tau_k = 1 - O(x_k^T y_k)$.

If $\{(x_k, y_k)\}$ *converges to* (x_*, y_*), *then the convergence is Q-quadratic.*

THEOREM 2.2 (see [11]). *Let* (x_*, y_*) *be a solution of problem* (3) *and* $\{(x_k, y_k)\}$ *be generated by Algorithm* 1. *Assume that*

(i) *strict complementarity holds at* (x_*, y_*);
(ii) *the sequence* $\{\eta_k\}$ *is bounded*;
(iii) $\sigma_k \to 0$ *and* $\tau_k \to 1$.

If $\{(x_k, y_k)\}$ *converges to* (x_*, y_*), *then the duality gap sequence* $\{x_k^T y_k\}$ *converges to zero Q-superlinearly.*

With some additional work, we can actually demonstrate that the sequence $\{X_k Y_k e\}$ componentwise converges to zero Q-superlinearly.

Several assumptions have been made in the above theorems. Our numerical experiments have led us to believe that the strict complementarity assumption is not restrictive. On the other hand, the nondegeneracy assumption is quite restrictive since degeneracy exists in most real-world problems. For degenerate solutions, the best convergence that has been established is Q-superlinear, as stated in Theorem 2.2.

Although many of the existing polynomial primal-dual interior-point algorithms satisfy assumption (ii) of Theorem 2.2, none of them satisfy assumption (iii), i.e., $\sigma_k \to 0$ and $\tau_k \to 1$. In fact, in several polynomial algorithms, for example, Todd and Ye's and Monteiro and Adler's, the values of σ_k are close to one. From Zhang, Tapia, and Dennis [11] it follows that these algorithms will most likely have slow Q-linear convergence. Hence while their global behavior may be excellent, their local behavior can be improved.

Recently, in a number of performance-oriented primal-dual algorithms, for example, those implemented by Choi, Monma, and Shanno [1]; McShane, Monma, and Shanno [6]; and Lustig, Marsten, and Shanno [5], very small values of σ_k were used and long steps were also taken. Impressive numerical results were obtained from these implementations, although polynomial complexity bounds are not known. Hence while their local behavior may be good, their global behavior is questionable from a theoretical standpoint.

In this work, we develop a primal-dual interior-point polynomial algorithm that gives quadratic convergence for nondegenerate solutions and gives superlinear convergence for degenerate solutions. Hence, from a mathematical point of view, both the global and the local behavior will be good. This new algorithm is still of a theoretical nature. However, the fact that polynomiality and quadratic or superlinear convergence can be achieved simultaneously by one algorithm provides motivation for practical implementations of the conditions $\sigma_k = O(x_k^T y_k)$ and $\tau_k = 1 - O(x_k^T y_k)$ for fast convergence.

3. Determining the step length.

In the previous section we mentioned that both polynomiality and superlinear convergence essentially require that the sequence $\{\eta_k\}$ be bounded. The most straightforward way of accomplishing this objective is to explicitly enforce a uniform bound on the quantity

$$\eta_{k+1} = \frac{x_{k+1}^T y_{k+1}/n}{\min(X_{k+1} Y_{k+1} e)}$$

during the process of choosing the step length α_k; i.e., ask that

$$\frac{1}{\eta_{k+1}} = \frac{\min(X_{k+1}Y_{k+1}e)}{x_{k+1}^T y_{k+1}/n} \geq \gamma \tag{11}$$

for some $\gamma > 0$.

Following the notation used in [3], let

$$\begin{array}{ll} x_k(\alpha) = x_k + \alpha\Delta x_k, & y_k(\alpha) = y_k + \alpha\Delta y_k, \\ f_k(\alpha) = X_k(\alpha)Y_k(\alpha)e, & f_k^{\text{ave}}(\alpha) = \frac{1}{n}x_k(\alpha)^T y_k(\alpha), \\ f_k^{\min}(\alpha) = \min(f_k(\alpha)), & f_k^{\max}(\alpha) = \max(f_k(\alpha)). \end{array} \tag{12}$$

Note that the above quantities actually also depend on the centering parameter σ because both Δx_k and Δy_k are functions of σ (see Step 3 of Algorithm 1). However, since we will always choose σ before we determine α, it will suffice to consider these quantities only as functions of α for a fixed value of σ.

Whenever $\alpha = 0$, we will drop the argument from the above functions. For example, $x_k \equiv x_k(0)$, $f_k^{\text{ave}} \equiv f_k^{\text{ave}}(0)$, and so on. From the formula for the iterates (Step 4 of Algorithm 1), we also have $x_{k+1} = x_k(\alpha_k)$, $f_{k+1}^{\text{ave}} = f_k^{\text{ave}}(\alpha_k)$, and so on.

Using the above notation, we choose the form of condition (11) as requiring α_k to satisfy

$$\frac{f_k^{\min}(\alpha)}{f_k^{\text{ave}}(\alpha)} \geq \gamma_k, \qquad \alpha > 0, \tag{13}$$

where

$$\gamma_k \in [\gamma, f_k^{\min}/f_k^{\text{ave}}], \quad 0 < \gamma \leq f_0^{\min}/f_0^{\text{ave}} \leq 1, \quad \text{and} \quad \gamma, \gamma_k < 1. \tag{14}$$

In the case $1/\eta_0 > \gamma$, we allow $1/\eta_k$ to decrease monotonically as long as $1/\eta_k > \gamma$.

In the following development, we use some of the techniques developed by Kojima, Mizuno, and Yoshise [3].

Using (12), (7), (8), and (9), and letting

$$s_k = \text{diag}(\Delta x_k)\Delta y_k,$$

we have

$$f_k^i(\alpha) = f_k^i - (f_k^i - \sigma_k f_k^{\text{ave}})\alpha + s_k^i \alpha^2 \tag{15}$$

and

$$f_k^{\text{ave}}(\alpha) = f_k^{\text{ave}}[1 - (1 - \sigma_k)\alpha]. \tag{16}$$

Hence $f_k^i(\alpha)$ is a quadratic (so $f_k^{\min}(\alpha)$ and $f_k^{\max}(\alpha)$ are piecewise quadratic) and $f_k^{\text{ave}}(\alpha)$ is linear. Clearly, if $f_k^{\text{ave}}(\hat{\alpha}_k) = 0$, then $(x_k(\hat{\alpha}_k), y_k(\hat{\alpha}_k))$ will solve problem (3). In the sequel, we always assume $f_k^{\text{ave}}(\hat{\alpha}_k) > 0$.

For notational convenience, we introduce the piecewise quadratic function

$$h(\alpha) \overset{\text{def}}{=} f_k^{\min}(\alpha) - \gamma_k f_k^{\text{ave}}(\alpha). \tag{17}$$

It follows that condition (13) is equivalent to

$$h(\alpha) \geq 0, \qquad \alpha > 0. \tag{18}$$

In determining α_k we use the following quantity:

$$\alpha_k^\gamma \overset{\text{def}}{=} \min\{\alpha > 0 : h(\alpha) = 0\}. \tag{19}$$

Recall that $\hat{\alpha}_k$ is defined in Step 3 of the General Algorithm (see §2).

LEMMA 3.1. *The quantity α_k^γ is well defined and $\alpha_k^\gamma \in (0, \hat{\alpha}_k)$. Moreover, condition* (13) *is satisfied for all $\alpha \in (0, \alpha_k^\gamma]$.*

Proof. Let us examine the function $h(\alpha)$. It follows from the definitions of γ_k and $\hat{\alpha}_k$ that

$$h(0) = f_k^{\min} - \gamma_k f_k^{\text{ave}} \geq 0$$

and

$$h(\hat{\alpha}_k) = f_k^{\min}(\hat{\alpha}_k) - \gamma_k f_k^{\text{ave}}(\hat{\alpha}_k) = -\gamma_k f_k^{\text{ave}}(\hat{\alpha}_k) < 0.$$

Hence it follows from the continuity of $h(\alpha)$ that $h(\alpha)$ has a root in $[0, \hat{\alpha}_k)$. When $h(0) > 0$, $h(\alpha)$ obviously has a root in $(0, \hat{\alpha}_k)$. When $h(0) = 0$, it can be verified that the right-derivative of $h(\alpha)$ at $\alpha = 0$ is

$$\begin{aligned} h'(0^+) &= -(f_k^{\min} - \sigma_k f_k^{\text{ave}}) + \gamma_k(1 - \sigma_k) f_k^{\text{ave}} \\ &= [(1 - \gamma_k)\sigma_k + (\gamma_k - f_k^{\min}/f_k^{\text{ave}})] f_k^{\text{ave}} \\ &= (1 - \gamma_k)\sigma_k f_k^{\text{ave}} > 0. \end{aligned}$$

Therefore, $h(\alpha) > 0$ for sufficiently small but positive α. Consequently, $\alpha_k^\gamma > 0$.

Since $h(\hat{\alpha}_k) < 0$, we have $\alpha_k^\gamma < \hat{\alpha}_k$. It is evident that $h(\alpha) \geq 0$ for $\alpha \in (0, \alpha_k^\gamma]$, i.e., condition (13) is satisfied. This completes the proof. □

An equivalent expression for α_k^γ is

$$\alpha_k^\gamma = \min\{\alpha > 0 : f_k^i(\alpha) - \gamma_k f_k^{\text{ave}}(\alpha) = 0,\ i = 1, 2, \ldots, n\}. \tag{20}$$

The computation of α_k^γ involves calculating the roots of at most n quadratics and therefore requires $O(n)$ operations.

In addition to a lower bound for $\{f_k^i(\alpha_k)/f_k^{\text{ave}}(\alpha_k)\}$ (i.e., condition (13)), we also impose an upper bound on these quantities; namely, we require α_k to satisfy

$$\frac{f_k^{\max}(\alpha)}{f_k^{\text{ave}}(\alpha)} \leq \Gamma_k, \qquad \alpha > 0, \tag{21}$$

where

$$\Gamma_k \in [f_k^{\max}/f_k^{\text{ave}}, \Gamma], \quad 1 \leq f_0^{\max}/f_0^{\text{ave}} \leq \Gamma \leq n, \quad \text{and} \quad \Gamma, \Gamma_k > 1. \tag{22}$$

Since $f_k^i(\alpha)/f_k^{\text{ave}}(\alpha) < n$ for all i, condition (21) will be redundant if $\Gamma_k = n$. We introduce condition (21) to improve our complexity bound. We do not feel that enforcing this condition will have much practical significance.

Following the treatment of condition (13), we introduce the piecewise quadratic function

$$H(\alpha) \overset{\text{def}}{=} f_k^{\max}(\alpha) - \Gamma_k f_k^{\text{ave}}(\alpha). \tag{23}$$

It follows that condition (21) is equivalent to

$$H(\alpha) \leq 0, \qquad \alpha > 0. \tag{24}$$

We will also use the following quantity in determining α_k:

$$\alpha_k^\Gamma \overset{\text{def}}{=} \min\{\alpha > 0 : H(\alpha) = 0\}. \tag{25}$$

Analogous to Lemma 3.1 for condition (13), we have the following lemma for condition (21).

LEMMA 3.2. *The quantity α_k^Γ is well defined and $\alpha_k^\Gamma \in (0, \hat{\alpha}_k)$. Moreover, condition* (21) *is satisfied by all $\alpha \in (0, \alpha_k^\Gamma]$.*

Proof. The proof is similar to that for Lemma 3.1, so we omit it. □

Analogous to the expression (20) for condition (13), we have for condition (21)

$$\alpha_k^\Gamma = \min\{\alpha > 0 : f_k^i(\alpha) - \Gamma_k f_k^{\text{ave}}(\alpha) = 0,\ i = 1, 2, \ldots, n\}. \tag{26}$$

For the sake of simplicity, we will enforce the conditions

$$\gamma_k \le \tfrac{1}{2} \quad \text{and} \quad \Gamma_k \ge 2. \tag{27}$$

The specific values in (27) do not constitute a loss of generality because they will only affect expressions for some constants in our analysis. These values of γ_k and Γ_k will result in much simplified expressions for those constants.

From (5), we see that for fixed σ_k a larger step length α_k will produce a larger reduction in the duality gap. So it is always desirable to take the largest step length possible as long as other requirements are satisfied. Our procedure for determining the step length α_k is summarized as follows.

PROCEDURE 1 (step-length criterion). Given positive constants γ and Γ such that

$$0 < \gamma \le \min(\tfrac{1}{2}, f_0^{\min}/f_0^{\text{ave}}), \qquad \max(2, f_0^{\max}/f_0^{\text{ave}}) \le \Gamma < n: \tag{28}$$

Step 1. Choose $\gamma_k \in [\gamma, \min(\frac{1}{2}, f_k^{\min}/f_k^{\text{ave}})]$ and $\Gamma_k \in [\max(2, f_k^{\max}/f_k^{\text{ave}}), \Gamma]$.

Step 2. Compute $\alpha_k^\gamma = \min\{\alpha > 0 : f_k^i(\alpha) - \gamma_k f_k^{\text{ave}}(\alpha) = 0,\ i = 1, 2, \ldots, n\}$ (i.e., (19)).

Step 3. Compute $\alpha_k^\Gamma = \min\{\alpha > 0 : f_k^i(\alpha) - \Gamma_k f_k^{\text{ave}}(\alpha) = 0,\ i = 1, 2, \ldots, n\}$ (i.e., (25)).

Step 4. Let $\alpha_k = \min(\alpha_k^\gamma, \alpha_k^\Gamma)$.

We note that the above procedure for choosing the step length bears a certain similarity to a procedure recently proposed by Mizuno, Todd, and Ye [8].

Now we prove two technical lemmas that will be needed in the later development.

LEMMA 3.3. *For $\alpha \in [0, 1]$,*

$$f_k^{\min}(\alpha) \ge f_k^{\min} - (f_k^{\min} - \sigma_k f_k^{\text{ave}})\alpha + \min(s_k)\alpha^2,$$
$$f_k^{\max}(\alpha) \le f_k^{\max} - (f_k^{\max} - \sigma_k f_k^{\text{ave}})\alpha + \max(s_k)\alpha^2.$$

Proof. We first look at the linear part of $f_k^i(\alpha)$. Since for all i,

$$f_k^i - (f_k^i - \sigma_k f_k^{\text{ave}})\alpha = \begin{cases} f_k^i, & \alpha = 0, \\ \sigma_k f_k^{\text{ave}}, & \alpha = 1, \end{cases}$$

it is evident that for $\alpha \in [0, 1]$,

$$f_k^{\min} - (f_k^{\min} - \sigma_k f_k^{\text{ave}})\alpha \le f_k^i - (f_k^i - \sigma_k f_k^{\text{ave}})\alpha \le f_k^{\max} - (f_k^{\max} - \sigma_k f_k^{\text{ave}})\alpha.$$

For the quadratic terms, we clearly have

$$\min(s_k)\alpha^2 \le s_k^i\alpha^2 \le \max(s_k)\alpha^2.$$

By adding the quadratic terms to their corresponding linear parts, we thus finish the proof. □

It is worth noting that $e^T s_k = 0$ by (8). Hence, $\min(s_k) \le 0$ and $\max(s_k) \ge 0$. In the sequel, we will adopt the convention that $\frac{1}{0} = +\infty$.

LEMMA 3.4. *Let α_k be given by Procedure* 1. *Then*

$$\alpha_k \ge \min\left(1, \frac{(1-\gamma_k)\sigma_k f_k^{\text{ave}}}{-\min(s_k)}, \frac{(\Gamma_k - 1)\sigma_k f_k^{\text{ave}}}{\max(s_k)}\right). \tag{29}$$

Moreover,

$$\alpha_k \ge \min\left(1, \frac{\sigma_k f_k^{\text{ave}}}{2\|s_k\|_\infty}\right). \tag{30}$$

Proof. From (19), α_k^γ is a positive root of $f_k^i(\alpha) - \gamma_k f_k^{\text{ave}}(\alpha)$ for some index i. Noticing that for $\alpha \in [0,1]$ $f_k^{\text{ave}}(\alpha)$ is positive, and using Lemma 3.3, for $\alpha \in [0,1]$, $\gamma_k \ge 0$, and for all indices i, we have

$$\begin{aligned} f_k^i(\alpha) - \gamma_k f_k^{\text{ave}}(\alpha) &\ge f_k^{\min} - (f_k^{\min} - \sigma_k f_k^{\text{ave}})\alpha + \min(s_k)\alpha^2 - \gamma_k f_k^{\text{ave}}(\alpha) \\ &= (f_k^{\min} - \gamma_k f_k^{\text{ave}})(1-\alpha) + (1-\gamma_k)\sigma_k f_k^{\text{ave}}\alpha + \min(s_k)\alpha^2 \\ &\ge (1-\gamma_k)\sigma_k f_k^{\text{ave}}\alpha + \min(s_k)\alpha^2. \end{aligned} \tag{31}$$

If $\min(s_k) = 0$, then $h(\alpha) > 0$ for $\alpha \in (0,1]$. Therefore, we will have $\alpha_k^\gamma > 1$. Now assume $\min(s_k) < 0$. Then the quadratic on the right-hand side of the last inequality in (31) has a unique positive root

$$\bar{\alpha}_k = \frac{(1-\gamma_k)\sigma_k f_k^{\text{ave}}}{-\min(s_k)}.$$

Hence, if $\alpha_k^\gamma \le 1$, from (31) we must have $\alpha_k^\gamma \ge \bar{\alpha}_k$. This proves that

$$\alpha_k^\gamma \ge \min\left(1, \frac{(1-\gamma_k)\sigma_k f_k^{\text{ave}}}{-\min(s_k)}\right). \tag{32}$$

Similarly, we can prove that

$$\alpha_k^\Gamma \ge \min\left(1, \frac{(\Gamma_k - 1)\sigma_k f_k^{\text{ave}}}{\max(s_k)}\right). \tag{33}$$

Combining (32) and (33), we obtain (29).

Finally, (30) follows from the facts that $\|s_k\|_\infty = \max\{-\min(s_k), \max(s_k)\}$ and

$$\tfrac{1}{2} \le 1 - \gamma_k < 1 \le \Gamma_k - 1.$$

This completes the proof. □

4. Choosing the centering parameter. We will use the following notation:

$$
\begin{aligned}
p_k &= X_k^{-1}\Delta x_k, & q_k &= Y_k^{-1}\Delta y_k, \\
p_k^N &= X_k^{-1}\Delta x_k^N, & q_k^N &= Y_k^{-1}\Delta y_k^N, \\
p_k^C &= X_k^{-1}\Delta x_k^C, & q_k^C &= Y_k^{-1}\Delta y_k^C,
\end{aligned} \tag{34}
$$

and

$$
\omega_k = \max_{1\le i\le n} (|(p_k^N)^i(q_k^N)^i|, |(p_k^N)^i(q_k^C)^i|, |(p_k^C)^i(q_k^N)^i|, |(p_k^C)^i(q_k^C)^i|). \tag{35}
$$

LEMMA 4.1. *If $f_k^{\min}/f_k^{\text{ave}} \ge \gamma$, then*

$$\omega_k \le n/\gamma^2.$$

Proof. Multiply both sides of (6) by $(X_kY_k)^{-\frac{1}{2}}$ and consider the square of the ℓ_2-norm of both sides. Using (8) and (34), we obtain

$$\|(X_kY_k)^{\frac{1}{2}}p_k^C\|_2^2 + \|(X_kY_k)^{\frac{1}{2}}q_k^C\|_2^2 = (\tfrac{1}{n}x_k^Ty_k)^2e^T(X_kY_k)^{-1}e;$$

or equivalently after dividing both sides by $\frac{1}{n}x_k^Ty_k$,

$$
\|T_k^{-\frac{1}{2}}p_k^C\|_2^2 + \|T_k^{-\frac{1}{2}}q_k^C\|_2^2 = e^TT_ke, \tag{36}
$$

where $T_k = \frac{1}{n}x_k^Ty_k(X_kY_k)^{-1}$ is a diagonal matrix. Our assumption implies that the maximum diagonal element of $\{T_k\}$ is bounded above by $1/\gamma$ and the minimum diagonal element of $\{T_k^{-1}\}$ is bounded below by γ. Therefore, from (36) we have

$$|(p_k^C)^i| \le \sqrt{n}/\gamma \quad \text{and} \quad |(q_k^C)^i| \le \sqrt{n}/\gamma.$$

Using the same technique, we can prove that

$$|(p_k^N)^i| \le \sqrt{n/\gamma} \le \sqrt{n}/\gamma \quad \text{and} \quad |(q_k^N)^i| \le \sqrt{n/\gamma} \le \sqrt{n}/\gamma.$$

From the definition of ω_k and the above estimates, Lemma 4.1 follows immediately. □

We now state our procedure for choosing the centering parameter σ_k.

PROCEDURE 2 (centering parameter criterion). Given

$$
\sigma \in (0,1), \quad \rho^l = \frac{\gamma^2\sigma}{2n}, \quad \rho^u \ge \frac{\gamma^2\sigma}{n}: \tag{37}
$$

Step 1. Compute ω_k from (35).
Step 2. Compute $\rho_k^u = \min(\rho^u, \sigma/\omega_k)$.
Step 3. Choose $\rho_k \in [(\rho^l + \rho_k^u)/2, \rho_k^u]$.
Step 4. Let $\sigma_k = \rho_k\omega_k$.

Since $\sigma_k = \rho_k\omega_k$ and $\rho_k \in [\rho^l, \rho_k^u]$, we have $\sigma_k \in [\rho^l\omega_k, \rho_k^u\omega_k]$. In addition, we require that σ_k be greater than the midpoint of the interval. This requirement is needed in our proof of superlinear convergence. It is evident that σ_k is bounded away from one because $\sigma_k \le \sigma < 1$. The reasons why the centering parameter is so chosen will hopefully become clear as our discussion proceeds.

5. Algorithm description. Now we formally state our primal-dual interior-point algorithm.

ALGORITHM 2. Suppose we are given a strictly feasible pair (x_0, y_0). Choose positive constants γ and Γ such that (see (28))

$$0 < \gamma \leq \min(1/2, f_0^{\min}/f_0^{\text{ave}}), \qquad \max(2, f_0^{\max}/f_0^{\text{ave}}) \leq \Gamma < n,$$

and choose $\sigma \in (0,1)$. Set $\rho^l = \gamma^2\sigma/2n$ and $\rho^u \geq \gamma^2\sigma/n$ (see (37)). For $k = 0, 1, 2, \dots,$ do:

Step 1. Compute the Newton step and the centering step from Algorithm 1.
Step 2. Choose σ_k by Procedure 2 and form $(\Delta x_k, \Delta y_k)$ from Algorithm 1.
Step 3. Choose α_k by Procedure 1.
Step 4. Form (x_{k+1}, y_{k+1}) from Algorithm 1.

The procedure for determining the step length α_k can be implemented in an effective manner. Its cost is somewhat higher than the ratio test that is used in most of the practical implementations. On the other hand, our procedure for choosing the centering parameter σ_k requires extra work when compared to the more standard method. The standard practice is to choose the centering parameter prior to computing the steps; then we only need to solve once for the combined step (Newton step plus the centering parameter times the centering step). Since Algorithm 2 requires the information obtained from both the Newton step and the centering step to choose the centering parameter, it requires us to solve for the two steps separately and then combine them.

6. Global linear convergence. The global linear convergence of Algorithm 2 is given in the following theorem.

THEOREM 6.1 (global linear convergence). *Let $\{(x_k, y_k)\}$ be generated by Algorithm 2. Then*

$$x_{k+1}^T y_{k+1} \leq (1 - \delta/n) x_k^T y_k$$

for some δ satisfying

$$\delta \geq \frac{\sigma(1-\sigma)\gamma^2}{16\Gamma}.$$

Proof. We need to estimate $\|s_k\|_\infty$ in (30). Let the index j be such that $\|s_k\|_\infty = |s_k^j|$. Observe that

$$\begin{aligned}\|s_k\|_\infty &= |\Delta x_k^j \Delta y_k^j| = |(x_k^j p_k^j)(y_k^j q_k^j)| = |(x_k^j y_k^j)(p_k^j q_k^j)| \\ &\leq \max(X_k Y_k e)\|\text{diag}(p_k) q_k\|_\infty \\ &= f_k^{\max}\|\text{diag}(p_k^N + \sigma_k p_k^C)(q_k^N + \sigma_k p_k^C)\|_\infty \\ &\leq f_k^{\max}\omega_k(1+\sigma_k)^2 \\ &\leq 4 f_k^{\max}\omega_k.\end{aligned}$$

Hence it follows from (21), (30), and Procedure 2 that

$$\alpha_k \geq \min\left(1, \frac{\rho_k f_k^{\text{ave}}}{8 f_k^{\max}}\right) \geq \min\left(1, \frac{\rho_k}{8\Gamma}\right) \geq \frac{\rho^l}{8\Gamma}. \tag{38}$$

Substituting ρ^l (see (37)) into the above expression, we obtain

$$\alpha_k \geq \frac{\sigma\gamma^2}{16\Gamma n}.$$

The proof is completed by substituting the above inequality into (9) and noticing that $\sigma_k \leq \sigma$. □

The following corollary follows immediately from Theorem 6.1. By a standard argument, it leads to polynomiality assuming integral data.

COROLLARY 6.2. *Assume that a strictly feasible pair (x_0, y_0), constants γ and Γ, both independent of n, are chosen such that* (28) *is satisfied and $x_0^T y_0 \leq 2^{\nu L}$, where $L > 0$ and ν is a positive constant independent of n. Then in at most $O(nL)$ iterations, Algorithm* 2 *will produce (x_k, y_k) such that $x_k^T y_k \leq 2^{-L}$.*

Proof. From Theorem 6.1,

$$x_k^T y_k \leq (1 - \delta/n)^k x_0^T y_0 \leq (1 - \delta/n)^k 2^{\nu L}.$$

Let $(1 - \delta/n)^k 2^{\nu L} = 2^{-L}$ and take the natural logarithm of both sides. We have $k = -(\ln 2)(1 + \nu)L/\ln(1 - \delta/n)$. Observe that for $x \in (0, 1)$,

$$\ln(1 - x) = -\sum_{k=1}^{\infty} \frac{x^k}{k} < -x.$$

Therefore,

$$k \leq (\ln 2)(1 + \nu)L/(\delta/n) = O(nL).$$

This completes the proof. □

7. Quadratic convergence.

In this section, we apply Theorem 2.1 to establish that under strict complementarity and nondegeneracy assumptions our algorithm converges Q-quadratically. It can be shown that the nondegeneracy and strict complementarity assumptions at optimality imply the uniqueness of both primal and dual solutions. We have already established convergence of the duality gap sequence to zero in the preceding section. With the uniqueness, it can be shown that the convergence of the duality gap implies that of the iterates to the unique solution $(x_*, y_*) \geq 0$. What we must verify is assumption (iii) of Theorem 2.1; namely,

$$\sigma_k = O(x_k^T y_k) \quad \text{and} \quad \tau_k = 1 - O(x_k^T y_k).$$

Since $\tau_k = \alpha_k/\hat{\alpha}_k$, for the latter it suffices to show that

$$\hat{\alpha}_k \to 1 \quad \text{and} \quad \hat{\alpha}_k - \alpha_k = O(x_k^T y_k). \tag{39}$$

The following lemma will be useful. It is a slightly modified version of Lemma 3.2 in [11]. We refer interested readers to the original paper for its proof.

LEMMA 7.1 (see [11]). *Let (x_*, y_*) be a solution of problem* (3) *and let $\{(x_k, y_k)\}$ be generated by Algorithm* 2. *Let p_k^N, p_k^C, q_k^N, and q_k^C be defined by* (34). *Assume that*

(i) *strict complementarity holds at (x_*, y_*);*
(ii) *x_* is a nondegenerate vertex of* (1).

Then

$$p_k^N = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ \vdots \\ -1 \end{pmatrix} + O(x_k^T y_k), \qquad p_k^C = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{x_k^T y_k/n}{[X_k Y_k e]^{m+1}} \\ \vdots \\ \frac{x_k^T y_k/n}{[X_k Y_k e]^{n}} \end{pmatrix} + O(x_k^T y_k),$$

and

$$q_k^N = \begin{pmatrix} -1 \\ \vdots \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + O(x_k^T y_k), \qquad q_k^C = \begin{pmatrix} \frac{x_k^T y_k/n}{[X_k Y_k e]^{1}} \\ \vdots \\ \frac{x_k^T y_k/n}{[X_k Y_k e]^{m}} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + O(x_k^T y_k),$$

where the number of zeros is m in p_k^N and p_k^C, and $n - m$ in q_k^N and q_k^C.

Now we are ready to state and prove our quadratic convergence theorem.

THEOREM 7.2 (quadratic convergence). *Let (x_*, y_*) be a solution of problem* (3) *and let $\{(x_k, y_k)\}$ be generated by Algorithm* 2. *Assume that*

(i) *strict complementarity holds at (x_*, y_*);*

(ii) *x_* is a nondegenerate vertex of* (1);

(iii) *ρ^u is sufficiently large, e.g., $\rho^u \geq 16\Gamma$.*

Then $\{(x_k, y_k)\}$ converges to (x_*, y_*) Q-quadratically.

Proof. We first prove $\sigma_k = O(x_k^T y_k)$. Observe from Lemma 7.1 that for each index i either the "p" terms ($(p_k^N)^i$ and $(p_k^C)^i$) or the "q" terms ($(q_k^N)^i$ and $(q_k^C)^i$) are $O(x_k^T y_k)$ while the other terms are bounded. Thus the quantity ω_k (see its definition (35)) is $O(x_k^T y_k)$; so is σ_k because $\sigma_k \leq \rho^u \omega_k$.

Since $\omega_k \longrightarrow 0$, from the choice of ρ_k^u in Step 2 of Procedure 2 we have for k sufficiently large

$$\rho_k^u = \rho^u \quad \text{and} \quad \rho_k \geq \tfrac{1}{2}(\rho^l + \rho^u). \tag{40}$$

We observe that if ρ^u is sufficiently large, e.g., $\rho^u \geq 16\Gamma$ (i.e., σ_k is not forced to approach zero too quickly), then the step length α_k will eventually be equal to or greater than one, as can be seen from (38).

Since $\sigma_k = O(x_k^T y_k)$ and $(x_k^T y_k/n)/\min(X_k Y_k e)$ is bounded, the elements of p_k and q_k are either $O(x_k^T y_k)$ or $-1 + O(x_k^T y_k)$. Therefore,

$$\min(X_k^{-1}\Delta x_k, Y_k^{-1}\Delta y_k) = \min(p_k, q_k) = -1 + O(x_k^T y_k). \tag{41}$$

By examining the definition of $\hat{\alpha}_k$ in Step 3 of Algorithm 1, we see $\hat{\alpha}_k = 1 + O(x_k^T y_k)$. Consequently, for k sufficiently large we have

$$1 \leq \alpha_k < \hat{\alpha}_k = 1 + O(x_k^T y_k).$$

This implies (39) and completes the proof. □

8. Superlinear convergence. In this section, we apply Theorem 2.2 to establish Q-superlinear convergence of Algorithm 2 for general problems. We must show that assumption (iii) of Theorem 2.2 holds; i.e.,

$$\sigma_k \to 0 \quad \text{and} \quad \tau_k \to 1.$$

For the latter, it will suffice to show $\hat{\alpha}_k \to 1$ and $\hat{\alpha}_k - \alpha_k \to 0$. Without the nondegeneracy assumption, we can no longer use Lemma 7.1. For technical reasons, we must further restrict the choice of ρ_k.

Denote the length of the interval $[\rho^l, \rho_k^u]$ by π_k. It follows from (37), Step 2 of Procedure 2, and Lemma 4.1 that

$$\rho_k^u \geq \frac{\gamma^2 \sigma}{n}. \tag{42}$$

Thus

$$\pi_k \overset{\text{def}}{=} \rho_k^u - \rho^l \geq \frac{\gamma^2 \sigma}{2n} > 0. \tag{43}$$

Let Σ_k be the following set of $2n$ points

$$\Sigma_k = \{-(p_k^N)^i/(p_k^C)^i, -(q_k^N)^i/(q_k^C)^i, \ i = 1, 2, \ldots, n\}$$

and define the distance from σ to the set Σ_k as

$$\operatorname{dist}(\sigma, \Sigma_k) = \min\{|\sigma - \varsigma| : \varsigma \in \Sigma_k\}.$$

We choose σ_k according to Procedure 2 with the additional restriction that

$$\operatorname{dist}(\sigma_k, \Sigma_k) \geq \pi_k \omega_k/(8n + 4). \tag{44}$$

In other words, we require not only that

$$\sigma_k \in [0.5(\rho^l + \rho_k^u)\omega_k, \rho_k^u \omega_k], \tag{45}$$

but also that σ_k be bounded away from the set Σ_k by at least $\pi_k \omega_k/(8n + 4)$. Since $\{\pi_k\}$ is bounded away from zero, we see from (44) that $\{\operatorname{dist}(\sigma_k, \Sigma_k)\}$ is bounded away from zero if $\{\omega_k\}$ is bounded away from zero.

We introduce condition (44) to avoid the situation where $p_k^i = (p_k^N)^i + \sigma_k (p_k^C)^i$ (say) converges to zero but $(p_k^N)^i$ and $(p_k^C)^i$ do not. Although we believe that it is extremely unlikely for this situation to occur, we have not been able to rule it out.

LEMMA 8.1. *The set of σ_k's satisfying* (44) *and* (45) *is nonempty.*

Proof. The length of the interval in (45) is $\pi_k \omega_k/2$. Partition this interval into $2n + 1$ equal subintervals, each having length $\pi_k \omega_k/(4n + 2)$. If the interior of any one of the subintervals does not intersect Σ_k, then the midpoint of that subinterval will satisfy (44) and (45). Since Σ_k has only $2n$ points, it cannot intersect the interiors of all the $2n + 1$ subintervals. This proves the lemma. □

Now we are well equipped to prove our superlinear convergence theorem.

THEOREM 8.2 (superlinear convergence). *Let (x_*, y_*) be a solution of problem* (3) *and let $\{(x_k, y_k)\}$ be generated by Algorithm* 2 *with the restriction* (44) *on the centering parameter σ_k. Assume that*

(i) *strict complementarity holds at (x_*, y_*);*

(ii) ρ^u *is sufficiently large, e.g.,* $\rho^u \geq 16\Gamma$.
If $\{(x_k, y_k)\}$ *converges to* (x_*, y_*)*, then the duality gap sequence* $\{x_k^T y_k\}$ *converges to zero Q-superlinearly.*

Proof. We first prove $\sigma_k \to 0$. It suffices to show $\omega_k \to 0$. Let $x_*^i > 0$. Obviously,

$$1 = \lim_{k \to \infty} \frac{x_{k+1}^i}{x_k^i} = \lim_{k \to \infty} (1 + \alpha_k p_k^i).$$

This implies $p_k^i \to 0$, because $\{\alpha_k\}$ is bounded away from zero. On the other hand, if $x_*^i = 0$, then $y_*^i > 0$ by strict complementarity. The same argument, interchanging the roles of p_k^i and q_k^i, gives $q_k^i \to 0$. Therefore, for each index i, either

$$p_k^i = (p_k^N)^i + \sigma_k (p_k^C)^i \to 0 \quad \text{or} \quad q_k^i = (q_k^N)^i + \sigma_k (q_k^C)^i \to 0. \tag{46}$$

We will prove $\omega_k \to 0$ by contradiction. Suppose the opposite. Then there exists a subsequence $\{\omega_{k_0}\} \subset \{\omega_k\}$ that is bounded away from zero. This in turn implies, from (44), that $\{\text{dist}(\sigma_{k_0}, \Sigma_{k_0})\}$ is bounded away from zero (recall that π_k is bounded away from zero).

We have shown that for each index i, either $p_k^i \to 0$ or $q_k^i \to 0$. Without loss of generality, assume $p_k^i \to 0$. We now show that $\{(p_{k_0}^C)^i\}$ converges to zero. Otherwise, there exists a subsequence $\{(p_{k_1}^C)^i\} \subset \{(p_{k_0}^C)^i\}$ such that $\{|(p_{k_1}^C)^i|\}$ is bounded away from zero. For this subsequence,

$$p_{k_1}^i = (p_{k_1}^N)^i + \sigma_{k_1} (p_{k_1}^C)^i = (p_{k_1}^C)^i [\sigma_{k_1} + (p_{k_1}^N)^i / (p_{k_1}^C)^i] \to 0.$$

This implies

$$\sigma_{k_1} + (p_{k_1}^N)^i / (p_{k_1}^C)^i \to 0.$$

However, this cannot be true because $\{\text{dist}(\sigma_{k_1}, \Sigma_{k_1})\} \subset \{\text{dist}(\sigma_{k_0}, \Sigma_{k_0})\}$ is bounded away from zero. Hence $(p_{k_0}^C)^i \to 0$.

Now in view of (46) we also have $(p_{k_0}^N)^i \to 0$. Similarly, we can prove that if $q_k^j \to 0$, then we have both $(q_{k_0}^N)^j \to 0$ and $(q_{k_0}^C)^j \to 0$. Therefore, for each index i, either $(p_{k_0}^N)^i$ and $(p_{k_0}^C)^i$, or $(q_{k_0}^N)^i$ and $(q_{k_0}^C)^i$ converge to zero. Since all these sequences are uniformly bounded (see the proof of Lemma 4.1), this leads to $\omega_{k_0} \to 0$ (see definition (35)), contradicting the hypothesis that $\{\omega_{k_0}\}$ is bounded away from zero. This proves that $\omega_k \to 0$. Consequently, $\sigma_k \to 0$.

Now we prove $\alpha_k \to 1$. Note that (7) can be written as

$$p_k + q_k = -e + \sigma_k \tfrac{1}{n} x_k^T y_k (X_k Y_k)^{-1} e.$$

Since $\frac{1}{n} x_k^T y_k (X_k Y_k)^{-1} e$ is bounded above by $1/\gamma$, as $\sigma_k \to 0$, we have

$$p_k + q_k \to -e.$$

We have shown that for each i, either $p_k^i \to 0$ or $q_k^i \to 0$. Therefore, all p_k^i and q_k^i converge to either 0 or -1. This again implies that $\hat{\alpha}_k \to 1$ (see (41)). In view of (38) and (40), α_k will eventually be equal to or greater than one if ρ^u is sufficiently large, e.g., $\rho^u \geq 16\Gamma$. Hence

$$1 \leq \alpha_k \leq \hat{\alpha}_k \to 1.$$

This completes the proof. □

9. Concluding remarks. In this paper, we have shown that the two fundamental parameters in primal-dual interior-point algorithms for linear programming can be chosen in such a way that both polynomiality and superlinear convergence are achieved. If the solution is a nondegenerate vertex, then in addition to superlinear convergence, we have quadratic convergence.

The current practices in some of the state-of-the-art implementations of primal-dual interior-point algorithms have the following common fundamental features. First, they allow iterates to be very close to the boundary of the positive orthant; second, they phase out the centering steps at a fast pace. The theory established in Zhang, Tapia, and Dennis [11] has already provided theoretical justification for such a practice from the viewpoint of fast convergence. This paper provides further theoretical justification for such a practice from the viewpoint of polynomiality. In summary, we can indeed, under reasonable conditions, accomplish both objectives: good global behavior and good local behavior.

We recently learned of a new result by Güler and Ye [4]. When applied to linear programming, it says that condition (11) will guarantee strict complementarity for any limit point of the iteration sequence generated by an interior-point algorithm. This result nicely complements the Zhang–Tapia–Dennis theory (i.e., Theorems 2.1 and 2.2) and, therefore, the strict complementarity assumptions in Theorems 7.2 and 8.2 are no longer necessary.

Acknowledgments. We gratefully acknowledge Florian Potra and Michael Todd for their useful comments on an earlier version of this paper. We also thank Yinyu Ye and two anonymous referees for their constructive suggestions.

REFERENCES

[1] I. C. Choi, C. L. Monma, and D. F. Shanno, *Further development of a primal-dual interior point method for linear programming*, ORSA J. Comput., 2 (1990), pp. 304–311.

[2] M. Iri and H. Imai, *A multiplicative barrier function method for linear programming*, Algorithmica, 1 (1986), pp. 455–482.

[3] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior point method for linear programming*, in Progress in Mathematical Programming, Interior-point and Related Methods, Nimrod Megiddo, ed., Springer-Verlag, New York, 1989, pp. 29–47.

[4] O. Güler and Y. Ye, *Convergence behavior of some interior-point algorithms*, Working paper series No. 91-4, College of Business Administration, Univ. of Iowa, Iowa City, IA, 1991.

[5] I. J. Lustig, R. E. Marsten, and D. F. Shanno, *Computational experience with a primal-dual interior point method for linear programming*, J. Linear Algebra Appl., 152 (1991), pp. 191–222.

[6] K. A. McShane, C. L. Monma, and D. F. Shanno, *An implementation of a primal-dual interior point method for linear programming*, ORSA J. Comput., 1 (1989), pp. 70–83.

[7] R. C. Monteiro and I. Adler, *Interior path-following primal-dual algorithms. Part* I: *Linear programming*, Math. Programming, 44 (1989), pp. 27–41.

[8] S. Mizuno, M. J. Todd, and Y. Ye, *On adaptive-step primal-dual interior-point algorithms for linear programming*, Tech. Rep. No. 944, School of Operations Research and Industrial Engineering, Cornell Univ., Ithaca, New York, 1990; Math. Oper. Res., to appear.

[9] M. J. Todd and Y. Ye, *A centered projective algorithm for linear programming*, Math. Oper. Res., 15 (1990), pp. 508–529.

[10] H. Yamashita, *A polynomially and quadratically convergent method for linear programming*, Report, Mathematical Systems Institute, Inc., Tokyo, Japan, 1986.

[11] Y. Zhang, R. A. Tapia, and J. E. Dennis, *On the superlinear and quadratic convergence of primal-dual interior-point linear programming algorithms*, SIAM J. Optimization, 2 (1992), pp. 304–324.

SIAM J. OPTIMIZATION
Vol. 3, No. 1, pp. 134–154, February 1993

NUMERICAL CONTINUATION AND SINGULARITY DETECTION METHODS FOR PARAMETRIC NONLINEAR PROGRAMMING*

BRUCE N. LUNDBERG[†] AND AUBREY B. POORE[‡]

Abstract. Numerical methods are developed for continuation, solution-type determination, and singularity detection in the parametric nonlinear programming problem. This problem is first converted to a closed, "active set" system of equations $\bar{F}(z, \alpha) = 0$, which includes a nonstandard normalization of the multipliers. A framework is then developed for combining various numerical continuation methods with a large number of null and range space methods from constrained optimization. By exploiting the special structure in the parametric optimization problem, solution-type classification and singularity detection are shown to require minimal additional expense beyond that involved in the continuation procedure itself. Due to the special structure of these problems, singularity detection methods are more comprehensive than those for general nonlinear equations. In this development, the Schur complement and related results play an important and unifying role. As an illustration, these methods are used to produce a "global" parametric analysis for a model problem from design optimization. This example exhibits an extensive number of solution paths, each of the basic types of singularities, multiple optima, regions of sensitivity, and jump phenomena.

Key words. active set system, numerical continuation, bifurcation, singularities, parametric optimization, Schur complement, null and range space methods

AMS(MOS) subject classifications. 65H10, 65K05, 90C31

1. Introduction.

The parametric nonlinear programming problem is that of determining the behavior of solution(s) as a parameter or vector of parameters $\alpha \in \mathbb{R}^r$ varies over a region of interest for the problem

$$\text{Minimize } \{f(x,\alpha) : \ c_E(x,\alpha) = 0, \ c_I(x,\alpha) \leq 0\}, \tag{1.1}$$

where $f : \mathbb{R}^{n+r} \rightarrow \mathbb{R}$, $c_E : \mathbb{R}^{n+r} \rightarrow \mathbb{R}^q$, and $c_I : \mathbb{R}^{n+r} \rightarrow \mathbb{R}^p$ are assumed to be at least twice continuously differentiable. Some of these parameters may be fixed but not precisely known and others may be varied or treated as control parameters. At a regular point for this system, we may use the implicit function theorem to rigorously justify the computation of the derivatives of the primal and dual variables with respect to the parameter α. These derivatives provide the basis for local sensitivity analysis as presented in the work of Fiacco [5], [6] and references therein. Many authors [2], [12]–[17], [27], [33] have used bifurcation and singularity theory to investigate the local behavior and persistence of minima at the singular points of (1.1), which are characterized by a loss of strict complementarity, a violation of the linear independence constraint qualification, or the singularity of the Hessian of the Lagrangian on the tangent space to the active constraints. The importance of these singularities is that they define the stability boundaries where a minimizer may be lost and where catastrophic failure, extreme sensitivity, and jumps to undesirable operating states can occur.

*Received by the editors August 30, 1991; accepted for publication (in revised form) December 6, 1991. This work was partially supported by Air Force Office of Scientific Research grants AFOSR-88-0059 and AFOSR-91-0138.

[†]Department of Mathematics, Grand Canyon University, 3300 W. Camelback, Phoenix, Arizona 85017.

[‡]Department of Mathematics, Colorado State University, Fort Collins, Colorado 80523.

Similar theoretical investigations have occurred within the fields of general nonlinear equations, dynamical systems, and for certain types of partial differential equations [10], [11], [23]. Numerical continuation and bifurcation techniques have been extensively and systematically developed for these latter fields and have played an integral part in investigating various phenomena [3], [4], [19]–[24], [31], [32]. Indeed, these methods should be equally helpful in the large areas of parametric nonlinear programming, abstract optimization, and control; however, only recently have these methods begun to appear in parametric nonlinear programming and in various applications [12], [25], [28]–[30]. Thus the overall objective in this work is to combine the analysis of the singular points in parametric nonlinear programming [14]–[17], [27], [33], numerical linear algebra methods from constrained optimization, and predictor-corrector continuation techniques to produce a collection of numerical methods specifically tailored to the parametric nonlinear programming problem. It is the utilization and modifications of the numerical methods from constrained optimization and the emphasis on numerical critical point classification and numerical singularity detection that differentiate this work from that of other authors [12], [28]–[30]. Numerical methods for branch switching, fold following, and singularity unfoldings will be treated in future work.

Since numerical continuation procedures are designed to follow solution paths of parameterized systems of nonlinear equations, the parametric nonlinear programming problem is first converted to a closed system of equations $F(z, \alpha) = 0$, which contains the complementarity slackness conditions and a nonstandard normalization of the multipliers [27]. For numerical purposes these equations are then converted to the "active set" system used by Lundberg and Poore in an earlier work [25]. These features, along with a brief discussion of the singular points, are presented in §2. For the single parameter problem ($r = 1$ in (1.1)), §3 describes a general class of predictor-corrector continuation schemes tailored specifically to the active set system for the parametric nonlinear programming problem. The solution of the linear systems arising in the continuation procedures is based on the bordering algorithm introduced by Keller [19]–[21]. This bordering algorithm, along with the modifications by Keller [20] and Chan [3], [4], allows us to exploit the large number of null and range space methods developed for constrained optimization, even when the systems are ill conditioned. To efficiently present numerical methods for determining critical point-type and singularity detection in §§5 and 6, respectively, we briefly review these null and range space methods in §4 along with the necessary modifications required on nonoptimal solution paths. In this development, the Schur complement and related ideas from linear algebra will play an important and unifying role.

The numerical determination of the solution type as an inexpensive by-product of the modified null or range space methods used in the continuation procedure is developed in §5. The special structure of the parametric programming problem is used in §6 to derive singularity detection tests that are more comprehensive and efficient than those for general nonlinear equations. In fact, the detection of singularities due to the loss of strict complementarity or the violation of the linear independence constraint qualification is shown to be immediately available from the computation of the solution points. The detection of a singularity of the Hessian of the Lagrangian on the tangent space to the active constraints is based on the inertia of the reduced Hessian, and we show how this may be computed at little additional expense when using either range or null space methods.

In §7, we illustrate these methods with a simple model problem arising from design optimization [30]. For this parametric nonlinear programming problem we produce a "global" analysis of sensitivity, stability, and multiplicity of minima which exhibits an extensive number of solution paths, each of the basic types of singularities, and jump phenomena arising from a loss of the linear independence constraint qualification.

2. Singularities and formulation of the active set system. The first objective is to convert the parametric nonlinear programming problem to a closed system of nonlinear equations whose solutions contain all local minima as well as saddle points, local maxima, and feasible and infeasible solutions. Following a characterization of the singular points in this system, an equivalent *active set* system [25] for the numerical continuation process will be presented.

The following notation will be needed. For a function $f : \mathbb{R}^{n+r} \to \mathbb{R}^1$, the gradient of $f(x,\alpha)$ with respect to $x \in \mathbb{R}^n$ will be a column vector denoted by $\nabla_x f(x,\alpha)$. The differential operator

$$D_x = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_n}\right)$$

denotes a row operator whose transpose is $D_x^T = \nabla_x$. Thus $D_x f(x,\alpha)$ is a row vector, $D_x^T f(x,\alpha) \equiv [D_x f(x,\alpha)]^T = \nabla_x f(x,\alpha)$ is a column vector, and $\nabla_x^2 f(x,\alpha) = D_x(\nabla_x f(x,\alpha)) = D_x[D_x^T f(x,\alpha)]$ denotes the Hessian of f. Also, if $F : \mathbb{R}^{n+r} \to \mathbb{R}^m$, then $D_x F(x,\alpha)$ is an m-by-n matrix whose element in the ith row and jth column is $\partial F_i(x,\alpha)/\partial x_j$.

Given the parametric nonlinear programming problem

$$\text{Minimize } \{f(x,\alpha) \mid c_i(x,\alpha) = 0 \text{ for } i \in E \; c_i(x,\alpha) \le 0 \text{ for } i \in I\} \tag{2.1}$$

where $E = \{1,\dots,p\}$ and $I = \{p+1,\dots,p+q\}$ represent the index sets for the equality and inequality constraints, respectively, the Fritz John first-order necessary conditions are that there exist $p+q+1$ real numbers in the scalar ν and the vector $\lambda = (\lambda_1,\dots,\lambda_p,\lambda_{p+1},\dots,\lambda_{p+q})$, not all zero, such that

$$\nabla_x \mathcal{L}(x,\lambda,\nu,\alpha) = 0, \tag{2.2}$$

$$\Lambda c(x,\alpha) = 0, \tag{2.3}$$

$$c_i(x,\alpha) \le 0 \quad \text{for } i \in I, \tag{2.4}$$

$$\lambda_i \ge 0 \quad \text{for } i \in I, \; \nu \ge 0, \tag{2.5}$$

where $\mathcal{L} = \mathcal{L}(x,\lambda,\nu,\alpha) = \nu f(x,\alpha) + \sum_{i=1}^{p+q} \lambda_i c_i(x,\alpha)$ is the Lagrangian and Λ is a diagonal matrix with $\Lambda_{ii} = 1$ for $i \in E$ and $\Lambda_{ii} = \lambda_i$ for $i \in I$. Observe that equations (2.2) and (2.3) represent $n+p+q$ equations in the $n+p+q+1$ unknowns, $x \in \mathbb{R}^n, \lambda \in \mathbb{R}^{p+q}$, and $\nu \in \mathbb{R}$. The usual normalization is to choose $\nu = \nu_0 > 0$; however, this can lead to infinite multipliers when the linear independence constraint qualification is violated. To resolve this difficulty, the normalization $\nu^2 + \lambda^T\lambda - \beta_0^2 = 0$, where β_0 is a fixed positive real number, is included with the equations (2.2) and (2.3) to obtain the closed system

$$F(x,\lambda,\nu,\alpha) = \begin{bmatrix} \nabla_x \mathcal{L}(x,\lambda,\nu,\alpha) \\ \Lambda c(x,\alpha) \\ \nu^2 + \lambda^T\lambda - \beta_0^2 \end{bmatrix} = 0. \tag{2.6}$$

In the sequel, the variable z will be used to denote the $n+p+q+1$ variables (x,λ,ν), i.e., $z \equiv (x,\lambda,\nu)$.

The next theorem gives necessary and sufficient conditions for $D_zF(z_0, \alpha_0)$ to be singular at a solution of $F(z, \alpha) = 0$. This requires the concept of an eigenvalue on a tangent space to the active constraints: Let $L : \mathbb{R}^m \to \mathbb{R}^m$ be a linear operator and let V denote a k-dimensional subspace of $\mathbb{R}^m$. The restriction of L to V is denoted by L_V and defined on V as PL where P is an orthogonal projection of $\mathbb{R}^m$ onto V. A scalar λ is an eigenvalue of L_V provided there exists a nonzero vector $y \in V$ such that $L_V y = \lambda y$. Thus we say that L is singular on the subspace V provided zero is an eigenvalue of L_V. If L also denotes the matrix representation of the operator L and the columns of a matrix $Z \in \mathbb{R}^{m\times k}$ form an orthonormal basis for V, then $P = ZZ^T$ is such a projection and the eigenvalues of L_V are those of the matrix Z^TLZ, which are invariant under changes in Z as long as the columns form an orthonormal basis for V.

THEOREM 2.1 (see [27]). *Let $(z_0, \alpha_0) = (x_0, \lambda_0, \nu_0, \alpha_0)$ be a solution of $F(z, \alpha) = 0$, i.e., a solution of equation* (2.6), *which combines* (2.2), (2.3), *and the normalization $\nu^2 + \lambda^T\lambda - \beta_0^2 = 0$. Assume that f and c are twice continuously differentiable in a neighborhood of (x_0, α_0) and define two index sets $\mathcal{A}$ and $\bar{\mathcal{A}}$ and a corresponding tangent space T by*

$$\begin{aligned}\mathcal{A} &= E \cup \{i \in I : c_i(x_0, \alpha_0) = 0\}, \qquad \bar{\mathcal{A}} = E \cup \{i \in \mathcal{A} \cap I : \lambda_i^0 \neq 0\},\\ T &= \{y \in \mathbb{R}^n : D_xc_i(x_0, \alpha_0)y = 0 \text{ for all } i \in \mathcal{A}\}.\end{aligned} \tag{2.7}$$

Then a necessary and sufficient condition that $D_zF(z_0, \alpha_0)$ be nonsingular is that each of the following three conditions hold:

(i) $\bar{\mathcal{A}} = \mathcal{A}$;

(ii) $S := \{\nabla_x c_i(x_0, \alpha_0)\}_{i\in\mathcal{A}}$ *is a linearly independent 'collection of $|\mathcal{A}|$ vectors where $|\mathcal{A}|$ denotes the cardinality of $\mathcal{A}$*;

(iii) *The Hessian $\nabla_x^2\mathcal{L}(z_0, \alpha_0)$ of the Lagrangian is nonsingular on the tangent space T.*

If $D_zF(z_0, \alpha_0)$ is nonsingular, there exist neighborhoods $\mathcal{B}_1$ of $\alpha = \alpha_0$ and $\mathcal{B}_2$ of $z_0 = (x_0, \lambda_0, \nu_0)$ and a function $\phi \in C^1(\mathcal{B}_1)$ such that $F(\phi(\alpha), \alpha) = 0$ for all $\alpha \in \mathcal{B}_1$ and $\phi(\alpha_0) = z_0$. This solution is unique in the sense that if $z \in \mathcal{B}_2$ and $F(z, \alpha) = 0$, then (z, α) belongs to the manifold defined by ϕ, i.e., $z = \phi(\alpha)$. Furthermore, if f and c are $C^k (k \geq 2)$ (C^∞ or real analytic) then ϕ is C^{k-1} (C^∞ or real analytic, respectively, on $\mathcal{B}_1$).

The importance of conditions (i)–(iii) in Theorem 2.1 is that they provide a set of necessary and sufficient conditions for a singularity in the system $F = 0$, and thus an initial classification into which all singularities and bifurcation problems fit. The term *critical point* will refer to any solution of system (2.6), *regular point* will describe any solution of (2.6) for which conditions (i)–(iii) of Theorem 2.1 are valid, and the term *singular point* is reserved for any solution of (2.6) at which D_zF is singular, i.e., one or more of (i)–(iii) is violated. Since these singularities have been investigated theoretically by many authors [14]–[17], [27], [33], we will focus only on the numerical aspects.

Since a multiplier corresponding to an inactive constraint is zero, the system (2.6) can be reduced in complexity by using an *active set strategy.* The inactive constraints, i.e., those c_i for which $i \in I - \mathcal{A}$ are thus removed, yielding the active set system

$$\bar{F}(z, \alpha) = \begin{bmatrix}\nabla_x\mathcal{L}(z, \alpha)\\ \bar{c}(x, \alpha)\\ B(\lambda, \nu)\end{bmatrix} = 0 \quad \text{where} \quad z = \begin{bmatrix}x\\ \lambda\\ \nu\end{bmatrix} \in \mathbb{R}^m, \tag{2.8}$$

$m = n + |\mathcal{A}| + 1$, $\lambda = (\lambda_1, \ldots, \lambda_p, \lambda_{i \in \mathcal{A} \cap I})$, and $\bar{c} = (c_1, \ldots, c_p, c_{i, i \in \mathcal{A} \cap I})$, $\mathcal{L}(z, \alpha) = \nu f(x, \alpha) + \sum_{i \in \mathcal{A}} \lambda_i c_i(x, \alpha)$, and $B(\lambda, \nu) = \nu^2 + \lambda^T \lambda - \beta_0^2$. Continuation for the active set system (2.8), along with detecting zeros in one or more of the active, inequality multipliers λ_i, $i \in \mathcal{A} \cap I$, or in an inactive constraint c_i for $i \in I - \mathcal{A}$ and changing the active set appropriately, is then equivalent to continuation for the full system (2.6).

3. Numerical continuation methods.

The subject of general numerical continuation methods has a formidable literature, and excellent introductions can be found in Allgower and Georg [1], Keller [21], and Rheinboldt [32]. Thus we forego a survey of this area and concentrate on building a framework for combining numerical linear algebra methods in optimization with predictor-corrector continuation methods. (This combination should also augment the work of several authors [12], [28]–[30], who also use predictor-corrector methods for parametric optimization.) Schur complements and related results [26] will be used to rederive the bordering algorithm of Keller [19]–[21] in a form more suitable for singularity detection and continuation in the parametric nonlinear programming problem.

3.1. Bordered matrices for predictor-corrector continuation.

The notation $w = (z, \alpha)$ is convenient for a discussion of predictor-corrector continuation and will be used in this section. Assume that $\bar{F}(w)$ is continuously differentiable and $\bar{F}(w) = 0$ has a smooth solution path $P = \{w \in \mathbb{R}^{m+1} : w = \Psi(s), s \in I, \Psi \in C^1\}$ where I is an interval of real numbers. Most path-following algorithms generate a sequence $\{(w_k, s_k)\}_{k=0}^N$ where w_k is a point on or near the path and w_0 is a known solution of $\bar{F}(w) = 0$. To go from a point w_k to a point w_{k+1}, we first obtain a predicted point of the form $wp_{k+1} = w_k + \Delta s d(\Delta s)$ where the predictor direction d is typically the current oriented unit tangent T_k or a combination of this and previously computed tangents [24]. In either case $d(\Delta s)$ is continuous at $\Delta s = 0$ and $\lim_{\Delta s \to 0} d(\Delta s) = T_k$. The predicted point is then used as the initial approximation for a Newton-like correction iteration back to the path, terminating with a solution w_{k+1}. At each point w_k on the path, the tangent T_k is a solution to

$$[D_w \bar{F}(w_k)] T_k = 0. \tag{3.1}$$

The correction back to the path can be achieved in many ways [1], [21], [32], but the work described here is based on solving the augmented system

$$G(w) = \begin{bmatrix} \bar{F}(w) \\ N(w) \end{bmatrix} = 0 \quad \text{where} \quad N(w) = (w - wp_{k+1})^T d(\Delta s), \tag{3.2}$$

which confines the correction to a hyperplane orthogonal to the prediction direction $d(\Delta s)$ [21], [24]. If $\{w_{k+1}^i\}_{i \geq 0}$ denotes the Newton-like corrector iterates for (3.2) with $w_{k+1}^0 = wp_{k+1}$, then a correction step $\Delta w = w_{k+1}^{i+1} - w_{k+1}^i$ is computed by solving a linear system of the form

$$J \Delta w = -G(w_{k+1}^i), \tag{3.3}$$

where $J = D_w G(w_{k+1}^i)$ is the Jacobian of G or some approximation to it.

The primary linear algebra requirements in a predictor-corrector step are thus the computation of the tangent vector T_k in (3.1) and corrections Δw in (3.3). Central to this linear algebra is the Lagrangian matrix W defined by

$$W = \begin{bmatrix} H & A \\ A^T & 0 \end{bmatrix}, \tag{3.4}$$

where $H = \nabla_x^2 \mathcal{L}(z, \alpha)$ or some approximation to it, $A^T = D_x \bar{c}(x, \alpha)$, and $\bar{c}(x, \alpha)$ denotes the equality and active inequality constraints, as in (2.8). The reason for this is that the $(m+1) \times (m+1)$ Jacobian matrix $J = D_w G$ can be partitioned by

$$J = \begin{bmatrix} M & D_\alpha \bar{F} \\ d_z^T & d_\alpha \end{bmatrix} \quad \text{where} \quad M = D_z \bar{F}(z, \alpha) = \begin{bmatrix} W & & \nabla_x f \\ & & 0 \\ 0 & 2\lambda^T & 2\nu \end{bmatrix} \tag{3.5a}$$

and by

$$J = \begin{bmatrix} W & B \\ C^T & D \end{bmatrix} \tag{3.5b}$$

where

$$B = \begin{bmatrix} \nabla_x f & D_\alpha \nabla_x \mathcal{L} \\ 0 & D_\alpha \bar{c}(x, \alpha) \end{bmatrix}, \quad C^T = \begin{bmatrix} 0 & 2\lambda^T \\ d_x^T & d_\lambda^T \end{bmatrix}, \quad \text{and} \quad D = \begin{bmatrix} 2\nu & 0 \\ d_\nu & d_\alpha \end{bmatrix}. \tag{3.5c}$$

The matrices M, B, C^T, and D have dimensions $m \times m$, $(m-1) \times 2$, $2 \times (m-1)$, and 2×2, respectively, and $d_z^T = (d_x^T, d_\lambda^T, d_\nu)$ where d_x, d_λ, d_ν, and d_α denote the x, λ, ν, and α components of the prediction direction d, respectively. Note that the function $\bar{F}$ corresponds to the active set system (2.8) and that it is this Lagrangian matrix W that plays a central role in nonlinear constrained optimization [7], [8], [18].

3.2. The Schur complement and the bordering algorithm. The partitioning of the Jacobian J given in (3.5b) suggests a block elimination algorithm for solving the systems (3.3) and (3.1) that exploits the underlying structure of the Lagrangian matrix W. Schur complements will be used in this subsection to rederive the bordering algorithm of Keller [19]–[21] in a form more suitable for singularity detection and continuation in nonlinear parametric programming. The following two theorems will be used repeatedly in this and the next three sections and can be found in the survey on Schur complements by Ouellette [26].

THEOREM 3.1 (see [26]). *If L is a nonsingular matrix and $S = D - C^T L^{-1} B$ is the Schur complement of L in*

$$\Phi = \begin{bmatrix} L & B \\ C^T & D \end{bmatrix},$$

then

$$\det(\Phi) = \det(L) \cdot \det(S), \tag{3.6a}$$

where $\det(\cdot)$ *denotes the determinant. If Φ is also real symmetric, then*

$$\text{in}(\Phi) = \text{in}(L) + \text{in}(S), \tag{3.6b}$$

where $\text{in}(\cdot)$ *denotes the inertia (the number of positive, negative, and zero eigenvalues).*

THEOREM 3.2 (see [26]). *Suppose L and Φ in the previous theorem are nonsingular. Then the Schur complement S of L in Φ is nonsingular and*

$$\Phi^{-1} = \begin{bmatrix} L^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} L^{-1} B \\ -I \end{bmatrix} S^{-1} [C^T L^{-1}, -I] \tag{3.7a}$$

$$= \begin{bmatrix} L^{-1} + L^{-1} B S^{-1} C^T L^{-1} & -L^{-1} B S^{-1} \\ -S^{-1} C^T L^{-1} & S^{-1} \end{bmatrix}. \tag{3.7b}$$

With this background, a version of the bordering algorithm [20] for solving (3.3) can be briefly described as follows. Consider partitioning (3.5b) and let $\tilde{y}, \tilde{v}, \tilde{u} \in \mathbb{R}^{m-1}$ be solutions of

(3.8)
$$W\tilde{y} = -\begin{bmatrix} \nabla_x \mathcal{L}(z,\alpha) \\ \bar{c}(x,\alpha) \end{bmatrix}, \quad W\tilde{v} = -\begin{bmatrix} \nabla_x f(x,\alpha) \\ 0 \end{bmatrix}, \quad W\tilde{u} = -\frac{\partial}{\partial\alpha}\begin{bmatrix} \nabla_x \mathcal{L}(z,\alpha) \\ \bar{c}(x,\alpha) \end{bmatrix}.$$

Define ℓ, y, v, and $u \in \mathbb{R}^{m+1}$ by

$$\ell = \begin{pmatrix} 0 \\ 2\lambda \\ 2\nu \\ 0 \end{pmatrix}, \quad y = \begin{pmatrix} \tilde{y} \\ 0 \\ 0 \end{pmatrix}, \quad v = \begin{pmatrix} \tilde{v} \\ 1 \\ 0 \end{pmatrix}, \quad \text{and} \quad u = \begin{pmatrix} \tilde{u} \\ 0 \\ 1 \end{pmatrix}, \tag{3.9}$$

where the bottom two entries in these vectors are scalars and $\binom{0}{2\lambda} \in \mathbb{R}^{m-1}$. Then the Schur complement S_W of W in J is given by

$$S_W = \begin{bmatrix} \ell^T v & \ell^T u \\ d^T v & d^T u \end{bmatrix}, \tag{3.10}$$

so that the correction step in (3.3) is given by

$$\Delta w = y + sv + tu, \tag{3.11a}$$

where s and t solve the two-dimensional system

$$S_W \begin{bmatrix} s \\ t \end{bmatrix} = -\begin{bmatrix} B + \ell^T y \\ N + d^T y \end{bmatrix}. \tag{3.11b}$$

(This formula is the result of applying (3.7a) to equation (3.3) for the partitioning (3.5b).)

If the vectors in (3.9) are computed at w_{k+1}, then the tangent T_{k+1} is given by

$$T_{k+1} = \pm\left[(\ell^T v)u - (\ell^T u)v\right] / \|(\ell^T v)u - (\ell^T u)v\|_2. \tag{3.12}$$

The sign in this formula determines the orientation of the continuation and is typically chosen so that $T_k^T T_{k+1} > 0$ [21]. (This representation of the solution of (3.1) can be derived by forming $J^{-1}[{}^0_1]$ using (3.5b) and (3.7b) to obtain a scalar multiple of $[(\ell^T v)u - (\ell^T u)v]$ and then normalizing to obtain T_{k+1}.)

The above formulas are *theoretically* valid if W, S_W, and J are nonsingular. Facts regarding the nonsingularity of these matrices, together with the matrix $M = D_z\bar{F}$, are given in the following theorem and subsequent discussion.

THEOREM 3.3. *Let $w_{k+1} = (x_{k+1}, \lambda_{k+1}, \nu_{k+1}, \alpha_{k+1})$ be a solution of the active set system (2.8), assume the objective function $f(x,\alpha)$ and constraints $\bar{c}(x,\alpha)$ are twice continuously differentiable in a neighborhood of (x_{k+1}, α_{k+1}), and let W, J, M, S_W, and T_{k+1} be defined as in (3.4), (3.5), (3.10), and (3.12) with all derivatives being evaluated at w_{k+1}. Then the following are equivalent:*

(i) *$A = D_x\bar{c}(x_{k+1}, \alpha_{k+1})^T$ has full rank and $H = \nabla_x^2\mathcal{L}(w_{k+1})$ is nonsingular on $\mathcal{N}(A^T)$.*

(ii) *W is nonsingular.*

(iii) *M is nonsingular.*

Furthermore, if W is nonsingular and d_k is the prediction direction at w_k, then the following are equivalent:

(iv) *J is nonsingular.*

(v) *S_W is nonsingular.*

(vi) *$d_k^T T_{k+1}$ is nonzero.*

Proof. The equivalence of parts (i) and (ii) is shown in [18] and is a corollary of Theorem 5.1 in §5. The equivalence of (i) and (iii) follows as a special case of Theorem 2.1. Thus (ii) and (iii) are equivalent. If W is nonsingular, Theorem 3.1 implies the equivalence of (iv) and (v). Finally, $d_k^T T_{k+1} = \pm d_k^T\left[(\ell^T v)u - (\ell^T u)v\right]/\|(\ell^T v)u - (\ell^T u)v\|_2 = \pm\left[(\ell^T v)d_k^T u - (\ell^T u)d_k^T v\right]/\|(\ell^T v)u - (\ell^T u)v\|_2 = \pm\det(S_W)\|(\ell^T v)u - (\ell^T u)v\|_2^{-1}$, which implies the equivalence of (v) and (vi). □

Geometrically, condition (vi) states that the previous predictor direction is not orthogonal to the path $w(s)$ at w_{k+1}. If w_k is a regular point of the smooth solution path P and if $d_k = T_k$ or $d_k = d_k(\Delta s)$ with $\lim_{\Delta s\to 0} d_k(\Delta s) = T_k$, then $d_k^T T_{k+1} > 0$ holds at w_{k+1} for a sufficiently small stepsize Δs [24]. The corrector procedure developed by the authors [24] terminates when $d_k^T T_{k+1}$ becomes small, and the corrector is reinitiated with a smaller predictor step Δs. Hereafter, we will assume that $d_k^T T_{k+1} > 0$ holds at every continuation point w_{k+1}, so that the bordering formulas above are all *theoretically* valid as long as the Lagrangian W is nonsingular. Numerically, these formulas perform well as long as W is well conditioned.

3.3. Ill-conditioned Lagrangian matrices. An ill-conditioned Lagrangian matrix W can occur at or near a singularity or may occur all along a path of singularities arising, e.g., from fold following [20], [21]. In these cases the procedures described in the work of Keller [20], or in the generalized deflated block elimination algorithm of Chan [3], [4] can be used to solve (3.1) and (3.3). Both procedures allow us to exploit the structure of the Lagrangian matrix W, even when W is ill conditioned.

4. Linear algebra for the Lagrangian matrix.

The linear systems that arise in the continuation steps are of the form $J\Delta w = -G$ and must be solved for several different values of G. The bordering algorithm (3.8)–(3.12) applied to the active set system (2.8) reduces the linear algebra requirements in the continuation procedure to the solution of systems of the form

$$W\begin{bmatrix}\Delta x\\ \Delta\lambda\end{bmatrix} = \begin{bmatrix}g\\ b\end{bmatrix} \quad\text{where}\quad W = \begin{bmatrix}H & A\\ A^T & 0\end{bmatrix}. \tag{4.1}$$

This section contains a brief review of the direct methods for solving systems of this form, since this allows an efficient presentation of the determination of critical point-type and singularity detection in the next two sections. Three classes of methods for solving these linear systems in constrained optimization [7], [8] are the symmetric factorization, null space or generalized elimination methods, and range space methods. It is important to stress that the formulas to be presented involve the inverses of certain matrices and that these formulas are not used directly in computation. Instead, when an inverse is required, a factorization is computed and the computations are rearranged to simplify the operations. We do not discuss the various iterative methods.

Since the Lagrangian matrix W is generally symmetric indefinite, a *symmetric factorization* algorithm such as either the Bunch–Parlett or Bunch–Kaufman algorithm [9] can be used to produce the factorization $PWP^T = LDL^T$ where P is a permutation matrix, L is unit lower triangular, and D is block diagonal with 1×1 and symmetric 2×2 blocks.

For *range space* methods, both W and H are assumed to be nonsingular. Then the Schur complement of H in W is

$$S = -A^T H^{-1} A, \tag{4.2}$$

which is also nonsingular by Theorem 3.2. Using (3.7b), the inverse of W can be expressed in the form

$$W^{-1} = \begin{bmatrix} K & T \\ T^T & U \end{bmatrix}, \tag{4.3a}$$

where

$$K = H^{-1} + H^{-1} A S^{-1} A^T H^{-1}, \tag{4.3b}$$
$$T = -H^{-1} A S^{-1}, \tag{4.3c}$$
$$U = S^{-1}. \tag{4.3d}$$

Range space methods, which are recommended for problems with few constraints [7], [8], make use of this representation of W^{-1} to solve (4.1). As an example, suppose the Bunch–Kaufman algorithm [9] is used to factor an indefinite H by $PHP^T = LDL^T$, so that $H^{-1} = P^T L^{-T} D^{-1} L^{-1} P$ and $A^T H^{-1} A = (L^{-1}PA)^T D^{-1} (L^{-1}PA)$. Next let

$$L^{-1}PA = QR = [Q_1 : Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1$$

be the QR factorization of $L^{-1}PA$. Then

$$S = -(A^T H^{-1} A)^{-1} = -R_1^{-1} (Q_1^T D^{-1} Q_1)^{-1} R_1^{-T}$$

and we must factor the expression $Q_1^T D^{-1} Q_1$ to complete the computation; however, this has small dimension when the number of active constraints (row size of A^T) is small. Finally, range space methods can also be viewed as a form of Keller's bordering algorithm applied to the system (4.1), which suggests that Chan's deflation algorithm [4] can be applied in case H is ill conditioned.

Null space methods, which are recommended for problems with many constraints [7], [8], may be described by constructing matrices $Y \in \mathbb{R}^{n \times a}$ and $Z \in \mathbb{R}^{n \times (n-a)}$ with the properties that $[Y : Z]$ is nonsingular, $A^T Y = I$, and $A^T Z = 0$. A general scheme for such a construction is to choose an $(n) \times (n-a)$ matrix V such that $[A : V]$ is nonsingular. Then

$$[A : V]^{-1} = \begin{bmatrix} Y^T \\ Z^T \end{bmatrix}.$$

By using these to solve (4.1), we obtain an alternate representation of K, T, and U in (4.3a) [7]:

$$K = Z(Z^T H Z)^{-1} Z^T, \tag{4.4a}$$
$$T = Y - Z(Z^T H Z)^{-1} Z^T H Y, \tag{4.4b}$$
$$U = Y^T H Z (Z^T H Z)^{-1} Z^T H Y - Y^T H Y. \tag{4.4c}$$

The matrix $Z^T H Z$ must be factored once Y and Z are chosen. In numerical constrained optimization, we generally assume that $Z^T H Z$ is positive definite as part of a second-order sufficient condition at a local minimizer; however, local minima, saddle points, and maxima are encountered as the solution paths of $\bar{F}(x, \alpha) = 0$ are traced during the course of the continuation procedure. Thus $Z^T H Z$ may now be positive definite, indefinite, or negative definite on some portions of the path. Thus we must again resort to factorizations for symmetric indefinite matrices such as that of Bunch and Kaufman [9]. As an example, the popular null space method based on a QR factorization

$$A = [Q_1 : Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

gives $Z = Q_2$ and $Y = Q_1 R_1^{-T}$.

5. Critical point type. The objective in this section is to explain how critical point type can be efficiently determined as an inexpensive by-product for the linear algebra methods presented in the previous section. Recall that a *regular point* for the nonlinear programming problem is a solution of the first-order necessary conditions (2.2) and (2.3) at which strict complementarity holds, the linear independence constraint qualification is valid, and the Hessian of the Lagrangian on the tangent space to the active constraints is nonsingular. Such regular points can be classified using:

(5.1a) sign ν,

(5.1b) signs of $c_i(x, \alpha)$ for $i \in I - \mathcal{A}$ and λ_i for $i \in I \cap \mathcal{A}$,

(5.1c) signs of the eigenvalues of $\nabla_x^2 \mathcal{L}_T$,

where $\mathcal{A}$ denotes the active set, I, the inequality constraints, and $\nabla_x^2 \mathcal{L}_T$, the restriction of the Hessian of the Lagrangian to the tangent space of the active constraints $\mathcal{N}(A^T)$. Since these signs can change only at a singularity in system (2.6), they are used for both solution-type classification and singular point detection. The first two sets of signs in (5.1a) and (5.1b) are determined from the solution by inspection; only part (5.1c) requires any additional computation. The following adaptation of a result of Jongen, Möbert, Rückmann, and Tammer [18] yields an efficient means of computing the inertia of $\nabla_x^2 \mathcal{L}_T$ for any of the linear algebra methods discussed in the previous section.

THEOREM 5.1 (see [18]). *Let $a = |\mathcal{A}|$ denote the number of active constraints in* (2.8) *and let W be a symmetric matrix of the form* (3.4), *where $H \in \mathbb{R}^{n \times n}$ and $A \in \mathbb{R}^{n \times a}$. The restriction of the map H to the null space $T = \mathcal{N}(A^T)$ is denoted by H_T. If A has rank k,*

$$\text{in}(W) = \text{in}(H_T) + (k, k, a - k), \tag{5.2}$$

so that when A has full rank,

$$\text{in}(H_T) = \text{in}(W) - (a, a, 0). \tag{5.3}$$

For a *symmetric factorization* of W, $\text{in}(H_T)$ can be obtained directly from $\text{in}(W)$ via (5.3) and this factorization. For example, if we have a factorization $PWP^T = LDL^T$, then $\text{in}(W) = \text{in}(D)$ by Sylvester's law of inertia, and the inertia of D is simply the sum of the inertias of the 1×1 and 2×2 diagonal blocks of D.

The computation of $\text{in}(H_T)$ for a *range space method* can be based on the formula $\text{in}(H_T) = \text{in}(H) + \text{in}(S) - (a, a, 0)$, which is obtained from equations (3.6b) and (5.3). The inertias of the symmetric matrices H and $S = -A^T H^{-1} A$ are easily obtained from their factorizations when using a range space method for (4.1). Continuing with the example following equation (4.3d), $\text{in}(H) = \text{in}(D)$ when using the Bunch–Kaufman factorization. Since $Q_1^T D^{-1} Q_1$ is to be factored as part of the range space method, $\text{in}(S) = \text{in}(-Q_1^T D^{-1} Q_1)$ is inexpensively computed as a by-product of this factorization.

In the *null space method* the $\text{in}(H_T)$ is even more easily computed since $\text{in}(H_T) = \text{in}(Z^T H Z)$ and $Z^T H Z$ must be factored as part of this method. If, for example, a symmetric factorization $Z^T H Z = P^T L D L^T P$ is employed, then $\text{in}(H_T) = \text{in}(Z^T H Z) = \text{in}(D)$ where $\text{in}(D)$ is, again, the sum of the inertias of the 1×1 and symmetric 2×2 diagonal blocks of D.

6. Singularity detection. A singular point of the system $F(w) = 0$ is a solution $w = (z, \alpha)$ at which $D_z F(z, \alpha)$ is singular. Let $P = \{w(s) = (z(s), \alpha(s)) : s_a < s < s_b,\ w \in C^1(s_a, s_b)\}$ be a smooth solution path of the system $F(w) = 0$. Most continuation codes are designed to step over singular points, detect their presence, and then either continue along the path, switch branches, or reverse the orientation as required. A popular singularity detection scheme [1], [21], [32] is based on detecting changes in

$$\text{sign} \det(D_z F(w)), \tag{6.1a}$$

$$\text{sign} \det(D_w G(w)) \tag{6.1b}$$

along the path P where $G(w) = \left[\begin{smallmatrix} F(w) \\ N(w) \end{smallmatrix}\right] = 0$ is the corrector system (3.2) and $N(w)$ is a normalization equation. Many such normalizations from the literature [1], [21], [24], [32] can be put in the form $N(w) = (w - wp_{k+1})^T d(s)$ and the direction $d(s)$ can be the prediction direction, e.g., Euler, secant, or higher-order predictor [21], [24], or a standard unit basis vector e_i near the tangent direction for the parameter switching techniques [32].

We next define the terms simple fold, simple quadratic fold, bifurcation, and simple bifurcation, which will be needed in the next theorem. Let $F : \mathbb{R}^{m+1} \to \mathbb{R}^m$ be a smooth mapping and $F(z_0, \alpha_0) = 0$. As defined in §2, (z_0, α_0) is a *regular point* if $D_z F(z_0, \alpha_0)$ is nonsingular. If the rank of $[D_z F(z_0, \alpha_0) : D_\alpha F(z_0, \alpha_0)]$ is m, but $D_z F(z_0, \alpha_0)$ is singular, then (z_0, α_0) is called a *simple fold* (*limit or turning*) *point.* Note that this occurs if and only if the dimension of the null space (kernel) of $D_z F(z_0, \alpha_0)$ is one and $D_\alpha F(z_0, \alpha_0)$ is not in the range of $D_z F(z_0, \alpha_0)$. Since we may interchange the parameter α and one of the coordinates, say z^j, in z and apply the implicit function treating z^j as a parameter, the solution set can be parameterized by $(z, \alpha) = (z(\epsilon), \alpha(\epsilon))$ where $\epsilon = z^j - z_0^j$. Let ψ and ψ^* span the null spaces of $D_z F(z_0, \alpha_0)$ and $D_z F(z_0, \alpha_0)^T$, respectively. Then [21]

$$\alpha(0) = 0, \quad \frac{d\alpha(0)}{d\epsilon} = 0, \quad \text{and} \quad \frac{d^2\alpha(0)}{d\epsilon^2} = -\frac{\langle D_z^2 F(z_0, \alpha_0)\psi\psi, \psi^* \rangle}{\langle D_\alpha F(z_0, \alpha_0), \psi^* \rangle}.$$

If

$$\frac{d^2\alpha(0)}{d\epsilon^2} \neq 0,$$

the simple fold is called a *simple quadratic fold* since α is quadratic in ϵ. (Note that $\langle D_\alpha F(z_0, \alpha_0), \psi^* \rangle \neq 0$ since $D_\alpha F(z_0, \alpha_0) \notin \mathcal{R}(D_z F(z_0, \alpha_0))$.)

For an explanation of the term bifurcation, let $w = w(\epsilon)$ be a smooth curve, defined on an open interval I containing the origin, and parameterized by a parameter

ϵ such that $F(w(\epsilon)) = 0$ for all $\epsilon \in I$. The point $w(0)$ is called a *bifurcation point* of the equation $F(w) = 0$ if there exists an $\epsilon_0 > 0$ such that every neighborhood of $w(0)$ contains solutions w of $F(w) = 0$ which are not on the path $\{w(\epsilon) : -\epsilon_0 < \epsilon < \epsilon_0\}$. To explain the term *simple bifurcation point*, suppose $\dim \mathcal{N}(D_zF(z_0, \alpha_0)) = 1$ and $D_\alpha F \in \mathcal{R}(D_zF(z_0, \alpha_0))$, and define $\mathcal{D} = b^2 - ac$ where $a = \langle D_\alpha^2 F + 2D_\alpha D_z F v + D_z^2 F vv, \psi^* \rangle$, $b = \langle D_z^2 F \psi v + D_\alpha D_z F \psi, \psi^* \rangle$, $c = \langle D_z^2 F \psi\psi, \psi^* \rangle$, the derivatives of F are evaluated at (z_0, α_0), v is the unique solution of $\langle v, \psi \rangle = 0$ and $D_zF(z_0, \alpha_0)v = -D_\alpha F(z_0, \alpha_0)$, and $w \equiv (z, \alpha)$. If $\mathcal{D} > 0$, then $w_0 = (z_0, \alpha_0)$ is called a *simple bifurcation point*, for then there are two smooth solution paths through (z_0, α_0) with linearly independent tangents [1], [21]. When $c = 0$, the simple bifurcation point gives rise to what is commonly called a pitchfork bifurcation in that one of the two solution branches has a vertical tangent with respect to the parameter α.

We can now proceed to the following theorem, which gives the standard results for bifurcation tests based on sign changes in the determinants in (6.1), extended for the aforementioned choices of the predictor directions $d = d(s)$.

THEOREM 6.1. *Assume that $P = \{w(s) : s_a < s < s_b\}$ is a C^1 parameterization of a path of solutions of a system $F(w) = 0$, where $F : \mathbb{R}^{m+1} \to \mathbb{R}^m$ is at least C^2. Suppose that this path is regular, except for the point $w(s_0)$ for some $s_0 \in (s_a, s_b)$, at which $D_zF(w(s))$ is singular. With regard to* $\det(D_zF(w(s)))$, *we have the following*:

(i) *Suppose the path P can be smoothly parameterized by the natural parameter α. If* $\det(D_zF(w(s)))$ *changes sign at s_0, then $w(s_0)$ is a bifurcation point.*

(ii) *If $w(s_0)$ is a simple bifurcation point and the zero eigenvalue of $D_zF(w(s_0))$ has algebraic multiplicity one, then at least one of the two smooth paths through $w(s_0)$ can be smoothly parameterized by the natural parameter α. Along such a path* $\det(D_zF(w(s)))$ *changes sign at s_0, but $d\alpha/ds$ remains of the same sign on (s_a, s_b). For pitchfork bifurcation, $d\alpha/ds$ changes sign but* $\det(D_zF(w(s)))$ *does not at s_0 along exactly one of the two paths.*

(iii) *If $w(s_0)$ is a simple quadratic fold point and the zero eigenvalue of $D_zF(w(s_0))$ has algebraic multiplicity one, then* $\det(D_zF(w(s)))$ *and $d\alpha/ds$ change sign at s_0. With regard to the determinant of $D_wG(w(s))$ we have*:

(iv) *If $w(s_0)$ is a simple fold point, i.e., $D_\alpha F \notin \mathcal{R}(D_zF)$ and if* $\dim \mathcal{N}(D_zF) = 1$ *at $w(s_0)$, and $d(s)^T \frac{dw(s)}{ds} > 0$, then* $\det(D_wG(w(s)))$ *does not change sign on (s_a, s_b).*

(v) *Let the point $w(s_0)$ be a simple bifurcation point as described above. If $d(s)^T \frac{dw(s)}{ds} > 0$ holds on (s_a, s_b), then* $\det(D_wG(w(s)))$ *changes sign at s_0 along both of the smooth paths through $w(s_0)$.*

Proof. The statements in (i), (ii), and (iii) are found in the work of Keller [21], and statement (vi) follows directly from Lemma 4.9 of [21]. For (v) note that

$$\frac{dw(s)}{d\alpha} = \begin{bmatrix} -D_zF(w(s))^{-1} D_\alpha F(w(s)) \\ 1 \end{bmatrix} \quad \text{for } s_0 \in (s_a, s_b) - \{s_0\}.$$

Consider the partitioning (3.5a) and observe that the Schur complement of D_zF in D_wG equals the scalar $d(s)^T \frac{dw(s)}{d\alpha} = d(s)^T \frac{dw(s)}{ds} \frac{ds}{d\alpha}$ for $s_0 \in (s_a, s_b) - \{s_0\}$. Since the sign of $d(s)^T \frac{dw(s)}{ds}$ does not change on $(s_a, s_b) - \{s_0\}$, the sign of the Schur complement changes at s_0 if and only if the sign of $\frac{ds}{d\alpha}$ changes there. Now use (3.6a) to obtain $\det(D_wG) = d(s)^T \frac{dw(s)}{d\alpha} \det(D_zF)$. Thus $\det(D_wG(w(s)))$ changes sign at s_0 since exactly one of $\det(D_zF(w(s)))$ and $\frac{ds}{d\alpha}$ changes sign at s_0. □

Finally, note that changes in the signs of the determinants in (6.1) do not occur when an *even* number of *real* eigenvalues of D_zF or D_wG cross zero. Additional techniques can be found in the works of Keller [21], Kupper, Mittelmann, and Weber [22], and references therein.

For the parametric nonlinear programming problem, singularity detection is based on the full system (2.6); however, we may equivalently consider the active set system (2.8) and monitor the sizes and signs in the inactive constraints c_i $(i \in I - \mathcal{A})$ and the active inequality multipliers λ_i $(i \in \mathcal{A} \cap I)$ to detect a zero and thus a loss of strict complementarity. The detection test for this singularity are further discussed in §6.1. In §6.2 we show that detecting a zero in the parameter ν yields a test for the loss of the linear independence constraint qualification that is far more comprehensive than determining a change in the sign of the determinant in (6.1a). Thus the normalization $\nu^2 + \lambda^T\lambda - \beta_0^2 = 0$ plays a fundamental role in this singularity detection scheme. Detection tests for the singularity of the Hessian of the Lagrangian on the tangent space to the active constraints are presented in §6.3 and are based on detecting changes in the inertia of the reduced Hessian as presented in §5. Thus these tests are able to detect an odd number of eigenvalues of the Lagrangian matrix W crossing zero and an even number crossing in the same direction.

6.1. Loss of strict complementarity and the active set system. A rather comprehensive investigation of the singularity associated with the loss of strict complementarity in the system (2.6) has been previously reported by Tiahrt and Poore [33], and the theoretical results will not be repeated here. Detection of the occurrence of this singularity is not based on the signs of the determinants in (6.1), but rather on detecting a zero in any of the inactive constraints or active inequality multipliers. *If an inactive constraint c_i $(i \in I - \mathcal{A})$ or if a multiplier λ_i corresponding to an active inequality constraint $(i \in \mathcal{A} \cap I)$ changes sign or becomes zero, then there is a loss of strict complementarity.* We change branches at the zero by either activating or deactivating the corresponding constraint, respectively, thereby changing the active set. Note that a change in the sign of an inactive constraint c_i indicates that the solution path has crossed the boundary of the feasible region for that constraint; a change in the sign in one or more of the multipliers λ_i $(i \in \mathcal{A} \cap I)$ indicates that the critical point type has changed.

6.2. Loss of the linear independence constraint qualification. A bifurcation analysis of singularities due to the violation of the linear independence constraint qualification can be found in the work of Tiahrt and Poore [33] as well as in that of Jongen, Jonker, and Twilt [14]–[16], and thus we forego a presentation of the theoretical bifurcation analysis. In this subsection, strict complementarity and nonsingularity of the Hessian of the Lagrangian on the tangent space will be assumed, so that we only need to consider the active set system (2.8). The theorems in this subsection establish a test that is far more comprehensive than monitoring changes in the signs of the determinants in (6.1) and is very simple: *a change in the sign or the occurrence of a zero of ν indicates a singularity due to violation of the linear independence constraint qualification.* The next theorem provides a basis for this test. The remainder of this subsection establishes the relation between this test and any test based on detecting changes in the signs of $\det(M)$ and $\det(J)$ defined in equation (3.5). The final results in Theorem 6.6 are fairly strong in that changes in the sign of $\det(M)$ and $\det(J)$ can be detected from sign changes in ν and $\det(S_W)$ where S_W is the 2×2 Schur complement of W in J defined in equation (3.10), thereby removing the need to compute the former determinants. This is particularly beneficial if indirect or iterative methods are used to solve the linear systems.

THEOREM 6.2. *Assume f and $c \in C^2$ at (x, α) and let $w = (x, \lambda, \nu, \alpha)$ be a solu-*

tion of the active set system (2.8). *If* $\nu = 0$, *then* $A^T = D_x\bar{c}(x,\alpha)$ *is rank deficient, the linear independence constraint qualification* (*condition* (ii) *of Theorem* 2.1) *is violated, and* W *is singular at* w. *On the other hand, if* W *is nonsingular at* w, *then* $\nu \neq 0$ *and*

$$\ell^T v = 2\beta_0^2/\nu, \tag{6.2}$$

where ℓ *and* v *are defined in* (3.9).

Proof. If $\nu = 0$ in (2.8), $\nabla_x\mathcal{L}(x,\lambda,\nu,\alpha) = 0$ and the normalization equation $\nu^2 + \|\lambda\|_2^2 = \beta_0^2$ imply $A\lambda = 0$ and $\lambda \neq 0$. Thus the columns of A are dependent. The remainder of the first statement follows from Theorem 3.3. On the other hand, when W is nonsingular, A has full rank (by Theorem 3.3). This, along with the contrapositive of the first statement, implies $\nu \neq 0$. The representations (4.3a) and (3.8) and the fact that T^T is a left inverse of A (see (4.3a)) yield $\frac{1}{2}\nu\ell^T v = \frac{1}{2}\nu[2\lambda^T T^T(-\nabla_x f) + 2\nu] = \lambda^T T^T(-\nu\nabla_x f) + \nu^2 = \lambda^T T^T A\lambda + \nu^2 = \lambda^T\lambda + \nu^2 = \beta_0^2$. Thus (6.2) holds.

The next theorem provides a basis for determining changes in the signs of $\det(M)$ and $\det(J)$ in terms of the signs of ν, $\det(S_W)$, and $\det(W)$.

THEOREM 6.3. *Assume* f *and* $c \in C^2$ *at* (x,α), *let* $w = (x,\lambda,\nu,\alpha)$ *be a solution of the active set system* $\bar{F}(w) = 0$ (*see* (2.8)), *and let* $M = D_z\bar{F}(w(s))$ *and* $J = D_wG(w(s))$ *be as in* (3.5). *Also, let the Lagrangian matrix* W (*see* (3.4)) *and the* 2×2 *Schur complement* S_W *of* W *in* J (*see* (3.10)) *be evaluated at* w. *If* W *is nonsingular, then*

$$\det(M) = \left(\frac{2\beta_0^2}{\nu}\right)\det(W) \tag{6.3a}$$

and

$$\det(J) = \det(S_W)\det(W). \tag{6.3b}$$

Proof. Since W is nonsingular, M is also nonsingular by Theorem 3.3. Now apply formula (3.6a) of Theorem 3.1 to the partitionings of M and J in (3.5a), (3.5b). For the former matrix,

$$\begin{aligned}\det(M) &= \det(W)\left(2\nu - (0, 2\lambda^T)W^{-1}\begin{bmatrix}\nabla_x f(x,\alpha)\\ 0\end{bmatrix}\right)\\ &= \det(W)\left(\begin{bmatrix}0\\ 2\lambda\end{bmatrix}^T \tilde{v} + 2\nu\right) = (\ell^T v)\det(W).\end{aligned}$$

Now apply formula (6.2) in Theorem 6.2 to obtain (6.3a). Then the partitioning (3.5b) of J implies $\det(J) = \det(W)\det(S_W)$. □

The next two theorems show that when only the linear independence constraint qualification is violated along a path, the sign of $\det(W)$ and $\text{in}(W)$ do not change. Thus a change in the sign of $\det(M)$ ($\det(J)$) occurs if and only if ν (respectively, $\det(S_W)$) changes sign. The need to compute determinants of M, W, or J is thereby removed. First, a linear algebra result is needed.

THEOREM 6.4. *Let*

$$W^i = \begin{bmatrix}H^i, A_i\\ A_i^T, 0\end{bmatrix}$$

for $i = 1$ *and* 2 *be symmetric matrices of the same size, and suppose* A_1 *and* A_2 *are of the same size and rank. Then* $\text{in}(W^1) = \text{in}(W^2)$ *if and only if* $\text{in}(H^1_{T_1}) = \text{in}(H^2_{T_2})$, *where the H_T denotes the restriction of H to the tangent space $\mathcal{N}(A^T)$. Moreover,* $\mathrm{in}(H_{T_1}^1) - \mathrm{in}(H_{T_2}^2) = \mathrm{in}(W^1) - \mathrm{in}(W^2)$.

Proof. Suppose A_1 and A_2 are $n \times a$ matrices with common rank k. Apply Theorem 5.1 to obtain $\mathrm{in}(W^i) = \mathrm{in}(H_{T_i}^i) + (k, k, a-k)$ for $i = 1$ and 2. Subtract these two equations to obtain the equation $\mathrm{in}(W^1) - \mathrm{in}(W^2) = \mathrm{in}(H_{T_1}^1) - \mathrm{in}(H_{T_2}^2)$, from which the conclusion follows. □

THEOREM 6.5. *Let f and c be C^2 functions and assume that $P = \{w(s) : s_a < s < s_b\}$ is a C^1 parameterization of a path of solutions of the active set system (2.8). Suppose that this path is regular, except for the point $w(s_0)$ for some $s_0 \in (s_a, s_b)$, at which the linear independence constraint qualification is violated, but strict complementarity holds. Let W represent the Lagrangian matrix (3.4), and H_T the reduced Hessian matrix, both of which are evaluated at $w(s)$, and suppose that* $\mathrm{in}(H_T)$ *remains constant on $(s_a, s_b) - \{s_0\}$. Then both sign* $\det(W)$ *and* $\mathrm{in}(W)$ *remain constant on $(s_a, s_b) - \{s_0\}$.*

Proof. Since strict complementarity holds, the active set $\mathcal{A}$, and hence the number of eigenvalues of W and H_T and the rank of $A = D_x\bar{c}(x, \alpha)$, remain constant on $(s_a, s_b) - \{s_0\}$. The application of Theorem 6.4 at regular points $w(s_1)$ and $w(s_2)$ for $s_a < s_1 < s_0 < s_2 < s_b$ yields $\mathrm{in}(W^1) = \mathrm{in}(W^2)$, which shows that $\mathrm{in}(W)$ remains constant on $(s_a, s_b) - \{s_0\}$. Theorem 3.3 implies that W is nonsingular on $(s_a, s_b) - \{s_0\}$, so that $\det(W)$ is nonzero there and its sign depends only on the number of negative eigenvalues of W, which is constant on $(s_a, s_b) - \{s_0\}$. □

That $\mathrm{in}(H_T)$ remains constant on $(s_a, s_b) - \{s_0\}$ is implied by a different hypothesis, that H_T is nonsingular at $w(s_0)$, as is shown in the work of Tiahrt and Poore [33, p. 127].

THEOREM 6.6. *Let the assumptions and notation of Theorem 6.5 hold, let $w = (x, \lambda, \nu, \alpha)$ be a solution of the active set system $\bar{F}(w) = 0$ (see (2.8)), let $M = D_z\bar{F}(w(s))$ and $J = D_wG(w(s))$ be as in (3.5), and let S_W be the 2×2 Schur complement of W in J (see (3.10)), evaluated at w. Then* $\det(M)$ *changes sign at s_0 if and only if ν changes sign at s_0. Also,* $\det(J)$ *changes sign at s_0 if and only if* $\det(S_W)$ *changes sign at s_0.*

If the path P can be smoothly parameterized by the natural parameter α, a sign change in ν at s_0 implies that $w(s_0)$ is a bifurcation point. If, in addition, $w(s_0)$ is a simple bifurcation point described in Theorem 6.1 (ii)*, then ν must change sign at s_0. Also, ν changes sign at s_0 if $w(s_0)$ is a simple quadratic fold described in Theorem* 6.1 (iii)*. Assume, in addition to the above, that the predictor stepsize Δs is sufficiently small so that the prediction direction $d_k(\Delta s)$ satisfies $d_k(\Delta s)^T dw(s)/ds > 0$. Then simple bifurcation and simple folds are distinguished by the fact that* $\det(S_W)$ *changes sign at s_0 for the former, but not for the latter.*

Proof. By Theorem 6.5, $\det(W)$ does not change sign as s crosses s_0. Thus (6.3a) of Theorem 6.3 implies that ν and $\det(M)$ change sign together, which proves the first statement. Similarly, by (6.3b) of Theorem 6.3, $\det(S_W)$ and $\det(J)$ change sign together, and this proves the second statement. The remaining statements now follow directly from Theorem 6.1. □

Finally, if ν does change sign and (i) and (iii) of Theorem 2.1 are not violated, then a complete reversal in the type of the critical point is indicated, i.e., the multipliers λ_i and the eigenvalues of H_T all change sign if the normalization $\nu > 0$ is reapplied [33].

It is possible that $\nu \neq 0$ when the linear independence constraint qualification fails, and this occurs exactly when $\nabla_x f \in \mathcal{R}(D_x\bar{c}^T)$. Generally, this yields a higher-order singularity. The two possible cases are described in the next theorem, whose proof follows from (2.8).

THEOREM 6.7. *Let $w = (x, \lambda, \nu, \alpha)$ be a solution of the active set system* (2.8) *and let $A^T = D_x\bar{c}(x,\alpha)$.*

(i) *If $\nabla_x f \notin \mathcal{R}(A)$, then $\nu = 0$ and A is rank deficient. Furthermore, if a matrix $U \in \mathbb{R}^{n\times j}$ is chosen with columns forming an orthonormal basis for $\mathcal{N}(A)$, then $\tilde{w} = (x, U\mu, 0, \alpha)$ is a solution of system* (2.8) *for any $\mu \in \mathbb{R}^j$ with $\|\mu\|_2 = \beta_0$.*

(ii) *Suppose that $\nabla_x f \in \mathcal{R}(A)$, A is rank deficient, and $U \in \mathbb{R}^{n\times j}$ has columns forming an orthonormal basis for $\mathcal{N}(A)$. Let $A\xi = \nabla_x f$ and define $\lambda_0 = (I - UU^T)\xi$. Then $\hat{w} = (x, \hat{\lambda}, \hat{\nu}, \alpha)$ is a solution of system* (2.8) *for any $\hat{\nu}$ and $\hat{\lambda} = \hat{\nu}\lambda_0 + U\mu$ with $\mu \in \mathbb{R}^j$ and $\hat{\nu}^2(1+\lambda_0^2) + \|\mu\|_2^2 = \beta_0^2$. Any of these solutions with $\|\mu\|_2 < \beta_0$ has $\hat{\nu} \neq 0$.*

6.3. Singularity of the Hessian of the Lagrangian on the tangent space. If the linear independence constraint qualification and strict complementarity conditions hold along a smooth solution path, the eigenvalues of $H_T = \nabla_x^2\mathcal{L}_T$ vary continuously along this path [33], so that any change in in(H_T) between two regular points on the solution path indicates that a singularity in H_T has been crossed. The computation of in(H_T) (see §5) at regular points has already been developed as an inexpensive by-product of the linear algebra for the continuation procedure. Thus, the detection test for this case is: *Changes in* in(H_T) *indicate singularity of the Hessian of the Lagrangian on the tangent space to the active constraints.*

The relation between this test and tests based on the signs of det(M) and det(J) in (6.1) can be explained as follows. As a consequence of Theorems 6.4 and 6.5, a change in the sign of det(W) is indicated by a change in in(H_T). Theorem 6.3 shows that changes in the signs of det(M) and det(J) are easily detected from changes in the signs of det(W), ν, and det(S_W) in the course of using (3.8)–(3.11). Note that the value of ν and the entries of the 2×2 matrix S_W are immediately available from the computations described in (3.8)–(3.11). Thus the basic bifurcation test is essentially free if type is monitored, as indicated in §5. Finally, note that these tests involving sign ν and in(H_T) are much stronger tests for singularities than are available using the determinants of M and J, since the latter tests cannot detect cases in which an even number of eigenvalues of M or J changes from positive to negative or negative to positive. Also, det(M) does not change sign when ν and det(W) change sign together; det(J) does not change sign when det(S_W) and det(W) change sign together. Each of these situations is detected by the tests involving sign ν and in(H_T).

7. A model problem from design optimization.

The numerical continuation techniques described in the previous sections will now be used to obtain a "global" analysis of the sensitivity, stability, and multiplicity of minima for a parametric nonlinear programming problem arising from design optimization. The problem, which is simple yet still exhibits the basic phenomena, involves the design of a two-bar planar truss with semispan 1, unloaded height h, and load p as indicated in Fig. 7.1.

Given a specific unloaded height h and load p, the deflection d is a minimizer of the potential energy $E(d,h;p) = -pd + \left(\sqrt{1+h^2} - \sqrt{1+(h-d)^2}\right)^2 / \sqrt{1+h^2}$. Rheinboldt [31] used this model problem to illustrate continuation methods in structural analysis and has given a rather complete solution to both the static and parametric problems. Rao and Papalambros [30] posed a corresponding optimal design problem as that of choosing the height h to minimize the deflection subject to $0 \leq h \leq 1.5$. This problem is posed mathematically as:

$$\begin{array}{ll} & \text{Minimize} \quad d \\ (7.1) & \text{Subject to} \quad \nabla_d E(d,h;p) = 0, \quad 0 \leq h \leq 1.5. \end{array}$$

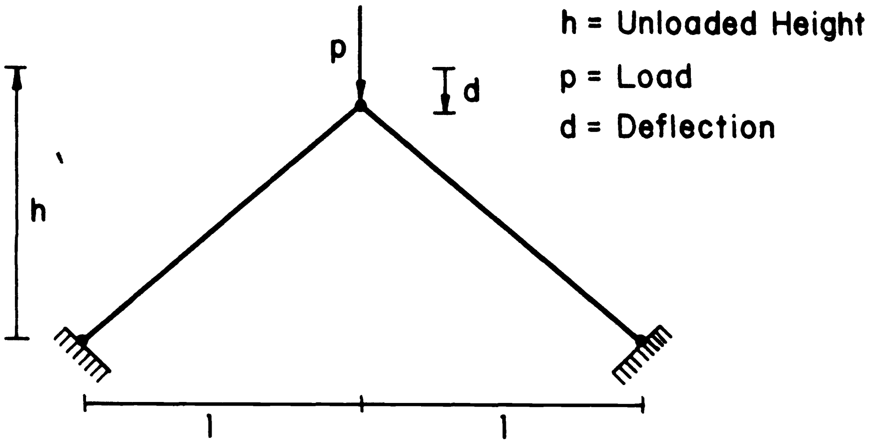


FIG. 7.1. *Loaded two-bar truss.*

In addition to selecting the minimizer, the state (d, h) must also be selected so that the potential energy $E(d, h; p)$ is minimized with respect to d. The corresponding parametric problem is to determine the solution and its properties as the load p varies over all physically important ranges. The remainder of this section contains a complete analysis of this problem.

A starting solution w_0 for continuation was obtained by solving this problem for $p = 0.05$, and then scaling the multipliers λ and ν so that $\beta_0 = 3$ in (2.8). The solution paths $w(p) = (z(p), p)$ of the active set system (2.8) were then tracked using an adaptation of the continuation code ABCON Chord described in [24]. Solution type was monitored and singularities detected, as described in the previous sections. Eight singular points, labeled (a)–(i) in Table 7.2 and Fig. 7.3, were encountered in addition to an entire path of singular points along which $p = d = 0$, but h, λ_1, and ν vary. In Table 7.2 SC, CQ, and HL denote *s*trict *c*omplementarity, linear independence *c*onstraint *q*ualification, and the nonsingularity of the *H*essian of the *L*agrangian on the tangent space to the active constraints, respectively, i.e., conditions (i)–(iii) of Theorem 2.1. The word "branch" refers to a singular point from which at least three distinct half-rays emerge. λ_a denotes the multiplier of the active inequality constraint.

Figure 7.3 gives the displacement d and unloaded height h as p varies and represents a projection of the solutions of (2.6) into (h, d, p) space. Solid and dashed lines indicate paths of local minimizers and maximizers, respectively. The dashed and dotted line represents a feasible singular path, and lines of small dots represent infeasible solutions. The solutions to the optimization problem need not be points of minimum potential energy $E(d, h; p)$, which is not minimized on the segments from (b) to (c) and from (d) to (c) to (e). However, all other feasible path segments do correspond to physical states of the truss where the potential energy is minimized.

We now describe these singularities, and the connecting path segments, beginning with those that occur along the solution branch where the constraint $h \leq 1.5$ is active. Loss of strict complementarity gives rise to the bifurcation points (g), (a), and (c), whose presence was indicated by a change in sign of the multiplier λ_a. At these points the inequality constraint becomes weakly active and solution paths bifurcate into the region $0 < h < 1.5$. The fold points (b) and (d) ($p = \pm .37$), which resulted from

TABLE 7.2
Singular points.

Point	Violations	Phenomena	d	h	λ_1	λ_a	ν	p
(a)	SC	Branch	0.09	1.5	−2.4	0.0	1.7	0.07
(b)	CQ	Fold	0.81	1.5	−2.8	1.0	0.0	0.37
(c)	SC	Branch	2.0	1.5	−2.9	0.0	−0.77	−0.35
(d)	CQ	Fold	2.2	1.5	−2.8	−1.0	0.0	−0.37
(e)	SC,CQ,HL	Branch	0.0	0.0	−3.0	0.0	0.0	0.0
(f)	HL	Branch	0.0	$\sqrt{2}$	−2.4	—	1.8	0.0
(g)	SC	Branch	0.0	1.5	−2.4	0.0	1.8	0.0
(i)	HL	Branch	0.0	$-\sqrt{2}$	−2.4	—	1.8	0.0

violation of the linear independence constraint qualification, were detected by a change in the sign of ν. The type of the solution along this solution branch is determined by the sign of λ_a/ν, which changes at each of these five singular points. This results in the alternating segments of minimizers and maximizers shown in Fig. 7.3.

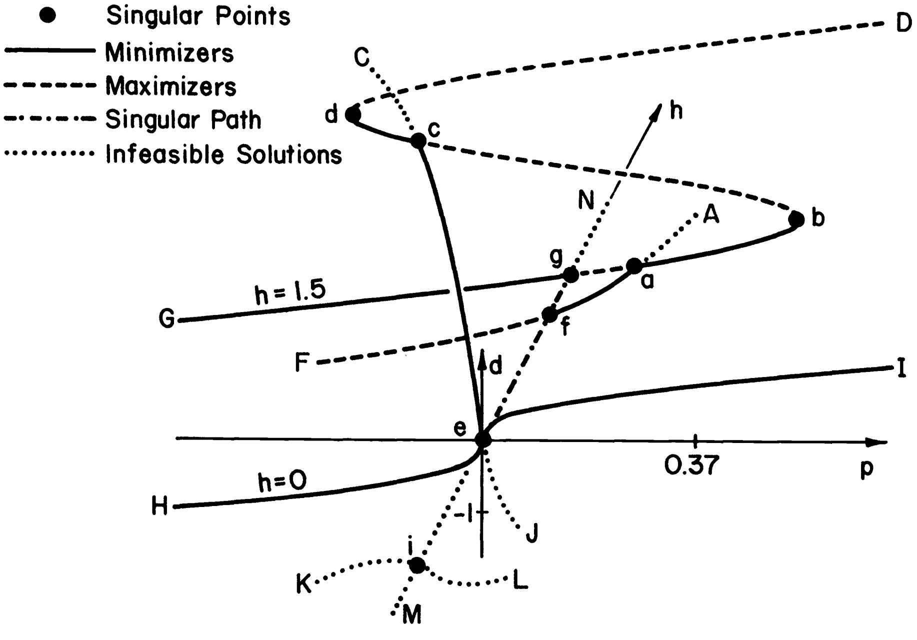


FIG. 7.3. *Solutions of* (2.6) *for problem* (7.1).

Along the solution branch corresponding to the constraint $h \geq 0$ being active, a multiple bifurcation point (e) occurs from which two additional paths emerge into the region $0 < h < 1.5$. At the point (e) all three conditions in Theorem 2.1 (strict complementarity, linear independence constraint qualification, and nonsingularity of the Hessian of the Lagrangian on the tangent space) are violated. Also, both ν and the multiplier λ_a become zero but do not change sign at (e), and this was detected in the continuation process by small values of these quantities. The fact that $\nabla_x^2\mathcal{L}_T$ is singular was detected by a change in in(W) along the path from (c) through (e) out of the feasible region along J.

In Fig. 7.3 three additional solution paths are shown which emanate from the constraints $h \geq 0$ and $h \leq 1.5$ into the region $0 < h < 1.5$: one path of minimizers branching from the bifurcation point (a), another from each of the bifurcation points (c) and (e), and one path of singular points branching from (e) and (g), along which $p = d = 0$. Singularity of $\nabla_x^2\mathcal{L}_T$ occurs at the bifurcation point (f), which was detected by a change in in(W) along the path from (a) through (f). At (f) there is a change in type resulting in the path of maximizers labeled F. Except for the bifurcations occurring at (e), (f), and (g), the path with $p = d = 0$ consists of degenerate fold point singularities.

Figure 7.3 also shows infeasible solutions of the system (2.8) which emerge at (a), (c), (g) ($h > 1.5$), and (e) ($h < 0$). Two infeasible solution branches pass through the bifurcation point (i) at which $\nabla_x^2\mathcal{L}_T$ is singular. In some problems infeasible paths may provide the opportunity for further branching to other feasible paths.

The number and location of the local minimizers is summarized in Table 7.4. The paths of global minimizers are indicated in boldface. It is important to stress that the multiple optima were found by continuation, not by reoptimizing from different starting points.

TABLE 7.4
Multiplicity of minima.

Parameter range	Number of minima	Paths
$-\infty < p < -0.37$	2	**G**, H
$-0.37 < p < 0$	3	**G-g**, d-e, H-e
$0 < p < 0.37$	2	**f-a-b**, I
$0.37 < p < \infty$	1	**I**

The solution to the parametric design problem can now be described for $p > 0$. Given a small but positive load p, the global minimum occurs on the branch of minimizers between singular points (f) and (a). As the load p is increased from zero, the height h increases from $\sqrt{2}$ to $h = 1.5$ where the constraints $h \leq 1.5$ become active. As the load p is increased further, the deflection d continues to increase along the path from (a) to (b) until p reaches 0.37 where the truss "snaps through" and there is no minimum beyond $p = 0.37$ corresponding to a height h near 1.5. (The only way to maintain an optimum locally beyond $p = 0.37$ is to increase the parameter $\beta = 1.5$ in the upper bound on the height h.) The local minimizer corresponding to $h = 0$ becomes the global minimizer for p beyond $p = 0.37$. Local sensitivity is surely present at points (a) and (b). Note that the path of minimizers is continuous but not differentiable at (a). (Near such points many optimization codes exhibit cycling.) At the fold point (b), the path of minimizers ceases to exist. Optimization codes would have difficulty here since the unnormalized multipliers will be large and go to infinity as p approaches 0.37. The conclusion with regard to the design of the truss is that for stability the loads must be less than $p = 0.37$ and that sensitivity occurs near the singular points (a) and (b) for the reasons stated. Clearly, the ability of the continuation procedure to locate such singular points and obtain such a global analysis is a major strength of the methodology.

8. Concluding remarks. Numerical methods have been developed for continuation, solution-type determination, and singularity detection in the parametric nonlinear programming problem. Starting from the Fritz John first-order necessary conditions, this problem was converted to a system of nonlinear equations (2.6) whose singularities are characterized in terms of the loss of strict complementarity, the violation of a linear independence constraint qualification, and/or the singularity of the Hessian of the Lagrangian on the tangent space to the active constraints. The singularity detection schemes developed in this work focus on these three singularities. The system of equations (2.6) employs a nonstandard normalization $\nu^2 + \lambda^T\lambda = \beta_0^2$ where ν and λ are multipliers in the Lagrangian $\mathcal{L}(z, \alpha) = \nu f(x, \alpha) + \sum_{i \in \mathcal{A}} \lambda_i c_i(x, \alpha)$. This normalization is central to the detection of singularities associated with a violation of the linear independence constraint qualification. For computational efficiency, the full system (2.6) was replaced by an active set system (2.8), and then a framework for combining various numerical continuation and bifurcation methods with a large number of null and range space methods from constrained optimization was developed using the bordering algorithm of Keller [19]–[21]. In this development, Schur complements [18], [26] have played an important and unifying role.

Using this framework, techniques for determining solution type were developed in §5. These involve little additional expense when symmetric factorization, null space, or range space methods are used in the continuation procedure. This framework was also used in §6 to develop singularity detection tests which are more comprehensive and efficient than those used for general nonlinear equations. In fact, the tests for singularities due to the loss of strict complementarity and the violation of the linear independence constraint qualification are shown to only require that we monitor the sizes and signs of multipliers λ_i, the inactive constraints c_i, the parameter ν, and the determinant of a 2×2 matrix S_W. These tests thus require very little additional computation, no factorizations, and no determinant computations other than that of the S_W. Hence, they can easily be applied when iterative solvers are used in the continuation procedure. Detection of a singularity of the Hessian of the Lagrangian on the tangent space to the active constraints is based on the inertia of the reduced Hessian, which we have shown how to compute at little extra expense when using symmetric factorization, null space, or range space methods.

In §7 these methods were applied to a model problem from design optimization [30] and were shown to yield a "global" analysis of the sensitivity, stability, and multiplicity of minima. This example also exhibits an extensive number of solution paths, each of the basic singularities, and jump phenomena arising from a loss of the linear independence constraint qualification. Finally, these methods should be equally applicable to discrete versions of abstract optimization and control problems.

REFERENCES

[1] E. L. Allgower and K. Georg, *Numerical Continuation Methods: An Introduction*, Springer-Verlag, Berlin, Heidelberg, New York, 1990.

[2] B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer, *Nonlinear Parametric Optimization*, Birkhäuser-Verlag, Basel, 1983.

[3] T. F. Chan, *Deflation techniques and block-elimination algorithms for solving bordered singular systems*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 121–134.

[4] T. F. Chan and D. C. Resasco, *Generalized deflated block-elimination*, SIAM J. Numer. Anal., 23 (1986), pp. 913–924.

[5] A. V. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Academic Press, New York, 1983.

[6] A. V. FIACCO, *Mathematical Programming Study* 21: *Sensitivity, Stability and Parametric Analysis*, North-Holland, Amsterdam, 1984.

[7] R. FLETCHER, *Practical Methods of Optimization*, 2nd ed., John Wiley, New York, 1987.

[8] P. E. GILL, W. MURRAY, AND M. H. WRIGHT, *Practical Optimization*, Academic Press, New York, 1981.

[9] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, 1989.

[10] M. GOLUBITSKY AND D. G. SCHAEFFER, *Singularities and Groups in Bifurcation Theory*, Vol. 1, Springer-Verlag, New York, 1985.

[11] ———, *Singularities and Groups in Bifurcation Theory*, Vol. 2, Springer-Verlag, New York, 1988.

[12] J. GUDDAT, F. GUERRA VAZQUEZ, AND H. TH. JONGEN, *Parametric Optimization: Singularities, Path Following, and Jumps*, John Wiley and Sons, Chichester, England, 1990.

[13] J. GUDDAT, ED., *Parametric Optimization and Related Topics* II, Mathematical Research 62, Akademie-Verlag, Berlin, 1991.

[14] H. TH. JONGEN, P. JONKER, AND F. TWILT, *Nonlinear Optimization in* $\mathbb{R}^n$*: I. Morse Theory, Chebyshev Approximation*, Verlag Peter Lang, New York, 1983.

[15] ———, *Nonlinear Optimization in* $\mathcal{R}^n$*: II. Transversality, Flows, Parametric Aspects*, Verlag Peter Lang, New York, 1986.

[16] ———, *One-parameter families of optimization problems: equality constraints*, J. Optim. Theory Appl., 48 (1986), pp. 141–161.

[17] H. TH. JONGEN AND G.-W. WEBER, *On parametric nonlinear programming*, Ann. Oper. Res., 27 (1990), pp. 253–284.

[18] H. TH. JONGEN, T. MÖBERT, J. RÜCKMANN, AND K. TAMMER, *On inertia and Schur compliment in optimization*, Linear Algebra Appl., 95 (1987), pp. 97–109.

[19] H. B. KELLER, *Numerical solution of bifurcation and nonlinear eigenvalue problems*, in Applications of Bifurcation Theory, P. H. Rabinowitz, ed., Academic Press, New York, 1977, pp. 359–394.

[20] ———, *The bordering algorithm and path following near singular points of higher nullity*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 573–582.

[21] ———, *Numerical Methods in Bifurcation Problems*, Springer-Verlag, Berlin, 1987.

[22] T. KUPPER, H. D. MITTELMANN, AND H. WEBER, *Numerical Methods for Bifurcation Problems*, Birkhäuser-Verlag, Boston, 1984.

[23] T. KUPPER, R. SEYDEL, AND H. TROGER, EDS., *Bifurcation: Analysis, Algorithms, Applications*, Birkhäuser-Verlag, Boston 1987.

[24] B; N. LUNDBERG AND A. B. POORE, *Variable order Adams–Bashforth predictors with an error-stepsize control for continuation methods*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 695–723.

[25] ———, *Bifurcations and sensitivity in parametric nonlinear programming*, Third Air Force/ NASA Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, San Francisco, CA, Sept. 24–26, 1990.

[26] D. V. OUELLETTE, *Schur complements in statistics*, Linear Algebra Appl., 36 (1981), pp. 187–295.

[27] A. B. POORE AND C. A. TIAHRT, *Bifurcation problems in nonlinear parametric programming*, Math. Programming, 39 (1987), pp. 189–205.

[28] J. RAKOWSKA, R. T. HAFTKA, AND L. T. WATSON, *An active set algorithm for tracing parameterized optima*, Struct. Optim., 3 (1991), pp. 29–44.

[29] ———, *Tracing the efficient curve for multi-objective control-structure optimization*, Comput. Systems Engrg., 2 (1991), pp. 461–472.

[30] J. R. J. RAO AND P. Y. PAPALAMBROS, *Extremal behavior of one parameter families of optimal design models*, ASME Design Automation Conference, Montreal, Sept. 17–20, 1989.

[31] W. C. RHEINBOLDT, *Numerical analysis of continuation methods for nonlinear structural problems*, Comput. Struct., 13 (1981), pp. 103–113.

[32] ———, *Numerical Analysis of Parameterized Nonlinear Equations*, John Wiley, New York, 1985.

[33] C. A. TIAHRT AND A. B. POORE, *A bifurcation analysis of the nonlinear parametric programming problem*, Math. Programming, 47 (1990), pp. 117–141.

SIAM J. OPTIMIZATION
Vol. 3, No. 1, pp. 155–163, February 1993

REMARKS ON CONVERGENCE OF THE MATRIX SPLITTING ALGORITHM FOR THE SYMMETRIC LINEAR COMPLEMENTARITY PROBLEM*

WU LI[†]

Abstract. In this paper it is shown how to remove a restriction on the perturbation vectors h^i imposed by Mangasarian and Luo and Tseng in their respective convergence analyses of the matrix splitting algorithms.

Key words. matrix splitting algorithm, linear complementarity problem, linear convergence, error estimate, power method

AMS(MOS) subject classifications. 90C33, 90C20

1. Introduction.

Consider the classical symmetric linear complementarity problem (LCP) of finding an x in the n-dimensional real space $\mathbb{R}^n$ such that

$$Mx + q \geq 0, \quad x \geq 0, \quad x(Mx+q) = 0, \tag{1.1}$$

where M is a given $n \times n$ real symmetric positive semidefinite matrix and q is a given vector in $\mathbb{R}^n$. Based on works done earlier [2], [9], Pang [14] proposed the following general matrix splitting method for solving (1.1):

$$Bx^{i+1} + Cx^i + q \geq 0, \quad x^{i+1} \geq 0, \quad x^{i+1}(Bx^{i+1} + Cx^i + q) = 0, \quad i = 0, 1, \ldots, \tag{1.2}$$

where $M = B + C$ is a *regular splitting* (cf. [13] and [3]), that is,

$$M = B + C, \quad B - C \ \text{ positive definite.}$$

The regular splitting of M guarantees that (1.2) is a descent method for the following equivalent convex quadratic programming problem of (1.1):

$$\begin{aligned} &f_{\min} = \min f(x) := \tfrac{1}{2} xMx + qx \\ &\text{subject to} \quad x \geq 0 \end{aligned}$$

and any accumulation point of $\{x^i\}$ is a solution of (1.1) [14]. The key to establishing the convergence of subsequences of $\{x^i\}$ is to prove the boundedness of $\{x^i\}$, which was investigated extensively [14], [15], [16], [4], [17]. Luo and Tseng were the first to prove the convergence of the whole sequence $\{x^i\}$ [8]. However, their convergence analysis is rather complex and requires that the subproblems (1.2) be solved exactly. Concerned about the practical implementation of (1.2), Mangasarian [10] considered solutions of the subproblems (1.2) with perturbations $\{h^{i+1}\}$:

$$\begin{aligned} &Bx^{i+1} + Cx^i + q - h^{i+1} \geq 0, \quad x^{i+1} \geq 0, \\ &x^{i+1}(Bx^{i+1} + Cx^i + q - h^{i+1}) = 0, \quad i = 0, 1, \ldots. \end{aligned} \tag{1.3}$$

*Received by the editors May 28, 1991; accepted for publication (in revised form) December 23, 1991.

[†]Department of Mathematics and Statistics, Old Dominion University, Norfolk, Virginia 23529 (li@xanth.cs.odu.edu).

He proved the convergence of $\{x^i\}$ generated by (1.3) under the assumption that B is symmetric, $\sum_{i=1}^{\infty} \|h^i\| < \infty$, and

$$\sum_{i=1}^{\infty} \|h^{i+1}\| \cdot (\|x^i - x^{i+1}\| + \|x^i - x^*\|) < \infty, \tag{1.4}$$

where x^* is any solution of (1.1). Note that (1.4) is not the exact assumption he made, but is an equivalent one.

A key step in Mangasarian's proof is to show that $x^{i+1} := \varphi(x^i)$ as a mapping defined by (1.2) is nonexpansive in the B-norm. However, φ is not a nonexpansive mapping in the B-norm if B is not symmetric. Thus, it would be difficult to extend his approach to general cases. At the same time, Luo and Tseng [5] improved their original approach and established the linear convergence of $\{x^i\}$ generated by (1.3) under the assumption that

$$\|h^{i+1}\| \leq (\gamma - \epsilon) \|x^i - x^{i+1}\|, \tag{1.5}$$

where 2γ is the smallest eigenvalue of $B - C$ and $\epsilon > 0$. Actually, their method is general enough to prove the linear convergence of the iterates generated by descent methods for convex essentially smooth minimization problems. There are two key steps in Luo and Tseng's proof of the linear convergence of $\{x^i\}$: one is a local error estimate:

$$d(x^i, X^*) \leq \tau \cdot \|x^i - (x^i - (Mx^i + q))_+\|, \qquad i \geq r, \tag{1.6}$$

and another is an estimate of the speed of convergence of $f(x^i)$:

$$f(x^i) - f_{\min} \leq \beta \cdot (d(x^i, X^*))^2, \qquad i \geq r, \tag{1.7}$$

where r is some positive integer, β is a fixed scalar, x_+ denotes the vector in $\mathbb{R}^n$ with components $(x_+)_i := \max\{x_i, 0\}, i = 1, \ldots, n$, X^* is the solution set of (1.1), and $d(x^i, X^*)$ is the distance from x^i to the set X^* defined as

$$d(x^i, X^*) := \min\{\|x^i - x^*\| : x^* \in X^*\}.$$

Based on (1.6) and (1.7) they derived the following key inequality in their convergence analysis:

$$f(x^i) - f_{\min} \leq \alpha \cdot (\|x^i - x^{i+1}\| + \|h^{i+1}\|)^2, \qquad i \geq r. \tag{1.8}$$

Recently, Luo and Tseng [6] showed that their approach can be used to prove the linear convergence (in the root sense [13]) of $\{x^i\}$ generated by (1.3) with the assumption that M is symmetric, h^i satisfy (1.5), f is bounded below on $\mathbb{R}^n_+$, and $M = B + C$ is a regular splitting. Their main effort was to show that (1.8) holds even when M is only symmetric. This further demonstrates the power of their approach.

In general, one cannot have a global error estimate of (1.6) as shown by Mangasarian and Shiau [11]. Luo and Tseng [7] studied cases when (1.6) holds for all $x \in \mathbb{R}^n_+$ and are able to obtain a characterization of such cases by using the index set of active constraints of the solution set. For the matrix splitting algorithms, they proved that (1.6) implies that (1.8) holds with α, depending only on τ, M, and B.

Therefore, Luo and Tseng were able to obtain a global estimate of $\|x^i - x^{i+1}\|$ (cf. Corollary 4 in [7]).

One of the main goals of this paper is to show that the sequence $\{x^i\}$ generated by (1.3) converges with the assumption

$$\sum_{i=1}^{\infty} \|h^i\| < \infty, \tag{1.9}$$

which is weaker than (1.5). Moreover, $\{x^i\}$ converges linearly if $\{h^i\}$ converges linearly to 0 (cf. Theorem 3.3). Since (1.5) and the linear convergence of $\{x^i\}$ imply that $\{h^i\}$ converges linearly to 0, the linear convergence of $\{h^i\}$ is a weaker restriction on h^i than (1.5). Also, it is easy to implement in practical situations, since the condition of linear convergence of h^i can be set a priori. The proof of our results is almost the same as the proof given by Luo and Tseng, except that we manipulate inequalities differently. In addition, we show that Mangasarian's method provides an alternative way to prove the convergence of $\{x^i\}$ under the additional assumption that B is symmetric. The alternative proof gives a different perspective of why $\{x^i\}$ converges.

In §2 we include some inequalities that are commonly used to study the convergence of $\{x^i\}$ generated by (1.3). In §3, we show how to modify Luo and Tseng's approach to eliminate the restriction (1.5) on h^i. Section 4 shows an alternative way to prove the convergence of $\{x^i\}$ generated by (1.3) under the additional assumption that B is symmetric. In §5 we give comments on how to extend the results to affine variational inequality problems.

Commonly used notations are included here. $\mathbb{R}^n_+ := \{x \in \mathbb{R}^n : x \geq 0\}$. A positive definite $n \times n$ real matrix B induces an elliptic norm $\|\cdot\|_B$ on $\mathbb{R}^n$, defined by $(xBx)^{1/2}$ for x in $\mathbb{R}^n$. When $B = I$, we have the Euclidean or two-norm $(xx)^{1/2}$, which we denote simply as $\|\cdot\|$. To avoid ambiguity, we make a standing assumption for the following sections.

ASSUMPTION 1.1. *M is symmetric positive semidefinite, $M = B + C$ is a regular splitting,* (1.1) *has at least one solution, and h^i satisfy* (1.9).

2. Preliminary lemmas.

In this section we review some inequalities commonly used to analyze the convergence of $\{x^i\}$. The first lemma is proved by Luo and Tseng [5].

LEMMA 2.1. *There is a constant $\beta > 0$ such that*

$$\|x^i - (x^i - (Mx^i + q))_+\| \leq \beta(\|x^i - x^{i+1}\| + \|h^{i+1}\|).$$

The following three lemmas are implicitly proved in [10] and [5]. For easy reference, we reproduce the proof here.

LEMMA 2.2. *Let 2γ be the smallest eigenvalue of $B - C$. Then*

$$\|x^i - x^{i+1}\|^2 \leq \frac{2}{\gamma}\left(f(x^i) - f(x^{i+1}) + \frac{1}{2\gamma}\|h^{i+1}\|^2\right).$$

LEMMA 2.3. *Let 2γ be the smallest eigenvalue of $B - C$. Then*

$$\|x^i - x^{i+1}\| \leq \sqrt{\tfrac{2}{\gamma}} \cdot |f(x^i) - f(x^{i+1})|^{1/2} + \tfrac{1}{\gamma} \cdot \|h^{i+1}\|.$$

LEMMA 2.4. *Any accumulation point of $\{x^i\}$ generated by* (1.3) *is a solution of* (1.1) *and* $\lim_{i\to\infty} \|x^i - x^{i+1}\| = 0$.

Proof. One can verify that (cf. [10] and [5])

$$f(x^i) - f(x^{i+1}) \geq \gamma \cdot \|x^i - x^{i+1}\|^2 - \|h^{i+1}\| \cdot \|x^i - x^{i+1}\|.$$

The above inequality can be rewritten as

$$(2.1) \qquad f(x^i) - f(x^{i+1}) \geq \gamma \left(\|x^i - x^{i+1}\| - \frac{1}{2\gamma}\|h^{i+1}\| \right)^2 - \frac{1}{4\gamma}\|h^{i+1}\|^2.$$

Since $(a+b)^2 \leq 2(a^2+b^2)$ for $a, b \geq 0$,

$$(2.2) \qquad \gamma \cdot \|x^i - x^{i+1}\|^2 \leq 2\left\{ \gamma \left(\|x^i - x^{i+1}\| - \frac{1}{2\gamma}\|h^{i+1}\| \right)^2 + \frac{1}{4\gamma}\|h^{i+1}\|^2 \right\}.$$

It follows from (2.1) and (2.2) that

$$(2.3) \qquad \|x^i - x^{i+1}\|^2 \leq \frac{2}{\gamma}\left(f(x^i) - f(x^{i+1}) + \frac{1}{2\gamma}\|h^{i+1}\|^2 \right).$$

Lemmas 2.2 and 2.3 follow from (2.3) and the inequality $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b \geq 0$.

Let $f_{\min}$ be the minimum value of f on $\mathbb{R}^n_+$. Then (2.1) implies

$$(2.4) \qquad 0 \leq (f(x^{i+1}) - f_{\min}) \leq (f(x^i) - f_{\min}) + \frac{1}{4\gamma}\|h^{i+1}\|^2.$$

Since $\sum_{i=1}^{\infty} \|h^{i+1}\| < \infty$ (cf. Assumption 1.1), $\{\|h^i\|\}$ is a bounded sequence; i.e., there is a constant $\beta > 0$ such that $\|h^i\| \leq \beta$ for all i. Therefore, $\sum_{i=1}^{\infty} \|h^{i+1}\|^2 \leq \sum_{i=1}^{\infty} \beta \cdot \|h^{i+1}\| < \infty$. By (2.4) and Lemma 2.1 in [1] (or Lemma 2, [18, p. 44]), $\{f(x^i) - f_{\min}\}$ converges. Thus $\{f(x^i) - f(x^{i+1})\}$ converges to 0. Since the right-hand side of (2.3) converges to 0, so does $\|x^i - x^{i+1}\|$. Finally, let x^* be an accumulation point of $\{x^i\}$. It follows from Lemma 2.1 that $x^* = (x^* - (Mx^* + q))_+$; i.e., x^* is a solution of (1.1). This completes the proof of Lemma 2.4. □

Remark. It is clear from the proof that the conclusions of Lemma 2.4 hold under the weaker assumption $\sum_{i=1}^{\infty} \|h^{i+1}\|^2 < \infty$. We only use this lemma as a preliminary result for the proof of convergence of $\{x^i\}$ (cf. Theorem 3.3).

3. Convergence of the matrix splitting algorithm. First we state the key inequality in Luo and Tseng's proof of the linear convergence of $\{x^i\}$ [5], [6]. The proof of the following lemma was implicitly given in the proofs of inequalities (3.18) and (3.19) in [6].

LEMMA 3.1. *There exist α and $r > 0$ such that*

$$f(x^i) - f_{\min} \leq \alpha \cdot (\|x^i - x^{i+1}\| + \|h^{i+1}\|)^2 \quad \textit{for } i \geq r.$$

Based on this lemma and the assumption (1.5), Luo and Tseng proved that the sequence $\{f(x^i)\}$ converges at least linearly, which implies the linear convergence of $\{x^i\}$. Here we want to show that careful manipulations of inequalities could replace the assumption (1.5) by the weaker assumption (1.9). The following lemma is a modification of the proof of Theorem 3.1 in [5].

LEMMA 3.2. *There exist constants $r > 0$, $0 \leq \lambda < 1$, and $\delta > 0$ such that*

$$\sqrt{f(x^{i+1}) - f_{\min}} \leq \lambda \cdot \sqrt{f(x^i) - f_{\min}} + \delta \cdot \|h^{i+1}\| \quad \textit{for } i \geq r.$$

Proof. By Lemma 3.1, there exist α and $r > 0$ such that

$$f(x^i) - f_{\min} \leq \alpha(\|x^{i+1} - x^i\| + \|h^{i+1}\|)^2 \quad \text{for } i \geq r.$$

Now, by Lemma 2.2 and the inequality $(a + b)^2 \leq 2(a^2 + b^2)$, we obtain

$$f(x^i) - f_{\min} \leq \frac{4\alpha}{\gamma}\left(f(x^i) - f(x^{i+1}) + \frac{1}{2}(\gamma^{-1} + \gamma)\|h^{i+1}\|^2\right),$$

which is equivalent to

$$f(x^{i+1}) - f_{\min} \leq \lambda^2 \cdot (f(x^i) - f_{\min}) + \delta^2 \cdot \|h^{i+1}\|^2, \tag{3.1}$$

where $\lambda^2 = 1 - \frac{\gamma}{4\alpha} < 1$ and $\delta^2 = \frac{1}{2}(\gamma^{-1} + \gamma)$. The lemma follows from (3.1) and the inequality $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b \geq 0$. □

THEOREM 3.3. *The sequence $\{x^i\}$ generated by* (1.3) *converges to a solution of* (1.1). *If $\|h^i\| \leq \alpha\theta^i$ with some $\alpha > 0$ and $0 \leq \theta < 1$, then $\{x^i\}$ converges linearly in the root sense.*

Proof. Let $e^i := \sqrt{f(x^i) - f_{\min}}$. It follows from Lemma 3.2 that

$$\sum_{i=r+1}^{m} e^i \leq \lambda \cdot \sum_{i=r}^{m-1} e^i + \delta \cdot \sum_{i=r+1}^{m} \|h^i\| \leq \lambda \cdot e^r + \lambda \cdot \sum_{i=r+1}^{m} e^i + \delta \cdot \sum_{i=r+1}^{m} \|h^i\| \quad \text{for } m > r.$$

Therefore,

$$\sum_{i=r+1}^{m} e^i \leq \frac{\lambda \cdot e^r}{1-\lambda} + \frac{\delta}{1-\lambda} \cdot \sum_{i=r+1}^{m} \|h^i\| \leq \frac{\lambda \cdot e^r}{1-\lambda} + \frac{\delta}{1-\lambda} \cdot \sum_{i=r+1}^{\infty} \|h^i\| < \infty \quad \text{for } m > r,$$

which implies the convergence of $\sum_{i=1}^{\infty} e^i$. Since $|f(x^{i+1}) - f(x^i)|^{1/2} \leq e^i + e^{i+1}$, we have $\sum_{i=1}^{\infty} |f(x^{i+1}) - f(x^i)|^{1/2} < \infty$. By Lemma 2.3, we know that $\sum_{i=1}^{\infty} \|x^i - x^{i+1}\| < \infty$. Thus $\{x^i\}$ is a convergence sequence. This proves the first part of Theorem 3.3.

Now, suppose $\{\|h^i\|\}$ converges linearly in the root sense to 0; i.e., $\|h^i\| \leq \alpha\theta^i$ with $0 \leq \theta < 1$. By Lemma 3.2, one can use the induction method to prove the following inequality:

$$e^i \leq \delta \sum_{j=r+1}^{i} \lambda^{i-j}\|h^j\| + e^r \cdot \lambda^{i-r} \quad \text{for } i > r.$$

Therefore, there is a constant $\beta > 0$ such that

$$|f(x^{i+1}) - f(x^i)|^{1/2} \leq \beta \sum_{j=1}^{i+1} (\lambda^{i+1-j}\|h^j\| + \lambda^i) \quad \text{for } i > r. \tag{3.2}$$

Let $\nu = \max\{\lambda, \theta\} < 1$. It follows from (3.2) that

$$|f(x^{i+1}) - f(x^i)|^{1/2} \leq \beta(1 + \alpha)(i + 1)\nu^i. \tag{3.3}$$

Since there is a constant $\delta > 0$ such that $(i+1)\nu^i \leq \delta\left(\frac{1+\nu}{2}\right)^i$, it follows from Lemma 2.3 and (3.3) that

$$\|x^i - x^{i+1}\| \leq \mu\left(\frac{1+\nu}{2}\right)^i$$

for some constant $\mu > 0$. Thus $\{x^i\}$ converges linearly in the root sense. □

4. Mangasarian's convergence analysis. In this section we give an alternative proof of the convergence of $\{x^i\}$ under the additional assumption that B is symmetric. The alternative proof is based on Mangasarian's convergence analysis and treats the matrix splitting algorithms as power methods. This provides a different perspective of why $\{x^i\}$ converges.

Let $\varphi_h(y)$ denote the solution x of the following LCP:

$$Bx + Cy + q - h \geq 0, \quad x \geq 0, \quad x(Bx + Cy + q - h) = 0.$$

For simplicity, denote $\varphi = \varphi_0$. It is easy to see that any fixed point x^* of φ (i.e., $x^* = \varphi(x^*)$) is a solution of (1.1).

The key step in Mangasarian's proof of the convergence of $\{x^i\}$ is to show that φ is a nonexpansive mapping in the B-norm [10]; i.e.,

$$\|\varphi(x) - \varphi(y)\|_B \leq \|x - y\|_B \quad \text{for } x, y \in \mathbb{R}^n.$$

LEMMA 4.1. *If B is symmetric, then φ is nonexpansive in the B-norm.*

The proof of this lemma is implicitly given in the proof of inequality (2.10) in [10]. If $\|h^i\| = 0$ for all i, then $x^{i+1} = \varphi(x^i)$; i.e., $\{x^i\}$ is generated by the power method with respect to φ. Let x^* be any fixed point of φ. Then

$$\|x^{i+1} - x^*\|_B = \|\varphi(x^i) - \varphi(x^*)\|_B \leq \|x^i - x^*\|_B \quad \text{for } i = 0, 1, 2, \ldots. \tag{4.1}$$

Therefore, $\{x^i\}$ is a bounded sequence. Since any accumulation point of $\{x^i\}$ is a fixed point of φ (cf. Lemma 2.4), we may assume that x^* in (4.1) is an accumulation point of $\{x^i\}$. Then $\{\|x^i - x^*\|_B\}$ is a decreasing sequence and has a subsequence converging to 0. Therefore, $\{\|x^i - x^*\|_B\}$ itself converges to 0. This provides a very simple proof of convergence of iterates generated by the power method for the nonexpansive mapping φ.

Moreover, it is proved by Mangasarian and Shiau [11] that $\varphi_h(x)$ is Lipschitz continuous with respect to h and x. As a consequence, there exists a constant $\alpha > 0$ such that

$$\|\varphi_h(x) - \varphi(x)\|_B \leq \alpha \cdot \|h\|_B \quad \text{for } x, h \in \mathbb{R}^n. \tag{4.2}$$

We could view φ_h as Lipschitz perturbations of φ. Now $\{x^i\}$ generated by (1.3) can be considered as the iterates generated by the following inexact power method for φ:

$$x^{i+1} = \varphi_{h^{i+1}}(x^i), \qquad i = 0, 1, 2, \ldots. \tag{4.3}$$

Then we can show that the iterates generated by the inexact power method with respect to φ converges. Here we do not need any lemmas stated in §§2 and 3.

4.1. Alternative proof of convergence of $\{x^i\}$ when B is symmetric. It follows from Lemma 4.1 and (4.2) that

$$\begin{aligned}
\|x^{i+j}-\varphi^i(x^j)\|_B &= \left\|\sum_{k=1}^{i}(\varphi^{k-1}(x^{i+j-k+1})-\varphi^k(x^{i+j-k}))\right\|_B \\
&= \left\|\sum_{k=1}^{i}(\varphi^{k-1}(\varphi_{h^{i+j-k+1}}(x^{i+j-k}))-\varphi^{k-1}(\varphi(x^{i+j-k})))\right\|_B \\
&\le \sum_{k=1}^{i}\|\varphi^{k-1}(\varphi_{h^{i+j-k+1}}(x^{i+j-k}))-\varphi^{k-1}(\varphi(x^{i+j-k}))\|_B \qquad (4.4)\\
&\le \sum_{k=1}^{i}\|\varphi_{h^{i+j-k+1}}(x^{i+j-k})-\varphi(x^{i+j-k})\|_B \\
&\le \sum_{k=1}^{i}\alpha\cdot\|h^{i+j-k+1}\|_B \le \alpha\sum_{k=1}^{\infty}\|h^{k+j}\|_B,
\end{aligned}$$

where φ^k denotes the composite of φ with itself k times. Let x^* be a solution of (1.1). Then $\varphi(x^*)=x^*$. Hence

$$\|\varphi^i(x^j)-x^*\|_B=\|\varphi^i(x^j)-\varphi^i(x^*)\|_B\le\|x^j-x^*\|_B. \tag{4.5}$$

It follows from (4.4) and (4.5) that

$$\|x^{i+j}-x^*\|_B\le\alpha\sum_{k=1}^{\infty}\|h^{k+j}\|_B+\|x^j-x^*\|_B. \tag{4.6}$$

Since any two norms on $\mathbb{R}^n$ are equivalent and $\sum_{i=1}^{\infty}\|h^i\|<\infty$, we have $\sum_{i=1}^{\infty}\|h^i\|_B<\infty$. Let $j=0$. Then (4.6) implies that $\{x^i\}$ is a bounded sequence. Let x be an accumulation point of $\{x^i\}$. Fix j and let $\{x^{i'+j}\}$ be a subsequence which converges to x. Then the remarks after Lemma 4.1 show that $\{\varphi^{i'}(x^j)\}$ converges to a solution x^* of (1.1). Letting $i'\to\infty$ in (4.4), we obtain

$$\min_{y\in X^*}\|x-y\|_B\le\|x-x^*\|_B\le\alpha\cdot\sum_{k=1}^{\infty}\|h^{k+j}\|_B, \tag{4.7}$$

where X^* is the set of solutions of (1.1). Since $\lim_{j\to\infty}\sum_{k=1}^{\infty}\|h^{k+j}\|_B=0$ and (4.7) holds for any $j\ge1$, we get $\min_{y\in X^*}\|x-y\|_B=0$ by letting $j\to\infty$ in (4.7). This implies $x\in X^*$ (i.e., x is a solution of (1.1)). Thus, any accumulation point x of $\{x^i\}$ is a solution of (1.1), i.e., $x=\varphi(x)$. Therefore, we may assume that x^* in (4.6) is an accumulation point of $\{x^i\}$. Let $\epsilon>0$ be any positive number. Since $\sum_{i=1}^{\infty}\|h^i\|_B<\infty$, there is an integer $r>0$ such that

$$\alpha\sum_{k=1}^{\infty}\|h^{k+j}\|_B<\epsilon/2\quad\text{for } j\ge r. \tag{4.8}$$

Since x^* is an accumulation point of $\{x^i\}$, there is $j>r$ such that $\|x^j-x^*\|_B<\epsilon/2$. Then it follows from (4.6) and (4.8) that

$$\|x^{i+j}-x^*\|_B<\epsilon\quad\text{for } i\ge1.$$

This proves that $\{x^i\}$ converges to a solution of (1.1). □

Remark. With the exception of Lemma 4.1, the above convergence analysis is self-contained. Moreover, the proof shows that, if φ_h is any perturbation of φ that satisfies

$$\|\varphi_{h^{i+1}}(x^i) - \varphi(x^i)\|_B \leq \alpha \cdot \|h^{i+1}\|_B$$

and $\{h^i\}$ satisfies (1.9), then the sequence $\{x^i\}$ generated by (4.3) converges to a solution of (1.1). This fact could be very useful in practice. For example, let x^{i+1} be an inexact solution of the following problem:

$$Bx + Cx^i + q \geq 0, \quad x \geq 0, \quad x(Bx + Cx^i + q) = 0, \quad i = 0, 1, \ldots,$$

where x^0 is given. If we consider x^{i+1} as a solution of (4.3) for some $\varphi_{h^{i+1}}$, then there exists $\alpha > 0$, depending only on B such that (cf. [12, Lemma 2])

$$\|\varphi_{h^{i+1}}(x^i) - \varphi(x^i)\|_B = \|x^{i+1} - \varphi(x^i)\|_B \leq \alpha \cdot \|\min\{Bx^{i+1} + Cx^i + q, x^{i+1}\}\|_B,$$

where $h^{i+1} := \min\{Bx^{i+1} + Cx^i + q, x^{i+1}\}$ and $\min\{y, z\}$ denotes the vector in $\mathbb{R}^n$ whose ith component is $\min\{y_i, z_i\}$. Therefore, if $\sum_{i=1}^{\infty} \|\min\{Bx^{i+1} + Cx^i + q, x^{i+1}\}\| < \infty$, then the sequence $\{x^i\}$ converges to a solution of (1.1). Notice that x^i might have negative components here, while the feasibility of x^i is crucial in Luo and Tseng's convergence analysis (cf. the proofs given in §§2 and 3). Therefore, the alternative proof gives more than a different proof of the convergence of $\{x^i\}$ generated by (1.3) under the assumption that B is symmetric.

5. Comments.

The results and methods used in this paper can be easily extended to the symmetric affine variational inequality problem associated with a symmetric matrix M, a vector q, and a convex polyhedral subset X of $\mathbb{R}^n$ (cf. [6] and references therein):

$$\text{find an } x^* \in X \text{ satisfying } (x - x^*)(Mx^* + q) \geq 0 \quad \text{for } x \in X.$$

One can easily modify the results and their proofs in [5] and [6], as we did in §3, to eliminate the assumption (1.5) on h^i. If we assume further that M is positive semidefinite and B is symmetric, then the corresponding matrix splitting method induces a nonexpansive mapping φ in the B-norm. The proof in §4 also works.

Acknowledgments. I am in debt to Prof. Paul Tseng for our many stimulating discussions on the subject. Also, I would like to thank Prof. Olvi Mangasarian for his instructive suggestions and comments.

REFERENCES

[1] Y. C. Cheng, *On the gradient-projection method for solving the nonsymmetric linear complementarity problem*, J. Optim. Theory Appl., 43 (1984), pp. 527–541.

[2] C. W. Cryer, *The solution of a quadratic programming problem using systematic overrelaxation*, SIAM J. Comput., 9 (1971), pp. 385–392.

[3] H. B. Keller, *On the solution of singular and semidefinite linear systems by iteration*, SIAM J. Numer. Anal., 2 (1965), pp. 281–290.

[4] Y. Y. Lin and J.-S. Pang, *Iterative methods for large quadratic programs: A survey*, SIAM J. Control Optim., 25 (1987), pp. 383–411.

[5] Z.-Q. Luo and P. Tseng, *On the linear convergence of descent methods for convex essentially smooth minimization*, J. Optim. Theory Appl., 72 (1992), pp. 7–35.

[6] 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.

[7] ———, *On global error bounds for a class of monotone affine variational inequality problems*, Oper. Res. Lett., 11 (1992), to appear.

[8] ———, *On the convergence of a matrix splitting algorithm for the symmetric monotone linear complementarity problem*, SIAM J. Control Optim., 29 (1991), pp. 1037–1060.

[9] O. L. MANGASARIAN, *Solution of symmetric linear complementarity problems by iterative methods*, J. Optim. Theory Appl., 22 (1977), pp. 465–485.

[10] ———, *Convergence of iterates of an inexact matrix splitting algorithm for the symmetric monotone linear complementarity problem*, SIAM J. Optim., 1 (1991), pp. 114–122.

[11] 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.

[12] R. MATHIAS AND J.-S. PANG, *Error bounds for the linear complementarity problem with a P-matrix*, Linear Algebra Appl., 132 (1990), pp. 123–136.

[13] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[14] J.-S. PANG, *On the convergence of a basic iterative method for the implicit complementarity problem*, J. Optim. Theory Appl., 37 (1982), pp. 149–162.

[15] ———, *Necessary and sufficient conditions for the convergence of iterative methods for the linear complementarity problem*, J. Optim. Theory Appl., 42 (1984), pp. 1–17.

[16] ———, *More results on the convergence of iterative methods for the symmetric linear complementarity problems*, J. Optim. Theory Appl., 49 (1986), pp. 107–134.

[17] J.-S. PANG AND J.-M. YANG, *Two-stage parallel iterative methods for the symmetric linear complementarity problem*, Ann. Oper. Res., 14 (1988), pp. 61–75.

[18] B. T. POLYAK, *Introduction to Optimization*, Optimization Software, New York, 1987.

SIAM J. OPTIMIZATION
Vol. 3, No. 1, pp. 164–221, February 1993

GLOBAL CONVERGENCE OF A CLASS OF TRUST REGION ALGORITHMS FOR OPTIMIZATION USING INEXACT PROJECTIONS ON CONVEX CONSTRAINTS*

A. R. CONN[†], NICK GOULD[‡], A. SARTENAER[§], AND PH. L. TOINT[¶]

Abstract. A class of trust region-based algorithms is presented for the solution of nonlinear optimization problems with a convex feasible set. At variance with previously published analyses of this type, the theory presented allows for the use of general norms. Furthermore, the proposed algorithms do not require the explicit computation of the projected gradient, and can therefore be adapted to cases where the projection onto the feasible domain may be expensive to calculate. Strong global convergence results are derived for the class. It is also shown that the set of linear and nonlinear constraints that are binding at the solution are identified by the algorithms of the class in a finite number of iterations.

Key words. trust region methods, projected gradients, convex constraints

AMS(MOS) subject classifications. 90C30, 65K05

1. Introduction.

Trust region methods for nonlinear optimization problems have become very popular over the last decade. One possible explanation of their success is their remarkable numerical reliability associated with the existence of a sound and complete convergence theory. The fact that they efficiently handle nonconvex problems has also been considered an advantage.

As an integral part of this growing interest, research in convergence theory for this class of methods has been very active. First, a substantial body of theory was built for the unconstrained case (see [19] for an excellent survey). Problems involving bound constraints on the variables were then considered (see [1], [9], and [21]), as well as the more general case where the feasible region is a convex set on which the projection (with respect to the Euclidean norm) can be computed at a reasonable cost (see [4], [20], and [29]). The studied techniques are based on the use of the explicitly calculated projected gradient as a tool to predict which of the inequality constraints are binding at the problem's solution. Moreover, trust region methods for nonlinear equality constraints have also been studied by several authors (see, e.g., [5], [8], [25], and [30]).

This paper also considers the case where the feasible set is convex. It presents a convergence theory for a class of trust region algorithms with the following new features.

- The theory does not depend on the explicit use of the projection operator in the Euclidean norm, but allows for the use of a uniformly equivalent family of arbitrary norms.
- The gradient of the objective function can be approximated if its exact value is either impossible or too costly to compute at every iteration.

[*] Received by the editors April 19, 1990; accepted for publication (in revised form) January 9, 1992. Travel support for this research was provided by NATO grant 890867.

[†] IBM T. J. Watson Research Centre, P.O. Box 218, Yorktown Heights, New York 10598-0218. This work was supported in part by Natural Sciences and Engineering Research Council of Canada grant A8639.

[‡] Rutherford Appleton Laboratory, Chilton, Oxfordshire, OX11 0QX England.

[§] Facultés Universitaires ND de la Paix, 61 rue de Bruxelles, B-5000, Namur, Belgium. This work was supported by the Belgian National Fund for Scientific Research.

[¶] Department of Mathematics, Facultés Universitaires ND de la Paix, 61 rue de Bruxelles, B-5000, Namur, Belgium.

- The calculation of the "projected gradient" (with respect to the chosen norms) need not be carried out to full accuracy.
- When the feasible set is described by a system of linear and/or nonlinear (in)equalities, conditions are presented that guarantee that the algorithms of the class identify, in a finite number of iterations, the set of inequalities that are binding at the solution. We note that this description of the feasible set does not need its partition into faces.

In this sense, we see that our theory applies to problems similar to those considered in [4], [9], [20], and [29], although in a more general setting.

An attractive aspect of this theory is that it covers the case where a polyhedral norm is chosen to define an analog of the projection operator, allowing the use of linear (or convex) programming methods for the approximate calculation of the projected gradients. This type of algorithm should be especially efficient in the frequent situation where the feasible set is defined by a set of linear equalities and inequalities, and where a basis for the nullspace of the matrix of the active constraints is cheaply available. In network problems, for example, this can be very cheaply obtained and updated using a spanning tree of the problem's underlying graph (see [17] for a detailed presentation of the relevant algorithms). Other examples include multiperiodic operation research models resulting in staircase matrices.

The problem and notation are introduced in §2, together with a general class of algorithms. The convergence properties of this class are then analyzed in §3. A particular practical algorithm of the class is discussed in §4. The identification of the active constraints is presented in §5. Section 6 presents an analysis of the conditions under which the whole sequence of iterates can be shown to converge to a single limit point. Additional points and extensions of the theory are discussed in §7. A glossary of symbols can be found in Appendix B. All the assumptions used in the paper are finally summarized in Appendix C.

2. A class of trust region algorithms for problems with convex feasible domain.

2.1. The problem.

The problem we consider is that of finding a local solution of

$$\min f(x) \tag{2.1}$$

subject to the constraint

$$x \in X, \tag{2.2}$$

where x is a vector of $\mathbf{R}^n$, $f(\cdot)$ is a smooth function from $\mathbf{R}^n$ into $\mathbf{R}$ and X is a nonempty closed convex subset of $\mathbf{R}^n$, also called the *feasible set.* We assume that we can compute the function value $f(x)$ for any feasible point x. We are also given a feasible starting point x_0, and we wish to start the minimization procedure from this point.

If we define $\mathcal{L}$ by

$$\mathcal{L} \stackrel{\text{def}}{=} X \cap \{x \in \mathbf{R}^n \mid f(x) \leq f(x_0)\}, \tag{2.3}$$

we may formulate our assumptions on the problem as follows.

AS.1. The set $\mathcal{L}$ is compact.

AS.2. The objective function $f(x)$ is continuously differentiable and its gradient $\nabla f(x)$ is Lipschitz continuous in an open domain containing $\mathcal{L}$.

In particular, we allow for unbounded X, provided the set $\mathcal{L}$ remains bounded.

We will denote by $\langle\cdot,\cdot\rangle$ the Euclidean inner product on $\mathbf{R}^n$ and by $\|\cdot\|_2$ the associated ℓ_2-norm.

We recall that a subset K of $\mathbf{R}^n$ is a cone if it is closed under positive scalar multiplication, that is, if $\lambda x \in K$ whenever $x \in K$ and $\lambda > 0$ (see [26, p. 13]). Given a cone K, one can define its *polar* (see [26, p. 121]) as

$$K^0 \stackrel{\text{def}}{=} \{y \in \mathbf{R}^n | \, \langle y, u\rangle \leq 0, \forall u \in K\} \tag{2.4}$$

and verify that K^0 is also a cone, and that $(K^0)^0 = K$ when K is a nonempty closed convex cone.

Given the closed convex set X, we can define $P_X(x)$, the *projection* of the vector $x \in \mathbf{R}^n$ onto X, as the unique minimizer of the problem

$$\min_{y\in X} \|y - x\|_2. \tag{2.5}$$

This projection operator is well known and has been much studied (see, e.g., [33]). We will also denote by $N(x)$ the *normal cone* of X at $x \in X$; that is,

$$N(x) \stackrel{\text{def}}{=} \{y \in \mathbf{R}^n \mid \langle y, u - x\rangle \leq 0, \ \forall u \in X\}. \tag{2.6}$$

The *tangent cone* of X at $x \in X$ is the polar of the normal cone at the same point; that is,

$$T(x) \stackrel{\text{def}}{=} N(x)^0 = \text{cl}\{\lambda(u - x) | \lambda \geq 0 \ \text{ and } \ u \in X\}, \tag{2.7}$$

where cl$\{S\}$ denotes the closure of the set S. We will also use the *Moreau decomposition* given by the identity

$$x = P_{T(y)}(x) + P_{N(y)}(x), \tag{2.8}$$

which is valid for all $x \in \mathbf{R}^n$ and all $y \in X$ (see [22]). This decomposition is illustrated in Fig. 1. In this figure and all subsequent ones, the boundary of the feasible set X is drawn with a bold line.

We conclude this subsection with a result extracted from the classical perturbation theory of convex optimization problems. This result is well known and can be found in [14, pp. 14–17] for example.

LEMMA 2.1. *Assume that D is a continuous point-to-set mapping from $S \subseteq \mathbf{R}^\ell$ into $\mathbf{R}^n$ such that the set $D(\epsilon)$ is convex and nonempty for each $\epsilon \in S$. Assume also that one is given a real-valued function $F(y,\epsilon)$, which is defined and continuous on the space $\mathbf{R}^n \times S$ and convex in y for each fixed ϵ. Then, the real-valued function F_* defined by*

$$F_*(\epsilon) \stackrel{\text{def}}{=} \inf_{y\in D(\epsilon)} F(y,\epsilon) \tag{2.9}$$

and the solution set mapping y_* defined by

$$y_*(\epsilon) \stackrel{\text{def}}{=} \{y \in D(\epsilon) | F(y,\epsilon) = F_*(\epsilon)\} \tag{2.10}$$

are both continuous on S.

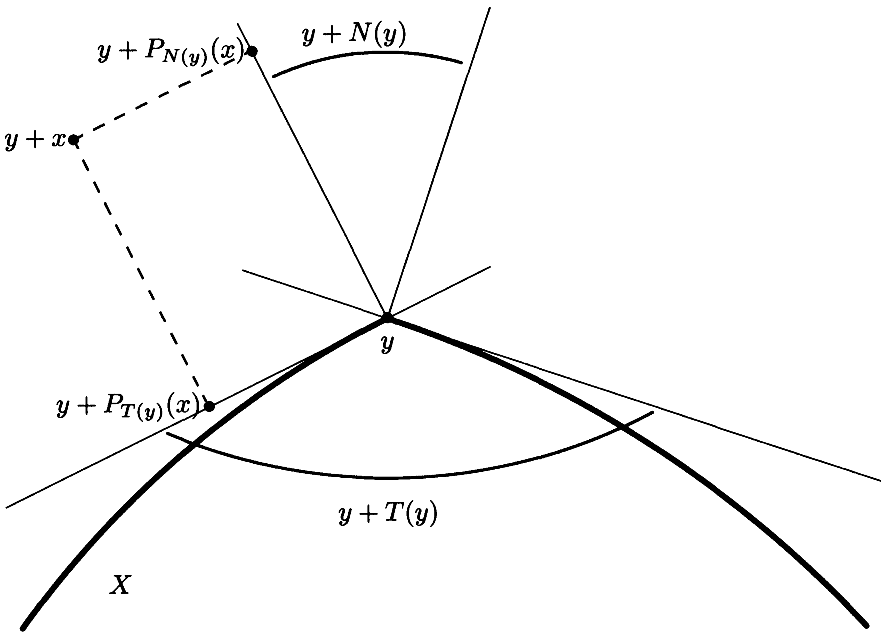


FIG. 1. *The normal and tangent cones at* y, *and the corresponding Moreau decomposition of* x (*translated to* y).

2.2. Defining a local model of the objective function. The algorithm we propose for solving (2.1) subject to the constraint (2.2) is iterative and of trust region type. Indeed, at each iteration, we define a *model* of the objective function $f(x)$, and a region surrounding the current iterate, say x_k, where we believe this model to be adequate. The algorithm then finds, in this region, a candidate for the next iterate that sufficiently reduces the value of the model of the objective. If the function value calculated at this point matches its predicted value closely enough, the new point is then accepted as the next iterate and the trust region is possibly enlarged; otherwise the point is rejected and the trust region size decreased. With each iteration of our algorithm will be associated a norm: we will denote by $\|\cdot\|_{(k)}$ the norm associated with the kth iteration.

We now specify the conditions we impose on the model of the objective function. This model, defined in a neighbourhood of the kth iterate x_k, is denoted by the symbol m_k and is meant to approximate the objective f in the *trust region*

$$B_k \overset{\text{def}}{=} \{x \in \mathbf{R}^n | \, \|x - x_k\|_{(k)} \leq \nu_1 \Delta_k\}, \tag{2.11}$$

where ν_1 is a positive constant and $\Delta_k > 0$ is the *trust region radius*. We will assume that m_k is differentiable and has Lipschitz continuous first derivatives in an open set containing B_k, that

$$m_k(x_k) = f(x_k), \tag{2.12}$$

and that $g_k \overset{\text{def}}{=} \nabla m_k(x_k)$ approximates $\nabla f(x_k)$ in the following sense: there exists a nonnegative constant κ_1 such that the inequality

$$\|e_k\|_{[k]} \le \kappa_1 \Delta_k \tag{2.13}$$

holds for all k, where the error e_k is defined by $e_k \overset{\text{def}}{=} g_k - \nabla f(x_k)$ and where the norm $\|\cdot\|_{[k]}$ is any norm that satisfies

$$|\langle x, y\rangle| \le \|x\|_{(k)} \|y\|_{[k]} \tag{2.14}$$

for all $x, y \in \mathbf{R}^n$. In particular, one can choose the *dual norm* of $\|\cdot\|_{(k)}$ defined by

$$\|y\|_{[k]} \overset{\text{def}}{=} \sup_{x \ne 0} \frac{|\langle x, y\rangle|}{\|x\|_{(k)}}. \tag{2.15}$$

Condition (2.13) is quite weak, as it merely requires that the first-order information on the objective function be reasonably accurate whenever a short step must be taken. Indeed, one expects this first-order behaviour to dominate for small steps.

Clearly, for the above conditions to be coherent from one iteration to the next, we need to assume some relationship between the various norms that we introduced. More precisely, we will assume that all these norms are *uniformly equivalent* in the following sense.

AS.3. There exist constants $\sigma_1, \sigma_3 \in (0, 1]$ and $\sigma_2, \sigma_4 \ge 1$ such that, for all $k_1 \ge 0$ and $k_2 \ge 0$,

$$\sigma_1 \|x\|_{(k_1)} \le \|x\|_{(k_2)} \le \sigma_2 \|x\|_{(k_1)} \tag{2.16}$$

and

$$\sigma_3 \|x\|_{[k_1]} \le \|x\|_{[k_2]} \le \sigma_4 \|x\|_{[k_1]} \tag{2.17}$$

for all $x \in \mathbf{R}^n$.

If (2.15) is chosen, then (2.17) immediately results from (2.16) with $\sigma_3 = 1/\sigma_2$ and $\sigma_4 = 1/\sigma_1$.

We also note that (2.16) and (2.17) necessarily hold if the norms $\|\cdot\|_{(k_2)}$ and $\|\cdot\|_{[k_2]}$ are replaced by the ℓ_2-norm.

We finally introduce, for given k and for any nonnegative t, the quantity $\alpha_k(t) \ge 0$ given by

$$\alpha_k(t) \overset{\text{def}}{=} \Big| \min_{\substack{x_k + d \in X \\ \|d\|_{(k)} \le t}} \langle g_k, d\rangle \Big|, \tag{2.18}$$

that is, the magnitude of the maximum decrease of the linearized model achievable on the intersection of the feasible domain with a ball of radius t (in the norm $\|\cdot\|_{(k)}$) centred at x_k.

We note here that $\alpha_k(t)$ can be defined using the notion of support function of the convex set $\{d | x_k + d \in X \text{ and } \|d\|_{(k)} \le t\}$. The properties that follow can then be derived in this framework. We have, however, chosen to use the more familiar vocabulary of classical optimization in order to avoid further prerequisites in convex analysis.

We then have the following simple properties.

LEMMA 2.2. *For all $k \ge 0$,*

(1) *the function* $t \mapsto \alpha_k(t)$ *is continuous and nondecreasing for* $t \geq 0$,
(2) *the function* $t \mapsto \alpha_k(t)/t$ *is nonincreasing for* $t > 0$,
(3) *the inequality*

$$\frac{\alpha_k(t)}{t} \leq \|P_{T(x_k)}(-g_k)\|_{[k]} \tag{2.19}$$

holds for all $t > 0$.

Proof. The first statement is an immediate consequence of the definition (2.18) and of Lemma 2.1 applied to the optimization problem of (2.18). In order to prove the second statement, consider $0 < t_1 < t_2$ and two vectors d_1 and d_2 such that

$$\alpha_k(t_1) = -\langle g_k, d_1 \rangle, \quad \|d_1\|_{(k)} \leq t_1, \quad x_k + d_1 \in X, \tag{2.20}$$

and

$$\alpha_k(t_2) = -\langle g_k, d_2 \rangle, \quad \|d_2\|_{(k)} \leq t_2, \quad x_k + d_2 \in X. \tag{2.21}$$

We observe that the point $x_k + (t_1/t_2)d_2$ lies between x_k and $x_k + d_2$, and therefore we have that $x_k + (t_1/t_2)d_2 \in X$. Furthermore,

$$\left\| \frac{t_1}{t_2} d_2 \right\|_{(k)} = \frac{t_1}{t_2} \|d_2\|_{(k)} \leq t_1 \tag{2.22}$$

and the point $x_k + (t_1/t_2)d_2$ thus lies in the feasible domain of the optimization problem associated with the definition of $\alpha_k(t_1)$ and d_1. As a consequence, we have that

$$\frac{\alpha_k(t_1)}{t_1} \geq \frac{1}{t_1} \left| \left\langle g_k, \frac{t_1}{t_2} d_2 \right\rangle \right| = \frac{\alpha_k(t_2)}{t_2}, \tag{2.23}$$

and the second statement of the lemma is proved.

The third statement is proved as follows. Applying the Moreau decomposition to $-g_k$, we obtain that, for any d such that $x_k + d \in X$ and $\langle g_k, d \rangle \leq 0$,

(2.24)
$$\langle g_k, d \rangle = -\left\langle P_{T(x_k)}(-g_k), d \right\rangle - \left\langle P_{N(x_k)}(-g_k), P_{T(x_k)} d \right\rangle \geq -\left\langle P_{T(x_k)}(-g_k), d \right\rangle,$$

where we used the fact that $d \in T(x_k)$ and the fact that the tangent cone is the polar of the normal cone to derive the last inequality. Taking absolute values and applying (2.14) thus yields that

$$|\langle g_k, d \rangle| \leq \|d\|_{(k)} \|P_{T(x_k)}(-g_k)\|_{[k]}. \tag{2.25}$$

We then obtain (2.19) by applying this inequality to any solution d of the optimization problem associated with the definition of $\alpha_k(t)$ in (2.18) and using the fact that $\|d\|_{(k)} \leq t$. □

2.3. A class of trust region algorithms. We are now ready to define our first algorithm in more detail. Besides κ_1 as used in (2.13), it depends on the constants

$$0 < \mu_1 < \mu_2 < 1, \quad \mu_3 \in (0,1], \quad \mu_4 \in (0,1], \tag{2.26}$$

$$0 < \nu_3 < \nu_2 \leq \nu_1, \quad \nu_4 \in (0,1], \tag{2.27}$$

$$0 < \eta_1 < \eta_2 < 1, \tag{2.28}$$

and

$$0 < \gamma_1 \le \gamma_2 < 1 < \gamma_3. \tag{2.29}$$

ALGORITHM 1.

Step 0. Initialization. The starting point x_0 is given, together with $f(x_0)$ and an initial trust region radius $\Delta_0 > 0$. Set $k = 0$.

Step 1. Model choice. Choose m_k, a model of the objective function f in the trust region B_k centred at x_k, satisfying (2.12) and (2.13).

Step 2. Determination of a *generalized Cauchy point* (GCP). If $\alpha_k \overset{\text{def}}{=} \alpha_k(1) = 0$, stop. Else, find a vector s_k^C such that, for some strictly positive $t_k \ge \|s_k^C\|_{(k)}$,

$$x_k + s_k^C \in X, \tag{2.30}$$

$$\|s_k^C\|_{(k)} \le \nu_2 \Delta_k, \tag{2.31}$$

$$\langle g_k, s_k^C \rangle \le -\mu_3 \alpha_k(t_k), \tag{2.32}$$

$$m_k(x_k + s_k^C) \le m_k(x_k) + \mu_1 \langle g_k, s_k^C \rangle, \tag{2.33}$$

and either

$$t_k \ge \min[\nu_3 \Delta_k, \nu_4] \tag{2.34}$$

or

$$m_k(x_k + s_k^C) \ge m_k(x_k) + \mu_2 \langle g_k, s_k^C \rangle. \tag{2.35}$$

Set the GCP

$$x_k^C = x_k + s_k^C. \tag{2.36}$$

Step 3. Determination of the step. Find a vector s_k such that

$$x_k + s_k \in X \cap B_k \tag{2.37}$$

and

$$m_k(x_k) - m_k(x_k + s_k) \ge \mu_4[m_k(x_k) - m_k(x_k^C)]. \tag{2.38}$$

Step 4. Determination of the model accuracy. Compute $f(x_k + s_k)$ and

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}. \tag{2.39}$$

Step 5. Trust region radius updating. In the case where

$$\rho_k > \eta_1, \tag{2.40}$$

set

$$x_{k+1} = x_k + s_k \tag{2.41}$$

and

$$\Delta_{k+1} \in [\Delta_k, \gamma_3 \Delta_k] \quad \text{if } \rho_k \geq \eta_2, \tag{2.42}$$

or

$$\Delta_{k+1} \in [\gamma_2 \Delta_k, \Delta_k] \quad \text{if } \rho_k < \eta_2. \tag{2.43}$$

Otherwise, set

$$x_{k+1} = x_k \tag{2.44}$$

and

$$\Delta_{k+1} \in [\gamma_1 \Delta_k, \gamma_2 \Delta_k]. \tag{2.45}$$

Step 6. Loop. Increment k by one and go to Step 1.

Of course, this only describes a relatively abstract algorithmic class. In particular, we note the following:

1. We have not been very specific about the model m_k to be used in the trust region. In fact, we have merely stated that its value should coincide with that of the objective at the current iterate, and that its gradient at this point should approximate the gradient of the objective at the same point. We will also impose additional necessary assumptions on its curvature in order to derive the desired convergence results. This still remains very broad and requires further specification for any practical implementation of the algorithm.

One very common model choice for a twice differentiable f is to use a quadratic of the form

$$m_k(x_k + s) = f(x_k) + \langle \nabla f(x_k), s \rangle + \tfrac{1}{2} \langle s, H_k s \rangle, \tag{2.46}$$

where H_k is a symmetric approximation to $\nabla^2 f(x_k)$. In particular, Newton's method corresponds to (2.46) with the choice of $H_k = \nabla^2 f(x_k)$.

Another interesting choice is

$$m_k(x_k + s) = f(x_k + s), \tag{2.47}$$

that is, the model and the objective must coincide on $X \cap B_k$. In that case, ρ_k will always be exactly one, and the trust region size Δ_k may be assumed to be very large. We then obtain a convergence theory of an algorithm which is no longer a trust region method in the classical sense. In particular, if the step s_k is determined by a linesearch procedure (see [1] and [29]), the present theory then covers both linesearch and trust region algorithms in a single context.

2. When $k = 0$ or $x_k \neq x_{k-1}$ or $\Delta_k < \Delta_{k-1}$, the definition of the model m_k at Step 1 and the condition that (2.13) be satisfied may require the computation of a new sufficiently accurate approximate gradient g_k.

3. We now briefly motivate the conditions (2.30)–(2.35). Our main idea is to avoid the repeated computation of the projection onto the feasible set X within the GCP calculation, which is a convex *nonlinear* program. Instead, we allow the repeated solution of convex *linear* programs. Furthermore, these linear programs need not be solved to full accuracy. These two relaxations may indeed allow for a substantially reduced amount of calculation. We have in mind the particular case where X is a polyhedral set and $\|\cdot\|_{(k)}$ is polyhedral for all k.

Condition (2.30) is imposed because we want our algorithm only to generate feasible points. This may be essential when some constraints are "hard," for instance, when the objective function is undefined outside X.

Condition (2.31) simply requires the step to be inside a ball contained in the trust region defined by (2.11). This is intended to leave some freedom for the calculation of s_k in Step 3, even when the GCP is on the boundary of that smaller ball.

Condition (2.32) introduces the desired relaxations, while relating the definition of x_k^C to that of a point along the projected gradient path

$$x_k(\theta) = P_X(x_k - \theta g_k) \qquad (\theta \geq 0). \tag{2.48}$$

Indeed, it can be shown that, if $\mu_3 = 1$ and $\|\cdot\|_{(k)} = \|\cdot\|_2$, then x_k^C achieves the same reduction in the linearized model as that obtained by the unique point $x_k(\theta_k)$ on the projected gradient path (2.48) having length t_k, if such a point exists. Condition (2.32) with $\mu_3 < 1$ can therefore be interpreted as a weakening of the condition (for example, required in [9], [21], and [29]) that x_k^C should be on the projected gradient path. This weakening is of great practical interest when the projection onto the feasible domain X is not readily computable.

An example is shown in Fig. 2 using the ℓ_∞-norm, where the set of admissible steps s_k^C is represented by the shaded area, and where (2.32) with $\mu_3 = 1$ is achieved for the step $d_k(t_k)$.

Conditions (2.33) and (2.35) are in the spirit of the classical Goldstein conditions for a "projected search" on the model along the approximation of the projected gradient path implicitly defined by varying t_k. This projected search is similar to that introduced in [29] and modified in [20]. Condition (2.34) completes (2.33) and (2.35) by allowing the search to terminate with a point that sufficiently reduces the model m_k while having a length comparable to the trust region radius.

We note here that the value of t_k is never used by Algorithm 1 except in the definition of s_k^C. It is unnecessary to explicitly define its numerical value, provided its existence is guaranteed for the computed s_k^C. We note also that condition (2.32) implies that both s_k^C and the denominator of (2.39) are nonzero.

The vector x_k^C in (2.36) is called a GCP because it plays a role similar to that of the GCP in [4], [9], [20], and [29].

At this stage, it is far from obvious how a vector s_k^C satisfying the conditions of Step 2 can be computed. The existence and computation of a suitable step will be addressed in §§4 and 7.1.

4. Again, much freedom is left in the calculation of the step s_k in Step 3, but this fairly broad outline is sufficient for our analysis. However, this freedom is crucial in practical implementations, as it allows a refinement of the GCP step based on second-order information, hence providing a possibly fast ultimate rate of convergence.

5. Only a theoretical stopping rule has been specified at the beginning of Step 2. (This criterion will be justified in §3.) Of course, any practical algorithm in our class must use a more practical test, which may depend on the particular class of models

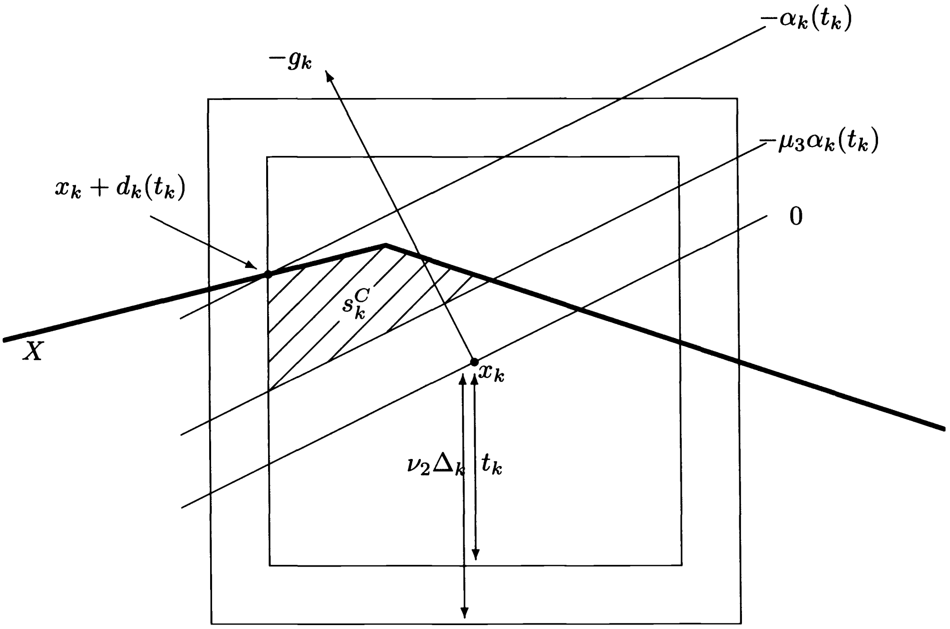


FIG. 2. *An illustration of condition* (2.33) *using the* ℓ_∞*-norm.*

being used. The present hypothesis is, however, natural in our context, where we want to analyze the behaviour of the algorithm as k tends to infinity. We will therefore assume in the sequel that the test at the beginning of Step 2 is never triggered.

6. From the practical point of view, it may be unrealistic to let the trust region radius Δ_k grow to infinity, and most implementations do impose a uniform upper bound on these radii. This is coherent with (2.42), where a strict increase of Δ_k is not required.

7. The condition (2.45) may seem inappropriate when $\|s_k\|_{(k)}$ is small compared with the trust region radius Δ_k. Analogously to the observation in [29], this condition may be replaced by the more practical

$$\Delta_{k+1} \in [\min(\gamma_0\|s_k\|_{(k)}, \gamma_1\Delta_k), \gamma_2\Delta_k] \tag{2.49}$$

for some $\gamma_0 \in (0, 1]$ without modifying the theory presented below.

8. The algorithm necessarily depends on several constants. Typical values for some of them are $\mu_1 = 0.1$, $\mu_2 = 0.9$, $\mu_4 = 1$, $\nu_1 = 1$, $\nu_3 = 10^{-5}$, $\nu_4 = 0.01$, $\eta_1 = 0.25$, $\eta_2 = 0.75$, $\gamma_1 = 0.01$, $\gamma_2 = \frac{1}{2}$, and $\gamma_3 = 2$. Suitable values for the remaining constants will only become clear after extensive testing.

We call an iteration of the algorithm *successful* if the test (2.40) is satisfied, that is when the achieved objective reduction $f(x_k) - f(x_k + s_k)$ is large enough compared to the reduction $m_k(x_k) - m_k(x_k + s_k)$ predicted by the model. If (2.40) fails, the iteration is said to be *unsuccessful*. In what follows, the set of indices of successful iterations will be denoted by $\mathcal{S}$.

3. Global convergence for Algorithm 1.

3.1. Criticality measures. If we are to prove that the iterates generated by Algorithm 1 converge to critical points for the problem (2.1)–(2.2), we clearly must specify how we will measure the "criticality" of a given feasible point. We say that a feasible point x_* is *critical* (or *stationary*) if and only if

$$-\nabla f(x_*) \in N(x_*). \tag{3.1}$$

We propose to use, as a measure of criticality, the quantity

$$\alpha_k[x] \overset{\text{def}}{=} \Big| \min_{\substack{x+d\in X\\ \|d\|_{(k)}\leq 1}} \langle \nabla f(x), d\rangle \Big|, \tag{3.2}$$

which can be interpreted as the magnitude of the maximum decrease of the *linearized objective function* achievable in the intersection of X with a ball of radius one (in the norm $\|\cdot\|_{(k)}$) centred at x. Observe that $\alpha_k[x]$ reduces to $\|\nabla f(x)\|_2$ when $X = \mathbf{R}^n$ and $\|\cdot\|_{(k)} = \|\cdot\|_2$.

LEMMA 3.1. *Assume that* AS.2 *holds. Then, for all $k \geq 0$, $\alpha_k[\cdot]$ is continuous with respect to its argument.*

Proof. The continuity of $\alpha_k[\cdot]$ with respect to its argument is a direct consequence of Lemma 2.1 and of the continuity of $\nabla f(x)$. □

We now show that all the norms $\|\cdot\|_{(k)}$ are formally equivalent.

THEOREM 3.2. *Assume that* AS.2 *and* AS.3 *hold. Then there exists a positive constant $c_1 \geq 1$ such that*

$$\frac{1}{c_1}\alpha_{k_1}[x] \leq \alpha_{k_2}[x] \leq c_1\alpha_{k_1}[x] \tag{3.3}$$

for all $x \in X$ and all $k_1 \geq 0$ and $k_2 \geq 0$.

Proof. We first observe that, using assumption AS.3,

$$\|d\|_{(k)} = 1 \Longrightarrow \sigma_1 \leq \|d\|_2 \leq \sigma_2. \tag{3.4}$$

The lower (respectively, upper) bound in this last inequality represents the smallest (respectively, largest) possible distance (induced by $\|\cdot\|_2$) between x and the boundary of any ball, $\|d\|_{(k)} = 1$, for $k \geq 0$. The ball $\{x + d \mid \|d\|_2 \leq \sigma_2\}$ then contains all the balls of the form

$$\|d\|_{(k)} \leq 1, \tag{3.5}$$

while the ball $\{x + d \mid \|d\|_2 \leq \sigma_1\}$ is contained in them all. Now consider

$$\alpha_{\max} \overset{\text{def}}{=} \Big| \min_{\substack{x+d\in X\\ \|d\|_2\leq\sigma_2}} \langle \nabla f(x), d\rangle \Big| \quad\text{and}\quad \alpha_{\min} \overset{\text{def}}{=} \Big| \min_{\substack{x+d\in X\\ \|d\|_2\leq\sigma_1}} \langle \nabla f(x), d\rangle \Big|. \tag{3.6}$$

Because of the second part of Lemma 2.2 (with $x_k = x$, $g_k = \nabla f(x)$ and $\|\cdot\|_{(k)} = \|\cdot\|_2$), we deduce that

$$\alpha_{\max} \leq \frac{\sigma_2}{\sigma_1}\alpha_{\min}. \tag{3.7}$$

Having established this property, we now return to the proof of Theorem 3.2 itself. If $\alpha_{k_1}[x] = \alpha_{k_2}[x]$, then (3.3) is trivially satisfied. We thus only consider the case where, say,

$$\alpha_{k_1}[x] < \alpha_{k_2}[x]. \tag{3.8}$$

In this situation, we will show that both d_1 and d_2, two vectors satisfying the relations

$$\alpha_{k_1}[x] = -\langle \nabla f(x), d_1 \rangle, \quad \|d_1\|_{(k_1)} \le 1, \quad x + d_1 \in X, \tag{3.9}$$

and

$$\alpha_{k_2}[x] = -\langle \nabla f(x), d_2 \rangle, \quad \|d_2\|_{(k_2)} \le 1, \quad x + d_2 \in X, \tag{3.10}$$

are such that

$$\sigma_1 \le \|d_1\|_2 \le \sigma_2 \quad \text{and} \quad \sigma_1 \le \|d_2\|_2 \le \sigma_2. \tag{3.11}$$

We note that the two upper bounds in these inequalities immediately result from AS.3 and (3.9)–(3.10). We therefore only consider the case where one or both lower bounds in (3.11) are violated. Assume, for instance, that $\|d_1\|_2 < \sigma_1$. This solution of the minimization problem associated with $\alpha_{k_1}[x]$ is therefore in the interior of all the possible balls of the form (3.5). The only binding constraint at this point must be $x + d \in X$, and this is still true if the ball defined by $\|\cdot\|_{(k_1)}$ is replaced by that defined by $\|\cdot\|_{(k_2)}$. But this implies that (3.8) cannot hold, which is impossible. The case where $\|d_2\|_2 < \sigma_1$ is entirely similar. The inequalities (3.11) are therefore valid, and we obtain that

$$\alpha_{\min} \le \alpha_{k_1}[x] \le \alpha_{\max} \quad \text{and} \quad \alpha_{\min} \le \alpha_{k_2}[x] \le \alpha_{\max}. \tag{3.12}$$

Combining these relations with (3.7) and (3.8), one deduces that

$$\alpha_{k_1}[x] < \alpha_{k_2}[x] \le \alpha_{\max} \le \frac{\sigma_2}{\sigma_1}\alpha_{\min} \le \frac{\sigma_2}{\sigma_1}\alpha_{k_1}[x] \tag{3.13}$$

and (3.3) is proved with $c_1 \overset{\text{def}}{=} \sigma_2/\sigma_1$. □

The fact that $\alpha_k[x]$ can now be used as a criticality measure results from the following lemma.

LEMMA 3.3. *Assume that* AS.1–AS.3 *hold. Then, x_* is critical if and only if*

$$\alpha_k[x_*] = 0. \tag{3.14}$$

Proof. Consider first the minimization problem of (3.2) where we choose $\|\cdot\|_{(k)} = \|\cdot\|_2$, and let us denote the analog of (3.2) by $\alpha_2[x]$.

The criticality conditions for this problem can be expressed as

$$0 \in 2\zeta d + \nabla f(x) + N(x + d), \tag{3.15}$$

$$x + d \in X, \tag{3.16}$$

$$\|d\|_2 \le 1, \tag{3.17}$$

and

$$\zeta\left(\|d\|_2^2 - 1\right) = 0. \tag{3.18}$$

Assume now that $\alpha_2[x_*] = 0$. Then the choice $d = 0$ is a solution of the minimization problem. The relation (3.1) then follows from (3.15).

Assume, on the other hand, that (3.1) holds. Then the conditions (3.15)–(3.18) are satisfied with $d = 0$ and $\zeta = 0$. It is then easy to verify that

$$\alpha_2[x_*] = 0 \tag{3.19}$$

follows.

As a consequence, x_* is critical if and only if (3.19) holds. But Theorem 3.2 and the fact that the ℓ_2-norm can be considered as one of the (k)-norms then yield the desired result. □

Lemmas 3.1 and 3.3 and Theorem 3.2 have the following important consequence.

COROLLARY 3.4. *Assume that* AS.1–AS.3 *hold and that the sequence* $\{x_k\}$ *is generated by Algorithm* 1. *Assume furthermore that there exists a subsequence of* $\{x_k\}$, $\{x_{k_i}\}$, *say, converging to* x_* *and that*

$$\lim_{i\to\infty} \alpha_{k_i}[x_{k_i}] = 0. \tag{3.20}$$

Then x_* *is critical.*

We note that, if formally equivalent, the criticality measures depending on k often differ from the practical point of view, when used in a stopping rule. If the problem's scaling is poor, a scaled measure is usually more appropriate. This scaling can be taken into account in the definition of the iteration-dependent norms.

On the other hand, if the only first-order information we can obtain is g_k (under the proviso (2.13)), then $\alpha_k[x]$ is unavailable, and one is naturally led to use

$$\alpha_k \stackrel{\text{def}}{=} \alpha_k(1) = \left| \min_{\substack{x_k+d\in X\\ \|d\|_{(k)}\le 1}} \langle g_k, d\rangle \right|, \tag{3.21}$$

which represents the amount of possible decrease for the *linearized model* in the intersection of the feasible domain with a ball of radius one. Clearly, $\alpha_k = \alpha_k[x_k]$ when $g_k = \nabla f(x_k)$, but this need not be the case in general. The value α_k was used in the "theoretical stopping rule" in Step 2 of Algorithm 1.

The replacement of $\alpha_k[x_k]$ by α_k has a price, however. It may well happen that an iterate x_k is a constrained critical point for the model m_k, although x_k is not critical for the true problem. In that case, Algorithm 1 will stop at the beginning of Step 2. The model m_k should therefore reflect the noncriticality of x_k. The discrepancy between α_k and $\alpha_k[x_k]$ cannot be arbitrarily large, however, as is shown by the following result.

LEMMA 3.5. *Let* $x_k \in X$ *be an iterate generated by Algorithm* 1. *Then*

$$|\alpha_k[x_k] - \alpha_k| \le \|e_k\|_{[k]}. \tag{3.22}$$

Proof. Define d_k^* and d_k as two vectors satisfying

$$\alpha_k[x_k] = -\langle \nabla f(x_k), d_k^*\rangle, \quad \|d_k^*\|_{(k)} \le 1, \quad x_k + d_k^* \in X, \tag{3.23}$$

and

$$\alpha_k = -\langle g_k, d_k\rangle, \quad \|d_k\|_{(k)} \le 1, \quad x_k + d_k \in X. \tag{3.24}$$

Assume first that $\alpha_k[x_k] \ge \alpha_k$. Then we can write that

$$\begin{aligned} 0 \le \alpha_k[x_k] - \alpha_k &= \langle g_k, d_k\rangle - \langle \nabla f(x_k), d_k^*\rangle \\ &= \langle g_k, d_k - d_k^*\rangle + \langle e_k, d_k^*\rangle \\ &\le \langle g_k, d_k - d_k^*\rangle + \|e_k\|_{[k]}, \end{aligned} \tag{3.25}$$

where we used the inequality (2.14). But the definitions of α_k, d_k, and d_k^* imply that

$$\langle g_k, d_k\rangle = -\alpha_k \leq \langle g_k, d_k^*\rangle, \tag{3.26}$$

and hence (3.22) follows from (3.25). On the other hand, if $\alpha_k[x_k] < \alpha_k$, then a similar argument can be used to prove (3.22), with (3.25) replaced by

$$0 < \alpha_k - \alpha_k[x_k] \leq \langle \nabla f(x_k), d_k^* - d_k\rangle + \|e_k\|_{[k]} \tag{3.27}$$

and (3.26) by

$$\langle \nabla f(x_k), d_k^*\rangle = -\alpha_k[x_k] \leq \langle \nabla f(x_k), d_k\rangle. \quad \Box \tag{3.28}$$

The bound (3.22) will be used at the end of our global convergence analysis.

3.2. The model decrease. The traditional next step in a trust region-oriented convergence analysis is to derive a lower bound on the reduction of the model value at an iteration where the current iterate x_k is noncritical. This lower bound usually involves the considered measure of criticality (α_k in our case), the trust region radius Δ_k, and the inverse of the curvature of the model m_k (see [9], [19], [21], [23], and [29] for examples of such bounds). To define this notion of curvature more precisely, we follow [29] and introduce, for an arbitrary continuously differentiable function q, the curvature at the point $x \in X$ along the step v, as defined by

$$\omega_k(q,x,v) \stackrel{\text{def}}{=} \frac{2}{\|v\|_{(k)}^2}\left[q(x+v) - q(x) - \langle \nabla q(x), v\rangle\right]. \tag{3.29}$$

If we assume that q is twice differentiable, the mean-value theorem (see, e.g., [16, p. 11]) implies that

$$\omega_k(q,x,v) = 2\int_0^1\int_0^1 \tau_2 \frac{\langle v, \nabla^2 q(x+\tau_1\tau_2 v)v\rangle}{\|v\|_{(k)}^2}\, d\tau_1\, d\tau_2. \tag{3.30}$$

It is also easy to verify that, if q is quadratic and $\|\cdot\|_{(k)} = \|\cdot\|_2$, then $\omega_k(q,x,v)$ is independent of x and of the norm of v, and reduces to the scaled Rayleigh quotient of $\nabla^2 q$ with respect to the direction v. We note that the Rayleigh quotient has already been used for similar purposes in the context of convergence analysis, namely, in [7], [28], and [29].

We then obtain the following simple result.

LEMMA 3.6. *If* AS.1–AS.3 *hold, then there exists a finite constant* $c_2 \geq 1$ *such that*

$$\omega_k(f, x_k, s) \leq c_2 \tag{3.31}$$

for all $k \geq 0$ *and all* s *such that* $x_k + s \in \mathcal{L}$.

Proof. The Lipschitz continuity of $\nabla f(x)$ implies that

$$|f(x_k+s) - f(x_k) - \langle \nabla f(x_k), s\rangle| \leq \tfrac{1}{2}L_f\|s\|_2^2, \tag{3.32}$$

where L_f is the Lipschitz constant of $\nabla f(x)$ in the norm $\|\cdot\|_2$. We may then deduce from (3.29) that

$$\omega_k(f, x_k, s) \leq L_f \frac{\|s\|_2^2}{\|s\|_{(k)}^2}, \tag{3.33}$$

which gives (3.31) with $c_2 = \max[1, \sigma_2^2 L_f]$, by using AS.3. □

We are now in position to state the main result of this section.

THEOREM 3.7. *Assume that* AS.1–AS.3 *hold. Consider any sequence* $\{x_k\}$ *produced by Algorithm* 1, *and select a* $k \geq 0$ *such that* x_k *is not critical in the sense that* $\alpha_k > 0$. *Then, if one defines*

$$\omega_k^C \overset{\text{def}}{=} \begin{cases} \omega_k(m_k, x_k, s_k^C) & \textit{if } s_k^C \textit{ satisfies } (2.35), \\ 0 & \textit{otherwise}, \end{cases} \tag{3.34}$$

one obtains that

$$\omega_k^C \geq 0. \tag{3.35}$$

Furthermore, there exists a constant $c_3 \in (0, 1]$ *such that*

$$m_k(x_k) - m_k(x_k + s_k) \geq c_3 \alpha_k \min\left[1, \Delta_k, \frac{\alpha_k}{1 + \omega_k^C}\right] \tag{3.36}$$

for all $k \geq 0$.

Proof. Let us first consider the case where $t_k \geq 1$. In this case, we obtain from (2.33), (2.32), the first statement of Lemma 2.2, and the definition (3.21) that

$$m_k(x_k) - m_k(x_k + s_k^C) \geq \mu_1 \mu_3 \alpha_k(t_k) \geq \mu_1 \mu_3 \alpha_k(1) = \mu_1 \mu_3 \alpha_k. \tag{3.37}$$

Assume now that $t_k < 1$. We first note that, because of (2.32), the second part of Lemma 2.2, (3.37), and (3.21), we have that

$$\frac{|\langle g_k, s_k^C \rangle|}{t_k} \geq \mu_3 \frac{\alpha_k(t_k)}{t_k} \geq \mu_3 \frac{\alpha_k(1)}{1} = \mu_3 \alpha_k. \tag{3.38}$$

Combining this inequality with (2.33), we obtain that

$$m_k(x_k) - m_k(x_k + s_k^C) \geq \mu_1 \frac{|\langle g_k, s_k^C \rangle|}{t_k} t_k \geq \mu_1 \mu_3 \alpha_k t_k. \tag{3.39}$$

Now, if condition (2.34) is satisfied, we can deduce, by using (3.39), that

$$m_k(x_k) - m_k(x_k + s_k^C) \geq \mu_1 \mu_3 \alpha_k \min[\nu_3 \Delta_k, \nu_4]. \tag{3.40}$$

On the other hand, if s_k^C satisfies (2.35), we observe that

$$\omega_k^C \geq \frac{2(1-\mu_2)}{\|s_k^C\|_{(k)}} \frac{|\langle g_k, s_k^C \rangle|}{\|s_k^C\|_{(k)}} \geq \frac{2(1-\mu_2)}{t_k} \frac{|\langle g_k, s_k^C \rangle|}{t_k}, \tag{3.41}$$

where we used the definition of ω_k^C and (2.35). Hence (3.35) is proved and, using (3.38), we have that

$$t_k \geq 2\mu_3(1-\mu_2) \frac{\alpha_k}{\omega_k^C} \geq 2\mu_3(1-\mu_2) \frac{\alpha_k}{1 + \omega_k^C}. \tag{3.42}$$

Substituting this bound into (3.39) then yields that

$$m_k(x_k) - m_k(x_k + s_k^C) \geq 2\mu_1 \mu_3^2 (1-\mu_2) \frac{\alpha_k^2}{1 + \omega_k^C}. \tag{3.43}$$

The inequality (3.36) now results from (3.37), (3.40), (3.43), (2.38), and $\nu_4 \leq 1$, with

$$c_3 = \mu_1\mu_3\mu_4 \min[\nu_3, \nu_4, 2\mu_3(1-\mu_2)] \leq 1. \qquad \square \tag{3.44}$$

We end this subsection by stating an easy corollary of Theorem 3.7, giving a lower bound on the decrease in the objective that is obtained on successful iterations.

COROLLARY 3.8. *Under the assumptions of Theorem* 3.7, *one obtains that*

$$f(x_k) - f(x_{k+1}) \geq \eta_1 c_3 \alpha_k \min\left[1, \Delta_k, \frac{\alpha_k}{1+\omega_k^C}\right] \tag{3.45}$$

for $k \in \mathcal{S}$.

Proof. The inequality (3.45) immediately results from (3.36), (2.39), (2.40), and (2.41). □

3.3. Convergence to critical points. This section is devoted to the proof of global convergence of the iterates generated by Algorithm 1 to critical points.

For developing our convergence theory, we will need to introduce additional assumptions on the curvature of the models m_k. These assumptions, and the rest of our convergence analysis, will be phrased in terms of the quantity

$$\beta_k = 1 + \max_{i=0,\ldots,k} \left[\max[\omega_i^C, |\omega_i(m_i, x_i, s_i)|]\right]. \tag{3.46}$$

We note that β_k only measures curvature of the model along the s_k^C and s_k vectors. We also observe that the sequence $\{\beta_k\}$ is nondecreasing by definition.

We first recall two useful preliminary results in the spirit of [29].

LEMMA 3.9. *Assume that* AS.1–AS.3 *hold and consider a sequence* $\{x_k\}$ *of iterates generated by Algorithm* 1. *Then there exists a positive constant* $c_4 \geq 1$ *such that, for all* $k \geq 0$,

$$|f(x_k+s_k) - m_k(x_k+s_k)| \leq c_4\beta_k\Delta_k^2. \tag{3.47}$$

Proof. We observe that

$$\begin{aligned} |f(x_k+s_k) - m_k(x_k+s_k)| &\leq |\langle \nabla f(x_k) - g_k, s_k\rangle| \\ &\quad + \tfrac{1}{2}\|s_k\|_{(k)}^2 |\omega_k(f, x_k, s_k) - \omega_k(m_k, x_k, s_k)| \\ &\leq \|e_k\|_{[k]}\, \|s_k\|_{(k)} \\ &\quad + \tfrac{1}{2}\|s_k\|_{(k)}^2 [|\omega_k(f, x_k, s_k)| + |\omega_k(m_k, x_k, s_k)|], \end{aligned} \tag{3.48}$$

where we used the definition (3.29), (2.12), and the inequality (2.14). But $\|s_k\|_{(k)} \leq \nu_1\Delta_k$, and hence we obtain from (3.48), (2.13), (3.46), and Lemma 3.6 that

$$|f(x_k+s_k) - m_k(x_k+s_k)| \leq \kappa_1\nu_1\Delta_k^2 + \tfrac{1}{2}\nu_1^2(c_2+\beta_k)\Delta_k^2, \tag{3.49}$$

which then yields (3.47) with

$$c_4 = 2\left(c_2 + \frac{\kappa_1}{\nu_1}\right)\max\left[1, \frac{1}{2}\nu_1^2\right]. \qquad \square \tag{3.50}$$

LEMMA 3.10. *Assume that* AS.1–AS.3 *hold and consider a sequence* $\{x_k\}$ *of iterates generated by Algorithm* 1. *Assume furthermore that there exists a constant* $\epsilon \in (0,1)$ *such that*

$$\alpha_k \geq \epsilon \tag{3.51}$$

for all k. Then there exists a positive constant c_5 such that

$$\Delta_k \geq \frac{c_5}{\beta_k} \tag{3.52}$$

for all k.

Proof. Assume, without loss of generality, that

$$\epsilon < \frac{c_4\beta_0\Delta_0}{\gamma_1 c_3(1-\eta_2)}, \tag{3.53}$$

where γ_1 and η_2 are defined in the algorithm (see (2.29) and (2.28)). In order to derive a contradiction, assume also that there exists a k such that

$$\beta_k\Delta_k \leq \frac{\gamma_1 c_3(1-\eta_2)}{c_4}\epsilon \tag{3.54}$$

and define r as the first iteration number such that (3.54) holds. (Note that $r \geq 1$ because of (3.53).) The mechanism of Algorithm 1 then ensures that

$$\beta_{r-1}\Delta_{r-1} \leq \beta_r\frac{\Delta_r}{\gamma_1} \leq \frac{c_3(1-\eta_2)}{c_4}\epsilon \leq \epsilon, \tag{3.55}$$

where we used the relations $\beta_{r-1} \leq \beta_r$, (2.45), (3.54) with $k = r$, $c_3 \leq 1$, and $c_4 \geq 1$. Combining the inequalities (3.51), (3.36), $\epsilon < 1$, $\beta_{r-1} \geq 1$, and (3.55), we now obtain that

$$m_{r-1}(x_{r-1}) - m_{r-1}(x_{r-1}+s_{r-1}) \geq c_3\epsilon\min\left[1, \Delta_{r-1}, \frac{\epsilon}{\beta_{r-1}}\right] = c_3\epsilon\Delta_{r-1}. \tag{3.56}$$

The relations (2.39), (3.47), (3.56), and the middle part of (3.55) together then imply that

(3.57)

$$|\rho_{r-1}-1| = \frac{|f(x_{r-1}+s_{r-1}) - m_{r-1}(x_{r-1}+s_{r-1})|}{|m_{r-1}(x_{r-1}) - m_{r-1}(x_{r-1}+s_{r-1})|} \leq \frac{c_4\beta_{r-1}\Delta_{r-1}}{c_3\epsilon} \leq 1-\eta_2.$$

Hence $\rho_{r-1} \geq \eta_2$, and thus $\Delta_r \geq \Delta_{r-1}$. However, we may deduce from this last inequality that

$$\beta_{r-1}\Delta_{r-1} \leq \beta_r\Delta_r \leq \frac{\gamma_1 c_3(1-\eta_2)}{c_4}\epsilon, \tag{3.58}$$

which contradicts the assumption that r is the first index with (3.54) satisfied. The inequality (3.54) therefore never holds, and we obtain that, for all k,

$$\beta_k\Delta_k > \frac{\gamma_1 c_3(1-\eta_2)}{c_4}\epsilon. \tag{3.59}$$

The inequality (3.52) then follows from (3.59) by setting

$$c_5 = \frac{\gamma_1 c_3(1-\eta_2)\epsilon}{c_4}. \qquad \square \tag{3.60}$$

We now formulate our first assumption on the model's curvatures.

AS.4. The series

$$\sum_{k=0}^{\infty} \frac{1}{\beta_k} \tag{3.61}$$

is divergent.

As shown in [29], this condition is necessary for guaranteeing convergence to a stationary point. It is clearly satisfied in the common case where quadratic models of the form (2.46) are used, whose Hessian matrices H_k are uniformly bounded. This last assumption obviously holds when $f(x)$ is twice continuously differentiable over the compact set $\mathcal{L}$ and $H_k = \nabla^2 f(x_k)$.

Before proving one of the major results of this section, we recall the following technical lemma, due to Powell [24] (proofs can also be found in [9] or [32]).

LEMMA 3.11. *Let $\{\Delta_k\}$ and $\{\beta_k\}$ be two sequences of positive numbers such that $\beta_k \Delta_k \geq c_5$ for all k, where c_5 is a positive constant. Let ϵ be a positive constant, $\mathcal{S}$ be a subset of $\{1, 2, \ldots\}$, and assume that, for some constants $\gamma_2 < 1$ and $\gamma_3 > 1$,*

$$\Delta_{k+1} \leq \gamma_3 \Delta_k \quad \textit{for } k \in \mathcal{S}, \tag{3.62}$$

$$\Delta_{k+1} \leq \gamma_2 \Delta_k \quad \textit{for } k \notin \mathcal{S}, \tag{3.63}$$

$$\beta_{k+1} \geq \beta_k \quad \textit{for all } k, \tag{3.64}$$

and

$$\sum_{k \in \mathcal{S}} \min \left[\Delta_k, \frac{\epsilon}{\beta_k} \right] < \infty. \tag{3.65}$$

Then

$$\sum_{k=1}^{\infty} \frac{1}{\beta_k} < \infty. \tag{3.66}$$

Using this lemma, we now show the following important result.

THEOREM 3.12. *Assume that* AS.1–AS.4 *hold. Then, if $\{x_k\}$ is a sequence of iterates generated by Algorithm* 1, *one has that*

$$\liminf_{k \to \infty} \alpha_k = 0. \tag{3.67}$$

Proof. Assume, for the purpose of obtaining a contradiction, that there exists an $\epsilon \in (0,1)$ such that (3.51) holds for all $k \geq 0$. Corollary 3.8 and the fact that the objective function is bounded below on $\mathcal{L}$ imply that

$$\eta_1 c_3 \epsilon \sum_{k \in \mathcal{S}} \min \left[1, \Delta_k, \frac{\epsilon}{\beta_k} \right] \leq \sum_{k \in \mathcal{S}} [f(x_k) - f(x_{k+1})] < \infty. \tag{3.68}$$

Thus, because of Lemma 3.10 and the inequalities $\epsilon < 1$ and $\beta_k \geq 1$, the sequences Δ_k and β_k then verify all the assumptions of Lemma 3.11, which then guarantees that

$$\sum_{k=0}^{\infty} \frac{1}{\beta_k} < \infty. \tag{3.69}$$

This last relation clearly contradicts AS.4, and hence our initial assumption must be false, yielding (3.67). □

This theorem has the following interesting consequences.

COROLLARY 3.13. *Assume that* AS.1–AS.4 *hold. Assume furthermore that* $\{x_k\}$ *is a sequence of iterates generated by Algorithm* 1 *that converges to* x_*, *and that*

$$\lim_{k\to\infty} \|e_k\|_{[k]} = 0. \tag{3.70}$$

Then x_* *is critical.*

Proof. This result follows directly from (3.70), Lemma 3.5, Theorem 3.12, and Corollary 3.4. □

COROLLARY 3.14. *Assume that* AS.1–AS.4 *hold. If* $\{x_k\}$ *is a sequence of iterates generated by Algorithm* 1 *and if* $\mathcal{S}$ *is finite, then the iterates* x_k *are all equal to some* x_* *for* k *large enough, and* x_* *is critical.*

Proof. If $\mathcal{S}$ is finite, it results from (2.44) that x_k is unchanged for k large enough, and therefore that $x_k = x_* = x_{j+1}$ for k sufficiently large, where j is the largest index in $\mathcal{S}$. The relations (2.45) and (2.29) also imply that the sequence $\{\Delta_k\}$ converges to zero. Hence (2.13) ensures that (3.70) holds. We then apply Corollary 3.13 to deduce the criticality of x_*. □

If we now assume that $\mathcal{S}$ is infinite, we wish to replace the lim inf in (3.67) by a true limit, taken on all successful iterations, but this requires a slight strengthening of our assumption on the model curvature.

AS.5. We assume that

$$\lim_{k\to\infty} \beta_k[f(x_k) - f(x_{k+1})] = 0. \tag{3.71}$$

As discussed in [9], this assumption is not very severe, as we always have that (3.71) holds with the limit replaced by the limit inferior. Also, AS.5 is obviously satisfied when using a model with bounded curvature, as is assumed in [20], for example.

THEOREM 3.15. *Assume that* AS.1–AS.5 *hold. Then, if* $\{x_k\}$ *is a sequence of iterates generated by Algorithm* 1 *and if the set* $\mathcal{S}$ *is infinite, one has that*

$$\lim_{\substack{k\to\infty\\ k\in\mathcal{S}}} \alpha_k = 0. \tag{3.72}$$

Proof. We proceed again by contradiction and assume that there exists an $\epsilon_1 \in (0,1)$ and a subsequence $\{m_i\}$ of successful iterates such that, for all m_i in this subsequence,

$$\alpha_{m_i} \geq \epsilon_1. \tag{3.73}$$

If we define

$$c_6 \stackrel{\text{def}}{=} \max\left[1 - \frac{1}{c_1}, c_1 - 1\right], \tag{3.74}$$

where c_1 is given by Theorem 3.2, and if we choose

$$\epsilon_2 \in \left(0, \frac{\epsilon_1}{2(c_6+1)}\right), \tag{3.75}$$

Theorem 3.12 then ensures the existence of another subsequence $\{\ell_i\}$ such that

$$\alpha_k \geq \epsilon_2 \quad \text{for } m_i \leq k < \ell_i \quad \text{and} \quad \alpha_{\ell_i} < \epsilon_2. \tag{3.76}$$

We now restrict our attention to the subsequence of successful iterations whose indices are in the set

$$\mathcal{K} \overset{\text{def}}{=} \{k \in \mathcal{S} \mid m_i \leq k < \ell_i\}, \tag{3.77}$$

where m_i and ℓ_i belong, respectively, to the two subsequences defined above. Applying Corollary 3.8 for $k \in \mathcal{K}$, we obtain that

$$f(x_k) - f(x_{k+1}) \geq \eta_1 c_3 \epsilon_2 \min\left[\Delta_k, \frac{\epsilon_2}{\beta_k}\right], \tag{3.78}$$

where we used the inequalities $\epsilon_2 < 1$ and $\beta_k \geq 1$. But AS.5 then implies that

$$\lim_{\substack{k\to\infty \\ k\in\mathcal{K}}} \beta_k \Delta_k = 0, \tag{3.79}$$

and hence, using (3.78), that

$$f(x_k) - f(x_{k+1}) \geq \eta_1 c_3 \epsilon_2 \Delta_k \tag{3.80}$$

for $k \in \mathcal{K}$ sufficiently large. As a consequence, we obtain, for i sufficiently large, that

$$\begin{aligned} \|x_{m_i} - x_{\ell_i}\|_2 &\leq \textstyle\sum_{k=m_i}^{\ell_i-1} \|x_{k+1} - x_k\|_2 \\ &\leq \sigma_2 \nu_1 \textstyle\sum_{k=m_i}^{\ell_i-1} {}^{(\mathcal{K})} \Delta_k \\ &\leq c_7 \textstyle\sum_{k=m_i}^{\ell_i-1} {}^{(\mathcal{K})} [f(x_k) - f(x_{k+1})] \\ &\leq c_7 [f(x_{m_i}) - f(x_{\ell_i})], \end{aligned} \tag{3.81}$$

where the sums with superscript $(\mathcal{K})$ are restricted to the indices in $\mathcal{K}$, and where

$$c_7 \overset{\text{def}}{=} \frac{\sigma_2 \nu_1}{\eta_1 c_3 \epsilon_2}. \tag{3.82}$$

Because of Lemma 3.1 and because the last right-hand side of (3.81) tends to zero as i tends to infinity, we deduce that

$$|\alpha_{m_i}[x_{m_i}] - \alpha_{m_i}[x_{\ell_i}]| \leq \frac{\epsilon_1}{2(c_6 + 3)} \tag{3.83}$$

for i sufficiently large. We note now that (3.79), $\beta_k \geq 1$, and (2.13) imply that g_{m_i} is arbitrarily close to $\nabla f(x_{m_i})$, and hence Lemma 3.5 gives that

$$|\alpha_{m_i} - \alpha_{m_i}[x_{m_i}]| \leq \frac{\epsilon_1}{2(c_6 + 3)} \tag{3.84}$$

for i large enough. We observe also that, because of (2.13) and (2.42),

$$\|e_{\ell_i}\|_{[\ell_i]} \leq \kappa_1 \Delta_{\ell_i} \leq \kappa_1 \gamma_3 \Delta_{k_i}, \tag{3.85}$$

where k_i is the largest integer in $\mathcal{K}$ that is smaller than ℓ_i. As before, we now deduce from (3.79), $\beta_k \geq 1$, Lemma 3.5, and (3.85) that

$$|\alpha_{\ell_i} - \alpha_{\ell_i}[x_{\ell_i}]| \leq \frac{\epsilon_1}{2(c_6 + 3)} \tag{3.86}$$

for large i. Hence, using Theorem 3.2, we obtain that

$$|\alpha_{m_i}[x_{\ell_i}] - \alpha_{\ell_i}[x_{\ell_i}]| \leq c_6 \alpha_{\ell_i}[x_{\ell_i}] \leq c_6 \left[\alpha_{\ell_i} + \frac{\epsilon_1}{2(c_6+3)}\right] \tag{3.87}$$

for i sufficiently large. Using the triangular inequality together with (3.84), (3.83), (3.87), and (3.86), we obtain that, for large enough i,

$$\alpha_{m_i} - \alpha_{\ell_i} \leq |\alpha_{m_i} - \alpha_{\ell_i}| \leq c_6 \alpha_{\ell_i} + \tfrac{1}{2}\epsilon_1. \tag{3.88}$$

We then deduce from (3.76) and (3.75), that, for large enough i,

$$\alpha_{m_i} \leq \alpha_{\ell_i}(c_6 + 1) + \tfrac{1}{2}\epsilon_1 < \epsilon_1, \tag{3.89}$$

which contradicts (3.73) and proves the desired result. □

As above, we can obtain conclusions about convergent subsequences where the first-order information is asymptotically correct. If $\mathcal{S}$ is finite, the convergence of the iterates to a critical point results from Corollary 3.14. Hence, we now restrict our attention to the case where $\mathcal{S}$ is infinite.

COROLLARY 3.16. *Assume that* AS.1–AS.5 *hold. Assume furthermore that $\mathcal{S}$ is infinite, that $\{x_{k_i}\}$ is a convergent subsequence of the successful iterates generated by Algorithm* 1, *and that*

$$\lim_{i\to\infty} \|e_{k_i}\|_{[k_i]} = 0. \tag{3.90}$$

Then x_*, the limit point of $\{x_{k_i}\}$, is critical.

Proof. The proof of this result is entirely similar to that of Corollary 3.13 except that we have to consider only the successful iterates. □

Finally, we are interested in what can be said on the criticality of limit points of $\{x_k\}$ if we do not assume (3.70).

COROLLARY 3.17. *Assume that* AS.1–AS.5 *hold, that $\{x_{k_i}\}$ is a subsequence of successful iterates generated by Algorithm* 1, *and that $\{x_{k_i}\}$ converges to x_*. Then*

$$\limsup_{i\to\infty} \alpha_{k_i}[x_*] \leq \limsup_{i\to\infty} \|e_{k_i}\|_{[k_i]}. \tag{3.91}$$

Proof. If $\mathcal{S}$ is finite, then the result immediately follows from Corollary 3.14 and Lemma 3.3. Assume, therefore, that $\mathcal{S}$ is infinite. Because of Lemma 3.1, Lemma 3.5, and Theorem 3.15, we have that

$$\begin{aligned} \limsup_{i\to\infty} \alpha_{k_i}[x_*] &= \limsup_{i\to\infty} \alpha_{k_i}[x_{k_i}] \\ &\leq \limsup_{i\to\infty} |\alpha_{k_i}[x_{k_i}] - \alpha_{k_i}| \\ &\leq \limsup_{i\to\infty} \|e_{k_i}\|_{[k_i]}. \qquad \square \end{aligned} \tag{3.92}$$

Keeping in mind that the dependence of $\|\cdot\|_{[k_i]}$ on k_i, and hence on i, is irrelevant because of Theorem 3.2, Corollary 3.17 thus guarantees that all limit points are as critical as the scaled accuracy of g_k as an approximation to $\nabla f(x_k)$ warrants.

4. A model algorithm for computing a generalized Cauchy point. A major difficulty in adapting the framework given by Algorithm 1 to a more practical setting is clearly the definition of a practical procedure to compute a GCP satisfying all the conditions of Step 2.

As indicated already, such procedures have been designed and implemented in the case where the projected gradient path defined by the classical ℓ_2-norm is explicitly available (see [1] and [29], for example).

We now consider the more general case presented in §§2 and 3, and we wish to find, at a given iteration, a GCP satisfying (2.30)–(2.35). The difficulty is then to produce a point that is not too far away from the *unavailable* projected gradient path. This cannot be done without considering the particular geometry of this path, which may closely follow the boundary of the feasible set. As a consequence, linear interpolation between two points on the projected gradient path is often unsuitable, and a specialized procedure is presented in this section.

For the sake of clarity, in this section we will drop the subscript k, corresponding to the iteration number.

4.1. The RS Algorithm. We first define the following *restriction operator* associated with the feasible set X and a *centre* $x \in X$. This operator is defined as

$$R_x[y] \overset{\text{def}}{=} \arg \min_{z \in [x,y] \cap X} \|z - y\|_2 \tag{4.1}$$

for any $y \in \mathbf{R}^n$, where $[x, y]$ is the segment between x and y. The definition of $R_x[y]$ uses the ℓ_2-norm, but any other norm can be used because the associated minimization problem is unidimensional. The action of the restriction operator (4.1) is illustrated in Fig. 3. It should be noted that computing $R_x[y]$ for a given y is often a very simple task.

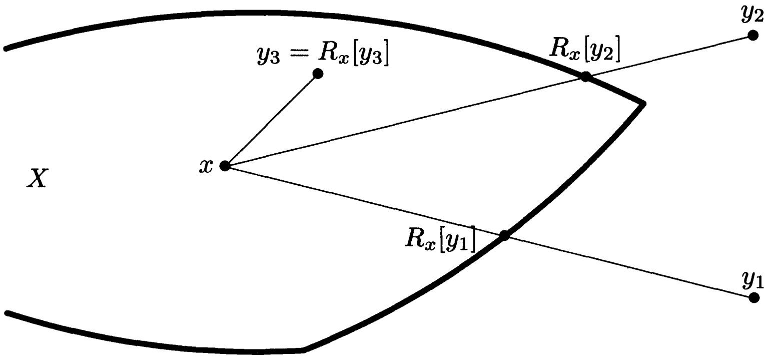


FIG. 3. *The restriction operator with centre x.*

The GCP Algorithm relies on a simple bisection linesearch algorithm on the restriction of a piecewise linear path with respect to a given center, called the RS Algorithm (Restricted Search Algorithm). Because of the definition of the restriction operator, this last algorithm closely follows the boundary of the feasible domain, as desired. It finds a point $x_* = x + z$ in $R_x[x^l, x^p, x^u]$, the restriction of a nonempty piecewise linear path consisting of the segment $[x^l, x^p]$ followed by $[x^p, x^u]$, where x^l, x^p, and x^u are defined below. The restriction is computed with respect to the centre x, and the resulting vector z is such that (2.33) and (2.35) hold with $s_k^C = z$. The RS Algorithm can be applied under the conditions that (2.35) is violated at $R_x[x^l]$ and that (2.33) is violated at $R_x[x^u]$. It therefore depends on the three points x^l, x^p, and x^u defining the piecewise linear path, the centre x, and on the current model m (and hence on its gradient g). It also depends on an arbitrary bijective parametrization of the path $[x^l, x^p, x^u]$. For example, one can choose the parameter to be the length of the arc along the path measured in the ℓ_2-norm. More formally, if

$$\delta_p = \|x^p - x^l\|_2 \quad \text{and} \quad \delta_u = \delta_p + \|x^u - x^p\|_2, \tag{4.2}$$

we can define

$$x(\delta) \overset{\text{def}}{=} \begin{cases} \frac{\delta}{\delta_p} x^p + (1 - \frac{\delta}{\delta_p}) x^l & \text{if } \delta \le \delta_p, \\ \frac{\delta - \delta_p}{\delta_u - \delta_p} x^u + (1 - \frac{\delta - \delta_p}{\delta_u - \delta_p}) x^p & \text{if } \delta \ge \delta_p \end{cases} \tag{4.3}$$

for any $\delta \in [0, \delta_u]$. The inner iterations of Algorithm RS will be denoted by the index j.

RS ALGORITHM.

Step 0. Initialization. Set $l_0 = 0$, $u_0 = \delta_u$, and $j = 0$. Then define $\delta_0 = \frac{1}{2}(l_0 + u_0)$.

Step 1. Check the stopping conditions. Compute $x_j = R_x[x(\delta_j)]$, using (4.1) and (4.3). If

$$m(x_j) > m(x) + \mu_1 \langle g, x_j - x \rangle, \tag{4.4}$$

then set

$$l_{j+1} = l_j \quad \text{and} \quad u_{j+1} = \delta_j, \tag{4.5}$$

and go to Step 2. Else, if

$$m(x_j) < m(x) + \mu_2 \langle g, x_j - x \rangle, \tag{4.6}$$

then set

$$l_{j+1} = \delta_j \quad \text{and} \quad u_{j+1} = u_j, \tag{4.7}$$

and go to Step 2; else (that is, if both (4.4) and (4.6) fail), set $x_* = x_j$ and STOP.

Step 2. Choose the next parameter value by bisection. Increment j by one, set

$$\delta_j = \tfrac{1}{2}(l_j + u_j), \tag{4.8}$$

and go to Step 1.

The fact that a vector x_* has been produced by the application of the RS Algorithm on the path $[x^l, x^p, x^u]$ with respect to the centre x and the model m will be denoted by

$$x_* = \text{RS}(x, m, x^l, x^p, x^u). \tag{4.9}$$

We have the following simple result.

LEMMA 4.1. *Assume that the* RS *Algorithm is applied on a piecewise linear path* $[x^l, x^p, x^u]$ *satisfying the conditions stated in the paragraph preceding its description, with centre* x *and model* m. *Then this algorithm terminates with a suitable vector* $x_* = x + z$ *at which* (2.33) *and* (2.35) *hold in a finite number of iterations.*

Proof. We first note that (2.35) is violated at $R_x[x^l]$ and that (2.33) is violated at $R_x[x^u]$. As a consequence, the validity of the result directly follows from the inequality $\mu_1 < \mu_2$, the continuity of the model m on the restriction of the path $[x^l, x^p, x^u]$, and from the fact that the length of the interval $[l_j, u_j]$ tends geometrically to zero, while its associated arc on the restricted path always contains a fixed connected set of acceptable points. □

4.2. The GCP Algorithm. We now describe the GCP Algorithm itself. It depends on the current iterate $x \in X$, on the current model m and its gradient g, on the current norm $\|\cdot\|$, and also on the current trust region radius, $\Delta > 0$. Its inner iterations will be identified by the index i. (Also recall that all subscripts k have been dropped, yielding, for instance, $\alpha(t)$ instead of $\alpha_k(t)$ and α instead of α_k.)

GCP ALGORITHM.

Step 0. Initialization. Set $i = 0$, $l_0 = 0$, $z_0^l = 0$, and $u_0 = \nu_2\Delta$. Also choose z_0^u as an arbitrary vector such that $\|z_0^u\| > \nu_2\Delta$ and an initial parameter $t_0 \in (0, \nu_2\Delta]$.

Step 1. Compute a candidate step. Compute a vector z_i such that

$$\|z_i\| \le t_i, \tag{4.10}$$

$$x + z_i \in X, \tag{4.11}$$

and

$$\langle g, z_i \rangle \le -\mu_3\alpha(t_i). \tag{4.12}$$

Step 2. Check the stopping rules on the model and step. If

$$m(x + z_i) > m(x) + \mu_1 \langle g, z_i \rangle, \tag{4.13}$$

then set

$$u_{i+1} = t_i, \qquad z_{i+1}^u = z_i \tag{4.14}$$

and

$$l_{i+1} = l_i, \qquad z_{i+1}^l = z_i^l, \tag{4.15}$$

and go to Step 3. Else, if

$$m(x + z_i) < m(x) + \mu_2 \langle g, z_i \rangle \tag{4.16}$$

and

$$t_i < \min[\nu_3\Delta, \nu_4], \tag{4.17}$$

then set

$$u_{i+1} = u_i, \qquad z_{i+1}^u = z_i^u \tag{4.18}$$

and

$$l_{i+1} = t_i, \qquad z^l_{i+1} = z_i, \tag{4.19}$$

and go to Step 3. Else (that is, if (4.13) and either (4.16) or (4.17) fail), then set

$$x^C = x + z_i \tag{4.20}$$

and STOP.

Step 3. Define a new trial step by bisection. We distinguish two mutually exclusive cases.

Case 1. $z^l_{i+1} = z^l_0$ or $z^u_{i+1} = z^u_0$. Set

$$t_{i+1} = \tfrac{1}{2}(l_{i+1} + u_{i+1}), \tag{4.21}$$

increment i by one and go to Step 1.

Case 2. $z^l_{i+1} \neq z^l_0$ and $z^u_{i+1} \neq z^u_0$. Define

$$z^p_{i+1} = \max\left[1, \frac{\|z^u_{i+1}\|}{\|z^l_{i+1}\|}\right] z^l_{i+1}, \tag{4.22}$$

set

$$x^C = \mathrm{RS}(x, m, x^l_{i+1}, x^p_{i+1}, x^u_{i+1}) \tag{4.23}$$

where

$$x^l_{i+1} = x + z^l_{i+1}, \quad x^p_{i+1} = x + z^p_{i+1}, \quad x^u_{i+1} = x + z^u_{i+1}, \tag{4.24}$$

and STOP.

The actual value of z^u_0 is irrelevant in practice: this quantity is merely used to detect if z^u_{i+1} has been updated in (4.14) at least once.

Figure 4 shows the situation at a given iteration of the GCP Algorithm in the case where $\|\cdot\|_{(k)} = \|\cdot\|_\infty$. In particular, the use of the point x^p as defined in Step 3 (Case 2) is illustrated. The symbols x^r, x^f, t^l, t^u, x_{t^l}, C_{t^l}, and C_{t^u} are not yet defined, but will be introduced in the proof of Theorem 4.5 below.

We note that linear interpolation between $x^l_{i+1} = R_x[x^l_{i+1}]$ and $x^u_{i+1} = R_x[x^u_{i+1}]$ cannot generally be used in Step 3 (Case 2), because the geometry of the boundary of the feasible domain may imply that the (unknown) projected gradient path considerably departs from the segment $[x^l_{i+1}, x^u_{i+1}]$. This is the reason why a call is made to the RS Algorithm, which closely follows this boundary.

We emphasize that this GCP Algorithm is only a model, intended to show feasibility of our approach, but is not optimized from the point of view of efficiency. Many additional considerations are possible and indeed necessary before implementing the algorithm, including

- the details of the all-important solver used to determine z_i in Step 1,
- a suitable choice of t_0,
- more efficient techniques for simple models (e.g., linear or quadratic), and also for specific choices of the norm $\|\cdot\|$.

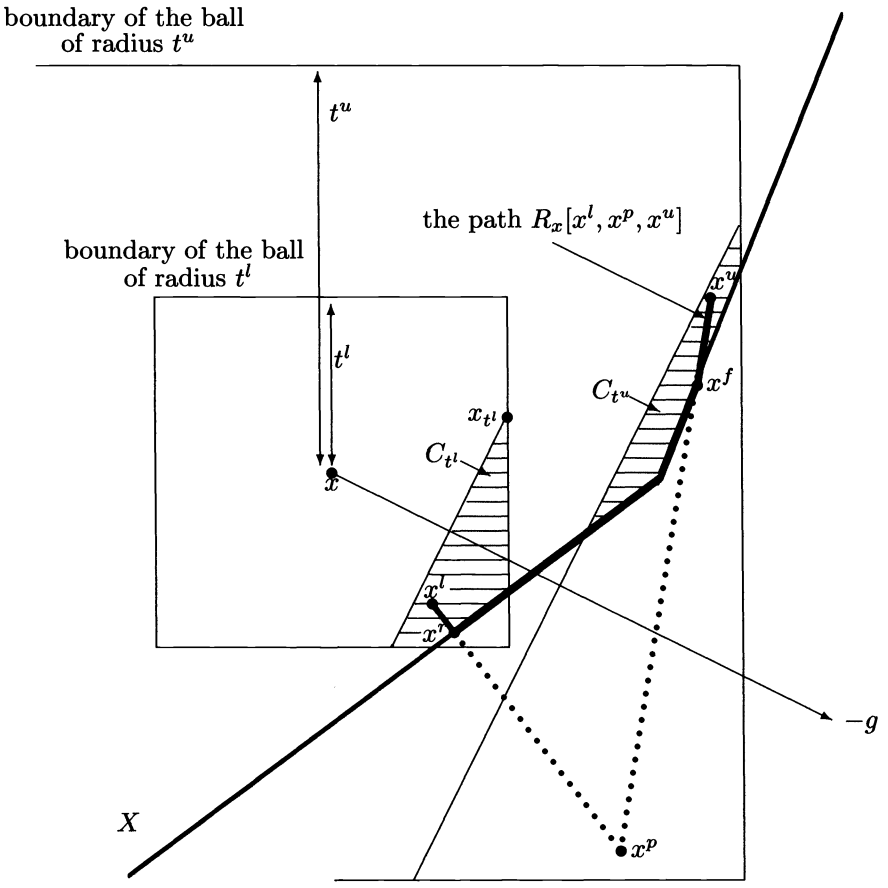


FIG. 4. *A "restricted path" with the ℓ_∞-norm.*

The solver used in Step 1 obviously depends on X and the norm $\|\cdot\|$. For example, Step 1 reduces to a linear programming problem if X is polyhedral and a polyhedral norm is used; the classical projected gradient may also be obtained when the ℓ_2-norm is used and $\mu_3 = 1$.

If we denote by

$$x^C = \text{GCP}(x, m, \|\cdot\|, \Delta) \tag{4.25}$$

the fact that the vector x^C has been obtained by the GCP Algorithm for the point x, the model m, the norm $\|\cdot\|$, and the radius Δ, we then replace Step 2 of Algorithm 1 by the simple call

$$x_k^C = \text{GCP}(x_k, m_k, \|\cdot\|_{(k)}, \Delta_k). \tag{4.26}$$

4.3. Properties of the GCP Algorithm. We now wish to show that the GCP Algorithm converges to a point satisfying (2.30)–(2.35) and terminates in a finite number of iterations.

The first result shows that, if a step z satisfies (2.32), then all prolongations of this step, that is, all vectors of the form τz with $\tau \geq 1$, also satisfy the same condition.

LEMMA 4.2. *Assume that there exists a $t \geq \|z\|$ such that*

$$\langle g, z \rangle \leq -\mu_3 \alpha(t) \tag{4.27}$$

for some $z \neq 0$. Then

$$\langle g, \tau z \rangle \leq -\mu_3 \alpha(\tau t) \tag{4.28}$$

for $\tau \geq 1$.

Proof. Using successively (4.27), the inequality $\tau \geq 1$, and the second part of Lemma 2.2, we obtain that

$$\langle g, \tau z \rangle \leq -\mu_3 \tau t \frac{\alpha(t)}{t} \leq -\mu_3 \tau t \frac{\alpha(\tau t)}{\tau t}, \tag{4.29}$$

yielding the desired bound. □

We are now in the position to prove that the GCP Algorithm is correctly stated, finite, and coherent with the theoretical framework presented in §§2 and 3.

LEMMA 4.3. *The* GCP *Algorithm has well-defined iterates.*

Proof. We have to verify that all the requested conditions for applying the RS Algorithm are fulfilled when a call to this algorithm is made. We first note that the RS Algorithm can only produce a feasible point because of the definition of the restriction operator. We also note that the mechanism of the GCP Algorithm ensures that the piecewise path to be restricted is nonempty, that (2.33) is always violated at $R_x[x_{i+1}^u] = x_{i+1}^u$, and, similarly, that (2.35) is always violated at $R_x[x_{i+1}^l] = x_{i+1}^l$. The RS Algorithm is therefore applied in the appropriate context. □

We now prove the desirable finiteness of the GCP Algorithm at noncritical points.

THEOREM 4.4. *Assume that $\alpha > 0$. Then the* GCP *Algorithm terminates with a suitable x^C in a finite number of iterations.*

Proof. Assume that an infinite number of iterations are performed. We first consider the case where

$$z_i^l = z_0^l \quad \text{for all } i \geq 0. \tag{4.30}$$

In this case, the mechanism of the GCP Algorithm implies that

$$t_i \leq (\tfrac{1}{2})^i \nu_2 \Delta. \tag{4.31}$$

Hence we obtain that

$$\|z_i\| \leq t_i \leq \min \left[1, \frac{2(1-\mu_1)\mu_3 \alpha}{L_m} \right] \tag{4.32}$$

for all $i \geq i_1$, say, where L_m is the Lipschitz constant of the gradient of m with respect to the norm $\|\cdot\|$. For all $i \geq 0$, we have that

$$m(x + z_i) - m(x) - \mu_1 \langle g, z_i \rangle \leq (1 - \mu_1) \langle g, z_i \rangle + \tfrac{1}{2} L_m \|z_i\|^2, \tag{4.33}$$

where we have used Taylor's expansion of m around x and the definition of L_m. But the second part of Lemma 2.2 implies that

$$\frac{\alpha(t_i)}{t_i} \geq \frac{\alpha(1)}{1} = \alpha \tag{4.34}$$

for all $i \geq i_1$, and hence that

$$\alpha(t_i) \geq \alpha \|z_i\| \tag{4.35}$$

for $i \geq i_1$, because of the inequality $t_i \geq \|z_i\|$. Condition (4.12) then gives, for such i, that

$$\langle g, z_i \rangle \leq -\mu_3 \alpha(t_i) \leq -\mu_3 \alpha \|z_i\|. \tag{4.36}$$

Introducing this inequality in (4.33), we obtain that

$$m(x + z_i) - m(x) - \mu_1 \langle g, z_i \rangle \leq -(1 - \mu_1)\mu_3 \alpha \|z_i\| + \tfrac{1}{2} L_m \|z_i\|^2 \tag{4.37}$$

for $i \geq i_1$. Using (4.32), we now deduce that

$$m(x + z_i) - m(x) - \mu_1 \langle g, z_i \rangle \leq 0 \tag{4.38}$$

for all $i \geq i_1$. As a consequence, (4.13) is always violated for sufficiently large i, and (4.30) is therefore impossible.

We thus consider the case where $z_i^u = z_0^u$ for all i. This implies that (4.13) is always false and that the algorithm either stops through (4.20) (in which case the convergence is clearly finite) or uses (4.19) at each iteration. But the effect of (4.19) is that l_i tends to $\nu_2 \Delta$ as i grows, and therefore (4.17) must fail for sufficiently large i because $\nu_3 < \nu_2$. The algorithm then terminates with (4.20) after finitely many iterations.

We conclude from these two arguments that, for the algorithm to be infinite, one must have that $z_{i_1}^l \neq z_0^l$ for some $i_1 > 0$ and also that $z_{i_2}^u \neq z_0^u$ must be defined for some $i_2 > 0$. But, because the mechanism of the algorithm guarantees that the sequence $\{l_i\}$ is nondecreasing and that the sequence $\{u_i\}$ is nonincreasing, Case 2 in Step 3 therefore occurs for $i = \max(i_1, i_2)$. The RS Algorithm is thus used in (4.23), and Lemma 4.1 again ensures finite temination. □

THEOREM 4.5. *The call* (4.26) *can be used as an implementation of Step* 2 *of Algorithm* 1.

Proof. We have to verify the compatibility of the GCP Algorithm with the conditions of Step 2 in Algorithm 1, that is, we have to check that the step $s_k^C = x_k^C - x_k$ produced by (4.26) does indeed satisfy the conditions (2.30)–(2.35). All these conditions except (2.32) are clearly enforced by the mechanism of the GCP and RS Algorithms. We can therefore restrict our attention to the verification of (2.32) for the two different possible exits of the GCP Algorithm and their associated $s_k^C = x_k^C - x_k$. Dropping again the subscripts k, we have to verify that (4.27) holds with $z = x^C - x$.

The first case is when the GCP Algorithm terminates using (4.20). Then (4.12) ensures that (4.27) holds for $z = z_i$.

The second and last case is when the algorithm terminates through (4.23). The condition (4.12) again ensures that, in this case, (4.27) holds for $z = z_{i+1}^l$ for some $t_{i+1}^l \geq \|z_{i+1}^l\|$, and for $z = z_{i+1}^u$ for some $t_{i+1}^u \geq \|z_{i+1}^u\|$. For clarity of notations, we drop the subscript $i + 1$ below.

We analyze the situation in the plane H containing x, x^l, and x^u, and define, for $t > 0$, the convex sets

$$H_t \overset{\text{def}}{=} \{x + z \in H | \, \langle g, z \rangle \leq -\mu_3 \alpha(t)\}, \tag{4.39}$$

$$S_t \overset{\text{def}}{=} \{x + z \in H | x + z \in X \text{ and } \|z\| \leq t\}, \tag{4.40}$$

and

$$C_t \overset{\text{def}}{=} H_t \cap S_t. \tag{4.41}$$

For a given $t > 0$, H_t is the half-plane of all vectors $x + z \in H$ such that z satisfies (4.27), irrespective of the constraints $t \geq \|z\|$ and $x + z \in X$, while C_t is the subset of H_t for which these constraints hold.

We again distinguish two cases. The first case is when

$$\|z^l\| \geq \|z^u\|. \tag{4.42}$$

Using the first part of Lemma 2.2, we deduce that

$$\langle g, z^u \rangle \leq -\mu_3 \alpha(t^u) \leq -\mu_3 \alpha(t^l), \tag{4.43}$$

and therefore, using the inequality $t^l \geq \|z^l\| \geq \|z^u\|$, that the complete segment $[x^l, x^u]$ belongs to the convex set C_{t^l}. Hence (4.27) holds for t^l at every point of the segment $[x^l, x^u] = R_x[x^l, x^p, x^u]$.

The more complicated second case is when (4.42) fails. The proof proceeds by showing the existence of a continuous feasible path between x^l and x^u, depending on the parameter t, such that, for each point on this path, there is a $t \in [t^l, t^u]$ for which (4.27) holds at this point. To find this path, we first define, for all $t \in [t^l, t^u]$,

$$x_t \overset{\text{def}}{=} \arg\min_{y \in C_t} \|y - x^u\|_2, \tag{4.44}$$

that is, the projection of x^u onto the convex set C_t. We note that both x^l and x_{t^l} belong to C_{t^l}, and hence that the segment $[x^l, x_{t^l}]$ lies in C_{t^l}. We also note that $x^u = x_{t^u} \in C_{t^u}$. Finally, x_t clearly belongs to C_t for all $t \in [t^l, t^u]$ because of (4.44). Furthermore, this set of x_t determines a continuous path, as can be seen by applying Lemma 2.1 to the minimization problem (4.44). The desired path from x^l to x^u then consists of the segment $[x^l, x_{t^l}]$ followed by the path determined by x_t for $t \in [t^l, t^u]$.

To complete the proof of the theorem for this second case, we use the path just obtained to show that (4.27) holds for some t at every point of $R_x[x^l, x^p, x^u]$. We observe here that this restriction belongs to the plane H. We successively consider three parts of the restricted path, and show the desired property for each part in turn. This restricted path is that used by the GCP Algorithm. A case where $\|\cdot\| = \|\cdot\|_\infty$ is illustrated in Fig. 4.

The first part of the restricted path consists of the segment $[x^l, x^r]$ (where $x^r = R_x[x^p]$) which is the restriction of the segment $[x^l, x^p]$. Using Lemma 4.2 and the fact that z^p is a multiple of z^l, we deduce that, for each point $y \in [x^l, x^r]$, there exists a t such that (4.27) is satisfied at this point for $z = y - x$. We also note that the same argument implies the existence of $t^p \geq \|z^p\| = \|z^u\|$ such that (4.27) also holds at z^p.

The second part of the restricted path consists of the segment $[x^f, x^u]$, where $x^f = R_x[x^f]$ is the first feasible point on the segment $[x^p, x^u]$. (Note that $[x^f, x^u]$ may be equal to $[x^p, x^u]$ when x^p is feasible or may be reduced to the point x^u if this is the only feasible point in $[x^p, x^u]$.) The segment $[x^f, x^u]$ is also contained in X and is therefore equal to its restriction. Because (4.27) holds with $t = \min[t^p, t^u]$ both for z^p and z^u, it must also hold, with the same t, for all z such that $z = y - x$ where $y \in [x^f, x^u] \subseteq [x^p, x^u]$.

The third part of the restricted path consists of the restriction of the segment $[x^p, x^f]$. If x^p is feasible, then the path reduces to $x^f = x^p$, and the desired property results from the analysis of the first part of the restricted path. Assume, therefore, that x^p is not feasible. Then the restriction of $[x^p, x^f]$ lies on the intersection of the boundary of X with H. It can therefore be viewed as the prolongation (as defined before Lemma 4.2) of a part of the path from x^l to x^u defined by the segment $[x^l, x_{t^l}]$ followed by $\{x_t | t \in [t^l, t^u]\}$. Lemma 4.2 then guarantees the existence, for each point $y = x + z$ on the restriction of $[x^p, x^f]$, of a t such that (4.27) holds for z. This finally completes the proof. □

The proof of this last theorem also shows that the path used by the GCP Algorithm is not the only possible one. This can be seen, for example, by choosing $\|\cdot\| = \|\cdot\|_2$, in which case the *projected gradient path* (see [29]) is also acceptable (in the sense that each of its points satisfies (4.12)) and may be different from the restricted path used by the GCP Algorithm.

5. Identification of the correct active set. In this section we consider the case where the convex set of feasible points X is defined as the intersection of a finite collection of larger convex sets X_i, that is,

$$X = \bigcap_{i=1}^{m} X_i. \tag{5.1}$$

AS.6. We assume that, for all $i \in \{1, \dots, m\}$, the convex set X_i is defined by

$$X_i = \{x \in \mathbf{R}^n | h_i(x) \geq 0\}, \tag{5.2}$$

where the function h_i is from $\mathbf{R}^n$ into $\mathbf{R}$ and is continuously differentiable.

We will be interested in the behaviour of the class of algorithms presented in §2 as the iterates $\{x_k\}$ approach a limit point x_*. More precisely, if we define the *active set* at the point $x \in X$ by

$$A(x) = \{i \in \{1, \dots, m\} | h_i(x) = 0\} \tag{5.3}$$

(note that $A(x)$ may be empty if X has a nonempty interior that contains x), the question we wish to analyze can then be phrased as "Is $A(x_k) = A(x_*)$ for k large enough?"

We temporarily restrict ourselves to the case where only inequality constraints are present. This is indeed the case where the constraints identification problem is most apparent. We will discuss the introduction of linear equality constraints in §7.2.

5.1. The assumptions. Clearly, our present assumptions are too general for such an analysis, and we need to strengthen them both from the algorithmic and the geometric point of view.

We first state precisely the additional conditions that are required in Algorithm 1. The idea is that the active constraints at the GCP x_k^C, indexed by $A(x_k^C)$, should be a good estimate of the constraints active at the limit point x_* when k is large enough, as in [4] and [9]. The test which ensures that the GCP asymptotically picks up the correct active constraints is motivated as follows. Assume that an iterative procedure is used to solve the linearized problem associated with $\alpha_k(t_k)$ in (2.18). When a step $\hat{s}_k^C$ satisfying condition (2.32) is obtained in the course of this iteration, we investigate if the correct active set has been found. If the current step $\hat{s}_k^C$ does not approximately minimize the linearized model *with respect to the constraints in* $A(x_k + \hat{s}_k^C)$, we anticipate that this is because the correct active set has not yet been determined. Consequently, additional constraints may need to be considered, for otherwise, the minimizer may be too far away—at infinity in the case of purely linear constraints. We may then choose to continue our iterative procedure. On the other hand, if $\hat{s}_k^C$ approximately minimizes the linearized model with respect to this restricted set of constraints, we may hope that the correct active set has been identified. In the worst case, this may result in finally solving the linearized problem exactly: at the solution $\hat{s}_k^C$, we know that (2.32) obviously holds, but also that this step solves the relaxed version of the same problem *where all constraints that are not in $A(x_k + \hat{s}_k^C)$ have been discarded.* This technique motivates our next assumption, in which we require not only that (2.32) holds at s_k^C, but also that this step approximately minimizes the linearized model with respect to the constraints in $A(x_k^C)$.

More precisely, if the quantity $\alpha_k^C(t)$ is defined, for a given x_k^C and for all $t \geq 0$, by

$$\alpha_k^C(t) \stackrel{\text{def}}{=} \Big| \min_{\substack{x_k+d\in X_k^C \\ \|d\|_{(k)}\leq t}} \langle g_k, d\rangle \Big|, \tag{5.4}$$

where

$$X_k^C \stackrel{\text{def}}{=} \bigcap_{i\in A(x_k^C)} X_i, \tag{5.5}$$

we can then formulate our assumption as follows.

AS.7. For all k sufficiently large, there exists a strictly positive $t_k \geq \|s_k^C\|_{(k)}$ such that

$$\langle g_k, s_k^C\rangle \leq -\mu_3 \alpha_k^C(t_k) \tag{5.6}$$

for some constant $\mu_3 \in (0, 1]$.

We note that, because $X \subseteq X_k^C$,

$$\alpha_k^C(t) \geq \alpha_k(t) \tag{5.7}$$

for all $t \geq 0$, and hence condition (5.6) is stronger than (2.32): it can therefore replace this condition, for large k, in the formulation of Algorithm 1. (This is the reason why the constant μ_3 has been reused in (5.6).)

We also note that it is always possible to satisfy AS.7 and (2.32) together because equality holds in condition (5.7) if x_k^C is chosen as the minimizer of the linearized problem associated with the definition of $\alpha_k(t)$ in (2.18) (see our motivation for AS.7 above).

Once the correct active constraints have been identified by the GCP, one must then make sure they are not dropped at Step 3 of Algorithm 1. This is ensured by the following condition.

AS.8. For all k sufficiently large,

$$A(x_k^C) \subseteq A(x_k + s_k). \tag{5.8}$$

In a way entirely similar to that used in the proof of Lemma 2.2, one can deduce the following properties of $\alpha_k^C(t)$ as a function of t.

LEMMA 5.1. *For all $k \geq 0$,*

1. *the function $t \mapsto \alpha_k^C(t)$ is continuous and nondecreasing for $t \geq 0$,*
2. *the function $t \mapsto \alpha_k^C(t)/t$ is nonincreasing for $t > 0$.*

By analogy with (3.21), we can also define

$$\alpha_k^C \overset{\text{def}}{=} \alpha_k^C(1). \tag{5.9}$$

Using this quantity, we obtain the following counterpart of Theorem 3.7 and Corollary 3.8.

THEOREM 5.2. *Assume that* AS.1–AS.3 *and* AS.7 *hold. Consider any sequence* $\{x_k\}$ *produced by Algorithm* 1 *and assume that* $\alpha_k^C > 0$ *for a* k *sufficiently large. Then there exists a constant* $c_8 \in (0,1]$ *such that*

$$m_k(x_k) - m_k(x_k + s_k) \geq c_8 \alpha_k^C \min\left[1, \Delta_k, \frac{\alpha_k^C}{1+\omega_k^C}\right], \tag{5.10}$$

for all k *sufficiently large. Furthermore, one has that*

$$f(x_k) - f(x_{k+1}) \geq \eta_1 c_8 \alpha_k^C \min\left[1, \Delta_k, \frac{\alpha_k^C}{1+\omega_k^C}\right] \tag{5.11}$$

for all $k \in \mathcal{S}$ *sufficiently large such that* $\alpha_k^C > 0$.

Proof. The proof is entirely similar to those of Theorem 3.7 and Corollary 3.8, with all α_k being replaced by α_k^C, Lemma 2.2 replaced by Lemma 5.1, and the references to (2.32) by references to (5.6). □

We note that we can then pursue the development of §3.3, using α_k^C instead of α_k, and deduce a counterpart of Theorem 3.12.

THEOREM 5.3. *Assume that* AS.1–AS.4 *and* AS.7 *hold. Then, if* $\{x_k\}$ *is a sequence of iterates generated by Algorithm* 1, *one has that*

$$\liminf_{k\to\infty} \alpha_k^C = 0. \tag{5.12}$$

Let us now examine the geometry of the feasible set. We will use the strong constraint qualification based on the independence of the constraint normals at the limit points of the sequence of iterates $\{x_k\}$ generated by Algorithm 1. We first define L to be the set of all limit points of this sequence. Clearly, L is compact because of AS.1.

AS.9. For all $x_* \in L$, the vectors $\{\nabla h_i(x_*)\}_{i \in A(x_*)}$ are linearly independent.

Assumptions AS.6 and AS.9 imply that the normal cone at any $x_* \in L$ is polyhedral and of the form

$$N(x_*) = \left\{ y \in \mathbf{R}^n | y = -\sum_{i \in A(x_*)} \lambda_i \nabla h_i(x_*), \lambda_i \geq 0 \right\}. \tag{5.13}$$

We complete our assumptions by requiring Dunn's *nondegeneracy condition* [13] at every limit point $x_* \in L$. Before stating this condition, we recall that the relative interior of a convex set Y (denoted ri$[Y]$) is its interior when Y is regarded as a subset of its affine hull, that is, the affine subspace with lowest dimensionality that contains Y (see [26, p. 44] for further details). Using this concept, we now express our condition as follows.

AS.10. For every limit point $x_* \in L$, one has that

$$-\nabla f(x_*) \in \text{ri}[N(x_*)]. \tag{5.14}$$

As discussed in [3], this last condition can be viewed as the generalization of the strict complementarity assumption used in [9] and [18]. It was also used in [2] and in [3] in a similar context. As in [2] and [3], we note that AS.9, AS.10, and (5.13) together imply the existence of a unique set of strictly positive multipliers. Thus, for every $x_* \in L$,

$$\nabla f(x_*) = \sum_{i \in A(x_*)} \lambda_i \nabla h_i(x_*) \tag{5.15}$$

for some uniquely defined $\lambda_i > 0$.

We finally assume that the gradient approximations are asymptotically exact.

AS.11.

$$\lim_{k \to \infty} \|e_k\|_{[k]} = 0. \tag{5.16}$$

This assumption is not the weakest one for obtaining the results on constraint identification presented below, but its presence simplifies the exposition. A weaker requirement will be discussed in §7.

We note that none of the above assumptions requires the feasible set to be polyhedral, or even that it has quasi-polyhedral faces (cf. [3]).

5.2. Connected components of limit points. Using the assumptions presented in the preceding subsection, we examine the properties of the unique connected component of limit points of L containing a given $x_* \in L$, which we denote by L_*. We first show the following remarkable fact.

LEMMA 5.4. *Assume that* AS.1–AS.10 *hold. Then, for each connected component of limit points L_*, there exists a set $A(L_*) \subseteq \{1, \ldots, m\}$ such that*

$$A(x_*) = A(L_*) \tag{5.17}$$

for all $x_* \in L_*$.

Proof. Consider two limit points $x_*, y_* \in L_*$ such that

$$A(x_*) \neq A(y_*) \tag{5.18}$$

and assume, without loss of generality, that there exists $j \in \{1, \ldots, m\}$ such that $j \in A(y_*)$ but $j \notin A(x_*)$. Because of the path-connectivity of L_*, we know that there exists a continuous path $z(t)$ such that

$$z(0) = x_*, \quad z(1) = y_*, \quad z(t) \in L_*, \quad \forall t \in [0, 1]. \tag{5.19}$$

Using the continuity of $z(\cdot)$ and $h_j(\cdot)$, the condition (5.18) and the definition of j also ensure the existence of $t_+ \in (0, 1]$ such that

$$j \notin A(z(t)) \quad \forall t \in [0, t_+) \quad \text{and} \quad j \in A(z(t_+)). \tag{5.20}$$

Let us also consider a sequence $\{t_j\}$ in the interval $[0, t_+)$ converging to t_+, and such that $A(z(t_j))$ is constant, and equal to A_- say, for all j. Equation (5.15) implies that

$$\nabla f(z(t_j)) = \sum_{i \in A_-} \lambda_i^-(t_j) \nabla h_i(z(t_j)) \tag{5.21}$$

for all t_j and for some uniquely defined $\lambda_i^-(t_j) > 0$. We now wish to show by contradiction that the sequences $\{\lambda_i^-(t_j)\}$ are bounded for all $i \in A_-$. Assume indeed that the sequence of vectors $\{\lambda^-(t_j)\}$ is unbounded, where these vectors have $\{\lambda_i^-(t_j)\}_{i\in A_-}$ for fixed j as components. In this case, we can select a subsequence $\{t_\ell\} \subseteq \{t_j\}$ such that

$$\|\lambda^-(t_\ell)\|_2 \longrightarrow \infty \quad \text{and} \quad \frac{\lambda^-(t_\ell)}{\|\lambda^-(t_\ell)\|_2} \longrightarrow \lambda^\diamond, \tag{5.22}$$

where $\lambda^\diamond$ is normalized and has at least one strictly positive component. We then obtain from (5.21) that

$$\frac{\nabla f(z(t_\ell))}{\|\lambda^-(t_\ell)\|_2} = \sum_{i\in A_-} \frac{\lambda_i^-(t_\ell)}{\|\lambda^-(t_\ell)\|_2} \nabla h_i(z(t_\ell)), \tag{5.23}$$

which gives in the limit that

$$0 = \sum_{i\in A_-} \lambda_i^\diamond \nabla h_i(z(t_+)), \tag{5.24}$$

using the continuity of $z(\cdot)$, $\nabla f(\cdot)$, and $\nabla h_i(\cdot)$. If we now define

$$A_+ \stackrel{\text{def}}{=} A(z(t_+)), \tag{5.25}$$

we note that (5.20) and the fact that the set $\{x \in \mathbf{R}^n | A(x) \supseteq A_-\}$ is closed ensure that $A_- \subset A_+$. Therefore, because of AS.9 and the fact that $z(t_+) \in L$, we may deduce from (5.24) that all the components of $\lambda^\diamond$ are zero, which we just showed to be impossible. Hence the sequence $\{\lambda^-(t_j)\}$ must be bounded, as well as the sequences of its components. From each of these component's sequences, we may thus extract converging subsequences with limit points λ_i^-. Using the continuity of $z(\cdot)$, $\nabla f(\cdot)$, and $\nabla h_i(\cdot)$, and again taking the limit in (5.21) for these subsequences, we obtain that

$$\nabla f(z(t_+)) = \sum_{i\in A_-} \lambda_i^- \nabla h_i(z(t_+)). \tag{5.26}$$

On the other hand, (5.15) implies that

$$\nabla f(z(t_+)) = \sum_{i\in A_+} \lambda_i^+ \nabla h_i(z(t_+)) \tag{5.27}$$

for some uniquely defined set of $\lambda_i^+ > 0$. But the fact that $A_- \subset A_+$ ensures that (5.26) and (5.27) cannot hold together. Our initial assumption (5.18) is thus impossible, which proves the lemma. □

We now define the distance from any vector x to any compact set Y by

$$\operatorname{dist}(x, Y) \stackrel{\text{def}}{=} \min_{y\in Y} \|x - y\|_2, \tag{5.28}$$

and the neighbourhood of any compact set Y of radius δ by

$$\mathcal{N}(Y, \delta) \stackrel{\text{def}}{=} \{x \in \mathbf{R}^n | \operatorname{dist}(x, Y) \le \delta\}. \tag{5.29}$$

After showing that different active sets cannot appear in a single connected component of limit points, we now show that connected components of limit points corresponding to different active sets are well separated.

LEMMA 5.5. *Assume that* AS.1–AS.10 *hold. Then there exists a $\psi \in (0,1)$ such that*

$$\operatorname{dist}(x_*, L'_*) \geq \psi \tag{5.30}$$

for every $x_* \in L$ and each compact connected component of limit points L'_* such that $A(L'_*) \neq A(x_*)$.

Proof. Consider any $x_* \in L$. To this x_*, we can associate the sets

$$D_i \overset{\text{def}}{=} \{x \in \mathcal{L} | i \in A(x)\} \tag{5.31}$$

for $i \notin A(x_*)$. For each $x_* \in L_*$, there is only a finite number of such sets, and each of them is compact. Because of Lemma 5.4, the sets D_i and L_* are disjoint for all $i \notin A(x_*)$. From the compactness of L, we then deduce the existence of $\psi > 0$ such that

$$\min_{x_* \in L} \min_{i \notin A(x_*)} \min_{x \in D_i} \|x_* - x\|_2 \geq \psi. \tag{5.32}$$

(Without loss of generality, we may assume that $\psi < 1$.) Hence the distance from x_* to any $L'_* \subset L$ such that $A(L'_*)$ contains some index $j \notin A(x_*)$ is bounded below by ψ, which then implies the desired result. □

We next show that, for k large enough, every iterate x_k lies in the neighbourhood of a well-defined connected component of limit points, and also that all constraints that are not binding for this component are also inactive at x_k.

LEMMA 5.6. *Assume that* AS.1–AS.10 *hold. Assume also that the sequence $\{x_k\}$ is generated by Algorithm* 1. *Then there exist a $\delta \in (0, \frac{1}{4}\psi)$, $\psi \in (0,1)$, and a $k_1 \geq 0$ such that, for all $k \geq k_1$, there exists a compact connected component of limit points $L_{*k} \subseteq L$ such that*

$$x_k \in \mathcal{N}(L_{*k}, \delta) \tag{5.33}$$

and

$$A(x) \subseteq A(L_{*k}) \quad \textit{for all } x \in \mathcal{N}(L_{*k}, \delta) \cap \mathcal{L}. \tag{5.34}$$

Proof. Because of the bounded nature of the sequence $\{x_k\}$ (ensured by AS.1), we may divide the complete sequence into a number of subsequences, each of which converges to a given connected component of limit points. For k large enough, x_k therefore lies in the neighbourhood of one such connected component, say L_{*k}. The inclusion (5.33) then follows for δ small enough and for k sufficiently large. We then obtain (5.34) by using (5.32) and imposing the additional requirement that $\delta < \psi/4$. □

We now prove that, if an iterate x_k is close to its associated connected component of limit points, but x_k^C has an incomplete set of active bounds, then α_k^C is bounded away from zero by a small constant independent of k.

LEMMA 5.7. *Assume that* AS.1–AS.11 *hold. Then there exists $k_2 \geq k_1$ (where k_1 is as defined in Lemma* 5.6 *with $\delta < \frac{1}{2}$) such that, if there exists $j \in \{1, \ldots, m\}$ with*

$$j \in A(L_{*k}) \quad \textit{and} \quad j \notin A(x_k^C) \tag{5.35}$$

for some $k \geq k_2$, then

$$\alpha_k^C \geq \epsilon_* \tag{5.36}$$

for some $\epsilon_* \in (0,1)$ independent of k and j.

Proof. Consider, for a given $x_* \in L$ with $A(x_*) \neq \emptyset$ and a given $i \in A(x_*)$, the quantity

$$\alpha_{*i}(x_*) \stackrel{\text{def}}{=} \Big| \min_{\substack{x_*+d \in X_{\{i\}} \\ \|d\|_{(k)} \leq 1/2}} \langle \nabla f(x_*), d \rangle \Big|, \tag{5.37}$$

where $X_{\{i\}}$ is defined by

$$X_{\{i\}} \stackrel{\text{def}}{=} \bigcap_{j \in \{1,\ldots,m\} \setminus \{i\}} X_j. \tag{5.38}$$

$\alpha_{*i}(x_*)$ is the magnitude of the decrease obtained by minimizing the linearized objective from x_* in a ball of radius $\frac{1}{2}$ (in the norm $\|\cdot\|_{(k)}$) when dropping the ith (active) constraint. Because of AS.9 and AS.10, one has that

$$\alpha_{*i}(x_*) > 0 \tag{5.39}$$

for all choices of $x_* \in L$ and $i \in A(x_*)$. Lemma 2.1 and the continuity of ∇f also ensure that $\alpha_{*i}(x_*)$ is a continuous function of x_*. We first minimize $\alpha_{*i}(x_*)$ on the compact set of all $x_* \in L$ such that $i \in A(x_*)$. For each such set, this produces a strictly positive result. We next take the smallest of these results on all i such that $i \in A(x_*)$ for some $x_* \in L$, yielding a strictly positive lower bound $2\epsilon_*$. In short,

$$\min_i \min_{x_*} \alpha_{*i}(x_*) \geq 2\epsilon_* \tag{5.40}$$

for some $\epsilon_* > 0$.

Now consider $k \geq k_1$. Then, by Lemma 5.6, we know that we can associate with x_k a unique connected component of limit points L_{*k} such that (5.33) holds. We then choose a particular $x_{*k} \in L_{*k} \cap \mathcal{N}(x_k, \delta)$, for which we have that

$$\{x_{*k} + d \in X_{\{i\}} \mid \|d\|_{(k)} \leq \tfrac{1}{2}\} \subset \{x_k + d \in X_{\{i\}} \mid \|d\|_{(k)} \leq 1\} \tag{5.41}$$

for all $i \in \{1, \ldots, m\}$, where we used the inequality $\delta < \frac{1}{2}$. Observe also that (5.38) implies that

$$X_{\{i\}} \subseteq X_k^C \tag{5.42}$$

for all $i \notin A(x_k^C)$.

Given a $k \geq k_1$ and such that x_k satisfies (5.35), we now distinguish two cases. The first is when $\alpha_k^C \geq \alpha_{*j}(x_{*k})$, in which case (5.36) immediately follows from (5.40). The second is when $\alpha_k^C < \alpha_{*j}(x_{*k})$. If we define d_k^C and d_* as two vectors satisfying

$$\alpha_k^C = -\left\langle g_k, d_k^C \right\rangle, \quad \|d_k^C\|_{(k)} \leq 1, \quad x_k + d_k^C \in X_k^C, \tag{5.43}$$

and

$$\alpha_{*j}(x_{*k}) = -\left\langle \nabla f(x_{*k}), d_* \right\rangle, \quad \|d_*\|_{(k)} \leq \tfrac{1}{2}, \quad x_{*k} + d_* \in X_{\{i\}}, \tag{5.44}$$

we can write that

$$\begin{aligned} 0 < \alpha_{*j}(x_{*k}) - \alpha_k^C &= \left\langle g_k, d_k^C \right\rangle - \left\langle \nabla f(x_{*k}), d_* \right\rangle \\ &= \left\langle g_k, d_k^C - d_* \right\rangle + \left\langle g_k - \nabla f(x_{*k}), d_* \right\rangle \\ &\leq \left\langle g_k, d_k^C - d_* \right\rangle + \tfrac{1}{2}\|g_k - \nabla f(x_{*k})\|_{[k]}, \end{aligned} \tag{5.45}$$

where we used the inequality (2.14). Now combining (5.41), (5.42), and the definitions of α_k^C, d_k^C, and d_*, we obtain that

$$\langle g_k, d_k^C \rangle = -\alpha_k^C \leq \langle g_k, d_* \rangle . \tag{5.46}$$

Substituting this last inequality in (5.45), using AS.11 and the Lipschitz continuity of ∇f (reducing δ if necessary), we can find $k_2 \geq k_1$ sufficiently large such that

$$0 < \alpha_{*j}(x_{*k}) - \alpha_k^C \leq \epsilon_* \tag{5.47}$$

when $k \geq k_2$. The inequality (5.36) then follows again from (5.40). □

5.3. Active constraints identification. We now wish to show that, given a limit point x_*, the set of active constraints at x_*, that is $A(L_*)$, is identified by Algorithm 1 in a finite number of iterations.

We first show that, if the trust region radius is small and the correct active set is not identified at x_k^C (k large enough), which implies, by Lemma 5.7, that (5.36) holds, then the kth iterate is successful.

LEMMA 5.8. *Assume that* AS.1–AS.9 *hold. Assume furthermore that* (5.36) *holds and*

$$\beta_k \Delta_k \leq \frac{c_8 \epsilon_* (1 - \eta_2)}{c_4} \tag{5.48}$$

for some $k \geq k_2$. *Then iteration* k *is successful* ($k \in \mathcal{S}$) *and* $\Delta_{k+1} \geq \Delta_k$.

Proof. We first observe that (2.28) and the inequalities $c_4 \geq 1$ and $c_8 \leq 1$ imply that

$$\frac{c_8(1 - \eta_2)}{c_4} \leq 1. \tag{5.49}$$

Using Theorem 5.2, (5.36), (5.48), (5.49), and the inequalities $\epsilon_* < 1$ and $\beta_k \geq 1$, one then deduces that

$$f(x_k) - m_k(x_k + s_k) \geq c_8 \epsilon_* \Delta_k. \tag{5.50}$$

But this last inequality, Lemma 3.9, and (5.48) then ensure that

$$|\rho_k - 1| \leq \frac{c_4 \beta_k \Delta_k}{c_8 \epsilon_*} \leq 1 - \eta_2. \tag{5.51}$$

Hence $\rho_k \geq \eta_2$, and the conclusion of the lemma follows. □

We also need the result that the gradient projected onto the tangent cone at a point y having the correct active set goes to zero as both this point and the iterates tend to a connected component of limit points.

LEMMA 5.9. *Assume that* AS.1–AS.11 *hold. Consider any subsequence whose indices form* $K \subseteq \mathbf{N}$ *such that*

$$\lim_{\substack{k \in K \\ k \to \infty}} \operatorname{dist}(x_k, L_*) = 0 \tag{5.52}$$

for some connected component of limit points L_*,

$$\lim_{\substack{k \in K \\ k \to \infty}} \|y_k - x_k\|_{(k)} = 0 \tag{5.53}$$

for some sequence $\{y_k\}_{k\in K}$ *such that* $y_k \in X$, *and*

$$A(y_k) = A(L_*) \tag{5.54}$$

for all $k \in K$. *Then one has that*

$$\lim_{\substack{k\in K\\ k\to\infty}} P_{T(y_k)}(-g_k) = 0. \tag{5.55}$$

Proof. We first note that Lemma 2.1 and the continuity of the constraints' normals imply the continuity of the operators $P_{T(\cdot)}$ and $P_{N(\cdot)}$ as functions of $\{y|A(y) = A(L_*)\}$ in a sufficiently small neighbourhood of L_*. We also observe that the Moreau decomposition of $-g_k$ gives that

$$-g_k = P_{T(y_k)}(-g_k) + P_{N(y_k)}(-g_k). \tag{5.56}$$

Equations (5.54) and (5.56), limits (5.52) and (5.53), and assumptions AS.10 and AS.11 then give (5.55) by continuity. □

Among the finitely many active sets $\{A(x_*)\}_{x_*\in L}$, we now consider a maximal one and denote it by A_*. This is to say that $A_* = A(x_*)$ for some $x_* \in L$ and that

$$A_* \not\subset A(y_*) \tag{5.57}$$

for any $y_* \in L$. We are now in the position to prove that A_* is identified at least on a subsequence of successful iterations.

LEMMA 5.10. *Assume that* AS.1–AS.11 *hold and that the sequence* $\{x_k\}$ *is generated by Algorithm* 1. *Then there exists a subsequence* $\{k_i\}$ *of successful iterations such that, for* i *large enough,*

$$A(x_{k_i}) = A_*. \tag{5.58}$$

Proof. We define the subsequence $\{k_j\}$ as the sequence of successful iterations whose iterates approach limit points with active set equal to A_*; that is,

$$\{k_j\} \overset{\text{def}}{=} \{k \in \mathcal{S}|A(L_{*k}) = A_*\}, \tag{5.59}$$

and assume, for the purpose of obtaining a contradiction, that

$$A(x_{k_j+1}) \neq A_* \tag{5.60}$$

for all j large enough. Assume now, again for the purpose of contradiction, that

$$A_* \subseteq A(x_{k_j}^C) \tag{5.61}$$

for such a j. Using successively AS.8, (5.60), and Lemma 5.6, we then deduce that, for j sufficiently large,

$$A_* \subset A(L_{*k_j+1}), \tag{5.62}$$

which is impossible because of (5.57). Hence (5.61) cannot hold, and there must exist a $p_j \in A_* = A(L_{*k_j})$ such that $p_j \notin A(x_{k_j}^C)$ for j large enough. From Lemma 5.7, we then deduce that (5.36) holds for all j sufficiently large. But Theorem 5.2 and the inequalities $\epsilon_* < 1$ and $\beta_{k_j} \geq 1$ then give that

$$\beta_{k_j}[f(x_{k_j}) - f(x_{k_j+1})] \geq \eta_1 c_8 \epsilon_* \min[\beta_{k_j}\Delta_{k_j}, \epsilon_*], \tag{5.63}$$

for j large enough, and thus, using AS.5, that

$$\lim_{j\to\infty} \beta_{k_j}\Delta_{k_j} = 0. \tag{5.64}$$

The inequality $\beta_{k_j} \geq 1$ and (2.11) then give that

$$\|s_{k_j}\|_{(k_j)} \leq \nu_1 \Delta_{k_j} \leq \frac{1}{2}\delta < \frac{\psi}{4} \tag{5.65}$$

for j larger than $j_1 \geq 1$, say. But this last inequality and Lemmas 5.5 and 5.6 imply that x_{k_j+1} cannot jump to the neighbourhood of any other connected component of limit points with a different active set, and hence x_{k_j+1} belongs to $\mathcal{N}(L_*, \delta)$ again for some L_* such that $A(L_*) = A_*$. The same property also holds for the next successful iterate, say x_{k_j+q}, and we have that $A(L_{*k_j+q}) = A_*$. Therefore, the subsequence $\{k_j\}$ is identical to the complete sequence of successful iterations with $k \geq k_{j_1}$. Hence we may deduce from (5.64) that

$$\lim_{\substack{k\to\infty \\ k\in\mathcal{S}}} \beta_k\Delta_k = 0. \tag{5.66}$$

In particular, we have that

$$\beta_k\Delta_k \leq \frac{c_8\gamma_1^2\epsilon_*(1-\eta_2)}{2c_4} \tag{5.67}$$

for all $k \in \mathcal{S}$ sufficiently large. But the mechanism of the algorithm and (5.66) also give the limit

$$\lim_{k\to\infty} \Delta_k = 0. \tag{5.68}$$

As a consequence, we note that, for k large enough, x_k, x_k^C, and $x_k + s_k$ all belong to $\mathcal{N}(L_*, \delta)$ for a single connected component of limit points L_*.

We also note that Lemma 5.8, the fact that (5.36) now holds for $k \in \mathcal{S}$, and (5.66) together imply that

$$k \in \mathcal{S} \Longrightarrow \Delta_{k+1} \geq \Delta_k \tag{5.69}$$

for k large enough.

We can therefore deduce the desired contradiction from (5.69) and (5.68) if we can prove that all iterations are eventually successful.

Assume, therefore, that this is not the case. It is then possible to find a subsequence K of sufficiently large k such that

$$k \notin \mathcal{S} \quad \text{and} \quad k+1 \in \mathcal{S}. \tag{5.70}$$

Note that, because of (2.45) and the nondecreasing nature of the sequence $\{\beta_k\}$, one has that

$$\beta_k\Delta_k \leq \frac{1}{\gamma_1}\beta_{k+1}\Delta_{k+1} \leq \frac{c_8\gamma_1\epsilon_*(1-\eta_2)}{2c_4} \tag{5.71}$$

for $k \in K$ sufficiently large, where we used (5.67) to deduce the last inequality. Now, if one has that

$$A(x_k^C) \subset A(L_*), \tag{5.72}$$

then Lemmas 5.7 and 5.8 together with (5.71) and (2.29) imply that $k \in \mathcal{S}$, which contradicts (5.70). Hence (5.72) cannot hold, and AS.8 together with Lemma 5.6 give that

$$A(x_k + s_k) = A(x_k^C) = A(L_*) \tag{5.73}$$

for all $k \in K$ sufficiently large. Observe now that, since $k \notin \mathcal{S}$, one has that $x_{k+1} = x_k$ because of (2.44), and hence, using (2.12), that

$$\begin{aligned} m_{k+1}(x_{k+1} + s_{k+1}) - m_k(x_k + s_k) &= m_{k+1}(x_k + s_{k+1}) - m_k(x_k + s_k) \\ &= \langle g_{k+1}, s_{k+1}\rangle - \langle g_k, s_k\rangle + \frac{1}{2}[\|s_{k+1}\|_{(k+1)}^2 \omega_{k+1}(m_{k+1}, x_k, s_{k+1}) \\ &\qquad - \|s_k\|_{(k)}^2 \omega_k(m_k, x_k, s_k)] \\ &\geq \langle g_{k+1} - g_k, s_{k+1}\rangle + \langle -g_k, s_k - s_{k+1}\rangle - \frac{1}{2}\nu_1^2 \beta_k \Delta_k^2 - \frac{1}{2}\nu_1^2 \beta_{k+1} \Delta_{k+1}^2. \end{aligned} \tag{5.74}$$

But, using successively the identity $x_k = x_{k+1}$, the Cauchy–Schwarz inequality, AS.3, (2.11), (2.13), and (2.45), we have that

$$\begin{aligned} \langle g_{k+1} - g_k, s_{k+1}\rangle &= \langle g_{k+1} - \nabla f(x_k), s_{k+1}\rangle + \langle \nabla f(x_k) - g_k, s_{k+1}\rangle \\ &= \langle e_{k+1}, s_{k+1}\rangle - \langle e_k, s_{k+1}\rangle \\ &\geq -\|e_{k+1}\|_{[k+1]}\|s_{k+1}\|_{(k+1)} - \|e_k\|_{[k+1]}\|s_{k+1}\|_{(k+1)} \\ &\geq -\|s_{k+1}\|_{(k+1)}\left[\|e_{k+1}\|_{[k+1]} + \sigma_4\|e_k\|_{[k]}\right] \\ &\geq -\nu_1\Delta_{k+1}\left[\kappa_1\Delta_{k+1} + \sigma_4\kappa_1\Delta_k\right] \\ &\geq -\nu_1\kappa_1\Delta_{k+1}^2\left[1 + \tfrac{\sigma_4}{\gamma_1}\right] \end{aligned} \tag{5.75}$$

for all $k \in K$, and also that

$$\begin{aligned} \langle -g_k, s_k - s_{k+1}\rangle &= \left\langle P_{T(x_k+s_k)}(-g_k), s_k - s_{k+1}\right\rangle + \left\langle P_{N(x_k+s_k)}(-g_k), s_k - s_{k+1}\right\rangle \\ &\geq -\|P_{T(x_k+s_k)}(-g_k)\|_{[k]}\|s_k - s_{k+1}\|_{(k)} \\ &\qquad - \left\langle P_{N(x_k+s_k)}(-g_k), P_{T(x_k+s_k)}(s_{k+1} - s_k)\right\rangle \\ &\geq -\|P_{T(x_k+s_k)}(-g_k)\|_{[k]}\|s_k - s_{k+1}\|_{(k)} \\ &\geq -(\sigma_2 + \tfrac{1}{\gamma_1})\|P_{T(x_k+s_k)}(-g_k)\|_{[k]}\nu_1\Delta_{k+1} \end{aligned} \tag{5.76}$$

for all $k \in K$, where we have used the Moreau decomposition of $-g_k$, the fact that $s_{k+1} - s_k \in T(x_k + s_k)$, (2.14), the fact that the cone $T(x_k + s_k)$ is the polar of $N(x_k + s_k)$, (2.11), AS.3, and (2.45). Using (2.45) again, (5.74), (5.75), (5.76), and the nondecreasing nature of $\{\beta_k\}$, we also deduce that, for such k,

$$\begin{aligned} &m_{k+1}(x_{k+1} + s_{k+1}) - m_k(x_k + s_k) \\ &\qquad \geq -\nu_1\Delta_{k+1}\Big[\kappa_1(1 + \tfrac{\sigma_4}{\gamma_1})\Delta_{k+1} + (\sigma_2 + \tfrac{1}{\gamma_1})\|P_{T(x_k+s_k)}(-g_k)\|_{[k]} \\ &\qquad\qquad + \tfrac{\nu_1}{2}(1 + \tfrac{1}{\gamma_1^2})\beta_{k+1}\Delta_{k+1}\Big]. \end{aligned} \tag{5.77}$$

We now observe that, because of (2.37) and (5.68), we have that $\|s_k\|_{(k)}$ tends to zero when k tends to infinity. Now applying Lemma 5.9 using (5.73) (with $y_k = x_k + s_k$) to the subsequence $k \in K$, we deduce from (5.77), (5.55), (5.68), and (5.66) that

$$m_{k+1}(x_{k+1} + s_{k+1}) - m_k(x_k + s_k) \geq -\tfrac{1}{2}c_8\epsilon_*\Delta_{k+1} \tag{5.78}$$

for k large enough in K. On the other hand, we can also apply Theorem 5.2 to iteration $k+1$ and obtain

$$f(x_{k+1}) - m_{k+1}(x_{k+1} + s_{k+1}) \geq c_8\epsilon_*\Delta_{k+1}, \tag{5.79}$$

where we used (5.66), the inequalities $\epsilon_* < 1$ and $\beta_{k+1} \geq 1$, and the fact that (5.36) holds for all sufficiently large $k \in \mathcal{S}$. Hence we obtain that

$$\begin{aligned} f(x_k) - m_k(x_k + s_k) &= f(x_{k+1}) - m_{k+1}(x_{k+1} + s_{k+1}) + m_{k+1}(x_{k+1} + s_{k+1}) \\ &\qquad - m_k(x_k + s_k) \\ &\geq \tfrac{1}{2}c_8\epsilon_*\Delta_{k+1} \\ &\geq \tfrac{1}{2}c_8\gamma_1\epsilon_*\Delta_k \end{aligned} \tag{5.80}$$

for all $k \in K$ sufficiently large. But then, using the definition of ρ_k, Lemma 3.9, and (5.71), one obtains that

$$|\rho_k - 1| \leq \frac{2c_4}{c_8\gamma_1\epsilon_*}\beta_k\Delta_k \leq 1 - \eta_2 \tag{5.81}$$

and hence that $\rho_k \geq \eta_2$ for all $k \in K$ large enough. But this last inequality implies that $k \in \mathcal{S}$, which contradicts (5.70). The condition (5.70) is thus impossible for k sufficiently large. All iterates are eventually successful, which produces the desired contradiction.

As a consequence, (5.60) cannot hold for all j, and we obtain that there exists a subsequence $\{k_p\} \subset \{k_j\}$ such that, for all p,

$$A_* = A(x_{k_p+1}) = A(x_{k_p+q}), \tag{5.82}$$

where $k_p + q$ is the first successful iteration after iteration k_p. The lemma is thus proved if we choose $\{k_i\} = \{k_p + q\}$. □

The last step in our analysis of the active set identification is to show that, once detected, the maximal active set A_* cannot be abandoned for sufficiently large k. This is the essence of the final theorem of this section.

THEOREM 5.11. *Assume that* AS.1–AS.11 *hold and that the sequence* $\{x_k\}$ *is generated by Algorithm* 1. *Then one has that*

$$A(x_*) = A_* \tag{5.83}$$

for all $x_* \in L$, *and*

$$A(x_k) = A_* \tag{5.84}$$

for all k *sufficiently large.*

Proof. Consider $\{k_i\}$, the subsequence of successful iterates such that (5.58) holds, as given by Lemma 5.10. Assume furthermore that this subsequence is restricted to sufficiently large indices, that is, $k_i \geq k_2$ for all i. Assume finally that there exists a subsequence of $\{k_i\}$, say $\{k_p\}$, such that, for each p, there is a j_p with

$$j_p \in A(x_{k_p}) = A_* \quad \text{and} \quad j_p \notin A(x_{k_p+1}). \tag{5.85}$$

Now Lemma 5.6, (5.57), and (5.58) give that $A(L_{*k_p}) = A_*$. Using this observation and AS.8, we obtain that

$$j_p \in A(L_{*k_p}) \quad \text{and} \quad j_p \notin A(x^C_{k_p}) \tag{5.86}$$

for all p. But Lemma 5.7 then ensures that

$$\alpha^C_{k_p} \geq \epsilon_* \tag{5.87}$$

for all p. Combining this inequality with Theorem 5.2 and the relations $\epsilon_* < 1$ and $\beta_{k_p} \geq 1$, one obtains that, for all p,

$$\beta_{k_p}[f(x_{k_p}) - f(x_{k_p+1})] \geq \eta_1 c_8 \epsilon_* \min[\beta_{k_p}\Delta_{k_p}, \epsilon_*]. \tag{5.88}$$

Using AS.5, we then deduce that

$$\lim_{p\to\infty} \beta_{k_p}\Delta_{k_p} = 0. \tag{5.89}$$

Theorem 5.2 and the inequalities $\epsilon_* < 1$ and $\beta_{k_p} \geq 1$ then imply that

$$f(x_{k_p}) - m_{k_p}(x_{k_p} + s_{k_p}) \geq c_8\epsilon_*\Delta_{k_p} \tag{5.90}$$

for all p sufficiently large. On the other hand, we have that, for all k,

$$\begin{aligned} f(x_k) - m_k(x_k + s_k) &\leq |\langle g_k, s_k\rangle| + \beta_k\|s_k\|^2_{(k)} \\ &\leq \alpha_k(\|s_k\|_{(k)}) + \beta_k\nu_1^2\Delta_k^2 \\ &\leq \frac{\alpha_k(\|s_k\|_{(k)})}{\|s_k\|_{(k)}}\nu_1\Delta_k + \beta_k\nu_1^2\Delta_k^2, \end{aligned} \tag{5.91}$$

where we used (3.29), (3.46), (2.18), and (2.11). Combining (5.90) with (5.91) taken at $k = k_p$, applying the third statement of Lemma 2.2, and dividing both sides by Δ_{k_p}, we obtain that

$$c_8\epsilon_* \leq \nu_1\|P_{T(x_{k_p})}(-g_{k_p})\|_{[k_p]} + \beta_{k_p}\nu_1^2\Delta_{k_p}. \tag{5.92}$$

Assuming that the sequence $\{x_{k_p}\}$ converges to some x_* in some L_* (or taking a further subsequence if necessary), using (5.89) and Lemma 5.9 (with $K = \{k_p\}$, $y_k = x_k$ and $A(L_*) = A_*$), we deduce that (5.92) is impossible for p large enough. As a consequence, no such subsequence $\{k_p\}$ exists, and we have that, for large i,

$$A_* \subseteq A(x_{k_i+1}) \subseteq A(L_{*k_i+1}), \tag{5.93}$$

where we used Lemma 5.6 to deduce the last inclusion. But (5.93) and the maximality of A_* impose that

$$A_* = A(x_{k_i+1}) = A(L_{*k_i+1}) \tag{5.94}$$

for i large enough. Hence we deduce that, for sufficiently large i,

$$A(x_{k_i+q}) = A_*, \tag{5.95}$$

where $k_i + q$ is the index of the first successful iteration after iteration k_i. Hence $k_i + q \in \{k_i\}$. We can therefore repeatedly apply (5.95) and deduce that

$$\{k_i\} = \{k \in \mathcal{S} | k \ \text{is sufficiently large}\}, \tag{5.96}$$

and also that $A(x_k) = A_*$ for all $k \in \mathcal{S}$ large enough, hence proving (5.84). Moreover, A_* is then the only possible active set for the limit points, which proves (5.83). □

6. Convergence to a minimizer. The purpose of this section is to analyze conditions under which the complete sequence of iterates produced by Algorithm 1 can be shown to converge to a single limit point. By Corollary 3.16 and AS.11, this limit point is, of course, critical. We will assume in this section that there are infinitely many successful iterations. Indeed, the convergence of the sequence of iterates is trivial if all iterations are unsuccessful for sufficiently large k.

We define C_* as the set of feasible points whose active set is the same as that of all the limit points, that is,

$$C_* \stackrel{\text{def}}{=} \{x \in X | A(x) = A_*\}. \tag{6.1}$$

We also define $V(x)$ as the plane tangent to the constraints indexed by A_*, that is

$$V(x) \stackrel{\text{def}}{=} \{z \in \mathbf{R}^n | J_*(x)z = 0\}, \tag{6.2}$$

where $J_*(x)$ is the Jacobian matrix whose rows are equal to $\{\nabla h_i(x)^T\}_{i \in A_*}$.

As we wish to use the second-order information associated with the objective function, we must clearly assume that it exists.

AS.12. The objective function $f(\cdot)$ is twice continuously differentiable in an open domain containing X.

We can now prove that if the model curvature along successful steps is asymptotically uniformly positive and if a limit point is an isolated local minimizer, then the complete sequence of iterates converges to this single limit point. In the statement of this result we use the second-order sufficiency condition that the Hessian of the objective is positive definite on the tangent plane to the constraints at the solution (see, e.g., Theorems 6.1 and 6.2 in [4]), which guarantees the isolated character of the minimizer.

THEOREM 6.1. *Assume that* AS.1–AS.12 *hold, that the sequence* $\{x_k\}$ *is generated by Algorithm* 1, *and that the set* S *is infinite. Assume also that there is an* $\epsilon > 0$ *such that*

$$\liminf_{\substack{k \in S \\ k \to \infty}} \omega_k(m_k, x_k, s_k) \geq \epsilon \tag{6.3}$$

and that, for some $x_* \in L$, $\nabla^2 f(x_*)$ *is positive definite on the corresponding tangent plane* $V(x_*)$. *Then*

$$\lim_{k \to \infty} x_k = x_*. \tag{6.4}$$

Proof. We first observe that x_* is a critical point because of AS.11 and Corollary 3.16. We consider $\{x_{k_i}\}$, a subsequence of successful iterates converging to x_*. We now choose $\delta_1 > 0$ small enough to ensure the following two conditions. The first is that we can define $Z(x)$, a matrix whose columns form a continuous basis for the tangent plane $V(x)$. The existence of such a basis is ensured in a sufficiently small neighbourhood $\mathcal{N}(x_*, \delta_1)$ of x_* by assumptions AS.6 and AS.9. The second condition is that $Z(x)^T \nabla^2 f(x) Z(x)$ (that is, $\nabla^2 f(x)$ restricted to the subspace $V(x)$) is uniformly positive definite in $\mathcal{N}(x_*, \delta_1) \cap C_*$.

We now introduce

$$\delta_* \stackrel{\text{def}}{=} \frac{\epsilon \delta_1}{4\sigma_2 + \epsilon} < \delta_1 \tag{6.5}$$

and define $f_{\mathcal{P}}$ to be the largest value of the objective such that the level set

$$\mathcal{P} \overset{\text{def}}{=} \{x \in \mathcal{N}(x_*, \delta_1) \cap C_* | f(x) \le f_{\mathcal{P}}\} \subset \mathcal{N}(x_*, \delta_*), \tag{6.6}$$

which is possible because the positive definiteness of $Z(x)^T \nabla^2 f(x) Z(x)$ in $\mathcal{N}(x_*, \delta_1) \cap C_*$ guarantees the strict convexity of $f(x)$ in this set.

We then use Theorem 5.11 and choose i_1 such that $k_{i_1} \ge 0$ is sufficiently large to guarantee that, for all $i \ge i_1$,

$$x_{k_i} \in \mathcal{P}, \tag{6.7}$$

and also, for all $k \in \mathcal{S}$ with $k \ge k_{i_1}$,

$$x_k \in C_* \tag{6.8}$$

and

$$\omega_k(m_k, x_k, s_k) \ge \tfrac{1}{2}\epsilon. \tag{6.9}$$

We note that, for $k \ge 0$,

$$s_k \in T(x_k). \tag{6.10}$$

Because of (6.8) and Lemma 5.9 with $y_k = x_k$, we deduce that

$$\|P_{T(x_k)}(-g_k)\|_{[k]} \le \delta_* \tag{6.11}$$

for all $k \in \mathcal{S}$ larger than $k_{i_2} \ge k_{i_1}$, say.

Now consider

$$0 > m_{k_i}(x_{k_i} + s_{k_i}) - m_{k_i}(x_{k_i}) = \langle g_{k_i}, s_{k_i}\rangle + \tfrac{1}{2}\|s_{k_i}\|^2_{(k_i)} \omega_{k_i}(m_{k_i}, x_{k_i}, s_{k_i}), \tag{6.12}$$

where the equality results from (3.29) and the inequality from the definition of the step s_{k_i}. Using successively (6.12), (6.9), the Moreau decomposition of $-g_{k_i}$, and (6.10), we then deduce that

$$\|s_{k_i}\|_{(k_i)} < \frac{-2}{\omega_{k_i}(m_{k_i}, x_{k_i}, s_{k_i})} \frac{\langle g_{k_i}, s_{k_i}\rangle}{\|s_{k_i}\|_{(k_i)}} \le \frac{4}{\epsilon} \frac{|\langle P_{T(x_{k_i})}(-g_{k_i}) s_{k_i}\rangle|}{\|s_{k_i}\|_{(k_i)}} \tag{6.13}$$

for $i \ge i_2$. Hence, using (2.14) and (6.11),

$$\|s_{k_i}\|_{(k_i)} \le \frac{4}{\epsilon}\|P_{T(x_{k_i})}(-g_{k_i})\|_{[k_i]} \le \frac{4\delta_*}{\epsilon} \tag{6.14}$$

for $i \ge i_2$. Using this last relation, the equivalence of norms, and the triangle inequality, we obtain that, for such i,

$$\|x_{k_i+1} - x_*\|_2 \le \|s_{k_i}\|_2 + \|x_{k_i} - x_*\|_2 \le \left[\frac{4\sigma_2}{\epsilon} + 1\right] \delta_* = \delta_1. \tag{6.15}$$

We now observe that $k_i \in \mathcal{S}$ implies $f(x_{k_i+1}) < f(x_{k_i}) \le f_{\mathcal{P}}$. Hence $x_{k_i+1} \in \mathcal{P}$ and all conditions that were satisfied at x_{k_i} are again satisfied at the next successful iteration after k_i. The argument can therefore be applied recursively to show that

$$x_{k_i+j} \in \mathcal{P} \subset \mathcal{N}(x_*, \delta_1) \tag{6.16}$$

for all $j \ge 1$. Since δ_1 is arbitrarily small, this proves the convergence of the complete sequence $\{x_k\}$ to x_*. □

7. Discussion and extensions.

The purpose of this section is to further discuss aspects of the theory presented above, from the point of view of both practical implementation and interesting theoretical extensions.

7.1. Simple relaxation-based tests for inexact projections.

A computational difficulty in the framework of Algorithm 1 is the practical enforcement of condition (4.12) in the GCP calculation. Indeed, although the left-hand side can be readily calculated for any vector z, the right-hand side contains the quantity $\alpha(t_i)$, which may not be available. However, an upper bound on $\alpha(t_i)$ can often be derived in the following way.

Assume, for example, that we have computed a candidate for the GCP step, say z_i, such that

$$\|z_i\| \leq t_i \quad \text{and} \quad |\langle g, z_i\rangle| = \alpha(\|z_i\|). \tag{7.1}$$

The last of these conditions says merely that z_i minimizes the linearized model in a "ball" of radius $\|z_i\|$. The aim is then to verify that z_i satisfies (4.12), i.e., that z_i gives a large enough reduction of this linearized model compared to that obtained by the minimizer in a ball of radius $t_i \geq \|z_i\|$. Using the definition of $\alpha(t_i)$ and the second part of Lemma 2.2, it is easy to see that

$$\alpha(t_i) \leq t_i \frac{|\langle g, z_i\rangle|}{\|z_i\|}, \tag{7.2}$$

and (4.12) can thus be guaranteed by checking the stronger condition

$$\langle g, z_i\rangle \leq -\mu_3 t_i \frac{|\langle g, z_i\rangle|}{\|z_i\|}, \tag{7.3}$$

which is equivalent to verifying that

$$\|z_i\| \geq \mu_3 t_i. \tag{7.4}$$

The situation described by (7.1) is far from being unrealistic. It may arise, for example, if $\alpha(t_i)$ is computed by an iterative method starting from x and ensuring (7.1) at each of its iterations.

Another interesting case is when X is polyhedral and $\|\cdot\|_{(k)}$ is the infinity norm for all k. We then find a vector z_i satisfying (4.12) by applying a simplex-like method to the linear programming problem (2.18). Using the fact that the current iterate is feasible and adding slack variables if necessary, this problem can then be rewritten (again dropping the k's) as

$$\min \langle g, d\rangle \tag{7.5}$$

subject to the constraint

$$Ad = 0 \tag{7.6}$$

and the componentwise inequalities

$$l \leq d \leq u \tag{7.7}$$

for some constraint matrix A and some vectors of lower and upper bounds l and u, depending on the value of t in (2.18) (or, equivalently, of t_i in (4.12)). If we use a simplex-based method for solving this problem, we calculate, at each iteration of this method, an admissible iterate d_ℓ and an associated admissible basis B_ℓ. It is then easy to compute

$$\pi_\ell = g_{B_\ell}^T B_\ell^{-1} \quad \text{and} \quad \mu_{\ell j} = \max(0, \pi_\ell A e_j - g_j) \quad (j = 1, \dots, n), \tag{7.8}$$

where g_{B_ℓ} is the basic part of g, and e_j is the jth vector of the canonical basis of $\mathbf{R}^n$. Remarkably, π_ℓ and the vector μ_ℓ (whose components are the $\mu_{\ell j}$) provide an admissible point for the problem

$$\max - \langle Al, \pi \rangle - \langle u - l, \mu \rangle + \langle g, l \rangle \tag{7.9}$$

subject to

$$\pi A - \mu \leq g \tag{7.10}$$

and

$$\mu \geq 0. \tag{7.11}$$

But this problem is the dual of problem (7.5)–(7.7) after the change of variables $d' = d - l$. As a consequence, we can use the weak duality theorem for linear programming (see, e.g., [17, p. 40]) and deduce that $\langle Al, \pi_\ell \rangle + \langle u - l, \mu_\ell \rangle - \langle g, l \rangle$ is an upper bound on the value of $\alpha(t_i)$ in (4.12). We may then stop our simplex-based algorithm as soon as

$$|\langle g, d_\ell \rangle| \geq \mu_3 \min_{r=1,\dots,\ell} \left[\langle Al, \pi_r \rangle + \langle u - l, \mu_r \rangle - \langle g, l \rangle \right], \tag{7.12}$$

since this condition implies

$$|\langle g, d_\ell \rangle| \geq \mu_3 \alpha(t_i), \tag{7.13}$$

thus ensuring (4.12) for $z_i = d_\ell$. This technique therefore allows for the inexact solution of the linear program implicit in (2.18).

We also note that the use of interior point methods for linear programming (see, e.g., [27]) seems quite attractive for solving the same problem in the case where $\|\cdot\|$ is a polyhedral norm and X is polyhedral. These algorithms indeed provide a sequence of feasible approximate solutions together with an estimate of the corresponding duality gaps, which can then be used to stop the process as soon as condition (4.12) is satisfied.

7.2. Constraint identification in the presence of linear equations. We now consider the case where the feasible domain X is defined not only by a set of convex inequalities (as in AS.6) but also by a set of independent linear equations of the form

$$p_i(x) = 0, \qquad i = 1, \dots, q, \tag{7.14}$$

where each of the p_i is an affine function from $\mathbf{R}^n$ into $\mathbf{R}$.

We first observe that identifying the active p_i at the solution is trivial: they are all active by definition. The only remaining question is then to examine whether their very presence can upset the theory developed in §5. We also note that representing an equation by two inequalities of opposite sign does not fit with this theory, because AS.9 is then automatically violated. We therefore need to discuss this case separately.

The simplest way to exploit the identification theory for inequalities is to eliminate the linear equations and view Algorithm 1 as restricted to the affine subspace, say W, where the equations (7.14) hold. We therefore consider the reduction of the original problem to W as follows. Assume that Z is an $n \times n - q$ matrix whose columns form an orthonormal basis of the linear subspace parallel to W. The problem can now be rewritten as

$$\min \hat{f}(y) \overset{\text{def}}{=} f(Zy) \tag{7.15}$$

subject to the constraints

$$\hat{h}_i(y) \overset{\text{def}}{=} h_i(Zy) \geq 0 \qquad (i = 1, \ldots, m), \tag{7.16}$$

where $y \in \mathbf{R}^{n-q}$ (see [15, p. 156] for an introduction to the variable reduction technique). The idea is to show that, if an adapted version of AS.6–AS.11 holds for the problem including the constraints (7.14), then AS.6–AS.11 hold for problem (7.15)–(7.16). The theory of §5 then applies without any modification.

Assumptions AS.6–AS.8 and AS.11 need not be modified for handling the constraints (7.14). Therefore, they also hold for problem (7.15)–(7.16). Assumption AS.9, however, requires the following modification.

AS.9b. For all $x_* \in L$, the vectors $\{\nabla h_i(x_*)\}_{i \in A(x_*)}$ and $\{\nabla p_i(x_*)\}_{i=1}^q$ are linearly independent.

The formal expression of AS.10 is unchanged, but AS.6 and AS.9b imply that the normal cone $N(x_*)$ is now defined by

$$N(x_*) = \left\{ y \in \mathbf{R}^n | y = - \sum_{i \in A(x_*)} \lambda_i \nabla h_i(x_*) - \sum_{i=1}^{q} \xi_i \nabla p_i(x_*), \lambda_i \geq 0 \right\} \tag{7.17}$$

instead of (5.13).

Defining $x_* \overset{\text{def}}{=} Zy_*$ and $\hat{A}(y_*) \overset{\text{def}}{=} A(x_*)$, we first note that AS.9 holds for problem (7.15)–(7.16) as a consequence of AS.9b.

THEOREM 7.1. *Assume that* AS.9b *holds. Then the vectors* $\{\nabla \hat{h}_i(y_*)\}_{i \in \hat{A}(y_*)}$ *are linearly independent.*

The proof of this result belongs to the folklore of mathematical programming, and an easy proof is given in the Appendix A.

Similarly, AS.9b and AS.10 with (7.17) imply that AS.10 holds for problem (7.15)–(7.16), as expressed in the following proposition.

THEOREM 7.2. *Assume that* AS.9b *and* AS.10 *hold with* (7.17). *Then*

$$-\nabla \hat{f}(y_*) \in \text{ri}[\hat{N}(y_*)], \tag{7.18}$$

where

$$\hat{N}(y_*) \overset{\text{def}}{=} \left\{ z \in \mathbf{R}^{n-q} | z = - \sum_{i \in \hat{A}(y_*)} \lambda_i \nabla \hat{h}_i(y_*), \lambda_i \geq 0 \right\}. \tag{7.19}$$

The proof of this result can also be found in Appendix A.

The conclusion of this simple reduction exercise is that all the conditions required for the theory of §5 to hold are satisfied for problems (7.15)–(7.16). The presence of equality constraints therefore does not affect the identification of active inequality constraints in a finite number of iterations of Algorithm 1.

7.3. Constraint identification without linear independence of constraint's normals. One may note that AS.9 is a rather strong constraint qualification and wonder if it can be weakened without affecting the result that "the correct active set" is identified in a finite number of iterations.

In order to answer this question, we first note that Algorithm 1 and the GCP and RS Algorithms do not depend in any way on the particular parametrization (description) of the feasible set X that is used. The constraints functions h_i were indeed introduced only in AS.6 and play no role in the theoretical algorithm. As a consequence, one can clearly add redundant constraints of the form

$$r_i(x) \geq 0 \qquad (i = 1, \ldots, m_r) \tag{7.20}$$

to the set $\{h_i\}_{i=1}^m$ without modifying the result that the algorithm will identify the correct active constraints in the set $\{1, \ldots, m\}$.

Identification of the active redundant constraints in $\{r_i\}_{i=1}^{m_r}$ will then depend on the existence, for each of these constraints, of a set $A_i \subseteq \{1, \ldots, m\}$ such that

$$\{x \in X | A(x) = A_i\} \subseteq \{x \in X | r_i(x) = 0\}. \tag{7.21}$$

If this property holds for r_i, and if $A_i = A_*$, then the activity of r_i will clearly be detected in a finite number of iterations.

For example, if $r_i(x)$ is a multiple of $h_j(x)$, say, and if $j \in A_*$, then r_i is identified as an active constraint in a finite number of iterations. Another example is given by the problem

$$\min x + y \tag{7.22}$$

subject to

$$h_1(x, y) = x \geq 0, \quad h_2(x, y) = y \geq 0, \quad r_1(x, y) = x + 4y \geq 0. \tag{7.23}$$

In this case, the constraint r_1 is active if and only if both h_1 and h_2 are active ($A_1 = \{1, 2\}$). It is therefore detected as an active constraint in a finite number of iterations because the activity of h_1 and h_2 is also.

On the other hand, if we consider the problem

$$\min y \tag{7.24}$$

subject to

$$h_1(x, y) = y - x^2 \geq 0 \quad \text{and} \quad r_1(x, y) = y \geq 0, \tag{7.25}$$

we note that the activity of r_1 at the solution may not be detected in a finite number of iterations. This is because there is no subset $A_1 \subseteq \{1, \ldots, m\} = \{1\}$ such that (7.21) holds.

The above arguments show that a weak active constraint identification is possible without the assumption of linear independence of the constraints' normals. In order to avoid this assumption and to obtain this identification property more directly, several researchers have used a purely geometrical description of the feasible domain for some less general cases (see [3], [4], and [31]). It would be quite interesting to develop such a geometric theory in our framework. This approach seems indeed possible, because a specialization of our identification results to linear inequalities shows that the correct active face of the corresponding convex polytope is identified by Algorithm 1 in a finite number of iterations. This geometric rephrasing of nonlinear constraint identification results is the subject of ongoing research.

7.4. A further discussion on the use of approximate gradients. The technique for handling inexact gradient information, as proposed in §2.2, is identical to that analyzed by Toint in [29], but is quite different from that proposed by Carter in [6] for the unconstrained case, where he only requires that, for all $k \geq 0$,

$$\|D_k^{-T} e_k\|_2 \leq \tau \|D_k^{-T} g_k\|_2 \tag{7.26}$$

for some $\tau \in [0, 1-\eta_2)$ and some symmetric positive definite scaling matrices D_k such that the norms $\|D_k^{-T}(\cdot)\|_2$ do satisfy AS.3. Convergence is proved under this remarkably weak condition by using the property that

$$\lim_{\Delta_k \to 0} (1-\rho_k) \leq \lim_{\Delta_k \to 0} \frac{\|D_k^{-T} e_k\|_2}{\|D_k^{-T} g_k\|_2 \cos \vartheta_k} \leq \lim_{\Delta_k \to 0} \frac{\tau}{\cos \vartheta_k}, \tag{7.27}$$

where ϑ_k is the angle between $D_k s_k$ and $-D_k^{-T} g_k$. The next step in Carter's development is to show that ϑ_k tends to zero when the trust region radius Δ_k tends to zero, for a large class of trust region schemes applied on unconstrained problems. The relation (7.27) then implies that $\rho_k \geq \eta_2$ for small enough Δ_k, and hence the kth iteration is successful, the trust region radius increases, and the algorithm can proceed.

This line of reasoning unfortunately does not apply to constrained problems, where it may well happen that the negative gradient and its approximation both point outside the feasible domain. As a consequence, if x_k lies on the boundary of X, the accuracy level τ requested for e_k may depend on ϑ_k, which can be bounded away from zero as it depends on the angle of $D_k^{-T} g_k$ with the plane tangent to the constraint boundary at x_k. For example, if one considers the problem

$$\min -2x_1 - 2x_2 \tag{7.28}$$

with the constraints

$$x_1 \leq 0 \quad \text{and} \quad x_2 \leq 3, \tag{7.29}$$

and if one assumes that $D_k = I$, x_k is the origin, and $m_k(s) = -2s_1 - \beta s_2$ for some $\beta > 0$, then it is not difficult to verify that

$$\tau \leq (1-\eta_2) \cos \vartheta_k \leq (1-\eta_2)\beta / \sqrt{4+\beta^2} \tag{7.30}$$

is required in (7.26) for the iteration to be successful with $\Delta_{k+1} \geq \Delta_k$, and this value depends on the geometry of the feasible set at x_k (see Fig. 5, where the shaded area corresponds to all steps that produce a model decrease).

A fixed value, as used in [6], is therefore insufficient to cope with a possibly complex geometry of the feasible set X, and an adaptive scheme, such as that suggested by (2.13), is necessary. Furthermore, our purposely broad assumptions (2.37) and (2.38) are too loose to guarantee a well-defined (isotonic, for example) behaviour of ϑ_k as Δ_k tends to zero. Finally, Carter also exploits in his theory the fact that the problem is unconstrained, and thus that $\|D_k^{-T} g_k\|_2$ can be viewed as a criticality measure for the problem at hand. When constraints are present, this is not the case anymore, and the lack of relation between a criticality measure and the right-hand side of (7.26) makes the direct adaptation of this criterion to the constrained framework quite difficult.

Condition (2.13) also differs from the more abstract condition used by Moré in [19], namely that e_k should tend to zero for a converging sequence of iterates. This condition is related to (3.70) and (3.90) in our analysis.

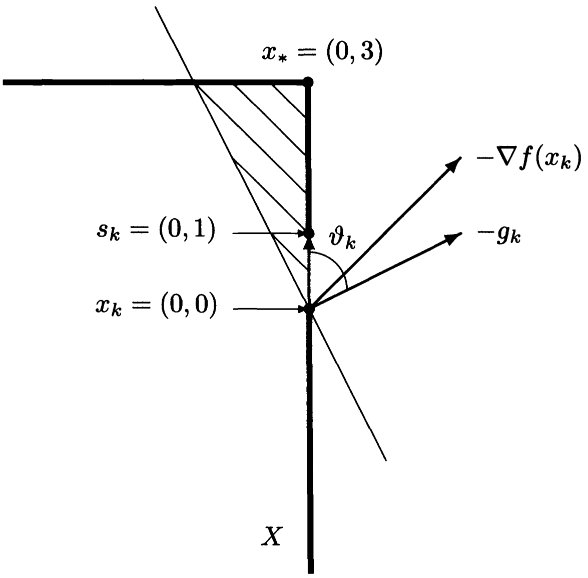


FIG. 5. *The impact of the feasible set geometry on the angle ϑ_k.*

One attractive feature of Carter's condition (7.26) is the fact that the accuracy requirement is relative to the size of the approximating vector g_k, and hence also to the size of the true gradient $\nabla f(x_k)$, as can be seen as follows. From (7.26), we have that

$$\frac{\|D_k^{-T} g_k\|_2}{\|D_k^{-T} \nabla f(x_k)\|_2} \le 1 + \frac{\|D_k^{-T} e_k\|_2}{\|D_k^{-T} \nabla f(x_k)\|_2} \le 1 + \frac{\tau \|D_k^{-T} g_k\|_2}{\|D_k^{-T} \nabla f(x_k)\|_2}, \tag{7.31}$$

and hence, using the fact that $\tau \in [0,1)$,

$$\|D_k^{-T} g_k\|_2 \le \frac{1}{1-\tau} \|D_k^{-T} \nabla f(x_k)\|_2, \tag{7.32}$$

yielding the desired inequality.

It is important to note that our condition (2.13) can be made relative as well, in the form of the criterion

$$\|e_k\|_{[k]} \le \min[\kappa_1 \Delta_k, \kappa_2] \; \|g_k\|_{[k]}, \tag{7.33}$$

where $\kappa_2 \in [0,1)$. This relative criterion does in fact imply (2.13). This implication is based on the following simple result.

LEMMA 7.3. *Assume that* AS.3 *and* (7.33) *hold. Then there exists a constant* $c_9 > 0$ *such that*

$$\|g_k\|_{[k]} \le c_9 \tag{7.34}$$

for all $k \ge 0$.

Proof. Because of (7.33), we have that

$$\|g_k\|_{[k]} \leq \|\nabla f(x_k)\|_{[k]} + \|e_k\|_{[k]} \leq \frac{1}{\sigma_3}\|\nabla f(x_k)\|_2 + \kappa_2\|g_k\|_{[k]}, \tag{7.35}$$

and hence the compactness of $\mathcal{L}$ implies that (7.34) holds with

$$c_9 = \frac{1}{\sigma_3(1-\kappa_2)} \max_{x\in\mathcal{L}} \|\nabla f(x)\|_2. \quad \Box \tag{7.36}$$

As a result of this lemma, we obtain from (7.33) that

$$\|e_k\|_{[k]} \leq c_9 \min[\kappa_1\Delta_k, \kappa_2] \leq c_9\kappa_1\Delta_k, \tag{7.37}$$

and (2.13) therefore holds with κ_1 replaced by $c_9\kappa_1$. The theory developed in this paper is therefore also valid when condition (7.33) is imposed instead of (2.13).

We end this subsection by noting that AS.11 can be omitted without altering the constraint identification result of Theorem 5.11 in the case where the complete sequence of iterates converges to a single limit point x_*, and where the model's gradients g_k converge to a well-defined limit g_* such that $-g_*$ belongs to the relative interior of the normal cone at x_*. This amounts to replacing AS.11 by the following.

AS.11b.

$$\lim_{k\to\infty} x_k = x_* \tag{7.38}$$

and

$$\lim_{k\to\infty} g_k = g_* \quad \text{and} \quad -g_* \in \text{ri}[N(x_*)]. \tag{7.39}$$

The theory of §5 must then be adapted accordingly. In particular, the proof of Lemma 5.7 is modified by replacing $\nabla f(x_*)$ by g_* in (5.37); the minimum over x_* then disappears from (5.40) and the rest of the proof follows.

The second crucial adaptation is the observation that Lemma 5.9 merely requires that

$$\lim_{\substack{k\in\mathcal{K}\\ k\to\infty}} \|e_k\|_{[k]} = 0, \tag{7.40}$$

which is weaker than (AS.11). Condition (7.40) fortunately holds whenever Lemma 5.9 is used: it is ensured by (5.68) and (2.13) in the proof of Lemma 5.10 and by (5.89) and (2.13) in the proof of Theorem 5.11 since $\beta_k \geq 1$ for all k.

Assumption AS.11b seems natural if the correct active set is to be identified at all, since the vectors g_k should clearly provide some consistent first-order information for this property to hold.

7.5. An extension to noisy objective function values. We note that (2.12) (specifying that the model and function values should coincide at the current iterate) is not used anywhere in the convergence theory of §3, except in Lemma 3.9. This leaves some room for a further generalization of Algorithm 1, where not only gradient vectors are allowed to be inexact, but also where the objective function values themselves are not known exactly.

Indeed, define the quantity E_k by

$$E_k \overset{\text{def}}{=} \frac{f(x_k) - m_k(x_k)}{m_k(x_k) - m_k(x_k+s_k)}. \tag{7.41}$$

E_k is therefore a measure of the uncertainty of the objective function value relative to the predicted model decrease for the current step s_k. Clearly, if $|E_k|$ is of the order of one or larger, then the predicted model reduction is comparable to the uncertainty in the objective, and the step s_k is then likely to be completely useless: the algorithm might as well stop at x_k. Conversely, if $|E_k|$ is small, then the predicted model reduction is significant compared to the uncertainty in the objective value, and the algorithm may proceed.

This argument is very nicely supported by the theory, as can be seen as follows. We first note that the term $|f(x_k) - m_k(x_k)|$ now appears in the right-hand side of (3.48) and (3.49), so that (3.47) becomes

$$|f(x_k + s_k) - m_k(x_k + s_k)| \le |f(x_k) - m_k(x_k)| + c_4\beta_k\Delta_k^2. \tag{7.42}$$

We then use this inequality instead of (3.47) to obtain that

$$|\rho_{r-1} - 1| \le 2|E_{r-1}| + \frac{c_4\beta_{r-1}\Delta_{r-1}}{c_3\epsilon} \tag{7.43}$$

instead of (3.57), and the right-hand side of this inequality is smaller than $1 - \eta_2$ provided that we assume the bound

$$|E_k| \le \tfrac{1}{2}\phi(1 - \eta_2) \tag{7.44}$$

for all k and for some $\phi \in [0, 1)$, and provided that (3.53) is replaced by

$$\epsilon < \frac{c_4\beta_0\Delta_0}{\gamma_1 c_3(1 - \eta_2)(1 - \phi)}, \tag{7.45}$$

and (3.54) by

$$\beta_k\Delta_k \le \frac{\gamma_1 c_3(1 - \eta_2)(1 - \phi)}{c_4}\epsilon. \tag{7.46}$$

One then can deduce (3.52) with

$$c_5 = \frac{\gamma_1 c_3(1 - \eta_2)(1 - \phi)}{c_4}\epsilon. \tag{7.47}$$

The rest of the global convergence theory of §3 then follows as before. Hence we conclude that, provided the relative uncertainty on the objective value E_k satisfies the typically very modest bound (7.44) ($|E_k| \le 0.1$ for $\phi = 0.8$ and $\eta_2 = 0.75$), Theorems 3.12 and 3.15 still hold.

8. Conclusions and perspectives. In this paper, we have presented a class of trust region algorithms for problems with convex constraints that uses general norms, approximate gradients, and inexact projections onto the feasible domain. We have proved global convergence of the iterates generated by this class to critical points. Identification of the final set of active inequality constraints in a finite number of iterations is also shown under slightly stronger assumptions. Interestingly, this theory does not assume the locally polyhedral character of the constrained set.

We have also considered practical implementation issues, including an explicit procedure for computing an approximate generalized Cauchy point. Application of these ideas to problems whose linear constraints represent the flow conservation laws in a network is presently being studied.

Appendix A. Proof of Theorems 7.1 and 7.2.

Appendix A. Proof of Theorems 7.1 and 7.2. Considering the variable reduction introduced in §7.2, we first note that

$$\nabla \hat{f}(y) = Z^T \nabla f(x) \quad \text{and} \quad \nabla \hat{h}_i(y) = Z^T \nabla h_i(x). \tag{A.1}$$

A.1. Proof of Theorem 7.2. Assumption AS.10 with (7.17) yields that

$$\nabla f(x_*) = \sum_{i \in A(x_*)} \lambda_i \nabla h_i(x_*) + \sum_{i=1}^{q} \xi_i \nabla p_i(x_*) \tag{A.2}$$

for some $\lambda_i > 0$ and $\xi_i \neq 0$. Applying Z^T to both sides of this relation and noting that $Z^T \nabla p_i(x_*) = 0$ by definition, we obtain the desired conclusion. □

A.2. Proof of Theorem 7.1. Assume that

$$\sum_{i \in \hat{A}(y_*)} \phi_i \nabla \hat{h}_i(y_*) = 0. \tag{A.3}$$

Premultiplying by Z and using (A.1), we obtain that

$$\sum_{i \in A(x_*)} \phi_i Z Z^T \nabla h_i(x_*) = 0. \tag{A.4}$$

Assume, furthermore, for the purpose of contradiction, that

$$\sum_{i \in A(x_*)} \phi_i (I - ZZ^T) \nabla h_i(x_*) \neq 0. \tag{A.5}$$

Since $I - ZZ^T$ is the orthogonal projection onto the subspace spanned by the vectors $\{\nabla p_i(x_*)\}$, we can write that

$$\sum_{i \in A(x_*)} \phi_i (I - ZZ^T) \nabla h_i(x_*) = \sum_{i=1}^{q} \chi_i \nabla p_i(x_*) \tag{A.6}$$

for some χ_i, not all χ_i being zero. Adding (A.4) to (A.6), we obtain

$$\sum_{i \in A(x_*)} \phi_i \nabla h_i(x_*) - \sum_{i=1}^{q} \chi_i \nabla p_i(x_*) = 0, \tag{A.7}$$

which contradicts AS.9b. Hence (A.5) does not hold, and

$$\sum_{i \in A(x_*)} \phi_i (I - ZZ^T) \nabla h_i(x_*) = 0. \tag{A.8}$$

Summing (A.4) and (A.8), and using assumption AS.9b, we deduce that $\phi_i = 0$ for all $i \in A(x_*)$, which yields the desired conclusion. □

Appendix B. Glossary.

Symbol	Definition	Purpose
$\|\cdot\|_{(k)}, \|\cdot\|_{[k]}$	§2.2	iteration dependent norm and its dual
$\alpha_k(t)$	(2.18)	the magnitude of the maximum linearized model decrease achievable in the intersection of X and a ball of radius t centred at x_k
α_k	(3.21)	$\alpha_k(1)$
$\alpha_k^C(t)$	(5.4)	the magnitude of the maximum linearized model decrease achievable in the intersection of X_k^C and a ball of radius t centred at x_k
α_k^C	(5.9)	$\alpha_k^C(1)$
$\alpha_k[x]$	(3.2)	the magnitude of the maximum linearized objective decrease achievable in the intersection of X and a ball of radius 1 centred at x
β_k	(3.46)	monotonically increasing upper bound on the model's curvature along relevant directions (at iteration k)
$\gamma_1, \gamma_2, \gamma_3$	(2.42), (2.43), (2.45)	contraction/expansion factors for trust region updating
δ	Lemma 5.6	
Δ_k	(2.11)	the trust region radius
η_1, η_2	(2.40), (2.42), (2.43)	model accuracy levels
κ_1	(2.13)	the model's gradient accuracy relative to the trust region radius
μ_1, μ_2	(2.33), (2.35)	Goldstein-like constants for the projected search
μ_3	(2.32)	the relative projection accuracy
μ_4	(2.38)	model value relaxation w.r.t. value at the GCP
ν_1	(2.11)	outer trust region radius definition parameter
ν_2	(2.31)	inner trust region radius definition parameter
ν_3, ν_4	(2.34)	minimum steplength condition parameter
ρ_k	(2.39)	ratio of actual (function) to predicted (model) decrease
$\sigma_1, \sigma_2, \sigma_3, \sigma_4$	(2.16), (2.17)	constants in the uniform equivalence of the norms $\|\cdot\|_{(k)}$ and $\|\cdot\|_{[k]}$
ψ	Lemma 5.5	lower bound on the distance between connected components of limit points
$\omega_k(q, x, v)$	(3.29)	the curvature of the function q from x along v
ω_k^C	(3.34)	$= \omega_k(m_k, x_k, s_k^C)$
$A(x)$	(5.3)	the active set at x
A_*	(5.57)	the maximal active set at limit points
B_k	(2.11)	the trust region at iteration k
c_1	Theorem 3.2, (3.3)	uniform equivalence constant for $\alpha_k[x]$
c_2	Lemma 3.6, (3.31)	uniform upper bound on $\omega_k(f, x_k, s)$
c_3	Theorem 3.7, (3.44)	model decrease parameter
c_4	Lemma 3.9, (3.50)	
c_5	Lemma 3.10, (3.60)	
c_6	(3.74)	
c_7	(3.82)	
c_8	Theorem 5.2, (5.10)	
c_9	Lemma 7.3, (7.36)	upper bound on the model's gradient norm
C_t	(4.41)	set of admissible GCP steps of length at most t
C_*	(6.1)	set of feasible points with active set equal to A_*
$\mathrm{dist}(x, Y)$	(5.28)	the distance from x to the compact set Y
D_k	after (7.26)	symmetric positive definite scaling matrix at iteration k
e_k	after (2.13)	difference between the model's and the objective's gradients
E_k	(7.41)	uncertainty of the objective value relative to the predicted model decrease
f	after (2.1)	the objective function
g_k	after (2.12)	the gradient of the model at iteration k, taken at x_k
h_i	(AS.6), (5.2)	inequality constraint functions
H_k	after (2.46)	symmetric approximation to the objective's Hessian at x_k
$J_*(x)$	after (6.2)	the Jacobian matrix of the h_i restricted to rows whose index is in A_* taken at x

Symbol	Definition	Purpose
k_1	Lemma 5.6	
k_2	Lemma 5.7	
K, K^0	(2.4)	cone and its polar
L	before (AS.9)	set of all limit points
$\mathcal{L}$	(2.3)	the intersection of the feasible domain with the level set associated with $f(x_0)$
L_f	after (3.32)	the Lipschitz constant of the objective's gradient
L_m	after (4.32)	the Lipschitz constant of the model's gradient
L_*	§5.2	the connected component of limit points containing x_*
L_{*k}	Lemma 5.6	the connected component of limit points associated with x_k
L'_*	Lemma 5.5, (5.30)	connected component of limit points *not* ćontaining x_*
m_k	§2.2	the model of the objective at iteration k
$N(x)$	(2.6)	the normal cone to X at the feasible point x
$\mathcal{N}(Y, \delta)$	(5.29)	neighbourhood of a compact set Y of radius δ
p_i	(7.14)	linear equality constraint functions
P_X	before (2.5)	the orthogonal projection onto X
r_i	(7.20)	redundant inequality constraint functions
$\text{ri}(Y)$	before (5.14)	relative interior of the convex set Y
$R_x[\cdot]$	(4.1)	the restriction operator
$R_x[x^l, x^p, x^u]$	§4.1	restriction of the path $[x^l, x^p, x^u]$
s_k^C	(2.30)–(2.35)	the step from x_k to the GCP
s_k	(2.37)–(2.38)	the step at iteration k
$\mathcal{S}$	end of §2.3	the set of indices of successful iterations
t_k	before (2.30)	upper bound on the length of the GCP step
$T(x)$	(2.7)	the tangent cone to X at the feasible point x
$V(x)$	(6.2)	the linear subspace such that $x + V(x)$ is the tangent plane at x to the constraints indexed by A_*
W	§7.2	affine subspace determined by the linear equality constraints p_i
x^f	§4.2	
x^l	§4.2	
x^p	§4.2	
x^r	§4.2	
x^u	§4.2	
x_k	§2.2	the iterate of Algorithm 1 at iteration k
$x_k(\theta)$	(2.48)	the projected gradient path starting from x_k
x_k^C	(2.36)	the Generalized Cauchy Point
x_t	(4.44)	the projection of x^u on the convex set C_t
x_*	(3.1)	a critical point
X	after (2.2)	the convex feasible domain
X_i	(5.1), (5.2)	convex sets whose intersection is the feasible domain
X_k^C	(5.5)	relaxation of the feasible domain determined by the constraints active at the GCP
Z	§7.2	matrix whose columns form an orthonormal basis of the linear subspace parallel to W
$Z(x)$	before (6.5)	matrix whose columns form a continuous basis for $V(x)$

Appendix C. Summary of the assumptions.

AS.1. The set $\mathcal{L}$ is compact.

AS.2. The objective function $f(x)$ is continuously differentiable and its gradient $\nabla f(x)$ is Lipschitz continuous in an open domain containing $\mathcal{L}$.

AS.3. There exist constants $\sigma_1, \sigma_3 \in (0,1]$ and $\sigma_2, \sigma_4 \geq 1$ such that, for all $k_1 \geq 0$ and $k_2 \geq 0$,

$$\sigma_1 \|x\|_{(k_1)} \leq \|x\|_{(k_2)} \leq \sigma_2 \|x\|_{(k_1)} \quad \text{and} \quad \sigma_3 \|x\|_{[k_1]} \leq \|x\|_{[k_2]} \leq \sigma_4 \|x\|_{[k_1]}$$

for all $x \in \mathbf{R}^n$.

AS.4. The series

$$\sum_{k=0}^{\infty} \frac{1}{\beta_k}$$

is divergent.

AS.5. The limit

$$\lim_{k \to \infty} \beta_k [f(x_k) - f(x_{k+1})] = 0$$

holds.

AS.6. For all $i \in \{1, \ldots, m\}$, the convex set X_i is defined by

$$X_i = \{x \in \mathbf{R}^n | h_i(x) \geq 0\},$$

where the function h_i is from $\mathbf{R}^n$ into $\mathbf{R}$ and is continuously differentiable.

AS.7. For all k sufficiently large,

$$\left\langle g_k, s_k^C \right\rangle \leq -\mu_3 \alpha_k^C(t_k),$$

for some strictly positive $t_k \geq \|s_k^C\|_{(k)}$ and some constant $\mu_3 \in (0,1]$.

AS.8. For all k sufficiently large,

$$A(x_k^C) \subseteq A(x_k + s_k).$$

AS.9. For all $x_* \in L$, the vectors $\{\nabla h_i(x_*)\}_{i \in A(x_*)}$ are linearly independent.

AS.10. For every limit point $x_* \in L$,

$$-\nabla f(x_*) \in \mathrm{ri}[N(x_*)].$$

AS.11.

$$\lim_{k \to \infty} \|e_k\|_{[k]} = 0.$$

AS.12. The objective function $f(\cdot)$ is twice continuously differentiable in an open domain containing X.

AS.9b. For all $x_* \in L$, the vectors $\{\nabla h_i(x_*)\}_{i \in A(x_*)}$ and $\{\nabla p_i(x_*)\}_{i=1}^q$ are linearly independent.

AS.11b.

$$\lim_{k \to \infty} x_k = x_*, \quad \lim_{k \to \infty} g_k = g_* \quad \text{and} \quad -g_* \in \mathrm{ri}[N(x_*)].$$

Acknowledgment. The authors are grateful to W. R. Pulleyblank for discussions about §7.1.

REFERENCES

[1] M. Bierlaire, Ph.L. Toint, and D. Tuyttens, *On iterative algorithms for linear least squares problems with bound constraints*, Linear Algebra Appl., 143 (1991), pp. 111-143.

[2] J. V. Burke, *On the identification of active constraints* II: *The nonconvex case*, SIAM J. Numer. Anal., 27 (1990), pp. 1081–1101.

[3] J. V. Burke and J. J. Moré, *On the identification of active constraints*, SIAM J. Numer. Anal., 25 (1988), pp. 1197–1211.

[4] J. V. Burke, J. J. Moré, and G. Toraldo, *Convergence properties of trust region methods for linear and convex constraints*, Math. Programming, 47 (1990), pp. 305–336.

[5] 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.

[6] R. G. Carter, *On the global convergence of trust region algorithms using inexact gradient information*, SIAM J. Numer. Anal., 28 (1991), pp. 251–265.

[7] ———, *Safeguarding Hessian approximations in trust region algorithms*, manuscript.

[8] 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., 1985, pp. 71–82.

[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. Correction, same journal, 26 (1989), pp. 764–767.

[10] ———, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comp., 50 (1988), pp. 399–430.

[11] A. R. Conn, N. I. M. Gould, M. Lescrenier, and Ph.L. Toint, *Performance of a multifrontal scheme for partially separable optimization*, Report 88/4, Dept. of Mathematics, FUNDP, Namur, Belgium, 1988.

[12] J. E. Dennis and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[13] J. C. Dunn, *On the convergence of projected gradient processes to singular critical points*, J. Optim. Theory Appl., 55 (1987), pp. 203–216.

[14] A. V. Fiacco, *Introduction to sensitivity and stability analysis in nonlinear programming*, Academic Press, New York, 1983.

[15] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, New York, 1981.

[16] W. A. Gruver and E. Sachs, *Algorithmic methods in optimal control*, Pitman, Boston, MA, 1980.

[17] J. L. Kennington and R. V. Helgason, *Algorithms for Network Programming*, John Wiley, New York, 1980.

[18] M. Lescrenier, *Partially separable optimization and parallel computing*, Report 86/5, Dept. of Mathematics, FUNDP, Namur, Belgium, 1986.

[19] J. J. Moré, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming: The State of the Art, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 258–287.

[20] ———, *Trust regions and projected gradients*, in System Modelling and Optimization, M. Iri and K. Yajima, eds., Proc. 13th IFIP Conf. System Modelling and Optimization, Tokyo, August 31–September 4, 1987, Lecture Notes in Control and Inform. Sci., 113, Springer-Verlag, Berlin, 1988, pp. 1–13.

[21] J. J. Moré and G. Toraldo, *Algorithms for bound constrained quadratic programming problems*, Numer. Math., 55 (1989), pp. 377–400.

[22] J. J. Moreau, *Décomposition orthogonale d'un espace hilbertien selon deux cônes mutuellement polaires*, Comptes-Rendus Académie des Sciences, 255 (1962), pp. 238–240.

[23] M. J. D. Powell, *A new algorithm for unconstrained optimization*, in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, New York, 1970.

[24] ———, *On the global convergence of trust region algorithms for unconstrained minimization*, Math. Programming, 29 (1984), pp. 297–303.

[25] M. J. D. Powell and Y. Yuan, *A trust region algorithm for equality constrained optimization*, Report DAMTP1986–NA2, Dept. of Applied Mathematics and Theoretical Physics, Univ. of Cambridge, UK, 1986.

[26] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[27] M. J. Todd, *Recent developments and new directions in linear programming*, in Mathematical Programming: Recent Developments and Applications, M. Iri and K. Tanabe, eds., Kluwer Academic Publishers, Norwell, MA, 1989.

[28] Ph.L. Toint, *Convergence properties of a class of minimization algorithms that use a possibly unbounded sequence of quadratic approximations*, Report 81/1, Dept. of Mathematics, FUNDP, Namur, Belgium, 1981.

[29] ———, *Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space*, IMA J. Numer. Anal., 8 (1988), pp. 231–252.

[30] A. Vardi, *A trust region algorithm for equality constrained minimization: convergence properties and implementation*, SIAM J. Numer. Anal., 22 (1985), pp. 575–591.

[31] S. WRIGHT, *Convergence of* SQP*-like methods for constrained optimization*, SIAM J. Control Optim., 27 (1989), pp. 13–26,

[32] Y. YUAN, *Conditions for convergence of trust region algorithms for nonsmooth optimization*, Math. Programming, 31 (1985), pp. 220–228.

[33] 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.

SIAM J. OPTIMIZATION
Vol. 3, No. 2, pp. 223-235, May 1993

A FINITE SMOOTHING ALGORITHM FOR LINEAR $\mathbb{l}_1$ ESTIMATION*

KAJ MADSEN† AND HANS BRUUN NIELSEN†

Abstract. In this paper a new method for solving the linear $\mathbb{l}_1$ problem is described, analysed, and tested. The method is based on smoothing the nondifferentiable $\mathbb{l}_1$ function. The smoothing can be done in a well-conditioned manner since the method has finite convergence. Extensive numerical tests demonstrate significant superiority to existing simplex-type codes. Furthermore, the tests show that the new algorithm is very well suited for vector processing.

Key words. Huber estimator, smoothing, Newton's method, finite convergence

AMS subject classifications. 62J05, 65F20, 90C05

1. Introduction. In this paper we consider the linear $\mathbb{l}_1$ estimation problem, i.e., we consider the problem of minimizing the functional

$$F(\mathbf{x}) \equiv \sum_{j=1}^{m} |r_j(\mathbf{x})|, \tag{1}$$

where

$$\begin{gathered} r_j(\mathbf{x}) = \mathbf{a}_j^T\mathbf{x} - b_j, \qquad j = 1, \dots, m, \\ \Updownarrow \\ \mathbf{r}(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b} \qquad (\mathbf{A}^T = [\mathbf{a}_1, \dots, \mathbf{a}_m]) \end{gathered}$$

is a set of linear functionals in R^n. We consider a continuation method for minimizing (1). At each iteration the nondifferentiable function F is approximated by a smooth function, the Huber M-estimator [7],

$$F_\gamma(\mathbf{x}) = \sum_{j=1}^{m} \rho_\gamma(r_j(\mathbf{x})), \tag{2}$$

where

$$\rho_\gamma(t) = \begin{cases} t^2/(2\gamma) & \text{if } |t| \leqq \gamma, \\ |t| - \gamma/2 & \text{if } |t| > \gamma, \end{cases} \tag{3}$$

and the *threshold* γ is a positive real number. Clearly, F_γ is continuously differentiable, and it can be demonstrated (see, e.g., Theorem 1 below) that a minimizer $\mathbf{x}_\gamma$ of (2) is close to a minimizer $\mathbf{x}_0$ of (1) when γ is close to zero. Furthermore, Theorem 1 shows that the $\mathbb{l}_1$ solution can be detected when $\gamma > 0$ is small enough, i.e., it is not necessary to let γ converge to zero in order to find a minimizer of (1). This observation is essential for the efficiency and the numerical stability of the algorithm to be described in this paper. The algorithm produces a sequence $\mathbf{x}_{\gamma_i}$, $i = 1, \dots, i_0$, of minimizers of (2), where $\{\gamma_i\}$ is a decreasing sequence of positive numbers. When the threshold is small enough, an $\mathbb{l}_1$ solution is detected and the computation stops. The minimizers of (2) are found through a Newton-type iteration [10], and since "warm starts" are used, very little work is necessary to find˙one $\mathbf{x}_{\gamma_i}$ when the previous is known. Extensive numerical testing of the algorithm indicates that the number of different threshold values used only increases very slowly with the size of the problem. For example, for problems with 1000 variables, i_0 is less than 20 on average.

* Received by the editors February 11, 1991; accepted for publication (in revised form) March 19, 1992.

† Institute for Numerical Analysis, Technical University of Denmark, 2800 Lyngby, Denmark.

During the past ten years several authors have used continuation methods for minimizing the $\mathbb{l}_1$ function, or for solving related special problems. Clark and Osborne [2], [14] also use the smoothing (2) in a method which can be used to minimize (1) by letting γ go to zero. However, this method can only decrease γ in very small steps, so it is computationally less efficient than the present method. Furthermore, Clark and Osborne only prove the finiteness of the algorithm in the case where (2) has a unique minimizer for each value of γ. In our corresponding result (Theorem 1) no such restriction is used. Other authors who use continuation methods for problems related to the present one are Knoth [9], Pinar and Zenios [15], and Chen and Harker [3].

It is well known that the linear $\mathbb{l}_1$ problem is closely connected to the linear programming problem. This fact is the basis for most algorithms for minimizing (1). One of the most efficient methods is given by Barrodale and Roberts [1]. It uses a specialized version of the simplex method to solve an LP formulation of the $\mathbb{l}_1$ problem. Various "interior point" algorithms, related to the Karmarkar [8] algorithm, have recently been developed for the linear $\mathbb{l}_1$ problem [4], [16].

We compare our algorithm with the version of the Barrodale-Roberts algorithm, which is in the Harwell Subroutine Library [6]. On a set of pseudo-randomly generated test problems our method seems to be significantly faster for large problems, by a factor which increases with n and m. For $n = 810$, $m = 1620$ our method is on average 19 times faster than the Barrodale-Roberts algorithm. We have not been able to compare this with [4] and [16] since no code is available, but we have compared it with the interior point algorithm OB1 (Marsten [11]) applied to the standard LP formulation of the $\mathbb{l}_1$ problem. However, the code of [11] is sparse, whereas our code and the test matrices are not, and this may be one reason why our method was already more than 50 times faster for $n = 50$, $m = 200$.

The paper is organised as follows. In § 2 the Huber estimator (2) and its relation to F is analysed. The main result is Theorem 1, which shows how an $\mathbb{l}_1$ solution can be calculated from a minimizer of F_γ when the threshold value γ is small enough. The algorithm is defined in § 3 and finite convergence is demonstrated. Finally, in § 4 our code is discussed and a large number of numerical tests with our algorithm and the Barrodale-Roberts algorithm are described. The tests include rank deficient problems.

2. The connection between F and F_γ. In this section γ always denotes a positive real number. When it is convenient we denote F by F_0. Without loss of generality we can assume that $\mathbf{a}_j \neq \mathbf{0}$, $j = 1, \ldots, m$, and that $\mathbf{A}$ has rank n. Otherwise the problem could easily be reformulated to have these properties.

When we analyse the function F_γ it is essential to determine whether $r_j(\mathbf{x}) < -\gamma$, $r_j(\mathbf{x}) > \gamma$, or $|r_j(\mathbf{x})| \leqq \gamma$. These inequalities divide R^n into subregions U_j^-, U_j^+, and U_j separated by the parallel hyperplanes $r_j(\mathbf{x}) = \pm\gamma$. The set of all such hyperplanes is denoted by B_γ:

$$(4) \qquad B_\gamma = \{\mathbf{x} \in R^n | \exists j : |r_j(\mathbf{x})| = \gamma\}.$$

Defining the *sign vector* $\mathbf{s}_\gamma(\mathbf{x}) = (s_1(\mathbf{x}), \ldots, s_m(\mathbf{x}))^T$ by

$$(5) \qquad s_j = s_j(\mathbf{x}) = \begin{cases} -1 & \text{for} \quad r_j(\mathbf{x}) < -\gamma, \\ 0 & \text{for} \quad |r_j(\mathbf{x})| \leqq \gamma, \\ 1 & \text{for} \quad r_j(\mathbf{x}) > \gamma, \end{cases}$$

and introducing

$$(6) \qquad w_j = w_j(\mathbf{x}) \equiv 1 - s_j^2(\mathbf{x}),$$

we can write

$$\rho_\gamma(r_j(\mathbf{x}))=\frac{1}{2\gamma}w_j r_j^2(\mathbf{x})+s_j\left[r_j(\mathbf{x})-\frac{1}{2}\gamma s_j\right]$$

yielding

$$F_\gamma(\mathbf{x})=\frac{1}{2\gamma}\mathbf{r}^T\mathbf{W}_\gamma\mathbf{r}+\mathbf{s}_\gamma^T\left[\mathbf{r}-\frac{1}{2}\gamma\mathbf{s}_\gamma\right], \tag{7}$$

where $\mathbf{W}_\gamma=\mathbf{W}_\gamma(\mathbf{x})$ is the diagonal $m\times m$ matrix with diagonal elements $w_j(\mathbf{x})$, i.e., $\mathbf{W}_\gamma$ has 1 in those diagonal elements that correspond to "small" residuals, and zero elsewhere.

For $\mathbf{x}\in R^n$ the gradient of F_γ is given by

$$\mathbf{F}_\gamma'(\mathbf{x})=\mathbf{A}^T\left[\frac{1}{\gamma}\mathbf{W}_\gamma(\mathbf{x})\mathbf{r}(\mathbf{x})+\mathbf{s}_\gamma(\mathbf{x})\right], \tag{8}$$

and for $\mathbf{x}\in R^n\backslash B_\gamma$ the Hessian exists and is given by

$$\mathbf{F}_\gamma''(\mathbf{x})=\frac{1}{\gamma}\mathbf{A}^T\mathbf{W}_\gamma(\mathbf{x})\mathbf{A}. \tag{9}$$

The gradient is a continuous function in R^n, whereas the Hessian is piecewise constant.

We say that $\mathbf{s}$ is a *γ-feasible sign vector* if there exists $\mathbf{x}\in R^n\backslash B_\gamma$ with $\mathbf{s}_\gamma(\mathbf{x})=\mathbf{s}$. If $\mathbf{s}$ is γ-feasible then $Q_\mathbf{s}$ is defined as the quadratic, which is deduced from (7) by inserting $\mathbf{s}$ instead of $\mathbf{s}_\gamma$. Thus, for any $\mathbf{x}$ with $\mathbf{s}_\gamma(\mathbf{x})=\mathbf{s}$, we have

$$Q_\mathbf{s}(\mathbf{y})=\tfrac{1}{2}(\mathbf{y}-\mathbf{x})^T\mathbf{F}_\gamma''(\mathbf{x})(\mathbf{y}-\mathbf{x})+\mathbf{F}_\gamma'(\mathbf{x})^T(\mathbf{y}-\mathbf{x})+\mathbf{F}_\gamma(\mathbf{x}). \tag{10}$$

Clearly $F_\gamma(\mathbf{y})=Q_\mathbf{s}(\mathbf{y})$ in the domain

$$C_\mathbf{s}=\mathrm{cl}\,\{\mathbf{y}\,|\,\mathbf{s}_\gamma(\mathbf{y})=\mathbf{s}\}. \tag{11}$$

For each $\gamma>0$ and $\mathbf{z}\in R^n$ we have one or several corresponding quadratics $Q_\mathbf{s}$. If $\mathbf{z}\notin B_\gamma$ then $Q_\mathbf{s}$ is characterized by $\mathbf{z}$ and γ only ($\mathbf{s}=\mathbf{s}_\gamma(\mathbf{z})$), but for $\mathbf{z}\in B_\gamma$ the quadratic is not unique. Therefore, we use a *reference*

$$(\gamma,\mathbf{z},\mathbf{s}) \tag{12}$$

to determine the quadratic. We say that

$(\gamma,\mathbf{z},\mathbf{s})$ is a *feasible reference* if $\mathbf{s}$ is a *γ-feasible sign vector* with $\mathbf{z}\in C_\mathbf{s}$, and
$(\gamma,\mathbf{z},\mathbf{s})$ is a *solution reference* if it is feasible and $\mathbf{x}=\mathbf{x}_\gamma$ minimizes F_γ.

The set of indices

$$A_\gamma(\mathbf{x})\equiv\{j\,|\,1\leqq j\leqq m\wedge s_j(\mathbf{x})=0\} \tag{13}$$

is called the *γ-active set* at $\mathbf{x}$ and the subspace

$$V_\gamma(\mathbf{x})\equiv\mathrm{span}\,\{\mathbf{a}_i\,|\,i\in A_\gamma(\mathbf{x})\} \tag{14}$$

is called the *γ-active subspace* at $\mathbf{x}$. (If $A_\gamma(\mathbf{x})$ is empty, then we let $V_\gamma(\mathbf{x})=\{\mathbf{0}\}$.) We can express the *necessary* condition for a minimum of F_γ as follows:

$$\mathbf{0}=\boldsymbol{F}_\gamma'(\mathbf{x})=\frac{1}{\gamma}\sum_{j\in A_\gamma(\mathbf{x})}r_j(\mathbf{x})\mathbf{a}_j+\sum_{j\notin A_\gamma(\mathbf{x})}s_j(\mathbf{x})\mathbf{a}_j, \tag{15}$$

where $j \notin A_\gamma(\mathbf{x})$ means $j \in \{i \mid 1 \leqq i \leqq m \wedge i \notin A_\gamma(\mathbf{x})\}$. Since $|r_j(\mathbf{x})|/\gamma \leqq 1$ for $j \in A_\gamma(\mathbf{x})$, this expression is similar to the necessary condition for a minimizer $\mathbf{y}$ of the $\mathbb{l}_1$ function F: There exists $\{\delta_j\}$ with $|\delta_j| \leqq 1$ such that

$$(16) \qquad \mathbf{0} = \sum_{j \in A_0(\mathbf{y})} \delta_j \mathbf{a}_j + \sum_{j \notin A_0(\mathbf{y})} s_j^* \mathbf{a}_j, \quad \text{with } |\delta_j| \leqq 1,$$

where $A_0(\mathbf{y}) = \{j \mid 1 \leqq j \leqq m \wedge r_j(\mathbf{y}) = 0\}$ and $s_j^* = s_j^*(\mathbf{y}) = \operatorname{sign}\{r_j(\mathbf{y})\}$ (see, e.g., Watson [18]). Since the objectives are convex these necessary conditions are also sufficient.

For each of the functions F and F_γ there exists a minimizer $\mathbf{x}_\gamma$ at which the active subspace has dimension n, i.e., $V_\gamma(\mathbf{x}_\gamma) = R^n$ (see, e.g., [18] and [10]). A minimizer $\mathbf{x}_\gamma$ for which $V_\gamma(\mathbf{x}_\gamma) \neq R^n$ is called a *degenerate* solution.

We denote by $\mathbf{x}_0$ a minimizer of (1), by $\mathbf{x}_\gamma$ a minimizer of (2), and by $\mathbf{r}_\gamma = \mathbf{r}(\mathbf{x}_\gamma)$ the residual corresponding to $\mathbf{x}_\gamma$. As indicated above, $\mathbf{x}_0$ and $\mathbf{x}_\gamma$ are not necessarily unique. However, convexity rules out possible existence of nonglobal local solutions. The set of all minimizers of F_γ is denoted by M_γ. When no confusion is possible we shall use the notation A_γ, $\mathbf{W}_\gamma$, etc., for $A_\gamma(\mathbf{x}_\gamma)$, $\mathbf{W}_\gamma(\mathbf{x}_\gamma)$, etc.

In the algorithm presented in § 3 F is minimized through minimizations of F_γ for a decreasing sequence of positive γ-values. For each new value of γ, information from the previous minimum of F_γ is utilized. The paper of Clark and Osborne [2] analyses the variation of a minimizer $\mathbf{x}_\gamma$ of F_γ as a function of γ. It is shown that $\mathbf{x}_\gamma$ is a piecewise linear function of γ. The proof is an easy consequence of the necessary condition (15) and of (8), which also provides formulae for following $\mathbf{x}_\gamma$ as γ varies: If $\mathbf{v}$ is a solution of

$$(17) \qquad (\mathbf{A}^T\mathbf{W}_\gamma\mathbf{A})\mathbf{v} = \mathbf{A}^T\mathbf{s}_\gamma,$$

where $\mathbf{W}_\gamma$ and $\mathbf{s}_\gamma$ are evaluated at $\mathbf{x}_\gamma$, then

$$(18) \qquad \mathbf{x}_{\gamma-\varepsilon} = \mathbf{x}_\gamma + \varepsilon\mathbf{v}$$

and

$$(19) \qquad \mathbf{r}(\mathbf{x}_{\gamma-\varepsilon}) = \mathbf{r}(\mathbf{x}_\gamma) + \varepsilon\mathbf{A}\mathbf{v}$$

are corresponding minimizers and residuals of $F_{\gamma-\varepsilon}$ provided $(\delta, \mathbf{x}_\delta, \mathbf{s}_\gamma)$ is a feasible reference for each intermediate δ, $\gamma - \varepsilon \leqq \delta \leqq \gamma$. This is utilized by Clark and Osborne to construct a method which follows the solution by updating the sign vector each time a hyperplane $\{\mathbf{x} \mid |\mathbf{r}_j(\mathbf{x})| = \delta\}$ is met.

We now list some useful properties of F_γ and its minimizers. Our main result, Theorem 1, shows that if $\gamma > 0$ is small enough then an $\mathbb{l}_1$ minimizer can easily be found from any minimizer $\mathbf{x}_\gamma$ of F_γ.

LEMMA 1. *If x_γ is a minimizer of F_γ, $\mathbf{W}_\gamma = \mathbf{W}_\gamma(\mathbf{x}_\gamma)$, and $\mathbf{s}_\gamma = \mathbf{s}_\gamma(\mathbf{x}_\gamma)$ then there always exists solution(s) to* (17).

Proof. From the necessary condition (15) it follows that (17) is equivalent to

$$(20) \qquad (\mathbf{A}^T\mathbf{W}_\gamma\mathbf{A})\mathbf{v} = -\frac{1}{\gamma}\mathbf{A}^T\mathbf{W}_\gamma\mathbf{r}(\mathbf{x}_\gamma).$$

This system is the normal equation for an overdetermined system of linear equations, and hence consistent.

LEMMA 2. *Let $\mathbf{x}_\gamma$ be a minimizer of F_γ. If $\mathbf{v}$ and $\mathbf{w}$ are solutions to* (17) *and $j \in A_\gamma(\mathbf{x}_\gamma)$, then*

$$r_j(\mathbf{x}_\gamma + \varepsilon\mathbf{v}) = r_j(\mathbf{x}_\gamma + \varepsilon\mathbf{w}) \quad \textit{for } \varepsilon \in R.$$

Proof. Since $(\mathbf{v} - \mathbf{w})$ is in the null space of $\mathbf{A}^T\mathbf{W}_\gamma\mathbf{A}$, we have, by the definition of $\mathbf{W}_\gamma$, that $\mathbf{a}_j^T(\mathbf{v} - \mathbf{w}) = 0$ for $j \in A_\gamma(\mathbf{x}_\gamma)$. Hence the result follows from the definition of r_j.

LEMMA 3. *If there exists a minimizer* $\mathbf{x}_\gamma \in M_\gamma$ *and* j, $1 \leqq j \leqq m$, *such that* $|r_j(\mathbf{x}_\gamma)| < \gamma$, *then* $r_j(\mathbf{x})$ *is constant for* $\mathbf{x} \in M_\gamma$.

Proof. Let $\mathbf{s} = \mathbf{s}(\mathbf{x}_\gamma)$. Then $Q_\mathbf{s}(\mathbf{y}) = F_\gamma(\mathbf{y})$ for $\mathbf{y} \in C_\mathbf{s}$. Let $\mathbf{y} \in C_\mathbf{s} \cap M_\gamma$; then $\mathbf{y}$ minimizes $Q_\mathbf{s}$. Hence it follows from (10) that

$$\mathbf{F}''_\gamma(\mathbf{x}_\gamma)(\mathbf{y}-\mathbf{x}_\gamma) = \mathbf{0}.$$

Then (9) implies $\mathbf{a}_j^T(\mathbf{y}-\mathbf{x}_\gamma) = 0$ since $j \in A_\gamma(\mathbf{x}_\gamma)$, and thus $r_j(\mathbf{y}) = r_j(\mathbf{x}_\gamma)$. Hence $r_j(\mathbf{y})$ is constant for $\mathbf{y} \in C_\mathbf{s} \cap M_\gamma$.

Let U be a neighbour subregion of $C_\mathbf{s}$, i.e., $\mathrm{cl}\,(U) \cap C_\mathbf{s} \neq \emptyset$. If $U \cap M_\gamma \neq \emptyset$ then there exist points $\mathbf{x} \in U \cap M_\gamma$ with $|r_j(\mathbf{x})| < \gamma$ because r_j is continuous and M_γ is a convex set. Hence $r_j(\mathbf{x})$ is constant in $U \cap M_\gamma$ because of the argument above, and hence the continuity of r_j implies that $r_j(\mathbf{x}) = r_j(\mathbf{x}_\gamma)$ for $\mathbf{x} \in U \cap M_\gamma$.

Repeating this argument, Lemma 3 follows because the set M_γ is connected.

Lemma 3 shows that the "small" solution residuals are easy to control when F_γ has several minimizers. The next lemma shows that the "large" solution residuals, although not being constant, remain "large" with constant sign. For easy notation we use the following alternative sign vector definition:

$$(21) \qquad \bar{s}_j(\mathbf{x}) = \begin{cases} -1 & \text{for} \quad r_j(\mathbf{x}) \leqq -\gamma, \\ 0 & \text{for} \quad |r_j(\mathbf{x})| < \gamma, \\ 1 & \text{for} \quad r_j(\mathbf{x}) \geqq \gamma. \end{cases}$$

$\bar{s}_\gamma(\mathbf{x})$ is the vector $(\bar{s}_1(\mathbf{x}), \dots, \bar{s}_m(\mathbf{x}))^T$.

LEMMA 4. $\bar{s}_\gamma(\mathbf{x})$ *is constant for* $\mathbf{x} \in M_\gamma$.

Proof. Let $\mathbf{x} \in M_\gamma$. If $|r_j(\mathbf{x})| < \gamma$ then $\bar{s}_j(\mathbf{y})$ is constant in M_γ because of Lemma 3. Next, assume $r_j(\mathbf{x}) \geqq \gamma$. If there exists $\mathbf{y} \in M_\gamma$ with $r_j(\mathbf{y}) < \gamma$ then there exists $\mathbf{z} \in M_\gamma$ with $|r_j(\mathbf{z})| < \gamma$ because of the convexity of M_γ and the continuity of r_j. But then we have a contradiction because of Lemma 3. Hence $r_j(\mathbf{y}) \geqq \gamma$ for $\mathbf{y} \in M_\gamma$, i.e., $\bar{s}_j(\mathbf{y}) = 1$ in M_γ. Finally, if $r_j(\mathbf{x}) \leqq -\gamma$ then the proof of $\bar{s}_j(\mathbf{y}) = -1$ for $\mathbf{y} \in M_\gamma$ is equivalent, and Lemma 4 is proved.

We denote by $\bar{\mathbf{s}}_\gamma$ the sign vector $\bar{\mathbf{s}}_\gamma(\mathbf{x}_\gamma)$, where $\mathbf{x}_\gamma \in M_\gamma$.

LEMMA 5. *Let* $0 < \delta \leqq \eta < \gamma$. *If* $\bar{s}_\delta = \bar{s}_\gamma$ *then* $\bar{s}_\eta = \bar{s}_\gamma$.

Proof. Let $\mathbf{x}_\delta \in M_\delta$, $\mathbf{x}_\gamma \in M_\gamma$, and

$$(22) \qquad \mathbf{x}_\eta = (1-\varepsilon)\mathbf{x}_\delta + \varepsilon \mathbf{x}_\gamma, \qquad \varepsilon = (\eta-\delta)/(\gamma-\delta).$$

Because of the linearity we have

$$(23) \qquad \mathbf{r}(\mathbf{x}_\eta) = (1-\varepsilon)\mathbf{r}(\mathbf{x}_\delta) + \varepsilon \mathbf{r}(\mathbf{x}_\gamma),$$

and hence since $0 \leqq \varepsilon \leqq 1$ and $\bar{\mathbf{s}}_\delta = \bar{\mathbf{s}}_\gamma$ we obtain $\bar{\mathbf{s}}_\eta(\mathbf{x}_\eta) = \bar{\mathbf{s}}_\delta$. Now the necessary condition (15) can be rewritten as follows:

$$(24) \qquad \mathbf{0} = \frac{1}{\gamma} \sum_{j \in \bar{A}_\gamma} r_j(\mathbf{x})\mathbf{a}_j + \sum_{j \notin \bar{A}_\gamma} \bar{s}_j(\mathbf{x})\mathbf{a}_j,$$

where $\bar{A}_\gamma \equiv \{i \mid 1 \leqq i \leqq m \wedge \bar{s}_i(\mathbf{x}) = 0\}$. Hence the sign vector identity, the optimality of $\mathbf{x}_\delta$ and $\mathbf{x}_\gamma$, and (24) imply that $\mathbf{x}_\eta$ satisfies the necessary condition (15) for F_η. Thus $\mathbf{x}_\eta$ is a minimizer of F_η since this function is convex. Now the result is a consequence of Lemma 4.

THEOREM 1. *Let $\mathbf{x}_\gamma$ be a minimizer of F_γ and let $\mathbf{v}_\gamma$ be a solution of* (17). *Then there exists $\gamma_0 > 0$ such that the following hold for $0 < \gamma < \gamma_0$:*

$$\text{(25)}\qquad \mathbf{x}_\delta = \mathbf{x}_\gamma + (\gamma - \delta)\mathbf{v}_\gamma \textit{ minimizes } F_\gamma \textit{ and } \mathbf{s}_\delta(\mathbf{x}_\delta) = \mathbf{s}_\gamma(\mathbf{x}_\gamma) \quad \textit{for } 0 < \delta \leqq \gamma,$$

$$\text{(26)}\qquad \frac{1}{\delta} r_j(\mathbf{x}_\delta) = -\mathbf{a}_j^T \mathbf{v}_\gamma \quad \textit{for } 0 < \delta \leqq \gamma \textit{ and } j \in A_0(\mathbf{x}_0),$$

$$\text{(27)}\qquad \mathbf{x}_0 = \mathbf{x}_\gamma + \gamma\, \mathbf{v}_\gamma \textit{ minimizes the } l_1 \textit{ function } F \textit{ and } A_\gamma(\mathbf{x}_\gamma) \subseteq A_0(\mathbf{x}_0).$$

Proof. It is a consequence of Lemma 5 that if $\bar{\mathbf{s}}_\delta \neq \bar{\mathbf{s}}_\gamma$, $0 < \delta < \gamma$, then $\bar{\mathbf{s}}_\theta \neq \bar{\mathbf{s}}_\gamma$ for all θ with $0 < \theta \leqq \delta$. Thus, since the number of different sign vectors is finite there must exist $\gamma_0 > 0$ such that $\bar{\mathbf{s}}_\delta$ is constant for $0 < \delta \leqq \gamma_0$.

Since $\bar{\mathbf{s}}_\delta$ is constant for $\delta \leqq \gamma_0$, $\mathbf{s}_\delta(\mathbf{x}_\delta)$ is constant for $\delta < \gamma_0$. This follows from the linearity in the definition (25) of $\mathbf{x}_\delta$ by inspecting the cases where $\bar{s}_j$ and s_j differ: If, for instance, $r_j(\mathbf{x}_{\gamma_0}) = \gamma_0 \wedge r_j(\mathbf{x}_\delta) > \delta$ for some $\delta < \gamma_0$ then the latter inequality holds for any $\delta < \gamma_0$. If $r_j(\mathbf{x}_{\gamma_0}) > \gamma_0$ and $r_j(\mathbf{x}_\delta) = \delta$ for some $\delta < \gamma_0$, then $r_j(\mathbf{x}_\theta) < \theta$ for $\theta < \delta$, which contradicts $\bar{\mathbf{s}}_\theta = \bar{\mathbf{s}}_\delta$. Thus, if $\bar{s}_j(\mathbf{x}_{\gamma_0}) = 1$ then $s_j(\mathbf{x}_\delta)$ is constant (0 or 1) for $\delta < \gamma_0$. If $\bar{s}_j = -1$ at γ_0 a similar argument applies. If $\bar{s}_j(\mathbf{x}_{\gamma_0}) = 0$ then $s_j(\mathbf{x}_\delta) = \bar{s}_j(\mathbf{x}_\delta) = 0$ for $\delta < \gamma_0$. Thus it is demonstrated that $s_\delta(\mathbf{x}_\delta)$ is constant for $\delta < \gamma_0$. Assume from now on that $\gamma < \gamma_0$. Then (25) is a consequence of (18).

Equation (26) is proved as follows: If $j \in A_0(\mathbf{x}_0)$, i.e., $r_j(\mathbf{x}_0) = 0$, then

$$\begin{aligned} \mathbf{r}_j(\mathbf{x}_\delta) &= \mathbf{a}_j^T \mathbf{x}_\delta - b_j = \mathbf{a}_j^T(\mathbf{x}_\gamma + (\gamma - \delta)\mathbf{v}_\gamma) - b_j \\ &= (\mathbf{a}_j^T \mathbf{x}_0 - b_j) - \delta \mathbf{a}_j^T \mathbf{v}_\gamma \\ &= -\delta \mathbf{a}_j^T \mathbf{v}_\gamma. \end{aligned}$$

Equation (27) is a consequence of the following: If $j \in A_\gamma(\mathbf{x}_\gamma)$ then $|r_j(\mathbf{x}_\delta)| \leqq \delta$ for $0 < \delta \leqq \gamma$, since $s_j(\mathbf{x}_\delta)$ is constant. Hence the continuity implies $r_j(\mathbf{x}_0) = 0$, i.e., $j \in A_0(\mathbf{x}_0)$. Thus $A_\gamma(\mathbf{x}_\gamma) \subseteq A_0(\mathbf{x}_0)$.

Since $\mathbf{x}_\delta$ minimizes F_δ we obtain from (15),

$$\text{(28)}\qquad \mathbf{0} = \sum_{j \in A_\delta} \frac{1}{\delta} r_j(\mathbf{x}_\delta)\mathbf{a}_j + \sum_{j \notin A_\delta} s_j(\mathbf{x}_\delta)\mathbf{a}_j.$$

Since $A_\delta(\mathbf{x}_\delta) \subseteq A_0(\mathbf{x}_0)$ this implies

$$\text{(29)}\qquad \mathbf{0} = \sum_{j \in A_0} \frac{1}{\delta} r_j(\mathbf{x}_\delta)\mathbf{a}_j + \sum_{j \notin A_0} s_j(\mathbf{x}_\delta)\mathbf{a}_j.$$

Because of the constant sign property (25), $s_j^*(\mathbf{x}_\delta) = s_j^*(\mathbf{x}_0)$ for $j \notin A_0$, and hence (29) is the necessary condition (16) for a minimizer of F. Therefore, $\mathbf{x}_0$ minimizes F because of the convexity. Thus Theorem 1 is proved.

Theorem 1 is the key to our algorithm since it shows that an l_1 solution can be detected directly from a minimizer of F_γ, $\gamma > 0$. Thus, we can avoid letting $\gamma \to 0$, which would give numerical instabilities. Furthermore, it is of course very easy, using (27) and inspecting the signs, to check whether the constant sign vector $\mathbf{s}_\delta$ has been found.

3. The algorithm. The new algorithm for minimizing the l_1 objective F is based on minimizing the smooth function F_γ for a set of decreasing values of γ. For every new value of γ information from the previous solution is utilized. Finally, when γ is small enough, an l_1 minimizer can be found from (27) of Theorem 1.

Thus our basic algorithm can be formulated as follows:

(30)

```
find an initial solution reference $(\gamma, \mathbf{x}_\gamma, \mathbf{s})$
**repeat**
    decrease $\gamma$
    find a solution reference $(\gamma, \mathbf{x}_\gamma, \mathbf{s})$
**until** $\gamma = 0$
{$\mathbf{x}_0$ is an $\ell_1$ minimizer}
```

The initial solution reference is found by letting $\mathbf{x}_\gamma$ be the least squares solution and then choosing γ and $\mathbf{s}$ appropriately.

In the strategy for decreasing γ we use (17)-(19). Let

$$\begin{aligned} \mathbf{x}(\delta) &= \mathbf{x}_\gamma + (\gamma - \delta)\mathbf{v}, \\ \mathbf{y}(\delta) &= \mathbf{r}(\mathbf{x}_\gamma) + (\gamma - \delta)A\mathbf{v}, \end{aligned} \qquad 0 \leqq \delta \leqq \gamma. \tag{31}$$

If $\mathbf{s}(\mathbf{x}(\delta)) = \mathbf{s}(\mathbf{x}_\gamma)$ for $0 < \delta \leqq \gamma$ then we let $\gamma = 0$ and $\mathbf{x}_\gamma = \mathbf{x}(0)$. Otherwise we choose a positive value of γ by inspecting some of the points where $\mathbf{y}(\delta)$ changes status, i.e., where $|y_j(\delta)| = \delta$ for some j, $1 \leqq j \leqq m$.

More precisely, let $\{\delta_i\}$, $i = 1, \ldots, N$, with $\gamma > \delta_1 > \delta_2 > \cdots > \delta_N > 0$ be the points in $]0, \gamma[$ where $|y_j(\delta_i)| = \delta_i$ for some value(s) of j. (If this set is empty then $N = 0$.) Let $\delta_{N+1} = 0$, let ν be the number of elements in $A_\gamma(\mathbf{x}_\gamma)$, and let ν_i be the number of elements in $\{j \mid |y_j(\delta_i)| \leqq \delta_i\}$. Then γ is chosen by the following procedure:

(32)

```
**if** $N = 0$ **then** $\gamma = 0$
**else**
    $i := 1$
    find $\delta_1$
    **while** not STOP **do**
        $i := i + 1$
        find $\delta_i$
    **end**
    $\gamma := \min[0.9 * \gamma, \frac{1}{2}(\delta_i + \delta_{i+1})]$
**end**
```

STOP is a function which returns true if one of the following conditions holds:

$$i = N, \quad \nu_{i+1} < \tfrac{1}{2}(\nu + n), \quad \nu_{i+1} \geqq \nu_i, \quad i > imax,$$

where *imax* is some fixed upper bound independent of n and m. (In our numerical experiments we have used *imax* = 20.)

The motivation for using the bound $\frac{1}{2}(\nu + n)$ in the stopping criterion is that normally $\nu > n$ when $N \neq 0$, and in nondegenerate cases there are n active residuals at the ℓ_1 solution [10]. So the philosophy is that when we have gone "half-way" from ν to n then we accept γ. A similar kind of argument motivates the condition $\nu_{i+1} \geqq \nu_i$. Note that this search guarantees that unless we have identified the ℓ_1 solution there is at least one change in the sign vector at the new value of γ. The reason for trying to choose γ outside of the set of kink points $\{\delta_i\}$ is that then the procedure for finding the new solution reference is more stable.

We have experimented with other strategies for reducing the threshold, e.g., the much simpler algorithm $\gamma := \frac{1}{2}\gamma$. The experiments show that it is inefficient to let the threshold decrease too quickly. The reason is that then the positive effect of the warm starts in the Newton iterations disappears and the whole iteration becomes less efficient. Although there was no evident difference between halving γ and (32) we chose the latter because it was the best on average.

The method for finding the new solution reference is a modified Newton iteration as given in [10] or [13]. The iteration is started from $\mathbf{z}=\mathbf{x}(\delta)$ as given by (31) with δ chosen as the new value of γ. The search direction $\mathbf{h}$ is normally found by minimizing Q_s where $\mathbf{s}=\mathbf{s}_\gamma(\mathbf{z})$. More precisely, we consider the equation

$$\mathbf{Q}_s''\mathbf{h}=-\mathbf{Q}_s'(\mathbf{z}). \tag{33}$$

If $\mathbf{Q}_s''$ has rank n then $\mathbf{h}$ is the solution to (33). Otherwise, if the system is consistent (i.e., the problem may be degenerate) then we use a *basic* solution to (33). Finally, if (33) is not consistent then we compute $\mathbf{h}$ by a Marquardt-like modification of the system. For details see §§ 4 and 5 in [10] or § 2 in [13]. The next iterate in the modified Newton iteration is found through a line search which is very cheap because of the simplicity of F_γ. It is shown in [10] that this iteration is finite, i.e., after a finite number of iterations we have $\mathbf{z}+\mathbf{h}\in C_{\mathbf{s}(\mathbf{z})}$ and thus $(\mathbf{z}+\mathbf{h})$ minimizes F_γ because of (10), (11), and the convexity of F_γ.

To summarize, the modified Newton iteration is the following:

```
        {there is given a reference ($\gamma$, $z$, s)}
        repeat
            find h from (33)
            if $(\mathbf{z}+\mathbf{h})\in C_s$ then
                $\mathbf{z}:=\mathbf{z}+\mathbf{h}$
(34)            stop := true
            else
                $\mathbf{z}:=\mathbf{z}+\alpha\mathbf{h}$      {line search}
                $\mathbf{s}:=\mathbf{s}_\gamma(\mathbf{z})$
        until stop
        {the new reference is ($\gamma$, z, s), i.e., $\mathbf{x}_\gamma=\mathbf{z}$}
```

The method of Clark and Osborne [2] follows a solution $\mathbf{x}_\gamma$ as γ varies, using (18) and updating the direction each time a change in the sign vector occurs. This is a strategy rather similar to using the simplex method of linear programming to solve the $\mathbb{l}_1$ problem. In the first version of our algorithm we used a combination of the method of [10] and the Clark-Osborne strategy, using the first method initially and the latter close to the solution. However, experiments showed that the simpler method (30)-(34) on the average is faster as well as more robust than the hybrid method. It enhances the numerical stability of our method that the values of γ used in (34) are chosen outside of the set of kink points. In the Clark-Osborne method directions are updated at the kink points, and this occasionally gives rise to numerical instabilities.

THEOREM 2. *The algorithm* (30)-(34) *stops at a minimizer* $\mathbf{x}_0$ *after a finite number of iterations.*

Proof. The number of loops in (30) must be finite as a consequence of Theorem 1 since γ is at least decreased by a factor of 0.9 in each loop. Thus Theorem 2 follows from the fact that the inner loop (34) is finite [10].

4. Numerical results. In this section we present results computed in Fortran 77 on an IBM PS/2 Model 55SX with an Intel 387 SX coprocessor and a Stardent Titan 1500 with a choice between scalar and vector mode computation. On the IBM PS/2 we have used the Lahey F77L version 4.00 compiler with production optimization, and on the Titan we have used release 2.2 of the Fortran compiler with "inlining" of subroutines. On both computers the machine accuracy is $\varepsilon_M=2^{-52}\approx 2.2_{10}-16$.

For details of our implementation of algorithm (30)-(34), see [13]. The major part of computing time is spent in solving the systems (33) and (17). We use the package AAFAC [12] for performing $\mathbf{LDL}^T$ factorization of the matrices $\mathbf{Q}_s'' = \mathbf{A}^T\mathbf{W}_\gamma(\mathbf{x})\mathbf{A}$, and in most of the iteration steps we only need simple down- and updatings of the factors **L** and **D**, corresponding to the equations leaving and entering the active set; in this case the cost of one iteration is $O(n^2)$. Occasionally a refactorization is needed (see § 2.2 in [13] and Tables 1 and 2 below); this is an $O(n^3)$ process.

In order to enhance accuracy we use one step of iterative refinement when solving a system of linear equations with matrix $\mathbf{Q}_s''$, and the $\mathbf{v}_\gamma$ of Theorem 1 is found by solving (20) rather than (17); cf. [13, §§ 2.3, 3.3].

The implementation makes intensive use of BLAS subroutines [5] for performing tasks like $\mathbf{y} := \alpha\mathbf{x}+\mathbf{y}$, $\mathbf{y} := \mathbf{A}\mathbf{x}+\beta\mathbf{y}$, etc. The computation of the **L** and **D** factors involves a series of updates of the form $(\mathbf{x}, \mathbf{y}) := (\mathbf{x}+\alpha\mathbf{y},\ \mathbf{y}+\beta\mathbf{x})$, where we use our own code [12]. For medium-sized problems ($m \approx 200$, $n \approx 100$) the BLAS routines and the computations of **L** and **D** each account for about 45% of the execution time on the IBM PS/2. Note that without "inlining" the calls of BLAS routines imply an overhead, but they lead to a simpler code, and on the Titan 1500 in vector mode we have used a vectorized version of the BLAS routines, thus helping to speed up computation.

First, consider the well-known stack loss data set; see, e.g., Table 5.1 in Osborne [14]. Our algorithm needs five different values of the threshold γ to obtain the ℓ_1 solution. Table 1 illustrates the five loops of the outer iteration (30). "iter." is the accumulated number of solutions of (33) or (17), and "refac." is the accumulated number of refactorizations in connection with computing the **L** and **D** factors.

TABLE 1
Results for the stack loss data set. $m = 21$, $n = 4$.

iter.	γ	refac.	$A_\gamma(\mathbf{x}_\gamma)$	$\mathbf{x}_\gamma$
1	7.238	1	$\{1, \ldots, 21\}$	(−39.92, .716, 1.295, −.152)
9	.226	4	$\{2, 8, 10, 12, 16, 18\}$	(−39.62, .833, .586, −.066)
11	.0232	4	$\{2, 8, 10, 16, 18\}$	(−39.81, .832, .574, −.060)
13	.0067	4	$\{2, 8, 16, 18\}$	(−39.73, .831, .576, −.061)
14	0	4	$\{2, 8, 16, 18\}$	(−39.69, .832, .574, −.061)

Note that three refactorizations are used during the Huber iteration to find $\mathbf{x}_{.226}$; they involve at most 6 out of the 21 equations. No further refactorizations are needed. Our results for ν and $\mathbf{x}_\gamma$ agree with Table 5.4 in Osborne [14].

In the other test problems the elements of **A** and **b** are computed by a random number generator, and modified so that condition (16) is satisfied for a given **y** and so that $A_0(\mathbf{y}) = \{1, \ldots, \nu_0\}$, with ν_0 given as input. This generator is similar to the generator for Huber problems in [10], and is based on ideas from [2] and [17]; details are given in [13, § 4.1].

In Fig. 1 we illustrate the typical behaviour of the algorithm: Each of the seven horizontal lines correspond to one loop in the outer iteration (30); the threshold values are shown in logarithmic scale. The circles indicate the current number of active equations ν. An "iteration" is counted as one solution of (33) or (17), corresponding to one loop in the inner iteration (34) or updating the threshold, respectively. The first iteration corresponds to finding the least squares solution; i.e., $\nu = m = 200$, which is outside of the figure.

Note that the first loop of (30) requires 14 loops in (34), and $\nu(\mathbf{x}_\gamma)$ is reduced from 200 to 120. In the remaining loops of (30) we have better approximations to the

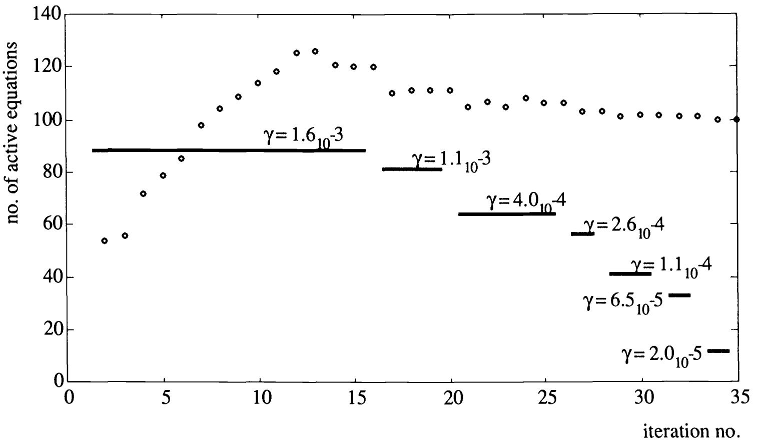


FIG. 1. *Typical behaviour of the algorithm.* $m = 200$, $n = \nu_0 = 100$. *Each horizontal line corresponds to one loop in* (30). *Each circle corresponds to one loop in* (34).

solution reference and therefore fewer loops in (34). For $\gamma = 2.0_{10}-5$ we find $\mathbf{s}(\mathbf{x}(\delta)) = \mathbf{s}(\mathbf{x}_\gamma)$ for $0 < \delta \leqq \gamma$ (cf. (31)), and the iteration stops.

In Table 2 we give results for a number of problems. For each set of (m, n, ν_0) we give average results for 10 different problems. "refac." and "iter." are explained above, and "i_0" is the number of threshold reductions, i.e., number of loops in (30). For comparison we also give results for the same problems when solved by the Harwell MA20AD implementation of the method of Barrodale and Roberts [1]. Here "iter." is the number of simplex iterations. For both methods the variation over the 10 problems is about 20%, and in all cases the solution is found with accuracy $O(\varepsilon_M)$.

In most of the problems the solution $\mathbf{x}_0$ found is unique and the number of elements in the active set $A_0(\mathbf{x}_0)$ is equal to n. These are all the problems with $\nu_0 = n$, and these we assume to be of greatest practical interest.

TABLE 2

Performance of Algorithm (30)-(34) *and* MA20AD. *Above line*: *Times on an* IBM PS/2 *model* 55SX; *Below line*: *Times on a Stardent Titan* 1500, *scalar mode.*

		Algorithm (30)-(34)					MA20AD	
m	n	ν_0	refac.	iter.	i_0	time (secs)	iter.	time (secs)
200	100	90	6.6	73.4	12.2	389	251	290
200	100	100	2.3	36.6	7.2	207	250	289
200	100	110	2.0	24.9	3.6	176	263	304
66	60	60	1.5	14.4	4.5	21	70	16
90	60	60	2.5	28.0	7.8	46	103	33
120	60	60	2.3	31.8	7.1	57	142	60
180	60	60	3.1	38.4	6.0	88	186	121
240	60	60	2.9	31.6	5.5	103	209	183
200	100	100	2.3	36.6	7.2	15	250	16
320	160	160	2.3	48.9	9.5	59	431	69
480	240	240	2.4	53.3	10.3	170	736	260
720	360	360	2.3	67.4	13.0	518	1211	946
1080	540	540	2.3	80.9	13.2	1606	2042	3553
1620	810	810	2.5	106.5	16.8	5319	3790	14700

In the first three sets of problems, however, we illustrate the effect of ν_0 being different from n. The first set of problems, for instance, is constructed such that there exist solutions $\mathbf{x}_0$ with *rank* less than n, i.e., the rank corresponding to the active set $A_0(\mathbf{x}_0)$ less than n. However, there also exist solutions with full rank in these cases since the matrix $\mathbf{A}$ has full rank. This means that the solutions are not unique in the first set of problems. The Barrodale-Roberts method seems to be almost unaffected by this, whereas our method is best when the solution is unique. The reason may be the following: if there are several solutions then our method will probably find a solution $\mathbf{x}_0$ where the rank corresponding to $A_0(\mathbf{x}_0)$ (and of $A_\gamma(\mathbf{x}_\gamma)$ for γ small) is less than n. This may slow down the rate of convergence in the inner iterations (34). The Barrodale-Roberts method, however, will find a solution at a "corner" of the simplex polytope, i.e., at a point where the rank is full. Therefore, the latter method may be unaffected by the fact that there exist solutions with rank less than n.

The next five sets of problems illustrate the dependence on

$$\mu = m/n. \tag{35}$$

Note that the numbers of refactorizations and of γ-reductions are almost constant, and that the required number of iterations seems to reach a maximum for $\mu \approx 3$. For the Barrodale-Roberts method the number of iterations grows monotonically with m.

In [13] we give more examples, and based on these experiments we have found that for both methods the number of iterations is roughly modeled by

$$\text{number of iterations} \approx A(\mu) \cdot n^{\alpha}, \tag{36}$$

where $\alpha = 0.5$ for our method and $\alpha = 1.25$ for the Barrodale-Roberts method. In Fig. 2 (double logarithmic scale) this relation is illustrated by a number of problems with $\mu = 2$, $\nu_0 = n$. The lines are fitted in two groups: the large problems are the last five sets of problems in Table 2, while the smaller values of n include the second set of problems in Table 2 and some smaller problems. There is good agreement with model (36). The lines correspond to $a(2) \approx 4$ for our method and $a(2) \approx 0.8$ for the Barrodale-Roberts method.

Now we can estimate execution times: For a typical iteration the time is $O(n^2)$ for both methods, but the occasional (although very few) refactorizations imply that for our method (denoted *T.m.*)

$$\text{time}_{\text{T.m.}} \approx b(\mu)n^3 + c(\mu)n^{2.5} \tag{37}$$

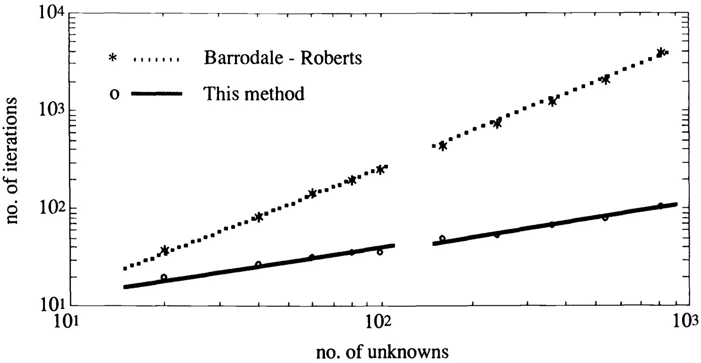


FIG. 2. *Number of iterations as functions of n. $m = 2n$, $\nu_0 = n$.*

should be a fairly good model, while

$$\text{time}_{\text{B-R}} \approx d(\mu) n^{3.25}. \tag{38}$$

For the last five sets of problems in Table 2 a least squares fit with these models gives $(b(2), c(2), d(2)) \approx (8.9, 32, 5.2) \cdot 10^{-6}$ seconds.

We have also solved the last five sets of problems in Table 2 using vector mode on the Titan 1500. In Table 3 we give results for run times of our method, the *speed-up* (i.e., ratio between times in scalar and vector mode), and the ratio between vector run times for the two methods.

TABLE 3
Times and speed-up (g). $m = 2n$. *Vector mode on a Stardent Titan* 1500.

n	$\text{time}_{\text{T.m.}}$ (secs)	$g_{\text{T.m.}}$	g_{B-R}	$\text{time}_{B-R}/\text{time}_{\text{T.m.}}$
160	8.6	6.8	1.45	5.5
240	21.7	7.8	1.45	8.2
360	59.2	8.8	1.45	11.0
540	172.4	9.3	1.45	14.2
810	530.2	10.0	1.45	19.1

For comparison it should be mentioned that on the Titan 1500 the theoretical limit for speed-up is $g = 16$, but $g \leqq 11$ is found in practice. We get very close to this limit, whereas the MA20AD is seen to vectorize very poorly.

5. Conclusion. We have defined a new algorithm for solving the linear $\mathbb{l}_1$ problem. The algorithm is efficient and compares favorably with well-known methods. For full matrix problems the new method seems to be better than Simplex-type methods by a factor $O(n^{0.25})$.

The method is very well suited for sparse matrix techniques, so future work will extend the method to the sparse case. A sparse version is interesting, especially because the relationship between the $\mathbb{l}_1$ problem and the linear programming problem indicates that our method may also be used to solve the latter class of problems. Introductory experiments in this respect are encouraging.

REFERENCES

[1] I. BARRODALE AND F. D. K. ROBERTS, *An improved algorithm for discrete $\mathbb{l}_1$ linear approximation*, SIAM J. Numer. Anal., 10 (1973), pp. 839-848.

[2] D. I. CLARK AND M. R. OSBORNE, *Finite algorithms for Huber's M-estimator*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 72-85.

[3] B. CHEN AND P. T. HARKER, *A non-interior-point continuation method for linear complementarity problems*, Working Paper 90-10-03, Decision Sciences Dept., Univ. of Pennsylvania, Philadelphia, PA, 1990.

[4] T. F. COLEMAN AND Y. LI, *A global and quadratic affine scaling method for linear $\mathbb{l}_1$ problems*, Report TR 89-1026, Dept. of Computer Science, Cornell Univ., Ithaca, NY, 1989.

[5] J. DONGARRA, J. DU CROZ, S. HAMARLING, AND R. HANSEN, *An extended set of Fortran basic linear algebra subprograms*, ACM Trans. Math. Software, 14 (1988), pp. 1-17.

[6] *Harwell Subroutine Library*, Report R 9185, 9th ed., Computer Science and Systems Division, Harwell Laboratory, England, 1989.

[7] P. HUBER, *Robust Statistics*, John Wiley, New York, 1981.

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

[9] O. KNOTH, *Newton-like methods for the Euclidean multifacility location problem*, presented at the 14th Internat. Symposium on Mathematical Programming, Amsterdam, 1991.

[10] K. MADSEN AND H. B. NIELSEN, *Finite algorithms for robust linear regression*, BIT, 30 (1990), pp. 682–699.

[11] R. E. MARSTEN, *User's manual for the research version of OB*1, Georgia Institute of Technology, Atlanta, GA, 1989.

[12] H. B. NIELSEN, *AAFAC, a package of Fortran 77 subprograms for solving* $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{c}$, Report NI 90-01, Institute for Numerical Analysis, Technical Univ. of Denmark, Lyngby, 1990.

[13] ———, *Implementation of a finite algorithm for linear* ℓ_1 *estimation*, Report NI 91-01, Institute for Numerical Analysis, Technical Univ. of Denmark, Lyngby, 1991.

[14] M. R. OSBORNE, *Finite Algorithms in Optimization and Data Analysis*, John Wiley, New York, 1985.

[15] M. C. PINAR AND S. A. ZENIOS, *On smoothing exact penalty functions for convex constrained optimization*, Report 91-05-03, Decision Sciences Dept., Univ. of Pennsylvania, Philadelphia, PA, 1991.

[16] S. A. RUZINSKY AND E. T. OLSEN, L_1 *and* L_∞ *minimization via a variant of Karmarkar's algorithm*, IEEE Trans. Acoust. Speech Signal Process., 37 (1989), pp. 245–253.

[17] D. F. SHANNO AND D. M. ROCKE, *Numerical methods for robust regression: Linear models*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 86–97.

[18] G. A. WATSON, *Approximation Theory and Numerical Methods*, John Wiley, New York, 1980.

SIAM J. OPTIMIZATION
Vol. 3, No. 2, pp. 236-247, May 1993

002

UNIFORMLY EXTREMAL SOLUTIONS IN SOBOLEV FUNCTION SPACES FOR THE QUADRATIC CASE: CHARACTERIZATION AND APPLICATIONS*

LAKSHMAN S. THAKUR†

Abstract. Important in optimization issues in many areas, the uniformly extremal solutions in a real Sobolev space which minimize $f^{(n)}$ in L_∞ norm and interpolate the given data $\{(x_i, y_i)\}_1^p$ are characterized for the quadratic case $n=2$. In contrast to comparable results, the characterization uses only elementary facts and adds some useful perspectives on extremal solutions. Its consideration of uniformly extreme splines suggests a simple and fast method for computing optimal solutions; problems with $p=200$ points can be solved in less than two seconds of cpu time. It also leads, in another application, to an elementary proof of Karlin's characterization theorem, which has so far relied on a wide range of advanced mathematical tools. Thus, this analysis of uniformly extremal solutions is fruitful for the quadratic case and offers a promising framework for generalizing the elementary proof and efficient solution approach for the higher-degree and other related problems.

Key words. optimal quadratic splines, uniformly extremal solutions in Sobolev spaces, Karlin's perfect spline theorem

AMS subject classifications. primary 90C20; secondary 41A15, 41A05, 46E35

1. Introduction. Often we need to infer about an interesting property of the underlying function from its values at a certain number of points. For example, what is the minimal value of $\|f^{(1)}\|_\infty$ or $\|f^{(2)}\|_\infty, \ldots,$ required by a function f to be able to interpolate the given points? It is obvious that this depends critically on the property of interest and the assumptions one can reasonably make in the context. For many problems, the most natural and important of these variables relate to the continuity, differentiability, and boundedness of the underlying function and its derivatives. Here we study a problem of this kind that has been discussed in the literature for a long time. We consider functions f, which pass through the p given points $\{(x_i, y_i)\}_1^p$ on the plane, whose $(n-1)$th derivatives are absolutely continuous, and whose bounded nth derivatives exist almost everywhere in $[x_1, x_p]$. Then we ask: What is the minimal value of $\|f^{(n)}\|_\infty$, and which f, among these, achieves the minimum?

More precisely, let $F_\infty^{(n)}[a, b]$, $-\infty < a < b < \infty$, be a subset of the real Sobolev space

$$W_\infty^{(n)}[a, b] = \{f: f \in C^{n-1}[a, b]; f^{(n-1)} \text{ abs. cont.}; f^{(n)} \in L_\infty[a, b]\},$$

defined by

$$F_\infty^{(n)}[a, b] = \{f: f \in W_\infty^{(n)}[a, b]; f(x_i) = y_i, i = 1, \ldots, p\},$$

where $\{y_i\}_1^p \in R^p$, $\{x_i\}_1^p \in R^p$ with $p \geqq 2$, $x_i < x_{i+n}$, $x_1 = a$, $x_p = b$, and all $x_i \in [a, b]$ are given problem data. Note that absolute continuity of $f^{(n-1)}$ simply implies that it can be obtained by integration of $f^{(n)}$, and $L_\infty[a, b]$ denotes the set of essentially bounded functions in $[a, b]$, that is, functions bounded everywhere in $[a, b]$, except, at most, at a set of points with measure zero. Then we define the *nth degree* minimization problem as

$$\inf\{\|f^{(n)}\|_\infty : f \in F_\infty^{(n)}[a, b]\}. \tag{1}$$

* Received by the editors January 28, 1991; accepted for publication (in revised form) March 24, 1992.

† Operations and Information Management Department, University of Connecticut, Storrs, Connecticut 06269-2041. This work was partially supported by the Frehn Fellowship of Shippensburg University, Pennsylvania.

Applications of such problems are in various fields, including operations research, statistics, control theory, and numerical analysis [10], [11], [14]-[16], [20], [21].

Probably due to the difficulty of the general problem, many special cases of (1) have been considered: Glaeser [7], Louboutin [12], Schoenberg [14], and others [8], [17], [18]. Here we introduce and give a characterization of uniformly extremal solutions for the quadratic case $n=2$ with increasing x_i. The motivation for doing so is manifold:

(i) Since the convexity (concavity) of a function f is determined by $f^{(2)}$, and since bounds on $f^{(2)}$ can be used to compute error bounds in often used approximations of nonlinear functions, the quadratic case is important. For example, we get a bound [15] on the function error: $\max_{a\leqq x\leqq b}|f^*(x)-\hat{f}(x)|\leqq\|f^{*(2)}\|_\infty\delta^2/8$, where $\hat{f}$, obtained from joining the adjacent points $\{(x_i, f(x_i))\}_1^p$, is the piecewise linear approximation of some $f\in F_\infty^{(2)}[a, b]$, f^* is a solution of (1) for $n=2$, and $\delta=\max_{1\leqq i\leqq p-1}(x_{i+1}-x_i)$. Since f^* is a solution, and therefore $\|f^{*(2)}\|_\infty\leqq\|f^{(2)}\|_\infty$ for all $f\in F_\infty^{(2)}[a, b]$, such a bound can be used in the error analysis of convex separable programs [15], [16].

(ii) The constructive nature of the arguments used to study uniformly extreme splines gives insight into the mechanism underlying the characterization theorem, in terms of both the perfectness of the solution, and the number of knots leading to a fast method of computing optimal solutions.

(iii) The resulting characterization depends only on elementary facts. This is in contrast to the current approaches to the general problem (as reflected in Karlin's characterization theorem discussed below) which require a substantial range of nonelementary mathematical tools [1], [3], [4], [9], [11], [13].

THEOREM (Karlin [10]). *There is a solution f of* (1) *which is a perfect spline of degree n with at most $(p-n-1)$ knots in $[a, b]$; that is, f maintains a constant absolute value of the nth derivative with a sign change at each of the at most $(p-n-1)$ knots in $[a, b]$.*

Karlin's [11] method to prove the theorem is based on deep analysis. It uses many total positivity properties of the kernel $(x-t)_+^n$ and advanced topological results on the degree of mapping of nonlinear transformations. The *existence* of the solution of (1) is also shown by Jerome [9] and Fisher and Jerome [3]. This approach uses representations for $f\in W_\infty^{(n)}[a, b]$ obtained from Peano's theorem, and shows that any nonempty intersection of $\{f^{(n)}: f\in F_\infty^{(n)}[a, b]\}$ with a closed ball in $L_\infty[a, b]$ is sequentially weak* closed in $L_\infty[a, b]$. Their other result [4], which *characterizes* solutions of (1), is accessible via calculus, but it uses their above existence result and further relies on such results as open mapping and Arzela-Ascoli theorems. In contrast to these works, DeBoor [1] has given a short proof of Karlin's theorem. However, in terms of tools, a representational theorem for the divided difference linear functional, the Riesz duality theorem, the Hahn-Banach theorem, Holder's inequality, the smoothing of functionals using Gaussian kernels, and finally a limiting process, are used in the proof.

In this paper we show that for our special case, consideration of uniformly extremal spline solutions leads to a characterization and a solution method based on the consequences of the following elementary observation. For a continuous function $f(x)$, $x\in[a, b]$, with a fixed mean value ν, its slope essentially bounded by a constant m, and the value $f(a)$ constrained in $[l, h]$, the maximum (minimum) attainable value $f(b)$ is realized by (i) f starting at a with the lowest (highest) possible value $l(h)$, (ii) having $-m(m)$ slope initially, and then (iii) switching to $m(-m)$ slope at an appropriate point in $[a, b]$ to have its mean value equal to ν. Note that this construction (i) is optimal in attaining the maximum (minimum) $f(b)$, (ii) has at most one knot in $[a, b]$, and (iii) is a perfect spline. Thus, we would expect it to be related to optimal solutions of the problem and to such a result as Karlin's theorem. Here we show that it actually leads to an efficient computational method and to an elementary characterization that adds up to the full statement of Karlin's theorem.

In the next section we consider a related problem and study some basic properties of its solutions. These properties are then used to motivate our characterization result in § 3.1, and show how its consideration of uniformly extremal solutions suggests a fast approach to computing optimal solutions of the problem in § 3.2. Section 3.3 shows how it leads to an elementary proof of Karlin's theorem. The summary and concluding remarks follow in § 4.

2. Preliminary analysis. Our case is $n=2$, with $x_i<x_{i+1}$, $i=1,\dots,p-1$. Thus, specifically, we want to characterize and compute extremal solutions of

$$\inf\{\|f^{(2)}\|_\infty : f\in F_\infty^{(2)}[a,b]\}, \tag{2}$$

and show, in terms of Karlin's theorem, that it has a solution which is a perfect spline of second degree with at most $(p-3)$ knots, for $p\geqq 3$. For $p=2$ the solution is trivial and has no knots.

2.1. A related problem. In the following analysis, we work with an easier problem, $(\hat{2})$, instead of (2). For degree n and p given points $\{x_i\}_1^p$, $\{y_i\}_1^p\in R^p$, with $a=x_1<\cdots<x_p=b$, consider problem $(\hat{2})$:

$$\inf\{\|f^{(1)}\|_\infty : f\in \hat{F}^{(1)}[a,b]\}, \tag{$\hat{2}$}$$

where

$$\hat{F}(n)[a,b]=\left\{f(x): f(x)\in W_\infty^{(n)}[a,b];\ \int_{x_i}^{x_{i+1}} f(x)\,dx=(y_{i+1}-y_i)\right\} \tag{3}$$

$$\text{for } i=1,\dots,p-1.$$

Note that $f^*(x)$, a solution of (2), is given by $f^*(x)=\int_{x_1}^x \hat{f}^*(x)\,dx+y_1$, where $\hat{f}^*(x)$ is a solution of $(\hat{2})$, with knots of f^* and $\hat{f}^*$ obviously the same in number and at identical x-coordinates. Finite-dimensional mathematical programming techniques have been used recently [17], [18] to solve problem $(\hat{2})$. Illiev and Pollul [8] also use $(\hat{2})$ for *convex* quadratic splines; however, as compared to Karlin's theorem adapted for convex splines, their characterization result is significantly weaker regarding the more difficult "perfectness" part [11, p. 27].

2.2. Notation and definitions. We use the following notation and definitions.

(a) For any positive integer i, $k\geqq 0$, $x_i<x_{i+1}$, and the given problem data, let $d_i=(y_{i+1}-y_i)/(x_{i+1}-x_i)=\Delta y_i/\Delta x_i$, $k_i=k(x_{i+1}-x_i)/2$, and $H_i=d_i+k(x_{i+1}-x_i)/2=d_i+k_i$, $L_i=d_i-k(x_{i+1}-x_i)/2=d_i-k_i$. Note that for $k>0$, $H_i>L_i$.

(b) For given real constants a, b, c, let $h[a,b,c]$ define a real affine funion on R, passing through the point (a,b) and having slope c everywhere. Thus $h[a,b,c](a)=b$, $h^{(1)}[a,b,c](x)=c$ for all $x\in R$.

(c) For a positive integer i, $x_i<x_{i+1}$ and real s, $k\geqq 0$; y_i^+, y_i^-; x_i^+, x_i^- both in the interval $[x_i, x_{i+1}]$, we define a pair of continuous piecewise affine functions on a single interval $[x_i, x_{i+1}]$:

$$\begin{aligned} g_i^+[s,k]&=\begin{cases} h[x_i,s,k](x), & x_i\leqq x\leqq x_i^+,\\ h[x_i^+,y_i^+,-k](x), & x_i^+\leqq x\leqq x_{i+1}; \end{cases}\\ g_i^-[s,k]&=\begin{cases} h[x_i,s,-k](x), & x_i\leqq x\leqq x_i^-,\\ h[x_i^-,y_i^-,k](x), & x_i^-\leqq x\leqq x_{i+1}, \end{cases} \end{aligned} \tag{4}$$

where $h[x_i,s,k](x_i^+)=y_i^+$ and $h[x_i,s,-k](x_i^-)=y_i^-$, ruling out any jumps in $g_i^+[s,k]$, $g_i^-[s,k]$ at x_i^+, x_i^-, respectively (see Fig. 1).

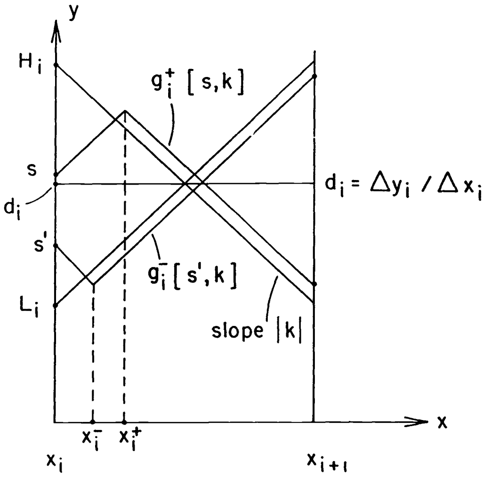


FIG. 1. *The g_i^+ and g_i^- functions.*

Since $[s, k]$ and $[x_i, s, \pm k]$ are frequently used, except when given for emphasis, we drop them (others will be given) and denote the corresponding functions by g_i^+, g_i^-, h^+, and h^-. For referring to both g_i^+ and g_i^-, we will use g_i. The values x_i^+, y_i^+, x_i^-, and y_i^- are determined by certain conditions to be satisfied by g_i's. Note that g_i^+ (g_i^-) is a perfect spline of order 1 on $[x_i, x_{i+1}]$ with at most one knot.

When $x_i^+ = x_{i+1}$, $x_i^- = x_i$, we have $g_i^+ = g_i^- = h^+$ for all $x \in [x_i, x_{i+1}]$, and we will denote this function by g_i^{++}. Similarly, when $g_i^+ = g_i^- = h^-$, it will be written g_i^{--}. These functions are needed for their extremal properties, explored below.

(d) For the given real values $s, s' < s, k \geqq 0$, positive integers $i, j (j \geqq i+1)$, $p := j - i + 1$, and a p-point problem data $\{x_r\}_i^j$, $\{y_r\}_i^j$, consider, in relation to problem $(\hat{2})$, the following sets of functions defined in $[x_i, x_j]$:

$$E_{ij}(k) = \left\{ f \in W_\infty^{(1)}[x_i, x_j] \colon \|f^{(1)}\|_\infty \leqq k;\ \int_{x_r}^{x_{r+1}} f(x)\, dx = y_{r+1} - y_r,\ r = i, \ldots, j-1 \right\},$$

$$E_{ij}(s, k) = \{ f \in E_{ij}(k) \colon f(x_i) = s \} \quad \text{and} \quad E_{ij}(s', s, k) = \{ f \in E_{ij}(t, k) \colon t \in [s', s] \}.$$

For a function in any of the above sets, it is often convenient to call $f(x_i)$ the *starting* value and $f(x_j)$ the *ending* value, and to say that f has *mean values* $d_i, \ldots, d_{j-1}$, *f covers* $d_i, \ldots, d_{j-1}$, or $d_i, \ldots, d_{j-1}$ are *coverable* by f (we may, to emphasize the value of k being used, add "with value k" or "with k") in $[x_i, x_{i+1}], \ldots, [x_{j-1}, x_j]$, or, briefly, that *$f$ covers* $[x_i, x_j]$. Thus $f \in E_{ij}(k)$ if it covers $[x_i, x_j]$ with value k; $f \in E_{ij}(s, k)$ if, in addition, it starts at s, and $f \in E_{ij}(s', s, k)$ if it starts somewhere in $[s', s]$.

Let $k = \min\{\|f^{(1)}\|_\infty \colon f \in \hat{F}_\infty^{(1)}[x_i, x_j]\}$; then we will denote the set of solutions of this p-point problem by $S_{ij}[k]$. Note that with this value of k, $E_{ij}(k) = S_{ij}[k]$. The solutions that satisfy the conditions in Karlin's theorem (for the corresponding problem (2)) will be called *Karlin solutions* of $(\hat{2})$. For a p-point problem the maximum number of knots a Karlin solution may have is $(p-3)$; it is called the *Karlin count* for the problem. From § 2.1 we know that a Karlin solution of $(\hat{2})$ gives us a Karlin solution of (2) with the same number of knots.

For these sets of functions when they are nonempty, and for $i \leqq r \leqq j$, we would need the supremum and infimum values, defined, for example, for the set $E_{ij}(k)$ by $h_{ij}^r(k)(l_{ij}^r(k)) = \sup(\inf)\{f(x_r) \colon f \in E_{ij}(k)\}$. Similar definitions apply for the other sets.

Except when given for emphasis, we will drop the second subscript j, when $j = i+1$.

A solution $G_{ij} \in S_{ij}[k]$, starting off with $G_{ij}(x_i) = \sup\{f(x_i): f \in S_{ij}[k]\}$, $G_{ij}(x_{i+1}) = \inf\{f(x_{i+1}): f \in S_{ij}[k]\}, \dots$, is called the (upper) *uniformly extreme solution* in $S_{ij}[k]$. Similarly, we can define $G'_{ij} \in S_{ij}[k]$ as the (lower) uniformly extreme solution which starts off at x_i with the *minimum* attainable value and achieves minimum and maximum attainable values (among all $f \in S_{ij}[k]$) at alternating points $x_i, \dots, x_j$. We call G_{ij}, G'_{ij} (though they may not always be distinct) the *uniformly extreme pair* in $S_{ij}[k]$ for the p-point problem $(\hat{2})$, $G_{ij}(x_i)$ the *extremal starting value at* x_i, $G_{ij}(x_{i+1}), \dots, G_{ij}(x_{j-1})$ the *extremal values* at $x_{i+1}, \dots, x_{j-1}$, respectively, and $G_{ij}(x_j)$ the *extremal ending value* at x_j. Similar terminology applies to G'_{ij} and uniformly extreme solutions in $E_{ij}(k)$, $E_{ij}(s, k)$ and $E_{ij}(s', s, k)$.

2.3. The underlying elementary properties. Using the above notation, we now give some basic results, which will be used to prove the main theorem in the next section. We begin by summarizing the elementary properties of functions g_i's and functions in $E_i(s, k)$, $E_i(s', s, k)$ in Proposition 1. The proofs of these basic facts require some details, but they are elementary and are given in the Appendix.

PROPOSITION 1. *For a positive integer i, $k \geqq 0$, and a two-point problem data (x_i, y_i), (x_{i+1}, y_{i+1}), $x_i < x_{i+1}$, we have*

(A) (i) *For $s = Hi$, $E_i(s, k) = \{g_i^{--}\}$.* (ii) *For $s = L_i$, $E_i(s, k) = \{g_i^{++}\}$.* (iii) *For $L_i < s < H_i$, there are unique real numbers $x_i^- \in (x_i, x_{i+1})$, $x_i^+ \in (x_i, x_{i+1})$, y_i^-, y_i^+; giving functions g_i^-, g_i^+, which are in $E_i(s, k)$, with $g_i^-(x_{i+1}) > g_i^+(x_{i+1})$; left-sided derivatives $g_i^{-(1)}(x_{i+1}) = k$, $g_i^{+(1)}(x_{i+1}) = -k$; and one knot in (x_i, x_{i+1}).* (iv) *$E_i(s, k)$ is nonempty if and only if $L_i \leqq s \leqq H_i$.*

(B) *Let $L_i \leqq s \leqq H_i$, g_i^+, $g_i^- \in E_i(s, k)$; then $g_i^+(x_{i+1}) \leqq f(x_{i+1}) \leqq g_i^-(x_{i+1})$ for any $f \in E_i(s, k)$.*

(C) *Let $L_i \leqq s' < s \leqq H_i$; then for any $t \in [s', s]$ there are unique $g_i^-[t, k] \in E_i(s', s, k)$ and $g_i^+[t, k] \in E_i(s', s, k)$.*

(D) *Let $L_i \leqq s' < s \leqq H_i$. Let $G: [s', s] \to R$, $G': [s', s] \to R$, be defined by $G(t) = g_i^-[t, k](x_{i+1})$ and $G'(t) = g_i^+[t, k](x_{i+1})$ for a fixed x_{i+1}, where $g_i^-[t, k]$ and $g_i^+[t, k]$ are both in $E_i(s', s, k)$ (note that by* (C) *above, $g_i^-[t, k]$ and $g_i^+[t, k]$ exist). Then* (i) *$G(t)$, $G'(t)$ are decreasing functions of t, $t \in [s', s]$.* (ii) *$G(t)$, $G'(t)$ are continuous functions of t, $t \in [s', s]$.* (iii) *The functions $g_i^-[s', k] \in E_i(s', k)$ and $g_i^+ \in E_i(s, k)$ have the following properties:*[1]

$$g_i^-[s', k](x_{i+1}) = \sup\{f(x_{i+1}): f \in E_i(s', s, k)\};$$

$$g_i^+(x_{i+1}) = \inf\{f(x_{i+1}): f \in E_i(s', s, k)\}.$$

(iv) *If $k' < k$, $t \in [s', s]$, $E_i(t, k') \neq \phi$, then $g_i^-[t, k'](x_{i+1}) < g_i^-[t, k](x_{i+1})$ and $g_i^+[t, k'](x_{i+1}) > g_i^+[t, k](x_{i+1})$.*

Note that Proposition 1 is stated in terms of the starting value s at x_i. However, just as a given *starting* value at x_i and a value of k determines g_i^- and g_i^+ functions uniquely, a given *ending* value s at x_{i+1} and the value of k also determine them uniquely. Hence all the *symmetrical* results in Proposition 1 can be similarly proved in terms of an ending value at x_{i+1}.

The above proposition describes the properties of an $(i=2)$-point problem. In Proposition 2 we discuss three others, needed for $i \geqq 2$.

[1] The often used arguments $[s, k]$ in $g_i^{\mp}$ functions and $[x_i, s, \pm k]$ in $h^{\pm}$ functions are not written (except for emphasis), hence $g_i^+ \equiv g_i^+[s, k]$, $h^- \equiv h[x_i, s, -k]$; see § 2.2(c).

PROPOSITION 2. *For the given i-point* $(i \geqq 2)$ *problem data* $\{x_r\}_1^i$, $\{y_r\}_1^i$, *and k with* $E_{1i}(k) \neq \phi$, *we have the following*:

(A) *Let there be some* $k' < k$ *with* $E_{1i}(k') \neq \phi$. *Let q be a uniformly extreme perfect spline* (UEPS) *in* $E_{1i}(k)$. (i) (a) *If q attains the lowest extremal value at the* last *data value* x_i: $q(x_i) = l_{1i}^i(k)$, *then the left-sided derivative at* x_i: $q^{(1)}(x_i) = -k$. (b) *If q attains the highest extremal value at the* last *data value* x_i:$q(x_i) = h_{1i}^i(k)$, *then the left-sided derivative* $q^{(1)}(x_i) = k$. (ii) (a) *If q attains the lowest extremal value at the* first *data value* x_1: $q(x_1) = l_{1i}^1(k)$, *then the right-sided derivative at* x_1: $q^{(1)}(x_1) = k$. (b) *If q attains the highest extremal value at the* first *data value* x_1: $q(x_1) = h_{1i}^1(k)$, *then the right-sided derivative* $q^{(1)}(x_1) = -k$.

(B) *Let* $q_1 \neq q_2$ *be* UEPS*s in* $E_{1i}(s', s, k) \neq \phi$ (*i even, so that the pattern—low at* x_i, *high at* $x_{i-1}, \ldots,$*—ends up high at* x_1) *such that the* lowest *ending value* $q_2(x_i) > q_1(x_i)$; *then we must have the* highest *starting value* $q_2(x_1) \leqq q_1(x_1)$. *That is, if the lowest ending value of a* UEPS *is not as low as that of another* UEPS, *its highest starting value cannot be higher than that of the other* UEPS. *Analogous statements also apply to the other highest* (*lowest*) *ending values and i odd* (*even*) *cases.*

(C) *Let q be a* UEPS *for k and the* $(i-1)$*-point problem defined by the first* $(i-1)$ *points of the given data*; *i.e.*, UEPS $q \in E_{1i-1}(k)$. *Let* $q(x_{i-1})$ *be the highest* (*lowest*) *attainable value* $h_{1i-1}^{i-1}(k)(l_{1i-1}^{i-1}(k))$ *at* x_{i-1}. *Let the last point* (x_i, y_i) *of the data be such that* $h_{1i-1}^{i-1}(k) = L_{i-1}$ $(l_{1i-1}^{i-1}(k) = H_{i-1})$; *then* $E_{1i}(k') = \phi$ *for any* $k' < k$; *i.e., k is the* minimal *value needed for the i-point problem.*

3. Main result and applications. Considering uniformly extremal splines, we discuss and prove the main theorem in § 3.1. Then follow the applications to computing optimal solutions of the problem and deriving an elementary proof of Karlin's characterization theorem.

3.1. Characterizing uniformly extremal solutions.

3.1.1. Outline of the proof. The basic idea of the proof of the main characterization theorem, given below, is as follows. Suppose that we have a UEPS q for an i-point problem. Let us incorporate an additional data point $\{x_{i+1}, y_{i+1}\}$. If the *new* data point is, say, *dominated* by the previous data $\{x_r, y_r\}_1^i$ (that is, the extremal value under consideration at x_i as determined by the previous data remains unaffected by the new data point, i.e., remains the same for the $(i+1)$-point problem), then q can be extended to the *last* (new) interval with exactly one additional knot to serve as the UEPS for all the data, that is, for the $(i+1)$-point problem. However, if the new data point *dominates* the previous data (the extremal value at x_i is changed by the new data point), we show that the *first* data point is *dominated* by the remaining i-point data $\{x_r, y_r\}_2^{i+1}$ with respect to the corresponding extremal value *at* x_2. Therefore, as in the previous case, we can use the UEPS for the data $\{x_r, y_r\}_2^{i+1}$ and extend it to the *first* interval with exactly one additional knot to be the UEPS for the $(i+1)$-point problem. Here we see a perfect symmetry of dominance between the first and the last single data points, or equivalently, between the first and last group of i-data points in the given $(i+1)$-data points. In the proof below, the new data point is dominated by the previous data in case (a), while it dominates the previous data in case (b).

THEOREM 1. *Let* k^* *be the minimal value* $k^* = \min\{\|f^{(1)}\|_\infty : f \in \hat{F}_\infty^{(1)}[x_1, x_i]\}$ *for the given i-point* $(i \geqq 2)$ *problem* $(\hat{2})$ *defined in* $[x_1, x_i]$; *then for any k larger than the minimal value* $k > k^* + \varepsilon$, $\varepsilon > 0$, *there exists a* UEPS *pair in* $E_{1i}(k)$, *each of which has* $(i-2)$ *knots.*

Proof. We will use induction to prove the theorem. Assuming that the i-point problem has a UEPS pair with $(i-2)$ knots we show that an $(i+1)$-point problem, obtained by adding a point (x_{i+1}, y_{i+1}) coverable with k (i.e., $E_{1i+1}(k) \neq \phi$) has a UEPS pair with $(i-1)$ knots. We do this by showing that the additional point necessitates exactly one more knot than the UEPS for the i-point problem. We will discuss the UEPS in $E_{1i+1}(k)$, which attains the *lowest* ending value $l^i_{1i+1}(k)$ at x_i (in our terminology, an upper (lower) uniformly extreme spline if i is even (odd)). A similar approach can be used for a UEPS that attains the highest ending value at x_i, giving us the pair.

Let $s_i := l^i_{1i}(k)$, the lowest extremal value at x_i for the i-point problem data. Let us add another data point (x_{i+1}, y_{i+1}), coverable with k. Let q be the UEPS in $E_{1i+1}(k)$ for all data with $q(x_i) = l^i_{1i+1}(k)$. We consider the only possible two cases (a) and (b) separately.

(a) $L_i < s_i \leqq H_i$. Consider the interval $[x_i, x_{i+1}]$. Clearly, if (i) $H_i < s_i$ for the new point, then $E_{1i+1}(k) = \phi$ (Proposition 1(A)(iv)). Therefore, this case cannot arise, since the value of k considered in the proposition covers the problem data. (ii) If $H_i = s_i$, then by Proposition 2(C), k *is* the minimal value; this eliminates this case and thereby the possibility that the additional point does not necessitate any additional knots at all. (iii) If $L_i < s_i < H_i$, then $q(x_i) = s_i$ since obviously in this case $q(x_i) = \max(s_i = l^i_{1i}(k), L_i = l^i_{ii+1}(k)) = s_i$; that is, $L_i < s_i =$ the starting value of q at $x_i < H_i$, hence q has a knot in (x_i, x_{i+1}) (Proposition 1(A)(iii)).Thus for an $(i+1)$-point problem, taking q to be the UEPS for the i-point problem in $[x_1, x_i]$, and extending it in the *last* interval $[x_i, x_{i+1}]$ by defining $q(x) = g_i^-[s_i, k](x)$, $x_i \leqq x \leqq x_{i+1}$, we see that q has as many knots as the UEPS for the i-point problem in $[x_1, x_i]$, no knot at x_i (by Proposition 2(A)(i)(a), the left- and right-sided derivatives match at x_i, both being $-k$), and one more in (x_i, x_{i+1}). Therefore, q has $(i-2)+1 = (i-1)$ knots, as asserted. (iv) If $L_i = s_i$, then again $q(x_i) = s_i$, and there is a knot *at* x_i, since at x_i, $g_i^{++}[s_i, k]$, needed to cover d_i (Proposition 1(A)(ii)), obviously has k as the value of its right-sided derivative, while the UEPS achieving the lowest value at x_i for the i-point problem has $-k$ as its left-sided derivative (Proposition 2(A)(i)(a)). In this case also (as in (iii) above), q has $(i-1)$ knots, the correct count for an $(i+1)$-point problem. Thus there exists a q as desired for $L_i \leqq s_i < H_i$, q having a new knot in the last interval $[x_i, x_{i+1})$. Now we consider the only possibility left out: $L_i > s_i$.

(b) $L_i > s_i$. In this case we show that the UEPS q of the i-point data $\{x_r, y_r\}_2^{i+1}$, which exists by our induction assumption, can be extended in the *first* interval $[x_1, x_2]$ with exactly one additional knot to give a UEPS with $(i-2)$ knots for the $(i+1)$-point problem data $\{x_r, y_r\}_1^{i+1}$.

We know that in $E_{ri+1}(k)$ for any $r = 1, \ldots, i$, the lowest extremal value at $x_i \geqq L_i$ because the lowest extremal value of a problem must be greater than or equal to the lowest extremal value of any of its subproblems. In particular, $l^i_{2i+1}(k) \geqq L_i$. But in this case $\max(s_i = l^i_{1i}(k), L_i) = L_i$, therefore, $l^i_{2i+1}(k) \geqq L_i \geqq l^i_{1i}(k)$. That is, the lowest extremal value at x_i for the i-point problem $\{x_r, y_r\}_2^{i+1}$ is *greater than* $l^i_{1i}(k)$, the lowest ending value for the i-point problem $\{x_r, y_r\}_1^i$. Therefore, by Proposition 2(B) for odd i, the highest starting extremal value at x_2 for the i-point problem $\{x_r, y_r\}_2^{i+1}$ is less than or equal to the highest starting extremal value at x_2 for the i-point problem $\{x_r, y_r\}_1^i$: $h^2_{2i+1}(k) \leqq h^2_{1i}(k)$. Now, obviously (again because the highest extremal value of a problem is less than or equal to the highest extremal value of any of its subproblems), $h^2_{1i}(k) \leqq H_2$, thus $h^2_{2i+1}(k) \leqq h^2_{1i}(k) \leqq H_2$, and by Proposition 2(A)(ii)(b) the right-sided derivative of a UEPS in $E_{2i+1}(k)$ attaining the highest extremal value at x_2 is $-k$. Therefore (similar to (a)(iii) above), for $h^2_{2i+1}(k) < H_2$ there is a starting value s_1 (see the observation following Proposition 1) such that $g_1^+[s_1, k](x)$ covers d_1 with the ending value $g_1^+[s_1, k](x_2) = h^2_{2i+1}(k) = q(x_2)$ and one knot in (x_1, x_2). And (similar to (a)(iv) above), for $h_{2i+1}^2(k) = H_2$, there is s_1 such that $g_1^{++}[s_1, k](x)$ covers d_1 with $q(x_2) = H_2 = g_1^{++}[s_1, k](x_2)$ and a knot *at* x_2. This gives us the desired UEPS. □

3.1.2. Different perspectives. As compared to Karlin's theorem, there are two useful, and perhaps more satisfying, perspectives from which we may view the statement of Theorem 1. First, it applies to all the values $k > k^*$ (the minimal value) in a positive way; that is, for any such k the claimed extremal solution *exists*. Second, there is no ambiguity about the number of knots. For any $k > k^*$, all the uniformly extreme perfect splines for the i-point problem have exactly $(i-2)$ knots (one more than the Karlin count). Recall that Karlin's characterization theorem (§ 1) emphasizes the minimal value $k = k^*$, and gives only an upper bound on the number of knots.

Of these, the certainty of the number of knots is crucial in Theorem 1. It compels us to consider uniformly extremal splines, which are amenable to fast computation by direct formulas and therefore useful in finding optimal solutions of the problem. We discuss this application below.

3.2. Computing optimal solutions. To compute optimal solutions of $(\hat{2})$ and thus of (2) (as shown in § 2.1), the above discussion suggests the approach outlined below.

For *all* values of k larger than the minimal value $k^* = \min\{\|f^{(1)}\|_\infty : f \in \hat{F}_\infty^{(1)}[x_1, x_i]\}$, Theorem 1 asserts that there exists a UEPS in $E_{1i}(k)$ "covering" the given data. If such a spline exists, let us call the given value of k *feasible*; otherwise, call it *infeasible*. To determine the minimal value k^*, we need to check whether a given value of k is feasible or not because, to approach the minimal value k^*, a given value of k needs to be increased if infeasible and decreased if feasible. Once found, the infeasible and feasible values will also serve as lower and upper bounds on the minimal value we want to compute. Therefore, *if* we can check the feasibility/infeasibility of a given value of k, we can easily use a search-type procedure, such as bisection, to reduce the difference between these upper and lower bounds in each iteration and find the minimal value within any desirable tolerance. Knowing the minimal value k^*, then, also allows us to compute an optimal spline [18], [19].

Now, checking the feasibility of a given value of k by considering uniformly extremal splines is simple. It is essentially based on the successive ($r = 2, \ldots$) application of the facts that to attain the lowest (highest) extremal value at x_r, we start at the highest (lowest) extremal value t at x_{r-1} of the $(r-1)$-point problem and use a $g_{r-1}^+[t, k]$ ($g_{r-1}^-[t, k]$)-type function (Proposition 1(D)(iii)). Using these, for a given interval and data to its left, we get formulas to compute the extremal ending values and conditions to check whether the data to its right is "coverable" or not; that is, whether a given value of k is feasible or not.

Using extremal value formulas and feasibility criteria implied above, a direct algorithm and its implementation in FORTRAN are given in [19]. The algorithm seems fast for even large values of p; problems of up to $p = 200$ points were solved in less than two seconds on an IBM3090 under CMS. The optimal k^* was obtained to within a tolerance of $\pm\varepsilon = 0.00001$ in about 15 to 25 iterations for all the problems solved. Once the minimal k^* is found, the method to compute an optimal spline function and maximum/minimum envelopes [5], [6] are also given in [19].

3.3. An elementary proof of Karlin's theorem. As discussed before, Theorem 1 offers perspectives on (uniformly) extremal solutions related to but somewhat different from Karlin's theorem. In fact, however, it is equivalent to Karlin's theorem. Since Theorem 1 depends only on elementary facts, this gives us, for the quadratic case treated here, an elementary proof of Karlin's theorem, which otherwise requires many advanced mathematical tools, as discussed in § 1.

THEOREM 2. *Theorem* 1 *is equivalent to Karlin's theorem.*

Proof. Consider an i-point problem, and some $k' > k$, where k is the minimal value for the i-point problem. Assume that Theorem 1 is false; that is, there is no UEPS in $E_{ij}(k')$ with $(i-2)$ knots. If knots of q, any arbitrary UEPS in $E_{1i}(k')$, are less than $(i-2)$ (i.e., *less than* or *equal* to $(i-3)$), then k' is minimal, since one can show (essentially using Rolle's theorem) that any perfect spline solution with knots less than or equal to $(i-3)$ is a solution of the problem [2], [8]. So assume that q attains the highest extremal value at x_i and one has one extra knot than what Theorem 1 asserts: it has $(i-2)+1 = i-1$ knots. We will show that this implies that Karlin's theorem is false for an $(i+1)$-point problem constructed below. Add the $(i+1)$th point (x_{i+1}, y_{i+1}) such that $d_i = q(x_i) + k'(x_{i+1} - x_i)/2$. Then it is clear by the construction (which implies $h^i_{1i}(k') = L_i(k')$) that k' has become minimal for the $(i+1)$-point problem (Proposition 2(C)), and that there is no knot at x_i since the left-sided derivative $q^{(1)}(x_i)$ and the right-sided derivative of a g_i^{++}-type function at x_i needed to cover d_i in $[x_i, x_{i+1}]$ both have the value k (Propositions 2(A)(i)(b) and 1(A)(iii)). Therefore, any perfect spline solution for the $(i+1)$-point problem has at least $(i-1)$ knots, one more than the Karlin count for an $(i+1)$-point problem, which implies that Karlin's theorem is false. This proves that Karlin's theorem implies Theorem 1.

Now assume that Karlin's theorem is false. We will show that this implies that Theorem 1 is false, showing that Theorem 1 implies Karlin's theorem. Again consider the $(i+1)$-point problem given above. Since we are assuming Karlin's theorem to be false, any perfect spline solution of the $(i+1)$-point problem has at least one more knot than its Karlin count equal to $(i+1)-3 = i-2$; that is, it has at least $(i-2)+1 = (i-1)$ knots, and by construction no knot at x_i. Now dropping the (x_{i+1}, y_{i+1}) point and considering the i-point problem we see, again by construction, that k' is larger than the minimal k for this problem, and since there was no knot at x_i, any uniformly extreme perfect spline will have at least $(i-1)$ knots. For any i-point problem, this is a contraction of Theorem 1. □

4. Summary and concluding remarks. For the quadratic case, the analysis of uniformly extremal splines in the above framework leads to a simple and effective computational method and to significant simplification of the arguments characterizing the optimal solutions of problem (1). The approach can be expected to be useful for the *higher-degree problems* in two parallel ways: (1) The constructive details of the analysis, e.g., how we attain the maximum/minimum ending or staring points (extremal value formulas), or how one checks if a given $k = \|f^{(2)}\|_\infty$ value is feasible for an additional data point (feasibility criteria), may be used to help compute the optimal solutions of such problems efficiently. As discussed before, this has been successfully implemented for the quadratic case. (2) By appropriately identifying the roles of $f^{(1)}$ with $f^{(n-1)}$ and $f^{(2)}$ with $f^{(n)}$ in a suitable framework, it may form a starting basis for an elementary proof of the general characterization theorem. In addition, (3) since nothing restricts uniformly extremal splines specifically to problem $(\hat{2})$, they can be used for *other related problems*, e.g., problem (1) with convexity/concavity requirements on f [8], [17], and the problem of finding maximum/minimum function value envelopes for a given a priori bound on $\|f^{(n)}\|_\infty$ [5], [6], [19]. The study of (1)-(3) would seem a fruitful and challenging research task for the future.

Appendix. Proofs of Propositions 1 and 2.

Proof (Proposition 1). (A)(i) By the definition of H_i, $g_i^{--} \in E_i(s, k)$. If there is any other $f \in E_i(s, k)$ different from g_i^{--}, let it have a different value at $x' \in (x_i, x_{i+1})$.

Then either $f(x') < g_i^{--}(x')$, or $f(x') > g_i^{--}(x')$. Let $x^* = x'$ in the former case. In the latter case, since $\int_{x_i}^{x_{i+1}} f(x)\,dx = \int_{x_i}^{x_{i+1}} g_i^{--}(x)\,dx = d_i$, we must have some $\bar{x} \in (x_i, x_{i+1})$ such that $f(\bar{x}) < g_i^{--}(\bar{x})$; let $x^* = \bar{x}$. Then $((f(x^*) - f(x_i))/(x^* - x_i)) < -k$, therefore, $f \in W_\infty^{(1)}[x_i, x_{i+1}]$ could not have $\|f^{(1)}\|_\infty \leqq k$, contradicting the statement that $f \in E_i(s, k)$. (ii) Similar to (i). (iii)(a) Define $M(t) = \int_{x_i}^{t} h^-(x)\,dx + \int_t^{x_{i+1}} h[t, h_t, k](x)\,dx$, where $h_t = h^-(t)$, then by taking $\delta < ((\varepsilon/k + \delta_1^2)^{1/2} - \delta_1)$, for any $\bar{x}_1$, $\bar{x}_2$ in $[x_i, x_{i+1}]$ with $|\bar{x}_1 - \bar{x}_2| < \delta$, and $\delta_1 = x_{i+1} - \bar{x}_2$, we can show that $M(t)$ is a decreasing continuous function of t in $[x_i, x_{i+1}]$, with $M(x_i) > d_i$, and $M(x_{i+1}) < d_i$. Therefore there must be a unique $x_i^- \in (x_i, x_{i+1})$ such that $M(x_i^-) = d_i$, giving us the desired g_i^-. Similarly we can find g_i^+. (b) Now, since in the neighborhood of x_i, $g_i^+ > g_i^-$ and $\int_{x_i}^{x_{i+1}} g_i^+(x)\,dx = \int_{x_i}^{x_{i+1}} g_i^-(x)\,dx = d_i$, there must be some $x' \in (x_i, x_{i+1})$ such that $g_i^+(x') < g_i^-(x')$. To do this, g_i^+ must have changed the sign of its slope in (x_i, x'). But g_i^+ can do this only once by definition, hence $g_i^+ < g_i^-$ for all $x \in [x', x_{i+1}]$. The other conclusions follow from this. (iv) Let $s < L_i, f \in E_i(s, k)$. Since $\int_{x_i}^{x_{i+1}} f(x)\,dx = \int_{x_i}^{x_{i+1}} g_i^{--}(x)\,dx$ and $f(x_i) = s < L_i = g_i^{--}(x_i)$, as in (i) above, $\|f^{(1)}\|_\infty \nleqq k$, contradicting the statement that $f \in E_i(s, k)$. The same is true if $s > H_i$. For $L_i \leqq s \leqq H_i$, functions in $E_i(s, k)$ have been defined in (i)-(iii) above.

(B) Let $f \in E_i(s, k)$ and $f(x_{i+1}) > g_i^-(x_{i+1})$. Since $\int_{x_i}^{x_{i+1}} f(x)\,dx = \int_{x_i}^{x_{i+1}} g_i^-(x)\,dx$, there must be $x' \in (x_i, x_{i+1})$ with $f(x') < g_i^-(x')$. Then (a) if $x' \in (x_i, x_i^-)$, considering points (x_i, s), $(x', f(x'))$, it is clear that $f^{(1)} < -k$ somewhere in (x_i, x'), and (b) if $x' \in [x_i^-, x_{i+1}]$, then considering points $(x', f(x'))$, $(x_{i+1}, f(x_{i+1}))$, we must have $f^{(1)} > k$ somewhere in (x', x_{i+1}). In both cases $\|f_{(1)}\|_\infty \nleqq k$, a contradiction to $f \in E_i(s, k)$. Similarly, one shows the other inequality.

(C) Since $[s', s] \subseteq [L_i, H_i]$, by A(i)-(iii), for any $t \in [s', s]$, $E_i(s', s, k)$ is nonempty containing unique $g_i^-[t, k]$ and $g_i^+[t, k]$.

(D)(i) If $t > t'$, we have $g_i^+[t, k] > g_i^+[t', k]$, and $g_i^-[t, k] > g_i^-[t', k]$ in the neighborhood of x_i. Then the arguments as in proof (A)(iii)(b) imply that $g_i^+[t, k](x_{i+1}) < g_i^+[t', k](x_{i+1})$ etc., and it follows that $G(t)$, $G'(t)$ are decreasing functions of $t \in [s', s]$. (ii) The continuity of $G(t)$, $G'(t)$ is also seen directly: for any $\varepsilon > 0$ we can take $\delta < \varepsilon(x_{i+1} - x_i^+)/(x_i^+ - x_i)$, $\delta' < \varepsilon(x_{i+1} - x_i^-)/(x_i^- - x_i)$, implying that $|g_i^+[t, k](x_{i+1}) - g_i^+[t', k](x_{i+1})| < \varepsilon$ for all $|t - t'| < \delta$, and $|g_i^-[t, k](x_{i+1}) - g_i^-[t', k](x_{i+1})| < \varepsilon$ for all $|t - t'| < \delta'$, t, t', both in $[s', s]$. (iii) Clearly, $[s', s]$ is compact, and as just shown, $G(t) = g_i^+[t, k](x_{i+1})$ and $G'(t) = g_i^-[t, k](x_{i+1})$ are both continuous and decreasing in t, $t \in [s', s]$. From this it is clear that g_i's defined by (4) have $g_i^+(x_{i+1}) \leqq f(x_{i+1}) \leqq g_i^-[s', k](x_{i+1})$, for any $f \in E_i(s', s, k)$. (iv) Let $g := g_i^-[t, k]$ and $g' := g_i^-[t, k']$. Since in the neighborhood of x_i, $g' > g$, therefore, as in the proof (A)(iii)(b), $g'(x) < g(x)$ for all $x \in [x', x_{i+1}]$. The other inequality follows similarly. □

Proof (Proposition 2). (A)(i)(a) $q(x_i) = l_{1i}^i(k)$. Consider $h := h_{1i}^{i-1}(k)$, the extremal value at x_{i-1}. Proposition 1(D)(iii) implies that $q(x) = g_{i-1}^+[h, k]$ in $x_{i-1} \leqq x \leqq x_i$, since we must start at the highest value at x_{i-1} and use the g_{i-1}^+ function to attain the lowest ending value at x_i (among all the functions in $E_{i-1i}(l_{1i}^{i-1}(k), h_{1i}^{i-1}(k), k)$). The value of the left-sided derivative at x_i is $-k$ by definition for the function $g_{i-1}^+[h, k]$, except, of course, when it coincides with $g_{i-1}^{++}[h, k]$. This occurs if and only if $h = L_{i-1}$ (Proposition 1(A)(ii)), which, due to $g_{i-1}^{++}[h, k'](x) < g_{i-1}^{++}[h, k](x)$ for $x_{i-1} < x < x_i$, and Proposition 1(D)(iv) implies that for any $k' < k$ we have $E_{i-1i}(l_{1i}^{i-1}(k), h_{1i}^{i-1}(k), k') = \phi$. This obviously means $E_{1i}(k') = \phi$. Since this is a contradiction, $g_{i-1}^+[h, k]$ cannot coincide with $g_{i-1}^{++}[h, k]$, and we always have $q^{(1)}(x_i) = -k$. (b) $q(x_i) = h_{1i}^i(k)$. Here we must have $q(x) = g_{i-1}^-[l_{1i}^{i-1}(k), k]$, $x_{i-1} \leqq x \leqq x_i$, and our conclusion follows parallel to (i)(a) above. (Note that this proposition holds for the *last* x_i value since it is unfettered by any data to its right; it may not be true at other values.) We can prove (ii)(a) and (b) similarly.

(B) This is a direct consequence of the following facts: (i) to attain the lowest (highest) extremal value at x_r, we start at the highest (lowest) extremal value t at x_{r-1} of the $(r-1)$-point problem and use the $g^+_{r-1}[t, k]$ ($g^-_{r-1}[t, k]$) function (Proposition 1(D)(iii)). (ii) Both $g^+_{r-1}[t, k](x_r)$ and $g^-_{r-1}[t, k](x_r)$ are *decreasing* functions of t (Proposition 1(D)(i)), thus the higher the starting value, the lower the ending value; and (iii) since, clearly, the lowest (highest) extremal value of a problem cannot be lower (higher) than the lowest (highest) extremal value of any of its subproblems, the maximum of $\{L_r, g^+_{r-1}[t, k](x_r)\}$ (minimum of $\{H_r, g^-_{r-1}[t, k](x_r)\}$) determines the lowest (highest) extremal value at x_r in $E_{1r}(k)$. Applying these in each interval $[x_r, x_{r+1}]$, successively, $r = 1, \dots, i-1$, it is clear that if q_2 starts off at a higher value than q_1 at x_1, it will end up attaining at least as low a value as $q_1$at x_i. Thus, if q_2 does not attain as low an ending value as q_1 at x_i, $q_2(x_i) > q_1(x_i)$, then we must have the starting value $q_2(x_1) \leqq q_1(x_1)$. (The other cases follow similarly.)

(C) We consider $q(x_{i-1}) = h^{i-1}_{1i-1}(k) = L_{i-1}$; the other case follows similarly. First note that we may have $E_{1i-1}(k') = \phi$ for any $k' < k$; in that case, obviously, $E_{1i}(k') = \phi$. Hence, let $E_{1i-1}(k') \neq \phi$ for some $k' < k$. Proposition 1(D)(iv) implies that the highest attainable ending value at x_{i-1} with a value $k' < k$ cannot be higher than when we use k; i.e., $h^{i-1}_{1i-1}(k') \leqq h^{i-1}_{1i-1}(k)$. Under *inequality* $h^{i-1}_{1i-1}(k') < h^{i-1}_{1i-1}(k) = L_{i-1}$, we have the highest possible starting value of a function, which is to cover d_{i-1} in (x_{i-1}, x_i), lower than L_{i-1}. But, by Proposition 1(A)(iv), this cannot be done, implying $E_{1i}(k') = \phi$. In the *equality* case $h^{i-1}_{1i-1}(k') = h^{i-1}_{1i-1}(k)$, note that by Proposition 1(A)(ii), when the starting value $s = L_{i-1}$, the only function $g \in E_{i-1i}(s, k)$ which covers d_{i-1} in (x_{i-1}, x_i) is $g^{++}_{i-1}[s, k]$. However, since $g^{++}_{i-1}[s, k'] < g^{++}_{i-1}[s, k]$ for $k' < k$, $x \in (x_{i-1}, x_i)$, it is clear that $g^{++}_{i-1}[s, k']$ cannot cover d_{i-1} in (x_{i-1}, x_i), and we must have $E_{1i}(k') = \phi$, and k is the *minimal* required value for the i-point problem. □

REFERENCES

[1] C. deBoor (1974), *A remark concerning perfect splines*, Bull. Amer. Math. Soc., 80, pp. 724-727.

[2] ——— (1977), *Computational aspects of optimal recovery*, in Optimal Estimation in Approximation Theory, Proc. Internat. Symposium, Freudenstadt, 1976, C. A. Micchelli and T. J. Rivlin, eds., Plenum, New York, pp. 69-91.

[3] S. D. Fisher and J. W. Jerome (1974), *The existence, characterization and essential uniqueness of solutions of L_∞ extremal problems*, Trans. Amer. Math. Soc., 187, pp. 391-404.

[4] ——— (1974), *Perfect spline solutions to L_∞ extremal problems*, J. Approx. Theory, 12, pp. 78-90.

[5] P. W. Gaffney (1978), *To compute the optimal interpolation formula*, Math. Comp., 32, pp. 763-777.

[6] P. W. Gaffney and M. J. D. Powell (1976), *Optimal Interpolation*, in Numerical Analysis, Lecture Notes in Math. 506, G. A. Watson, ed., Springer-Verlag, Berlin, New York, pp. 90-99.

[7] G. Glaeser (1967), *Prolongement extremal de fonctions differentiables*, Publ. Sect. Math. Faculté des Sciences Rennes, Rennes, France.

[8] G. L. Illiev and W. Pollul (1984), *Convex interpolation with minimal L_∞-norm of the second derivative*, Math. Z., 186, pp. 49-56.

[9] J. Jerome (1973), *Minimization problems and linear and nonlinear spline functions.* I: *Existence*, SIAM J. Numer. Anal., 10, pp. 808-819.

[10] S. Karlin (1973), *Some variational problems on certain Sobolev spaces and perfect splines*, Bull. Amer. Math. Soc., 79, pp. 124-128.

[11] ——— (1975), *Interpolation properties of generalized perfect splines and the solutions of certain extremal problems.* I, Trans. Amer. Math. Soc., 206, pp. 25-66.

[12] R. Louboutin (1967), *Sur une bonne partition de l'unite*, in Le Prolongateur de Whitney, Vol. II, G. Glaeser, ed., University of Rennes, Rennes, France.

[13] C. A. Micchelli (1977), *Best L_1 approximation by weak Chebyshev systems and the uniqueness of interpolating perfect spline*, J. Approx. Theory, 19, pp. 1-14.

[14] I. J. Schoenberg (1971), *The perfect B-spline and a time-optimal control problem*, Israel J. Math., 10, pp. 261-274.

[15] L. S. THAKUR (1978), *Error analysis for convex separable programs: The piecewise linear approximation and the bounds on the optimal objective value*, SIAM J. Appl. Math., 34, pp. 704–714.

[16] ——— (1980), *Error analysis for convex separable programs: bounds on optimal and dual optimal solutions*, J. Math. Anal. Appl., 75, pp. 486-496.

[17] ——— (1986), *Optimal interpolation with convex splines of second degree*, SIAM J. Control Optim., 24, pp. 157–168.

[18] ——— (1986), *A computable convex programming characterization of optimal interpolatory quadratic splines with free knots*, J. Math. Anal. Appl., 114, pp. 278-288.

[19] ——— (1990), *A direct algorithm for optimal quadratic splines*, Numer. Math., 57, pp. 313-332.

[20] ——— (1990), *Quadratic spline functions; Applications and computations*, Working paper WP90-014, Operations and Information Management Dept., Univ. of Connecticut, Storrs, CT, pp. 1-26.

[21] E. J. WEGMAN AND I. W. WRIGHT (1983), *Splines in statistics*, J. Amer. Statist. Assoc., 78, pp. 351-365.

SIAM J. OPTIMIZATION
Vol. 3, No. 2, pp. 248-267, May 1993

PARTIALLY-FINITE PROGRAMMING IN L_1 AND THE EXISTENCE OF MAXIMUM ENTROPY ESTIMATES*

J. M. BORWEIN† AND A. S. LEWIS†

Abstract. Best entropy estimation is a technique that has been widely applied in many areas of science. It consists of estimating an unknown density from some of its moments by maximizing some measure of the entropy of the estimate. This problem can be modelled as a partially-finite convex program, with an integrable function as the variable. A complete duality and existence theory is developed for this problem and for an associated extended problem which allows singular, measure-theoretic solutions. This theory explains the appearance of singular components observed in the literature when the Burg entropy is used. It also provides a unified treatment of existence conditions when the Burg, Boltzmann-Shannon, or some other entropy is used as the objective. Some examples are discussed.

Key words. convex analysis, duality, existence, generalized solution, image reconstruction, maximum entropy method, moment problem, partially finite program, spectral estimation

AMS subject classifications. primary 49A55, 90C25; secondary 65K05, 49B27

1. Introduction: Best entropy estimation.

A very common problem in many areas of the physical sciences consists of trying to estimate an unknown density by measuring some of its moments. More precisely, given a number of integrals of an unknown function with respect to known weight functions, and a real interval in which the function is known to take its values, we seek to estimate the function. Typically, the weight functions are trigonometric polynomials, frequently multidimensional (so the given moments are Fourier coefficients), or algebraic polynomials (giving power moments), and the given interval is often (though not exclusively) the nonnegative reals.

Given only a finite number of moments this estimation problem is clearly underdetermined. One extremely popular method for selecting an estimate from the family of all functions satisfying the prescribed moment constraints is to choose it to minimize some objective functional (subject to the given constraints). This objective is typically some measure of entropy—hence the term "best entropy estimation." This approach has been widely and successfully used in such diverse areas as astronomy, crystallography, speech processing, tomography, geophysics, and many others. For surveys, see [31] and [35] (containing in total almost 700 references), and the recent collections, [54], [53], [17], and [51].

Phrased mathematically, the best entropy estimation problem becomes, in its simplest form,

$$(1.1)\qquad \begin{array}{ll} \text{minimize} & \displaystyle\int \phi(x(s)) \\ \text{subject to} & \displaystyle\int a_i x = b_i \quad \text{for } i = 1, \dots, n. \end{array}$$

The variable density to be chosen is x, the a_i's are the known weight functions, and the b_i's are the measured moments. The function ϕ reflects our choice of entropy: it may take the value $+\infty$ to incorporate the known range constraint on x. For reasons

* Received by the editors May 10, 1991; accepted for publication (in revised form) February 21, 1992. This research was partially supported by the Natural Sciences and Engineering Research Council of Canada.

† Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1.

discussed in [6] we may as well restrict ourselves to closed, proper, convex functions ϕ. The two classical choices correspond to the Boltzmann-Shannon entropy, perhaps first suggested in this context in [27],

$$\phi(u) := \begin{cases} u \log u & \text{if } u > 0, \\ 0 & \text{if } u = 0, \\ +\infty & \text{if } u < 0, \end{cases} \tag{1.2}$$

and the Burg entropy, first proposed in [12],

$$\phi(u) := \begin{cases} -\log u & \text{if } u > 0, \\ 0 & \text{if } u \leqq 0, \end{cases} \tag{1.3}$$

although numerous other entropies have appeared in the literature, including L_2 and L_r entropies [25], [29], and [3], and the general families proposed in [40], [39], and [13].

The debate over the relative merits of the various entropies has been intense, as the above references will testify. The choice between (1.2) and (1.3) has been particularly controversial (see, for example, [28] and [52]). The issues in this debate can be grouped into three rather distinct areas. The first might be termed a priori reasons for selecting a particular entropy, generally involving a probabilistic, statistical, or information-theoretic discussion of the underlying phenomenon we seek to measure (see for example, [40], [52], [28], [39], and [13]). The second area of debate is empirical: the performance of the method is judged by its ability to reconstruct a known density from its moments (see, for example, [40], [28], [52], and [29]). Both of these areas lie outside our current scope.

The third area might be called a posteriori reasons: mathematical properties of the estimates arising from a particular choice of entropy are studied. Two particular properties have attracted attention: the existence of the optimal estimates, and their convergence to the underlying density as the number of given moments grows. For questions of convergence, see [52], [37], [22], [50], [18], [34], [19], [13], [7], [5], and [11]. In this paper we shall concentrate on the first property: the existence of an optimal solution for the estimation problem (1.1).

The basic idea for solving (1.1) has been explained widely in the applied literature, although for the most part without any degree of rigour: the form of the optimal solution is derived by attaching Lagrange multipliers $\lambda_1, \dots, \lambda_n$ to the constraints, and then differentiating (formally), giving

$$\bar{x}(s) := (\phi')^{-1}\left(\sum_1^n \lambda_i a_i(s)\right), \tag{1.4}$$

where the λ_i's are chosen to ensure that $\bar{x}$ is feasible. Two existence questions need to be addressed to make this rigorous. First, when do the multipliers $\lambda_1, \dots, \lambda_n$ exist? Put differently, we require the existence of an optimal solution to the dual problem for (1.1). As usual in convex programming, the required condition is a primal constraint qualification for (1.1). This is straightforward to check: a general theory for "partially-finite programs" (convex programs with an infinite-dimensional variable subject to a finite number of linear constraints) is developed in [9] and [6].

The second question is more delicate: when does (1.4) give the optimal solution? Under mild conditions, it does so provided that we know a priori that an optimal solution exists. This is the case, for example, when the objective function has weakly compact level sets, as is the case with the Boltzmann-Shannon entropy [7], but the important case of the Burg entropy is not covered by this idea. Existence was shown for important special cases in [15] and [56], and a general condition ensuring existence was introduced in [32] together with a demonstration that it may fail in general.

A fascinating concrete example of the nonexistence of a best Burg entropy estimate appeared in [40] (see also [52] and [14]). The problem was very simple: the unknown function was a probability density on the unit cube in $\mathbb{R}^3$, with three of its (multidimensional) Fourier coefficients given equal to a parameter α in $[0, 1)$. It turns out that the Lagrange multipliers always exist, and, at least for small α, (1.4) gives the correct best Burg entropy estimate. However, as α increases to a certain critical value the solution becomes more and more concentrated, and beyond this value (1.4) fails to give even a feasible estimate.

The explanation given in the above papers in a self-professed nonrigorous fashion is that part of the real solution has condensed to a point mass, a claim also supported by considering discretized versions of the problem. The initial motivation of this work is to give a rigorous explanation of this phenomenon. In the course of this explanation we will develop a rather general duality and existence theory for the problem (1.1).

If, as the above example suggests, we should accept the possibility of measure-theoretic solutions to (1.1), then the question arises of how to reformulate the objective function. The constraints give no difficulty providing the a_i's are continuous, and the case where ϕ is piecewise linear and continuous is also clear—there is a strong analogy with semi-infinite linear programming, where point-mass solutions are familiar (see, for example, [1]).

The correct approach in the general case turns out to be to replace the objective function in (1.1) by what is essentially its second conjugate, which becomes a functional defined on measures. This idea is not in itself particularly new: see, for example, the discussion of "generalized solutions" in [16] and [47]. What is more remarkable is the simplicity and tractability of the resulting problem. In the first three sections of this work, relying heavily on the work of Rockafellar [43]-[47], we derive this extended primal problem, and investigate its relationship with the original primal and dual problems.

The next section returns to the underlying question of the existence of an optimal solution for the original problem (1.1). Using the extended solutions, we provide a general theory linking the boundary behaviour of the entropy and the local geometry of the underlying measure space with the existence question. This provides a unified and illuminating explanation of previous results in the literature [15], [56], [32], [6]. The last section discusses how extended solutions can be computed, and ends with some examples including a resolution of the example described above.

Just prior to submitting this article for publication, the authors became aware of recent unpublished work [20], [21] on some similar questions. The approach therein is very different from the purely convex analytic attack employed here. It relies on discretization and a Bayesian statistical interpretation, which lead to the application of large deviation theory (building on results in [13]). This probabilistic method, while seemingly less constructive than the convex programming approach, suggests intriguing connections between the two.

Problem (1.1) is a very general partially-finite program. As such, it models very many problems other than best entropy estimation. In particular, as outlined in [9], it includes numerous examples from constrained approximation, interpolation, and smoothing (see, for example, [38], [26], and [10]); the duality theory developed here also applies to some of these problems. The theory in this paper also allows an arbitrary linear functional to be added to the objective function. There has been recent interest in log-barrier penalty methods for semi-infinite linear programming, in the context of the asymptotic behaviour of Karmarkar's method [42], [55], and our results may be applied here.

In the interests of economy, many reasonably routine computations and proofs are omitted; they can be found in [8] and [33].

2. Preliminaries. The measures of entropy with which we shall be concerned are integral functionals of the form $\int \phi(x(s))$, where $\phi:\mathbb{R}\to(-\infty,+\infty]$ is a closed, proper, convex function. We shall use the notation and terminology of [45] throughout. The conjugate function is denoted by ϕ^*, and the recession function $\phi 0+:\mathbb{R}\to(-\infty,+\infty]$ is given by $(\phi 0+)(u)=\lim_{\lambda\to+\infty}(1/\lambda)\phi(u_0+\lambda u)$, where u_0 is arbitrary in the domain of ϕ (see [45, Thm. 8.5]). The following result defines the constants p and q, which will be crucial in this paper. (These are entirely unrelated to the notation for the spaces L_p and L_q.) The proof is standard (see [8, Lemma 2.2]).

LEMMA 2.1. *The following limits exist*:

$$q:=\lim_{u\to+\infty}\phi(u)/u\in(-\infty,+\infty],$$
$$p:=\lim_{u\to-\infty}\phi(u)/u\in[-\infty,+\infty). \tag{2.2}$$

Furthermore, $p\leqq q$,

$$(\phi 0+)(u)=\begin{cases} qu & \textit{if } u>0,\\ 0 & \textit{if } u=0,\\ pu & \textit{if } u<0,\end{cases} \tag{2.3}$$

and int (dom (ϕ^*)) $=(p,q)$. *The function* ϕ *is affine if and only if* $p=q$ (*so* dom $(\phi^*)=\{p\}$).

Lemma 2.1 characterizes dom (ϕ^*). It will also be helpful to have some notation for dom (ϕ), so define β in $(-\infty,+\infty]$ as sup (dom ϕ) and α in $[-\infty,+\infty)$ as inf (dom ϕ), so int (dom (ϕ)) $=(\alpha,\beta)$. The ideas of *essential strict convexity* and *essential smoothness* [45] will be useful to us. These concepts are particularly simple for univariate functions. We have that ϕ is essentially strictly convex (or, equivalently, ϕ is strictly convex on dom (ϕ)) if and only if ϕ^* is essentially smooth. This in turn is equivalent to $p<q$ and ϕ^* differentiable on (p,q) with $\lim_{v\downarrow p}(\phi^*)'(v)=-\infty$ if $p>-\infty$, and $\lim_{v\uparrow q}(\phi^*)'(v)=+\infty$ if $q<+\infty$. In this case,

$$\partial\phi^*(v)=\begin{cases}\{(\phi^*)'(v)\} & \text{if } v\in(p,q),\\ \emptyset & \text{otherwise.}\end{cases} \tag{2.4}$$

One particularly well behaved class of convex functions is that of Legendre type [45].

DEFINITION 2.5. We say ϕ is of *Legendre type* if it is essentially smooth and essentially strictly convex.

LEMMA 2.6. *Suppose* ϕ *is of Legendre type. Then, so is* ϕ^*, *and* $\phi':(\alpha,\beta)\to(p,q)$, $(\phi^*)':(p,q)\to(\alpha,\beta)$ *are continuous, strictly increasing, and mutually inverse maps between the interiors of the domains of* ϕ *and* ϕ^*. *Also*, $q<+\infty$ *if and only if* $\beta=+\infty$ *and* $p>-\infty$ *if and only if* $\alpha=-\infty$.

Proof. See [45, Thm. 26.5]. The last part is immediate. □

We will use the notation, for u in $\mathbb{R}$, $u^+:=\max\{u,0\}$, and $u^-:=-\min\{u,0\}$, so $u=u^+-u^-$ and $|u|=u^++u^-$. If we adopt the convention that $(\pm\infty)\ 0=0$, we can rewrite (2.3) as $(\phi 0+)(u)=qu^+-pu^-$.

The results in this paper will revolve around the computation of the conjugates and subdifferentials of various convex integral functions. We will rely heavily on the ideas and results of Rockafellar [45], [46]. For convenience, we will summarize the notation to be used throughout the paper before proving the technical results that will be applied.

S is a compact Hausdorff space, with $z_0 \in C(S)$, the Banach space of continuous functions on S. Furthermore, $0 \leqq \rho \in M(S)$, the Banach space of regular Borel measures on S, and ρ has full support [48]. $I_\phi : L_1(S, \rho) \to (-\infty, +\infty]$ is defined by $I_\phi(x) := \int_S \phi(x(s))\, d\rho$, and $I_{\phi^*} : L_\infty(S, \rho) \to (-\infty, +\infty]$ is defined by $I_{\phi^*}(z) := \int_S \phi^*(z(s))\, d\rho$. $J_{\phi^*} : C(S) \to (-\infty, +\infty]$ is defined as the restriction of I_{ϕ^*} to $C(S)$. We have $b \in \mathbb{R}^n$, and $a = (a_1, \dots, a_n) \in (C(S))^n$. The map $A : L_1(S, \rho) \to \mathbb{R}^n$ is defined by $(Ax)_i := \int_S a_i(s) x(s)\, d\rho$ for $i = 1, \dots, n$. Finally, $B : \mathbb{R}^n \to C(S)$ is defined by $B\lambda := \lambda^T a$.

Some comments are in order concerning these definitions. We will often treat $C(S)$ with its usual supremum norm as a subspace of $L_\infty(S, \rho)$. We can regard $M(S)$, with its usual norm, as the dual of $C(S)$. The continuous linear map A has adjoint $A^* : \mathbb{R}^n \to L_\infty(S, \rho)$, which may be identified with the continuous linear map B, as is easily checked. Also, $B^* : M(S) \to \mathbb{R}^n$ is continuous and given by $(B^*\rho)_i := \int a_i\, d\rho$. For the relevant ideas, see, for example, [48] and [49].

The function ϕ is a normal convex integrand, so the integral functional I_ϕ is a well-defined, convex, lower semicontinuous function, with conjugate I_{ϕ^*} [46]. The function J_{ϕ^*} is also well defined and convex (see, for example, [47, Thm. 3]). Much of this section will be devoted to studying its conjugate.

We will write, for any μ in $M(S)$, $\mu = \mu^+ - \mu^-$ for the Jordan decomposition, $\mu = \mu_\alpha + \mu_\sigma$ for the Lebesgue decomposition with respect to ρ (so $\mu_\alpha \ll \rho$ and $\mu_\sigma \perp \rho$), and $(d\mu_\alpha / d\rho) \in L_1(S, \rho)$ for the Radon-Nikodym derivative [48].

THEOREM 2.7. *The function J_{ϕ^*} is well defined, lower semicontinuous, and convex. It is continuous on the set $\{z \in C(S) \mid z(s) \in (p, q)$ for all $s \in S\}$. The conjugate function $J^*_{\phi^*} : M(S) \to (-\infty, +\infty]$ is given by*

$$J^*_{\phi^*}(\mu) = \int_S \phi\left(\frac{d\mu_\alpha}{d\rho}(s)\right) d\rho + q\mu^+_\sigma(S) - p\mu^-_\sigma(S). \tag{2.8}$$

The proof of this result in the affine case is a straightforward calculation, while the case $p < q$ is a direct application of [46, Thm. 5] (see also [8, Thm. 3.1]).

COROLLARY 2.9. *Suppose $x \in L_1(S, \rho)$ and $0 \leqq \nu$, $\xi \in M(S)$. If $d\mu = x\, d\rho + d\nu - d\xi$ then $J^*_{\phi^*}(\mu) \leqq I_\phi(x) + q\nu(S) - p\xi(S)$ with equality if ρ, ν, and ξ are mutually singular.*

Proof. Let $\gamma := \nu - \xi$. Then $\nu \geqq \gamma^+$ and $\xi \geqq \gamma^-$ (see [48, p. 127]). By Theorem 2.7 and the definition of $\phi 0+$,

$$\begin{aligned} J^*_{\phi^*}(\mu) &= \int_S \phi\left(x(s) + \frac{d\gamma_\alpha}{d\rho}(s)\right) d\rho + q\gamma^+_\sigma(S) - p\gamma^-_\sigma(S) \\ &\leqq \int_S \left[\phi(x(s)) + (\phi 0+)\left(\frac{d\gamma_\alpha}{d\rho}(s)\right)\right] d\rho + q\gamma^+_\sigma(S) - p\gamma^-_\sigma(S) \\ &= I_\phi(x) + \int_S \left[q\frac{d\gamma^+_\alpha}{d\rho}(s) - p\frac{d\gamma^-_\alpha}{d\rho}(s)\right] d\rho + q\gamma^+_\sigma(S) - p\gamma^-_\sigma(S) \\ &= I_\phi(x) + q\gamma^+_\alpha(S) - p\gamma^-_\alpha(S) + q\gamma^+_\sigma(S) - p\gamma^-_\sigma(S) \\ &= I_\phi(x) + q\gamma^+(S) - p\gamma^-(S) \\ &\leqq I_\phi(x) + p(\nu(S) - \xi(S)) + (q - p)\nu(S). \end{aligned}$$

If ρ, ν, and ξ are mutually singular, then $\gamma_\alpha = 0$, so $d\gamma_\alpha / d\rho(s) = 0$ almost everywhere, $[\rho]$ on S, and $\nu = \gamma^+$ (by Hahn decomposition), so we have equality above. □

We now compute the subdifferential of J_{ϕ^*}. This will be fundamental in deriving optimality conditions.

THEOREM 2.10. *Suppose $z \in C(S)$ and $\mu \in M(S)$. Then $\mu \in \partial J_{\phi^*}(z)$ (or equivalently, $J_{\phi^*}(z) + J^*_{\phi^*}(\mu) = \int_S z(s)\, d\mu$) if and only if*

$$z(s) \in [p, q] \quad \textit{for all } s \in S,$$

$$\frac{d\mu_\alpha}{d\rho}(s) \in \partial\phi^*(z(s)) \quad \textit{a.e.}\ [\rho]\ \textit{on}\ S,$$

$$\text{support}\,(\mu_\sigma^+) \subset \{s \mid z(s) = q\},$$

and

$$\text{support}\,(\mu_\sigma^-) \subset \{s \mid z(s) = p\}.$$

Proof. We will assume that ϕ is not affine: the affine case, like Theorem 2.7, is a straightforward calculation, and we will not use this case in what follows. We assume, therefore, that $p < q$, and apply Corollary 5A of [46]. As in the proof of Theorem 2.7, we will apply Rockafellar's result with $D(s) := (p, q)$ for all s in S. Writing $\mathbb{R}_+$ for the nonnegative reals, the normal cone to cl $(D(s))$ is given by

$$N_{[p,q]}(v) = \begin{cases} -\mathbb{R}_+ & \text{if } v = p, \\ \{0\} & \text{if } v \in (p, q), \\ \mathbb{R}_+ & \text{if } v = q, \end{cases} \tag{2.11}$$

for v in $[p, q]$. Applying Rockafellar's result shows that $\mu \in \partial J_{\phi^*}(z)$ is equivalent to the first two statements along with μ_σ being $N_{[p,q]}$-valued: in other words (using the fact that $\mu_\sigma \ll |\mu_\sigma|$),

$$\frac{d\mu_\sigma}{d|\mu_\sigma|} \in N_{[p,q]}(z(s)) \quad \text{a.e. } [|\mu_\sigma|] \text{ on } S. \tag{2.12}$$

The remainder of the proof is reasonably straightforward measure theory (see [8, Thm. 3.5]). □

3. Primal and dual constraint qualifications. The optimization problem that we wish to consider is

$$\begin{aligned} &\inf && \int_S [\phi(x(s)) + z_0(s)x(s)]\, d\rho, \\ &\text{subject to} && \int_S a_i(s)x(s)\, d\rho = b_i \quad \text{for } i = 1, \ldots, n, \\ &&& x \in L_1(S, \rho), \end{aligned}$$

or in our previous notation,

$$(P) \qquad \inf\{I_\phi(x) + \langle z_0, x\rangle \mid Ax = b \text{ and } x \in L_1(S, \rho)\}.$$

The extra linear functional corresponding to z_0 in the objective is introduced to allow us to model some best entropy estimation problems where a prior estimate is given (see, for example, [28] and [29]), and to consider the log-barrier penalty function for semi-infinite linear programming [55]. We could consider the problem (P) posed in any of the spaces $L_r(S, \rho)$, for $1 \leqq r \leqq \infty$, or even in $C(S)$, but since $L_1(S, \rho)$ is the largest of these spaces, it is the natural choice if we wish to find an optimal solution.

Unfortunately, $L_1(S, \rho)$ is not typically a dual space, so we are unable to use weak-star compactness arguments to prove attainment. Furthermore, unless $p = -\infty$ and $q = +\infty$, the level sets of I_ϕ will not typically be weakly compact in L_1 (see [7]). In this case special arguments are needed to prove attainment, dependent on the underlying measure space (S, ρ) and the constraint map A (see, for example, [6]).

The idea of considering solutions to optimization problems in L_1 which may have singular components is not new. An example in optimal control appears in [4], and was extended in [41]. In this latter thesis the approach taken is to consider the problem in Fenchel form and then to solve the second dual. This gives a so-called "weak" solution (see [16, § III.6]).

For this reason we introduce the following "extended primal problem:"

$$(P_E) \qquad \inf\{J^*_{\phi^*}(\mu) + \langle z_0, \mu\rangle \mid B^*\mu = b \text{ and } \mu \in M(S)\}.$$

Using Corollary 2.9 we can rewrite this as

$$\begin{array}{lll} & \inf & \int_S [\phi(x(s)) + z_0(s)x(s)]\, d\rho + q\nu^+(S) - p\nu^-(S) + \int_S z_0(s)\, d\nu \\ (P_E^1) & \text{subject to} & \int_S a_i(s)x(s)\, d\rho + \int_S a_i(s)\, d\nu = b_i \quad \text{for } i = 1, \ldots, n, \\ & & x \in L_1(S, \rho), \quad \nu \in M(S), \quad \nu \perp \rho. \end{array}$$

Notice that (P_E^1) is exactly (P) if we require the singular component $\nu = 0$. Under reasonable conditions (P_E) will always have an optimal solution: as we shall see, the singular component corresponds with singularities observed in practice when (P) fails to have an optimal solution. In fact, Corollary 2.9 allows us to omit the constraint $\nu \perp \rho$ if so desired (see [8, Thm. 5.3]).

Our arguments are based on duality techniques. The dual problem for (P) (see [6]) is

$$(P^*) \qquad \sup\{b^T\lambda - I_{\phi^*}(A^*\lambda - z_0) \mid \lambda \in \mathbb{R}^n\},$$

which we may write as

$$(3.1) \qquad \sup\{b^T\lambda - J_{\phi^*}(B\lambda - z_0) \mid \lambda \in \mathbb{R}^n\},$$

or as

$$(3.2) \qquad \sup\left\{b^T\lambda - \int_S \phi^*(\lambda^T a(s) - z_0(s))\, d\rho \mid \lambda \in \mathbb{R}^n\right\}.$$

We denote the value of an optimization problem (Q) by $V(Q) \in [-\infty, +\infty]$. We say (Q) is *consistent* if there is a choice of the variable that satisfies the constraints and has finite objective value.

As usual, we have an easy weak duality result. The problems (P_E) and (P^*) (written in the form (3.1)) are Fenchel duals of each other, so a simple dual constraint qualification ensures that $V(P_E) = V(P^*)$ and $V(P_E)$ is attained (the motivation for its introduction). We will henceforth ignore the case where ϕ is affine, which is trivial.

Dual Constraint Qualification. The function ϕ is not affine, and there exists a $\hat{\lambda}$ in $\mathbb{R}^n$ with $\hat{\lambda}^T a(s) - z_0(s) \in (p, q)$ for all s in S.

Note that the assumption that ϕ is not affine ensures that $p < q$. If, as frequently occurs in practice, one of the a_i's is a nonzero constant function, $z_0 = 0$, and ϕ is not affine then the Dual Constraint Qualification will hold.

In order to ensure attainment in the dual problem (P^*) we need a primal constraint qualification. We recall from [9] that if x lies in a convex subset C of a topological vector space X, then x is a *quasi-relative interior* point of C ($x \in \operatorname{qri}(C)$) if cl (cone $(C-x)$) is a subspace.

We will write $[\alpha, \beta]_{L_1}$ for the order interval $\{x \in L_1 \mid \alpha \leqq x(s) \leqq \beta$ almost everywhere$\}$. The usual constraint qualification for (P) is written

$$(PCQ_1) \qquad b \in \operatorname{ri}(A \operatorname{dom}(I_\phi))$$

(see, for example, [43]). Since this condition may be difficult to check, we will rewrite it in a more familiar Slater-type form.

(PCQ_2) There exists $\tilde{x} \in \operatorname{qri}(\operatorname{dom}(I_\phi))$, which is feasible for (P).

This in turn can be stated in the following equivalent but more applicable form.

Primal Constraint Qualification. There exists a function $\hat{x}$ in $L_1(S, \rho)$ such that $\hat{x}(s) \in \operatorname{ri}(\operatorname{dom}(\phi))$ almost everywhere, and $A\hat{x} = b$.

(Of course, $\operatorname{ri}(\operatorname{dom}(\phi)) = (\alpha, \beta)$ unless ϕ is the indicator function of a point.) The following result may be found in [33].

LEMMA 3.3. *The Primal Constraint Qualification,* (PCQ_1), *and* (PCQ_2) *are equivalent. Furthermore,* $\operatorname{ri}(A \operatorname{dom}(I_\phi)) = \operatorname{ri}(A[\alpha, \beta]_{L_1})$, *and providing* $\alpha < \beta$,

$$\operatorname{aff}(A \operatorname{dom}(I_\phi)) = \operatorname{aff}(A[\alpha, \beta]_{L_1}) = \operatorname{Range}(A) = \operatorname{Range}(B^*).$$

If the constraint functions $a_1, \ldots, a_n$ are *pseudo-Haar*, or, in other words, linearly independent on every subset of S with positive measure (see [6]), then the Primal Constraint Qualification can be weakened to:

$$(PCQ_3) \qquad \begin{array}{l} \text{There exists an } \hat{x} \text{ in } L_1(S, \rho) \text{ with } A\hat{x} = b, \text{ and} \\ \rho\{s \in S \mid \alpha < \hat{x}(s) < \beta\} > 0. \end{array}$$

For a proof, see [33]. In summary, the Primal Constraint Qualification is easy to check in practice.

THEOREM 3.4 (duality). *$V(P) \geqq V(P_E) \geqq V(P^*)$. If the Dual Constraint Qualification holds, then $V(P_E) = V(P^*)$, and if, furthermore, (P_E) is consistent then $V(P_E)$ is attained. If, on the other hand, the Primal Constraint Qualification holds then $V(P) = V(P_E) = V(P^*)$, and if, furthermore, (P^*) is consistent then $V(P^*)$ is attained.*

Proof. The first claim (weak duality) is straightforward (see [8, Prop. 4.3]). Suppose the Dual Constraint Qualification holds. By Theorem 2.7, J_{ϕ^*} is finite and continuous at $B\hat{\lambda} - z_0$, where $\hat{\lambda}$ is the point in the Dual Constraint Qualification. Thus by [46, Thm. 3],

$$\begin{aligned} &\min\{J^*_{\phi^*}(\mu) + \langle z_0, \mu\rangle \mid B^*\mu = b, \text{ and } \mu \in M(S)\} \\ &\quad = \sup\{b^T\lambda - J_{\phi^*}(B\lambda - z_0) \mid \lambda \in \mathbb{R}^n\}, \end{aligned}$$

which is exactly the required result. If, on the other hand, the Primal Constraint Qualification holds, then $V(P) = V(P^*)$ by Corollary 2.6 of [6]. It follows by weak duality that $V(P) = V(P_E) = V(P^*)$. □

Our next step is to derive the optimality conditions. The proof is an easy application of weak duality and Theorem 2.10 (see [8, Thm. 4.10]).

$$(OCP_E) \qquad \begin{cases} \bar{\mu} \text{ is feasible for } (P_E), \text{ and} \\ \bar{\mu} \in \partial J_{\phi^*}(B\bar{\lambda} - z_0), \end{cases}$$

$$(OCP_E^1) \qquad \begin{cases} (\bar{x}, \bar{\nu}) \text{ is feasible for } (P_E^1), \\ \bar{x}(s) \in \partial\phi^*(\bar{\lambda}^T a(s) - z_0(s)) \quad \text{a.e. on } S, \\ \text{support}\,(\bar{\nu}^+) \subset \{s \in S \,|\, \bar{\lambda}^T a(s) - z_0(s) = q\}, \text{ and} \\ \text{support}\,(\bar{\nu}^-) \subset \{s \in S \,|\, \bar{\lambda}^T a(s) - z_0(s) = p\}, \end{cases}$$

$$(OCP) \qquad \begin{cases} \bar{x} \text{ is feasible for } (P), \text{ and} \\ \bar{x}(s) \in \partial\phi^*(\bar{\lambda}^T a(s) - z_0(s)) \quad \text{a.e. on } S. \end{cases}$$

THEOREM 3.5. (i) (OCP_E) *holds if and only if* $\bar{\mu}$ *is optimal for* (P_E) *and* $\bar{\lambda}$ *is optimal for* (P^*), *with equal objective value.*

(ii) (OCP_E^1) *holds if and only if* $(\bar{x}, \bar{\mu})$ *is optimal for* (P_E^1) *and* $\bar{\lambda}$ *is optimal for* (P^*), *with equal objective value.*

(iii) (OCP) *holds if and only if* $\bar{x}$ *is optimal for* (P) *and* $\bar{\lambda}$ *is optimal for* (P^*), *with equal objective value.*

COROLLARY 3.6 (strong duality). *Suppose that the Primal and Dual Constraint Qualifications hold. Then the two primal problems* (P) *and* (P_E) *(and* (P_E^1)*) and the dual problem* (P^*) *all have equal, finite value, and there exist optimal solutions* $\bar{\mu}$ *for* (P_E) *(and* $(\bar{x}, \bar{\nu})$ *for* (P_E^1)*), and* $\bar{\lambda}$ *for* (P^*), *satisfying* (OCP_E) *(or* (OCP_E^1), *respectively).*

Part (iii) of Theorem 3.5 is extremely instructive. In practice ϕ^* is usually differentiable, so the last condition of (OCP) becomes

$$\bar{x}(s) = (\phi^*)'(\bar{\lambda}^T a(s) - z_0(s)). \tag{3.7}$$

It has been a frequent error in the more practical literature to assume that if $\bar{\lambda}$ is dual optimal then (3.7) gives the optimal solution of the primal problem (P). The feasibility of this $\bar{x}$ is justified by differentiating under the integral in (3.2) with respect to λ. Unfortunately, as we shall see, in quite simple examples (satisfying the Primal and Dual Constraint Qualifications) the $\bar{x}$ given by (3.7) can lie in L_1 and yet *fail to be feasible.*

Theorem 3.6 shows that, under reasonable conditions, the $\bar{x}$ given by (3.7) corresponds to the *absolutely continuous part* of an optimal solution of the *extended* primal problem (P_E). It will be optimal for the original primal problem (P) if and only if it is *feasible.* If it fails to be feasible this is due to singular components of the optimal solution, supported on the set where $\bar{\lambda}^T a(s) - z_0(s)$ hits the boundary of the domain of ϕ^*. In principle, if this set is large, these singular components could be very unpleasant, making any practical application or interpretation impossible. In fact, we can generally restrict our attention to singular components consisting of finitely many point masses (see [8]).

For the time being we confine ourselves to interpreting the singular components in terms of primal optimizing sequences (cf. [16, Prop. III.6.1]). A standard argument (see [8, Thm. 4.13]) gives the following result.

THEOREM 3.8. *Suppose the sequence* $(x_r)_1^\infty$ *in* $L_1(S, \rho)$ *is an* optimizing sequence *for the primal problem* (P): $Ax_r \to b$ *and* $I_\phi(x_r) + \langle z_0, x_r \rangle \to V(P)$ *as* $r \to \infty$. *Suppose also that the Primal Constraint Qualification holds. Then the limit of any weak-star convergent subsequence of* $(x_r\,d\rho)_1^\infty$ *in* $M(S)$ *is optimal for the extended primal problem* (P_E).

Standard compactness arguments show that there will exist weak-star convergent subsequences in the above result if, for example, S is metrizable, $\phi(u) = +\infty$ for $u < 0$, and for some j, $a_j(s) > 0$ on S (see [8, Cor. 4.14]).

In [33] these results are applied to progressively refined discretizations of the primal problem: it is shown that the corresponding optimal solutions typically have weak-star convergent subsequences, any of which converge to an optimal solution of the extended problem. This provides another more concrete justification for considering this extension of the primal problem.

4. Primal attainment. As we saw in § 3, the existence of an optimal solution of the extended primal problem (P_E) (or any of its equivalent formulations) is a straightforward consequence of the Dual Constraint Qualification. By contrast, attainment in the original primal problem (P) is a much more delicate matter: as we shall see, there may fail to be an optimal solution in even very simple examples. The existence question depends not only on the function ϕ in the objective but also on the smoothness of the constraint functions $a_1, \dots, a_n$, on z_0, and on geometric and measure-theoretric properties of the underlying space (S, ρ). This question was addressed in [6], where the existence of an optimal solution was demonstrated in particular for classical (algebraic and trigonometric) moment problems with the Burg entropy as objective, when (S, ρ) is a one-dimensional interval with Lebesgue measure. This had been known previously for the trigonometric case (where the interval is $[-\pi, +\pi]$ and the moment conditions consist of the first n Fourier coefficients of x) using very special contour integral techniques [15], and for the two-dimensional trigonometric case in [56], and more generally in [32]. The approach of the latter two papers is a direct investigation of the map that takes a polynomial to the moments of its reciprocal. A contrasting, duality-based approach is taken in [36]: some technical difficulties remain, as discussed after Corollary 3.6.

In this section we will extend and clarify the results in [6] by using the results in § 3 on the existence of *extended* primal solutions. In particular, our new results will give an entirely rigorous proof that the Burg entropy also entails the existence of an optimal solution in the two-dimensional trigonometric case. By contrast, as we shall see, simple three-dimensional problems fail to have optimal solutions. The idea is very simple: given an extended primal solution $(\bar{x}, \bar{\nu})$, we need a condition to ensure, via Theorem 3.5, that the singular part $\bar{\nu}$ vanishes.

To summarize, the approach here has three substantial advantages over [6]. First, it is extremely natural, unlike the techniques in [6]. Second, it generalizes the results in [6] to other important practical cases. Third, it reveals exactly the sense in which existence can fail.

We begin with an informal discussion. Let us denote by $\Psi : \mathbb{R}^n \to (-\infty, +\infty]$ the function $I_{\phi^*}(A^*(\cdot) - z_0)$, so the dual problem (P^*) consists of minimizing the convex function $\Psi(\lambda) - b^T\lambda$. Suppose for simplicity that $a(s)$ is nonzero for every s in S. Then it is easily checked that the interior of the domain of the dual objective function is equal to

$$\operatorname{int}(\operatorname{dom}(\Psi)) = \{\lambda \in \mathbb{R}^n \mid \lambda^T a(s) - z_0(s) \in (p, q) \text{ for all } s \in S\}.$$

Suppose the Primal Constraint Qualification holds so there exists a dual optimum, say $\bar{\lambda}$, with $b \in \partial\Psi(\bar{\lambda})$. As is usual in convex analysis, the difficulties, if any, occur at the boundary of dom (Ψ), while if $\bar{\lambda} \in \operatorname{int}(\operatorname{dom}(\Psi))$ easy arguments identical to those that follow show the existence of a solution to the primal problem (P). Of course, this must be the case if dom (Ψ) is open. However, it will be true more generally, provided there are no boundary points in dom Ψ at which subgradients exist. The difficulty is in checking this, since boundary subgradients may exist even when ϕ^* is essentially smooth.

That is the origin of the following condition; we will work with it directly, but similar arguments show it implies that $\partial\Psi(\lambda)=\emptyset$ whenever $\lambda\notin \operatorname{int}(\operatorname{dom}(\Psi))$. This also has important computational consequences. In practice, (P) is generally solved via the dual, so we seek to minimize Ψ. When the condition below holds any minimizer must lie in $\operatorname{int}(\operatorname{dom}(\Psi))$. Thus we can apply unconstrained search techniques (appropriately safeguarded).

INTEGRABILITY CONDITION. *For any function* $z:=\lambda^Ta-z_0$ (*with* λ *in* $\mathbb{R}^n$), *if* $z(s)\in(p,q)$ *almost everywhere on* S *and* $(\phi^*)'(z(\cdot))\in L_1(S,\rho)$, *then it follows that* $z(s)\in(p,q)$ *for all* s *in* S *where* $a(s)$ *is nonzero.*

THEOREM 4.1. (i) *Suppose* ϕ *is essentially strictly convex. Then if* $(\bar{x}_1,\bar{\nu})$ *and* $(\bar{x}_2,\bar{\nu})$ *are both optimal for the extended primal problem* (P_E^1) *then* $\bar{x}_1=\bar{x}_2$, *so in particular the original primal problem* (P) *has at most one solution.*

(ii) *Let us suppose furthermore that the Primal Constraint Qualification holds. Then the dual problem* (P^*) *has an optimal solution, and if* $(\bar{x}.\bar{\nu})$ *is optimal for* (P_E^1) *and* $\bar{\lambda}$ *is optimal for* (P^*), *then*

$$\bar{x}(s)=(\phi^*)'(\bar{\lambda}^Ta(s)-z_0(s))\quad a.e.\ on\ S; \tag{4.2}$$

so, in particular, if (P) *has an optimal solution it is given uniquely by* (4.2).

(iii) *Moreover, suppose also that the Dual Constraint Qualification holds. Then* (P_E^1) *has an optimal solution* $(\bar{x},\bar{\nu})$ *with the absolutely continuous part given uniquely by* (4.2).

(iv) *If, in addition, the Integrability Condition holds, then the singular part* $\bar{\nu}$ *vanishes, so* (4.2) *gives the unique optimal solution of* (P).

Proof. Part (i) follows by strict convexity.

Parts (ii) and (iii) follow by Theorem 3.5 and (2.4). Assume finally that $(\bar{x},\bar{\nu})$ is optimal for (P_E^1) with $\bar{x}$ given by (4.2), and suppose the Integrability Condition holds. If we write $S_0:=\{s\in S\,|\,a(s)=0\}$ then, from (OCP_E^1), $\bar{\nu}$ is supported on S_0, and by the Dual Constraint Qualification, $-z_0(s)\in(p,q)$ for all s in S_0.

But now $(\bar{x},0)$ is also feasible for (P_E^1), with a corresponding drop in the objective value of

$$q\bar{\nu}^+(S)-p\bar{\nu}^-(S)+\int_S z_0(s)\,d\bar{\nu}=\int_{S_0}(q+z_0(s))\,d\bar{\nu}^+-\int_{S_0}(p+z_0(s))\,d\bar{\nu}^->0,$$

unless $\bar{\nu}=0$. Hence the result. □

The Integrability Condition actually turns out to be necessary, as well as sufficient, for the existence of a primal solution in general. That is the substance of the next result.

THEOREM 4.3. *Suppose* ϕ *is of Legendre type and the Integrability Condition fails. Then there exists a right-hand side* b *in* $\mathbb{R}^n$ *such that the primal problem* (P) *satisfies the Primal Constraint Qualification, but has no optimal solution.*

Proof. Since the Integrability Condition fails, there exists a function $\bar{z}:=\bar{\lambda}^Ta-z_0$ satisfying $\bar{z}(s)\in(p,q)$ almost everywhere and with $\bar{x}(\cdot):=(\phi^*)'(\bar{z}(\cdot))$ in $L_1(S,\rho)$, but with $S_1:=\{s\in S\,|\,a(s)\neq 0,\ \bar{z}(s)=p \text{ or } q\}$ nonempty. Define $\bar{b}:=A\bar{x}$. Note that $\alpha<\bar{x}(s)<\beta$ almost everywhere by Lemma 2.6. Thus $\bar{b}\in\operatorname{ri}(A\operatorname{dom}(I_\phi))$ by Lemma 3.3.

Now choose any ν in $M(S)$, with

$$\operatorname{support}(\nu^+)\subset\{s\in S\,|\,a(s)\neq 0,\ \bar{z}(s)=q\},$$
$$\operatorname{support}(\nu^-)\subset\{s\in S\,|\,a(s)\neq 0,\ \bar{z}(s)=p\},$$

and $B^*\nu=\int_S a\,d\nu\neq 0$. For example, a point mass at any point of S (with the appropriate sign) will do. It follows by Lemma 3.3 that

$$b:=\bar{b}+\varepsilon B^*\nu\in\operatorname{ri}(A\operatorname{dom}(I_\phi)),$$

provided that $\varepsilon>0$ is sufficiently small.

Clearly now, the Primal Constraint Qualification holds (Lemma 3.3). Furthermore, if we write $\bar{\nu} := \varepsilon\nu$, then $(\bar{x}, \bar{\nu})$ and $\bar{\lambda}$ satisfy (OCP_E^1) and thus are optimal for (P_E^1) and (P^*), respectively, so $\bar{x}$ is the only possible optimal solution of (P), by Theorem 3.5(ii) and (iii). However, $\bar{x}$ is not feasible for (P), since $b \neq \bar{b}$. □

We now pursue a slight digression, to discuss the approach of [56] and [32]. We will show that their key supporting result, which is of some independent interest, can be subsumed by this approach. The idea of Woods, and Lang and McClellan (working in the special case where the a_i's are multidimensional trigonometric polynomials and ϕ is the Burg entropy) is to consider the nonlinear system of equations in $\lambda \in \mathbb{R}^n$ derived (formally in these references but rigorously above) from the optimality conditions (OCP):

$$(NLE) \qquad \int_S a_i(s)(\phi^*)'(\lambda^T a(s) - z_0(s))\, d\rho = b_i \quad \text{for } i = 1, \dots, n.$$

Assuming the existence of a dual optimal λ (a difficulty not addressed in the above papers), the primal optimal solution $\bar{x}$ (if it exists) must have the form $(\phi^*)'(\lambda^T a(s) - z_0(s))$, so it may be obtained by solving (NLE) for λ.

Assuming ϕ is of Legendre type it is clear, as in the proof of Theorem 4.3, that (NLE) is certainly not solvable unless $b \in \mathrm{ri}(A[\alpha, \beta]_{L_1})$. The point (obvious from an optimization viewpoint but surprising ab initio) is that the Integrability Condition gives a complete characterization.

COROLLARY 4.4. *Suppose ϕ is of Legendre type and the Dual Constraint Qualification holds. Then (NLE) is solvable for every b in* $\mathrm{ri}(A[\alpha, \beta]_{L_1})$ *if and only if the Integrability Condition holds.*

Proof. The first direction follows from Theorem 4.3 and the comments above. The converse follows from Theorem 4.1. □

Taking ϕ to be the Burg entropy and the a_i's as (multidimensional) trigonometric polynomials, we obtain the result in the Appendix of [32].

These results demonstrate the importance of the Integrability Condition for the question of attainment in the original primal problem. The remainder of this section will be devoted to investigating for what spaces (S, ρ), objectives ϕ and z_0, and constraints a it holds. We shall see that the important features are the local geometry of the set S, and the growth rate of $(\phi^*)'$ near p and q. We adopt an approach which gives unified conditions for the cases of common interest, namely, $S \subset \mathbb{R}^m$ for $m = 1, 2, 3$. We shall suppose for the remainder of this section that S is a compact metric space with metric $d(\cdot, \cdot)$, and we write $B(s, r)$ for the open ball, centre s, radius r. For any s in S we define $\chi_s(r) := \rho(S \cap B(s, r))$. The following result is derived from an elementary estimate of the integral in the Integrability Condition (see [8, Thm. 6.6]).

THEOREM 4.5. *Suppose S is a compact metric space, $a_1, \dots, a_n$ and z_0 are Lipschitz on S, ϕ is essentially strictly convex, and the following two conditions hold for any s_0 in S and $k > 0$:*

$$\lim_{\delta \downarrow 0} \inf_{0 < \varepsilon \leq r \leq \delta} r[(1/\varepsilon)(\chi_{s_0}(r + \varepsilon) - \chi_{s_0}(r))][(\phi^*)'(q - kr)] > 0, \quad \textit{if } q < +\infty;$$

$$\lim_{\delta \downarrow 0} \inf_{0 < \varepsilon \leq r \leq \delta} (-r)[(1/\varepsilon)(\chi_{s_0}(r + \varepsilon) - \chi_{s_0}(r))][(\phi^*)'(p + kr)] > 0, \quad \textit{if } p > -\infty.$$

Then the Integrability Condition holds.

In practice S is often a compact subset of $\mathbb{R}^m$ with Lebesgue measure, and in this case we can often simplify the required conditions. The *Dubovitskij-Miljutin* (DM) *cone* will be useful in what follows (see, for example, [2]).

DEFINITION 4.6. For a subset K of a normed space V, and for s in cl K, we define $D_K(s) := \{v \in V \mid s + (0, \varepsilon]B(v, \varepsilon) \subset K$ for some $\varepsilon > 0\}$. We say that K is DM-*regular* if $D_K(s)$ is nonempty for all s in K.

The condition of DM-regularity ensures that the sets in which we are interested have no cusps. In a normed space it is easily checked that any convex set with nonempty interior is DM-regular. A subset of a normed space defined by inequalities will be DM-regular providing a suitable constraint qualification holds everywhere (see [2, p. 126], for example). Obviously, an arbitrary union of DM-regular sets is DM-regular.

Let us denote m-dimensional Lebesgue measure by τ_m. Using the fact that an open convex cone must intersect the surface of the unit sphere with positive area, we obtain the following (see [8, Lemma 6.11]).

LEMMA 4.7. *Suppose* $S \in \mathbb{R}^m$ (*with Euclidean distance*) *is compact and* DM-*regular, and for some* $k > 0$, $\rho \geqq k\tau_m$ *on* S. *Then for any* s_0 *in* S,

$$\lim_{\delta \downarrow 0} \inf_{0 < \varepsilon \leqq r \leqq \delta} r^{1-m}[(1/\varepsilon)(\chi_{s_0}(r+\varepsilon) - \chi_{s_0}(r))] > 0. \tag{4.8}$$

We can now derive a more useful version of Theorem 4.5.

THEOREM 4.9. *Suppose that S is a compact,* DM-*regular subset of* $\mathbb{R}^m$, *and* ρ *dominates a positive multiple of Lebesgue measure on S. Suppose that* $a_1, \ldots, a_n$ *and* z_0 *are Lipschitz on S. Finally, suppose that* ϕ *is essentially strictly convex, with* $\underline{\lim}_{r\downarrow 0}\, r^m(\phi^*)'(q-r) > 0$ *if* $q < +\infty$, *and if* $p > -\infty$, $\underline{\lim}_{r \downarrow 0} - r^m(\phi^*)'(p+r) > 0$. *Then the Integrability Condition holds, so if in addition the Primal and Dual Constraint Qualifications hold, the original primal problem* (P) *has a unique optimal solution.*

Proof. By Lemma 4.7, for any s_0 in S, (4.8) holds. Now for any $k > 0$, if $q < +\infty$,

$$\lim_{\delta \downarrow 0} \inf_{0<\varepsilon \leqq r \leqq \delta} r[(1/\varepsilon)(\chi_{s_0}(r+\varepsilon) - \chi_{s_0}(r))][(\phi^*)'(q-kr)]$$

$$\geqq \left\{\lim_{\delta \downarrow 0} \inf_{0<\varepsilon \leqq r \leqq \delta} r^{1-m}[(1/\varepsilon)(\chi_{s_0}(r+\varepsilon) - \chi_{s_0}(r))]\right\}\left\{\lim_{\delta \downarrow 0} \inf_{0<r \leqq \delta} r^m(\phi^*)'(q-r)\right\}$$

(since both factors are nonnegative). The first factor is strictly positive by (4.8), and the second is strictly positive by assumption. The first condition of Theorem 4.5 follows, and a similar argument shows the second condition. The result now follows from Theorem 4.5. □

Probably the most important application of this result is when S is a compact interval of $\mathbb{R}$ and ϕ is the Burg entropy (1.3). In particular, we obtain the original existence result of [15].

The periodic case. In many cases in practice the moment conditions are given by Fourier coefficients. In other words, the constraint functions $a_1, \ldots, a_n$ are trigonometric polynomials (possibly multidimensional) and hence periodic. In these cases it is often possible to weaken the conditions for attainment in the original problem.

DEFINITION 4.10. Suppose $e^1, \ldots, e^m$ form a basis of $\mathbb{R}^m$. With respect to this basis, we say $S \subset \mathbb{R}^m$ is *covering* if $\bigcup_{\theta \in \mathbb{Z}^m} (S + \sum_{j=1}^m \theta_j e^j) = \mathbb{R}^m$. We say a function $z: S \to \mathbb{R}$ is *periodic* if $z(s) = z(s')$ whenever $s - s' = \sum_{j=1}^m \theta_j e^j$ for some θ in $\mathbb{Z}^m$.

The idea is to use the following simple result [8, Lemma 6.16].

LEMMA 4.11. *Suppose* $z: S \to \mathbb{R}$ *is periodic, with* ∇z *Lipschitz on S, and suppose S is covering. Suppose* $z(s) \leqq q$ *for all s in S and* $z(s_0) = q$. *Then for some* $k_1 > 0$, $z(s) \geqq q - k_1\|s - s_0\|^2$ *for all s in S.*

Using this to estimate $(\phi^*)'(z(s))$ we arrive at the following refinement of Theorem 4.9.

THEOREM 4.12. *Suppose that S is a compact,* DM-*regular, covering subset of* $\mathbb{R}^m$, *and* ρ *dominates a positive multiple of Lebesgue measure on S. Suppose that* $\nabla a_1, \ldots, \nabla a_n$

and ∇z_0 are Lipschitz on S and $a_1, \ldots, a_n$ and z_0 are periodic. Finally, suppose that ϕ is essentially strictly convex, with $\underline{\lim}_{r\downarrow 0}\, r^m(\phi^*)'(q-r^2)>0$ if $q<+\infty$, and $\underline{\lim}_{r\downarrow 0} -r^m(\phi^*)'(p+r^2)>0$ if $p>-\infty$. Then the Integrability Condition holds, so if in addition the Primal and Dual Constraint Qualifications hold, the original primal problem (P) has a unique optimal solution.

Again, probably the most important application of the above result is when ϕ is the Burg entropy and $S=[-\pi, \pi]^2$ with trigonometric polynomials $a_1, a_2, \ldots, a_n, z_0$. In particular, we obtain the existence result in [56].

Theorems 4.9 and 4.12 involve growth conditions on $(\phi^*)'$. It is easy to translate these into conditions on ϕ', if so desired, using Lemma 2.6 (see [8, Lemma 6.19]).

5. Computation, primal uniqueness, and examples. In this section we will discuss how to solve the extended primal problem (P_E), and give conditions ensuring it has a unique solution. Suppose that the Primal and Dual Constraint Qualifications hold and that ϕ is essentially strictly convex, so ϕ^* is essentially smooth. From the Strong Duality Theorem (Corollary 3.6) we know that the dual problem (P^*) has an optimal solution $\bar{\lambda}$, and (P_E^1) has an optimal solution $(\bar{x}, \bar{\nu})$, where the absolutely continuous part $\bar{x}$ is given uniquely by

$$\bar{x}(s)=(\phi^*)'(\bar{\lambda}^T a(s)-z_0(s)) \tag{5.1}$$

(by Theorem 4.1), and the singular part $\bar{\nu}$ can be chosen arbitrarily, provided that it satisfies the optimality conditions (OCP_E^1). In order to compute a solution we first solve the dual problem. This is a concave maximization problem, and the objective function is continuously differentiable on the interior of its domain, so a wide variety of standard numerical techniques may be applied.

The continuous part of the primal solution $\bar{x}$ is now given by (5.1), while the singular part is any measure $\bar{\nu}$ which is singular with respect to ρ and satisfies

$$\text{support}\,(\bar{\nu}^+)\subset\{s\in S|\bar{\lambda}^T a(s)-z_0(s)=q\}, \tag{5.2}$$

$$\text{support}\,(\bar{\nu}^-)\subset\{s\in S\,|\,\bar{\lambda}^T a(s)-z_0(s)=p\}, \text{and} \tag{5.3}$$

$$\int_S a_i(s)\, d\nu=b_i-\int_S a_i(s)\bar{x}(s)\, d\rho \quad \text{for } i=1, \ldots, n. \tag{5.4}$$

(We know there exists a solution.) It can be shown using techniques analogous to those used in semi-infinite programming (see, for example, [1]) that we can restrict attention to $\bar{\nu}$ for which $\bar{\nu}^+$ and $\bar{\nu}^-$ are supported on $n+1$ points in S (see [8, § 5]). Conditions (5.2)-(5.4) then form a semi-infinite linear problem for which standard numerical techniques are available (see, for example, [23]).

The idea of a Tchebycheff system will be useful for our discussion of uniqueness. Working on a fixed, finite interval S in $\mathbb{R}$, for a continuous function f we denote by $\tilde{Z}(f)$ the number of distinct zeros of f, counting twice the zeros in the interior of S at which f does not change sign. The following result [30, Thm. I.4.2] essentially characterizes Tchebycheff systems.

THEOREM 5.5. *If $\{a_1, \ldots, a_n\}$ is a Tchebycheff system on S then $\tilde{Z}(\lambda^T a)\leqq n-1$, provided that λ is nonzero.*

COROLLARY 5.6. *Suppose $\{a_1, \ldots, a_n\}$ is a Tchebycheff system on S, $a_1\equiv 1, p\leqq \lambda^T a(s)\leqq q$ for all s in S, and $\lambda^T a$ is not identically p or q. Then we have $|\{s\in S\,|\,\lambda^T a(s)=p$ or $q\}|\leqq n$.*

Proof. Denote the number of endpoints of S at which $\lambda^T a(s)=p$ or q by n_p^e and n_q^e, respectively, and the number of interior points by n_p^i and n_q^i, respectively. Then

$n_p^e + n_q^e \leqq 2$, and by Theorem 5.5,

$$n_p^e + 2n_p^i = \tilde{Z}(\lambda^T a - p) \leqq n-1,$$

and

$$n_q^e + 2n_q^i = \tilde{Z}(q - \lambda^T a) \leqq n-1.$$

Adding gives $2(n_p^e + n_p^i + n_q^e + n_q^i) \leqq 2n$, from which the result follows. □

THEOREM 5.7. *Suppose the Primal and Dual Constraint Qualifications hold, ϕ is essentially strictly convex, S is a finite, closed interval in $\mathbb{R}$, $\{a_1, \dots, a_n\}$ is a Tchebycheff system on S, $a_1 \equiv 1$, and $z_0 \equiv 0$. Then the extended primal problem (P_E) (or (P_E^1)) has a unique optimal solution.*

Proof. If $\bar\lambda$ is a dual optimal solution then, from the above discussion, the absolutely continuous part of any extended primal optimal solution $(\bar x, \bar\nu)$ is given uniquely by $\bar x := (\phi^*)'(\bar\lambda^T a)$, and since ϕ^* is essentially smooth, $\bar\lambda^T a$ is not identically p or q. The singular part $\bar\nu$ must satisfy (5.2)-(5.4), and Corollary 5.6 shows that it is supported on at most n points, determined by $\bar\lambda$. The set of linear equations resulting from (5.4) then has a unique solution for $\bar\nu$ since $\{a_1, \dots, a_n\}$ is a Tchebycheff system. □

Analogous results could be proved when $\{a_1, \dots, a_n\}$ is a periodic Tchebycheff system, as in the trigonometric moment problem.

Examples. We begin by discussing two examples from [6]. The first is a simple semi-infinite linear program, where $\phi(u) := u$ if $u \geqq 0$ and $+\infty$ otherwise (note this is not an affine function):

$$(E1) \qquad \begin{array}{ll} \inf & \displaystyle\int_0^1 x(s)\,ds, \\ \text{subject to} & \displaystyle\int_0^1 sx(s)\,ds = 1, \\ & 0 \leqq x \in L_1[0,1]. \end{array}$$

The dual problem is

$$(E1^*) \qquad \sup\left\{\lambda - \int_0^1 \delta(\lambda \mid (-\infty, 1])\,ds \,\middle|\, \lambda \in \mathbb{R}\right\},$$

where $\delta(\cdot \mid C)$ is the indicator function of C. The Primal and Dual Constraint Qualifications are both satisfied, the unique dual optimal solution is $\bar\lambda = 1$, and both problems have value 1, but the primal value is not attained. The extended primal problem is

$$(E1_E) \qquad \begin{array}{ll} \inf & \displaystyle\int_0^1 x(s)\,ds + \nu[0,1], \\ \text{subject to} & \displaystyle\int_0^1 sx(s)\,ds + \int_0^1 s\,d\nu = 1, \\ & 0 \leqq x \in L_1[0,1], \quad 0 \leqq \nu \in M[0,1], \quad d\nu \perp ds, \end{array}$$

and our results show that the unique optimal solution is a unit point mass at 1, giving the value 1.

The second example uses the objective function $\phi(u) := 1/u$ if $u > 0$ and $+\infty$ otherwise:

$$(E2)\qquad \begin{array}{ll} \inf & \displaystyle\int_0^{2\pi} (1/x(s))\,ds, \\ \text{subject to} & \displaystyle\int_0^{2\pi} \sin(s)x(s)\,ds = 1, \\ & 0 \leqq x \in L_1[0, 2\pi]. \end{array}$$

The dual problem is

$$(E2^*)\qquad \sup\left\{\lambda - \int_0^{2\pi} \phi^*(\lambda \sin(s))\,ds \,\middle|\, \lambda \in \mathbb{R}\right\},$$

where $\phi^*(v) = -2(-v)^{1/2}$ if $v \leqq 0$ and $+\infty$ otherwise. The only dual feasible solution is $\bar{\lambda} = 0$, which is therefore optimal, with value zero. Note that, although the Primal Constraint Qualification is satisfied, the value of the primal problem is zero and is unattained. Furthermore, the extended primal problem (not considered in [6]) is

$$(E2_E)\qquad \begin{array}{ll} \inf & \displaystyle\int_0^{2\pi} (1/x(s))\,ds, \\ \text{subject to} & \displaystyle\int_0^{2\pi} \sin(s)x(s)\,ds + \int_0^{2\pi} \sin(s)\,d\nu = 1, \\ & 0 \leqq x \in L_1[0, 2\pi], \quad 0 \leqq \nu \in M[0, 2\pi], \quad d\nu \perp ds, \end{array}$$

and this problem also does not attain its value of zero. The reason is, of course, that the Dual Constraint Qualification is not satisfied.

The final two examples are particularly interesting since the objective function is the Burg entropy, which is widely used in practice. The first problem is extremely simple, and demonstrates the importance of the assumption that the constraint functions are Lipschitz in Theorem 4.9. We consider the primal problem

$$(E3)\qquad \begin{array}{ll} \inf & \displaystyle\int_0^{1} -\log(x(s))\,ds, \\ \text{subject to} & \displaystyle\int_0^{1} x(s)\,ds = 1, \\ & \displaystyle\int_0^{1} s^{1/2}x(s)\,ds = \alpha, \\ & 0 \leqq x \in L_1[0, 1], \end{array}$$

where $\alpha \in (0, 1)$. The dual problem is

$$(E3^*)\qquad \begin{array}{ll} \sup & \lambda_0 + \alpha\lambda_1 + \displaystyle\int_0^{1} [1 + \log(-\lambda_0 - \lambda_1 s^{1/2})]\,ds, \\ \text{subject to} & -\lambda_0 \leqq 0, \quad \lambda_0 + \lambda_1 \leqq 0, \\ & \lambda_0, \lambda_1 \in \mathbb{R} \end{array}$$

(where the extra constraint is implicit in the objective function).

It is straightforward to check that both the Primal and Dual Constraint Qualifications hold. We would expect, from the form of the constraints in ($E3$), that the weight in the optimal density will shift from right to left in the interval $[0,1]$ as we decrease α in $(0,1)$, and this is indeed what happens. We know from (4.2) that the absolutely continuous part of any extended primal solution is given by $\bar{x}(s):=(-\bar{\lambda}_0-\bar{\lambda}_1 s^{1/2})^{-1}$, where $\bar{\lambda}_0$ and $\bar{\lambda}_1$ are dual optimal, and it may be checked that for α in $(\frac{1}{2},1)$, $\bar{\lambda}_0+\bar{\lambda}_1 s^{1/2}>0$ on $[0,1]$, so $\bar{x}$ is the unique primal solution. At $\alpha=\frac{2}{3}$ the optimal solution $\bar{x}(s)\equiv 1$, and as α decreases the weight shifts to the left until at $\alpha=\frac{1}{2}$ the unique optimal solution is $\bar{x}(s)=(1/2)s^{-1/2}$. For α in $(0,\frac{1}{2}]$ it can be checked that the dual optimum is $\bar{\lambda}_0=0$, $\bar{\lambda}_1=-\alpha^{-1}$, and $\bar{x}$ is no longer feasible for $(E3)$.

What has happened is that part of the optimal solution has condensed into a point mass at the origin, as would be shown by discretization. The extended primal problem is

$$(E3_E)\qquad \begin{array}{ll} \inf & \displaystyle\int_0^1 -\log(x(s))\,ds, \\[2ex] \text{subject to} & \displaystyle\int_0^1 x(s)\,ds+\nu[0,1]=1, \\[2ex] & \displaystyle\int_0^1 x(s)s^{1/2}\,ds+\int_0^1 s^{1/2}\,d\nu=\alpha, \\[2ex] & 0\leqq x\in L_1[0,1],\quad 0\leqq \nu\in M[0,1],\quad d\nu\perp ds, \end{array}$$

and this has a unique optimal solution for α in $[0,\frac{1}{2}]$, $\bar{x}(s)=\alpha s^{-1/2}$, and $\bar{\nu}$ a point mass of $(1-2\alpha)$ at the origin.

The last example was presented in [40] to demonstrate the problems associated with the Burg entropy for three-dimensional density reconstruction, and it has also been discussed in [52] and [14]. The underlying set S is the unit cube in $\mathbb{R}^3$, $[0,1]^3$, with Lebesgue measure ds, and the problem has simple trigonometric moment constraints:

$$(E4)\qquad \begin{array}{ll} \inf & \displaystyle\int_S -\log(x(s))\,ds, \\[2ex] \text{subject to} & \displaystyle\int_S x(s)\,ds=1, \\[2ex] & \displaystyle\int_S x(s)\cos(2\pi s_i)\,ds=\alpha,\quad \text{for } i=1,2,3, \\[2ex] & 0\leqq x\in L_1(S), \end{array}$$

where $\alpha\in[0,1)$. The dual problem is

$$(E4^*)\qquad \begin{array}{ll} \sup & \displaystyle\lambda_0+\alpha\sum_1^3\lambda_i+\int_S\left[1+\log\left(-\lambda_0-\sum_1^3\lambda_i\cos(2\pi s_i)\right)\right]ds, \\[2ex] \text{subject to} & \displaystyle -\lambda_0\geqq\sum_1^3|\lambda_i|, \\[2ex] & \lambda_0,\lambda_1,\lambda_2,\lambda_3\in\mathbb{R} \end{array}$$

(where again the extra constraint is implicit in the objective function).

Straightforward calculations will now verify the following assertions. The Primal and Dual Constraint Qualifications both hold, and the unique dual optimal solution has the form $(\bar{\lambda}_0(\alpha), \bar{\lambda}(\alpha), \bar{\lambda}(\alpha), \bar{\lambda}(\alpha))$ for each α. Thus the absolutely continuous part $\bar{x}_\alpha$ of any extended primal solution $(\bar{x}_\alpha, \bar{\nu}_a)$ is given uniquely (see (4.2)) by

$$\bar{x}_\alpha(s) := \left(-\bar{\lambda}_0(\alpha) - \bar{\lambda}(\alpha)\sum_1^3 \cos(2\pi s_i)\right)^{-1}. \tag{5.8}$$

The interesting phenomenon is to observe what happens as α increases. The trigonometric polynomial in (5.8) is strictly positive for small α, and $\bar{x}_\alpha$ is feasible for the primal problem $(E4)$, as is the unique optimal solution. As α approaches a certain critical value $\bar{\alpha}$, the minimum value of the polynomial decreases to zero, until the point when $\alpha = \bar{\alpha}$, where the polynomial has a zero when $s_i = 0$ or 1 for $i = 1, 2, 3$. The unique optimal solution of the primal is still $\bar{x}_{\bar{\alpha}}$. However, as α increases past $\bar{\alpha}$ the character of the solution changes. For $\alpha \in (\bar{\alpha}, 1)$ the unique dual optimal solution is $(1/[1-\alpha]) \times (-1, \frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, so (5.8) becomes

$$\bar{x}_\alpha(s) = \frac{1-\alpha}{(1-\frac{1}{3}\sum_1^3 \cos(2\pi s_i))}, \tag{5.9}$$

which is no longer primal feasible.

The extended primal problem is

$$(E4_E)\qquad \begin{aligned} &\inf && \int_S -\log(x(s))\,ds, \\ &\text{subject to} && \int_S x(s)\,ds + \nu(S) = 1, \\ &&& \int_S x(s)\cos(2\pi s_i)\,ds + \int_S \cos(2\pi s_i)\,d\nu = \alpha \quad \text{for } i = 1, 2, 3, \\ &&& 0 \leqq x \in L_1(S), \quad 0 \leqq \nu \in M(S), \quad d\nu \perp ds. \end{aligned}$$

Our results show that the absolutely continuous part of the optimal solution is given by (5.9), and the optimality conditions ensure that the singular part $\bar{\nu}_\alpha$ is supported on the zeros of the denominator of (5.9), namely, $s_i = 0$ or 1 for $i = 1, 2, 3$. These points are equivalent up to periodicity, so essentially the unique singular part is a point mass at the origin with weight $((\alpha - \bar{\alpha})/(1 - \bar{\alpha}))$.

The critical value of α is given by

$$\bar{\alpha} = 1 - \left(\int_S \left(1 - \tfrac{1}{3}\sum_1^3 \cos(2\pi s_i)\right)^{-1} ds\right)^{-1} \approx .34. \tag{5.10}$$

(This integral is actually Green's integral for the cubic lattice, and has the closed form $\Gamma(1/24)\Gamma(5/24)\Gamma(7/24)\Gamma(11/24)(6)^{1/2}/32\pi^3$; see [24].) In the case discussed in [52] and [40] an optimal solution was proposed informally for the case $\alpha = .5$; our solution agrees exactly.

As a final comment, numerous different measures of entropy ϕ have appeared in the literature. A survey of some of these, with their conjugates and associated p and q, may be found in [6].

REFERENCES

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

[2] J.-P. AUBIN AND H. FRANKOWSKA, *Set-Valued Analysis*, Birkhäuser, Boston, 1990.

[3] A. BEN-TAL, J. M. BORWEIN, AND M. TEBOULLE, *A dual approach to multi-dimensional L_p spectral estimation problems*, SIAM J. Control Optim., 26 (1988), pp. 985-996.

[4] M.-F. BIDAUT, *Un problème de contrôle optimal à fonction coût en norme L_1*, Comptes Rend. Acad. Sci. Paris, Sér. A, 281 (1975), pp. 273-276.

[5] J. M. BORWEIN AND A. S. LEWIS, *Convergence of best entropy estimates*, SIAM J. Optim., 1 (1991), pp. 191-205.

[6] ———, *Duality relationships for entropy-like minimization problems*, SIAM J. Control Optim., 29 (1991), pp. 325-338.

[7] ———, *On the convergence of moment problems*, Trans. Amer. Math. Soc., 325 (1991), pp. 249-271.

[8] ———, *Partially finite programming in L_1: Entropy maximization*, Tech. Rep. CORR 91-05, Dept. of Combinatorics and Optimization, Univ. of Waterloo, Ontario, Canada, 1991.

[9] ———, *Partially finite convex programming*, Math. Programming, 57 (1992), pp. 15-83 (in two parts).

[10] J. M. BORWEIN AND H. WOLKOWICZ, *A simple constraint qualification in infinite dimensional programming*, Math. Programming, 35 (1986), pp. 83-96.

[11] P. BORWEIN AND A. S. LEWIS, *Moment-matching and best entropy estimation.* Tech. Rep. CORR 91-03, Univ. of Waterloo, Ontario, Canada, 1991; J. Math. Anal. Appl., submitted.

[12] J. P. BURG, *Maximum entropy spectral analysis*, Paper presented at 37th Meeting of the Society of Exploration Geophysicists, Oklahoma City, OK, 1967.

[13] D. DACUNHA-CASTELLE AND F. GAMBOA, *Maximum d'entropie et problème des moments*, Ann. Inst. Henri Poincaré, 26 (1990), pp. 567-596.

[14] A. DECARREAU, D. HILHORST, C. LEMARÉCHAL, AND J. NAVAZA, *Dual methods in entropy maximization: Application to some problems in crystallography*, SIAM J. Optim., 2 (1992), pp. 173-197.

[15] J. A. EDWARD AND M. M. FITELSON, *Notes on maximum entropy processing*, IEEE Trans. Inform. Theory, 19 (1973), pp. 232-234.

[16] I. EKELAND AND R. TEMAN, *Convex Analysis and Variational Problems*, North-Holland, Amsterdam, 1976.

[17] G. J. ERICKSON AND C. R. SMITH, EDS., *Maximum-Entropy and Bayesian Methods in Science and Engineering*, Vols. I and II, Kluwer, Dordrecht, The Netherlands, 1988.

[18] B. FORTE, W. HUGHES, AND Z. PALES, *Maximum entropy estimators and the problem of moments*, Rendiconti Mat., Ser. VII, 9 (1989), pp. 689-699.

[19] F. GAMBOA, *Methode du Maximum d'Entropie sur la Moyenne et Applications*, Ph.D. thesis, Univ. Paris Sud, Centre d'Orsay, France, 1989.

[20] F. GAMBOA AND E. GASSIAT, *Extension of the maximum entropy method on the mean and a Bayesian interpretation of the method*, preprint, Statistics Laboratory, Univ. of Orsay, France, 1991.

[21] ———, M.E.M. *techniques for solving moment problems*, preprint, Statistics Laboratory, Univ. of Orsay, France, 1991.

[22] E. GASSIAT, *Probléme sommatoire par maximum d'entropie*, Comptes Rend. Acad. Sci. Paris Sér. I, 303 (1986), pp. 675-680.

[23] K. GLASHOFF AND S.-A. GUSTAFSON, *Linear Optimization and Approximation*, Springer-Verlag, New York, 1983.

[24] M. L. GLASSER AND I. J. ZUCKER, *Extended Watson integrals for the cubic lattices*, Proc. Nat. Acad. Sci., USA, 74 (1977), pp. 1800-1801.

[25] R. K. GOODRICH AND A. STEINHARDT, *L_2 spectral estimation*, SIAM J. Appl. Math., 46 (1986), pp. 417-428.

[26] L. D. IRVINE, S. P. MARIN, AND P. W. SMITH, *Constrained interpolation and smoothing*, Constructive Approx., 2 (1986), pp. 129-151.

[27] E. T. JAYNES, *Prior probabilities*, IEEE Trans. 4 (1968), pp. 227-241.

[28] R. W. JOHNSON AND J. E. SHORE, *Which is the better entropy expression for speech processing*: $-S \log S$ *or* $\log S$?, IEEE Trans. Acoust. Speech Signal Process., 32 (1984), pp. 129-137.

[29] L. K. JONES AND V. TRUTZER, *Computationally feasible high-resolution minimum distance procedures which extend the maximum-entropy method*, Inverse Problems, 5 (1989), pp. 749-766.

[30] S. KARLIN AND W. J. STUDDEN, *Tchebycheff Systems: With Applications in Analysis and Statistics*, Wiley-Interscience, New York, 1966.

[31] S. M. KAY AND S. L. MARPLE, *Spectrum analysis—a modern perspective*, IEEE Proc., 69 (1981), pp. 1380-1419.

[32] S. W. LANG AND J. H. MCCLELLAN, *Multidimensional* MEM *spectral estimation*, IEEE Trans. Acoust. Speech Signal Process., 30 (1984), pp. 880-887.

[33] A. S. LEWIS, *Pseudo-Haar functions and partially-finite programming*, manuscript.

[34] ———, *The convergence of entropic estimates for moment problems*, in Workshop/Miniconference on Functional Analysis/Optimization, S. Fitzpatrick and J. Giles, eds., Centre for Mathematical Analysis, Australian National University, Canberra, 1989, pp. 100-115.

[35] D. M. LIN AND E. K. WONG, *A survey on the maximum entropy method and parameter spectral estimation*, Phys. Rep., 193 (1990), pp. 41-135.

[36] J. H. MCCLELLAN AND S. W. LANG, *Multi-dimensional* MEM *spectral estimation*, in Spectral Analysis and Its Use in Underwater Acoustics, Institute of Acoustics, Imperial College, London, U.K., 1982, pp. 10.1-10.8.

[37] L. R. MEAD AND N. PAPANICOLAOU, *Maximum entropy in the problem of moments*, J. Math. Phys., 25 (1984), pp. 2404-2417.

[38] C. A. MICCHELLI, P. W. SMITH, J. SWETITS, AND J. D. WARD, *Constrained L_p approximation*, Constructive Approx., 1 (1985), pp. 93-102.

[39] J. NAVAZA, *The use of non-local constraints in maximum-entropy electron density reconstruction*, Acta Crystallographica, A42 (1986), pp. 212-223.

[40] R. NITYANANDA AND R. NARAYAN, *Maximum entropy image reconstruction—a practical non-information-theoretic appraoch*, J. Astrophys. Astron., 3 (1982), pp. 419-450.

[41] P. R. OLIVEIRA, *Contrôle de processus de vieillissement*, Ph.D. thesis, La Faculté des Sciences de Paris IX, France, 1977.

[42] M. POWELL, *Karmarkar's algorithm for semi-infinite programming*, Tech. Rep., University of Cambridge, Cambridge, UK, 1990.

[43] R. T. ROCKAFELLAR, *Duality and stability in extremum problems involving convex functions*, Pacific J. Math., 21 (1967), pp. 167-187.

[44] ———, *Integrals which are convex functionals*, Pacific J. Math., 24 (1968), pp. 525-539.

[45] ———, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[46] ———, *Integrals which are convex functionals*, II, Pacific J. Math., 39 (1971), pp. 439-469.

[47] ———, *Conjugate Duality and Optimization*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1974.

[48] W. RUDIN, *Real and Complex Analysis*, McGraw-Hill, New York, 1966.

[49] ———, *Functional Analysis*, McGraw-Hill, New York, 1973.

[50] A. SEGHIER, *Reconstruction de la densité spectrale par maximum d'entropie cas d-dimensionnel*, Comptes Rend. Acad. Sci. Paris, Sér. I, 305 (1987), pp. 517-520.

[51] J. SKILLING, ED., *Maximum-Entropy and Bayesian Methods*, Kluwer, Dordrecht, the Netherlands, 1989.

[52] J. SKILLING AND S. F. GULL, *The entropy of an image*, SIAM-AMS Proc., 14 (1984), pp. 167-189.

[53] C. R. SMITH AND G. J. ERICKSON, EDS., *Maximum-Entropy and Bayesian Spectral Analysis and Estimation Problems*, Kluwer, Dordrecht, the Netherlands, 1987.

[54] C. R. SMITH AND W. T. GANDY, EDS., *Maximum-Entropy and Bayesian Methods in Inverse Problems*, Kluwer, Dordrecht, the Netherlands, 1985.

[55] M. TODD, *Interior point methods for semi-infinite programming*, Tech. Rep., Cornell Univ., Ithaca, NY, 1991.

[56] J. W. WOODS, *Two-dimensional Markov spectral estimation*, IEEE Trans. Inform. Theory, 22 (1976), pp. 552-559.

SIAM J. OPTIMIZATION
Vol. 3, No. 2, pp. 268-297, May 1993

REVERSE AUCTION AND THE SOLUTION OF INEQUALITY CONSTRAINED ASSIGNMENT PROBLEMS*

DIMITRI P. BERTSEKAS†, DAVID A. CASTAÑON‡, AND HARALAMPOS TSAKNAKIS§

Abstract. In this paper auction algorithms for solving several types of assignment problems with inequality constraints are proposed. Included are asymmetric problems with different numbers of persons and objects, and multiassignment problems, where persons may be assigned to several objects and vice versa. A central new idea in all these algorithms is to combine regular auction, where persons bid for objects by raising their prices, with reverse auction, where objects compete for persons by essentially offering discounts. Reverse auction can also be used to accelerate substantially (and sometimes dramatically) the convergence of regular auction for symmetric assignment problems.

Key words. assignment problem, network optimization, auction, linear programming

AMS subject classifications. primary, 90C47; secondary, 90C05

1. Introduction. Let us consider the classical symmetric assignment problem where we want to match n persons and n objects on a one-to-one basis. The benefit for matching a person with an object is given, and we want to assign all persons to distinct objects so as to maximize the total benefit. The auction algorithm is a method for solving this problem that was first proposed in [Ber79], and was subsequently developed in [Ber85], [Ber88], and [BeE88]. It operates like a real-life auction. There is a price for each object, and at each iteration, unassigned persons bid simultaneously for their "best" objects (the ones offering maximum benefit minus price), thereby raising the corresponding prices. Objects are then awarded to the highest bidder. The bidding increments must be at least equal to a positive parameter ε, and are chosen so as to preserve an ε-complementary slackness property. For good theoretical as well as practical performance, it may be important to use ε-scaling, which consists of applying the algorithm several times, starting with a large value of ε and successively reducing ε up to an ultimate value that is less than some threshold ($1/n$ when a_{ij} are integer). Each scaling phase provides good initial prices for the next. For tutorial presentations of the auction algorithm, we refer to [Ber90], [Ber91], and [Ber92a].

We note that there are several extensions of the auction algorithm, e.g., to transportation problems [BeC89a] and to minimum cost flow problems (the ε-relaxation method of [Ber86a] and [Ber86b], and the network auction algorithm of [BeC89b]). Computational studies on serial and parallel machines [BeC89b], [BeC89c], [CSW89], [Cas92], [KKZ89], [PhZ88], [WeZ90], [WeZ91], [Zak90] have shown that the algorithm is very effective, particularly for sparse symmetric assignment problems and special types of transportation problems.

In this paper we consider several new extensions of the auction algorithm for variations of the assignment problem described above. For some of these problems, no effective adaptation of the auction algorithm has been known so far, while for other

* Received by the editors March 23, 1991; accepted for publication (in revised form) March 23, 1992. This work was supported in part by the BM/C3 Technology branch of the United States Army Strategic Defense Command.

† Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139.

‡ Department of Electrical Engineering, Boston University, and ALPHATECH, Inc., Burlington, Massachusetts 01803.

§ ALPHATECH, Inc., Burlington, Massachusetts 01803.

problems, including the symmetric assignment problem, the ideas of this paper have resulted in auction algorithms with substantially improved performance over the ones previously known.

Central to the present paper is an alternative form of the auction algorithm, called *reverse auction*, where, roughly, the *objects* compete for persons by *lowering* their prices. In particular, objects decrease their prices to a level that is sufficiently low to lure a person away from his/her currently held object. We can show that forward and reverse auctions are mathematically equivalent, but their combination results in algorithms that can solve problems that forward or reverse auction by themselves either cannot solve at all or can solve but much more slowly.

In the next section, we show how to combine forward and reverse auctions to solve symmetric assignment problems. In particular we provide mechanisms for switching gracefully between the two types of auction, using a special type of ε-complementary slackness condition. As shown by computational results given in § 5, the combined forward/reverse method substantially outperforms the regular (forward) method. The reason appears to be that the combined method suffers much less from "price wars," that is, protracted bid sequences involving a small number of persons competing for a smaller number of objects using small bidding increments. In fact, it may not be necessary to resort to ε-scaling, involving the solution of several subproblems, to improve the performance of the method.

In § 3, we consider asymmetric assignment problems, where the number of persons is less than the number of objects. As a result, in a feasible assignment, we require that every person, but not necessarily every object, be assigned. The original paper on the auction algorithm [Ber79] showed that this problem can be solved by the auction algorithm provided the prices of all objects start at zero. This approach is often very effective in practice, particularly when the number of persons is much less than the number of objects, but unfortunately it precludes the use of ε-scaling. As a result, it is ineffective for problems where price wars are likely to arise. By suitably combining forward and reverse auctions, we eliminate this drawback. In particular, we give a new auction algorithm for solving the asymmetric assignment problem, where the starting object prices can be arbitrary, so that ε-scaling can be used in the same way as for symmetric problems.

In § 4, we consider an interesting class of assignment-like problems, called *mutiassignment problems*, which arise in multitarget tracking applications (see the comments of § 4). There are no specialized network flow methods that can solve these problems at present, although they can be solved by general purpose network methods such as primal-simplex, primal-dual, or relaxation methods. We develop new classes of auction algorithms for multiassignment problems by combining the ideas of forward and reverse auctions.

Finally, in § 5, we present computational results using various experimental codes implementing the new algorithms of this paper. For each of the problems considered (symmetric and asymmetric assignment, and two types of multiassignment problems), we show that the new methods of this paper substantially (and often dramatically) outperform current state-of-the-art codes.

2. Reverse auction for symmetric assignment problems. In the symmetric assignment problem there are n persons and n objects. The benefit or value of assigning person i to object j is a_{ij}. The set of objects to which person i can be assigned is a nonempty set denoted $A(i)$. An *assignment* S is a (possibly empty) set of person-object pairs (i, j) such that $j \in A(i)$ for all $(i, j) \in S$; for each person i there can be at most one pair

$(i,j)\in S$; and for every object j there can be at most one pair $(i,j)\in S$. Given an assignment S, we say that person i is *assigned* if there exists a pair $(i,j)\in S$; otherwise we say that i is *unassigned*. We use similar terminology for objects. An assignment is said to be *feasible* if it contains n pairs, so that every person and every object is assigned; otherwise the assignment is called *partial*. We want to find an assignment $\{(1,j_1),\dots,(n,j_n)\}$ with maximum total benefit $\sum_{i=1}^{n} a_{ij_i}$.

The auction algorithm for the symmetric assignment problem proceeds iteratively and terminates when a feasible assignment is obtained. At the start of the generic iteration we have a partial assignment S and a price vector $p=(p_1,\dots,p_n)$ satisfying *ε-complementary slackness* (ε-CS). This is the condition

$$a_{ij}-p_j \geqq \max_{k\in A(i)} \{a_{ik}-p_k\}-\varepsilon \quad \forall (i,j)\in S. \tag{1}$$

As an initial choice, one can use an arbitrary set of prices together with the empty assignment, which trivially satisfies ε-CS. The iteration consists of two phases: the *bidding phase* and the *assignment phase*, described in the following.

BIDDING PHASE. Let I be a nonempty subset of persons i that are unassigned under the assignment S. For each person $i\in I$:

(1) Find a "best" object j_i having maximum value, that is,

$$j_i=\arg\max_{j\in A(i)} \{a_{ij}-p_j\},$$

and the corresponding value

$$v_i=\max_{j\in A(i)} \{a_{ij}-p_j\}, \tag{2}$$

and find the best value offered by objects other than j_i,

$$w_i=\max_{j\in A(i), j\neq j_i} \{a_{ij}-p_j\}. \tag{3}$$

(If j_i is the only object in $A(i)$, we define w_i to be $-\infty$ or, for computational purposes, a number that is much smaller than v_i.)

(2) Compute the "bid" of person i given by

$$b_{ij_i}=p_{j_i}+v_i-w_i+\varepsilon=a_{ij_i}-w_i+\varepsilon. \tag{4}$$

(We characterize this situation by saying that person i bid for object j_i, and that object j_i received a bid from person i. The algorithm works if the bid has any value between $p_{j_i}+\varepsilon$ and $p_{j_i}+v_i-w_i+\varepsilon$, but it tends to work fastest for the maximal choice of (4).)

ASSIGNMENT PHASE. For each object j:

Let $P(j)$ be the set of persons from which j received a bid in the bidding phase of the iteration. If $P(j)$ is nonempty, increase p_j to the highest bid,

$$p_j := \max_{i\in P(j)} b_{ij}, \tag{5}$$

remove from the assignment S any pair (i,j) (if j was assigned to some i under S), and add to S the pair (i_j,j), where i_j is a person in $P(j)$ attaining the maximum above.

Note that there is some freedom in choosing the subset of persons I that bid during an iteration. One possibility is to let I consist of a single unassigned person. This version, known as the *Gauss-Seidel version* in view of its similarity with Gauss-Seidel methods for solving systems of nonlinear equations, usually works best in a serial computing environment. The version where I consists of all unassigned persons is the one best suited for parallel computation, and is known as the *Jacobi version*, in view of its similarity with Jacobi methods for solving systems of nonlinear equations.

The choice of bidding increment $v_i - w_i + \varepsilon$ for a person i [cf. (4)] is such that ε-CS is preserved, as stated in the following well-known proposition.

PROPOSITION 1. *The auction algorithm preserves* ε-CS *throughout its execution*; *that is, if the assignment and price vector available at the start of an iteration satisfy* ε-CS, *the same is true for the assignment and price vector obtained at the end of the iteration.*

Proof. See [Ber79], [Ber88], [BT89], or [Ber91] for the proof.

Furthermore, the algorithm is valid in the sense stated below.

PROPOSITION 2. *If at least one feasible assignment exists, the auction algorithm terminates in a finite number of iterations with a feasible assignment that is within* $n\varepsilon$ *of being optimal* (*and is optimal if the problem data is integer and* $\varepsilon < 1/n$).

Proof. See [Ber79], [Ber88], [BeT89], or [Ber91] for the proof.

The auction algorithm can be shown to have an $O(A(n + nC/\varepsilon))$ worst-case running time, where A is the number of arcs of the assignment graph, and

$$C = \max_{(i,j)\in\mathcal{A}} |a_{ij}|$$

is the maximum absolute object value; see [Ber79], [BeE88], and [BeT89]. Thus, the amount of work needed to solve the problem can depend strongly on the value of ε as well as of C. In practice, the dependence of the running time on ε and C is often significant, particularly for sparse problems.

To obtain polynomial complexity, we can use *ε-scaling*, which consists of applying the algorithm several times, starting with a large value of ε and successively reducing ε up to an ultimate value that is less than $1/n$. Each application of the algorithm, called a *scaling phase*, provides good initial prices for the next application. For integer data, it can be shown that the worst-case running time of the auction algorithm using scaling and appropriate data structures is $O(nA \log(nC))$; see [BeE88] and [BeT89]. We note that while ε-scaling was suggested in the original proposal of the auction algorithm [Ber79], it was first analyzed in [Gol87] (see also [GoT90]) in the context of the ε-relaxation method. This minimum cost flow algorithm (also known as preflow-push) was proposed in [Ber86a] and [Ber86b], and is essentially equivalent to the auction algorithm [Ber92b]. Not much is known about the average complexity of the auction algorithm. However, an interesting analysis of [Sch90] suggests that for uniformly distributed arc costs its running time grows proportionally to something like $A \log n$ or $A \log n \log(nC)$; this is roughly consistent with computational results using randomly generated problems.

2.1. Reverse auction. In the auction algorithm, persons compete for objects by bidding and raising the price of their best object. It is possible to use an alternative form of the auction algorithm, called *reverse auction*, where *objects* compete for persons. In particular, objects decrease their prices to a level that is sufficiently low to either attract an unassigned person or lure a person away from its currently held object.

In order to describe reverse auction, we introduce a *profit* variable π_i for each person i. The role that profits play for persons is analogous to the role prices play for objects. We can describe the reverse auction algorithm in two equivalent ways: one where unassigned objects lower their prices as much as possible to attract a person without violating ε-CS, and another where unassigned objects select a best person and raise his/her profit as much as possible without violating ε-CS. For analytical convenience, we will adopt the second description rather than the first.

Let us consider the following ε-CS condition for a (partial) assignment S and a profit vector π:

$$a_{ij} - \pi_i \geqq \max_{k \in B(j)} \{a_{kj} - \pi_k\} - \varepsilon \quad \forall (i,j) \in S, \tag{6}$$

where $B(j)$ is the set of persons that can be assigned to object j,

$$B(j) = \{i | (i,j) \in \mathcal{A}\}.$$

For feasibility, we assume that this set is nonempty for all j. Note the symmetry of this condition with the corresponding one for prices; cf. (1). The reverse auction algorithm starts with and maintains an assignment and a profit vector π satisfying the above ε-CS condition. It terminates when the assignment is feasible. At the beginning of each iteration, we have an assignment S and a profit vector π satisfying the ε-CS condition (6).

Typical Iteration of Reverse Auction. Let J be a nonempty subset of objects j that are unassigned under the assignment S. For each object $j \in J$:

(1) Find a "best" person i_j such that

$$i_j = \arg \max_{i \in B(j)} \{a_{ij} - \pi_i\},$$

and the corresponding value

$$\beta_j = \max_{i \in B(j)} \{a_{ij} - \pi_i\}, \tag{7}$$

and find

$$\omega_j = \max_{i \in B(j), i \neq i_j} \{a_{ij} - \pi_i\}. \tag{8}$$

(If i_j is the only person in $B(j)$, we define ω_j to be $-\infty$ or, for computational purposes, a number that is much smaller than β_j.)

(2) Each object $j \in J$ bids for person i_j an amount

$$b_{i_j j} = \pi_{i_j} + \beta_j - \omega_j + \varepsilon = a_{i_j j} - \omega_j + \varepsilon. \tag{9}$$

(3) For each person i that received at least one bid, increase π_i to the highest bid

$$\pi_i := \max_{j \in P(i)} b_{ij}, \tag{10}$$

where $P(i)$ is the set of objects from which i received a bid; remove from the assignment S any pair (i, j) (if i was assigned to some j under S), and add to S the pair (i, j_i), where j_i is an object in $P(i)$ attaining the maximum above.

Note that reverse auction is identical to (forward) auction with the roles of persons and objects, as well as profits and prices, interchanged. Thus, by using the corresponding (forward) auction result (cf. Proposition 2), we have the following.

Proposition 3. *If at least one feasible assignment exists, the reverse auction algorithm terminates in a finite number of iterations. The feasible assignment obtained upon termination is within $n\varepsilon$ of being optimal* (*and is optimal if the problem data are integer and $\varepsilon < 1/n$*).

2.2. Combined forward and reverse auction. One of the reasons we are interested in reverse auction is to construct algorithms that switch from forward to reverse auction and back. Such algorithms must simultaneously maintain a price vector p satisfying the ε-CS condition (1) and a profit vector π satisfying the ε-CS condition (6). To this end we introduce an ε-CS condition for the *pair* (π, p), which, as we will see, implies the other two. Maintaining this condition is essential for switching gracefully between forward and reverse auction.

DEFINITION 1. An assignment S and a pair (π, p) are said to satisfy ε-CS if

$$\pi_i + p_j \geqq a_{ij} - \varepsilon \quad \forall (i, j) \in \mathcal{A}, \tag{11a}$$

$$\pi_i + p_j = a_{ij} \quad \forall (i, j) \in S. \tag{11b}$$

We have the following proposition.

PROPOSITION 4. *Suppose that an assignment S, together with a profit-price pair (π, p) satisfies ε-CS. Then*

(a) *S and π satisfy the ε-CS condition*

$$a_{ij} - \pi_i \geqq \max_{k \in B(j)} \{a_{kj} - \pi_k\} - \varepsilon \quad \forall (i, j) \in S. \tag{12}$$

(b) *S and p satisfy the ε-CS condition*

$$a_{ij} - p_j \geqq \max_{k \in A(i)} \{a_{ik} - p_k\} - \varepsilon \quad \forall (i, j) \in S. \tag{13}$$

(c) *If S is feasible, then S is within $n\varepsilon$ of being an optimal assignment.*

Proof. (a) In view of (11b), for all $(i, j) \in S$, we have $p_j = a_{ij} - \pi_i$, so (11a) implies that $a_{ij} - \pi_i \geqq a_{kj} - \pi_k - \varepsilon$ for all $k \in B(j)$. This shows (12).

(b) The proof is the same as that of part (a) with the roles of π and p interchanged.

(c) Since by part (b), the ε-CS condition (13) is satisfied, by Proposition 2, S is within $n\varepsilon$ of being optimal. □

We now introduce a combined forward/reverse algorithm. The algorithm starts with and maintains an assignment S and a profit-price pair (π, p) satisfying the ε-CS condition (11). It terminates when the assignment is feasible.

COMBINED FORWARD/REVERSE AUCTION ALGORITHM.

Step 1 (run forward auction): Execute several iterations of the forward auction algorithm (subject to the termination condition), and at the end of each iteration (after increasing the prices of the objects that received a bid), set

$$\pi_i = a_{ij_i} - p_{j_i}, \tag{14}$$

for every person-object pair (i, j_i) that entered the assignment during the iteration. Go to Step 2.

Step 2 (run reverse auction): Execute several iterations of the reverse auction algorithm (subject to the termination condition), and at the end of each iteration (after increasing the profits of the persons that received a bid), set

$$p_j = a_{i_j j} - \pi_{i_j}, \tag{15}$$

for every person-object pair (i_j, j) that entered the assignment during the iteration. Go to Step 1.

Note that the additional overhead of the combined algorithm over the forward or the reverse algorithm is minimal; just one update of the form (14) or (15) is required per iteration for each object or person that received a bid during the iteration. An alternative but probably less efficient possibility is to update the profits π_i of the assigned persons via (14) (or the prices p_j of the assigned objects via (15)) just before switching to reverse auction (or forward auction, respectively). An important property is that the updates of (14) and (15) maintain the ε-CS condition (11) for the pair

(π, p), and therefore, by Proposition 4, maintain the required ε-CS conditions (12) and (13) for π and p, respectively. This is shown in the following proposition.

PROPOSITION 5. *If the assignment and profit-price pair available at the start of an iteration of either the forward or the reverse auction algorithm satisfy the ε-CS condition* (11), *the same is true for the assignment and profit-price pair obtained at the end of the iteration, provided* (14) *is used to update π (in the case of forward auction), and* (15) *is used to update p (in the case of reverse auction).*

Proof. Assume for concreteness that forward auction is used, and let (π, p) and $(\bar{\pi}, \bar{p})$ be the profit-price pair before and after the iteration, respectively. Then, $\bar{p}_j \geqq p_j$ for all j (with strict inequality if and only if j received a bid during the iteration). Therefore, we have $\bar{\pi}_i + \bar{p}_j \geqq a_{ij} - \varepsilon$ for all (i, j) such that $\pi_i = \bar{\pi}_i$. Furthermore, we have $\bar{\pi}_i + \bar{p}_j = \pi_i + p_j = a_{ij}$ for all (i, j) that belong to the assignment before as well as after the iteration. Also, in view of the update (14), we have $\bar{\pi}_i + \bar{p}_{j_i} = a_{ij_i}$ for all pairs (i, j_i) that entered the assignment during the iteration. What remains is to verify that the condition

$$(16) \qquad \bar{\pi}_i + \bar{p}_j \geqq a_{ij} - \varepsilon \quad \forall j \in A(i)$$

holds for all persons i that submitted a bid and were assigned to an object, say j_i, during the iteration. Indeed, for such a person i, we have by (4),

$$\bar{p}_{j_i} = a_{ij_i} - \max_{j \in A(i), j \neq j_i} \{a_{ij} - p_j\} + \varepsilon,$$

which implies that

$$\bar{\pi}_i = a_{ij_i} - \bar{p}_{j_i} \geqq a_{ij} - p_j - \varepsilon \geqq a_{ij} - \bar{p}_j - \varepsilon \quad \forall j \in A(i).$$

This shows the desired relation (16). □

Note that during forward auction, the object prices p_j increase, while the profits π_i decrease, but exactly the opposite happens in reverse auction. For this reason, the termination proof used for forward auction (see, e.g., [BeT89, p. 371]) does not apply to the combined method. Indeed, it is possible to construct examples of feasible problems where the combined method never terminates if the switch between forward and reverse auctions is done arbitrarily. However, it is easy to guarantee that the combined algorithm terminates finitely for a feasible problem; it is sufficient to ensure that some "irreversible progress" is made before switching between forward and reverse auction. One easily implementable possibility is to refrain from switching until at least one more person-object pair has been added to the assignment. In this way there can be a switch at most $(n-1)$ times between the forward and reverse steps of the algorithm. Since for a feasible problem, forward and reverse auction by themselves have guaranteed finite termination, the final step will terminate with a feasible assignment satisfying ε-CS.

The combined forward/reverse auction algorithm often works substantially faster than the forward version. It seems to be affected less by "price wars," that is, protracted sequences of small price rises by a number of persons bidding for a smaller number of objects. Price wars can still occur in the combined algorithm, but they arise through more complex and unlikely problem structures than in the forward algorithm. For this reason the combined forward/reverse auction algorithm depends less on ε-scaling for good performance than its forward counterpart. One consequence of this is that starting with $\varepsilon = 1/n$ and bypassing ε-scaling is often the best choice. Another consequence is that a larger ε-reduction factor can typically be used with no price war effects in ε-scaled forward/reverse auction than in ε-scaled forward auction. As a result, fewer

ε-scaling phases are typically needed in forward/reverse auction to deal effectively with price wars.

3. Auction algorithms for asymmetric assignment problems. Reverse auction can be used in conjunction with forward auction to provide algorithms for solving the asymmetric assignment problem, where the number of objects n is larger than the number of persons m. Here we still require that each person be assigned to some object, but we allow objects to remain unassigned. As before, an assignment S is a (possibly empty) set of person-object pairs (i,j) such that $j \in A(i)$ for all $(i,j) \in S$; for each person i there can be at most one pair $(i,j) \in S$; and for every object j there can be at most one pair $(i,j) \in S$. The assignment S is said to be feasible if all persons are assigned under S.

The corresponding linear programming problem is

$$\begin{aligned} &\text{maximize} \quad \sum_{(i,j)\in\mathcal{A}} a_{ij}x_{ij} \\ &\text{subject to} \\ &\sum_{j\in A(i)} x_{ij} = 1 \quad \forall i = 1,\dots,m, \\ &\sum_{i\in B(j)} x_{ij} \leqq 1 \quad \forall j = 1,\dots,n, \\ &0 \leqq x_{ij} \quad \forall (i,j)\in\mathcal{A}. \end{aligned} \tag{17}$$

We can convert this program to the minimum cost flow problem

$$\begin{aligned} &\text{minimize} \quad \sum_{(i,j)\in\mathcal{A}} (-a_{ij})x_{ij} \\ &\text{subject to} \\ &\sum_{j\in A(i)} x_{ij} = 1 \quad \forall i = 1,\dots,m, \\ &\sum_{i\in B(j)} x_{ij} + x_{sj} = 1 \quad \forall j = 1,\dots,n, \\ &\sum_{j=1}^{n} x_{sj} = n - m, \\ &0 \leqq x_{ij} \quad \forall (i,j)\in\mathcal{A}, \\ &0 \leqq x_{sj} \quad \forall j = 1,\dots,n, \end{aligned} \tag{18}$$

by replacing maximization by minimization, by reversing the sign of a_{ij}, and by introducing a supersource node s, which is connected to each object node j by an arc (s,j) of zero cost and feasible flow range $[0,\infty)$.

Using the duality theory for minimum cost network flow problems (see, e.g., [BeT89, p. 335] or [Ber91, p. 35]), it can be verified that the corresponding dual problem is

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} \pi_i + \sum_{j=1}^{n} p_j - (n-m)\lambda \\ &\text{subject to} \quad \pi_i + p_j \geqq a_{ij} \quad \forall (i,j)\in\mathcal{A}, \\ &\qquad\qquad\quad \lambda \leqq p_j \quad \forall j = 1,\dots,n, \end{aligned} \tag{19}$$

where we have converted maximization to minimization, we have used $-\pi_i$ in place of the price of each person node i, and we have denoted by λ the price of the supersource node s.

We now introduce an ε-CS condition for an assignment S and a pair (π, p).

DEFINITION 2. An assignment S and a pair (π, p) are said to satisfy ε-CS if

$$\pi_i + p_j \geqq a_{ij} - \varepsilon \quad \forall (i, j) \in \mathcal{A}, \tag{20a}$$

$$\pi_i + p_j = a_{ij} \quad \forall (i, j) \in S, \tag{20b}$$

$$p_j \leqq \min_{k:\,\text{assigned under}\,S} p_k \quad \forall j \quad \text{unassigned under } S. \tag{20c}$$

The following proposition clarifies the significance of the preceding ε-CS condition.

PROPOSITION 6. *If a feasible assignment S satisfies the ε-CS conditions (20) together with a pair (π, p), then S is within $m\varepsilon$ of being optimal for the asymmetric assignment problem. The triplet $(\hat{\pi}, \hat{p}, \lambda)$, where*

$$\lambda = \min_{k:\,\text{assigned under}\,S} p_k, \tag{21a}$$

$$\hat{\pi}_i = \pi_i + \varepsilon \quad \forall i = 1, \dots, m, \tag{21b}$$

$$\hat{p}_j = \begin{cases} p_j & \text{if } j \text{ is assigned under } S, \\ \lambda & \text{if } j \text{ is unassigned under } S \end{cases} \quad \forall j = 1, \dots, n, \tag{21c}$$

is within $m\varepsilon$ of being an optimal solution of the dual problem (19).

Proof. For any feasible assignment $\{(i, k_i) | i = 1, \dots, m\}$ and for any triplet $(\bar{\pi}, \bar{p}, \lambda)$ satisfying the dual feasibility constraints $\bar{\pi}_i + \bar{p}_j \geqq a_{ij}$ for all $(i, j) \in \mathcal{A}$ and $\lambda \leqq \bar{p}_j$ for all j, we have

$$\sum_{i=1}^{m} a_{ik_i} \leqq \sum_{i=1}^{m} \bar{\pi}_i + \sum_{i=1}^{m} \bar{p}_{k_i} \leqq \sum_{i=1}^{m} \bar{\pi}_i + \sum_{j=1}^{n} \bar{p}_j - (n - m)\lambda.$$

By maximizing over all feasible assignments $\{(i, k_i) | i = 1, \dots, m\}$ and by minimizing over all dual-feasible triplets $(\bar{\pi}, \bar{p}, \lambda)$, we see that

$$A* \leqq D*,$$

where $A*$ is the optimal assignment value and $D*$ is the minimal dual cost.

Let now $S = \{(i, j_i) | i = 1, \dots, m\}$ be the given assignment satisfying ε-CS together with (π, p), and consider the triplet $(\hat{\pi}, \hat{p}, \lambda)$ defined by (21). Since for all i, we have $\hat{\pi}_i + \hat{p}_{j_i} = a_{ij} + \varepsilon$, we obtain

$$A* \geqq \sum_{i=1}^{m} a_{ij_i} = \sum_{i=1}^{m} \hat{\pi}_i + \sum_{i=1}^{m} \hat{p}_{j_i} - m\varepsilon = \sum_{i=1}^{m} \hat{\pi}_i + \sum_{j=1}^{n} \hat{p}_j - (n - m)\lambda - m\varepsilon \geqq D* - m\varepsilon,$$

where the last inequality holds because the triplet $(\hat{\pi}, \hat{p}, \lambda)$ is feasible for the dual problem. Since we showed earlier that $A* \leqq D*$, the desired conclusion follows. □

Consider now trying to solve the asymmetric assignment problem by means of auction. We can start with any assignment S and pair (π, p) satisfying the first two ε-CS conditions (20a) and (20b), and perform a forward auction (as defined earlier for the symmetric assignment problem) up to the point where each person is assigned to a distinct object. For a feasible problem, it can be seen that this will yield, in a finite number of iterations, a feasible assignment S satisfying the first two conditions (20a) and (20b). If we select initially all object prices to be zero, then upon termination of the algorithm, the prices of the unassigned objects will still be at zero, while the prices of the assigned objects will be nonnegative. Therefore, the ε-CS condition (20c) will also be satisfied, and by Proposition 6 the assignment S obtained will be optimal. Unfortunately, the use of zero initial prices precludes the use of ε-scaling, and leaves the method susceptible to price wars. To be able to use ε-scaling we must be able to use arbitrary initial prices, but then the assignment S obtained by forward auction may not be optimal because the prices of the unassigned objects may not be minimal, that is, they may not satisfy the third ε-CS condition (20c). Roughly, what is happening here is that forward auction cannot resolve whether the objects that were left unassigned upon termination are intrinsically "undesirable" because they offer relatively low benefit to the persons, or whether they were left unassigned because their initial prices were high relative to the initial prices of the assigned objects.

To resolve this dilemma, we use a modified form of reverse auction to lower the prices of the objects that were left unassigned upon termination of the forward auction. After several reverse auction iterations in which persons may be reassigned to other objects, the third condition (20c) will be satisfied. We will show that the assignment thus obtained satisfies all the ε-CS conditions (20a)-(20c) and by Proposition 6 is optimal within $m\varepsilon$ (and thus optimal if the problem data are integer and $\varepsilon < 1/m$).

The modified reverse auction starts with a feasible assignment S and with a pair (π, p) satisfying the first two ε-CS conditions (20a) and (20b). (For a feasible problem, such an S and (π, p) can be obtained by regular forward or reverse auction, as discussed earlier.) Let us denote by λ the minimal assigned object price under the initial assignment,

$$\lambda = \min_{j:\,\text{assigned under the initial assignment } S} p_j. \tag{22}$$

The typical iteration of modified reverse auction is the same as the one of reverse auction, except that only unassigned objects j with $p_j > \lambda$ participate in the auction. In particular, the algorithm maintains a feasible assignment S and a pair (π, p) satisfying (20a) and (20b), and terminates when all unassigned objects j satisfy $p_j \leqq \lambda$, in which case it will be seen that the third ε-CS condition (20c) will be satisfied as well. The scalar λ will be kept fixed throughout the algorithm.

Typical Iteration of Modified Reverse Auction for Asymmetric Assignment. Select an object j that is unassigned under the assignment S, and satisfies $p_j > \lambda$ (if no such object can be found, the algorithm terminates). Find a "best" person i_j such that

$$i_j = \arg\max_{i \in B(j)} \{a_{ij} - \pi_i\},$$

and the corresponding value

$$\beta_j = \max_{i \in B(j)} \{a_{ij} - \pi_i\}, \tag{23}$$

and find

$$\omega_j = \max_{i \in B(j), i \neq i_j} \{a_{ij} - \pi_i\}. \tag{24}$$

(If i_j is the only person in $B(j)$, we define ω_j to be $-\infty$.) If $\lambda \geqq \beta_j - \varepsilon$, set $p_j := \lambda$ and go to the next iteration. Otherwise, let

$$\delta = \min\{\beta_j - \lambda, \beta_j - \omega_j + \varepsilon\}. \tag{25}$$

Set

$$p_j := \beta_j - \delta, \tag{26}$$

$$\pi_{i_j} := \pi_{i_j} + \delta, \tag{27}$$

add to the assignment S the pair (i_j, j), and remove from S the pair (i_j, j'), where j' is the object that was assigned to i_j under S at the start of the iteration.

Note that the formula (25) for the bidding increment δ is such that the object j enters the assignment at a price which is no less than λ (and is equal to λ if and only if the minimum in (25) is attained by the first term). Furthermore, we have $\delta \geqq \varepsilon$ (when δ is calculated, that is, when $\lambda > \beta_j - \varepsilon$), so it can be seen from (26) and (27) that throughout the algorithm, prices are monotonically decreasing and profits are monotonically increasing. The following proposition establishes the validity of the method.

PROPOSITION 7. *The modified reverse auction algorithm for the asymmetric assignment problem terminates in a finite number of iterations and the assignment obtained is within $m\varepsilon$ of being optimal.*

Proof. In view of Proposition 6, the result will follow once we prove the following:

(a) The modified reverse auction iteration preserves the first two ε-CS conditions (20a) and (20b), as well as the condition

$$\lambda \leqq \min_{j:\,\text{assigned under the current assignment } S} p_j, \tag{28}$$

so upon termination of the algorithm (necessarily with the prices of all unassigned objects less or equal to λ), the third ε-CS condition (20c) is satisfied.

(b) The algorithm terminates finitely.

We will prove these facts in sequence.

We assume that the conditions (20a), (20b), and (28) are satisfied at the start of an iteration, and we will show that they are also satisfied at the end of the iteration. First, consider the case where there is no change in the assignment, which happens when $\lambda \geqq \beta_j - \varepsilon$. Then (20b) and (28) are automatically satisfied at the end of the iteration; only p_j changes in the iteration according to

$$p_j := \lambda \geqq \beta_j - \varepsilon = \max_{i \in B(j)} \{a_{ij} - \pi_i\} - \varepsilon,$$

so the condition (20a) is also satisfied at the end of the iteration.

Next consider the case where there is a change in the assignment during the iteration. Let (π, p) and $(\bar{\pi}, \bar{p})$ be the profit-price pair before and after the iteration, respectively, and let j and i_j be the object and person involved in the iteration. By construction [cf. (26) and (27)], we have $\bar{\pi}_{i_j} + \bar{p}_j = a_{i_j j}$, and since $\bar{\pi}_i = \pi_i$ and $\bar{p}_k = p_k$ for all $i \neq i_j$ and $k \neq j$, we see that the condition (20b) $(\bar{\pi}_i + \bar{p}_k = a_{ik})$ is satified for all assigned pairs (i, k) at the end of the iteration.

To show that the condition (20a) is satisfied at the end of the iteration, that is,

$$\bar{\pi}_i + \bar{p}_k \geqq a_{ik} - \varepsilon \quad \forall (i, k) \in \mathcal{A}, \tag{29}$$

consider first objects $k \neq j$. Then, $\bar{p}_k = p_k$ and since $\bar{\pi}_i \geqq \pi_i$ for all i, the above condition holds, since at the start of the iteration, we have $\pi_i + p_k \geqq a_{ik} - \varepsilon$ for all (i, k). Consider next the case $k = j$. Then, condition (29) holds for $i = i_j$, since $\bar{\pi}_{i_j} + \bar{p}_j = a_{i_j j}$. Also using (23)-(26) and the fact $\delta \geqq \varepsilon$, we have for all $i \neq i_j$,

$$\begin{aligned} \bar{\pi}_i + \bar{p}_j &= \pi_i + \bar{p}_j \geqq \pi_i + \beta_j - (\beta_j - \omega_j + \varepsilon) \\ &= \pi_i + \omega_j - \varepsilon \geqq \pi_i + (a_{ij} - \pi_i) - \varepsilon = a_{ij} - \varepsilon, \end{aligned}$$

so condition (29) holds for $i \neq i_j$ and $k = j$, completing the proof.

To see that condition (28) is maintained by the iteration, note that by (23), (24), and (26), we have

$$\bar{p}_j = \beta_j - \delta \geqq \beta_j - (\beta_j - \lambda) = \lambda.$$

Finally, to show that the algorithm terminates finitely, we note that in the typical iteration involving object j and person i_j, there are two possibilities:

(1) The price of object j is set to λ without the object entering the assignment; this occurs if $\lambda \geqq \beta_j - \varepsilon$.

(2) The profit of person i_j increases by at least ε (this is seen from the definition (25) of δ; we have $\lambda < \beta_j - \varepsilon$ and $\beta_j \geqq \omega_j$, so $\delta \geqq \varepsilon$).

Since only objects j with $p_j > \lambda$ can participate in the auction, possibility (1) can occur only a finite number of times. Thus, if the algorithm does not terminate, the profits of some persons will increase to ∞. This is impossible, since when person i is assigned to object j, we must have by (20b) and (28)

$$\pi_i = a_{ij} - p_j \leqq a_{ij} - \lambda,$$

so the profits are bounded from above by $\max_{(i,j)\in\mathscr{A}} a_{ij} - \lambda$. Thus the algorithm must terminate finitely. □

As mentioned earlier, forward auction followed by modified reverse auction can start with arbitrary initial prices. As a result, one can use ε-scaling, performing a sequence of auctions with decreasing values of ε. This can be shown to improve the theoretical worst-case complexity of the method, and is often beneficial in practice, particularly for sparse problems. Out of several possible variations of the method, the one we have tested most uses the modified reverse auction only in the last ε-scaling phase. In all other ε-scaling phases just forward auction is used.

Reverse auction also can be used to solve the variation of the two-sided inequality constrained assignment problem, where persons (as well as objects) need not be assigned if this degrades the assignment's value. This problem can be converted to an asymmetric assignment problem where all persons must be assigned by introducing for each person i an artificial object i' and a zero cost arc (i, i'). One can then use the algorithm given earlier to solve this problem. The algorithm can be streamlined so that the calculations involving the artificial objects and arcs are handled efficiently.

4. Auction algorithms for multiassignment problems. An interesting type of assignment problem is described by the linear program

$$\begin{aligned} &\text{maximize} \quad \sum_{(i,j)\in\mathscr{A}} a_{ij}x_{ij} \\ &\text{subject to} \\ (30)\qquad &\sum_{j\in\mathscr{A}(i)} x_{ij} \geqq 1 \quad \forall i = 1, \ldots, m, \\ &\sum_{i\in B(j)} x_{ij} = 1 \quad \forall j = 1, \ldots, n, \\ &0 \leqq x_{ij} \quad \forall (i,j) \in \mathscr{A}, \end{aligned}$$

where $m < n$. For feasibility, we assume that the sets $A(i)$ and $B(j)$ are nonempty for all i and all j, respectively. This is known as the *multiassignment* problem, and is characterized by the possibility of assignment of more than one object to a single person; such a person is said to be *multiassigned.* Problems of this type arise in military applications, such as multitarget tracking with sensors of limited resolution [Bla86], where objects correspond to tracked vehicles and persons correspond to data points, each representing at least one vehicle (but possibly more than one, because of the sensor's limited resolution). The multiassignment problem results when we try to associate data points with vehicles so as to match as closely as possible these data points with our prior knowledge of the vehicles' positions.

We can convert the multiassignment problem to the minimum cost flow problem

$$\text{minimize} \quad \sum_{(i,j)\in\mathcal{A}} (-a_{ij})x_{ij}$$

$$\text{subject to}$$

$$(31) \qquad \begin{aligned} &\sum_{j\in A(i)} x_{ij} - x_{si} = 1 \quad \forall i = 1,\ldots,m,\\ &\sum_{i\in B(j)} x_{ij} = 1 \quad \forall j = 1,\ldots,n,\\ &\sum_{i=1}^{m} x_{si} = n - m,\\ &0 \leqq x_{ij} \quad \forall (i,j)\in\mathcal{A},\\ &0 \leqq x_{si} \quad \forall i = 1,\ldots,n, \end{aligned}$$

by replacing maximization by minimization, by reversing the sign of a_{ij}, and by introducing a supersource node s, which is connected to each person node i by an arc (s, i) of zero cost and feasible flow range $[0, \infty)$ (see Fig. 1).

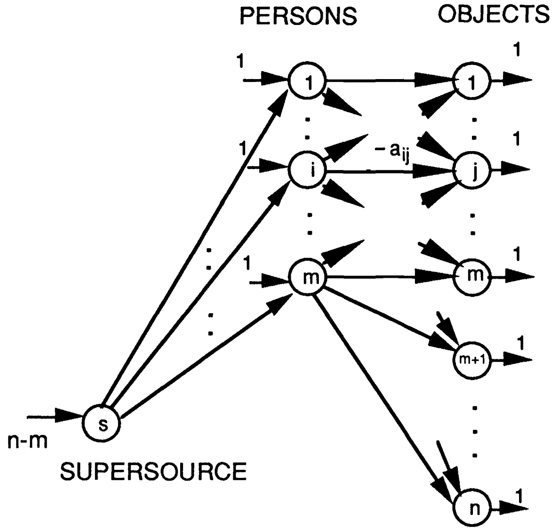


FIG. 1. *Converting a multiassignment problem into a minimum cost flow problem involving a supersource node s and a zero cost artificial arc (s, i) with feasible flow range $[0, \infty)$ for each person i.*

Using duality theory again and appropriately redefining the price variables corresponding to the nodes, it can be verified that the corresponding dual problem is

$$(32) \qquad \begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m} \pi_i + \sum_{j=1}^{n} p_j + (n-m)\lambda\\ \text{subject to} \quad & \pi_i + p_j \geqq a_{ij} \quad \forall (i,j)\in\mathcal{A},\\ & \lambda \geqq \pi_i \quad \forall i = 1,\ldots,m. \end{aligned}$$

We now introduce an ε-CS condition for an assignment S and a pair (π, p).

DEFINITION 3. A multiassignment S and a pair (π, p) are said to satisfy ε-CS if

$$\pi_i + p_j \geqq a_{ij} - \varepsilon \quad \forall (i,j) \in \mathcal{A}, \tag{33a}$$

$$\pi_i + p_j = a_{ij} \quad \forall (i,j) \in S, \tag{33b}$$

$$\pi_i = \max_{k=1,\dots,m} \pi_k \quad \text{if } i \text{ is multiassigned under } S. \tag{33c}$$

We have the following result.

PROPOSITION 8. *Assume that the benefits a_{ij} are integer. If a feasible assignment S satisfies the ε-*CS *conditions* (33) *together with a pair (π, p) for $\varepsilon < 1/m$, then S is optimal for the multiassignment problem.*

Proof. If S is not optimal, there must exist a cycle Y in the equivalent network of Fig. 1 with no repeated nodes along which the assignment S can be modified to result in a new feasible assignment S' with improved primal cost. Assume for the moment that the supersource s is in the cycle; thus, let Y be

$$Y = (s, i_1, j_2, i_2, \dots, i_{k-1}, j_k, i_k, s).$$

In the above cycle, the nodes i_q represent distinct persons, the nodes j_q represent distinct objects and

$$(i_q, j_q) \in S, \quad j_q \in A(i_{q-1}), \quad (i_{q-1}, j_q) \notin S, \quad q = 2, \dots, k.$$

Augmentation along Y results in replacing the pairs $(i_q, j_q) \in S$, $q = 2, \dots, k$, by the pairs (i_{q-1}, j_q), $q = 2, \dots, k$, in the assignment. It can be seen that i_k must be multiassigned prior to the augmentation; the reason is that with the augmentation along Y, the arc (i_k, j_k) will exit the assignment, so person i_k will be left unassigned and feasibility will be violated after the augmentation. Because Y has no repeated nodes, we have $k \leqq m$, which, based on the hypothesis, implies $k\varepsilon < 1$.

Since the augmentation results in strict cost improvement and the benefits are integer, we must have

$$\sum_{q=2}^{k} a_{i_q j_q} + 1 \leqq \sum_{q=2}^{k} a_{i_{q-1} j_q},$$

or equivalently,

$$\sum_{q=2}^{k} (a_{i_q j_q} - p_{j_q}) + 1 \leqq \sum_{q=2}^{k} (a_{i_{q-1} j_q} - p_{j_q}).$$

Using the above relation and the ε-CS condition (33a), it follows that

$$\sum_{q=2}^{k} \pi_{i_q} + 1 = \sum_{q=2}^{k} (a_{i_q j_q} - p_{j_q}) + 1 \leqq \sum_{q=2}^{k} (a_{i_{q-1} j_q} - p_{j_q}) \leqq \sum_{q=1}^{k-1} \pi_{i_q} + (k-1)\varepsilon.$$

From this relation, we obtain

$$1 - (k-1)\varepsilon \leqq \pi_{i_1} - \pi_{i_k}.$$

This is a contradiction because we argued earlier that $k\varepsilon < 1$, and that i_k is multiassigned, which implies that $\pi_{i_k} \geqq \pi_{i_1}$ (cf. (33c)).

If Y does not contain s, a similar argument establishes the result. □

Consider now trying to solve the multiassignment problem by means of auction. We can start with any assignment S and profit-price pair (π, p) satisfying the first two ε-CS conditions (33a) and (33b), and perform a forward auction up to the point where each person is assigned to a (single) distinct object, while satisfying the conditions (33a) and (33b). However, this assignment will not be feasible, because some objects will still be unassigned.

To make further progress, we use a modified reverse auction, which starts with the final results of the forward auction, that is, with an assignment S, where each person is assigned to a single distinct object, and with a pair (π, p) satisfying the first two ε-CS conditions (33a) and (33b). Let us denote by λ the maximal initial person profit,

$$\lambda = \max_{i=1,\ldots,m} \pi_i. \tag{34}$$

The typical iteration, given below, is the same as the one of reverse auction, except that unassigned objects j that bid for a person may not necessarily displace the object assigned to the person but may instead *share* the person with its already assigned object(s). In particular, the algorithm maintains an assignment S, for which each person is assigned to at least one object, and a pair (π, p) satisfying (33a) and (33b); it terminates when all unassigned objects j have been assigned. It will be seen that upon termination, the third ε-CS condition (33c) will be satisfied as well. The scalar λ is kept fixed throughout the algorithm.

Typical Iteration of Modified Reverse Auction for Multiassignment. Select an object j that is unassigned under the assignment S (if all objects are assigned, the algorithm terminates). Find a "best" person i_j such that

$$i_j = \arg\max_{i \in B(j)} \{a_{ij} - \pi_i\}, \tag{35}$$

and the corresponding value

$$\beta_j = \max_{i \in B(j)} \{a_{ij} - \pi_i\}, \tag{36}$$

and find

$$\omega_j = \max_{i \in B(j), i \neq i_j} \{a_{ij} - \pi_i\}. \tag{37}$$

(If i_j is the only person in $B(j)$, we define ω_j to be $-\infty$.) Let

$$\delta = \min\{\lambda - \pi_{i_j}, \beta_j - \omega_j + \varepsilon\}. \tag{38}$$

Add (i_j, j) to the assignment S, set

$$p_j := \beta_j - \delta, \tag{39}$$

$$\pi_{i_j} := \pi_{i_j} + \delta \tag{40}$$

and if $\delta > 0$, remove from the assignment S the pair (i_j, j'), where j' was assigned to i_j under S.

Note that in an iteration, the number of assigned objects increases by one if and only if $\delta = 0$ (which is equivalent to $\pi_{i_j} = \lambda$, since the second term $\beta_j - \omega_j + \varepsilon$ in (38) is always greater than or equal to ε). The following proposition establishes the validity of the method.

Proposition 9. *The modified reverse auction algorithm for the multiassignment problem with integer benefits terminates in a finite number of iterations with an optimal assignment when $\varepsilon < 1/m$.*

Proof In view of Proposition 8, the result will follow once we prove the following:

(a) The modified reverse auction iteration preserves the ε-CS conditions (33), as well as the condition

$$\lambda = \max_{i=1,\dots,m} \pi_i. \tag{41}$$

(b) The algorithm terminates finitely (necessarily with a feasible assignment).

To show (a) above, we use induction. In particular, we show that if the conditions (33) and (41) are satisfied at the start of an iteration, they are also satisfied at the end of the iteration. Indeed, this is easily seen to be true for (33a) and (33b). Equations (33c) and (41) are preserved since we have $\lambda = \max_{i=1,\dots,m} \pi_i$ at the start of the iteration and the only profit that changes is π_{i_j}, which by (38) and (40) is set to something that is less than or equal to λ, and is set to λ if and only if i_j is multiassigned at the end of the iteration.

To show finite termination, we observe that a person i can receive a bid only a finite number of times after the profit π_i is set to λ, since at each of these times the corresponding object will get assigned to i without any object already assigned to i becoming unassigned. On the other hand, by (38) and (40), at an iteration where a person i receives a bid, the profit π_i is either set equal to λ or else increases by at least ε. Since profits are bounded above by λ throughout the algorithm, it follows that each person can receive only a finite number of bids, proving finite termination. □

4.1. Two-sided multiassignment problem. There are several variations of the multiassignment problem and the preceding algorithm. For example, the problem where there is an upper bound α_i on the number of objects person i can be assigned to, that is,

$$\begin{aligned}
&\text{maximize} \quad \sum_{(i,j)\in\mathcal{A}} a_{ij}x_{ij} \\
&\text{subject to} \\
&1 \leqq \sum_{j\in A(i)} x_{ij} \leqq \alpha_i \quad \forall i=1,\dots,m, \\
&\sum_{i\in B(j)} x_{ij} = 1 \quad \forall j=1,\dots,n, \\
&0 \leqq x_{ij} \quad \forall (i,j)\in\mathcal{A},
\end{aligned} \tag{42}$$

where α_i are given integers. This multiassignment problem admits solution by a similar auction algorithm as the preceding one; we will not give the details.

Another interesting variation of the multiassignment problem arises when objects, as well as persons, can be multiassigned, up to a certain limit. This problem, referred to as *two-sided multiassignment*, can be written as

$$\begin{aligned}
&\text{maximize} \quad \sum_{(i,j)\in\mathcal{A}} a_{ij}x_{ij} \\
&\text{subject to} \\
&\sum_{j\in A(i)} x_{ij} \geqq 1 \quad \forall i=1,\dots,m, \\
&1 \leqq \sum_{i\in B(j)} x_{ij} \leqq \alpha_j \quad \forall j=1,\dots,n, \\
&0 \leqq x_{ij} \leqq 1 \quad \forall (i,j)\in\mathcal{A},
\end{aligned} \tag{43}$$

where α_j are given integers less. Note that if $\alpha_j = 1$, this problem is identical to the earlier problem (30).

Again, the above problem can be converted to a minimum cost network flow problem

$$
\begin{aligned}
&\text{minimize} \quad \sum_{(i,j)\in\mathcal{A}} (-a_{ij}x_{ij})\\
&\text{subject to}\\
&\sum_{j\in A(i)} x_{ij} - x_{si} = 1 \quad \forall i = 1,\dots,m,\\
&\sum_{i\in B(j)} x_{ij} - x_{js} = 1 \quad \forall j = 1,\dots,n,\\
(44)\qquad &\sum_{i=1}^{m} x_{si} - \sum_{j=1}^{n} x_{js} = n - m,\\
&0 \leqq x_{ij} \leqq 1 \quad \forall (i,j)\in\mathcal{A},\\
&0 \leqq x_{si} \quad \forall i = 1,\dots,m,\\
&0 \leqq x_{js} \leqq \alpha_j - 1 \quad \forall j = 1,\dots,n,
\end{aligned}
$$

by replacing maximization by minimization, by reversing the sign of a_{ij}, by introducing a supersource node s with supply $n-m$, an arc (s, i) for each person i of zero cost and feasible flow range $[0, \infty)$, and an arc (j, s) for each object node j of zero cost and feasible flow range $[0, \alpha_j - 1]$ (see Fig. 2).

Using duality theory and appropriately redefining the price variables corresponding to the nodes, it can be seen that the corresponding dual problem is

$$
\begin{aligned}
&\text{minimize} \quad \sum_{i=1}^{m} \pi_i + \sum_{j=1}^{n} (p_j + \max\{0, (p_j+\lambda)(\alpha_j - 1)\})\\
(45)\qquad &\qquad\qquad + \sum_{(i,j)\in\mathcal{A}} \max\{0, a_{ij} - p_j - \pi_i\} + (n-m)\lambda\\
&\text{subject to} \quad \lambda \geqq \pi_i \quad \forall i = 1,\dots,m,
\end{aligned}
$$

where λ is the dual price of the supersource node s. The above dual problem is similar to the earlier dual problem (32), with the exception of the cost terms introduced by the upper bounds on the arcs.

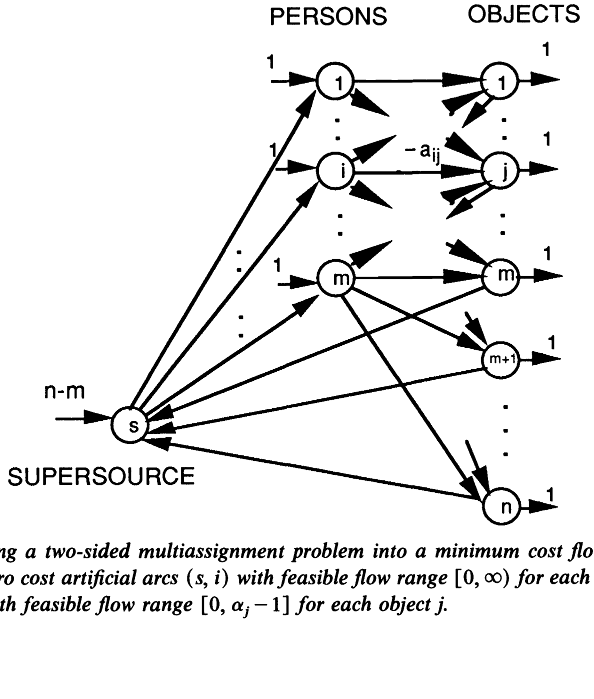


FIG. 2. *Converting a two-sided multiassignment problem into a minimum cost flow problem involving a supersource node s, zero cost artificial arcs (s, i) with feasible flow range $[0, \infty)$ for each person i, and zero cost artificial arcs (j, s) with feasible flow range $[0, \alpha_j - 1]$ for each object j.*

For the two-sided multiassignment problem, we introduce the following ε-CS condition for an assignment S and a pair (π, p).

DEFINITION 4. A multiassignment S and a pair (π, p) are said to satisfy ε-CS for the two-sided multiassignment problem if

$$\pi_i + p_j \geqq a_{ij} - \varepsilon \quad \forall (i,j) \in \mathcal{A}, \tag{46a}$$

$$\pi_i + p_j \leqq a_i \quad \forall (i,j) \in S, \tag{46b}$$

$$\pi_i = \max_{k=1,\ldots,m} \pi_k = \lambda \quad \text{if } i \text{ is multiassigned under } S, \tag{46c}$$

$$p_j + \lambda \geqq 0 \quad \text{if } j \text{ is multiassigned under } S, \tag{46d}$$

$$\text{if } p_j + \lambda > 0 \quad j \text{ must be assigned to } \alpha_j \text{ persons under } S. \tag{46e}$$

Using an argument similar to the proof of Proposition 8, we can establish the following result.

PROPOSITION 10. *Assume that the benefits a_{ij} are integers. If a feasible assignment S satisfies the ε-CS conditions (46) together wih a pair (π, p) for $\varepsilon < 1/m$, then S is optimal for the multiassignment problem.*

Proof. If S is not optimal, there must exist a cycle Y in the equivalent network of Fig. 2 with no repeated nodes along which the assignment S can be modified to result in a new feasible assignment S' with improved primal cost. There are five possible cases: (1) the cycle Y does not include node s; (2) the cycle Y includes s followed by a person and preceded by another person; (3) the cycle Y includes s followed by a person and preceded by an object; (4) the cycle Y includes s followed by an object and preceded by a person; and (5) the cycle Y includes s followed by an object and preceded by another object.

Assume for the moment that the node s is in the cycle and that it is followed and preceded by persons (case (2)); thus, let Y be

$$Y = (s, i_1, j_2, i_2, \ldots, i_{k-1}, j_k, i_k, s).$$

In order for augmentation along Y to result in a feasible assignment, we must have $(i_q, j_q) \in S$, $q = 2, \ldots, k$, $j_{q+1} \in A(i_q)$, $q = 1, \ldots, k-1$; furthermore, i_k must be multiassigned. Because Y has no repeated nodes, we have $k \leqq m$, which, based on the hypothesis, implies $k\varepsilon < 1$.

Augmentation along Y results in replacing the pairs (i_q, j_q), $q = 2, \ldots, k$, by the pairs (i_{q-1}, j_q), $q = 2, \ldots, k$, in the assignment. Since following augmentation along Y, the primal cost is strictly improved, we must have

$$\sum_{q=2}^{k} a_{i_q j_q} + 1 \leqq \sum_{q=2}^{k} a_{i_{q-1} j_q},$$

or equivalently,

$$\sum_{q=2}^{k} (a_{i_q j_q} - p_{j_q}) + 1 \leqq \sum_{q=2}^{k} (a_{i_{q-1} j_q} - p_{j_q}).$$

Using this relation, and the ε-CS conditions (46a) and (46b), we obtain

$$\begin{aligned} \sum_{q=1}^{k} \pi_{i_q} - \pi_{i_1} + 1 \leqq \sum_{q=2}^{k} (a_{i_q j_q} - p_{j_q}) + 1 &\leqq \sum_{q=2}^{k} (a_{i_{q-1} j_q} - p_{j_q}) \\ &\leqq \sum_{q=1}^{k} \pi_{i_q} - \pi_{i_k} + (k-1)\varepsilon. \end{aligned}$$

This yields

$$1 - (k-1)\varepsilon \leqq \pi_{i_1} - \pi_{i_k},$$

which is a contradiction because $k\varepsilon < 1$, and $\pi_{i_1} \leqq \pi_{i_k}$, since i_k is multiassigned (cf. (46c)).

Similarly, assume that node s is preceded and followed by an object in Y (case (5)); thus,

$$Y = (s, j_1, i_1, j_2, i_2, \dots, i_{k-1}, j_k, i_k, j_{k+1}, s).$$

In order for augmentation along Y to produce a feasible assignment, we must have $(i_q, j_q) \in S$, $q = 1, \dots, k$, $j_{q+1} \in A(i_q)$, $q = 1, \dots, k-1$; furthermore, we must have $(i_{q-1}, j_q) \notin S$, $q = 2, \dots, k+1$, j_1 must be multiassigned, and j_{k+1} must be assigned to less than $\alpha_{j_{k+1}}$ persons. Because Y has no repeated nodes, we have $k \leqq m$, which, based on the hypothesis, implies $k\varepsilon < 1$.

Augmentation along Y results in replacing the pairs (i_q, j_q), $q = 1, \dots, k$, by the pairs (i_q, j_{q+1}), $q = 1, \dots, k$, in the assignment; note that since j_1 is multiassigned and j_{k+1} can be assigned to at least one more person, the resulting modified assignment is feasible. Thus, we must have

$$\sum_{q=1}^{k} a_{i_q j_q} + 1 \leqq \sum_{q=1}^{k} a_{i_q j_{q+1}},$$

or equivalently,

$$\sum_{q=1}^{k} (a_{i_q j_q} - \pi_{i_q}) + 1 \leqq \sum_{q=1}^{k} (a_{i_q j_{q+1}} - \pi_{i_q}).$$

Using the above relation and the ε-CS conditions (46a) and (46b), we obtain

$$\sum_{q=1}^{k} p_{j_q} + 1 \leqq \sum_{q=1}^{k} (a_{i_q j_q} - \pi_{i_q}) + 1 \leqq \sum_{q=1}^{k} (a_{i_q j_{q+1}} - \pi_{i_q}) \leqq \sum_{q=1}^{k} p_{j_{q+1}} + k\varepsilon.$$

This yields $1 - k\varepsilon \leqq p_{j_{k+1}} - p_{j_1}$, which is a contradiction because $k\varepsilon < 1$, while by the CS conditions (46d) and (46e), we have $p_{i_{k+1}} = -\lambda \leqq p_{j_1}$ since j_{k+1} is assigned to less than α_{k+1} persons.

The proof for cases (1), (3), and (4) is similar. □

Consider now trying to solve the two-sided multiassignment problem using an auction algorithm. We start from any assignment S that has at most one person assigned to each object and at most one object assigned to each person, and a profit-price pair (π, p) satisfying the first two ε-CS conditions associated with regular auction (cf. (20a) and (20b)). We then use a forward auction algorithm up to the point where each person is assigned to a single (distinct) object, while satisfying the first two ε-CS conditions (46a) and (46b) (condition (46b) will actually be satisfied with equality). Note that this assignment will not be feasible since $m < n$.

At this point, we switch to using a modified reverse auction; denote by λ the maximal initial person profit

$$\lambda = \max_{i=1,\dots,m} \pi_i. \tag{47}$$

Using this value of λ, we can determine which objects have prices p_j indicating that they can be multiassigned; in particular, the ε-CS condition (46e) suggests that any object with price p_j greater than $-\lambda$ should be assigned to as many persons as possible. In order to determine these persons, we use a reverse auction where each unassigned object, and each assigned object with price p_j greater than $-\lambda$ and assigned to less than α_j persons will bid to be assigned to an additional person. This reverse auction is modified in order to satisfy the ε-CS conditions at termination, as follows.

TYPICAL ITERATION OF MODIFIED REVERSE AUCTION FOR TWO-SIDED MULTI-ASSIGNMENT. Select an object j that is unassigned, or is assigned to at least one and less than α_j persons, and has p_j greater than $-\lambda$ (if no such object can be found, the algorithm terminates). If the set $\{i \in B(j) \mid (i,j) \notin S\}$ is empty, set $p_j = -\lambda$ and go to the next iteration. Otherwise, find a "best" person i_j such that

$$i_j = \arg \max_{i \in B(j),(i,j) \notin S} \{a_{ij} - \pi_i\}, \tag{48}$$

and the corresponding value

$$\beta_j = \max_{i \in B(j),(i,j) \notin S} \{a_{ij} - \pi_i\}, \tag{49}$$

and find

$$\omega_j = \max_{i \in B(j), i \neq i_j,(i,j) \notin S} \{a_{ij} - \pi_i\}. \tag{50}$$

(If the set $i \in B(j)$, $i \neq i_j$, $(i,j) \notin S$ is empty, we define ω_j to be $-\infty$.)

If j is unassigned, let

$$\delta = \min\{\lambda - \pi_{i_j}, \beta_j - \omega_j + \varepsilon\}. \tag{51a}$$

Add (i_j, j) to the assignment S, set

$$p_j := \omega_j - \varepsilon, \tag{51b}$$

$$\pi_{i_j} := \pi_{i_j} + \delta, \tag{51c}$$

and if $\delta > 0$, remove from the assignment S the pair (i_j, j'), where j' was assigned to i_j under S.

If j is assigned to at least one and less than α_j persons, and $p_j + \lambda > 0$, let

$$\delta = \min\{\lambda - \pi_{i_j}, \beta_j - \omega_j + \varepsilon, \beta_j + \lambda\}, \tag{52a}$$

and distinguish two cases:

(a) $\delta < \beta_j + \lambda$: In this case, add (i_j, j) to the assignment S, set

$$p_j := \max\{\omega_j - \varepsilon, -\lambda\}, \tag{52b}$$

$$\pi_{i_j} := \pi_{i_j} + \delta, \tag{52c}$$

$$\pi_i := \min\{a_{ij} - \max\{\omega_j - \varepsilon, -\lambda\}, \lambda\} \quad \forall i \text{ such that } (i,j) \in S, \tag{52d}$$

and if $\delta > 0$, remove from S the pair (i_j, j'), where j' was assigned to i_j under S.

(b) $\delta = \beta_j + \lambda$: In this case, set

$$p_j := -\lambda, \tag{53a}$$

$$\pi_{i_j} := \pi_{i_j} + \max\{0, \delta\}, \tag{53b}$$

$$\pi_i := \min\{a_{ij} + \lambda, \lambda\} \quad \forall i \text{ such that } (i,j) \in S, \tag{53c}$$

and, if $\delta > 0$, add (i_j, j) to the assignment S and remove from S the pair (i_j, j'), where j' was assigned to i_j under S.

Note that the above algorithm uses two types of iterations. The first type occurs when the bidding object is unassigned; then the number of unassigned objects decreases by one when δ is zero, which is equivalent to $\pi_i = \lambda$, so that person i can be multi-assigned. The second type of iteration occurs when the bidding object is already assigned, but has price $p_j > -\lambda$; then either j is assigned to an additional person, or else p_j is reduced to the threshold price $-\lambda$.

The following proposition establishes the validity of the method.

PROPOSITION 11. *The modified reverse auction algorithm for the two-sided multiassignment problem with integer benefits terminates in a finite number of iterations with an optimal assignment when $\varepsilon < 1/m$.*

Proof. In view of Proposition 10, the result will follow once we prove the following:

(a) The modified reverse auction iteration preserves the ε-CS conditions (46a)-(46d).

(b) The algorithm terminates finitely (necessarily with a feasible assignment).

(c) Upon termination, the ε-CS condition (46e) must be satisfied.

To show (a) above, we use induction. Let (π, p) and $(\bar\pi, \bar p)$ be the profit-price pair before and after an iteration of the modified reverse auction algorithm, respectively, and let j and i_j be the object and person involved in the iteration. At the beginning of the first iteration, S and (π, p) satisfy

$$\pi_i + p_j \geqq a_{ij} - \varepsilon \quad \forall (i,j) \in \mathcal{A},$$

$$\pi_i + p_j = a_{ij} \quad \forall (i,j) \in S.$$

By construction, we also have

$$\pi_i \leqq \lambda \quad \forall i = 1, \dots, m;$$

furthermore, every person is assigned to exactly one object, and every object is assigned to at most one person. Thus, the ε-CS conditions (46a)-(46d) are satisfied.

Assume that the ε-CS conditions (46a)-(46d) are satisfied at the beginning of an iteration. We consider three cases: (a) object j is currently unassigned, (b) object j is assigned and the bid increment δ satisfies $\delta < \beta_j + \lambda$, and (c) object j is assigned and the bid increment δ satisfies $\delta = \beta_j + \lambda$.

In case (a), by construction we have

$$\bar\pi_{i_j} + \bar p_j \leqq \pi_{i_j} + \beta_j - \omega_j + \varepsilon + p_j \leqq a_{i_j j}.$$

Furthermore, since π_{i_j} does not decrease, the ε-CS condition (46a) will be satisfied for all $k \in A(i_j)$, $k \neq j$. In addition, for any $i' \neq i_j$, $i' \in B(j)$, we have by (50)

$$\pi_{i'} + p_j = \pi_{i'} + \omega_j - \varepsilon \geqq a_{i'j} - \varepsilon,$$

establishing that the ε-CS conditions (46a) and (46b) are satisfied at the end of the iteration. In addition, the ε-CS condition (46c) is guaranteed to be satisfied by (51a) and (51c), and the ε-CS condition (46d) continues to be satisfied, since the prices of multiassigned objects were not affected.

In case (b), the price p_j, and the profits π_{i_j} and π_i, $(i,j) \in S$ are modified. Conditions (46c) and (46d) will be satisfied by the modified profits and prices at the end of the iteration by construction (cf. (52a)-(52d)). Assume $\delta = \beta_j - \omega_j + \varepsilon$; then $-\lambda \leqq \omega_j - \varepsilon$, so

$$\bar p_j = \max\{\omega_j - \varepsilon, -\lambda\} = \omega_j - \varepsilon \leqq p_j,$$

$$\bar\pi_{i_j} + \bar p_j = a_{ij} - \delta - \beta_j + \omega_j - \varepsilon = a_{ij},$$

establishing that the ε-CS condition (46b) holds for the new pair (i_j, j) entering the assignment. Similarly, the ε-CS condition (46b) is satisfied for all $(i,j) \in S$ by construction (cf. (52d)). Since $\bar p_j \leqq p_j$, equation (52d) implies that $\pi_i \leqq \bar\pi_i$ for all i, which in turn implies that the ε-CS condition (46a) is satisfied for all $k \in A(i)$, with $(i,k) \notin S$

and $(i, j) \in S$. Also, the ε-CS condition (46a) is satisfied for all $(i, j) \notin S$, $i \neq i_j$ because (50) implies

$$p_j = \omega_j - \varepsilon \geqq a_{ij} - \pi_i - \varepsilon \quad \forall (i, j) \notin S, i \neq i_j.$$

If, on the other hand, $\delta = \lambda - \pi_i$, then $a_{i_j j} \geqq 0$ and $\omega_j - \varepsilon \leqq a_{i_j j} - \lambda$. Thus, (46b) is satisfied for (i_j, j) at the end of the iteration, since

$$\bar{\pi}_{i_j} + \bar{p}_j = \lambda + \max\{-\lambda, \omega_j - \varepsilon\} \leqq a_{i_j j}.$$

Similarly, the ε-CS condition (46b) is satisfied for all $(i, j) \in S$ by (52d). Since $p_j + \lambda > 0$, p_j is not increased during the iteration. Thus, (52d) implies that $\pi_i \leqq \bar{\pi}_i$ for all i, so that the ε-CS condition (46a) is satisfied for all $k \in A(i)$, with $(i, k) \notin S$, and $(i, j) \in S$. Furthermore, since $p_j \geqq \omega_j - \varepsilon$, the ε-CS condition (46a) is satisfied for all $k \in B(j)$, with $(k, j) \notin S$.

In case (c), assume $\delta = \beta_j + \lambda > 0$. Then, the ε-CS condition (46b) is satisfied for the pair (i_j, j) because

$$\pi_{i_j j} + p_j = a_{i_j j} - \beta_j + \max\{0, \delta\} - \lambda = a_{i_j j}.$$

By assumption, the iteration decreases the price p_j and increases the profit π_{i_j}. Furthermore, (53c) implies that the profits $\pi_i \leqq \bar{\pi}_i$ for all i, so that the ε-CS condition (46a) is satisfied for all $(i, k) \in \mathscr{A}$, $(i, k) \notin S$, $(i, j) \in S$. Equation (53c) also guarantees that the ε-CS condition (46b) will be satisfied at the end of the iteration. If $\delta \leqq 0$, the assignment S is not modified; only the price p_j is decreased and the profits π_i, $(i, j) \in S$ are modified. The ε-CS condition (46b) is satisfied because of (53c); in addition, the ε-CS condition (46a) is satisfied because

$$0 \geqq \beta_j + \lambda \geqq a_{ij} - \pi_i + \lambda = a_{ij} - \pi_i - p_j \quad \forall (i, j) \notin S,$$

and the profits π_i, $i \in B(j)$, are nondecreasing.

The above arguments establish that the ε-CS conditions (46a)–(46d) are preserved by each modified reverse auction iteration. To complete the proof, we must show that the algorithm terminates finitely, and that at termination, the ε-CS condition (46e) is satisfied. It is easy to verify that the number of assigned pairs is nondecreasing, and, as shown above, the profits π_i are nondecreasing, while the prices p_j are nonincreasing. Furthermore, each iteration is guaranteed to produce one (or more) of the following three outcomes: (a) at least one profit π_i increases, (b) one additional pair is assigned, and (c) S remains unchanged, but the price p_j is set to the minimum value $-\lambda$. By construction, the profits π_i cannot rise above λ; furthermore, the prices p_j can only be reduced to $-\lambda$ once per object, and there is a finite maximum number of assigned pairs, which establishes finite termination. To show that the ε-CS condition (46e) is satisfied at termination, note that iterations occur until this condition is satisfied. This completes the proof. □

5. Numerical results. In this section we present some numerical results on the computational performance of the new auction algorithms described in the previous sections. The algorithms have been implemented in FORTRAN and have been compared with state-of-the-art algorithms for every class of problems considered in this paper.

5.1. Symmetric assignment problems. We first tested the two versions of forward/reverse auction (a scaled and an unscaled version) applied to symmetric assignment problems versus two other state-of-the-art codes: a forward auction code and the code of Jonker and Volgenant [JoV87]. The latter, abbreviated as JV code, consists of two phases: an initialization phase, which is based on the naive auction algorithm (the forward auction algorithm with $\varepsilon = 0$), and a sequential shortest path method phase, which assigns the persons that are left unassigned by the initialization phase. It is widely believed that through the combination of the auction and the sequential shortest path algorithms, the JV code is substantially faster than the best pure sequential shortest path and Hungarian assignment codes (for some comparative evidence, see [Ber90]).

Our results for symmetric assignment problems are summarized in Figs. 3-6, where each data point represents an average over ten to thirty random problems with identical characteristics. In Figs. 3-6, a different characteristic (number of nodes, average node degree, and benefit range) of the problem was allowed to vary: the number of nodes in Fig. 3, the average node degree in Fig. 4, and the benefit range in Figs. 5 and 6. Experiments with problems of constant density and varying numbers of nodes and arcs have produced results that are qualitatively intermediate between the results of Figs. 3 and 4. Figures 5 and 6 are similar but they correspond to sparse and fully dense problems, respectively. It can be seen that the unscaled forward/reverse auction is running considerably faster than the other codes. The auction algorithms (remarkably, including the unscaled forward/reverse algorithm) are also quite insensitive to the benefit range; a similar conclusion regarding scaled forward auction was reached in [WeZ91]. Furthermore, all the auction codes run much faster than the JV code except when the problem is quite dense (cf. Fig. 4 when the number of arcs is large). Still, even for fully dense problems the unscaled forward/reverse algorithm is faster than the JV code, except when the benefit range is relatively small ([0, 100] in Fig. 6). There is an explanation for the excellent performance of the JV code for a fully dense problem with a small benefit range. What happens here is that the problem is solved essentially in the naive auction initialization phase of the code and the sequential shortest path phase plays no role. Thus, in this case, the JV code behaves like a very efficient auction algorithm.

In the test problems of Figs. 3-6 the arc benefits are uniformly distributed over the benefit range. In Fig. 7 we tested the effect of a two-level arc benefit distribution

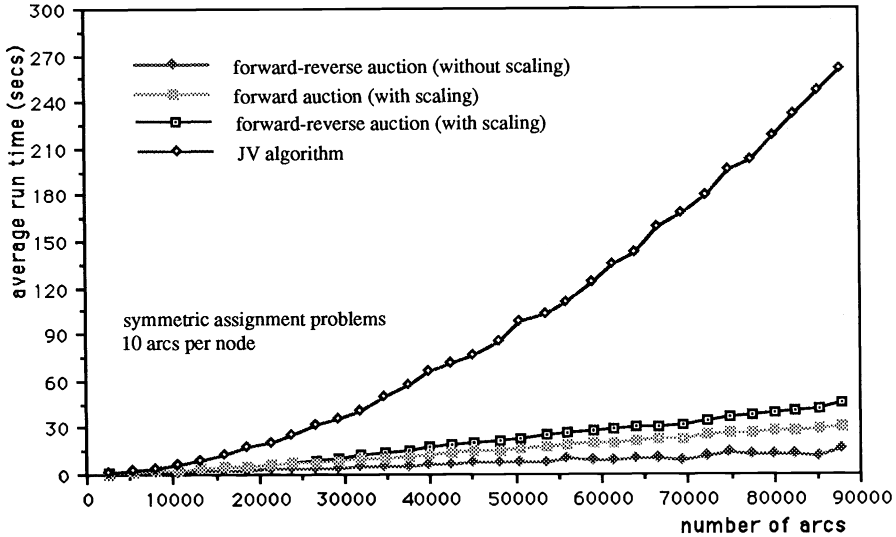


FIG. 3. *Run times for symmetric assignment problems on a* MAC II. *The degree of each person node is* 10. *Each data point represents an average of ten randomly generated problems. The arc benefits are drawn from the range* [0, 1000] *according to a uniform distribution.*

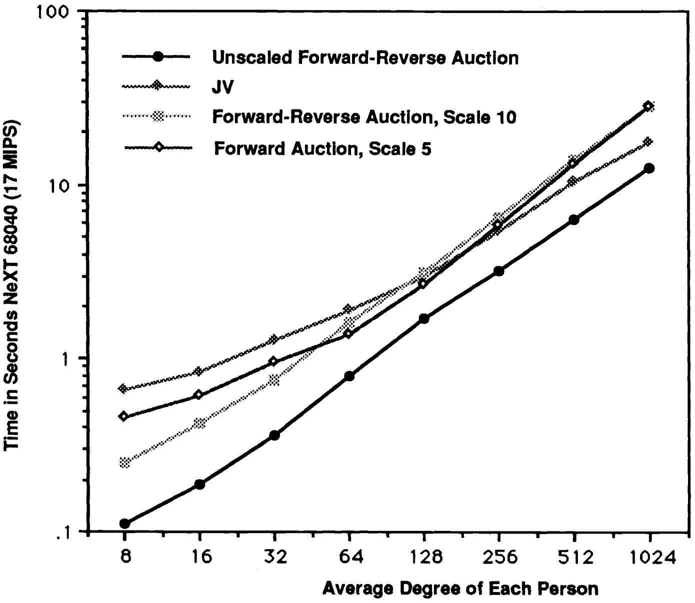


FIG. 4. *Run times for symmetric assignment problems on a* NeXT 68040. *The number of person nodes is* 1024 *and the average node degree varies. Each data point represents an average of* 30 *randomly generated problems. The arc benefits are drawn from the range* [0, 100000] *according to a uniform distribution.*

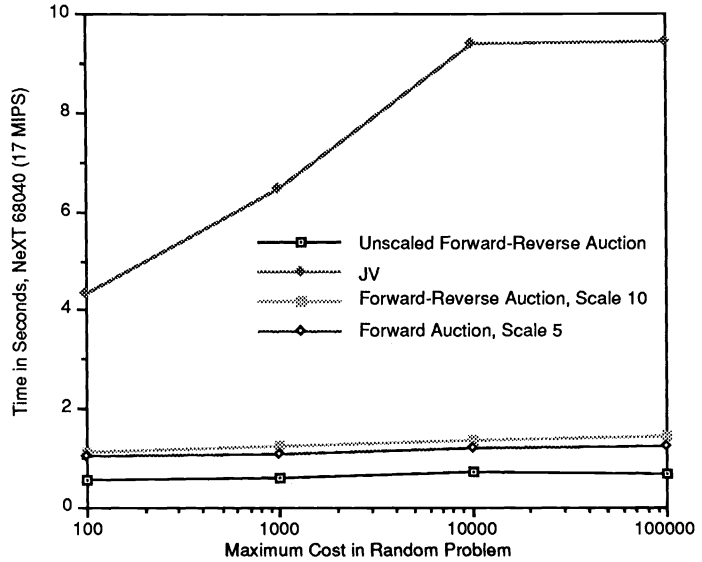


FIG. 5. *Run times for symmetric assignment problems on a* NeXT 68040. *The number of person nodes is* 4000 *and the degree of each is* 8. *Each data point represents an average of* 30 *randomly generated problems. The arc benefits are drawn from the range indicated according to a uniform distribution.*

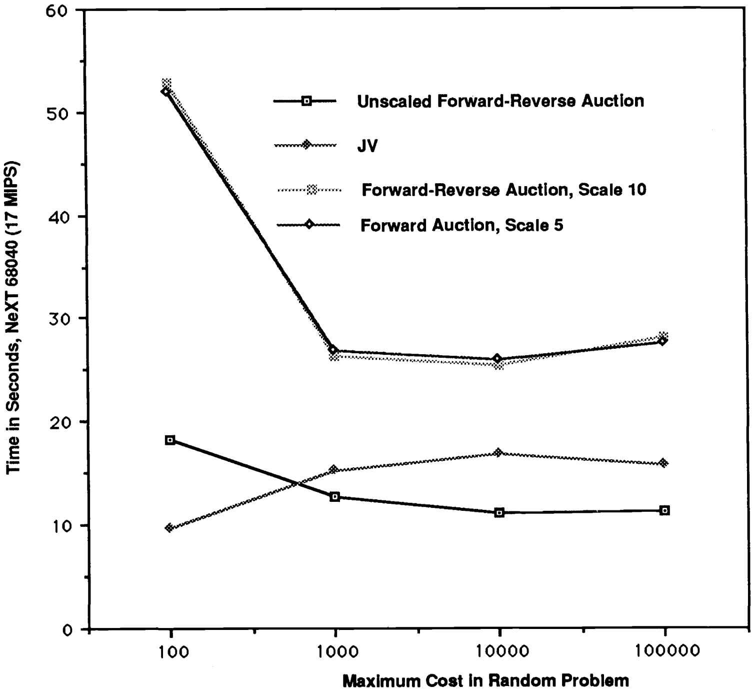


FIG. 6. *Run times for fully dense symmetric assignment problems with* 1024 *persons on a* NeXT 68040. *Each data point represents an average of* 30 *randomly generated problems. The arc benefits are drawn from the range indicated according to a uniform distribution.*

	FR Mean	FR St. Dev.	SFR10 Mean	SFR10 Std.	SF3 Mean	SF3 Std.	SF5 Mean	SF5 Std.	SF10 Mean	SF10 Std.
Easy	0.27	0.14	0.46	0.08	0.51	0.04	0.45	0.04	0.46	0.12
Difficult	0.25	0.05	1.15	0.09	1.77	0.11	1.91	0.32	2.99	0.40

FIG. 7. *Mean and standard deviation of run times for* 30 *experiments with symmetric assignment problems on a* NeXT 68040. *The number of person nodes is* 2000 *and the degree of each is* 8. *For the easy problems, the arc benefits are drawn from the range* [0, 100]. *For the difficult problems,* 80% *of the arc benefits are drawn from the range* [0, 100] *and* 20% *of the arcs have benefit* 100000. *The codes are as follows*: FR: *Unscaled forward/reverse auction.* SFRk: *Scaled forward/reverse auction with* ε*-reduction factor* k. SFk: *Scaled forward auction with* ε*-reduction factor* k.

on the performance of the auction algorithms. Here 80% of the arcs are drawn from the benefit range [0, 100] and 20% of the arcs have benefit 100000. Such arc benefit distributions are generally considered "difficult" for auction algorithms since they tend to stimulate price wars. As mentioned earlier, forward/reverse auction tends to resolve price wars faster than forward auction, and this advantage is manifested dramatically in the results of Fig. 7 for the difficult problems. It should be noted that, in the difficult problem experiments in Fig. 7, the scaling parameters of forward auction were optimized. This optimization resulted in an improvement of roughly a factor of 6 in run time over the codes with the default scaling parameters given in [Ber91].

Except on artificially constructed examples, we have found the performance of unscaled forward/reverse auction remarkably robust. Indeed, it is only in very special classes of problems that the performance of this algorithm is significantly hampered by the occurrence of price wars. The paper [Cas92] provides a comprehensive computational study of the performance and the robustness of the forward/reverse algorithms for a variety of problem structures.

5.2. Asymmetric assignment problems. The new forward/reverse auction algorithm for asymmetric one-on-one assignment problems was tested versus the asymmetric version of the JV algorithm. We performed tests with two types of randomly generated problems. For both classes of problems, each person node has 10 incident arcs. However, in the first class of problems, the end nodes of the arcs and the arc benefits were generated in a completely random fashion. Figure 8 gives the running times of the scaled forward reverse auction algorithm, the unscaled auction algorithm, and the JV code for this class of problems. A comparison of this figure with Fig. 3 indicates that this class of problems is relatively "easy" for all methods. In particular, price wars were very infrequent, and the unscaled auction algorithm outperformed its scaled version as well as the JV code by a large margin.

The second class of asymmetric assignment problems was specially designed to create price wars by making some nodes difficult to assign. In particular, we introduced two levels of arc benefits that are different by approximately three orders of magnitude. This kind of bipartite problems is quite typical in many applications where nearly infeasible problems frequently arise, e.g., in target tracking applications where potentially false measurements or tracks cannot be matched to confirmed targets. Figure 9 gives the run times of the various codes versus the number of arcs. It can be seen that the run times of both (scaled and unscaled) auction algorithms again grow almost linearly with the number of arcs, but the scaled auction algorithm outperforms the unscaled one by an almost constant factor of 25. The run time of the JV algorithm grows almost quadratically, as it did for symmetric problems. The performance of scaled auction is significantly better than that of the JV algorithm, but unscaled auction is worse than JV in these experiments. Note, however, that our unscaled auction for asymmetric assignment problems does not involve a reverse portion. The initial object prices in all runs were zero, and as mentioned in § 3, upon termination of the forward

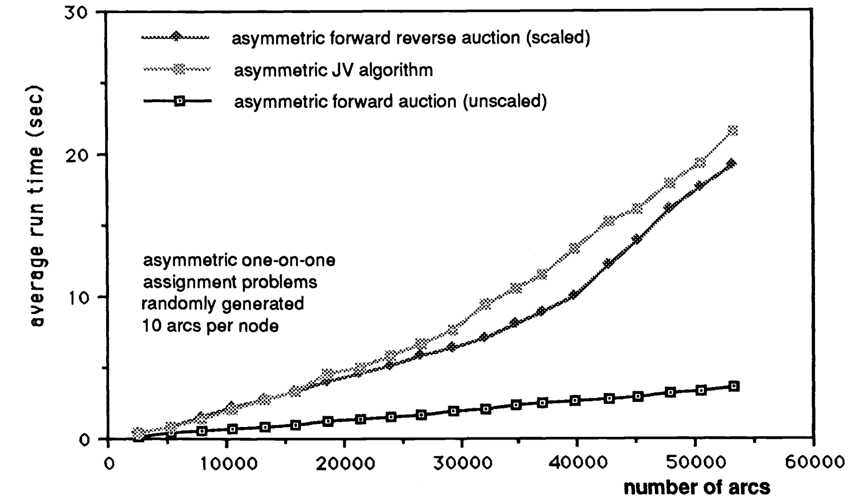


FIG. 8. *Run times for "easy" asymmetric assignment problems on a* MAC II. *The degree of each person node is* 10 *and the arc benefit range is* [0, 1000]. *Each data point represents an average of ten randomly generated problems. The arc benefits are drawn from the benefit range according to a uniform distribution.*

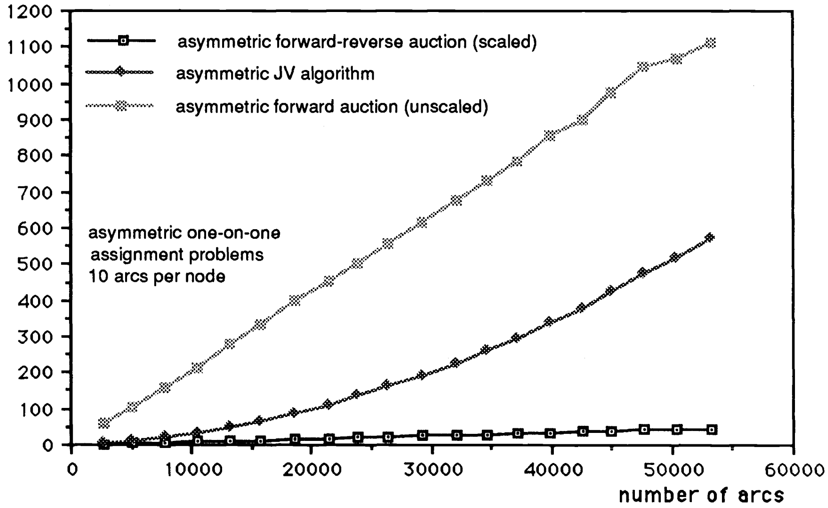


FIG. 9. *Run times for "difficult" asymmetric assignment problems on a* MAC II. *The degree of each person node is* 10. *Each data point represents an average of ten randomly generated problems. The standard deviation for each point was typically less than* 5% *of the corresponding run time. There are two levels of arc benefits that are different by approximately three orders of magnitude.*

auction part of the code, all the ε-CS conditions are satisfied, and the reverse auction part is never used. Thus the mechanism that helped the forward/reverse unscaled auction algorithm to avoid price wars in the difficult problems of Fig. 7 was not employed in the unscaled asymmetric auction algorithm for the difficult problems of Fig. 9.

5.3. Multiassignment problems. Next, we tested the multiassignment auction algorithm (abbreviated *multiauction*) for one-sided asymmetric problems versus the state-of-the-art relaxation code RELAX [BeT88], which solves the equivalent minimum cost network flow problems, and versus the state-of-the-art primal-simplex code NETFLO due to Kennington and Helgason [KeH80], which solves the same equivalent network flow problems. We do not know of any specialized assignment code (including the Hungarian or mixed auction/Hungarian code such as JV) that can be easily modified to handle multiassignment problems. It was found that the NETFLO run times were approximately 40 times higher than those of RELAX for about 5000 arcs and that factor was growing for higher numbers of arcs. In Fig. 10 we show the solution times versus the number of arcs for RELAX and the new multiassignment algorithm for a sequence of randomly generated asymmetric problems with a fixed number of arcs per node. The run times for multiauction grow almost linearly as the number of arcs increases, and this behavior is very consistent across all runs. Furthermore, multiauction is approximately four times faster than RELAX, whose run times also grow roughly linearly but with quite a bit more fluctuation.

Scaling is important for multiassignment problems as it is for one-on-one assignment problems. Although this may not be apparent for randomly generated problems, it is frequently needed in applications, particularly for nearly infeasible problems. In Fig. 11 we show run time results of scaled and unscaled one-sided multiauction algorithms applied to a practical dynamic multi-target tracking and correlation problem over a fixed period of time. At each point in time a sensor's scan produces a set of measurements of target positions that are to be matched with another set of existing

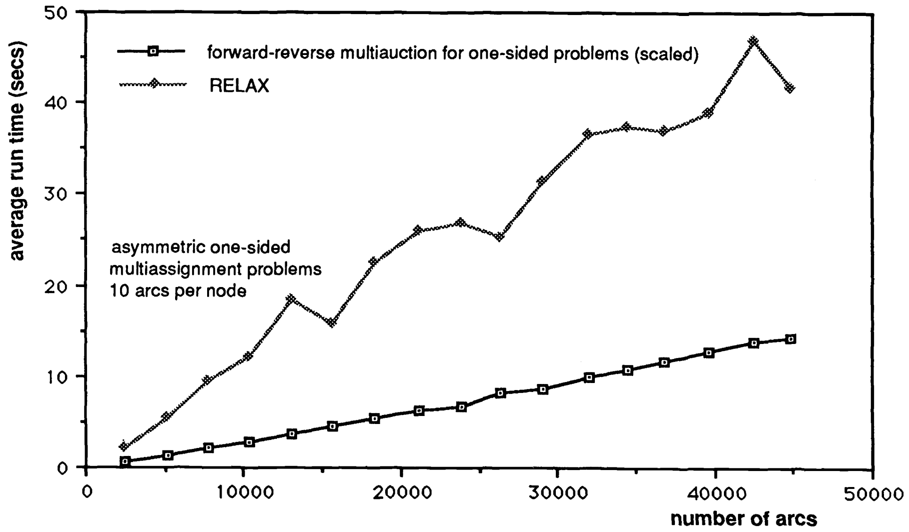


FIG. 10. *Run times for one-sided asymmetric multiassignment problems on a* MAC II. *Each data point represents an average of ten randomly generated problems. The degree of each person node is* 10 *and the arc benefit range is* [0, 1000]. *The standard deviation for each point in the multiauction curve was typically around* 5% *of the corresponding run time, while for the* RELAX *curve it was between* 10 *and* 30%.

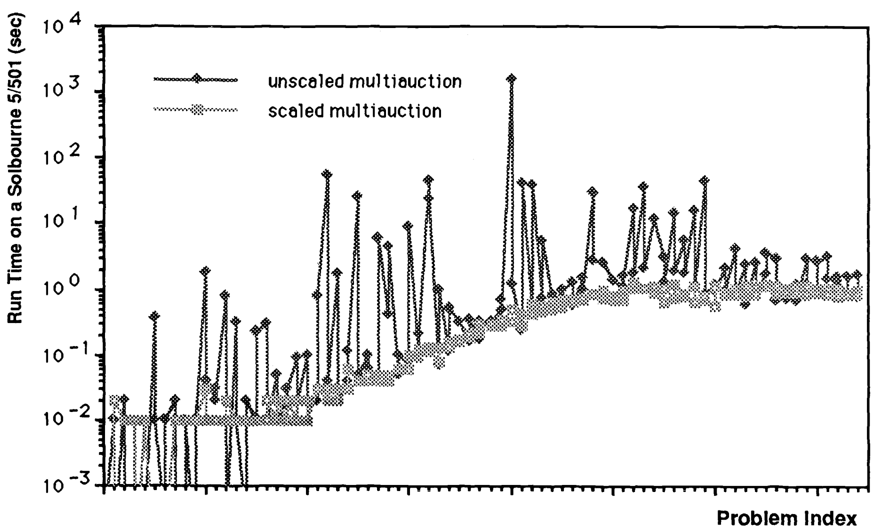


FIG. 11. *Run times for practical one-sided asymmetric multiassignment problems arising in multitarget tracking on a Solbourne* 5/501 *computer, rated at* 22 MIPS. *Each point on the horizontal axis corresponds to a different problem. The number of object nodes ranges from* 10 *to* 125, *and the number of arcs ranges from* 100 *to* 10,000.

tracks by making use of the multiauction algorithms. It can be seen from Fig. 11 that the unscaled multiauction performs worse and far less consistently than the scaled version for this class of practical problems.

Finally, the new two-sided multiassignment algorithm was tested versus RELAX for randomly generated problems. The results, shown in Fig. 12, indicate a substantial speed advantage for the new multiauction algorithm.

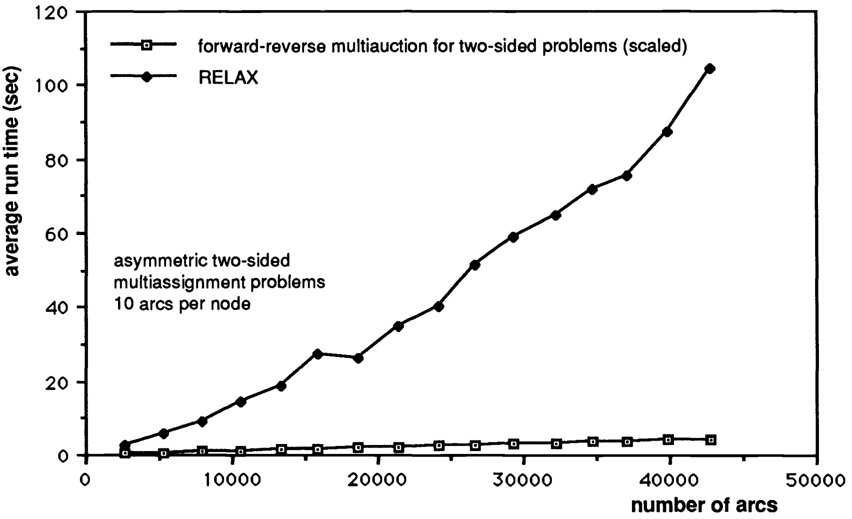


FIG. 12. *Run times for two-sided asymmetric multiassignment problems (cf. (43)) on a* MAC II. *The degree of each person node is* 10 *and the arc benefit range is* [0, 1000]. *In these problems, the upper flow bounds* α_j *are all* ∞.

REFERENCES

[Ber79] D. P. BERTSEKAS, *A distributed algorithm for the assignment problem*, Laboratory for Information and Decision Systems Working Paper, Massachusetts Institute of Technology, Cambridge, MA, March 1979.

[Ber81] ———, *A new algorithm for the assignment problem*, Math. Programming, 21 (1981), pp. 152–171.

[Ber85] ———, *A distributed asynchronous relaxation algorithm for the assignment problem*, Proc. 24th IEEE Conf. Decision and Control, 1985, pp. 1703–1704.

[Ber86a] ———, *Distributed asynchronous relaxation methods for linear network flow problems*, Laboratory for Information and Decision Systems Report LIDS P-1606, Massachusetts Institute of Technology, Cambridge, MA, Nov. 1986.

[Ber86b] ———, *Distributed relaxation methods for linear network flow problems*, Proc. 25th IEEE Conf. Decision and Control, 1986, pp. 2101–2106.

[Ber88] ———, *The auction algorithm: A distributed relaxation method for the assignment problem*, Ann. Oper. Res., 14 (1988), pp. 105–123.

[BeC89a] D. P. BERTSEKAS AND D. A. CASTAÑON *The auction algorithm for transportation problems*, Ann. Oper. Res., 20 (1989), pp. 67–96.

[BeC89b] ———, *The auction algorithm for the minimum cost network flow problem*, Laboratory for Information and Decision Systems Report LIDS-P-1925, Massachusetts Institute of Technology, Cambridge, MA, Nov. 1989.

[BeC89c] ———, *Parallel synchronous and asynchronous implementations of the auction algorithm*, Alphatech Report, Burlington, MA, Nov. 1989; also in Parallel Comput., 17 (1991), pp. 707–732.

[Ber90] D. P. BERTSEKAS, *The auction algorithm for assignment and other network flow problems: A tutorial*, Interfaces, 20 (1990), pp. 133–149.

[Ber91] ———, *Linear Network Optimization: Algorithms and Codes*, M.I.T. Press, Cambridge, MA, 1991.

[Ber92a] ———, *Auction algorithms for network flow problems: A tutorial introduction*, Comput. Optim. Appl., 1 (1992), pp. 7–66.

[Ber92b] ———, *Mathematical equivalence of the auction algorithm for assignment and the* ε*-relaxation (preflow-push) method for min cost flow*, LIDS Report P-2147, Massachusetts Institute of Technology, Cambridge, MA, Nov. 1992.

[BeE88] D. P. BERTSEKAS and J. ECKSTEIN, *Dual coordinate step methods for linear network flow problems*, Math. Programming, Ser. B, 42 (1988), pp. 203–243.

[BeT85] D. P. Bertsekas and P. Tseng, *Relaxation methods for minimum cost ordinary and generalized network flow problems*, Laboratory for Information and Decision Systems Report LIDS P-1462, Massachusetts Institute of Technology, Cambridge, MA, May 1985; also in Oper. Res. J., 36 (1988), pp. 93-114.

[BeT88] ———, RELAX: *A computer code for minimum cost network flow problems*, Ann. Oper. Res., 13 (1988), pp. 127-190.

[BeT89] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[Bla86] S. S. Blackman, *Multi-target Tracking with Radar Applications*, Artech House, Dedham, MA, 1986.

[CSW89] D. Castañon, B. Smith, and A. Wilson, *Performance of parallel assignment algorithms on different multiprocessor architectures*, Alphatech Inc. Report, Burlington, MA, 1989.

[Cas92] D. A. Castañon, *Reverse auction algorithms for assignment problems*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1992.

[Dan63] G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963.

[Gol87] A. V. Goldberg, *Efficient graph algorithms for sequential and parallel computers*, Tech. Rep. TR-374, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, Feb., 1987.

[GoT90] A. V. Goldberg and R. E. Tarjan, *Solving minimum cost flow problems by successive approximation*, Math. Oper. Res., 15 (1990), pp. 430-466.

[JoV87] R. Jonker and A. Volegnant, *A shortest augmenting path algorithm for dense and sparse linear assignment problems*, Computing, 38 (1987), pp. 325-340.

[KKZ89] D. Kempa, J. Kennington, and H. Zaki, *Performance characteristics of the Jacobi and Gauss-Seidel versions of the auction algorithm on the Alliant* FX/8, Report OR-89-008, Dept. of Mechanics and Industrial Engineering, Univ. of Illinois, Urbana, IL, 1989.

[KeH80] J. Kennington and R. Helgason, *Algorithms for Network Programming*, John Wiley, New York, 1980.

[PaS82] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[PhZ88] C. Phillips and S. A. Zenios, *Experiences with large scale network optimization on the connection machine*, Report 88-11-05, Dept. of Decision Sciences, The Wharton School, Univ. of Pennsylvania, Philadelphia, PA, Nov., 1988.

[Roc84] R. T. Rockafellar, *Network Flows and Monotropic Programming*, Wiley-Interscience, New York, 1984.

[Sch90] B. L. Schwartz, *A computational analysis of the auction algorithm*, unpublished manuscript, 1990.

[WeZ90] J. Wein and S. A. Zenios, *Massively parallel auction algorithms for the assignment problem*, Proc. Third Symposium on the Frontiers of Massively Parallel Computation, MD, Nov. 1990.

[WeZ91] ———, *On the massively parallel solution of the assignment problem*, J. Parallel Distrib. Comput., 13 (1991), pp. 228-236.

[Zak90] H. Zaki, *A comparison of two algorithms for the assignment problem*, Report ORL 90-002, Dept. of Mechanical and Industrial Engineering, Univ. of Illinois, Urbana, IL.

SIAM J. OPTIMIZATION
Vol. 3, No. 2, pp. 298–321, May 1993

005

A GLOBALLY AND SUPERLINEARLY CONVERGENT ALGORITHM FOR CONVEX QUADRATIC PROGRAMS WITH SIMPLE BOUNDS*

THOMAS F. COLEMAN† AND LAURIE A. HULBERT‡

Abstract. A globally and superlinearly convergent algorithm for solving convex quadratic programs with simple bounds is presented. The algorithm is developed using a new formulation of the problem: the minimization of an unconstrained piecewise quadratic function that has the same optimality conditions as the original problem. The major work at each iteration is the Cholesky factorization of a positive definite matrix with the size and structure of the Hessian of the quadratic. Hence, the algorithm is suitable for solving large sparse problems and for implementation on parallel computers. The numerical results indicate that the new approach has promise.

Key words. quadratic programming, interior point methods, simple bounds, box constraints, large sparse minimization

AMS subject classifications. 90C20, 65K05

1. Introduction. In this paper, we present a new algorithm for solving the problem

$$\begin{aligned} &\min \tfrac{1}{2}x^T Ax + b^T x \\ &-1 \le x \le 1, \end{aligned} \tag{1}$$

where A is an $n \times n$ symmetric positive definite matrix. In theory our approach can be applied to problems with general upper and lower bounds after a simple transformation to yield form (1). In practice this works without difficulty provided the ranges are not extreme. When there are large ranges, numerical difficulties may prevent an accurate solution. However, we believe that in many practical instances it is often the case that reasonable feasibility ranges are known in advance.

Many algorithms, both finite and infinite, have been proposed for (1). *Finite* algorithms (assuming exact arithmetic), usually involving pivoting and determination of an "active-set," are the most common. Recent contributions include: Björck [1], Coleman and Hulbert [3], Dembo and Tulowitzki [5], Júdice and Pires [9], Lötstedt [11], Moré and Toraldo [12], Öreborn [14], O'Leary [13], and Yang and Tolle [17].

Following Karmarkar's [10] development of an (infinite) "interior-point" algorithm for linear programming, there has been increased interest in infinite interior-point algorithms for quadratic programs. Interior-point algorithms for quadratic programs are typically based on affine scaling, path following or barrier functions, potential reduction, or projection techniques and are in general simpler to implement than active-set methods because they require less data structure manipulation. For a discussion of recent interior-point algorithms for this and other quadratic programs, see the survey paper by Ye [19]. Some of these interior-point algorithms have polynomial time bounds,[1] but

*Received by the editors February 10, 1990; accepted for publication (in revised form) January 22, 1992. This research was partially supported by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under grant DE-FG02-86ER25013.A000 and by the Computational Mathematics Program of the National Science Foundation under grant DMS-8706133.

†Computer Science Department, Cornell University, Ithaca, New York 14853.

‡Department of Mathematics and Computer Science, James Madison University, Harrisonburg, Virginia 22807.

[1] One can also take a finite view of such algorithms, assuming integer data, exact arithmetic, and a formal final "rounding" to the exact solution. This view leads to a complexity analysis; e.g., is the number of steps bounded by a polynomial in the size of the problem? This is not our concern in this paper.

their asymptotic rates of convergence have not been studied. The affine scaling method proposed by Ye [18] which is similar to but simpler than the polynomial algorithm of Ye and Tse [20] and has no proven polynomial time bound, displays linear convergence in practice. While few numerical results are available for the recent polynomial algorithms, those presented by Han, Pardalos, and Ye [8] show that the performance of their polynomial algorithm is more consistent than that of active-set algorithms.

We present a new infinite algorithm here that is not an interior-point method. In general, infeasible iterates are generated. Our algorithm is globally and superlinearly convergent; however, we do not claim that it has a polynomial time bound. We develop our algorithm using a new formulation of the problem: the minimization of an unconstrained piecewise quadratic function that has the same optimality conditions as the original problem. Our algorithm has similarities to an l_1 penalty function method, and is quite similar in development to the quadratically convergent affine scaling method for the linear l_1 problems of Coleman and Li [4]. The major work at each iteration of our algorithm is the Cholesky factorization of a positive definite matrix with the size and structure of the matrix A. Hence our algorithm is suitable for solving large sparse problems and for implementation on parallel computers.

There are three basic ideas underlying our new approach. The major purpose of this paper is to expose these ideas and to begin to explore their potential in constrained optimization. The first idea is the observation, detailed below, that a simple transformation changes (1) into an unconstrained minimization problem involving a piecewise quadratic function $f(y)$. This allows for the possibility of using unconstrained minimization strategies. The second idea, discussed in §2, is that there is a well-defined unconstrained Newton process in a neighborhood of the solution. This Newton process is defined with respect to the optimality conditions. The third idea is the definition of a descent direction, and a piecewise line search procedure that ultimately leads to full Newton steps, thereby ensuring superlinear convergence.

The paper is organized as follows. In the rest of this section, we present our new formulation of the problem and introduce some notation and definitions. In §2, we describe and motivate our algorithm. We prove global convergence in §3 and superlinear convergence in §4. Section 5 contains numerical results and a discussion of the behavior of the algorithm. Finally, in §6, we discuss possible improvements and make some concluding remarks.

1.1. A related problem. Consider the quadratic program (1). Let $q_x(x) = \frac{1}{2}x^T Ax + b^T x$, and hence $\nabla q_x(x) = Ax + b$. If we assume that x^* satisfies $\nabla q_x(x^*)_i \neq 0$ for all i such that $|x_i^*| = 1$, then the following conditions are sufficient to guarantee that x^* is a local minimum of (1):

$$\text{feasibility:} \quad -1 \leq x^* \leq 1,$$

$$\text{first order:} \quad \begin{cases} \nabla q_x(x^*)_i = 0 & \text{if } -1 < x_i^* < 1, \\ \nabla q_x(x^*)_i < 0 & \text{if } x_i^* = 1, \\ \nabla q_x(x^*)_i > 0 & \text{if } x_i^* = -1. \end{cases}$$

Now for a vector v, define the vector-valued function sign(v), where

$$\text{sign}(v)_i = \begin{cases} 1 & \text{if } v_i \geq 0, \\ -1 & \text{if } v_i < 0. \end{cases}$$

Then letting $d_i = x_i^* + \text{sign}(\nabla q_x(x^*))_i$, and $D = \text{diag}(d_i)$, we can express the first-order condition as

$$D\nabla q_x(x^*) = 0. \tag{2}$$

Now consider the following piecewise quadratic minimization problem:

$$\begin{aligned} \min f(y) &= \tfrac{1}{2}y^T A^{-1} y + y^T A^{-1} b + \|y\|_1 \\ &= q_y(y) + \|y\|_1, \end{aligned} \tag{3}$$

where $q_y(y) = \frac{1}{2}y^T A^{-1} y + y^T A^{-1} b$. The following conditions are sufficient to guarantee that y^* is a minimum of (3) [2]: there exists a vector λ^* such that

$$A^{-1}y^* + A^{-1}b + \sum_{i \ni y_i^* \neq 0} \text{sign}(y_i^*)e_i = - \sum_{i \ni y_i^* = 0} \lambda_i^* e_i, \qquad -1 \leq \lambda^* \leq 1.$$

We can reformulate these conditions into the following equivalent conditions: there exists a vector λ^* such that

$$\begin{aligned} &Y^*(-\lambda^* + \text{sign}(y^*)) = 0, \\ &\lambda^* = -(A^{-1}y^* + A^{-1}b), \qquad -1 \leq \lambda^* \leq 1, \end{aligned} \tag{4}$$

where $Y^* = \text{diag}(y^*)$. Thus if we equate λ^* with x^* and hence y^* with $-\nabla q_x(x^*)$, then it is apparent that (4) is equivalent to (2) plus feasibility.

This new formulation gives us a new perspective from which to approach solving (1) and this is the view we take in this paper.

For convenience in what follows, we sometimes switch between the original variables x and $-\nabla q_x(x)$ and the new variables λ and y:

$$x = \lambda, \qquad y = -\nabla q_x(x). \tag{5}$$

In general, we develop our algorithm and prove things about it in the λ and y variables and describe the characteristics of the quadratic programs in the x and $\nabla q_x(x)$ variables.

1.2. Some notation and definitions. In what follows, subscripts denote vector and matrix components and superscripts denote iteration number. We omit superscripts whenever the iteration number is clear or irrelevant. For any vector v, the matrix $\text{diag}(v)$ is a diagonal matrix whose diagonal elements are the components of v. If V is a matrix, let $|V|$ be the matrix whose ijth element is $|v_{ij}|$.

For any point y with $y_i \neq 0$ for all i, $\nabla f(y)$ is defined and $\nabla f(y) = A^{-1}y + A^{-1}b + \text{sign}(y)$. Given y and s, define a *breakpoint* of f along s to be any α where $f(y + \alpha s)$ is nondifferentiable, i.e., $(y + \alpha s)_i = 0$ for some i. For $\alpha > 0$ define $S(\alpha, y, s)$ to be the set of indices to breakpoints along s that occur at or before α, i.e.,

$$S(\alpha, y, s) = \{i \mid 0 < -y_i/s_i \leq \alpha\}. \tag{6}$$

Define $\sigma(\alpha, y, s) = \text{sign}(y + \alpha s)$. For any direction s, define

$$g(\alpha, y, s) = \lim_{\eta \to \alpha^+} \nabla f(y + \eta s).$$

Notice that if $(y + \alpha s)_i \neq 0$ for all i, then $g(\alpha, y, s) = \nabla f(y + \alpha s)$. For conciseness, we write $g(\alpha)$, $S(\alpha)$, and $\sigma(\alpha)$ when y and s are clear from context, and, in particular, $g^k(\alpha)$, $S^k(\alpha)$, and $\sigma^k(\alpha)$ when $y = y^k$ and $s = s^k$. Also, since we use it so frequently, we let σ denote $\sigma(0)$, i.e., $\sigma = \text{sign}(y)$.

A point satisfying $|x_i| = 1$ and $\nabla q_x(x)_i = 0$ for some i is called a *degenerate* point. We call a quadratic program of the form in (1) *nondegenerate on a closed bounded set* C if at every point $x \in C$ either $|x_i| \neq 1$ or $\nabla q_x(x)_i \neq 0$.

The nondegeneracy assumption. Given a closed bounded set C, the nondegeneracy assumption, with respect to C, is that at every point $x \in C$ either $|x_i| \neq 1$ or $\nabla q_x(x)_i \neq 0$.

2. The algorithm. Problem (3) is an unconstrained optimization problem; therefore, a descent direction algorithm can be developed without regard to maintaining feasibility. On the other hand, $f(y)$ is not everywhere differentiable due to the l_1-term $\|y\|_1$. The challenge is to deal with this piecewise nature of f. In response, our algorithm restricts iterates to differentiable points; i.e., $y_i^k \neq 0$ for all iterations k and components i.

2.1. The search direction. From (4), we see that a solution[2] to (3) is also a zero of

$$F(y) = Y(A^{-1}y + A^{-1}b + \text{sign}(y)) = 0. \tag{7}$$

Although F is not differentiable whenever $y_i = 0$ for some i, at all other points $F(y) = Y\nabla f(y)$ and is twice continuously differentiable. This naturally suggests using Newton's method, at least in a neighborhood of y^*. Where it is defined, the Jacobian of $F(y)$ is

$$J(y) = YA^{-1} + \text{diag}(\nabla f(y)),$$

and thus the Newton step for F at y is

$$s_N = -(YA^{-1} + \text{diag}(\nabla f(y)))^{-1} Y \nabla f(y). \tag{8}$$

The following lemma shows that in a neighborhood of the solution of (3), the Newton step for F is a descent direction for $f(y)$. This is not an obvious result since the Newton process does not come directly from f but from the nonlinear system of equations (7). The idea behind the proof is that $\nabla f(y)_i / y_i$ either converges to zero or to (positive) infinity as $y \to y^*$. Specifically, if $y_i^* \neq 0$, then $\nabla f(y)_i / y_i$ converges to 0; if $y_i^* = 0$, then $\nabla f(y)_i / y_i$ converges to $+\infty$. Consequently, the matrix $(A^{-1} + \text{diag}(\nabla f(x))/Y)$ is positive definite in a neighborhood of y^*; therefore, by (8), s_N will be a descent direction.

LEMMA 2.1. *Assume nondegeneracy of* (1) *at the solution. Then, there exists* $\epsilon > 0$ *such that whenever* $y_i \neq 0$ *for all* i *and* $\|y - y^*\| < \epsilon$, *we have* $-s_N^T \nabla f(y) > 0$.

Proof. Rewriting (8) as

$$s_N = -(A^{-1} + Y^{-1}\text{diag}(\nabla f(y)))^{-1} \nabla f(y), \tag{9}$$

we can see that if $\nabla f(y)_i / y_i > -(1/\|A\|_2)$ for all i, where $(1/\|A\|_2)$ is the smallest eigenvalue of A^{-1}, then $-s_N^T \nabla f(y) > 0$. Set

$$\epsilon = \frac{1}{2} \min\left(\frac{xdg}{\|A^{-1}\|_2}, \frac{ydg}{\|A\|_2 \|A^{-1}\|_2} \right),$$

[2] Recall: We have assumed A is symmetric positive definite, so A^{-1} exists.

where

$$ydg = \min_{\{i:y_i^* \neq 0\}} |y_i^*| \quad \text{and} \quad xdg = \min_{\{i:y_i^*=0\}} (1 - |x_i^*|).$$

Assume $\|y - y^*\|_2 < \epsilon$ and $y_i \neq 0$ for all i.

If $y_i^* \neq 0$ then $\text{sign}(y^*)_i = \text{sign}(y)_i$ by our choice of ϵ, so

$$\begin{aligned} |\nabla f(y)_i| &= |\nabla f(y)_i - \nabla f(y^*)_i| \\ &= |x_i - x_i^*| \\ &\leq \|x - x^*\|_2 \\ &\leq \|A^{-1}\|_2 \|y^* - y\|_2 \\ &< \|A^{-1}\|_2\, \epsilon \\ &\leq ydg/(2\|A\|_2). \end{aligned}$$

Since

$$|y_i| \geq ydg - \epsilon \geq ydg/2,$$

we have

$$\left| \frac{\nabla f(y)_i}{y_i} \right| < \frac{1}{\|A\|_2}.$$

Hence $\nabla f(y)_i / y_i > -(1/\|A\|_2)$ if $y_i^* \neq 0$.

If $y_i^* = 0$ then

$$|\, |x_i| - |x_i^*|\, | \leq |x_i - x_i^*| < \|A^{-1}\|_2\, \epsilon \leq xdg/2,$$

so

$$|x_i| < |x_i^*| + xdg/2 \leq 1 - xdg + xdg/2 < 1.$$

And since $\nabla f(y)_i = (\sigma_i - x_i)$, we conclude that $\sigma_i = \text{sign}(\nabla f(y))_i$ and hence $\nabla f(y)_i / y_i > 0$. Thus $\nabla f(y)_i / y_i > -(1/\|A\|_2)$ for all i, so we are done. □

Of course, the Newton step may not be a descent direction far from the solution. Therefore, we consider a "modified" Newton step. Specifically, we choose a step of the form

$$s = -(|Y|A^{-1} + R)^{-1} |Y| \nabla f(y), \tag{10}$$

where R is a diagonal matrix satisfying $r_{ii} > 0$ for all i. Thus we have the following lemma.

LEMMA 2.2. *For any diagonal matrix R with positive diagonal entries, the search direction s, defined by* (10), *is a descent direction, i.e.*, $-s^T \nabla f(y) > 0$.

In order for s to approach the Newton step, we choose $R = \text{diag}(r)$, where

$$r_i = \theta + (1 - \theta)|\nabla f(y)_i|, \tag{11}$$

$\theta \geq 0$, and $\theta = 0$ only at the optimal solution y^*. We define η to quantify the nonoptimality of the current point,

$$\eta = \rho \|Y \nabla f(y)\|_1 + \sum_i \max_i((|\lambda_i| - 1), 0),$$

where $\rho = 1/\|Y\nabla f(y)\|_1$, evaluated at a "typical" value of y. Our choice of a "typical" value of y is $-(A \cdot \text{sign}(-b) + b)$. Then we choose θ to be between 0 and some small constant $c_1 \in (0, 1)$ by setting

$$\theta = c_1\eta/(0.99 + \eta).$$

Notice that either $\nabla f(y^*)_i = 0$ or $\text{sign}(\nabla f(y^*))_i = \sigma_i$, so R, as defined by (11), approaches $\Sigma \cdot \text{diag}(\nabla f(y))$, where Σ =diag(σ_i), thus ensuring that the Newton step is approached. We can prove the following useful lemma about R.

LEMMA 2.3. *If r is defined by* (11), *then for all i, $r_i = 0 \iff \theta = 0$ and $\nabla f(y)_i = 0$.*

Proof. By definition, $r_i = \theta + (1-\theta)|\nabla f(y)_i|$. Since $0 \le \theta \le 1$, each term is greater than or equal to zero. Thus $r_i = 0$ if and only if $\theta = 0$ and $\nabla f(y)_i = 0$. □

2.2. The line search. The basic iteration in our overall procedure has the form

$$y^{k+1} = y^k + \alpha^k s^k, \tag{12}$$

where α^k is the step length, determined after computing the search direction s^k. Before describing this line search procedure, we introduce some notation and describe the geometry of the line search (we drop the superscript k in this discussion since we are referring to a single iteration of the overall procedure).

Define the function $f_{y,s}(\nu)$ to be the restriction of the function f to the line through y along s, i.e.,

$$f_{y,s}(\nu) = f(y + \nu s).$$

Thus $f_{y,s}(\nu)$ is continuous, convex, and piecewise quadratic. Define β to be the vector of positive values of ν where $f_{y,s}(\nu)$ is nondifferentiable, i.e.,

$$\beta_i = \begin{cases} -y_i/s_i & \text{if sign}(y_i) = -\text{sign}(s_i), \\ \infty & \text{otherwise.} \end{cases}$$

On the interval between any two adjacent breakpoints, say β_i and β_j, $f_{y,s}(\nu)$ is a quadratic. (The breakpoints β_i and β_j are adjacent, with $\beta_i \le \beta_j$, if there does not exist an index k such that $\beta_i < \beta_k < \beta_j$.) Label this quadratic $f_{(i,j)}(\nu)$. Hence the minimum of $f_{y,s}(\nu)$ occurs either at a breakpoint or at the minimum of one of the quadratic segments $f_{(i,j)}(\nu)$. Furthermore, $f'_{(i,j)}(\nu) = s^T g(\nu, y, s)$ and $f''_{(i,j)}(\nu) = s^T A^{-1} s$. Thus on each interval, the function $f'_{y,s}(\nu)$ is a line with slope $s^T A^{-1} s$, i.e., the curvature of f is the same for all intervals. For any ν, let β_i and β_j be the two adjacent breakpoints surrounding ν (i.e., β_i is the largest breakpoint equal to or to the left of ν, β_j is the smallest breakpoint strictly to the right of ν), and define $\gamma(\nu)$ to be the step from ν to the minimum of $f_{(i,j)}(\nu)$, i.e.,

$$\gamma(\nu) = \frac{-s^T g(\nu, y, s)}{s^T A^{-1} s}.$$

For notational convenience, define $\beta_0 = 0$ and $\gamma_i = \gamma(\beta_i)$. Figures 1 and 2 illustrate these quantities where we assume $\beta_0 < \beta_1 < \beta_2$.

In the next lemma, we show that γ is monotonically decreasing. This implies that as we move along the direction s during the line search, the distance to the optimal point

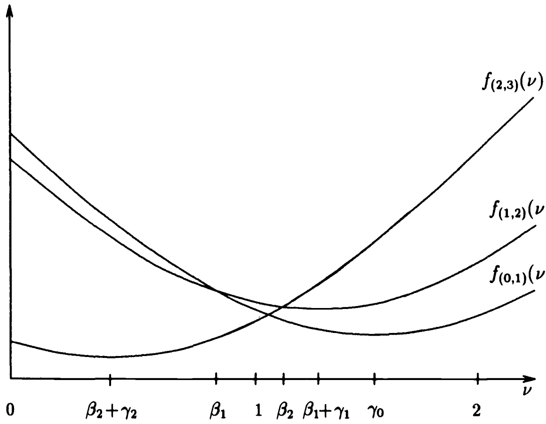


FIG. 1. *The quadratic functions that comprise the piecewise quadratic* $f_{y,s}(\nu)$.

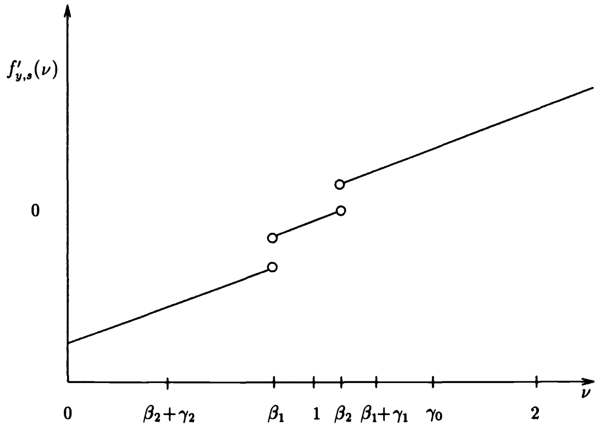


FIG. 2. *The function* $f'_{y,s}(\nu)$.

of the current quadratic is less than the distance to the optimal points of the previously encountered quadratics.

LEMMA 2.4. *Let s be a descent direction for f at the current point. Then, the functions $-s^Tg(\nu,y,s)$ and $\gamma(\nu)$ are monotonically decreasing functions of ν.*

Proof. We have

$$-s^Tg(\nu,y,s) = -s^Tg(0) - \nu s^TA^{-1}s - s^T(\sigma(\nu) - \sigma(0)). \tag{13}$$

But

$$\sigma(\nu)_i - \sigma(0)_i = \begin{cases} 0 & \text{if } \nu < \beta_i, \\ -2\sigma(0)_i & \text{otherwise.} \end{cases}$$

So

$$s^T(\sigma(\nu)-\sigma(0)) = \sum_{i\in S(\nu)} (-2\sigma_i s_i) = 2\sum_{i\in S(\nu)} |s_i| \tag{14}$$

is greater than zero and is monotonically increasing as ν increases. Thus, since A^{-1} is positive definite, $-s^Tg(\nu)$ is a monotonically decreasing function of ν. Furthermore, since

$$\gamma(\nu) = \frac{-s^Tg(\nu)}{s^TA^{-1}s},$$

$\gamma(\nu)$ is also monotonically decreasing. □

An algorithm to determine the optimal point along the descent direction s can now be described. That is, we can determine the global minimizer of the piecewise quadratic function as follows. First compute the vector β, and sort it so that the sequence $\beta_{p(1)},\ldots,\beta_{p(n)}$ is increasing, where p is the appropriate permutation vector, i.e.,

$$\beta_{p(1)} \le \beta_{p(2)} \le \cdots \le \beta_{p(n)}. \tag{15}$$

Ties can be broken arbitrarily. Examine each successive interval $(\beta_{p(i)},\beta_{p(i+1)})$ to determine if the minimum of $f_{y,s}$ occurs within it or at the end point $\beta_{p(i+1)}$ as follows. If $\gamma_{p(i)} \le \beta_{p(i+1)} - \beta_{p(i)}$ then the minimum of $f_{y,s}$ occurs at the minimum of $f_{(p(i),p(i+1))}$, so set $\alpha = \beta_{p(i)} + \gamma_{p(i)}$. Otherwise, if $-s^Tg(\beta_{p(i+1)}) \le 0$ the minimum of $f_{y,s}$ occurs at the breakpoint $\beta_{p(i+1)}$, so set $\alpha = \beta_{p(i+1)}$.

Our line search procedure follows this description *with one important modification*: in order to avoid stopping at a point of nondifferentiability,[3] a *near-optimal* point is computed. Specifically, if the minimum of $f_{y,s}$ occurs at the breakpoint $\beta_{p(i+1)}$, instead of setting $\alpha = \beta_{p(i+1)}$, set $\alpha = \beta_{p(i)} + \tau(\beta_{p(i+1)} - \beta_{p(i)})$, where $\tau = \min(c_2, 1-\theta/c_1)$, and $0 < c_2 < 1$. This guarantees that $f(y)$ is differentiable at the new point; moreover, the distance to the optimal point along the line goes to zero with θ.

```
$y = y^0$
$\rho = 1/\|Y\nabla f(y)\|_1$ evaluated at a "typical" value of $y$
while not optimal do
    $\lambda = -(A^{-1}y + A^{-1}b)$
    $\nabla f(y) = -\lambda + \mathrm{sign}(y)$
    $\eta = \rho\|Y\nabla f(y)\|_1 + \sum_i \max_i((|\lambda_i| - 1), 0)$
    $\theta = c_1\eta/(0.99 + \eta)$
    $R = (\theta I + (1-\theta)\mathrm{diag}(|\nabla f(y)|))$
    $s = -(|Y|A^{-1} + R)^{-1}(|Y|\nabla f(y))$
    determine $\alpha$ by the line search described above
    $y = y + \alpha s$
enddo
```

FIG. 3. *The proposed algorithm.*

[3] By (10) the search direction s is not defined if any component of y is zero. Therefore, we avoid such points.

2.3. Implementation details. In this section, we describe an efficient and numerically stable way to implement our algorithm (see Fig. 3). At each iteration, our algorithm requires the computation of a step

$$s = -(|Y|A^{-1} + R)^{-1}(|Y|\nabla f(y))$$

and this computation is the dominant work. However, for reasons of numerical stability, efficiency, and space we do not want to form A^{-1}. (If A is sparse, generally A^{-1} will not be sparse.) Since, by Lemma 2.3, R is nonsingular if y is not optimal, we have the following equivalent linear system of equations:

$$\begin{aligned} (|Y|A^{-1} + R)s &= -|Y|\nabla f(y), \\ (|Y| + RA)A^{-1}s &= -|Y|\nabla f(y), \\ R^{-\frac{1}{2}}(|Y| + RA)R^{\frac{1}{2}}R^{-\frac{1}{2}}(A^{-1}s) &= -R^{-\frac{1}{2}}(|Y|\nabla f(y)), \\ (|Y| + R^{\frac{1}{2}}AR^{\frac{1}{2}})R^{-\frac{1}{2}}(A^{-1}s) &= -R^{-\frac{1}{2}}(|Y|\nabla f(y)). \end{aligned}$$

Thus if we solve the symmetric positive definite system

$$(|Y| + R^{\frac{1}{2}}AR^{\frac{1}{2}})v = -R^{-\frac{1}{2}}(|Y|\nabla f(y)), \tag{16}$$

then we can easily compute $s = AR^{1/2}v$. Furthermore, this approach is well suited to sparse problems, since the structure of the matrix in (16) is always the same as that of A, and hence one data structure can be used to store all necessary Cholesky factors. (Note that a similar type of scaling can be used to improve the conditioning of the linear systems to be solved in many other interior-point quadratic programs. See Ye [19] for the general form of these systems.)

When performing the line search, we must compute $(s^k)^T g^k(\beta_j^k)$ at each breakpoint β_j^k that we cross. From (13) and (14), we have

$$-(s^k)^T g^k(\beta_j^k) = -(s^k)^T g^k(0) - \beta_j^k (s^k)^T A^{-1}(s^k) + \sum_{i \in S^k(\beta_j^k)} (2\sigma_i^k s_i^k). \tag{17}$$

(Recall that $S^k(\alpha) = \{i \mid 0 < \beta_i^k \leq \alpha\}$.) Hence we can efficiently obtain $(s^k)^T g^k(\beta_j^k)$ from $(s^k)^T g^k(\beta_{j-1}^k)$ without computing a matrix-vector product. (Note that $(s^k)^T A^{-1} s^k = (s^k)^T R^{1/2} v^k$, where v^k is given by (16).) The work of performing the line search is therefore dominated by the sorting of the breakpoints,[4] which costs $n \cdot \log n$.

3. Global convergence. In this section, we prove that our algorithm converges to the optimal point. We begin by proving some useful bounds.

A notational note: In all subsequent discussion in this section, vector s or s^k refers to the definition given by (10) and (11), unless otherwise noted.

LEMMA 3.1. *There exists $M > 0$ such that for all k, $\|y^k\|_1 \leq M$.*

Proof. Since s is a descent direction, the line search insures that $f(y^k) > f(y^{k+1})$. Thus $\{f(y^k)\}$ is monotonically decreasing. So we have

$$f(y^0) = q_y(y^0) + \|y^0\|_1 \geq f(y^k) = q_y(y^k) + \|y^k\|_1 \geq q_y(-b) + \|y^k\|_1.$$

[4] Even this cost could be reduced, on average, by avoiding the full sort and recursively choosing the minimum breakpoint, i.e., employing a heapsort mechanism.

Thus $\|y^k\|_1 \le M$, where $M = \|y^0\|_1 + q_y(y^0) - q_y(-b)$. □

COROLLARY 3.2. *$\|\nabla f(y^k)\|_2$, $\|R^k\|_2$, and $\|\lambda^k\|_2$ are bounded above.*

We use this lemma to define the domain of the problem for our nondegeneracy assumption. Let the domain C used in the nondegeneracy assumption be induced by

$$\{y \mid \|y\|_1 \le M + \epsilon\},$$

where ϵ is an arbitrarily small positive constant. We need ϵ because the proof of superlinear convergence requires nondegeneracy on an open set.

LEMMA 3.3. *$\|s^k\|_2$ is bounded above.*

Proof. We have

$$\begin{aligned}\|s^k\|_2 &= \|(A^{-1} + |Y^k|^{-1}R^k)^{-1}\nabla f(y^k)\|_2\\ &\le \|(A^{-1} + |Y^k|^{-1}R^k)^{-1}\|_2\|\nabla f(y^k)\|_2\\ &= \frac{\|\nabla f(y^k)\|_2}{\min\limits_{\|x\|_2=1} x^T(A^{-1} + |Y^k|^{-1}R^k)x}\\ &\le \frac{\|\nabla f(y^k)\|_2}{\text{min eigenvalue of } (A^{-1})}\\ &= \|\nabla f(y^k)\|_2\|A\|_2.\end{aligned}$$

Thus $\|s^k\|_2$ is bounded above. □

Next we show that the function values of the sequence of iterates converge, and the distance between iterates converges to zero.

LEMMA 3.4. *The sequence $\{f(y^k)\}$ is bounded above and below and converges.*

Proof. Since $\{f(y^k)\}$ is monotonically decreasing,

$$f(y^0) \ge f(y^k) = q_y(y^k) + \|y^k\|_1 \ge q_y(y^k) \ge q_y(-b).$$

Thus f is bounded above and below, and hence $\{f(y^k)\}$ converges. □

LEMMA 3.5. *The sequence $\{\|\alpha^k s^k\|_2\} \to 0$.*

Proof. By definition,

$$y^{k+1} = y^k + \alpha^k s^k.$$

Recalling our notation $\sigma^k(\alpha) = \text{sign}(y^k + \alpha s^k)$ and $S^k(\alpha) = \{i \mid 0 < \beta_i^k \le \alpha\}$, we have

$$\begin{aligned}f(y^k) - f(y^{k+1}) &= f(y^k) - f(y^k + \alpha^k s^k)\\ &= -(\alpha^k s^k)^T(A^{-1}y^k + A^{-1}b) + \tfrac{1}{2}(\alpha^k s^k)^T A^{-1}(\alpha^k s^k)\\ &\quad +\|y^k\|_1 - \|y^k + \alpha^k s^k\|_1.\end{aligned}$$

But for all $i \in S^k(\alpha)$,

$$|y_i^k| - |y_i^k + \alpha^k s_i^k| = 2|y_i^k| - \alpha^k s_i^k \sigma_i^k(\alpha),$$

and for $i \notin S^k(\alpha)$,

$$|y_i^k| - |y_i^k + \alpha^k s_i^k| = -\alpha s_i^k \sigma_i^k(\alpha).$$

Therefore, recalling that $\nabla f(y) = A^{-1}y + A^{-1}b +$ sign(y),

$$f(y^k) - f(y^{k+1}) = -(\alpha^k s^k)^T \nabla f(y^k + \alpha^k s^k) + \frac{1}{2}(\alpha^k s^k)^T A^{-1}(\alpha^k s^k) + 2 \sum_{i\in S^k(\alpha)} |y_i^k|.$$

Since $\sum_{i\in S^k(\alpha)} |y_i^k|$ is nonnegative, and our choice of α^k insures that

$$-(s^k)^T \nabla f(y^k + \alpha^k s^k) \geq 0,$$

we have

$$f(y^k) - f(y^{k+1}) \geq \tfrac{1}{2}(\alpha^k s^k)^T A^{-1}(\alpha^k s^k).$$

Since A^{-1} is positive definite, and $\{f(y^k)\}$ converges, then we must have $\{\|\alpha^k s^k\|_2\} \to 0$. □

Up to this point, none of our results depend on the nondegeneracy assumption; beginning with the next lemma, we will require this assumption. Now we show that under the nondegeneracy assumption, the step s converges to zero. Using this, we can then show that in the limit, complementary slackness is satisfied.

Lemma 3.6. *Under the nondegeneracy assumption*, $\{\|s^k\|_2\} \to 0$.

Proof. Suppose that $\{\|s^k\|_2\} \not\to 0$. Then Lemma 3.5 implies that a subsequence of $\{\alpha^k\}$ converges to zero. Let $\hat{\alpha}^k = \min(\gamma_0, c_2\beta_{p(1)}^k)$ where p is the permutation vector defined in (15). Note that p depends on the iteration k. Then $\alpha^k \geq \hat{\alpha}^k > 0$ and so zero is a limit point of $\{\hat{\alpha}^k\}$. However, since $g^k(0) = \nabla f(y^k) = -(A^{-1} + |Y^k|^{-1}R^k)(s^k)$, we have

$$\gamma_0 = \frac{-(s^k)^T g^k(0)}{(s^k)^T A^{-1}(s^k)} = \frac{(s^k)^T (A^{-1} + |Y^k|^{-1}R^k)(s^k)}{(s^k)^T A^{-1}(s^k)} \geq 1.$$

Thus zero must be a limit point of $\{\beta_{p(1)}^k\}$. From the definition of s^k, we have $|y_i^k|(\nabla f(y^k)_i + (A^{-1}s^k)_i) = r_i^k s_i^k$, so for each k,

$$\beta_{p(1)}^k = \beta_i^k = \frac{-y_i^k}{s_i^k} = \frac{\sigma_i^k r_i^k}{\nabla f(y^k)_i + (A^{-1}s^k)_i}$$

for some i. Since there are only a finite number of choices of index i, there must be a subsequence of $\{\beta_{p(1)}^k\}$ with $p(1) = j$ for some fixed j. Thus zero must be a limit point of $\{\beta_j^k\}$. Then since $\|s^k\|_2$ is bounded above, a subsequence of $\{y_j^k\}$ must converge to zero. If we assume that the nondegeneracy assumption holds, then the corresponding subsequence of $\{r_j^k\}$ does not have zero as a limit point. Hence a subsequence of it is bounded away from zero. Since $\{\nabla f(y^k)_j\}$ is bounded above, the corresponding subsequence of $(A^{-1}s^k)_j$ must diverge to infinity. However, $\|s^k\|_2$ is bounded above, so this is a contradiction. Therefore we must have $\{\|s^k\|_2\} \to 0$. □

Theorem 3.7. *Under the nondegeneracy assumption, the sequence* $\{Y^k \nabla f(y^k)\} \to 0$.

Proof. We have

$$\begin{aligned} \|\,|Y^k|\nabla f(y^k)\|_2 &= \|(|Y^k|\,A^{-1} + R^k)\,s^k\|_2 \\ &\leq \|(|Y^k|\,A^{-1} + R^k)\|_2 \|s^k\|_2 \\ &\leq (\|Y^k\|_2\,\|A^{-1}\|_2 + \|R^k\|_2)\,\|s^k\|_2. \end{aligned}$$

Since $\|Y^k\|_2$ and $\|R^k\|_2$ are bounded above, if $\|s^k\|_2 \to 0$, we conclude that $\{Y^k \nabla f(y^k)\} \to 0$. □

The next major result is that the sequence $\{y^k\}$ actually converges. Before we can prove this, we need the following two lemmas.

LEMMA 3.8. *Let $\nu \in \mathbb{R}^n$ be such that for all i, $v_i = 1$ or $v_i = -1$. Then the set $Z_\nu = \{y \mid Y(A^{-1}y + A^{-1}b + \nu) = 0\}$ contains a finite number of distinct points.*

Proof. Let $y \in Z_\nu$ and let J be the set of indices of the zero components of y, i.e., $J = \{j \mid y_j = 0\}$. Then since A^{-1} is positive definite, the equation $Y(A^{-1}y + A^{-1}b + \nu) = 0$ uniquely defines the remaining components of y. Hence, Z_ν contains no more points than there are unique subsets of the first n integers, so Z_ν is a finite set. □

The next lemma is standard. See, for example, [15, Note 14.1.2, p. 478], in which Ostrowski [16] is credited.

LEMMA 3.9. *Let $\{y^k\}$ be any bounded sequence of points with the following two properties. The sequence $\{y^k\}$ has a finite number of limit points and $\|y^{k+1} - y^k\|_2 \to 0$. Then the sequence $\{y^k\}$ converges.*

Finally we can show that the sequence of iterates produced by our algorithm converges.

THEOREM 3.10. *Under the nondegeneracy assumption, the sequence $\{y^k\}$ converges.*

Proof. Since Lemma 3.1 implies that the sequence $\{y^k\}$ is bounded, it must have at least one limit point. Let $\hat{y}$ be a limit point of $\{y^k\}$. Thus there is a subsequence of $\{y^k\}$ that converges to $\hat{y}$. Since there are only a finite number of distinct vectors $\text{sign}(y^k)$, there must be an infinite subsequence of this subsequence with $\text{sign}(y^k) = \nu$ for some fixed ν. Hence the corresponding subsequence of $\{Y^k(A^{-1}y^k + A^{-1}b + \nu)\}$ converges to zero, so $\hat{Y}(A^{-1}\hat{y} + A^{-1}b + \nu) = 0$. Since there are only finitely many choices of ν and Lemma 3.8 shows that for each ν the set $Z_\nu = \{y \mid Y(A^{-1}y + A^{-1}b + \nu) = 0\}$ is finite, the sequence $\{y^k\}$ can have only finitely many limit points. Hence Lemmas 3.5 and 3.9 imply that the sequence $\{y^k\}$ converges. □

The next major result is that $\{\lambda^k\}$ converges to a feasible point. We prove this by assuming the contrary and showing that the line search forbids this. First we show that if $|\lambda_j^*| > 1$, then for large enough k, the jth breakpoint will not be crossed during the line search.

LEMMA 3.11. *If $|\lambda_j^*| < 1$, then for large enough k,*

$$\text{sign}(y_j^k) = -\text{sign}(s_j^k) = \text{sign}(\nabla f(y^k)_j).$$

If $|\lambda_j^*| > 1$, then for large enough k,

$$\text{sign}(y_j^k) = -\text{sign}(s_j^k) = \text{sign}(\nabla f(y^k)_j) = -\text{sign}(\lambda_j^k) = -\text{sign}(\lambda_j^*),$$

and during the line search, the jth breakpoint will not be crossed.

Proof. If $|\lambda_j^*| \neq 1$, then since $\nabla f(y^k)_j = -\lambda_j^k + \sigma_j^k$, we have $\{\nabla f(y^k)_j\} \not\to 0$. From the definition of s^k,

$$s_j^k = \frac{-|y_j^k|}{r_j^k}(\nabla f(y^k)_j + (A^{-1}s^k)_j).$$

So since $\{(A^{-1}s^k)_j\} \to 0$, for large enough k, $\text{sign}(s_j^k) = -\text{sign}(\nabla f(y^k)_j)$.

Now suppose $|\lambda_j^*| < 1$. Then for large enough k, $|\lambda_j^k| < 1$, and so $\text{sign}(\nabla f(y^k)_j) = \sigma_j^k$, and hence $\text{sign}(y_j^k) = -\text{sign}(s_j^k) = \text{sign}(\nabla f(y^k)_j)$.

Next suppose $|\lambda_j^*| > 1$. Then for large enough k, we must have $|\lambda_j^k| > 1$. Since $\nabla f(y^k)_j = -\lambda_j^k + \sigma_j^k$ and $|\sigma_j^k| = 1$, we see that for large enough k, $\text{sign}(\nabla f(y^k)_j) = -\text{sign}(\lambda_j^k) = -\text{sign}(\lambda_j^*)$. Hence, $-\text{sign}(s_j^k) = \text{sign}(\nabla f(y^k)_j) = -\text{sign}(\lambda_j^k) = -\text{sign}(\lambda_j^*)$. This says that after some iteration, the sign of s_j^k will remain constant. Since by definition $y^{k+1} = y^k + \alpha^k s^k$, $\{y_j^k\}$ is a monotonic sequence. Since $\{\nabla f(y^k)_j\} \not\to 0$, Theorem 3.7 implies that $\{y_j^k\} \to 0$. In order for this to occur, the sign of s_j^k must be opposite that of y_j^k, otherwise $\{y_j^k\}$ would converge to a nonzero number with the same sign as s_j^k. Thus we must have $\text{sign}(y_j^k) = -\text{sign}(s_j^k) = -\text{sign}(\lambda_j^*)$ and so the jth breakpoint cannot be crossed in the line search. □

The next two lemmas will be used to show that if $|\lambda_i^*| > 1$, and $|\lambda_i^*| > |\lambda_j^*| \neq 1$, then $\beta_i^* < \beta_j^*$. From this we conclude that if $|\lambda_i^*| > 1$ and $\beta_j^k < \beta_i^k$,then for large enough k, $|\lambda_j^k| \geq |\lambda_i^k| > 1$.

LEMMA 3.12. *If* $0 < \lambda_1 < \lambda_2$, *then*

$$\frac{\lambda_1}{1+\lambda_1} < \frac{\lambda_2}{1+\lambda_2}.$$

LEMMA 3.13. *Assume that the nondegeneracy assumption holds. If* $|\lambda_i^*| = 1$ *then any limit points of the sequence* $\{\beta_i^k\}$ *are in the set* $\{-\infty, \infty\}$. *If* $|\lambda_i^*| < 1$ *then any limit points of the sequence* $\{\beta_i^k\}$ *are in the set*

$$\left\{ 1 + \frac{\theta^* |\lambda_i^*|}{1 - |\lambda_i^*|},\ 1 - \frac{\theta^* |\lambda_i^*|}{1 + |\lambda_i^*|} \right\}.$$

If $|\lambda_i^*| > 1$ *then the limit point of the sequence* $\{\beta_i^k\}$ *is*

$$1 - \frac{\theta^* |\lambda_i^*|}{1 + |\lambda_i^*|}.$$

Proof. Suppose $|\lambda_i^*| = 1$, and so using the nondegeneracy assumption, $\{y_i^k\} \not\to 0$. Since

$$\beta_i^k = \frac{-y_i^k}{s_i^k}$$

and $\{s_i^k\} \to 0$, the sequence $\{\beta_i^k\}$ can have only $-\infty$ or $+\infty$ as limit points.

If $|\lambda_i^*| < 1$, then Lemma 3.11 shows that for large enough k, $\text{sign}(\nabla f(y^k)_i) = \sigma_i^k$. Letting $\mu_i^k = \sigma_i^k \text{sign}(\lambda_i^k)$, we can write $\nabla f(y^k)_i = \sigma_i^k(1 - \mu_i^k |\lambda_i^k|)$. We can express β_i^k as

$$\begin{aligned}
\beta_i^k &= \frac{-y_i^k}{s_i^k} = \frac{\sigma_i^k r_i^k}{\nabla f(y^k)_i + (A^{-1}s^k)_i} \\
&= \frac{\sigma_i^k(|\nabla f(y^k)_i| + \theta^k(1 - |\nabla f(y^k)_i|))}{\nabla f(y^k)_i + (A^{-1}s^k)_i} \\
&= \frac{\sigma_i^k(1 - \mu_i^k|\lambda_i^k| + \theta^k \mu_i^k |\lambda_i^k|)}{\sigma_i^k(1 - \mu_i^k|\lambda_i^k| + \sigma_i^k(A^{-1}s^k)_i)} \\
&= 1 + \frac{\theta^k \mu_i^k |\lambda_i^k| - \sigma_i^k (A^{-1}s^k)_i}{1 - \mu_i^k |\lambda_i|^k + \sigma_i^k (A^{-1}s^k)_i}.
\end{aligned}$$

Since $\{(A^{-1}s^k)_i\} \to 0$, $\{\theta^k\} \to \theta^*$, and $\{|\lambda_i^k|\} \to |\lambda_i^*| < 1$, the sequence $\{\beta_i^k\}$ can have as limit points only

$$1 + \frac{\theta^*|\lambda_i^*|}{1 - |\lambda_i^*|} \quad \text{and} \quad 1 - \frac{\theta^*|\lambda_i^*|}{1 + |\lambda_i^*|}.$$

If $|\lambda_i^*| > 1$, then Lemma 3.11 shows that for large enough k, $\text{sign}(\lambda_i^k) = -\sigma_i^k$. Thus $\mu_i^k = -1$. So the limit point of the sequence $\{\beta_i^k\}$ is

$$1 - \frac{\theta^*|\lambda_i^*|}{1 + |\lambda_i^*|}. \qquad \square$$

Now we show that if $|\lambda_j^*| > 1$, then for large enough k, the line search will cause the jth breakpoint to be crossed.

LEMMA 3.14. *Assume that the nondegeneracy assumption holds and that* $|\lambda_j^*| > 1$. *Then for large enough* k,

$$-(s^k)^T g^k(\beta_j^k) > 0. \tag{18}$$

Proof. From (17), we have

$$-(s^k)^T g^k(\beta_j^k) = -(s^k)^T g^k(0) - \beta_j^k (s^k)^T A^{-1}(s^k) + \sum_{i \in S^k(\beta_j^k)} (2\sigma_i^k s_i^k). \tag{19}$$

Using the fact that $-(s^k)^T g^k(0) = -(s^k)^T \nabla f(y^k) = (s^k)^T (A^{-1} + |Y^k|^{-1} R^k)(s^k)$, we can rewrite the right-hand side of (19) as

$$(1 - \beta_j^k)(s^k)^T A^{-1}(s^k) + (s^k)^T (|Y^k|^{-1} R^k)(s^k) + \sum_{i \in S^k(\beta_j^k)} (2\sigma_i^k s_i^k). \tag{20}$$

For large enough k, Lemma 3.13 shows that $\beta_j^k < 1$, so the first term in (20) is greater than zero. We can express the second and third terms as

$$\sum_{i \in S^k(\beta_j^k)} \left(\frac{r_j^k |s_j^k|}{|y_j^k|} - 2 \right) |s_j^k| \; + \sum_{i \notin S^k(\beta_j^k)} \frac{r_i^k}{|y_i^k|} (s_i^k)^2. \tag{21}$$

The second sum in (21) is obviously greater than zero. Thus the only thing remaining to show is that the first sum in (21) is greater than zero. We can simplify the summand in the first sum as follows:

$$\left(\frac{r_i^k |s_i^k|}{|y_i^k|} - 2 \right) |s_i^k| = (|\nabla f(y^k)_i + (A^{-1}s^k)_i| - 2)|s_i^k|. \tag{22}$$

For large enough k, Lemmas 3.12 and 3.13 show that if $i \in S^k(\beta_j^k)$ then $|\lambda_i^k| \geq |\lambda_j^k| > 1$. Hence Lemma 3.11 implies that $|\nabla f(y^k)_i| = (1 + |\lambda_i^k|) > 2$. Since $\{(A^{-1}s^k)\} \to 0$, for large enough k,

$$(|\nabla f(y^k)_i + (A^{-1}s^k)_i| - 2) > 0.$$

Therefore, for large enough k, each term in the first sum in (21) is greater than zero, and the proof is complete. $\square$

Now we can prove that λ is feasible and derive some corollaries that we will use to prove superlinear convergence and to show that the step length converges to unity.

THEOREM 3.15. *Under the nondegeneracy assumption,* $|\lambda_i^*| \leq 1$ *for all* i.

Proof. Suppose $|\lambda_i^*| > 1$ for some i. Lemma 3.11 shows that for large enough k, the ith breakpoint cannot be crossed in the line search. Lemma 3.14 shows that for large enough k,

$$-(s^k)^T g^k(\beta_i^k) > 0,$$

and so our line search would cause the ith breakpoint to be crossed. These are contradictory statements and hence for all i, $|\lambda_i^*| \leq 1$. □

COROLLARY 3.16. *Under the nondegeneracy assumption,* $\{\theta^k\} \to 0$.

COROLLARY 3.17. *Under the nondegeneracy assumption, if* $y_i^* = 0$ *then* $\beta_i^* = 1$, *and if* $y_i^* \neq 0$ *then* $\beta_i^* = \pm\infty$.

Proof. The first statement follows immediately from Lemma 3.13, the definition of θ, and Theorems 3.7 and 3.15. The second statement follows from Theorem 3.7 and Lemma 3.13. □

4. Superlinear convergence. In this section we establish that under the nondegeneracy assumption, the sequence $\{y^k\}$ produced by our algorithm converges to y^* superlinearly. Consider the following finite set $\mathcal{F}$ of functions

$$F_\nu(y) = Y(A^{-1}y + A^{-1}b + \nu),$$

where $\nu \in \mathbb{R}^n$ is defined as

$$\nu_i = \begin{cases} +1 \text{ or } -1 & \text{if } y_i^* = 0, \\ \operatorname{sign}(y_i^*) & \text{otherwise.} \end{cases}$$

Each function F_ν is twice continuously differentiable and, furthermore, $F_\nu(y^*) = 0$.

The Jacobian of $F_\nu(y)$ is $J_\nu(y) = YA^{-1} + G_\nu(y)$, where $G_\nu(y) = \operatorname{diag}(A^{-1}y + A^{-1}b + \nu)$. Note that the nondegeneracy assumption implies that $J_\nu(y)$ is nonsingular. The Newton step at y^k for finding a zero of F_ν is

$$(Y^k A^{-1} + G_\nu(y^k))s_N^k = -F_\nu(y^k).$$

Lemma 3.11 shows that for large enough k, $F_{\sigma^k} \in \mathcal{F}$, and hence our search direction s^k satisfies

$$(Y^k A^{-1} + \Sigma^k R^k)s^k = -F_{\sigma^k}(y^k),$$

where $\Sigma^k = \operatorname{diag}(\sigma^k)$. Thus s^k is very similar to a Newton step at y^k and, in fact, we will show that s^k converges to a Newton step. But first we state a more general result about superlinear convergence of a family of functions. This result follows easily from Theorem 3.4 in Dennis and Moré [6].

THEOREM 4.1. *Let* $\mathcal{F} = \{F_\nu : \mathbb{R}^n \to \mathbb{R}^n\}$ *be a finite set of functions satisfying the following assumptions:*

- *Each* F_ν *is continuously differentiable in an open convex set* C.
- *There is a* y^* *in* C *such that* $F_\nu(y^*) = 0$ *and* $\nabla F_\nu(y^*)$ *is nonsingular.*
- *There is a constant* κ *such that for all* $F_\nu \in \mathcal{F}$,

$$\|\nabla F_\nu(y) - \nabla F_\nu(y^*)\| \leq \kappa \|y - y^*\|$$

for $y \in C$.

Let $\{W^k\}$ in $L(\mathbb{R}^n)$ be a sequence of nonsingular matrices. Suppose that for some y^0 in C the sequence

$$y^{k+1} = y^k - (W^k)^{-1}F_{\nu^k}(y^k), \qquad k = 0, 1, \ldots,$$

remains in C and converges to y^*, and that $y^k \neq y^*$ for $k > 0$. Then, if

$$\{\|W^k - \nabla F_{\nu^k}(y^*)\|\} \to 0, \tag{23}$$

$\{y^k\}$ converges superlinearly to y^*.

Now we show that our set of functions and the sequence generated by our algorithm satisfy the hypotheses of Theorem 4.1. For the convex open set, we take the region

$$C = \{y \mid \|y\|_1 < M + \epsilon\},$$

where M is as in Lemma 3.1 and ϵ is an arbitrarily small positive constant. We have seen that the first two assumptions hold. The next lemma shows that the third one holds.

LEMMA 4.2. *There is a constant κ such that for all $F_\nu \in \mathcal{F}$,*

$$\|\nabla F_\nu(y) - \nabla F_\nu(y^*)\|_1 \leq \kappa\|y - y^*\|_1$$

for $y \in C$.

Proof. Set $\delta = \max(M + \epsilon, \|b\|_1)$. We have

$$\begin{aligned}\|\nabla F_\nu(y) - \nabla F_\nu(y^*)\|_1 &= \|Y(A^{-1}y + A^{-1}b + \nu) - Y^*(A^{-1}y^* + A^{-1}b + \nu)\|_1\\ &\leq \|Y - Y^*\|_1\,\|A^{-1}y\|_1 \;+\; \|Y^*A^{-1}\|_1\,\|y - y^*\|_1\\ &\quad + \;\|Y - Y^*\|_1\,\|A^{-1}b + \nu\|_1\\ &\leq (3\,\|A^{-1}\|_1\,\delta + \|\nu\|_1)\,\|y - y^*\|_1\\ &= \kappa\,\|y - y^*\|_1,\end{aligned}$$

where $\kappa = 3\,\|A^{-1}\|_1\,\delta + n$. Thus we have the desired result. □

Before we can prove that (23) holds, we must show that the step length converges to one. The next lemma shows that for any fixed $\alpha > 1$, for large enough k, a step of length α takes us beyond the minimum of $f_{y,s}$.

LEMMA 4.3. *Assume that the nondegeneracy assumption holds and that $\alpha > 1$. Then for large enough k,*

$$\gamma^k(\alpha) = \frac{-(s^k)^T g(\alpha, y^k, s^k)}{(s^k)^T A^{-1}(s^k)} < 0.$$

Proof. From (13) and (14), we have

$$\gamma^k(\alpha) = \frac{1}{(s^k)^T A^{-1}(s^k)}\left(-(s^k)^T g^k(0) - \alpha(s^k)^T A^{-1}(s^k) - \sum_{i \in S^k(\alpha)} (-2\sigma_i^k s_i^k)\right). \tag{24}$$

Using the fact that $-(s^k)^T g^k(0) = -(s^k)^T \nabla f(y^k) = (s^k)^T(A^{-1} + |Y^k|^{-1}R^k)(s^k)$, we can rewrite (24) as

$$\gamma^k(\alpha) = (1 - \alpha) + \frac{(s^k)^T(|Y^k|^{-1}R^k)(s^k)}{(s^k)^T A^{-1}(s^k)} + \sum_{i \in S^k(\alpha)} \frac{-2\sigma_i^k s_i^k}{(s^k)^T A^{-1}(s^k)}.$$

Reorganizing, we get

$$\begin{aligned}\gamma^k(\alpha) &= (1-\alpha) + \sum_{i\in S^k(\alpha)} \left(\frac{r_i^k|s_i^k|}{|y_i^k|} - 2\right)\frac{|s_i^k|}{(s^k)^TA^{-1}(s^k)} + \sum_{i\notin S^k(\alpha)} \frac{r_i^k}{|y_i^k|}\frac{(s_i^k)^2}{(s^k)^TA^{-1}(s^k)}\\ &= (1-\alpha) + \sum_{i\in S^k(\alpha)} (|\nabla f(y^k)_i + (A^{-1}s^k)_i| - 2)\frac{|s_i^k|}{(s^k)^TA^{-1}(s^k)}\\ &\quad + \sum_{i\notin S^k(\alpha)} \frac{r_i^k}{|y_i^k|}\frac{(s_i^k)^2}{(s^k)^TA^{-1}(s^k)}.\end{aligned}$$

If $i \in S^k(\alpha)$, Theorem 3.15 and our nondegeneracy assumption show that for large enough k, $|\lambda_i^k| < 1$, and hence $|\nabla f(y^k)_i| = |(\sigma_i^k - \lambda_i^k)| < 2$. Furthermore, since $\{(A^{-1}s^k)\} \to 0$, for large enough k,

$$(|\nabla f(y^k)_i + (A^{-1}s^k)_i| - 2) < 0.$$

Thus, we can bound $\gamma^k(\alpha)$ as follows:

$$\begin{aligned}\gamma^k(\alpha) &\le (1-\alpha) + \sum_{i\notin S^k(\alpha)} \frac{r_i^k}{|y_i^k|}\frac{(s_i^k)^2}{(s^k)^TA^{-1}(s^k)}\\ &\le (1-\alpha) + \sum_{i\notin S^k(\alpha)} \left(\frac{r_i^k}{|y_i^k|}\right)\left(\frac{1}{\text{min eigenvalue of } A^{-1}}\right)\left(\frac{(s_i^k)^2}{\|s^k\|_2^2}\right)\\ &\le (1-\alpha) + \|A\|_2 \sum_{i\notin S^k(\alpha)} \frac{r_i^k}{|y_i^k|}.\end{aligned}$$

For large enough k, if $i \notin S^k(\alpha)$, then $y_i^k \not\to 0$ and hence $r_i^k \to 0$. Thus $\sum_{i\notin S^k(\alpha)}(r_i^k/|y_i^k|)$ converges to zero as $k \to \infty$. So, since $\alpha > 1$, for large enough k, $\gamma^k(\alpha) < 0$. □

THEOREM 4.4. *Under the nondegeneracy assumption*, $\{\alpha^k\} \to 1$.

Proof. Corollary 3.17 implies that $\{\beta_{p(1)}^k\} \to 1$ or $\{\beta_{p(1)}^k\} \to \infty$ and Corollary 3.16 implies that $\{\theta^k\} \to 0$. By definition, $\alpha^k \ge (1 - \theta^k/c_2)\beta_{p(1)}^k$, hence $\{\alpha\}$ cannot have a limit point that is less than 1. Furthermore, the properties of the line search, combined with Lemma 4.3, show that for any ϵ there exists $k(\epsilon)$, such that for $k > k(\epsilon)$, α^k cannot be greater than $1 + \epsilon$. Thus $\{\alpha^k\} \to 1$. □

The last thing necessary to prove superlinear convergence is to show that (23) holds and we show this in the next lemma.

LEMMA 4.5. *Let*

$$W^k = \frac{1}{\alpha^k}(Y^kA^{-1} + \Sigma^kR^k).$$

Then, under the nondegeneracy assumption, $\|W^k - \nabla F_{\sigma^k}(y^*)\|_2 \to 0$.

Proof. Notice that $G_{\sigma^k}(y^k) = \text{diag}(\nabla f(y^k))$. From the definitions of W^k, $\nabla F_{\sigma^k}(y^*)$, and R^k, we have

$$\|W^k - \nabla F_{\sigma^k}(y^*)\|_2 = \left\|\frac{1}{\alpha^k}(Y^kA^{-1} + \Sigma^kR^k) - (Y^*A^{-1} + G_{\sigma^k}(y^*))\right\|_2$$

$$\leq \left\| \left(\left(\frac{1}{\alpha^k}\right) Y^k - Y^* \right) A^{-1} \right\|_2 + \left\| \frac{1}{\alpha^k} (\Sigma^k \Gamma^k G_{\sigma^k}(y^k)) - G_{\sigma^k}(y^*) \right\|_2$$
$$+ \frac{\theta^k}{\alpha^k} \| \Sigma^k \left(I - |G_{\sigma^k}(y^k)| \right) \|_2,$$

where $\Gamma^k = \mathrm{diag}(\mathrm{sign}(\nabla f(y^k)))$. Theorem 4.4 shows that $\{\alpha^k\} \to 1$, and Corollary 3.16 shows that $\{\theta^k\} \to 0$. Furthermore, since $\{y^k\} \to \{y^*\}$ and $\|\nabla f(y^k)\|_2$ is bounded above, the first and third terms on the right-hand side of the above inequality converge to zero. Hence it suffices to show that

$$\left\| \frac{1}{\alpha^k} (\Sigma^k \Gamma^k G_{\sigma^k}(y^k)) - G_{\sigma^k}(y^*) \right\|_2 \to 0 \tag{25}$$

to obtain the desired result. But (25) follows immediately from Lemmas 3.7 and 3.11. □

Thus the conditions of Theorem 4.1 hold and we have the following theorem.

THEOREM 4.6. *Under the nondegeneracy assumption, the sequence $\{y^k\}$ generated by our algorithm converges superlinearly to y^*.*

Our numerical experiments suggest that our algorithm may indeed be quadratically convergent in the nondegenerate case; however, we have not been able to establish this yet. It is easy to see what needs to be proved. Under the assumptions of Theorem 4.1, if

$$\|W^k - \nabla F_{\nu^k}(y^*)\| = O(\|y^k - y^*\|), \tag{26}$$

then $\{y^k\}$ converges quadratically to y^*. It is straightforward to show that $\|Y^k - Y^*\| = O(\|y^k - y^*\|)$, $\|\Sigma^k \Gamma^k G_{\sigma^k}(y^k) - G_{\sigma^k}(y^*)\| = O(\|y^k - y^*\|)$, and $\theta^k = O(\|y^k - y^*\|)$. Hence from the proof of Lemma 4.5 it suffices to show that $1 - \alpha^k = O(\|y^k - y^*\|)$. The rate at which $\alpha^k \to 1$ depends on the rates at which γ^k decreases and $\beta^k \to 1$.

5. Numerical results.

5.1. The test problems. We generate test problems of the form (1) in the manner suggested by Moré and Toraldo [12]. They describe how to generate problems, varying four parameters: n, the number of variables; *lcnd*, the logarithm base 10 of the condition number of A; *nb*, the number of variables at their bound at the solution x^*; and $y$mag, the magnitude of the nonzero components of y^*.

To generate a test problem whose solution has certain properties, choose A to have the desired properties, generate x^* and $y^* = \nabla q_x(x^*)$ such that either x_i^* is at a bound (i.e., $|x_i^*| = 1$) or $y_i^* = 0$, but not both, and then set $b = -Ax^* + y^*$. In particular, set

$$A = QDQ \quad \text{where } Q = I - \frac{2}{\|y\|^2} yy^T,$$

D is a diagonal matrix with

$$d_{ii} = 10^{k_i \cdot lcnd}, \quad k_i = \frac{i-1}{n-1}, \quad i = 1, \ldots, n,$$

and the components of y are randomly generated in the interval $(-1, 1)$. Thus A is a positive definite matrix with condition number 10^{lcnd}.

Given *nb*, the number of variables at bounds at the solution, generate x^* as follows. Let B be the index set identifying the components of y that are zero at the solution:

$i \in B \iff y_i^* = 0$. Let B^c be the complementary set. First choose B and B^c by generating a random number μ_i in $(0,1)$ for each $i = 1,\ldots,n$ and include i in B^c if $\mu_i < nb/n$. Then choose x^* by setting those components in B^c randomly to $+1$ or -1, and selecting the remaining components by randomly generating x_i in $(-1,1)$.

Generate y^* as follows. If $i \in B$, set $y_i^* = 0$. Otherwise randomly generate μ_i in $(-1,1)$ and ν_i in $(0,1)$ and set

$$y^* = \text{sign}(\mu_i) \times 10^{-\nu_i \cdot ymag}. \tag{27}$$

Then, by setting $b = y^* - Ax^*$, we have a problem with the desired characteristics.

5.2. Numerical results. In this section, we examine the numerical behavior of our algorithm. Our implementation is in Pro-Matlab[5] and all experiments were performed using a collection of Sun Sparcstations.

Our implemention follows the algorithm described in §2. We use a single stopping criterion based on the change in objective function value: the algorithm is terminated at y^{k+1} if

$$|f(y^{k+1}) - f(y^k)| \leq \text{ tol } \cdot (1 + |f(y^k)|). \tag{28}$$

We set tol $= 10^{-15}$ for all the experiments except for the "low-precision" results where we use tol $= 10^{-8}$.

As our starting point, we choose the origin, i.e., $x^0 = 0$. Empirically, we determine that $c_1 = 10^{-3}$ and $c_2 = 0.90$ are reasonable choices of these parameters and we use them in our tests.

To capture the behavior of the algorithm, we vary each of the problem parameters, in turn, while keeping the others fixed. For the results quoted in the first six tables, we fix $n = 100$, restrict *lcnd* to the values 0, 3, 6, 9, and 12, and assign to nb the values 10, 50, and 90. We restrict $y$mag to be 1, 3, 6, 9, or 12, where the magnitude of the nonzero components of y is about $10^{-y\text{mag}}$. Therefore, the test problems become increasingly near-degenerate as $y$mag increases.

First, in order to compare the results of [8] we consider problems run to low-accuracy; i.e., tol $= 10^{-8}$. We consider 10 problems for each set of problem parameters; therefore, Tables 1–3 represent a total of 750 test problems. We report the average, maximum, and minimum number of iterations required to achieve the convergence criterion in (28).

The iteration averages in Tables 1–3 can be compared to the results given in [8] in which problems with identical characteristics (though not identical problems) were generated to test the feasible-point algorithm proposed in [8]. The stopping criteria used in both cases are comparable, as is the cost of an iteration, since in both algorithms the cost of an iteration is dominated by the solution of linear systems with identical structures. Inspection reveals that our proposed algorithm requires fewer average iterations in 71 out of 75 cases with an average differential of about 3; therefore, we feel confident in concluding that our method is at least competitive with [8] in the low-accuracy setting. (This comes with the caveat that while the method proposed in [8] maintains feasibility, our new method is only near-feasible on termination. However, the simple strategy of setting all infeasible variables to their nearest bounds upon termination is possible: on our test set this technique did not significantly increase the function value in any case.)

[5] Our results involving sparse matrices were obtained using an experimental version of Matlab [7] in which sparse matrices can be easily manipulated.

TABLE 1
Iterations for nb = 10 (*low accuracy,* tol $= 10^{-8}$).

	ymag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	10	9	11	11.5	11	12	11.5	11	12	12	11	13	11.9	11	13
3	10.6	9	12	10.7	10	12	11.7	11	13	11.6	11	12	11.9	10	13
6	9.9	9	11	11.2	10	13	11.8	11	13	11.4	10	12	11.8	11	13
9	9.7	9	10	10.3	9	12	10.9	10	12	11.2	11	12	11.6	10	12
12	9.3	8	11	9.9	9	13	10.6	9	14	11.6	9	12	11	10	12

TABLE 2
Iterations for nb = 50 (*low accuracy,* tol $= 10^{-8}$).

	ymag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	9.6	9	11	10.7	10	11	10.8	10	12	10.8	10	12	10.8	10	12
3	9.4	8	10	10.6	10	13	10.8	10	12	11.4	11	12	11.5	10	15
6	8.8	8	9	10.1	9	11	10.9	10	12	11.1	10	12	11.5	10	16
9	8.3	7	9	9.7	9	11	11.4	10	15	11.1	10	13	10.4	10	11
12	8.1	7	9	9.4	9	10	10.2	9	11	10.2	9	11	10.1	9	12

TABLE 3
Iterations for nb = 90 (*low accuracy,* tol $= 10^{-8}$).

	ymag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	8.3	7	9	9.2	8	10	8.8	8	10	8.8	8	9	8.4	8	9
3	8.0	7	9	9.8	9	11	10.5	8	16	9.7	8	11	9.5	9	10
6	8.1	7	9	12.3	8	23	10.2	9	13	9.7	9	11	9.7	9	11
9	7.2	6	8	10.2	8	15	10.6	9	16	9.3	8	10	9.8	8	15
12	7	6	8	9.9	8	18	9.8	8	14	9.7	9	10	9.2	8	11

Due to the second-order nature of our algorithm it is usually possible to obtain significantly greater accuracy at reasonable cost. Our next experiments, reported in Tables 4–6, involve exactly the same test problems as above, except that now tol $= 10^{-15}$; again, condition (28) is our sole stopping criterion.

In most cases the step from low accuracy (tol $= 10^{-8}$) to high accuracy (tol $= 10^{-15}$) involves only a modest increase in effort. The better-conditioned problems require one to two extra iterations. As *lcnd* and *ymag* increase, the number of extra iterations increases to about four or five, typically. The maximum number of iterations required by any problem, out of 750, is 41; the worst average is 20.4. Most of the problems require 17 or fewer iterations.

In our test set the accuracy achieved in the objective function value, $q_x(\bar{x})$, where $\bar{x}$ indicates the computed solution, is always acceptable. Specifically, out of 750 test problems the following bound is achieved:

$$\max \left| \frac{q_x(\bar{x}) - \text{opt}}{\text{opt}} \right| \leq 10^{-10}, \tag{29}$$

TABLE 4
Iterations for nb = 10 (*high accuracy*, tol = 10^{-15}).

	*y*mag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	11.4	11	12	13.5	13	15	15.7	14	18	17.2	16	18	16.1	15	17
3	12.6	11	15	12.5	12	14	15.2	13	18	16	14	18	16.4	14	18
6	12.9	11	15	13.7	12	16	15.5	13	19	16.6	15	21	16.8	14	18
9	13.5	13	15	13.6	11	16	14.3	11	16	15.8	14	20	16.5	14	17
12	12.4	11	15	13.2	11	15	13.7	11	17	17.3	14	28	15.4	12	18

TABLE 5
Iterations for nb = 50 (*high accuracy*, tol = 10^{-15}).

	*y*mag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	10.8	10	12	13.7	13	15	15.9	15	17	16.3	15	17	16.1	15	17
3	11.1	10	12	13	12	15	15.6	14	17	16.5	15	18	16.7	15	19
6	11.9	10	14	12.5	11	15	15.2	14	16	16	15	18	16.5	14	22
9	12.5	10	15	13.1	12	15	15.7	14	19	17.0	14	21	15.8	14	17
12	12.3	11	13	12.1	11	14	14.3	13	17	15.6	14	17	16.2	13	20

TABLE 6
Iterations for nb = 90 (*high accuracy*, tol = 10^{-15}).

	*y*mag														
	1			3			6			9			12		
lcnd	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max
0	10.1	9	13	12.2	11	14	14.1	13	15	14	13	15	13.9	13	15
3	9.2	9	10	12.9	11	17	20.4	14	41	16.6	14	22	15.9	15	22
6	10.4	9	12	14.8	11	25	17.5	13	31	16.8	15	21	16.2	14	21
9	10.3	9	11	13.6	11	19	15.6	13	22	16.7	14	22	16	13	21
12	11.2	10	13	13	10	20	15.6	13	23	19.3	15	32	16.3	12	24

where opt is the true optimal value, opt $\neq 0$. Moreover, in the vast majority of cases we achieve

$$\max \left| \frac{q_x(\bar{x}) - \text{opt}}{\text{opt}} \right| \leq 10^{-15}, \tag{30}$$

which is essentially full accuracy in the objective function value. Of course, the accuracy achieved in x varies depending on the conditioning of the problem. The worst feasibility result, over all 750 test cases, is

$$\max\{\max\left(|\bar{x}_i| - 1, 0\right)\} = 10^{-5}.$$

In all cases, setting infeasible variables to their nearest bound upon termination changed the objective function value only mildly; our worst-case bound (29) is maintained after this correction as well as the observation that (30) holds in the vast majority of the cases.

In order to test the sensitivity of our algorithm to problem size, we consider larger test cases, involving sparse matrices, and present our results in Tables 7–11. Thanks to Cleve Moler of The Mathworks, Inc., we were able to perform our experiments using an experimental version of Matlab, in which sparse matrices are easily generated and manipulated [7].

In our sparse experiments we strive for high accuracy, i.e., tol $= 10^{-15}$, and we hold the percentage of bound constraints, *nb*, fixed at 50 %. The matrices are generated using a Matlab subroutine SPRAND supplied to us by Rob Schreiber. Given the density of the matrix (dens) as well as *lcnd* and the base-10 exponent of the condition number of the matrix, SPRAND produces a sparse symmetric positive definite matrix with the given condition number and a random sparsity pattern with number of nonzeros approximately equal to dens $\times n^2$. In our tests dens $= \frac{5}{n}$ and $lcnd = 4, 8$. Our test suite consists of 5 test problems for each setting of the problem parameters, yielding a total of 100 test problems.

TABLE 7
Sparse problems, iterations for $n = 100$.

	ymag					
	1			5		
lcnd	avg	min	max	avg	min	max
4	11.6	10	13	15.4	13	18
8	12.4	12	13	15	13	20

TABLE 8
Sparse problems, iterations for $n = 200$.

	ymag					
	1			5		
lcnd	avg	min	max	avg	min	max
4	12.4	11	14	21.4	17	31
8	12.4	12	13	21.2	14	30

TABLE 9
Sparse problems, iterations for $n = 500$.

	ymag					
	1			5		
lcnd	avg	min	max	avg	min	max
4	14.4	13	17	21.4	16	29
8	14	13	16	29.6	18	45

TABLE 10
Sparse problems, iterations for $n = 1000$.

	ymag					
	1			5		
lcnd	avg	min	max	avg	min	max
4	14.8	14	16	21.8	19	24
8	15.2	17	13	25.4	22	31

TABLE 11
Sparse problems, iterations for $n = 2000$.

	*y*mag					
	1			5		
lcnd	avg	min	max	avg	min	max
4	15.2	14	16	28.6	23	33
8	16.2	14	18	33.4	23	47

The average number of iterations grows rather mildly with n. For example, in the moderately ill conditioned setting $lcnd = 4$, $y\text{mag} = 5$, the average number of iterations goes from $15.4(n = 100)$ to $21.4(n = 500)$ to $28.6(n = 2000)$.

The accuracy achieved on this set of large sparse problems is quite good. In particular, essentially full accuracy in the objective function value is achieved in every case:

$$\max \left| \frac{q_x(\bar{x}) - \text{opt}}{\text{opt}} \right| \leq 10^{-15}, \tag{31}$$

where $\bar{x}$ is the computed solution and opt is the true optimal value; feasibility was also respectable:

$$\max \{\max (|\bar{x}_i| - 1, 0)\} = 10^{-9}.$$

6. Conclusions. This paper presents a new algorithm for solving box-constrained convex quadratic programs. The method shows promise: beyond global and superlinear convergence results, the numerical experiments indicate practical potential. Specifically, high accuracy can usually be achieved with a modest number of iterations.

The real promise of this approach is in the large-scale setting where questions of exploiting sparsity or parallelism can be centered on the Cholesky factorization alone. Work outside of the factorization/solve is bounded by $\text{nnz}(A) + n \cdot \log n$, where $\text{nnz}(A)$ is the number of nonzeros of A. This work is usually negligible compared to the factorization.The Cholesky factorization is a standard linear algebra task in both the sparse and parallel settings; therefore, we need only "plug into" a standard routine to achieve efficiency.

Further research needs to be done. For example, we believe the degeneracy assumption can be greatly relaxed without weakening the theoretical properties; the question of quadratic convergence should be resolved (probably in the affirmative); more work is needed on the handling of different bounds, including one-sided bounds.

Despite the promising results of this paper with respect to our new algorithm, the most important contribution may lie elsewhere. Specifically, the *ideas* underpinning this algorithm are new (or are used in a novel way) and their full domain of applicability is unknown. To summarize, the basic underlying ideas are: the transformation of a constrained problem to a piecewise differentiable problem,[6] the notion of a Newton process for this nondifferentiable function, the definition of a descent direction in combination with an efficient line search procedure. We expect that many more problems can be approached in this way. For example, the successful l_1 algorithm in [7] also follows these lines.

[6] The lack of penalty parameter (or, equivalently, penalty parameter equal to unity) is due to two things. First, it is not hard to see that minimization of a quadratic function subject to finite box-constraints on every variable is equivalent to minimization of an unconstrained piecewise quadratic function with an easily computed penalty parameter. Second, the homogeneous unit bounds in (1) yield a unit penalty parameter.

Acknowledgment. The authors thank Mike Todd for pointing out a flaw in the proof of convergence in a previous draft and Yuying Li for numerous helpful discussions.

REFERENCES

[1] Å. Björck, *A direct method for sparse least squares problems with lower and upper bounds*, Numer. Math., 54 (1988), pp. 19–32.

[2] T. F. Coleman and A. R. Conn, *Second-order conditions for an exact penalty function*, Math. Programming, 19 (1980), pp. 178–185.

[3] T. F. Coleman and L. A. Hulbert, *A direct active set algorithm for large sparse quadratic programs with simple bounds*, Math. Programming, 45 (1989), pp. 373–406.

[4] T. F. Coleman and Y. Li, *A globally and quadratically convergent affine scaling method for linear l_1 problems*, Math. Programming, 56, Ser. A (1992), pp. 189–222.

[5] R. S. Dembo and U. Tulowitzki, *On the minimization of quadratic functions subject to box constraints*, Tech. Rep., B 71, School of Organization and Management, Yale University, New Haven, CT, 1983.

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

[7] J. Gilbert, C. Moler, and R. Schreiber, *Sparse matrices in matlab: Design and implementation*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 333–356.

[8] C.-G. Han, P. M. Pardalos, and Y. Ye, *Computational aspects of an interior point algorithm for quadratic programming problems with box constraints*, Large-Scale Numerical Optimization, T. F. Coleman and Y. Li, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990, pp. 92–112.

[9] J. J. Júdice and F. M. Pires, *Direct methods for convex quadratic programs subject to box constraints*, Departamento de Matemática, Univ. de Coimbra, Coimbra, Portugal, 1989.

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

[11] P. Lotstedt, *Solving the minimal least squares problem subject to bounds on the variables*, BIT, 24 (1984), pp. 206–224.

[12] J. J. Moré and G. Toraldo, *Algorithms for bound constrained quadratic programming problems*, Numer. Math., 55 (1989), pp. 377–400.

[13] D. P. O'Leary, *A generalized conjugate gradient algorithm for solving a class of quadratic programming problems*, Linear Algebra Appl., 34 (1980), pp. 371–399.

[14] U. Öreborn, *A Direct Method for Sparse Nonnegative Least Squares Problems*, Ph.D. thesis, Dept. of Mathematics, Linköping Univ., Linköping, Sweden, 1986.

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

[16] A. Ostrowski, *Solution of Equations and Systems of Equations*, Academic Press, New York, 1966.

[17] E. K. Yang and J. W. Tolle, *A class of methods for solving large convex quadratic programs subject to box constraints*, Tech. Rep., Dept. of Operations Research, Univ. of North Carolina, Chapel Hill, NC, 1988.

[18] Y. Ye, *An extension of Karmarkar's algorithm and the trust region method for quadratic programming*, in *Progress in Mathematical Programming*, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 49–63.

[19] ———, *Interior point algorithms for quadratic programming*, Tech. Rep. 89-29, Dept. of Management Sciences, Univ. of Iowa, Iowa City, IA, 1989.

[20] Y. Ye and E. Tse, *An extension of Karmarkar's projective algorithm for convex quadratic programming*, Math. Programming, 44 (1989), pp. 157–179.

SIAM J. OPTIMIZATION
Vol. 3, No. 2, pp. 322–358, May 1993

A NEW METHOD FOR OPTIMAL TRUSS TOPOLOGY DESIGN*

AHARON BEN-TAL[†] AND MARTIN P. BENDSØE[‡]

Abstract. Truss topology optimization formulated in terms of displacements and bar volumes results in a large, nonconvex optimization problem. For the case of maximization of stiffness for a prescribed volume, this paper presents a new equivalent, an unconstrained and convex minimization problem in displacements only, where the function to be minimized is the sum of terms, each of which is the maximum of two convex, quadratic functions. Existence of solutions is proved, as is the convergence of a nonsmooth steepest descent-type algorithm for solving the topology optimization problem. The algorithm is computationally attractive and has been tested on a large number of examples, some of which are presented.

Key words. truss topology design, nonsmooth optimization

AMS subject classifications. 90C31, 90C50, 73K40

1. Introduction. Recent years have seen a revived interest in methods for finding optimal topologies of structures [9]. Most work in optimal design of structures is related to optimization of sizes or boundary curves even though it is recognized that optimization of a structural layout (geometry and topology) has an immense impact on the performance of a structure. Analytical methods have been established for the study of fundamental properties of gridlike continua and this field goes back to the work of Michell [12], and is described in monographs by Hemp [8] and Rozvany [17]. Applications of numerical methods to discrete models, especially truss problems, are more recent, with initial studies by Dorn, Gromory, and Greenberg [5]; Fleron [6]; and Pedersen [13]. The last couple of years have seen the development of the so-called homogenization method for generating optimal topologies of structural elements (cf. Bendsøe and Kikuchi [3] and Suzuki and Kikuchi [19]), again emphasizing the great importance of topology design for the performance of a structure.

In this paper, we will consider the problem of finding the stiffest truss which is carrying a given load and which consists of perfect, slender bars of a given total volume. The bars of the truss are a subset of bars connecting all of a number of a priori chosen nodal points, this basic set of bars being the *ground structure* (cf. Fig. 1), and the topology of the truss is generated by varying the cross-sectional areas of the truss, allowing for zero cross-sectional areas. The truss is subject to an external nodal force vector f and the deformation of the truss is described by the vector x of nodal displacements. Figure 2 shows a simple three-bar truss with four nodes, of which three are fixed in all directions. The deformation is thus described by the displacement at the node Z and this displacement is controlled via the equation of equilibrium at this node.

Let a_i, ℓ_i denote the cross-sectional area and length of bar number i, respectively, and assume that all bars are made of the same linear elastic material with Young's modulus E. In order to define equilibrium and to compute bar elongations, construct the *compatibility matrix* $\boldsymbol{B}$, which is a projection matrix that relates nodal forces f and bar forces q by

$$\boldsymbol{B}^T q = f,$$

* Received by the editors April 17, 1991; accepted for publication (in revised form) January 30, 1992.

[†] Faculty of Industrial Engineering and Management, Technion–Israel Institute of Technology, Haifa 32000, Israel. This work was supported in part by the German–Israeli Foundation for Scientific Research and Development.

[‡] Mathematical Institute, Technical University of Denmark, DK-2800 Lyngby, Denmark. This work was supported in part by the Danish Technical Research Council (Programme of Research on Computer-Aided Design).

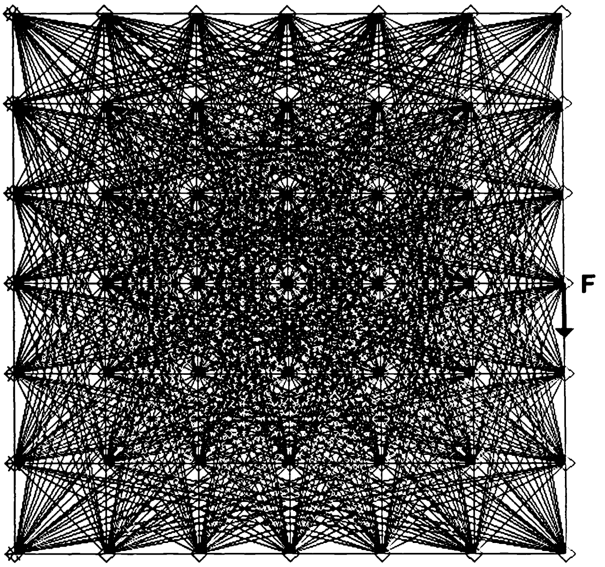


FIG. 1(a). *A ground structure with all possible node connections.*

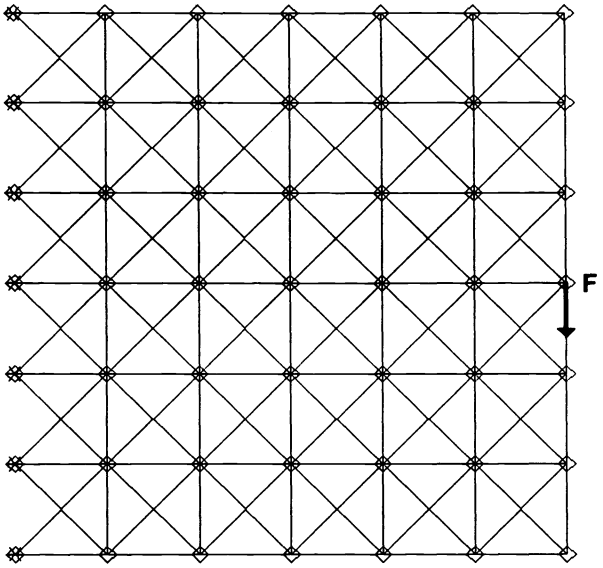


FIG. 1(b). *A ground structure with only neighboring nodes connected.*

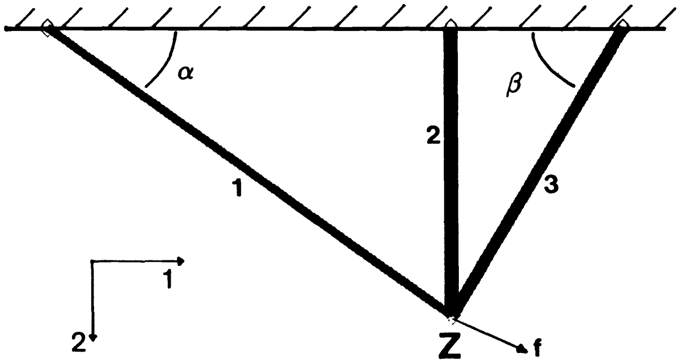


FIG. 2. *A three-bar truss.*

and which relates nodal displacements x and bar elongations Δ by

$$\boldsymbol{B}x = \Delta.$$

For the truss in Fig. 2,

$$\boldsymbol{B} = \begin{pmatrix} \cos\alpha & \sin\alpha \\ 0 & 1 \\ -\cos\beta & \sin\beta \end{pmatrix},$$

as we have three bars and two degrees of freedom. Generally $\boldsymbol{B}$ is an $m \times n_b$ matrix, m being the number of bars and n_b the degrees of freedom; $n_b = (n = \text{no. of nodes}) \times (\text{dim} = \text{dimension of space} = 2 \text{ or } 3) \div (b = \text{no. of support conditions})$.

With a member elongation Δ_i the member force q_i is

$$q_i = \frac{Ea_i}{\ell_i} \cdot \Delta_i, \tag{1.1}$$

so with $\boldsymbol{D} = \text{diag}\,(Ea_i/\ell_i)$, equilibrium is expressed as

$$f = \boldsymbol{B}^T q = \boldsymbol{B}^T \boldsymbol{D}\Delta = \boldsymbol{B}^T \boldsymbol{D}\boldsymbol{B}x = \boldsymbol{K}x,$$

where $\boldsymbol{K} = \boldsymbol{B}^T\boldsymbol{D}\boldsymbol{B}$ is called the *stiffness matrix*. The volume of the truss is given as

$$\text{Vol} = \sum_{i=1}^{m} a_i\ell_i,$$

and we thus introduce the volume of each bar, $t_i = a_i\ell_i$, as a more natural variable. Now setting (with $\delta_{k\ell}$ denoting the Kronecker index)

$$(\boldsymbol{D}_i)_{k\ell} = \frac{E}{\ell_i^2}\,\delta_{ik}\,\delta_{k\ell},$$
$$\boldsymbol{K}_i = \boldsymbol{B}^T\boldsymbol{D}_i\boldsymbol{B},$$

the stiffness matrix is written as

$$\boldsymbol{K} = \sum_{i=1}^{m} t_i\boldsymbol{K}_i,$$

where $t_i \boldsymbol{K}_i$ is the element stiffness matrix for element i. For the structure in Fig. 2, the matrices $\boldsymbol{K}_i$ are

$$\boldsymbol{K}_1 = \frac{E}{\ell_1^2}\begin{pmatrix} \cos^2\alpha & \cos\alpha\sin\alpha \\ \cos\alpha\sin\alpha & \sin^2\alpha \end{pmatrix}, \qquad \boldsymbol{K}_2 = \frac{E}{\ell_2^2}\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix},$$

$$\boldsymbol{K}_3 = \frac{E}{\ell_3^2}\begin{pmatrix} \cos^2\beta & -\cos\beta\sin\beta \\ -\cos\beta\sin\beta & \sin^2\beta \end{pmatrix}.$$

Clearly the matrices $\boldsymbol{K}_i$ are all positive semidefinite. Moreover, it is standard to assume that $\boldsymbol{B}$ has full rank (this depends on the geometry only), so as to exclude rigid body motion or mechanisms. This assumption implies that $\boldsymbol{K}$ is positive definite if all t_i satisfy $t_i > 0$.

The number $f^T x$, called the *compliance* of the structure, is a measure of the work done by the external forces and is thus inversely related to the stiffness of the truss. Finding the stiffest truss for a given total material volume v is thus covered by the formulation

$$(P1) \qquad \min_{x,t} \tfrac{1}{2} f^T x$$

subject to

$$\sum_{i=1}^{m} t_i \boldsymbol{A}_i x = f,$$
$$\sum_{i=1}^{m} t_i = v,$$
$$0 \le L_i \le t_i \le U_i \le \infty,$$

where the design variables t *and* the deformation variables x appear as independent variables, and where $\boldsymbol{A}_i$ are positive semidefinite matrices satisfying the assumption that $\sum t_i \boldsymbol{A}_i$ is positive definite if $t_i > 0$ for all $i = 1, \ldots, m$.

If the truss is supposed to carry a set of different loads, $f^1, \ldots, f^k$, a so-called *multi-load problem* can be formulated for the minimization of a weighted average of the compliances for these loads:

$$(Pm) \qquad \min \sum_{p=1}^{k} \tfrac{1}{2} W^p f^p x^p$$

subject to

$$\sum_{i=1}^{m} t_i \boldsymbol{K}_i x^p = f^p, \qquad p = 1, \ldots, k,$$
$$\sum_{i=1}^{m} t_i = v,$$
$$0 \le L_i \le t_i \le U_i \le \infty,$$

where W^p, $p = 1, \ldots, k$, denote suitable weights on the individual compliance values, and x^p are the displacements corresponding to load case f^p. This problem is of a form similar to problem $(P1)$; by introducing an extended displacement vector $x = (x^1, \ldots, x^k)$, an extended, weighted force vector $f = ((W^1)^{1/2}F^1, \ldots, (W^k)^{1/2}f^k)$, and extended unit element stiffness matrices $\boldsymbol{A}_i$ as the block-diagonal matrices with k copies of $\boldsymbol{K}_i$ in the diagonal, problem (Pm) takes the form of problem $(P1)$. In typical applications, the number of loads k is not great, in the order of 2 to 10.

In this paper our main interest is topology design, so we will typically allow for zero cross-sectional areas, i.e., $\ell_i = 0$ for all i. Also, we are primarily seeking to solve problems with a large number of nodal points (e.g., 100) and truss bars, typically taking all connecting bars in the ground structure. With n nodes, we can have up to $m = \frac{1}{2}n(n-1)$ connecting bars, with the total number of variables being $(n_b + m)$ (or $k \cdot n_b + m$ for multiload problems). Thus, for example, a successive quadratic programming (SQP) method typically will not be a suitable method for solving problem $(P1)$ and one should seek to exploit the special structure of the problem, as done in this paper.

The standard approach in structural optimization for a solution procedure for $(P1)$ (see Haftka, Kamat, and Gürdal [7]; and Rozvany and Zhou [18]) is to assume that $L_i > 0$, for all i, so that the state variable x can be eliminated by solving $\boldsymbol{A}x = f$. The derivatives of $f^T x$ are obtained through an adjoint equation, as in optimal control, or through implicit differentiation of the equilibrium equation, and we have

$$\frac{\partial}{\partial t_i}(f^T x) = -x^T \boldsymbol{A}_i x.$$

The problem is then a problem in the design variables t only, but with topology design in mind this is only a very modest reduction in problem size. For many other structural design problems, the number of state variables is much larger than the number of design variables. This is the case in boundary shape optimization with a finite element state model and a boundary defined through a rather small set of spline control points. For such problems, the matrix $\boldsymbol{A}$ is also typically sparse and banded. Again, for topology design, the situation is different because $\boldsymbol{A}$ will typically be neither banded nor sparse, as all nodes are connected.

Note that the topology optimization problem could also be formulated as a discrete optimization problem, but this has mostly been attempted in connection with material selection and cross-section-type selection problems (cf. Kirsch [9]). In addition, the homogenization method developed for topology design of continuum structures has turned out to be capable of generating truss-like thin structures; cf. Suzuki and Kikuchi [19]. This latter method automatically generates the nodal points of the truss and has a discretized formulation analogous to problem $(P1)$, but with $\boldsymbol{A}$ and volume depending nonlinearly on the design parameters. Finally, a natural extension of problem $(P1)$ is to consider the geometric location of the nodal points as design variables as well. These variables would enter the problem through the stiffness matrix $\boldsymbol{A}$ or, rather, through the compatibility matrix $\boldsymbol{B}$. Such a combination has attracted a great deal of attention (see Kirsh [9], Topping [20], and Vanderplaats [21]), but the resulting problem is extremely difficult to solve. With efficient methods for solving high-dimensional problems of type $(P1)$ in its present form, it may be more attractive to introduce a high number of nodal points in the ground structure, and in this way allow for the prediction of the optimal geometric location of nodes.

2. Summary of results. In this paper, we show that the nonconvex optimization problem $(P1)$ can be formulated in terms of an equivalent convex problem in the variables x only, thus achieving a considerable reduction in problem size. The new problem is an unconstrained problem and consists of the minimization of a nondifferentiable function $F(x)$, where $F(x)$ itself is the sum of terms, each of which is the maximum of two convex quadratic functions. For the special case of a problem $(P1)$ with only a zero lower bound on the t_i's (denoted $(P1)_s$), the new formulation is

$$(P2)_s \qquad \min_{x \in R^n} \max_{i=1,\ldots,m} \left\{ \tfrac{v}{2}\, x^T \boldsymbol{A}_i x - f^T x \right\},$$

where each term $x^T \boldsymbol{A}_i x$ is the *energy* of the bar number i. Note that the optimality conditions for problem $(P1)_s$ are

$$(2.1) \qquad \begin{aligned} &x^T \boldsymbol{A}_i x = \wedge \quad if \;\; t_i > 0, \\ &x^T \boldsymbol{A}_i x \le \wedge \quad \text{if} \;\; t_i = 0, \\ &\sum t_i \boldsymbol{A}_i x = f, \\ &\sum t_i = v, \qquad t_i \ge 0, \end{aligned}$$

where $\wedge$ is the constant (positive) Lagrange multiplier for the volume constraint. We thus see that for the optimal truss topology, no more than $n+1$ active bars (i.e., bars with $t_i > 0$) are needed. (This follows from the optimality conditions (2.1) and Caratheodory Theorem; see, e.g., [14].) Moreover, the active bars all have the same specific energy $x^T \boldsymbol{A}_i x$, and that energy level $\wedge$ is the maximum of the energies in all of the bars. This is reflected in problem $(P2)_s$, as is the fact that the conditions

$$(2.2) \qquad \sum \frac{t_i}{v} \boldsymbol{A}_i x = f/v, \qquad \sum \frac{t_i}{v} = 1$$

imply that a convex combination of the gradients of the energies of active bars equals the load f/v; equation (2.2) thus expresses the fact that the subgradient of the objective function in problem $(P2)_s$ contains zero. As problem $(P1)_s$ is not convex, this equivalence of necessary conditions does not in itself imply equivalence of $(P1)_s$ and $(P2)_s$, but this stronger result is proven in §3, where existence of solutions is also proved. In §4 we present a nonsmooth "steepest descent" algorithm for problem $(P2)$, *which simultaneously solves the original truss topology problem* $(P1)$. Section 5 contains the proof of the convergence of this algorithm. In §6 the algorithm is specialized to problems $(P2)_s$. For this special case, the algorithm is very similar to minmax algorithms, as in Demyanov and Malozemov [4] and Pshenichny and Danilin [14].

Each step of the algorithm consists of a computation of a subset J of bars which for the current estimate of x have a certain fixed energy level. The descent direction can then be computed from a quadratic programming problem with n_b variables and with the number of constraints controlled by J. This QP problem is thus of the same size as the equilibrium equation $\boldsymbol{A}x = f$, but the data of the problem only involves the bars of the set J, which typically contains many fewer than the total number of bars. Alternatively, the dual to this QP can be solved. This dual is also a QP problem, being a least-squares problem in the design variables t_i, $i \in J$, that will generate equilibrium in a least-squares sense for the current estimate of deformation x. It is advantageous to solve the dual problem, as the cardinality of J is usually considerably smaller than n_b. With the descent direction in hand, the steplength of the descent can be computed by an inexact linesearch of the Armijo–Goldstein type. For problem $(P2)_s$ we, in fact, derive an *analytical formula* for the stepsize. Alternatively an exact linesearch (e.g., golden section) can be performed, taking advantage of the fact that in most cases only "almost active" bars will influence the search. In §7 we present a number of computational examples and discuss implementation.

The algorithm is computationally very attractive because the values $x^T \boldsymbol{A}_i x$ require only a few additions and multiplications, and because we avoid assembly of the entire stiffness matrix $\boldsymbol{A} = \sum t_i \boldsymbol{A}_i$ at any stage. The algorithm thus never requires a solution of $\boldsymbol{A}x = f$, and equilibrium is actually first achieved when the algorithm has converged.

For the case of a *single load*, the matrices $\boldsymbol{A}_i$ of problem $(P1)$ are the element stiffness matrices $\boldsymbol{K}_i$, which can be written as

$$\boldsymbol{K}_i = \frac{E}{\ell_i^2} b_i b_i^T, \tag{2.3}$$

where b_i^T is the ith row of the compatibility matrix $\boldsymbol{B}$. In this case, it can be shown (see [1]) that $(P2)_s$ is equivalent to a *linear programming* problem:

$$(LPx) \qquad \begin{aligned} &\min_x -fx \\ &\text{subject to} \\ &1 \le \frac{\sqrt{E}}{\ell_i} b_i^T x \le 1, \quad i = 1, \dots, m, \end{aligned}$$

and this equivalence follows from the nontrivial equivalence between problems $(P1)_s$ and $(P2)_s$. Problem (LPx) has a rather low number of variables, but a very high number of constraints. It should be noted that for multiple load cases and/or upper (and/or lower) bounds on the bar volumes, a similar equivalence to linear programs does *not* hold.

Traditionally, truss topology optimization problems have been formulated in terms of member forces (cf. (1.1)) as a linear programming problem:

$$(LPq) \qquad \begin{aligned} &\min_{q,t} \sum_{i=1}^m t_i \\ &\text{subject to} \\ &\boldsymbol{B}^T q = f, \\ &-t_i \sigma \le \ell_i q_i \le t_i \sigma, \qquad i = 1, \dots, m, \\ &t_i \ge 0 \end{aligned}$$

for minimizing the weight, subject to equilibrium and stress constraints, σ being the limit stress value (see Dorn, Gromory, and Greenberg [5]; Fleron [6]; Kirsch [9], [10]; Pedersen [13]; Ringertz [15]; Topping [20]; and Vanderplaats [21]). *Problem* (LPq) *is the dual of problem* (LPx), written in terms of the variables $q_i = q_i^+ - q_i^-$, $t_i = (q_i^+ + q_i^-)\ell_i/\sigma$, where q_i^+, q_i^- are the dual variables of (LPx).

The equivalence mentioned above (and in other studies [1]) shows that for any solution (t, q) to (LPq) there exists a displacement field x so that (t, x) is a minimum compliance design, i.e., a solution to problem $(P1)$, as it is readily seen that for the member forces $\hat{q}$ corresponding to x, $(t, \hat{q})$ is a solution to problem (LPq). From a design point of view, the variables of primary interest are the bar volumes t_i, so (LPq) is a suitable formulation for plastic as well as elastic design.

3. A displacement-based formulation for truss topology design. The mixed formulation (simultaneous analysis and design) of the truss topology design problem is the following.[1]

PROBLEM $(P1)$.

$$\min_{x\in\Re^n, t\in\Re^m} \tfrac{1}{2}fx$$

subject to

$$\sum_{i=1}^{m} t_i A_i x = f, \tag{3.1}$$

$$\sum_{i=1}^{m} t_i = v, \tag{3.2}$$

$$L_i \le t_i \le U_i, \qquad i = 1, 2, \ldots, m. \tag{3.3}$$

The assumptions on the problem data are

(A1) $0 \le L_i < U_i \le v\,,\;\; i = 1, 2, \ldots, m;$

(A2) $\sum_{i=1}^m L_i < v < \sum_{i=1}^m U_i;$

(A3) for every i, the matrix A_i is $n \times n$ symmetric positive semidefinite;

(A4) if $t_i > 0,\;\; i = 1, \ldots, m$, then the matrix $\sum_{i=1}^m t_i A_i$ is positive definite;

(A5) $f \in \Re^n,\; f \ne 0\,.$

Problem $(P1)$ has a large number of variables $(m + n)$, and is nonconvex in the variables (x, t) due to the constraint (3.1). The main result of this section (Theorem 4) shows that Problem $(P1)$ can be solved by considering an equivalent *convex* programming problem (Problem $(P2)$ below), which has only $n + 1$ variables. Since typically m is much larger than n, Problem $(P2)$ offers an attractive way to solve the truss topology design problem. The formulation of Problem $(P2)$ is as follows.

PROBLEM $(P2)$.

$$\min_{x\in\Re^n, \lambda\in R} \left\{ F(x,\lambda) := \lambda v - fx + \sum_{i=1}^{m} \max\left\{ (\tfrac{1}{2}xA_ix - \lambda)U_i, (\tfrac{1}{2}xA_ix - \lambda)L_i \right\} \right\}.$$

The objective function $F(x, \lambda)$ is a nonsmooth convex function; in fact, it is a piecewise quadratic function, thus of "mild" nonsmoothness. The relation between Problems $(P1)$ and $(P2)$ is given in the following theorem and in Theorem 4 below.

THEOREM 1.

$$\min(P1) = -\min(P2).$$

Proof. Problem $(P1)$ can be written as

$$\min(P1) = \min_t \left\{ g(t) : \sum t_i = v,\; L \le t \le U \right\}, \tag{3.4}$$

where

$$g(t) := \min_x \left\{ \tfrac{1}{2}fx : \sum t_i A_i x = f \right\}. \tag{3.5}$$

[1] To simplify notation, we omit in the sequel the transpose symbol in inner products, matrix multiplications, etc.

We first derive an equivalent expression for $g(t)$. Let $\bar{x} = \bar{x}(t)$ be a solution of (3.5), so that

$$g(t) = \tfrac{1}{2} f\bar{x}(t).$$

Consider the convex problem

$$h(t) := \max_{x}\{fx - \tfrac{1}{2}x\boldsymbol{A}(t)x\}, \qquad \boldsymbol{A}(t) := \sum t_i \boldsymbol{A}_i. \tag{3.6}$$

The set of optimal solutions of (3.6), $X(t)$, is

$$X(t) = \{x : \boldsymbol{A}(t)x = f\},$$

and since $\bar{x}(t) \in X(t)$,

$$X(t) = \bar{x}(t) + N(\boldsymbol{A}(t))$$

where N denotes "null space." Now $h(t)$ can be computed as

$$\begin{aligned} h(t) &= \max_{x\in N(\boldsymbol{A}(t))}\{f(x+\bar{x}) - \tfrac{1}{2}(x+\bar{x})\boldsymbol{A}(t)(x+\bar{x})\} \\ &= f\bar{x} - \tfrac{1}{2}\bar{x}\boldsymbol{A}(t)\bar{x} + \max_{x\in N(\boldsymbol{A}(t))}\{x(f - \boldsymbol{A}(t)\bar{x}) - \tfrac{1}{2}x\boldsymbol{A}(t)x\} \\ &= \tfrac{1}{2}f\bar{x} \quad \text{since } \boldsymbol{A}(t)\bar{x} = f, \quad \boldsymbol{A}(t)x = 0 \; (x \in N(\boldsymbol{A}(t))). \end{aligned}$$

Thus

$$g(t) = h(t) = \max_{x}\{fx - z\tfrac{1}{2}x\boldsymbol{A}(t)x\};$$

and substituting this into (3.4),

$$\min(P1) = \min_{\substack{\sum t_i = v \\ L\le t\le U}} \max_{x\in\Re^n} \left\{ fx - \tfrac{1}{2}\sum_{i=1}^{m} t_i(x\boldsymbol{A}_i x) \right\}.$$

This is a minmax problem, which is convex (in fact, linear) in t and concave (quadratic) in x. Moreover, the constraint set of t is compact; hence a minmax theorem (Rockafellar [16, Cor. 37.3.2]) implies

$$\min(P1) = \max_{x\in\Re^n} \min_{L\le t\le U} \left\{ fx - \tfrac{1}{2}\sum_{i=1}^{m} t_i(x\boldsymbol{A}_i x) : \sum_{i=1}^{m} t_i = v \right\}. \tag{3.7}$$

By Lagrange duality, the inner minimization is equal to

$$\begin{aligned} &fx + \max_{\lambda\in R}\min_{L\le t\le U}\left\{ -\tfrac{1}{2}\sum_{i=1}^{m} t_i(x\boldsymbol{A}_i x) + \lambda\left(\sum t_i - v\right) \right\} \\ &= fx + \max_{\lambda\in R}\left\{ -\sum_{i=1}^{m} \max_{L_i\le t_i\le U_i}\{t_i(\tfrac{1}{2}x\boldsymbol{A}_i x - \lambda)\} - \lambda v \right\} \\ &= fx + \max_{\lambda}\left\{ -\lambda v - \sum_{i=1}^{m} \max\{(\tfrac{1}{2}x\boldsymbol{A}_i x - \lambda)U_i,\; \tfrac{1}{2}(x\boldsymbol{A}_i x - \lambda)L_i\} \right\}. \end{aligned}$$

Substituting the latter into (3.7),

$$\begin{aligned}\min(P1) &= \max_{x,\lambda}\left\{ fx - \lambda v - \sum_{i=1}^{m} \max\left\{ \left(\tfrac{1}{2}x\boldsymbol{A}_i x - \lambda\right) U_i,\ \left(\tfrac{1}{2}x\boldsymbol{A}_i x - \lambda\right) L_i \right\} \right\} \\ &= \max_{x,\lambda}\{-F(x,\lambda)\} = -\min(P2). \qquad \square\end{aligned}$$

The next result shows that for Problem $(P2)$ an optimal solution always exists.

THEOREM 2. *There exist $\bar{x} \in \Re^n$, and $\bar{\lambda} \in R$ such that*

$$F(\bar{x},\bar{\lambda}) = \min F(x,\lambda). \tag{3.8}$$

Proof. Let $t^0 \in \Re^m$ be a vector such that

$$t^0 > 0, \quad L \le t^0 \le U, \quad \sum_{i=1}^{m} t_i^0 = v,$$

and let $x^0 \in \Re^n$ be the unique solution of

$$\sum t_i^0 \boldsymbol{A}_i x = f, \tag{3.9}$$

i.e.,

$$x^0 = \left(\sum t_i^0 \boldsymbol{A}_i\right)^{-1} f.$$

(Recall that by assumption (A4), $\sum t_i^0 \boldsymbol{A}_i$ is positive definite and hence nonsingular.)

Let $\lambda^0 \in R$ be fixed but arbitrary. Consider the set

$$S_0 = \{(x,\lambda) \in \Re^n \times R \ :\ F(x,\lambda) \le F(x^0,\lambda^0)\}. \tag{3.10}$$

Then

$$\min_{x\in\Re^n,\lambda\in R} F(x,\lambda) = \min_{(x,\lambda)\in S_0} F(x,\lambda).$$

The function $F(x,\lambda)$ is continuous, and so to prove the existence of a solution $(\bar{x},\bar{\lambda})$, it remains to show that S_0 is bounded. Now

$$\begin{aligned} F(x^0,\lambda^0) &= \lambda^0 v - fx^0 + \sum_{i=1}^{m} \max\{(\tfrac{1}{2}x^0\boldsymbol{A}_i x^0 - \lambda^0)U_i,\ (\tfrac{1}{2}x_0\boldsymbol{A}_i x_0 - \lambda_0)L_i\} \\ &\ge \lambda^0 v - fx^0 + \sum t_i^0(\tfrac{1}{2}x^0\boldsymbol{A}_i x^0 - \lambda^0) \quad (\text{since } L_i \le t_i^0 \le U_i) \\ &= \lambda^0\left(v - \sum t_i^0\right) - x^0\left(f - \sum t_i^0 \boldsymbol{A} x^0\right) - \tfrac{1}{2}\sum t_i^0 x^0 \boldsymbol{A}_i x^0 \\ &= 0 - 0 - \tfrac{1}{2} f x^0. \end{aligned}$$

So,

$$a_0 := F(x^0,\lambda^0) \ge -\tfrac{1}{2} f x^0. \tag{3.11}$$

Let $(x, \lambda) \in S_0$; then

$$\begin{aligned} a_0 &\geq F(x,\lambda) \geq \lambda v - fx + \sum t_i^0(\tfrac{1}{2}xA_ix - \lambda) \\ &= -fx + \tfrac{1}{2}x\left(\sum t_i^0 A_i\right)x \\ &\geq -\|f\|\;\|x\| + \tfrac{1}{2}\tau_0\;\|x\|^2, \end{aligned}$$

where $0 < \tau_0$ is the minimum eigenvalue of the (positive definite) matrix $\sum t_i^0 A_i$. The above showed that if $(x, \lambda) \in S_0$ then

$$\tfrac{1}{2}\tau_0\|x\|^2 - \|f\|\;\|x\| - a_0 \leq 0. \tag{3.12}$$

Consider the polynomial

$$p(\alpha) = \tfrac{1}{2}\tau_0 \cdot \alpha^2 - \|f\|\alpha - a_0.$$

Its discriminant Δ is

$$\Delta := \|f\|^2 + 2\tau_0 a_0 \geq \|f\|^2 - \tau_0 fx^0$$

by (3.11),

$$\geq \|f\|(\|f\| - \tau_0\|x^0\|)$$

by Cauchy–Schwartz inequality, but

$$\|f\|\;\|x^0\| \geq fx^0 = x^0\left(\sum t_i^0 A_i\right)x^0 \geq \tau_0\|x_0\|^2,$$

hence $\|f\| \geq \tau_0\|x^0\|$, and so, from the above,

$$\Delta \geq 0.$$

Therefore, $p(\alpha)$ has real roots, the larger of which, ρ, is

$$\rho = \frac{1}{\tau_0}\left(\|f\| + \Delta^{1/2}\right) > 0.$$

Now, since $p(\cdot)$ is a convex (quadratic) function ($\tau_0 > 0$), the inequality $p(\alpha) \leq 0$ implies that

$$\alpha \leq \rho < \infty.$$

This shows that (3.12) implies that

$$\|x\| \leq \rho.$$

To derive a bound for λ, whenever $(x, \lambda) \in S_0$, we use the two inequalities

$$\begin{aligned} a_0 &\geq \lambda v - fx + \sum L_i(\tfrac{1}{2}xA_ix - \lambda), \\ a_0 &\geq \lambda v - fx + \sum U_i(\tfrac{1}{2}xA_ix - \lambda). \end{aligned}$$

By assumption (A2), these inequalities imply

$$\frac{-a_0 - fx + \frac{1}{2}\sum U_ixA_ix}{\sum U_i - v} \leq \lambda \leq \frac{a_0 + fx - \frac{1}{2}\sum L_ixA_ix}{v - \sum L_i},$$

which further imply

$$-\frac{a_0 + \|f\|\rho}{\sum U_i - v} \leq \lambda \leq \frac{a_0 + \|f\|\rho}{v - \sum L_i}. \qquad \square$$

Next we derive the necessary and sufficient conditions for $(\bar{x}, \bar{\lambda})$ to be a solution of Problem $(P2)$.

THEOREM 3. *A pair* $(\bar{x}, \bar{\lambda})$, $\bar{x} \in \Re^n$, $\bar{\lambda} \in R$ *is an optimal solution of Problem* $(P2)$ *if and only if there exist multipliers* $\{\bar{t}_i : i = 1, \dots, m\}$ *such that*

$$\bar{t}_i = L_i \quad \textit{if } i \in J^- := \{j : \tfrac{1}{2}\bar{x}\boldsymbol{A}_j\bar{x} < \bar{\lambda}\}, \tag{3.13}$$

$$\bar{t}_i = U_i \quad \textit{if } i \in J^+ := \{j : \tfrac{1}{2}\bar{x}\boldsymbol{A}_j\bar{x} > \bar{\lambda}\}, \tag{3.14}$$

$$L_i \leq \bar{t}_i \leq U_i \quad \textit{if } i \in J := \{j : \tfrac{1}{2}\bar{x}\boldsymbol{A}_j\bar{x} = \bar{\lambda}\}, \tag{3.15}$$

$$\sum_{i=1}^{m} \bar{t}_i \boldsymbol{A}_i \bar{x} = f, \tag{3.16}$$

$$\sum_{i=1}^{m} \bar{t}_i = v, \tag{3.17}$$

Proof. Since $F(x, \lambda)$ is a convex function, $(\bar{x}, \bar{\lambda})$ solves $(P2)$ if and only if

$$0 \in \partial F(\bar{x}, \bar{\lambda}), \tag{3.18}$$

where ∂F is the subgradient set of F. From well-known results on the subgradient of a sum and of max-functions (see, e.g., Rockafellar [16]), condition (3.18) becomes here

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} -f \\ v \end{pmatrix} + \begin{pmatrix} \sum_{J^-} L_i \boldsymbol{A}_i \bar{x} \\ -\sum_{J^-} L_i \end{pmatrix} + \begin{pmatrix} \sum_{J^+} U_i \boldsymbol{A}_i \bar{x} \\ -\sum_{J^+} U_i \end{pmatrix}$$
$$+ \sum_{J} \operatorname{conv} \left\{ \begin{pmatrix} L_i \boldsymbol{A}_i \bar{x} \\ -L_i \end{pmatrix}, \begin{pmatrix} U_i \boldsymbol{A}_i \bar{x} \\ -U_i \end{pmatrix} \right\}.$$

The latter inclusion holds if and only if numbers $\{\tau_i : i \in J\}$ exist such that

$$\begin{aligned} &0 \leq \tau_i \leq 1, \qquad i \in J, \\ &f = \sum_{J^-} L_i \boldsymbol{A}_i \bar{x} + \sum_{J^+} U_i \boldsymbol{A}_i \bar{x} + \sum_{J} (\tau_i L_i + (1 - \tau_i) U_i) \boldsymbol{A}_i \bar{x}, \\ &v = \sum_{J^-} L_i + \sum_{J^+} U_i + \sum_{J} (\tau_i L_i + (1 - \tau_i) U_i). \end{aligned}$$

This system is equivalent to (3.13)–(3.17) with

$$\bar{t}_i = \tau_i L_i + (1 - \tau_i) U_i, \qquad i \in J. \qquad \square$$

The main result follows.

THEOREM 4. *Let $(\bar{x}, \bar{\lambda})$ be an optimal solution of problem $(P2)$, with a corresponding multiplier vector $\bar{t} \in \Re^m$ (see Theorem 3). Then $(\bar{x}, \bar{t})$ is a (global) optimal solution of Problem $(P1)$.*

Proof. Clearly, by (3.13)–(3.17), the pair $(\bar{x}, \bar{t})$ is a feasible solution of $(P1)$. Moreover, by Theorem 1,

$$\begin{aligned}\min(P1) &= -\min(P2) = -F(\bar{x}, \bar{\lambda}) \\ &= -\bar{\lambda}v + f\bar{x} - \sum_{i=1}^{m} \max\{(\tfrac{1}{2}\bar{x}\boldsymbol{A}_i\bar{x} - \bar{\lambda})U_i,\ (\tfrac{1}{2}\bar{x}\boldsymbol{A}_i\bar{x} - \bar{\lambda})L_i\} \\ &= -\bar{\lambda}v + f\bar{x} - \sum_{J^-}(\tfrac{1}{2}\bar{x}\boldsymbol{A}_i\bar{x} - \bar{\lambda})L_i - \sum_{J+}(\tfrac{1}{2}\bar{x}\boldsymbol{A}_i\bar{x} - \bar{\lambda})U_i \\ &\quad - \sum_{J} \bar{t}_i(\tfrac{1}{2}\bar{x}\boldsymbol{A}_i\bar{x} - \bar{\lambda}).\end{aligned}$$

(The last summation is equal to zero by the definition of J.)

$$\begin{aligned}&= \tfrac{1}{2}f\bar{x} - \bar{\lambda}\left(v - \sum_{J^-} L_i - \sum_{J+} U_i - \sum_{J} \bar{t}_i\right) \\ &\quad + \tfrac{1}{2}\bar{x}\left(f - \sum_{J^-} L_i\boldsymbol{A}_i\bar{x} - \sum_{J+} U_i\boldsymbol{A}_i\bar{x} - \sum \bar{t}_i\boldsymbol{A}_i\bar{x}\right) \\ &= \tfrac{1}{2}f\bar{x} - \bar{\lambda}\left(v - \sum_{i=1}^{m} \bar{t}_i\right) + \tfrac{1}{2}\bar{x}\left(f - \sum_{i=1}^{m} \bar{t}_i\boldsymbol{A}_i\bar{x}\right) \qquad \text{(by (3.13), (3.14))} \\ &= \tfrac{1}{2}f\bar{x} \qquad \text{(by (3.16), (3.17))}\,.\end{aligned}$$

So, $(\bar{x}, \bar{t})$ is feasible for $(P1)$ and attains the minimal value: $\min(P1) = \frac{1}{2}f\bar{x}$; hence it is globally optimal. □

The optimality condition for $(\bar{x}, \bar{\lambda})$ to solve $(P2)$ (Theorem 3) reveals that $\bar{\lambda}$ is a *threshold energy level*. All truss members i with energy level $\frac{1}{2}\bar{x}\boldsymbol{A}_i\bar{x}$ below $\bar{\lambda}$ have the minimal volume L_i; all those with energy level above $\bar{\lambda}$ have the maximal volume U_i; all the rest have the *same* energy level $\bar{\lambda}$. We now show how to obtain the threshold value $\lambda = \lambda(x)$ for a given displacement vector x, i.e.,

$$\lambda(x) = \arg\min_{\lambda} F(x, \lambda).$$

The derivation is based on the following lemma.

LEMMA 1. *Let $\bar{v} > 0$, $T_i \geq 0$, $\alpha_i \in R$ $(i = 1, 2, \ldots, m)$ be numbers such that*

$$\alpha_1 \leq \alpha_2 < \cdots \leq \alpha_m,$$

$$\sum_{i=1}^{m} T_i > \bar{v}.$$

Let K be the largest integer such that

$$\bar{v} \leq \sum_{i=K}^{m} T_i \qquad (K \leq m).$$

Then, the optimal solution $\bar{\lambda}$ *of*

$$\min_{\lambda \in R} \left\{ \lambda \bar{v} + \sum_{i=1}^{m} (\alpha_i - \lambda)_+ T_i \right\} \tag{3.19}$$

is $\bar{\lambda} = \alpha_K$.

Proof. Since

$$\sum_{i=K+1}^{m} T_i < \bar{v} \leq \sum_{i=K}^{m} T_i,$$

we may write

$$\bar{v} = \theta \sum_{i=K}^{m} T_i + (1 - \theta) \sum_{i=K+1}^{m} T_i \quad \text{for some } 0 < \theta \leq 1.$$

Now,

$$\begin{aligned} \min_{\lambda} \{\lambda \bar{v} + \sum_{1}^{m} (\alpha_i - \lambda)_+ T_i\} &= \min_{\lambda} \left\{ \lambda \left(\theta \sum_{K}^{m} T_i + (1 - \theta) \sum_{K+1}^{m} T_i \right) \right. \\ &\qquad \left. + \sum_{1}^{m} (\alpha_i - \lambda)_+ T_i \right\} \\ &\geq \theta \min_{\lambda} \left\{ \lambda \sum_{K}^{m} T_i + \sum_{1}^{m} (\alpha_i - \lambda)_+ T_i \right\} \\ &\quad + (1 - \theta) \min_{\lambda} \left\{ \lambda \sum_{K+1}^{m} T_i + \sum_{1}^{m} (\alpha_i - \lambda)_+ T_i \right\} \\ &\geq \theta \min_{\lambda} \left\{ \lambda \sum_{K}^{m} T_i + \sum_{K}^{m} (\alpha_i - \lambda) T_i \right\} \\ &\quad + (1 - \theta) \min_{\lambda} \left\{ \lambda \sum_{K+1}^{m} T_i + \sum_{K+1}^{m} (\alpha_i - \lambda) T_i \right\} \\ &= \theta \sum_{K}^{m} \alpha_i T_i + (1 - \theta) \sum_{K+1}^{m} \alpha_i T_i \\ &= \theta \alpha_K T_K + \sum_{K+1}^{m} \alpha_i T_i := \gamma. \end{aligned}$$

Substituting $\lambda = \bar{\lambda} = \alpha_K$ in the objective function of (3.19) we get

$$\alpha_K \left(\theta \sum_K^m T_i + (1-\theta) \sum_{K+1}^m T_i \right) + \sum_{i=1}^m (\alpha_i - \alpha_K)_+ T_i$$

$$= \theta \alpha_K T_K + \alpha_K \sum_{K+1}^m T_i + \sum_{K+1}^m (\alpha_i - \alpha_K) T_i$$

$$= \theta \alpha_K T_K + \sum_{K+1}^m \alpha_i T_i = \gamma,$$

so $\bar{\lambda} = \alpha_K$ achieves the lower bound γ. Hence, it is optimal. □

THEOREM 5. *Let $\bar{x} \in \Re^n$ be given, and let*

$$\bar{\lambda} = \arg\min_{\lambda} F(\bar{x}, \lambda).$$

Let $\{i_1, i_2, \ldots, i_m\}$ be a permutation of $\{1, 2, \ldots, m\}$ such that

$$\bar{x} \boldsymbol{A}_{i_1} \bar{x} \le \bar{x} \boldsymbol{A}_{i_2} \bar{x} \cdots \le \bar{x} \boldsymbol{A}_{i_m} \bar{x}$$

and let K be the largest integer such that

$$\sum_{j=K}^m U_{i_j} + \sum_{j=1}^{K-1} L_{i_j} \ge v \qquad (K \le m);$$

then

$$\bar{\lambda} = \tfrac{1}{2} \bar{x} \boldsymbol{A}_{i_K} \bar{x}.$$

Proof. $F(x, \lambda)$ can be written as

$$F(x, \lambda) = -fx + \tfrac{1}{2} \sum L_i x \boldsymbol{A}_i x + \lambda(v - \sum L_i) + \sum (\tfrac{1}{2} x \boldsymbol{A}_i x - \lambda)_+ (U_i - L_i).$$

Hence

$$\bar{\lambda} = \arg\min_{\lambda} \left\{ \lambda(v - \sum L_i) + \sum (\tfrac{1}{2} \bar{x} \boldsymbol{A}_i \bar{x} - \lambda)_+ (U_i - L_i) \right\}.$$

Define

$$\alpha_j := \tfrac{1}{2} \bar{x} \boldsymbol{A}_{i_j} \bar{x}, \qquad j = 1, \ldots, m,$$

$$\bar{v} := v - \sum L_i,$$

$$T_j := U_{i_j} - L_{i_j}, \qquad j = 1, \ldots, m;$$

thus the conclusion in the theorem follows immediately from Lemma 1. □

4. An algorithm for solving Problems $(P2)$ **and** $(P1)$**.** We describe an algorithm for solving the nonsmooth problem $(P2)$

$$(P2) \qquad \min_{x\in\Re^n,\lambda\in R}\left\{F(x,\lambda) := \lambda v - fx + \sum_{i=1}^{m} F_i(x,\lambda)\right\},$$

$$F_i(x,\lambda) := \max\{(\tfrac{1}{2}x\boldsymbol{A}_i x - \lambda)U_i, (\tfrac{1}{2}x\boldsymbol{A}_i x - \lambda)L_i\}.$$

The algorithm will find the optimal solution $(\bar{x},\bar{\lambda})$ and will simultaneously generate an optimal solution pair $(\bar{x},\bar{t})$ for Problem $(P1)$. The basic iteration step is

$$\begin{pmatrix} x^{\ell+1} \\ \lambda^{\ell+1} \end{pmatrix} = \begin{pmatrix} x^{\ell} \\ \lambda^{\ell} \end{pmatrix} + \alpha_\ell \begin{pmatrix} d^{\ell} \\ \delta_\ell \end{pmatrix}, \qquad \ell = 0,1,2,\ldots,$$

where (d^ℓ,δ_ℓ) is a *direction of descent* of F at (x^ℓ,λ^ℓ), and $\alpha_\ell \geq 0$ is the *stepsize*. The direction vector (d^ℓ,δ_ℓ) is generated by solving a quadratic programming problem.

At a given point (x^ℓ,λ^ℓ), the *directional derivative* of F in the direction (d,δ), denoted by $F'(x^\ell,\lambda^\ell,d,\delta)$, is given (using well-known results on the directional derivative of a max-function) by

$$(4.1) \qquad \begin{aligned} F'(x^\ell,\lambda^\ell;d,\delta) = v\delta - fd &+ \sum_{J_\ell^-} L_i(d\boldsymbol{A}_i x^\ell - \delta) + \sum_{J_\ell^+} U_i(d\boldsymbol{A}_i x^\ell - \delta) \\ &+ \sum_{J_\ell} \max\{U_i(d\boldsymbol{A}_i x^\ell - \delta), L_i(d\boldsymbol{A}_i x^\ell - \delta)\}, \end{aligned}$$

where the index sets $\bar{J}_\ell^-$, $\bar{J}_\ell^+$, and $\bar{J}_\ell$ are defined by

$$\begin{aligned} \bar{J}_\ell^- &: = \{i : \tfrac{1}{2}x^\ell\boldsymbol{A}_i x^\ell < \lambda^\ell\}, \\ \bar{J}_\ell^+ &: = \{i : \tfrac{1}{2}x^\ell\boldsymbol{A}_i x^\ell > \lambda^\ell\}, \\ \bar{J}_\ell &: = \{i : \tfrac{1}{2}x^\ell\boldsymbol{A}_i x^\ell = \lambda^\ell\}. \end{aligned}$$

A *steepest descent direction* of F at (x^ℓ,λ^ℓ) is a vector $(\bar{d}^\ell,\bar{\delta}_\ell)$, which solves the minimization problem

$$(4.2) \qquad \min_{d\in\Re^n,\delta}\{F'(x^\ell,\lambda^\ell;\, d,\delta) + \tfrac{1}{2}(\|d\|^2 + \delta^2)\}.$$

The second term in the objective function is added to bound the length of the direction vector (d,δ). Let

$$\begin{aligned} \bar{v}^\ell &: = v - \sum_{\bar{J}_\ell^-} L_i - \sum_{\bar{J}_\ell^+} U_i, \\ \bar{f}^\ell &: = f - \sum_{\bar{J}_\ell^-} L_i\boldsymbol{A}_i x^\ell - \sum_{\bar{J}_\ell^+} U_i\boldsymbol{A}_i x^\ell. \end{aligned}$$

Then, by (4.1),

$$F'(x^\ell,\lambda^\ell,d,\delta) = \bar{v}^\ell\delta - \bar{f}^\ell d + \sum_{\bar{J}_\ell} \mu_i,$$

where

$$\mu_i = \max\{U_i(d\boldsymbol{A}_i x^\ell - \delta), L_i(d\boldsymbol{A}_i x^\ell - \delta)\}.$$

Then, problem (4.2) can be written as a quadratic program in the variables $d \in \Re^n, \delta \in R, \{\mu_i : i \in J_\ell\}$:

$$(P_\ell) \qquad \min\left\{ \bar{v}^\ell \delta - \bar{f}^\ell d + \sum_{\bar{J}_\ell} \mu_i + \tfrac{1}{2}\|d\|^2 + \tfrac{1}{2}\delta^2 \right\},$$

subject to

$$\mu_i \geq U_i d\boldsymbol{A}_i x^\ell - U_i \delta,$$

$$\mu_i \geq L_i d\boldsymbol{A}_i x^\ell - L_i \delta, \qquad i \in \bar{J}_\ell.$$

One can obtain the optimal solution of (P_ℓ), $(\bar{d}^\ell \bar{\delta}_\ell)$, by solving the dual problem of (P_ℓ), which is as follows (we omit the details):

$$(D_\ell) \qquad \max_t \left\{ -\tfrac{1}{2} \left\| \sum_{\bar{J}_\ell} t_i \boldsymbol{A}_i x^\ell - \bar{f}^\ell \right\|^2 - \frac{1}{2} \left\| \sum_{\bar{J}_\ell} t_i - \bar{v}^\ell \right\|^2 \right\},$$

$$L_i \leq t_i \leq U_i, \qquad i \in \bar{J}_\ell.$$

From the primal-dual relations between (P_ℓ)–(D_ℓ), if $\bar{t}^\ell$ is the optimal solution of (D_ℓ), then the optimal solution of (P_ℓ) is

$$\bar{d}^\ell = -\left(\sum_{\bar{J}_\ell} \bar{t}_i^\ell \boldsymbol{A}_i x^\ell - \bar{f}^\ell \right),$$

$$\bar{\delta}_\ell = \left(\sum_{\bar{J}_\ell} \bar{t}_i - \bar{v}^\ell \right).$$

It is easy to verify, from the optimality conditions in Theorem 3, and the result of Theorem 4, that the following result holds.

THEOREM 6. $\bar{d}^\ell = 0, \bar{\delta}_\ell = 0$ *if and only if* (x^ℓ, λ^ℓ) *solves Problem* $(P2)$ *and* $(x^\ell, \bar{t}^\ell)$ *solves Problem* $(P1)$.

The last result is of theoretical value since an algorithm based on the iteration step

$$x^{\ell+1} = x^\ell + \alpha_\ell \bar{d}^\ell,$$

$$\lambda^{\ell+1} = \lambda^\ell + \alpha_\ell \bar{\delta}^\ell$$

does not necessarily converge. Indeed, unlike the smooth case, for which the steepest descent algorithm is convergent, this is not the case for nonsmooth problems such as $(P2)$ (see, e.g., Lemarechal [11]).

The cure is to introduce a perturbation of the "active constraint set" J_ℓ. This will prevent the solution of problem (P_ℓ) or (D_ℓ) to change discontinuously when a constraint becomes inactive. The specific way this perturbation is chosen here is described next. In what follows, $\epsilon > 0$ is a fixed parameter controlling the "activity" index sets defined below:

$$J_\ell := \{i : |\tfrac{1}{2}x^\ell A_i x^\ell - \lambda^\ell| \le \epsilon/(U_i - L_i)\},$$

$$J_\ell^+ := \{i : \tfrac{1}{2}x^\ell A_i x^\ell - \lambda^\ell > \epsilon/(U_i - L_i)\},$$

$$J_\ell^- := \{i : \tfrac{1}{2}x^\ell A_i x^\ell - \lambda^\ell < -\epsilon/(U_i - L_i)\}.$$

Also let

$$v^\ell := v - \sum_{i \in J_\ell^+} U_i - \sum_{i \in J_\ell^-} L_i,$$

$$f^\ell := f - \sum_{i \in J_\ell^+} U_i A_i x^\ell - \sum_{i \in J_\ell^-} L_i A_i x^\ell.$$

An *ϵ-steepest descent direction* (d^ℓ, δ_ℓ) for $(P2)$ at (x^ℓ, λ^ℓ) is the solution of the quadratic program $(\hat{P}_\ell)$:

$$(\hat{P}_\ell) \qquad \begin{aligned} &\min_{d,\mu,\delta} \left\{ v^\ell \delta - df^\ell + \sum_{J_\ell} \mu_i + \frac{1}{2}\|d\|^2 + \frac{1}{2}\delta^2 \right\}, \\ &\text{subject to} \\ &U_i(dA_i x^\ell - \delta + p_i^\ell) - \mu_i \le 0, \qquad i \in J_\ell, \\ &L_i(dA_i x^\ell - \delta + p_i^\ell) - \mu_i \le 0, \end{aligned}$$

where

$$p_i^\ell = \tfrac{1}{2}x^\ell A_i x^\ell - \lambda^\ell.$$

Note that problem $(\hat{P}_\ell)$ is a perturbation of problem (P_ℓ). Indeed $|p_i^\ell| \le \epsilon/(U_i - L_i)$ for $i \in J_\ell$, and $J_\ell \approx \bar{J}_\ell$ for ϵ small; the problems coincide for $\epsilon = 0$.

A dual problem of $(\hat{P}_\ell)$ is the following quadratic program:

$$(\hat{D}_\ell) \qquad \begin{aligned} &\max_t \left\{ \sum_{i \in J_\ell} t_i p_i^\ell - \tfrac{1}{2} \left\| \sum_{i \in J_\ell} t_i A_i x^\ell - f^\ell \right\|^2 - \tfrac{1}{2} \left\| \sum_{i \in J_\ell} t_i - v^\ell \right\|^2 \right\}, \\ &L_i \le t_i \le U_i, \qquad i \in J_\ell. \end{aligned}$$

If t^ℓ is the solution of $(\hat{D}_\ell)$, then the solution (d^ℓ, δ_ℓ) of $(\hat{P}_\ell)$ is given by

$$(4.3) \qquad \begin{aligned} d^\ell &= -\left(\sum_{i \in J_\ell} t_i^\ell A_i x^\ell - f^\ell \right), \\ \delta_\ell &= \left(\sum_{i \in J_\ell} t_i^\ell - v^\ell \right). \end{aligned}$$

We now demonstrate that a result similar to Theorem 6 holds for problems $(\hat{P}_\ell)$ and$(\hat{D}_\ell)$.

THEOREM 7. *$d_\ell = 0, \delta_\ell = 0$ if and only if (x^ℓ, λ^ℓ) solves Problem $(P2)$ and (x^ℓ, t^ℓ) solves Problem $(P1)$.*

Proof. The optimality conditions for (d^ℓ, δ_ℓ) to solve problem $(\hat{P}_\ell)$ are

$$\sum_{J_\ell} t_i^\ell - \delta_\ell = v^\ell, \tag{4.4}$$

$$\sum_{J_\ell} t_i^\ell \boldsymbol{A}_i x^\ell + d^\ell = f^\ell, \tag{4.5}$$

$$(t_i^\ell - L_i)\left(U_i(h_i^\ell + p_i^\ell) - \mu_i^\ell\right) = 0, \qquad i \in J_\ell, \tag{4.6}$$

$$(U_i - t_i^\ell)\left(L_i(h_i^\ell + p_i^\ell) - \mu_i^\ell\right) = 0, \qquad i \in J_\ell, \tag{4.7}$$

$$L_i \le t_i^\ell \le U_i, \qquad i \in J_\ell, \tag{4.8}$$

$$\mu_i^\ell = \max\left\{U_i(h_i^\ell + p_i^\ell),\ L_i(h_i^\ell + p_i^\ell)\right\}, \qquad i \in J_\ell, \tag{4.9}$$

where

$$p_i^\ell := \tfrac{1}{2} x^\ell \boldsymbol{A}_i x^\ell - \lambda^\ell, \qquad h_i^\ell := d^\ell \boldsymbol{A}_i x^\ell - \delta_\ell.$$

Define

$$t_i^\ell = L_i, \quad \mu_i^\ell = L_i(h_i^\ell + p_i^\ell), \quad i \in J_\ell^-,$$

$$t_i^\ell = U_i, \quad \mu_i^\ell = U_i(h_i^\ell + p_i^\ell), \quad i \in J_\ell^+.$$

Then, using the definition of v^ℓ and f^ℓ, the system (4.4)–(4.9) can be written as follows:

$$\sum_{i=1}^{m} t_i^\ell - \delta^\ell = v, \tag{4.10}$$

$$\sum_{i=1}^{m} t_i^\ell \boldsymbol{A}_i x^\ell + d^\ell = f, \tag{4.11}$$

$$(t_i^\ell - L_i)\left(U_i(h_i^\ell + p_i^\ell) - \mu_i^\ell\right) = 0, \qquad i = 1, \ldots, m, \tag{4.12}$$

$$(U_i - t_i^\ell)\left(L_i(h_i^\ell + p_i^\ell) - \mu_i^\ell\right) = 0, \qquad i = 1, \ldots, m, \tag{4.13}$$

$$\begin{array}{l} L_i \le t_i \le U_i \qquad i = 1, \ldots, m \\ \text{with} \\ t_i^\ell = L_i, \quad i \in J_\ell^-, \quad t_i^\ell = U_i, \quad i \in J_\ell^+, \end{array} \tag{4.14}$$

$$\begin{aligned}
&\mu_i^\ell = \max\{U_i(h_i^\ell + p_i^\ell),\ L_i(h_i^\ell + p_i^\ell)\}, \qquad i \in J_\ell,\\
(4.15)\qquad &\mu_i^\ell = L_i(h_i^\ell + p_i^\ell), \qquad i \in J_\ell^-,\\
&\mu_i^\ell = U_i(h_i^\ell + p_i^\ell), \qquad i \in J_\ell^+.
\end{aligned}$$

The optimality condition at $(\bar{x}, \bar{\lambda})$ for Problem $(P2)$ can be written as the system

$$(4.16)\qquad \sum_{i=1}^{m} \bar{t}_i = v,$$

$$(4.17)\qquad \sum_{i=1}^{m} \bar{t}_i \boldsymbol{A}_i \bar{x} = f,$$

$$(4.18)\qquad (\bar{t}_i - L_i)[U_i \bar{p}_i - \bar{z}_i] = 0, \qquad i = 1, \dots, m,$$

$$(4.19)\qquad (U_i - \bar{t}_i)[L_i \bar{p}_i - \bar{z}_i] = 0, \qquad i = 1, \dots, m,$$

$$(4.20)\qquad L_i \le \bar{t}_i \le U_i, \qquad i = 1, \dots, m,$$

$$(4.21)\qquad \bar{z}_i = \max\{U_i \bar{p}_i, L_i \bar{p}_i\}, \qquad i = 1, \dots, m,$$

where

$$\bar{p}_i := \tfrac{1}{2} \bar{x} \boldsymbol{A}_i \bar{x} - \bar{\lambda}.$$

Let $d^\ell = 0,\ \delta_\ell = 0$. Then $h_i^\ell = 0$. Also $i \in J_\ell^- \Rightarrow p_i^\ell < 0$ and $i \in J_\ell^+ \Rightarrow p_i^\ell > 0$ and therefore (4.15) reduces to

$$\mu_i^\ell = \max\{U_i p_i^\ell, L_i p_i^\ell\}, \qquad i = 1, \dots, m.$$

It is easily seen, by comparing the systems (4.10)–(4.15) with (4.16)–(4.21), that

$$\bar{x} = x^\ell, \quad \bar{\lambda} = \lambda^\ell, \quad \bar{z}_i = \mu_i^\ell, \quad i = 1, \dots, m$$

is an optimal solution of $(P2)$.

Conversely, let $\bar{x} = x^\ell, \bar{\lambda} = \lambda^\ell$ be a solution of $(P2)$. Then $p_i^\ell = \bar{p}_i$ and it follows from (4.21) that

$$\begin{aligned}
\bar{z}_i &= U_i p_i^\ell \quad \text{if } p_i^\ell > 0, \quad \text{in particular, if } i \in J_\ell^+;\\
\bar{z}_i &= L_i p_i^\ell \quad \text{if } p_i^\ell < 0, \quad \text{in particular, if } i \in J_\ell^-.
\end{aligned}$$

Hence, $d^\ell = 0, \delta_\ell = 0$ (which makes $h_i^\ell = 0$) with corresponding multipliers $t_i^\ell = \bar{t}_i,\ \mu_i^\ell = \bar{z}_i$ satisfy the optimality condition (4.10)–(4.15) for $(\hat{P}_\ell)$. □

Once an ϵ-steepest descent direction (d^ℓ, δ_ℓ) has been computed, the stepsize α_ℓ can be computed by

$$\alpha_\ell = \arg\min_{\alpha \ge 0} F(x^\ell + \alpha d^\ell,\ \lambda^\ell + \alpha \delta_\ell).$$

Here, we employ an *inexact linesearch* of the Armijo–Goldstein type. The stopping rule for the algorithm is based on Theorem 7.

ALGORITHM A [For solving Problems ($P2$) and ($P1$)]
Parameters: $\epsilon > 0$ ("activity" parameter), $\delta > 0$ (for the stopping rule), $0 < \theta < \frac{1}{2}$ (for the stepsize rule).
Initialization

(0.1) *Choose* an initial design vector t^0

$$t^0 > 0, \quad L \leq t^0 \leq U, \quad \sum_{i=1}^{m} t_i^0 = v.$$

(0.2) *Solve* the linear system

$$\sum_{i=1}^{m} t_i^0 \boldsymbol{A}_i x = f$$

to obtain its (unique) solution x^0.

(0.3) *Compute* λ^0 as follows (see Theorem 5). Compute a permutation $(i_1, i_2, \ldots, i_m)$ of $\{1, 2, \ldots, m\}$ such that

$$x^0 \boldsymbol{A}_{i_1} x^0 \leq x^0 \boldsymbol{A}_{i_2} x^0 \leq \cdots \leq x^0 \boldsymbol{A}_{i_m} x^0.$$

Let K be the largest integer such that

$$\sum_{j=K}^{m} U_{i_j} + \sum_{j=1}^{K-1} L_{i_j} \geq v \qquad (K \leq m).$$

Then

$$\lambda_0 = \tfrac{1}{2} x^0 \boldsymbol{A}_{i_K} x^0.$$

Step ℓ (x^ℓ, λ^ℓ given)

(ℓ.1) Generate the index sets J_ℓ, J_ℓ^+, J_ℓ^-, compute v^ℓ and f^ℓ.

(ℓ.2) Solve ($\hat{P}_\ell$) to obtain (d^ℓ, δ_ℓ) [OR: solve ($\hat{D}_\ell$) to obtain t^ℓ and then compute d^ℓ, δ_ℓ by the formula (4.3)].

(ℓ.3) If $\max(\|d^\ell\|, |\delta_\ell|) < \delta$ stop, *else* go to (ℓ.4).

(ℓ.4) Compute the stepsize α_ℓ as the largest $\alpha > 0$ such that

$$F(x^\ell + \alpha d^\ell,\ \lambda^\ell + \alpha\delta_\ell) \leq F(x^\ell,\ \lambda^\ell) - \alpha\theta(\|d^\ell\|^2 + \delta_\ell^2). \tag{4.22}$$

Note: An approximation of α_ℓ can be computed as follows. Let $K(\ell)$ = smallest integer K such that $\alpha = (\frac{1}{2})^K$ satisfies (4.22), then

$$\alpha_\ell = (\tfrac{1}{2})^{K(\ell)}.$$

(ℓ.5)

$$x^{\ell+1} = x^\ell + \alpha_\ell d^\ell,$$

$$\lambda^{\ell+1} = \lambda^\ell + \alpha_\ell \delta_\ell.$$

(ℓ.6) $\ell \leftarrow \ell + 1$, go to (ℓ.1).

5. Convergence of Algorithm A. In this section we show that convergent subsequences generated by Algorithm A produce an optimal solution of $(P2)$ and simultaneously (by Theorem 3), an optimal solution of the original truss topology design Problem $(P1)$.

THEOREM 8. *The sequence* $\{x^\ell, \lambda^\ell\}_0^\infty$, *generated by Algorithm A, has a convergent subsequence. The limit point of any such subsequence is an optimal solution of problem* $(P2)$.

Proof. In the proof of Theorem 3 it was shown that the set

$$S_0 = \{(x, \lambda) \; : \; F(x, \lambda) \leq F(x^0, \lambda^0)\}$$

is compact. Since, by (4.22),

$$F(x^{\ell+1}, \lambda^{\ell+1}) \leq F(x^\ell, \lambda^\ell) \qquad \text{for all } \ell = 0, 1, 2, \ldots,$$

it follows that

$$\{x^\ell, \lambda^\ell\}_0^\infty \subset S_0,$$

and by the compactness of S_0, this implies the existence of a convergent subsequence. For simplicity of notation, we denote this subsequence also by $\{x^\ell, \lambda^\ell\}_0^\infty$. Let $(\bar{x}, \bar{\lambda})$ be its limit point. Consider an index $i \in J_\ell$; then

$$\begin{aligned} F_i(x^\ell + \alpha d^\ell, \lambda^\ell + \alpha\delta_\ell) &= \max \left\{ \begin{array}{l} U_i(p_i^\ell + \alpha h_i^\ell + \frac{1}{2}\alpha^2 d^\ell \boldsymbol{A}_i d^\ell)\,, \\ L_i(p_i^\ell + \alpha h_i^\ell + \frac{1}{2}\alpha^2 d^\ell \boldsymbol{A}_i d^\ell) \end{array} \right\} \\ &\leq \max \left\{ \begin{array}{l} (1-\alpha) U_i p_i^\ell + \alpha \mu_i^\ell + \frac{U_i}{2}\alpha^2 d^\ell \boldsymbol{A}_i d^\ell\,, \\ (1-\alpha) L_i p_i^\ell + \alpha \mu_i^\ell + \frac{L_i}{2}\alpha^2 d^\ell \boldsymbol{A}_i d^\ell \end{array} \right\} \quad \text{by (4.9)} \\ &\leq (1-\alpha) F_i(x^\ell, \lambda^\ell) + \alpha \mu_i^\ell + \tfrac{\alpha^2}{2} M_1 \|d^\ell\|^2 \quad \text{for } 0 < \alpha \leq 1, \end{aligned}$$

where

$$M_1 = \max_{i=1,\ldots,m} \{U_i \lambda_{\max}(\boldsymbol{A}_i)\}, \qquad \lambda_{\max}(\boldsymbol{A}_i) := \text{maximal eigenvalue of } \boldsymbol{A}_i,$$

and with $p_i^\ell, \mu_i^\ell, h_i^\ell$ defined as in the proof of Theorem 7 (see (4.9)).

From the above inequality

$$\begin{aligned} A &:= \sum_{i \in J_\ell} F_i(x^\ell + \alpha d^\ell, \lambda^\ell + \alpha \delta^\ell) \\ &\leq (1-\alpha) \sum_{J_\ell} F_i(x^\ell, \lambda^\ell) + \alpha \sum_{J_\ell} \mu_i^\ell + \tfrac{1}{2}\alpha^2 M_1 \|d^\ell\|^2 m_A, \end{aligned} \tag{5.1}$$

where

$$m_A := \text{card}\,(J_\ell).$$

Note that $d = 0$, $\delta = 0$, $\mu_i = F_i(x^\ell, \lambda^\ell) (= \max\{U_i p_i^\ell, L_i p_i^\ell\})$ is a feasible solution of $(\hat{P}_\ell)$. Hence

$$v^\ell \delta^\ell - d^\ell f^\ell + \sum_{J_\ell} \mu_i^\ell + \tfrac{1}{2}\|d^\ell\|^2 + \tfrac{1}{2}\delta_\ell^2 \leq \sum_{J_\ell} F_i(x^\ell, \lambda^\ell), \tag{5.2}$$

so (5.1) and (5.2) imply

$$A \le \sum_{J_\ell} F_i(x^\ell, \lambda^\ell) - \alpha v^\ell \delta_\ell + \alpha d^\ell f^\ell + \tfrac{1}{2}\alpha(\alpha M_1 m_A - 1)\|d^\ell\|^2 - \tfrac{1}{2}\alpha\delta_\ell^2. \tag{5.3}$$

We now evaluate

$$B := \sum_{J_\ell^+} F_i(x^\ell + \alpha d^\ell, \lambda^\ell + \alpha\delta_\ell) \quad \text{and} \quad C := \sum_{J_\ell^-} F_i(x^\ell + \alpha d^\ell, \lambda^\ell + \alpha\delta_\ell).$$

First, we obtain a bound on $|h_i^\ell|$. Recall

$$h_i^\ell = d^\ell A_i x^\ell - \delta_\ell.$$

Then, by the Cauchy–Schwartz inequality,

$$|h_i^\ell| \le \|A_i x^\ell\|\,\|d^\ell\| + |\delta_\ell|; \tag{5.4}$$

since

$$(x^\ell, \lambda^\ell) \subset S_0$$

we conclude that (see the proof of Theorem 2)

$$\|x^\ell\| \le \rho < \infty,$$

and hence

$$M_2 := \max_i \left\{\|A_i x^\ell\|\right\} \le \rho \max_i \|A_i\| < \infty.$$

By (5.4), then,

$$|h_i^\ell| \le M_2\|d^\ell\| + |\delta_\ell|. \tag{5.5}$$

Consider the following implications, valid for all $\eta \in R$.

$$0 < \alpha \le \frac{\epsilon}{(U_i - L_i)|\eta|} \Rightarrow \begin{cases} -\alpha L_i\eta - \epsilon \le -\alpha U_i\eta, \\ -\alpha U_i\eta - \epsilon \le -\alpha L_i\eta. \end{cases}$$

Choose $\eta = h_i^\ell = d^\ell A_i x^\ell - \delta_\ell$ and use the bound (5.5) to obtain, for all $i = 1, \dots, m$

$$\begin{aligned} &0 < \alpha \le \alpha_1^\ell := \frac{\epsilon}{\max\limits_{i=1,\dots,m} (U_i - L_i)[M_2\|d^\ell\| + |\delta_\ell|]} \\ &\Rightarrow \begin{cases} -\alpha L_i h_i^\ell - \epsilon \le -\alpha U_i h_i^\ell & \text{(a)}, \\ -\alpha U_i h_i^\ell - \epsilon \le -\alpha L_i h_i^\ell & \text{(b)}. \end{cases} \end{aligned} \tag{5.6}$$

Let $i \in J_\ell^+$, i.e.,

$$U_i p_i^\ell > L_i p_i^\ell + \epsilon, \tag{5.7}$$

and let $0 < \alpha \leq \alpha_1^\ell$. Then

$$F_i(x^\ell + \alpha d^\ell, \lambda^\ell + \alpha\delta_\ell) = \max \left\{ \begin{array}{l} U_i p_i^\ell + \alpha U_i h_i^\ell + \frac{1}{2}\alpha^2 U_i d^\ell \boldsymbol{A}_i d^\ell \,; \\ \\ L_i p_i^\ell + \alpha L_i h_i^\ell + \frac{1}{2}\alpha^2 L_i d^\ell \boldsymbol{A}_i d^\ell \end{array} \right\}$$
$$\leq \tfrac{1}{2}\alpha^2 M_1 \|d^\ell\|^2 + \max\{U_i p_i^\ell + \alpha U_i h_i^\ell,\; U_i p_i^\ell - \epsilon + \alpha L_i h_i^\ell\}$$

by definition of M_1, (5.7), and the Cauchy–Schwartz inequality,

$$\leq \tfrac{1}{2}\alpha^2 M_1 \|d^\ell\|^2 + U_i p_i^\ell + \alpha U_i h_i^\ell,$$

by (5.6b).

Since for $i \in J_\ell^+$, $U_i p_i^\ell = F_i(x^\ell, \lambda^\ell)$ and using the definition of h_i^ℓ, the last inequality yields

$$F_i(x^\ell + \alpha d^\ell, \lambda^\ell + \alpha\delta_\ell) \leq \tfrac{1}{2}\alpha^2 M_1 \|d^\ell\|^2 + F_i(x^\ell, \lambda^\ell)$$
$$+\alpha U_i d^\ell \boldsymbol{A}_i x^\ell - \alpha U_i \delta_\ell \qquad \text{for all } i \in J_\ell^+.$$

Summing for all $i \in J_\ell^+$ we get

$$B \leq \sum_{i \in J_\ell^+} F_i(x^\ell, \lambda^\ell) + \alpha \left(\sum_{J_\ell^+} U_i \boldsymbol{A}_i x^\ell \right) d^\ell - \alpha \left(\sum_{J_\ell^+} U_i \right) \delta_\ell$$
$$+ \tfrac{1}{2}\alpha^2 M_1 \|d^\ell\|^2 m_B,$$

where $m_B = \text{card}\,(J_\ell^+)$. Similarly we can obtain

$$C \leq \sum_{i \in J_\epsilon^-} F_i(x^\ell, \lambda^\ell) + \alpha \left(\sum_{J_\ell^-} L_i \boldsymbol{A}_i x^\ell \right) d^\ell - \alpha \left(\sum_{J_\ell^-} L_i \right) \delta_\ell$$
$$+ \tfrac{1}{2} M_1 \|d^\ell\|^2 m_c,$$

where $m_c = \text{card}\,(J_\ell^-)$. Combining the above inequalities for A, B, C, we get

$$F(x^\ell + \alpha d^\ell,\ \lambda^\ell + \alpha\delta_\ell) = A + B + C + \lambda^\ell v - f x^\ell + \alpha\delta^\ell v - \alpha f d^\ell$$
$$\leq \sum_{i=1}^{m} F_i(x^\ell, \lambda^\ell) + \lambda^\ell v - f x^\ell + \alpha\delta_\ell \left(v - \sum_{J_\ell^+} U_i - \sum_{J_\ell^-} L_i - v^\ell \right)$$
$$-\alpha \left(f - \sum_{J_\ell^+} U_i \boldsymbol{A}_i x^\ell - \sum_{J_\ell^-} L_i \boldsymbol{A}_i x^\ell - f^\ell \right) d^\ell$$
$$+\tfrac{1}{2}\alpha(M_1 m\alpha - 1)\|d^\ell\|^2 - \tfrac{1}{2}\alpha\delta_\ell^2.$$

By the definitions of f^ℓ, v^ℓ the last inequality is

$$(5.8) \qquad F(x^\ell + \alpha d^\ell, \lambda^\ell + \alpha\delta_\ell) \leq F(x^\ell, \lambda^\ell) + \tfrac{1}{2}\alpha(M_1 m\alpha - 1)\|d^\ell\|^2 - \tfrac{1}{2}\alpha\delta_\ell^2,$$

which holds for all $0 < \alpha \le 1, \alpha \le \alpha_1^\ell$.

Let $\alpha_2^\ell := ((1-2\theta)/(M_1 m))$ where $0 < \theta < \frac{1}{2}$. Then for $\alpha \le \alpha_2^\ell$,

$$\tfrac{1}{2}\alpha(M_1 m\alpha - 1) \le -\theta\alpha.$$

Therefore, by (5.8), for all $0 < \alpha \le \bar{\alpha}_\ell, 0 < \theta < \frac{1}{2}$,

$$F(x^\ell + \alpha d^\ell, \lambda^\ell + \alpha\delta_\ell) \le F(x^\ell, \lambda^\ell) - \theta\alpha\|d^\ell\|^2 - \alpha\theta\delta_\ell^2, \tag{5.9}$$

where

$$\bar{\alpha}_\ell = \min(1, \alpha_1^\ell, \alpha_2^\ell).$$

The stepsize α_ℓ in the algorithm is chosen to be the largest $\alpha > 0$ satisfying (5.9). Hence

$$\alpha_\ell \ge \bar{\alpha}_\ell \tag{5.10}$$

and

$$F(x^{\ell+1}, \lambda^{\ell+1}) = F(x^\ell + \alpha_\ell d^\ell, \lambda^\ell + \alpha_\ell\delta_\ell) \le F(x^\ell, \lambda^\ell) - \theta\alpha_\ell\left(\|d^\ell\|^2 + \delta_\ell^2\right). \tag{5.11}$$

As $\ell \to \infty$, it follows from (5.11) that

$$\alpha_\ell\left(\|d^\ell\|^2 + \delta_\ell^2\right) \to 0. \tag{5.12}$$

Now, by (4.3) and the facts

$$L_i \le t_i^\ell \le U_i,$$

$$\|\boldsymbol{A}_i x^\ell\| \le M_2,$$

it follows that $\|d^\ell\|$ and $|\delta^\ell|$ are bounded above. Hence (see definition of α_1^ℓ in (5.6)) α_1^ℓ is bounded away from zero, and hence also $\bar{\alpha}_\ell$. It follows from (5.10) and (5.11) that, when $\ell \to \infty, \delta_\ell \to 0$ and $d^\ell \to 0$.

As $\ell \to \infty$, we also have

$$x^\ell \to \bar{x}, \quad \lambda^\ell \to \bar{\lambda}, \quad h_i^\ell \to 0, \quad p_i^\ell \to \bar{p}_i, \quad t_i^\ell \to \bar{t}_i.$$

Also, by (4.15), with $\bar{\mu}_i := \lim_{\ell\to\infty} \mu_i^\ell$

$$\begin{aligned}
\bar{\mu}_i &= \max\{U_i\bar{p}_i, L_i\bar{p}_i\} && \text{if } |\bar{p}_i| \le \tfrac{\epsilon}{U_i - L_i},\\
\bar{\mu}_i &= L_i\bar{p}_i && \text{if } \bar{p}_i < \tfrac{-\epsilon}{U_i - L_i},\\
\bar{\mu}_i &= U_i\bar{p}_i && \text{if } \bar{p}_i > \tfrac{\epsilon}{U_i - L_i}.
\end{aligned}$$

Hence

$$\bar{\mu}_i = \max\{U_i\bar{p}_i, L_i\bar{p}_i\} \qquad \text{for all } i = 1, \ldots, m.$$

Letting $\ell \to 0$ in (4.10)–(4.15), we see that $\bar{x}, \bar{\lambda}, \bar{z}_i = \bar{\mu}_i, \bar{t}_i$ satisfy the optimality condition (4.16)–(4.21) for Problem $(P2)$. Hence $(\bar{x}, \bar{\lambda})$ is its optimal solution. □

6. Truss topology with free design variables. An important special case of the truss topology problem is where the design variable $\{t_i\}$ is free of the upper and lower bounds constraints $L_i \leq t_i \leq U_i$, i.e., t_i is only required to be nonnegative. Problem $(P1)$ reduces then to

$$(P1)_s \qquad \min\left\{\tfrac{1}{2}fx : \sum_{i=1}^{m} t_i \boldsymbol{A}_i x = f,\ \sum_{i=1}^{m} t_i = v,\ \ t_i \geq 0\right\}.$$

Note that the volume constraint indirectly imposes an upper bound $t_i \leq v$. Hence problem $(P1)_s$ is a special case of $(P1)$ with

$$L_i = 0, \quad U_i = v, \quad i = 1, 2, \ldots, m.$$

The equivalent displacement-based problem $(P2)$ is then

$$(6.1) \qquad \min_{x,\lambda}\left\{\lambda v - fx + \sum_{i=1}^{m} \max\left\{\left(\tfrac{1}{2}x\boldsymbol{A}_i x - \lambda\right) v, 0\right\}\right\}.$$

From Theorem 5, it follows easily that for any given x, the minimizing λ in (6.1) is

$$\lambda = \max_{i=1,\ldots,m}\left\{\tfrac{1}{2}x\boldsymbol{A}_i x\right\}.$$

Substituting this value in (6.1), we see that Problem $(P2)$ reduces to a simple convex minmax problem involving only displacement variables:

$$(P2)_s \qquad \min_{x}\left\{F(x) := \max_{i=1,\ldots,m}\left\{\tfrac{v}{2}x\boldsymbol{A}_i x - fx\right\}\right\}.$$

For this problem, an ϵ-steepest descent direction $d^\ell \in \Re^n$ of $F(\cdot)$ at x^ℓ is the solution of the quadratic program.

$$(6.2) \qquad \min\{\mu + \tfrac{1}{2}\|d\|^2\}$$

$$(\hat{P}_\ell)_s \qquad \text{subject to}$$

$$d^T(v\boldsymbol{A}_i x^\ell - f) + q_i^\ell - \mu \leq 0, \qquad i \in I_\ell,$$

where

$$q_i^\ell := \tfrac{v}{2}x^\ell \boldsymbol{A}_i x^\ell - fx^\ell,$$

$$I_\ell := \{i \,:\, q_i^\ell > F(x^\ell) - \epsilon\}.$$

The dual problem of $(\hat{P}_\ell)_s$ is here

$$(\hat{D}_\ell)_s \qquad \max_t\left\{\frac{1}{2}\sum_{i\in I_\ell} t_i x^\ell \boldsymbol{A}_i x^\ell - \frac{1}{2}\left\|\sum_{i\in I_\ell} t_i \boldsymbol{A}_i x^\ell - f\right\|^2\right\}$$

$$\text{subject to} \sum_{i\in I_\ell} t_i = v, \quad t_i \geq 0, \quad i \in I_\ell.$$

ALGORITHM B [For solving $(P2)_s$]
Parameters: $\epsilon > 0$ (activity), $\delta > 0$ (stopping rule), $0 < \theta < \frac{1}{2}$ (stepsize rule),
Initialization: *Choose* $t^0 > 0$, $\sum_{i=1}^{m} t^0 = v$, *compute* x^0, the unique solution of

$$\sum t_i^0 A x = f.$$

Step ℓ (x^ℓ given)

(ℓ.1) *Compute* q_i^ℓ, $F(x^\ell) = \max_{i=1,\dots,m}\{q_i^\ell\}$ and the index set I_ℓ.

(ℓ.2) *Compute* the search direction d^ℓ by solving the quadratic program $(\hat{P}_\ell)_s$, or by solving the dual $(\hat{D}_\ell)_s$, to obtain the solution t^ℓ, and then set

$$d^\ell = -\left(\sum_{I_\ell} t_i^\ell A_i x^\ell - f\right).$$

(ℓ.3) If $\|d^\ell\| < \delta$ stop, x^ℓ is the solution of $(P2)_s$ [t^ℓ is the solution of $(P1)_s$] *else*, go to (ℓ.4).

(ℓ.4) *Compute* the stepsize α_ℓ by the formula

$$\alpha_\ell = \min_{i=1,\dots,m} \{\alpha_i^\ell\}, \tag{6.3}$$

where

$$\alpha_i^\ell = \begin{cases} -c_i^\ell / b_i^\ell & \text{if } a_i^\ell = 0, \quad b_i^\ell > 0, \\ \dfrac{-b_i^\ell + \sqrt{(b_i^\ell)^2 - 4a_i^\ell c_i^\ell}}{2a_i^\ell} & \text{if } a_i^\ell > 0, \\ \infty & \text{if } a_i^\ell = 0, \quad b_i \le 0; \end{cases}$$

here the numbers a_i^ℓ, b_i^ℓ, c_i^ℓ are given by

$$a_i^\ell = \tfrac{v}{2} d^\ell A_i d^\ell \ge 0,$$

$$b_i^\ell = d^\ell(vA_i x^\ell - f) + \theta\|d^\ell\|^2,$$

$$c_i^\ell = q_i^\ell - F(x^\ell) \le 0;$$

(ℓ.5) $x^{\ell+1} = x^\ell + \alpha_\ell d^\ell$;

(ℓ.6) $\ell \leftarrow \ell + 1$, go to (ℓ.1).

To explain the analytic formula (4.3) for the stepsize α_ℓ, we first note that the stepsize rule (4.22) in Algorithm A reduces in our special case to

(6.4) α_ℓ is the largest $\alpha \ge 0$ such that $F(x^\ell + \alpha d^\ell) \le F(x^\ell) - \alpha\theta\|d^\ell\|^2$.

We now prove the following theorem.

THEOREM 9. *The stepsize given by* (6.3) *is the solution of* (6.4).

Proof. Inequality (6.4) is specifically

$$\tfrac{v}{2}(x^\ell + \alpha d^\ell) A_i (x^\ell + \alpha d^\ell) - f(x^\ell + \alpha d^\ell) \le F(x^\ell) - \alpha\theta\|d^\ell\|^2, \qquad i = 1, \dots, m,$$

which further reduces to

$$\alpha d^\ell(vA_ix^\ell - f) + \tfrac{v}{2}\alpha^2 d^\ell A_i d^\ell + q_i^\ell \le F(x^\ell) - \alpha\theta\|d^\ell\|^2, \qquad i = 1,\dots,m. \tag{6.5}$$

Define, for $i = 1,\dots,m$,

$$\varphi_i(\alpha) := \alpha^2\left(\tfrac{v}{2}d^\ell A_i d^\ell\right) + \alpha\left[d^\ell(vA_ix^\ell - f) + \theta\|d^\ell\|^2\right] + q_i^\ell - F(x^\ell).$$

Then (6.5) is just

$$\varphi_i(\alpha) \le 0, \qquad i = 1,\dots,m. \tag{6.6}$$

Now,

$$\varphi_i(0) = q_i^\ell - F(x^\ell)\begin{cases} = 0 & \text{if } i \in I^0(x^\ell), \\ < 0 & \text{otherwise;}\end{cases}$$
$$I^0(x^\ell) := \{i : q_i^\ell = F(x^\ell)\},$$

and

$$\varphi_i'(0) = d^\ell(vA_ix^\ell - f) + \theta\|d^\ell\|^2.$$

Recall that d^ℓ (together with μ_ℓ) is an optimal solution of $(\hat{P}_\ell)_s$; since $d = 0$, $\lambda = \max\{q_i^\ell\} = F(x^\ell)$ is a feasible solution of $(\hat{P}_\ell)_s$, we have

$$\mu_\ell + \tfrac{1}{2}\|d^\ell\|^2 \le F(x^\ell). \tag{6.7}$$

Therefore, for $i \in I^0(x^\ell)$, it follows from (6.2), (6.7) that

$$d^\ell(vA_ix^\ell - f) + \tfrac{1}{2}\|d^\ell\|^2 \le 0,$$

and since $0 < \theta < \frac{1}{2}$,

$$d^\ell(vA_ix^\ell - f) + \theta\|d^\ell\|^2 < 0,$$

i.e.,

$$\varphi_i'(0) < 0 \quad \text{for } i \in I^0(x^\ell). \tag{6.8}$$

From the above discussion, the stepsize α_ℓ solving (6.4) is given by

$$\alpha_\ell = \arg\max\{\alpha : \varphi_i(\alpha) \le 0, \alpha > 0\}.$$

Each function φ_i is convex, and

$$\begin{aligned} &\varphi_i(0) = 0, \quad \varphi_i'(0) < 0, \quad i \in I^0(x^\ell), \\ &\varphi_i(0) < 0, \qquad i \notin I^0(x^\ell). \end{aligned} \tag{6.9}$$

Thus (see Fig. 3), each φ_i has at most one root α_i^ℓ in $(0,\infty)$ and

$$\alpha_\ell = \min_{i=1,\dots,m}\{\alpha_i^\ell\}. \tag{6.10}$$

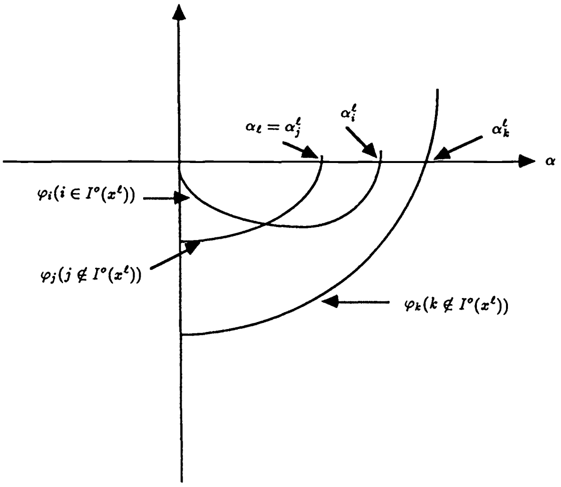

$\alpha_\ell = \alpha_j^\ell$
α_i^ℓ
α_k^ℓ
α
$\varphi_i(i \in I^\circ(x^\ell))$
$\varphi_j(j \notin I^\circ(x^\ell))$
$\varphi_k(k \notin I^\circ(x^\ell))$

FIG. 3. *Computation of the stepsize.*

Denote the coefficients of the quadratic function $\varphi_i(\cdot)$ by

$$a_i^\ell = \tfrac{v}{2} d^\ell A_i d^\ell \geq 0,$$

$$b_i^\ell = d^\ell(vA_i x^\ell - f) + \theta \|d^\ell\|^2,$$

$$c_i^\ell = q_i^\ell - F(x^\ell) \leq 0.$$

Then α_i^ℓ is given by

$$\alpha_i^\ell = \begin{cases} -c_i^\ell / b_i^\ell & \text{if } a_i^\ell = 0, \quad b_i^\ell > 0, \\ \frac{-b_i^\ell + \sqrt{(b_i^\ell)^2 - 4a_i^\ell c_i^\ell}}{2a_i^\ell} & \text{if } a_i^\ell > 0, \\ \infty & \text{if } a_i = 0, \quad b_i \leq 0, \end{cases}$$

and so (6.10) agrees with (6.3). □

7. Computational results. In this section, we will present a number of results obtained by using Algorithm B. For clarity, we concentrate on $(P1)_s$ with free design variables $(t_i \geq 0)$. Thus, this section will deal with the implementation of Algorithm B.

First, we note that the algorithm only requires computation of vectors $A_i x$ and numbers $y^T A_i x$. Thus we need not assemble nor store the matrices A_i, nor must we assemble the entire matrix A at any iteration step. The compatibility matrix should also not be stored (each column contains at most 2× dim nonzero elements), but instead one works with the $2 \times m$ matrix of connectivities, giving the numbers of the nodal points to which a given bar is connected, as well as a matrix of bar cosines. This means that even though our primal variables are connected to the nodal points, all computations and storage are based on bar numbers. In our implementation, the search vector d was always computed by solving the dual problem (D_ℓ) (or $(\hat{D}_\ell)_s$) in the active bar volumes t_i (i.e., J_ℓ or I_ℓ), as it is our experience that the number of (almost) active bars is considerably less than the number of degrees of freedom for the full truss. Finally, for the linesearch, both the (analytic) Armijo–Goldstein search and an exact linesearch have been tried. It turns out that the inexact search is typically very conservative and that the exact linesearch, especially for larger problems, gives a better performance. In the implementation of the linesearch (golden section method), in order to save costly function-calls, we do not use *all* bars, but only a subset of the $\hat{\epsilon}$-active one ($\hat{\epsilon}$ is larger than ϵ, typically $\hat{\epsilon} = 10\epsilon$). The full set of bars is used only if such a search does not improve the value of the objective function.

For the truss topology optimization (with $L_i = 0$), we are interested in the ultimate set of active bars

$$I_0(x^*) = \{i : t_i^* > 0\}.$$

It is true, however (see similar claims in, e.g., [4]), that for all sufficiently small $\epsilon > 0$, there exists a neighborhood N^* of x^* such that

$$I_\epsilon(x) = I_0(x^*) \quad \text{for all } x \in N^*.$$

It is thus natural to work with a decreasing sequence of ϵ-values. It was found that it is important not to choose ϵ too small for the first iterations, and that it is a good strategy to work with a sequence of alternatingly decreasing values of the ϵ-parameter as well as the stopping parameter δ. We note here that the final δ should be at least small enough that we can accept δ as an error in the satisfaction of the equilibrium equations.

The special problem $(P1)_s$ is made up of expressions which are elementwise linear in all variables, except geometric data. Thus, for a specific choice of ground structure geometry and load vector *direction*, the optimal topology only needs to be computed for one set of assigned values of Young's modulus E, volume v, load size f, and geometric scale; for any other values of these variables, the optimal values of the design variables t, the deformation x, and the compliance fx can be derived by a simple scaling. Thus, $(P1)_s$ lends itself to the creation of a "catalogue of optimal topologies" for both single and multiple loads. The optimal compliance may then conveniently be given in terms of the nondimensional compliance ϕ,

$$\phi = (f^T x) v E / (\|f\|^2 \ell^2),$$

where ℓ is a typical length dimension (horizontal length of truss in the examples that follow). For Problem $(P1)$, the optimal compliance should also be given in terms of ϕ and the bounds, L_i, U_i, in terms of ratios of the volume v.

Examples of optimal topologies are shown in Figs. 4–7. In these examples, where all connections between nodal points are used as the ground structure, overlapping connecting bars between two nodal points have been removed so as to avoid a redundancy in the model and a trivial possibility of subspaces of optimal solutions. In the optimal topologies, some straight bars appear with intermediate nodal points with no other connecting bars. Such bars should be thought of as straight bars without these intermediate nodal points, as a truss model under the given load will not be able to distinguish between the two configurations.

The final topology and the performance of the optimal structure depend intimately on the choice of ground structure, as does the performance of the algorithm. If the optimal topology consists of only a very low number of bars, the algorithm predicts this very quickly, even though the potential number of bars is large. However, it is also required

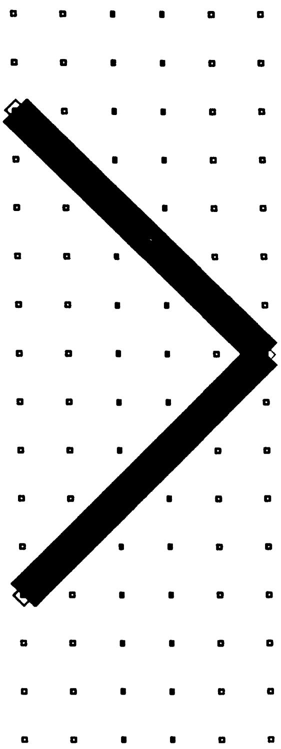


FIG. 4(a). *The optimal truss for a ground structure with* 2852 *potentials bars.*

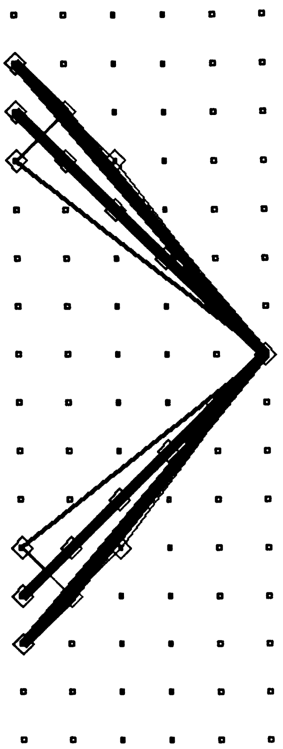


FIG. 4(b). *The optimal truss for the same ground structure as in Fig.* 4(a) *but with upperbounds on bar volumes.*

that "nature's optimal topology" is indeed a subset of the bars in the ground structure; if not, the algorithm will find approximations (however, the topologies are optimal for each choice of ground structure), usually involving many bars. It is well known that the best structure for carrying a single load which is parallel to a line of possible support is a two-bar truss with trusses at 45^o to the line of support (cf. Rosvany [17]). Such a situation is mimicked in all the examples shown, but only the structure in Fig. 4 allows for this optimum as part of its ground structure. The ground structure of Fig. 4 consists of all 2852 nonoverlapping connections between the equally spaced 6×16 nodes in a 10×30 rectangle. All left-hand nodes are possible supports and the single vertical force is at the mid right-hand node. Figure 4(a) shows the optimal, two-bar truss obtained when no constraints on the bar volumes t_i are imposed and the optimal nondimensional compliance is 4.0. In Fig. 4(b), upper bounds on the bar volumes are imposed, as $U_i = 0.01 \cdot \ell_i \cdot v$, and the compliance ϕ increases to 4.1092. The result in Fig. 4(a) was computed using Algorithm B and the result in Fig. 4(b) is the result of using Algorithm A; for the latter example, the deformation field x of Fig. 4(a) was used as the starting point of the algorithm. Notice that introducing upper bounds on the design variables, as expected,

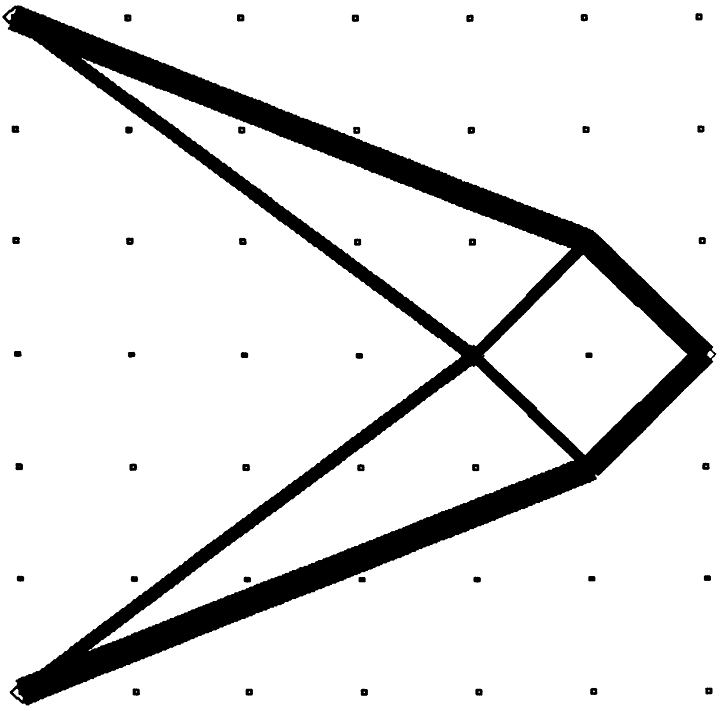


FIG. 5(a). *The optimal truss for the ground structure of Fig.* 1(a).

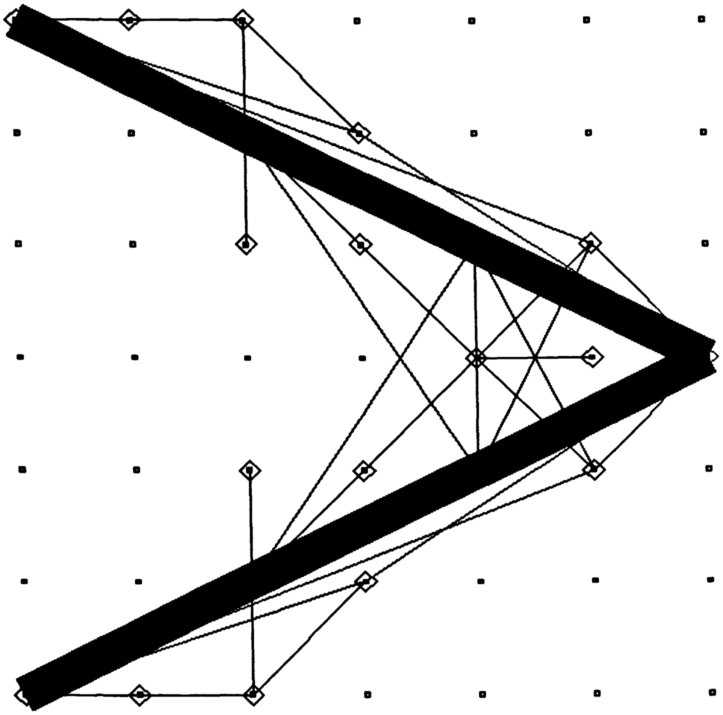


FIG. 5(b). *The optimal multiload design of a truss corresponding to the ground structure of Fig.* 1(a).

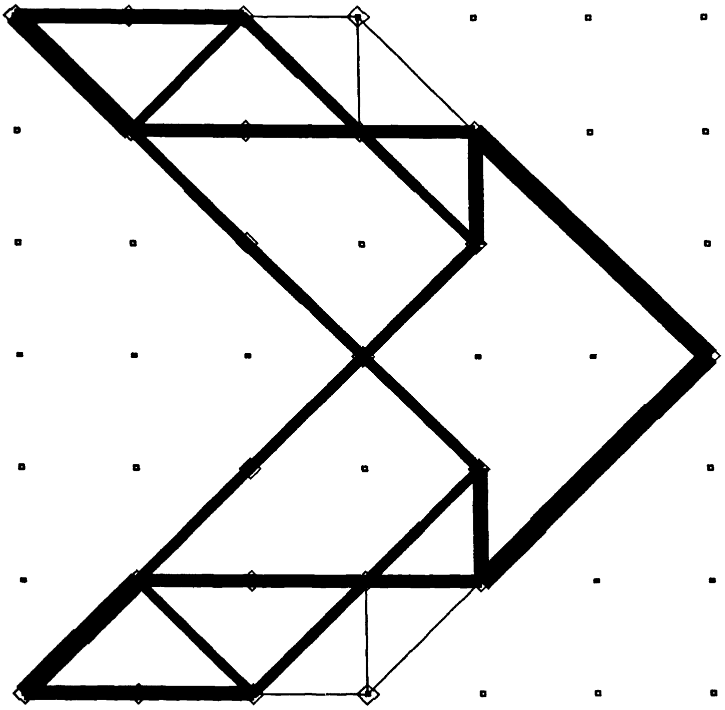


FIG. 6(a). *The optimal truss for the ground structure of Fig.* 1(b)*—single-load case.*

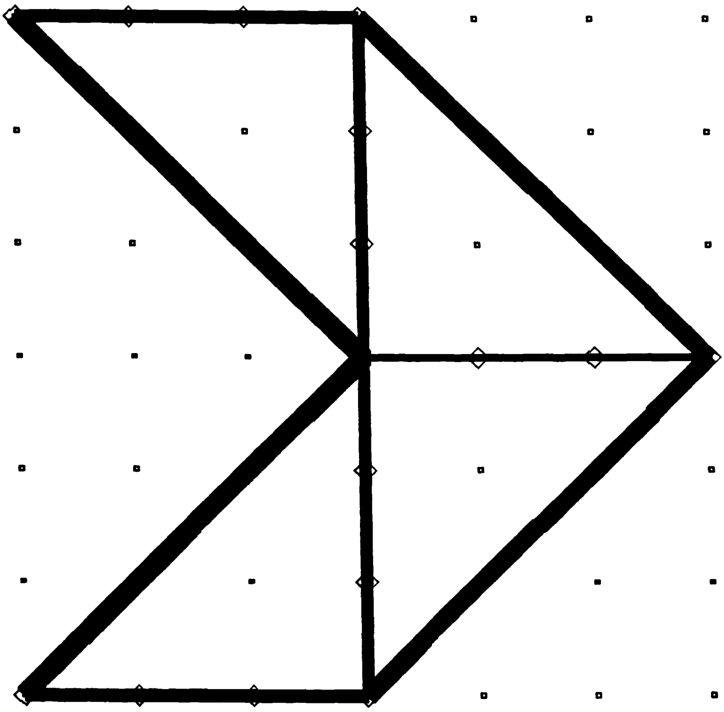


FIG. 6(b). *The optimal truss for the ground structure of Fig.* 1(b)*—three-load cases.*

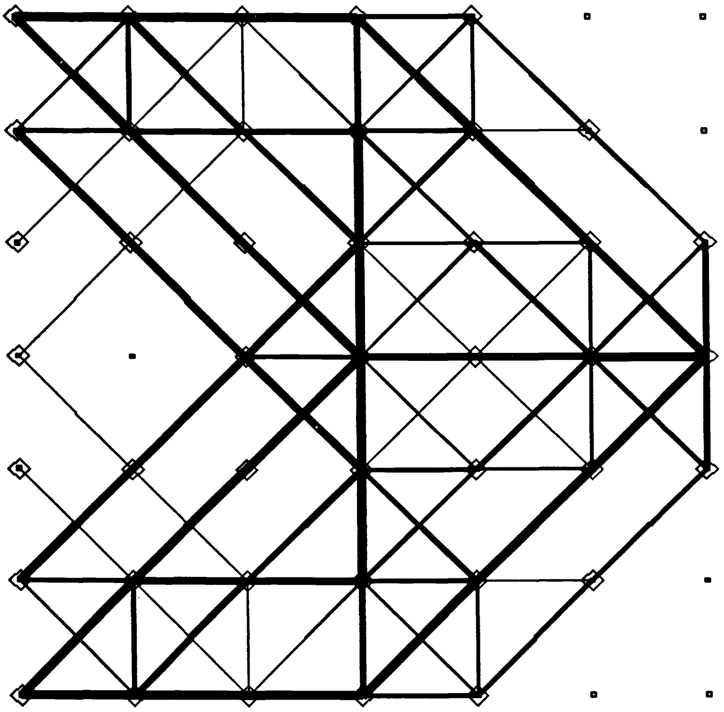


FIG. 6(c). *The optimal truss for the structure of Fig.* 6(b) *but with upper bounds on bar volumes.*

increases the number of bars in the structure as well as increasing the number of bars in the active set J_ℓ.

In Fig. 5(a) we show the optimal, unconstrained truss topology for the ground structure and loading condition of Fig. 1(a). The compliance ϕ is 6.0134, i.e., 1.5 times greater than for the two-bar truss of Fig. 4(a). In Fig. 5(b), an extra, horizontal load has been added at the loaded node and the figure shows the multiload design obtained for unconstrained design variables. The horizontal and vertical loads are equal in size and the weights on the compliances are 1.0 and 2.0, respectively. The average nondimensional compliance is 4.6943 and the compliances for each of the loads are 6.2541 and 1.5747, respectively. The multiload problem results in what is in practice a two-bar truss (trusses at ± 30° with horizontal direction), thus giving a simpler geometric layout. This feature is even more apparent in the example of Fig. 6, where we use the ground structure of Fig. 1(b). In Figure 6(a), we have the one-load case corresponding to the ground structure in Fig. 1(b), while in Fig. 6(b), we have three load cases: a horizontal and a vertical load at the mid, a right-hand node and a vertical load at the mid node, all of equal size and weighted 1.0, 2.0 and 1.0, respectively. Finally, in Fig. 6(c), we have a design-constrained ($U_i = 0.01 \cdot v \cdot \ell_i$) topology for the same ground structure and set of loads. For the unconstrained problem, the average compliance is 6.3737 and the individual compliances are 3.752, 9.4577, and 2.8273; with constraints the values are 7.2108 and 4.1401, 10.5957, and 3.5117.

In Fig. 7 we illustrate the effect of increasing the number of nodal points (and potential bars) for a ground structure geometry for which "nature's optimal topology" is

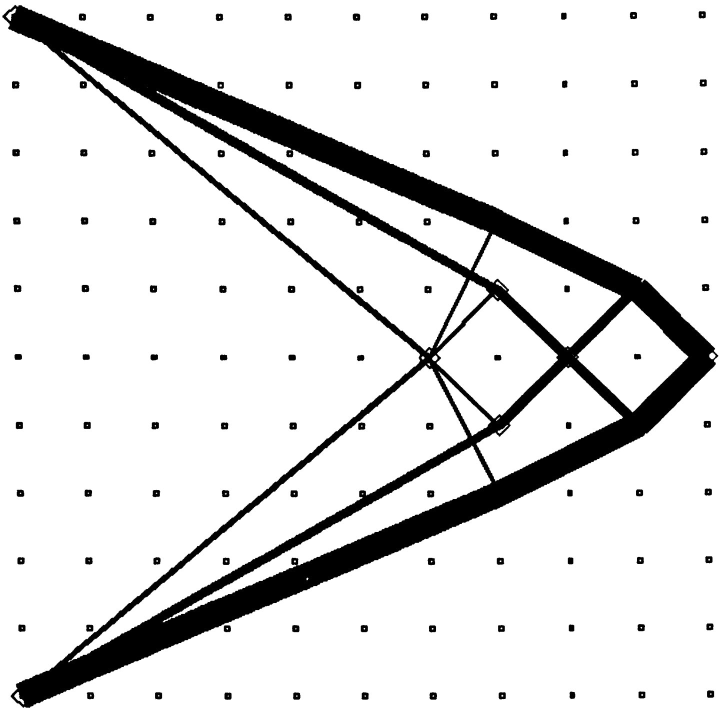


FIG. 7(a). *The optimal truss for the structure with the same geometry and load as in Fig.* 1(a), *with* 11 × 11 *nodes* (4492 *potential bars*).

a so-called Michell truss [12], [9], i.e., a curve-linear layout of a continuum of unidirectional load-bearing members. In Fig. 7, we have the same geometry and load as in Fig. 1(a). We allow all connections between nodes and have increased the number of nodes to an 11 × 11 (Fig. 7(a)) and a 15 × 15 (Fig. 7(b)) equidistant layout of nodes, giving 4492 and 15556 nonoverlapping connections and 5.9646 and 5.9344 nondimensional compliances, respectively. The number of bars in the optimal topology increases dramatically as the layout tries to mimic the curved layout of the optimum Michell truss, thus approximating a layout which is at the limit of the range of a truss model; similar behavior is seen in plate optimization and shape design (cf. [3]). The high number of active bars in the final topology slows the algorithms considerably and indicates that it is important to make a suitable choice of ground structure when optimizing topology.

Finally, it should be noted that the optimal compliance value fx^* is not very sensitive to variations in the values of the design variables. Small variations in the cross-sectional areas of the bars in the optimal topology and even the addition or deletion of thin bars have very little influence on the stiffness of the truss, as measured by compliance. Also, multiple solutions seem to exist, especially in cases with possible symmetry. These remarks are but experimental observations. However, some of them can be substantiated thoeretically by using results from, e.g., [2].

Acknowledgements. A guest professorship at the Technical University of Denmark is gratefully acknowledged by A. Ben-Tal.

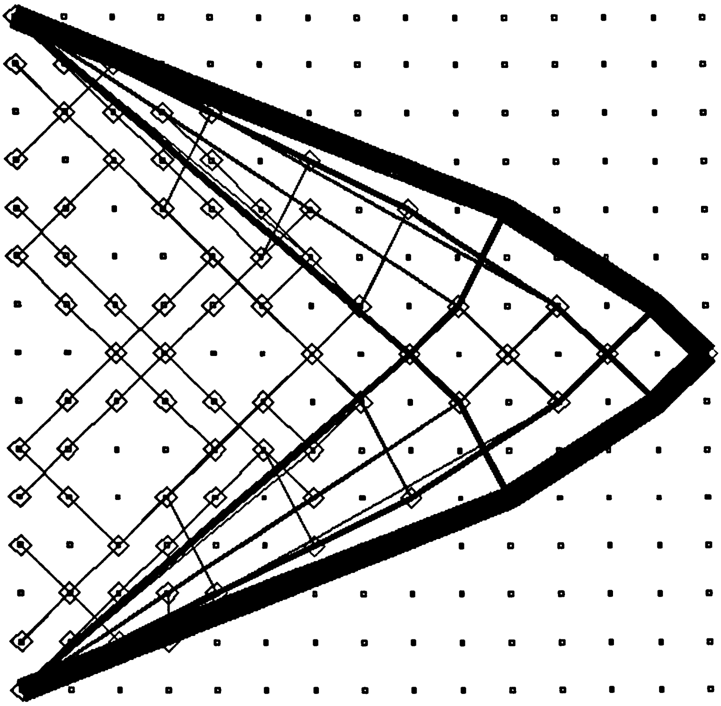


FIG. 7(b). *The optimal truss for the structure with the same geometry and load as in Fig.* 1(a), *with* 15 × 15 *nodes* (15556 *potential bars*).

REFERENCES

[1] W. ACHTZIGER, M. P. BENDSØE, A. BEN-TAL, AND J. ZOWE (1991), *New formulations of truss topology design problems*, Tech. Rep., Mathematical Institute, The Technical Univ. of Denmark, Lyngby, Denmark.

[2] H. ATTOUCH (1984), *Variational Convergence of Functions and Operations*, Pitman, Boston.

[3] M. P. BENDSØE AND N. KIKUCHI (1988), *Generating optimal topologies in structural design using a homogenization method*, Comput. Methods Appl. Mech. Engrg., 71, pp. 197–224.

[4] V. F. DEMYANOV AND V. H. MALOZEMOV (1974), *Introduction to Minimax*, John Wiley, New York.

[5] W. DORN, R. GROMORY, AND H. GREENBERG (1964), *Automatic design of optimal structures*, J. Mécanique, 3, pp. 25–52.

[6] P. FLERON (1964), *The minimum weight of trusses*, Bygnings Statiske Meddelelser, 35, pp. 81–96.

[7] R. T. HAFTKA, Z. GÜRDAL, AND M. P. KAMAT (1990), *Elements of Structural Optimization*, 2nd ed., Kluwer Academic Publishers, Dordrecht, the Netherlands.

[8] W. S. HEMP (1973), *Optimum Structures*, Clarendon Press, Oxford, U.K.

[9] U. KIRSCH (1989a), *Optimal topologies of structures*, Appl. Mech. Rev., 42, pp. 223–239.

[10] ——— (1989b), *Optimal topologies of truss structures*, Comp. Meth. Appl. Mech. Engrg., 72, pp. 15–28.

[11] C. LEMARECHAL (1989), *Nondifferentiable optimization*, in Optimization, Vol. 1, G. L. Nemhauser, H. A. G. Rinnooy Kan, and M. J. Todd, eds., North Holland, Amsterdam.

[12] A. G. M. MICHELL (1904), *The limits of economy of material in frame structures*, Philosophical Magazine, Ser. 6, Vol. 8, pp. 589–597.

[13] P. PEDERSEN (1970), *On the minimum mass layout of trusses*, AGARD Conf. Proc., No. 36, Symposium on Structural Optimization, AGARD–CP–36–70 Paris, France.

[14] B. N. PSHENICHNY AND Y. M. DANILIN (1978), *Numerical Methods in Extremal Problems*, MIR Publishers, Moscow.

[15] U. RINGERTZ (1985), *On Topology Optimization of Trusses*, Engrg. Optimization, 9, pp. 21–36.

[16] R. T. ROCKAFELLAR (1970), *Convex Analysis*, Princeton University Press, Princeton, NJ.

[17] G. I. N. ROZVANY (1989), *Structural Design Via Optimality Criteria*, Kluwer, Dordrecht, The Netherlands.

[18] G. I. N. ROZVANY AND M. ZHOU (1990), *Applications of the* COC *Algorithm in Layout Optimization*, Proc. Internat. Conf. Engrg. Optimization in Design Processes, Karlsruhe; Lecture Notes in Engineering, Springer-Verlag, Berlin, New York, 63 (1991), pp. 59–70.

[19] K. SUZUKI AND N. KIKUCHI (1990), *A homogenization method for shape and topology optimization*, Comput. Methods Appl. Mech. Engrg., 93 (1991), pp. 291–318.

[20] B. H. V. TOPPING (1983), *Shape optimization of skeletal structures: A review*, ACSE J. Struct. Engrg., 109, pp. 1933–1951.

[21] G. N. VANDERPLAATS (1984), *Numerical methods for shape optimization: An assessment of the state of the art*, in New Directions in Optimum Structural Design, E. Atrek, R. H. Gallagher, K. M. Ragsdell, O. C. Zienkiewicz, eds., John Wiley, Chichester, U.K.

SIAM J. OPTIMIZATION
Vol. 3, No. 2, pp. 359–381, May 1993

QUANTITATIVE STABILITY OF VARIATIONAL SYSTEMS II. A FRAMEWORK FOR NONLINEAR CONDITIONING*

HEDY ATTOUCH† AND ROGER J.-B. WETS‡

Abstract. Stability results of Lipschitz and Hölder type are obtained for the solutions and optimal values of optimization problems when perturbations are measured in terms of the ρ-epi-distance.

Key words. quantitative stability, epi-convergence, epi-distance, projection, optimization, variational problems

AMS subject classifications. 65K10, 65K05, 49J40, 58E35

1. Introduction. Much has been said about the continuity properties of the optimal value and of the set of (optimal) solutions of optimization problems as a function of various perturbations. This is also the purpose of this paper, as well as its companion [8]. However, we make a break with the standard approach in at least two ways. First, we do not consider a particular class of perturbations, but allow for perturbations of a global character. The reference to *variational systems* in the title is aimed at stressing this concern; the term "variational systems" was used in [45] to designate a mapping $u \mapsto f_u$, with each f_u to be viewed as representing a certain optimization problem that depends on u. Second, we are concerned with *quantitative* results that could be used to obtain error bounds in the case of an approximating scheme or error estimates for the current solution of an algorithmic procedure.

An overview of the stability results that are topological in nature could be gathered from the work of Evans and Gould [19], Fiacco [20], Bank et al. [14], Dolecki [17], Gauvin [23], Hogan [27], and Zolezzi [53]; for a recent survey one should consult [21]. Epi-convergence, a concept exploited relatively recently, has allowed for the consolidation of a large number of these results; cf. Mosco [34], Wets [50], Attouch and Wets [4], Attouch [2], Robinson [38], Kall [28], and Beer and Lucchetti [15]. It will also provide the framework for this analysis.

The literature on quantitative stability results is much less abundant. The results are *local* in character and rely mostly on quantities associated with first- or higher-order (sub)derivatives, either of the functions defining the optimization problem or of the infimal (= marginal) function. There is the extensive work of Robinson [36], [37] on obtaining Lipschitz constants for solution mappings; see also Klatte and Kummer [30]. In some cases, one can apply results that come from the study of the (sub)derivatives of the marginal functions; cf. Rockafellar [41], [42] and Gauvin and Janin [24], [25]. Finally, there is the work based on the inverse function and implicit function theorems beginning with the Robinson–Urescu theorem; in the introduction of [39] Robinson sketches out a brief review. For example, Aubin [9] exploits the fact that locally, the optimal solutions of the optimization problem $\min_x f(x)$ are characterized by the optimality conditions

$$0 \in \partial f(x),$$

where $\partial f(x)$ is some set of generalized (sub)gradients of f at x; see also Aubin and Frankowska [10] and Aubin and Wets [11]. Then, with a surjectivity assumption on the

*Received by the editors December 1, 1991; accepted for publication (in revised form) January 31, 1992.

†Mathématiques, Université Montpellier II, 34095 Montpellier Cédex 5, France.

‡Department of Mathematics, University of California, Davis, California 95616. This work was supported in part by the National Science Foundation and the Air Force Office of Scientific Research.

tangent cone to the graph of ∂f at $(x, 0)$, one shows that the solution set $\partial^{-1} f$ is pseudo-Lipschitz at 0. The counterpart of the great generality and flexibility attained through this approach is the need to calculate second-order (generalized) derivatives of f and these calculations could be quite involved; there are also intrinsic limitations to the applicability of this approach brought out in [41].

We are mostly motivated by approximation questions, and thus we are interested in perturbations of a global character, and this leads to results of a somewhat different flavor than those mentioned above. However, in some situations one can profitably exploit "localized" versions of our results: the statement of all basic results always allows the replacement of any given function f by a function f^a that coincides with f on a neighborhood of the point of interest and is $+\infty$ outside this neighborhood.

To measure the distance between optimization problems, we rely on the ρ-epi-distance $\hat{dl}_\rho$, a distance notion introduced in [7] and briefly reviewed in §2; $\hat{dl}_\rho$ is also used in [8] to obtain Lipschitz continuity results for the approximate ε-solutions of convex optimization problems.

The main results are derived in §3. In particular, it is shown that the function $f \mapsto \inf f$ has locally Lipschitz properties with respect to $\hat{dl}_\rho$, and that for *well-conditioned* problems the optimal solutions have locally Hölder—and in some cases Lipschitz—continuity properties with respect to $\hat{dl}_\rho$. In §4, we apply these results to a constrained convex optimization problem, viz. the projection of a point on a moving convex set, and to the analysis of the convergence of algorithmic procedures based on penalization. Conditioning questions are raised in §5 and connections are established with well-posedness and the conditioning number associated with a (nonlinear) optimization problem. Finally, in §6 we show that the Hölder-like stability result is, in a certain sense, the best possible.

2. The epi-distance. We introduced the notion of epi-distance in [7] and made first use of it in deriving Lipschitz properties for the ε-approximate solutions of convex optimization problems [8]. We briefly recall the definition and state the Kenmochi conditions that will be needed in the sequel.

Unless otherwise specifically mentioned, X will be a normed linear space with norm $\|\cdot\|$ and d the metric induced by the norm. For any $C \subset X$,

$$d(x, C) := \inf_{y \in C} \|x - y\|$$

denotes the distance from x to C; if $C = \emptyset$ we set $d(x, C) = \infty$. $\mathbb{B} := \{\, x \mid \|x\| \leq 1 \,\}$ is the unit ball, $\rho\mathbb{B}$ the ball of radius $\rho \geq 0$, and $\mathbb{B}(x, \rho)$ the (closed) ball centered at x and with radius ρ. For any set $C \subset X$ and $\rho \geq 0$, we set

$$C_\rho := C \cap \rho\mathbb{B}.$$

For $C, D \subset X$, the "excess" of C on D is

$$e(C, D) := \sup_{x \in C} d(x, D),$$

with the (natural) convention that $e = 0$ if $C = \emptyset$; note that these definitions imply $e = \infty$ if D is empty and $C \neq \emptyset$. For $\rho \geq 0$, the ρ-distance between C and D is defined to be

$$\hat{dl}_\rho(C, D) := \sup\{\, e(C_\rho, D), e(D_\rho, C) \,\}.$$

We are going to define the ρ-epi-distance between two functions in terms of the ρ-distance between their epigraphs viewed as subsets of $X \times \mathbb{R}$. In this context, it is convenient to define the norm of $(x, \alpha) \in X \times \mathbb{R}$ as $\max[\,\|x\|, |\alpha|\,]$ with the associated metric, also denoted by d, defined accordingly. The unit ball $\mathbb{B} := \mathbb{B}_{X\times\mathbb{R}}$ is the (cylindrical) set $\{\,(x, \alpha) \mid \|x\| \leq 1, |\alpha| \leq 1\,\}$.

DEFINITION 2.1. For $\rho \geq 0$, the ρ-epi-distance between two extended real-valued functions f and g defined on X is

$$\hat{d\!l}_\rho(f, g) := \hat{d\!l}_\rho(\operatorname{epi} f, \operatorname{epi} g).$$

Thus, $\hat{d\!l}_\rho(f, g) \leq \eta$ means that

$$(\operatorname{epi} f)_\rho \subset \operatorname{epi} g + \eta\mathbb{B}, \qquad (\operatorname{epi} g)_\rho \subset \operatorname{epi} f + \eta\mathbb{B},$$

where

$$(\operatorname{epi} f)_\rho = \{\,(x, \alpha) \mid \alpha \geq f(x),\ x \in \rho\mathbb{B},\ |\alpha| \leq \rho\,\}.$$

Epi-convergence is, in the sense that can be made precise, the weakest notion of functional convergence that will guarantee the convergence of optimal solutions; cf. for example, [45], [2], and [28]. Convergence with respect to the family of "distances" $\{\,\hat{d\!l}_\rho, \rho > 0\,\}$ implies epi-convergence. When X is finite-dimensional the two notions of convergences coincide, and when X is infinite-dimensional, convergence of convex functions with respect to $\hat{d\!l}_\rho$ (for all ρ) implies Mosco–epi-convergence (epi-convergence with respect to both the weak and the strong topology on X); refer to [3] and [7, §4] for further details.

A very useful criterion, which allows us to compute or at least estimate the ρ-epi-distance, is provided by the *Kenmochi conditions.*

THEOREM 2.2 [7, Thm. 2.1]. *Let X be a normed linear space $f, g : X \to \overline{\mathbb{R}}$ proper and bounded below by $-\alpha(\|\cdot\|^p + 1)$ for some $\alpha \in \mathbb{R}_+$ and $p \geq 1$, and let*

$$\rho_0 > \max[\,d((0,0), \operatorname{epi} f), d((0,0), \operatorname{epi} g)\,].$$

Then,

(a) *for all $\rho > \rho_0$ and $x \in \operatorname{dom} f$ such that $\|x\| \leq \rho$, $|f(x)| \leq \rho$, and all $\varepsilon > 0$, there exists $\tilde{x}_\varepsilon \in \operatorname{dom} g$ that satisfies*

$$\|x - \tilde{x}_\varepsilon\| \leq \hat{d\!l}_\rho(f, g) + \varepsilon,$$
$$g(\tilde{x}_\varepsilon) \leq f(x) + \hat{d\!l}_\rho(f, g) + \varepsilon,$$

and similar conditions hold when the roles of f and g are interchanged;

(b) *if for all $\rho > \rho_0$ there exists $\eta_\rho \in \mathbb{R}_+$ such that for all $x \in \operatorname{dom} f$ with $\|x\| \leq \rho$, $|f(x)| \leq \rho$, there exists $\tilde{x} \in \operatorname{dom} g$ that satisfies*

$$\|x - \tilde{x}\| \leq \eta_\rho,$$
$$g(\tilde{x}) \leq f(x) + \eta_\rho,$$

and similar conditions hold for all $x \in \operatorname{dom} g$ with $\|x\| \leq \rho$, $|g(x)| \leq \rho$, then with $\rho_\dagger := \rho + \alpha(\rho^p + 1)$,

$$\hat{d\!l}_\rho(f, g) \leq \eta_{\rho_\dagger}.$$

We know [5], [6], [3] that there are many other ways to introduce distance notions on the space of lower semicontinuous (lsc) functions that induce epi-convergence. In fact, those discussed in [7, §3] are known to induce the same uniformities [7, Thms. 3.4, 3.7, 3.9]. We state our results in terms of $\widehat{d\!l}_\rho$ because in many applications $\widehat{d\!l}_\rho$ is easier to calculate or estimate, and usually comes with a more immediate geometric interpretation.

3. Stability results. We now turn to the basic results. Because the ρ-epi-distance $\widehat{d\!l}_\rho$ is not invariant under translations, and the origin plays a special role, the results cannot very well be localized away from the origin. It will thus be convenient to consider first the *canonical* case, which means that the point of reference will be a function f such that $0 = \min f = f(0)$. When the location of $\operatorname{argmin} f$ and the value of $\inf f$ are arbitrary, we shall translate f (and all other functions) so as to bring us back to the canonical case. We record the results for the general case at the end of this section.

We begin by obtaining a bound for the optimal value of a function g that lies in a certain neighborhood of f.

THEOREM 3.1. *Let X be a normed linear space and $f : X \to \overline{\mathbb{R}}$ be a proper function such that* $\min f = f(0) = 0$*. Given $\rho > 0$, for all functions $g : X \to \overline{\mathbb{R}}$ such that*

$$\operatorname{argmin} g \cap \rho\mathbb{B} \neq \emptyset, \qquad |\inf g| < \rho,$$

we have

$$|\inf g - \min f| = |\inf g| \leq \widehat{d\!l}_\rho(f, g).$$

More generally, given $\rho > 0$, for all g such that $|\inf_{\rho\mathbf{B}} g| < \rho$,

$$|\inf_{\rho\mathbf{B}} g - \inf f| = |\inf_{\rho\mathbf{B}} g| < \widehat{d\!l}_\rho(f, g).$$

Proof. If $\operatorname{argmin} g \cap \rho\mathbb{B} \neq \emptyset$ and $|\inf g| < \rho$, then $|\inf_{\rho\mathbf{B}} g| < \rho$. It thus suffices to consider the general case.

From the definition of $\widehat{d\!l}_\rho$ and $(0,0) \in \operatorname{epi} f$, we have $d((0,0), \operatorname{epi} g) \leq \widehat{d\!l}_\rho(f,g)$. On the other hand, since $|\inf_{\rho\mathbf{B}} g| < \rho$, we have $d((0,0), \operatorname{epi} g) \leq \rho$. Thus,

$$d((0,0), \operatorname{epi} g) = d((0,0), \operatorname{epi} g \cap \rho\mathbb{B}) \leq \widehat{d\!l}_\rho(f, g),$$

and for all $\varepsilon > 0$, there exists $(u_\varepsilon, \alpha_\varepsilon) \in \operatorname{epi} g$ such that $\|u_\varepsilon\| \leq \widehat{d\!l}_\rho(f,g) + \varepsilon$, $|\alpha_\varepsilon| \leq \widehat{d\!l}_\rho(f,g) + \varepsilon$, and $\max[\|u_\varepsilon\|, |\alpha_\varepsilon|] \leq \rho$. Since $\inf_{\rho\mathbf{B}} g \leq g(u_\varepsilon) \leq \alpha_\varepsilon \leq \widehat{d\!l}_\rho(f,g) + \varepsilon$ holds for all $\varepsilon > 0$: $\inf_{\rho\mathbf{B}} g \leq \widehat{d\!l}_\rho(f,g)$.

Let us now prove that $\inf_{\rho\mathbf{B}} g \geq -\widehat{d\!l}_\rho(f,g)$ also. For $\varepsilon > 0$, let $x_\varepsilon \in \rho\mathbb{B}$ be such that $g(x_\varepsilon) \leq \inf_{\rho\mathbf{B}} g + \varepsilon$. Choosing $\varepsilon \in (0, \rho - \inf_{\rho\mathbf{B}} g)$, we have that $g(x_\varepsilon) \leq \rho$, using here the assumption $|\inf_{\rho\mathbf{B}} g| < \rho$. On the other hand, $g(x_\varepsilon) \geq \inf_{\rho\mathbf{B}} g > -\rho$ and hence $(x_\varepsilon, g(x_\varepsilon)) \in \operatorname{epi} g \cap \rho\mathbb{B}$. By definition of $\widehat{d\!l}_\rho$, there exists $(v_\varepsilon, \beta_\varepsilon) \in \operatorname{epi} f$ such that

$$\|x_\varepsilon - v_\varepsilon\| \leq \widehat{d\!l}_\rho(f,g) + \varepsilon, \qquad |g(x_\varepsilon) - \beta_\varepsilon| \leq \widehat{d\!l}_\rho(f,g) + \varepsilon.$$

From this it follows that

$$g(x_\varepsilon) \geq \beta_\varepsilon - \widehat{d\!l}_\rho(f,g) - \varepsilon \geq f(v_\varepsilon) - \widehat{d\!l}_\rho(f,g) - \varepsilon \geq -\widehat{d\!l}_\rho(f,g) - \varepsilon,$$

since $f \geq 0$. It now remains to combine this inequality with $g(x_\varepsilon) \leq \inf_{\rho\mathbf{B}} g + \varepsilon$ and let $\varepsilon \downarrow 0$ to conclude that $\inf_{\rho\mathbf{B}} g \geq -\widehat{d\!l}_\rho(f,g)$. □

The preceding proof has a simple geometric interpretation. The assumption that $|\inf_{\rho\mathbb{B}} g| < \rho$ tells us that we need only be concerned with what happens in $\rho\mathbb{B} \times [-\rho, \rho]$. The argument relies on the fact that the projection $\mathrm{prj}_{\mathbb{R}} : (x, \alpha) \to \alpha$ is a contraction with respect to $\hat{d\!l}_\rho$, i.e., for any sets $A_1, A_2 \subset \rho\mathbb{B} \times [-\rho, \rho]$, $\hat{d\!l}_\rho(\mathrm{prj}_{\mathbb{R}} A_1, \mathrm{prj}_{\mathbb{R}} A_2) \leq \hat{d\!l}_\rho(A_1, A_2)$.

Remark 3.2. When g is proper and bounded below, $\inf g$ is finite, and for ρ sufficiently large, $|\inf_{\rho\mathbb{B}} g| < \rho$. Indeed, $\inf_{\rho\mathbb{B}} g$ is monotonically decreasing as $\rho \to \infty$. Since for all x, $g(x) \geq \lim_{\rho\to\infty} \inf_{\rho\mathbb{B}} g \geq \inf g$ (for ρ large enough $x \in \rho\mathbb{B}$), we obtain $\lim_{\rho\to\infty} \inf_{\rho\mathbb{B}} g = \inf g$. Thus the condition $|\inf_{\rho\mathbb{B}} g| < \rho$ is not restrictive; it is a "minimal" assumption that will allow us to estimate $|\inf f - \inf_{\rho\mathbb{B}} g|$ in terms of $\hat{d\!l}_\rho$.

COROLLARY 3.3 (from local to global minimization). *Let X be a normed linear space, $f, g : X \to \overline{\mathbb{R}}$ proper with $\min f = f(0) = 0$ and g bounded below. Given $\varepsilon > 0$, let ρ_ε be such that*

$$\inf g \leq \inf\{\, g(x) \mid x \in \rho_\varepsilon \mathbb{B} \,\} < \inf g + \varepsilon.$$

Then

$$|\inf g - \min f| = |\inf g| \leq \hat{d\!l}_{\gamma_\varepsilon}(f, g) + \varepsilon, \quad \textit{where } \gamma_\varepsilon := \max\,[\rho_\varepsilon, |\inf g| + \varepsilon].$$

Proof. The assumptions yield the following string of inequalities:

$$-\gamma_\varepsilon \leq -(|\inf g| + \varepsilon) < \inf g \leq \inf\nolimits_{\gamma_\varepsilon \mathbb{B}} g \leq \inf\nolimits_{\rho_\varepsilon \mathbb{B}} g < \inf g + \varepsilon \leq \gamma_\varepsilon.$$

Hence $|\inf_{\gamma_\varepsilon \mathbb{B}} g| < \gamma_\varepsilon$. The theorem implies that $|\inf_{\gamma_\varepsilon \mathbb{B}} g| \leq \hat{d\!l}_{\gamma_\varepsilon}(f, g)$, and consequently,

$$-\hat{d\!l}_{\gamma_\varepsilon}(f, g) - \varepsilon \leq \inf\nolimits_{\gamma_\varepsilon \mathbb{B}} g - \varepsilon \leq \inf\nolimits_{\rho_\varepsilon \mathbb{B}} g - \varepsilon \leq \inf g \leq \inf\nolimits_{\gamma_\varepsilon \mathbb{B}} g \leq \hat{d\!l}_{\gamma_\varepsilon}(f, g),$$

i.e., $|\inf g| \leq \hat{d\!l}_{\gamma_\varepsilon}(f, g) + \varepsilon$. □

Note that in order to satisfy the condition $\inf g < \inf\{\, g(x) \mid x \in \rho_\varepsilon \mathbb{B} \,\} \leq \inf g + \varepsilon$ in Corollary 3.3, it may be necessary to let $\rho_\varepsilon \uparrow \infty$ when $\varepsilon \downarrow 0$, for example, if g is not coercive.

Remark 3.4. We can replace the condition $|\inf_{\rho\mathbb{B}} g| < \rho$ in Theorem 3.1 by "$\inf_{\rho\mathbb{B}} g > -\rho$ and $\hat{d\!l}_\rho(f, g) < \rho$" without affecting the conclusion. In fact, we can show that these assumptions are very nearly the same. More precisely, under the same hypotheses as in Theorem 3.1,

$$(\inf\nolimits_{\rho\mathbb{B}} g > -\rho,\ \hat{d\!l}_\rho(f, g) < \rho) \Longrightarrow |\inf\nolimits_{\rho\mathbb{B}} g| < \rho \Longrightarrow (\inf\nolimits_{\rho\mathbb{B}} g > -\rho,\ \hat{d\!l}_\rho(f, g) \leq 2\rho).$$

To obtain the first implication, it suffices to show that $\inf_{\rho\mathbb{B}} g < \rho$. The definitions of $\hat{d\!l}_\rho$ and $(0, 0) \in \operatorname{epi} f \cap \rho\mathbb{B}$ yield $d((0, 0), \operatorname{epi} g) < \rho$. Thus there exists $(u, \alpha) \in \operatorname{epi} g$ such that $\|u\| < \rho$ and $|\alpha| < \rho$, i.e., $u \in \rho\mathbb{B}$ and $\rho > \alpha \geq g(u) \geq \inf_{\rho\mathbb{B}} g$.

To obtain the second implication, we need only show that $\hat{d\!l}_\rho(f, g) \leq 2\rho$. Let $(v, \beta) \in \operatorname{epi} f \cap \rho\mathbb{B}$. Since $|\inf_{\rho\mathbb{B}} g| < \rho$ there exists $u \in \rho\mathbb{B}$ such that $-\rho < g(u) < \rho$, i.e., $(u, \rho) \in \operatorname{epi} g$ and

$$d((v, \beta), \operatorname{epi} g) \leq \max\,[\|v - u\|, |\beta - \rho|] \leq 2\rho.$$

This shows that $e(\operatorname{epi} f \cap \rho\mathbb{B}, \operatorname{epi} g) \leq 2\rho$. To see that $e(\operatorname{epi} g \cap \rho\mathbb{B}, \operatorname{epi} f) \leq 2\rho$, let $(u, \alpha) \in \operatorname{epi} g \cap \rho\mathbb{B}$ and observe that $d((u, \alpha), \operatorname{epi} f) \leq d((u, \alpha), (0, 0)) \leq \rho \leq 2\rho$.

To be able to consider any possible perturbation of f and still obtain an estimate of the distance between the minimizers of f and g, we must know something about the geometric shape of f in a neighborhood of a minimizer. We need to control the "curvature" of f. We are going to assume that this can be achieved radially, i.e., that we know of a function $\psi : \mathbb{R}_+ \to \overline{\mathbb{R}}_+$ with $\psi(0) = 0$ and of a neighborhood of $\bar{x} \in \operatorname{argmin} f$, say $\mathbb{B}(\bar{x}, \rho)$ with $\rho > 0$, such that

$$f(x') - f(\bar{x}) \geq \psi(\|\bar{x} - x'\|) \quad \forall x' \in \mathbb{B}(\bar{x}, \rho).$$

We refer to such a function ψ as a *conditioning function* and call $\bar{x}$ a *ψ-minimizer* of f. The terminology has been chosen to suggest a certain relationship with the notion of conditioning in numerical linear algebra; the connection will come to light after the discussion in §5. In the canonical case, it means that $f \geq \psi(\|\cdot\|)$ on some neighborhood of zero. Of course, $\psi \equiv 0$ is then such a function, but, as we shall see, no information can be gained from such a universal lower bound for $f - f(\bar{x})$. The best estimates will be obtained by choosing ψ as large as possible; a problem that admits a conditioning function that is strictly increasing could be called *well conditioned*. Figures 1 and 2 illustrate two typical situations.

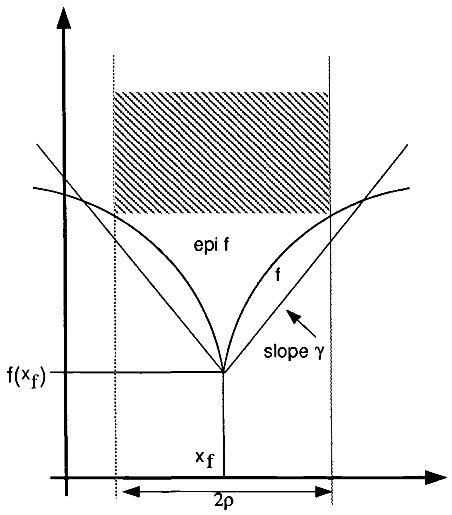


FIG. 1. $\psi(r) = \gamma|r|$.

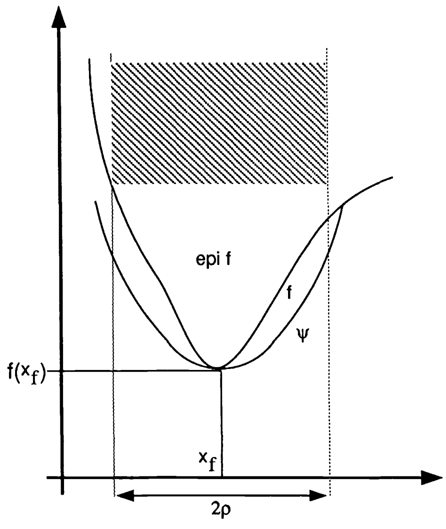


FIG. 2. $\psi(r) = cr^2$.

When $\psi(r) = \gamma r$ for some $\gamma > 0$, the function f is sharply pointed at $\bar{x}$ (see Fig. 1); this will lead to Lipschitz continuity properties for the optimal solutions. When $\psi(r) = \gamma r^2$, the function f may be smooth at $\bar{x}$ (see Fig. 2); this will only allow for Hölder-type continuity.

THEOREM 3.5. *Let X be a normed linear space and $f : X \to \overline{\mathbb{R}}$ a proper function such that* $\min f = 0 = f(0)$. *Let ψ be a conditioning function such that $\psi(\|\cdot\|) \leq f$ on $2\rho_f \mathbb{B}$ for some $\rho_f > 0$, and for $t \in \mathbb{R}_+$, let $\psi^{\square}(t) := \inf\{\psi(s) + |t - s| \mid s \in \mathbb{R}_+\}$. Given any $\rho \in (0, \rho_f]$, for all functions $g : X \to \overline{\mathbb{R}}$ such that*

$$\operatorname{argmin} g \cap \rho\mathbb{B} \neq \emptyset, \qquad |\inf g| < \rho,$$

we have

$$\psi^{\square}(\|\hat{x}\|) \leq 4\widehat{dl}_\rho(f, g) \quad \forall \hat{x} \in \operatorname{argmin} g \cap \rho\mathbb{B}.$$

More generally, given $\rho \in (0, \rho_f]$, for all $g : X \to \overline{\mathbb{R}}$ such that $|\inf_{\rho\mathbb{B}} g| < \rho$, we have

$$\psi^\square(\|\hat{x}\|) \leq 4\widehat{dl}_\rho(f, g) \quad \forall\, \hat{x} \in \operatorname{argmin}_{\rho\mathbb{B}} g.$$

Proof. We have already shown in the proof of Theorem 3.1 that $\operatorname{argmin} g \cap \rho\mathbb{B} \neq \emptyset$ and $|\inf g| < \rho$ imply $|\inf_{\rho\mathbb{B}} g| < \rho$, and that consequently it suffices to consider the weaker hypothesis, $|\inf_{\rho\mathbb{B}} g| < \rho$.

By assumption, $|g(\hat{x})| = |\inf_{\rho\mathbb{B}} g| < \rho$, and thus $(\hat{x}, g(\hat{x})) \in \operatorname{epi} g \cap \rho\mathbb{B}$. This means that $d((\hat{x}, g(\hat{x})), \operatorname{epi} f) \leq \widehat{dl}_\rho(f, g)$; cf. Fig. 3. From Theorem 3.1, we know that $|g(\hat{x})| = |\inf_{\rho\mathbb{B}} g| \leq \widehat{dl}_\rho(f, g)$. With the triangle inequality, this yields $d((\hat{x}, 0), \operatorname{epi} f) \leq 2\widehat{dl}_\rho(f, g)$. Let us now observe that $d((\hat{x}, 0), \operatorname{epi} f) \leq \|(\hat{x}, 0) - (0, 0)\| \leq \rho$, and that for any $(y, \gamma) \in X \times \mathbb{R}$ with $\|(y, \gamma)\| \geq 2\rho$,

$$\begin{aligned}
\|(\hat{x}, 0) - (y, \gamma)\| &= \max\,[\|\hat{x} - y\|, |\gamma|] \\
&\geq \max\,[\|y\| - \|\hat{x}\|, |\gamma|] \geq \max\,[\|y\| - \rho, |\gamma|] \\
&\geq \max\,[\|y\| - \rho, |\gamma| - \rho] \geq \max\,[\|y\|, |\gamma|] - \rho \geq \rho.
\end{aligned}$$

Hence, $d((\hat{x}, 0), \operatorname{epi} f) = d((\hat{x}, 0), \operatorname{epi} f \cap 2\rho\mathbb{B})$. Now, since $\psi(\|\cdot\|) \leq f$ on $2\rho\mathbb{B}$ (recall that $\rho \leq \rho_f$), $d((\hat{x}, 0), \operatorname{epi} f) \geq d((\hat{x}, 0), \operatorname{epi} \psi(\|\cdot\|))$, and in turn this implies

$$d((\hat{x}, 0), \operatorname{epi} \psi(\|\cdot\|)) \leq 2\,\widehat{dl}_\rho(f, g).$$

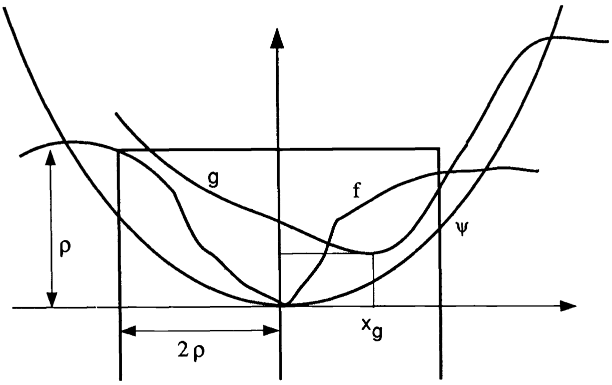


FIG. 3. *Functions f, g, and $\psi(\|\cdot\|)$.*

Next we calculate a lower bound for $d((\hat{x}, 0), \operatorname{epi} \psi(\|\cdot\|))$:

$$\begin{aligned}
d((\hat{x}, 0), \operatorname{epi} \psi(\|\cdot\|)) &= \inf_{x,\alpha}\{\max[\,\|x - \hat{x}\|, |\alpha|\,] \mid \alpha \geq \psi(\|x\|)\} \\
&= \inf_{x}\{\max[\,\|x - \hat{x}\|, \psi(\|x\|)\,]\} \\
&\geq \tfrac{1}{2} \inf_{x}\{\,\|x - \hat{x}\| + \psi(\|x\|)\} \\
&\geq \tfrac{1}{2} \inf_{x}\{\,|\,\|x\| - \|\hat{x}\|\,| + \psi(\|x\|)\} \\
&= \tfrac{1}{2}\psi^\square(\|\hat{x}\|).
\end{aligned}$$

Thus

$$\tfrac{1}{2}\psi^{\square}(\|\hat{x}\|) \le d((\hat{x},0),\operatorname{epi}\psi(\|\cdot\|)) \le 2\widehat{dl}_\rho(f,g),$$

which yields the asserted inequality. □

Remark 3.6. In certain situations, when some rough bound is available on the distance between $(0,0)$ and $(\hat{x},\inf g)$ with $\hat{x}\in\operatorname{argmin} g$, it may be possible to relax the condition $\psi(\|\cdot\|)\le f$ on $2\rho_f\mathbb{B}$, to one requiring that the inequality only be satisfied on $\rho_f\mathbb{B}$. More specifically, if $\max[\,\|\hat{x}\|,|\inf g|\,]\le \rho_f/3$ and $\widehat{dl}_{\rho_f}(f,g) =: \eta < 2\rho_f/3$ and there exists a conditioning function $\psi(\|\cdot\|)$ that minorizes f on $\rho\mathbb{B}$, we can reach the same conclusion as in Theorem 3.5. The argument is essentially the same as in the proof of Theorem 3.5, except that we need to show that $d((\hat{x},\inf g),\operatorname{epi} f) = d((\hat{x},\inf g),(\operatorname{epi} f)_\rho)$, where again $\inf g = g(\hat{x})$. To do this we proceed as follows. Since these conditions imply those of Theorem 3.1, we know that $(\hat{x},\inf g)\in(\operatorname{epi} g)_\rho$. For any $(x,\alpha)\notin\rho\mathbb{B}$, we have

$$\begin{aligned}\|(\hat{x},\inf g)-(x,\inf g)\| &= \max[\,\|\hat{x}-x\|,\ |\inf g-\alpha|\,]\\ &\ge \max[\,\|x\|-\rho/3,\ |\alpha|-\rho/3]\\ &\ge \|(x,\alpha)\|-\rho/3\ge 2\rho/3,\end{aligned}$$

where we have used the assumption that $\max[\,\|\hat{x}\|,|\alpha|\,]\le\rho/3$ to obtain the first inequality. The distance from $(\hat{x},\inf g)$ to any point of the epigraph of f outside the ρ-ball is greater than or equal to $2\rho/3$. By assumption, $\widehat{dl}_\rho(f,g) =: \eta < 2\rho/3$, hence $d((\hat{x},\inf g),\operatorname{epi} f)\le\eta<2\rho/3$, and $d((\hat{x},\inf g),\operatorname{epi} f) = d((\hat{x},\inf g),(\operatorname{epi} f)_\rho)$.

As already suggested by the examples in Figs. 1 and 2, the estimates for $\|\hat{x}\|$ depend on the properties of the conditioning function ψ we are able to come up with. We discuss this in further detail in §5, but there are a few observations that are in order at this point, in particular about the relationship between ψ and $\psi^{\square}$. Let us note that $\psi^{\square}$ is itself a conditioning function and that from the definition of $\psi^{\square}$, it follows that $\psi^{\square}\le\psi$. This means that, in general, the inequality $\psi^{\square}(\|\hat{x}\|)\le 4\widehat{dl}_\rho(f,g)$ yields a weaker bound for $\|\hat{x}\|$ than if one could assert $\psi(\|\hat{x}\|)\le 4\widehat{dl}_\rho(f,g)$. But, in most "practical" situations, it turns out that $\psi\equiv\psi^{\square}$, at least in a certain neighborhood (in $\mathbb{R}_+$) of 0. In fact, we are only interested in comparing ψ and $\psi^{\square}$ at the point $\|\hat{x}\|$. If ψ is finite valued, convex on $\mathbb{R}_+$, or at least convex on $[0,2\rho]$, and $\psi(t)>0$ for $t>0$, then ψ must be monotonically (strictly) increasing and directional derivatives exist at every point in $\mathbb{R}_+$. Let

$$\psi'_+ \text{ denote the right-hand side derivative of } \psi.$$

In such a situation, we have

$$\psi^{\square}(t)=\psi(t)\quad\text{whenever } \psi'_+(t)\le 1,$$

and thus $\psi^{\square}=\psi$ on $[0,t^\dagger]$ where $t^\dagger := \sup\{t\mid\psi'_+(t)\le 1\}$ [26]. Assuming that $\|\hat{x}\|\le t^\dagger$, Theorem 3.5 can also be interpreted as asserting that $\psi(\|\hat{x}\|)\le 4\widehat{dl}_\rho(f,g)$, or even that $\|\hat{x}\|\le\psi^{-1}(4\widehat{dl}_\rho(f,g))$, since ψ is then invertible. We summarize these observations in the following corollary.

COROLLARY 3.7. *If in addition to the assumptions in Theorem* 3.5, *the conditioning function ψ is finite valued, convex on* $[0,2\rho]$, $\psi>0$ *on* $(0,2\rho]$, *and* $\psi'_+(\|\hat{x}\|)\le 1$, *then*

$$\|\hat{x}\|\le\psi^{-1}(4\widehat{dl}_\rho(f,g)).$$

If $\psi(t) = \gamma t^p$ for $\gamma > 0$, $p > 1$, and $\rho \le \frac{1}{2}(p\gamma)^{1/1-p}$, then $\|\hat{x}\| \le ((4/\gamma)\hat{dl}_\rho(f,g))^{1/p}$. In particular, if $p = 2$,

$$\|\hat{x}\| \le \Big((4/\gamma)\hat{dl}_\rho(f,g)\Big)^{\frac{1}{2}}.$$

If $\psi(t) = \gamma t$, then

$$\|\hat{x}\| \le (4/\gamma')\hat{dl}_\rho(f,g) \quad \textit{with } \gamma' = \min[\gamma, 1].$$

Proof. The discussion preceding the corollary provides the proof when $\psi(t) = \gamma t^p$ and $p > 1$. When $\psi = \gamma|\cdot|$, $\psi^{\square} = \gamma'|\cdot|$ and we can apply the theorem directly. □

The remaining statements of this section are here for convenient reference. We rephrase the basic results for the case when the reference function f does not necessarily achieve its minimum at 0. We will rely on translates of f and g. Let f be such that $\operatorname{argmin} f =: x_f$ and let $\alpha_f = f(x_f)$, and define the *translation mapping* τ_f as follows:

$$\text{for a function } h : X \to \overline{\mathbb{R}}, \qquad (\tau_f h)(x) := h(x + x_f) - \alpha_f.$$

The function $(\tau_f f)$ then has a minimum at 0 and $(\tau_f f)(0) = 0$. This leads to the following reformulation of Theorems 3.1 and 3.5 and Corollary 3.7, which we now combine in one statement.

THEOREM 3.8. *Let X be a normed linear space, $f : X \to \overline{\mathbb{R}}$ a proper function that achieves its minimum only at x_f with $\alpha_f := \min f = f(x_f)$, and ψ a conditioning function such that for some $\rho_f > 0$,*

$$f(x) \ge \alpha_f + \psi(\|x - x_f\|) \quad \textit{whenever } \|x - x_f\| \le 2\rho_f.$$

Let $\psi^{\square}(t) := \inf\{\psi(s) + |t - s| \mid s \in \mathbb{R}_+\}$ and denote by τ_f the translation mapping $h \mapsto h(\cdot + x_f) - \alpha_f$. Given $\rho \in (0, \rho_f]$, for all functions $g : X \to \overline{\mathbb{R}}$ such that

$$\operatorname{argmin} g \cap \mathbb{B}(x_f, \rho) \ne \emptyset, \qquad |\inf g - \min f| < \rho,$$

we have

$$|\min g - \min f| \le \hat{dl}_\rho(\tau_f f, \tau_f g),$$
$$\psi^{\square}(\|\hat{x} - x_f\|) \le 4\hat{dl}_\rho(\tau_f f, \tau_f g) \quad \forall\, \hat{x} \in \operatorname{argmin} g \cap \mathbb{B}(x_f, \rho).$$

More generally, given $\rho \in (0, \rho_f]$, for all $g : X \to \overline{\mathbb{R}}$ such that $|\inf_{\mathbb{B}(x_f,\rho)} g - \inf f| < \rho$, we have

$$|\inf\nolimits_{\mathbb{B}(x_f,\rho)} g - \min f| \le \hat{dl}_\rho(\tau_f f, \tau_f g),$$
$$\psi^{\square}(\|\hat{x} - x_f\|) \le 4\hat{dl}_\rho(\tau_f f, \tau_f g) \quad \forall\, \hat{x} \in \operatorname{argmin}_{\mathbb{B}(x_f,\rho)} g.$$

Moreover, if ψ is convex, $0 < \psi(t) < \infty$ for $t \in (0,\infty)$, and $\psi'_+(\|\hat{x} - x_f\|) \le 1$, the preceding inequality can also be written as

$$\|\hat{x} - x_f\| \le \psi^{-1}(4\hat{dl}_\rho(\tau_f f, \tau_f g)) \quad \forall\, \hat{x} \in \operatorname{argmin}_{\mathbb{B}(x_f,\rho)} g.$$

Remark 3.9. The best bounds, of course, are achieved by choosing ψ as large as possible. We shall come back to this in §5. But, in particular, this applies to the case when we know of conditioning functions ψ_f and ψ_g, with x_f the unique ψ_f-minimizer of f and x_g the unique ψ_g-minimizer of g. Let us also assume that the conditioning functions ψ_f, ψ_g are finite valued, convex, and positive on $(0, \infty)$, and that

$$\max[\,(\psi_f)'_+(\|x_f - x_g\|),\ (\psi_g)'_+(\|x_f - x_g\|)\,] \leq 1.$$

From the theorem we obtain

$$\|x_f - x_g\| \leq \min[\psi_f^{-1}(4\widehat{dl}_\rho(\tau_f f, \tau_f g)),\ \psi_g^{-1}(4\widehat{dl}_\rho(\tau_g f, \tau_g g))\,],$$

which shows that all other quantities being equal, the better bound is obtained with the larger of the two conditioning functions.

Remark 3.10. Let us stress the fact that to apply Theorem 3.8, we only need insist that one of the two functions f or g have a unique minimum, say f. When g is a perturbation of f, it could very well happen that $\operatorname{argmin}_{\mathbb{B}(x_f,\rho)} g$ is not a singleton. Theorem 3.8 tells us that, under the appropriate conditions,

$$\operatorname{argmin}_{\mathbb{B}(x_f,\rho)} g \subset \mathbb{B}(x_f, \psi^{-1}(\widehat{dl}_\rho(\tau_f f, \tau_f g))).$$

Finally, Theorem 3.8 can be extended to a statement about global optimization.

COROLLARY 3.11. *Let X be a normed linear space and $f : X \to \overline{\mathbb{R}}$ a proper function. Suppose that f achieves its unique minimum at x_f with $\alpha_f := f(x_f)$, and that there exists a conditioning function ψ such that*

$$f(x) \geq f(x_f) + \psi(\|x - x_f\|) \quad \textit{whenever } \|x - x_f\| \leq 2\rho$$

for some $\rho > 0$. Let τ_f be the translation mapping: $h \mapsto h(\cdot + x_f) - \alpha_f$. Let $g : X \to \overline{\mathbb{R}}$ be a proper function that achieves its global minimum at $\hat{x}$ such that

$$\|\hat{x} - x_f\| \leq \rho, \qquad |f(x_f) - g(\hat{x})| \leq \rho;$$

then

$$|\inf f - \inf g| \leq \widehat{dl}_\rho(\tau_f f, \tau_f g) \quad \textit{and} \quad \psi^{\square}(\|x_f - \hat{x}\|) \leq 4\widehat{dl}_\rho(\tau_f f, \tau_f g),$$

where for $t \in \mathbb{R}_+$, $\psi^{\square}(t) := \inf\{\psi(s) + |t - s| \mid s \in \mathbb{R}_+\}$.

Proof. Simply note that since $\hat{x} \in \mathbb{B}(x_f, \rho)$, $g(\hat{x}) = \inf g \leq \inf_{\mathbb{B}(x_f,\rho)} g \leq g(\hat{x})$. This, with $|\inf f - \inf g| \leq \rho$, guarantees that the assumptions of Theorem 3.8 are satisfied. □

4. Examples. The most important potential applications of the results of §3 lie in their use for obtaining error estimates when dealing with approximation schemes for infinite-dimensional problems (control problems, stochastic optimization problems, variational inequalities, etc.). They also provide the tools for asymptotic analysis, such as in the development of large deviations results for stochastic optimization problems [29], as well for the asymptotic study of nonclassical differential equations. It is not possible, in the framework of this study, to pursue such developments. We will illustrate the application of the results of §3 in two simple situations. The first example involves the projection of a point on a moving convex set; this problem arises in many guises in various applications, for example, in mechanics [33]. The second example confirms, at the theoretical level, what is already known experimentally about the slow convergence rate of penalization methods in nonlinear programming.

Example 4.1. Let X be a Hilbert space, C a nonempty closed convex subset of X, and x_0 an arbitrary point in X. The optimization problem

$$\text{minimize } \|x - x_0\| \quad \text{for } x \in C$$

has a unique solution $p_C(x_0)$ called the *projection* of x_0 on C. With

$$f(x) = \|x - x_0\| + \delta_C(x) = \begin{cases} \|x - x_0\| & \text{if } x \in C, \\ \infty & \text{otherwise,} \end{cases}$$

we are in the setting of §§2 and 3. It is well known that $x_0 \mapsto p_C(x_0)$ is a contraction. We are going to consider $p_C(x_0)$ as a function of C measuring the perturbations by means of the ρ-distance. We show that if C and D are two nonempty closed convex sets and

$$\rho \geq \max[\,1/6, d(x_0, C) + d(x_0, D)\,],$$

then

$$\|p_C(x_0) - p_D(x_0)\| \leq 5\rho^{\frac{1}{2}} \widehat{dl}_\rho(C - p_C(x_0), D - p_C(x_0))^{\frac{1}{2}}.$$

To obtain this inequality as a consequence of the basic stability results, we use the fact that

$$p_C(x_0) \in \operatorname{argmin}_x\{\,\|x - x_0\|^2 \,|\, x \in C\,\},$$

as follows from

$$(\|p_C(x_0) - x_0\| \leq \|x - x_0\|) \Longrightarrow (\|p_C(x_0) - x_0\|^2 \leq \|x - x_0\|^2).$$

We are going to apply the results of §3 with $f := \|\cdot - x_0\|^2 + \delta_C$, $g := \|\cdot - x_0\|^2 + \delta_D$, and $\psi(t) = t^2$ as the conditioning function. Note that we always have

$$f(x) \geq f(p_C(x_0)) + \|x - p_C(x_0)\|^2.$$

This is certainly the case if $x \notin C$, and if $x \in C$, then this follows from a basic trigonometric identity for triangles that yields

$$\|x - x_0\|^2 = \|p_C(x_0) - x_0\|^2 + \|p_C(x_0) - x\|^2 - 2\|p_C(x_0) - x_0\|(\|p_C(x_0) - x\|)\cos\beta,$$

where β is the angle between the line segments $[x_0, p_C(x_0)]$ and $[p_C(x_0), x]$. Because $p_C(x_0)$ is the projection of x_0 on C, $x \in C$, and C is convex, it follows that $\pi/2 \leq \beta \leq \pi$, and thus $\cos\beta \leq 0$, i.e., $\|x - x_0\|^2 \geq \|p_C(x_0) - x_0\|^2 + \|p_C(x_0) - x\|^2$. This tells us that $p_C(x_0)$ is a ψ-minimizer of f.

From Theorem 3.8 and Corollary 3.11, we have

$$\|p_C(x_0) - p_D(x_0)| \leq \left(4\widehat{dl}_\rho(\tau_f f, \tau_f g)\right)^{\frac{1}{2}}$$

with

$$\tau_f f(x) := \|x - (x_0 - p_C(x_0))\|^2 + \delta_{C - p_C(x_0)}(x) - \|x_0 - p_C(x_0)\|^2,$$
$$\tau_f g(x) := \|x - (x_0 - p_C(x_0))\|^2 + \delta_{D - p_C(x_0)}(x) - \|x_0 - p_C(x_0)\|^2.$$

It remains for us to show that $\widehat{dl}_\rho(\tau_f f, \tau_f g) \leq 6\rho\widehat{dl}_\rho(C - p_C(x_0), D - p_C(x_0))$.

Let $C_0 := C - p_C(x_0)$, $D_0 := D - p_C(x_0)$, and $\tilde{x}_0 := x_0 - p_C(x_0)$. First note that $\hat{d\!l}_\rho(C_0, D_0) \leq 2\rho$. To see this, pick $x \in (C_0)_\rho$ and let $\tilde{x} = p_{D_0}(x)$. Then

$$\begin{aligned}\|x - \tilde{x}\| \leq \|x - p_{D_0}(\tilde{x}_0)\| &\leq \|x - \tilde{x}_0\| + \|\tilde{x}_0 - p_{D_0}(\tilde{x}_0)\| \\ &\leq \|x\| + (\|\tilde{x}_0\| + d(x_0, D)) \leq 2\rho,\end{aligned}$$

where the first term in the penultimate expression is less than or equal to ρ because $x \in (C_0)_\rho$ and the second term is less than or equal to ρ by definition of ρ.

To show that $\hat{d\!l}_\rho(\tau_f f, \tau_f g) \leq 6\rho\hat{d\!l}_\rho(C_0, D_0)$, we use Theorem 2.2(b) (the Kenmochi conditions). We show that for every $x \in (C_0)_\rho$ with $\|x - \tilde{x}_0\|^2 - \|\tilde{x}_0\|^2 \leq \rho$, there exists $\tilde{x} \in D_0$ such that $\|x - \tilde{x}\| \leq 6\rho\hat{d\!l}_\rho(C_0, D_0)$ and $\|\tilde{x} - \tilde{x}_0\|^2 - \|\tilde{x}_0\|^2 \leq \|x - \tilde{x}_0\|^2 - \|\tilde{x}_0\|^2 + 6\rho\hat{d\!l}_\rho(C_0, D_0)$. Since $(C_0)_\rho \subset D_0 + \hat{d\!l}_\rho(C_0, D_0)\mathbb{B}$, there always exists $\tilde{x} \in D_0$ such that $\|\tilde{x} - x\| \leq \hat{d\!l}_\rho(C_0, D_0)$, and a fortiori $\leq 6\rho\hat{d\!l}_\rho(C_0, D_0)$ since $\rho \geq \frac{1}{6}$. For this $\tilde{x}$, we have that

$$\begin{aligned}\|\tilde{x} - \tilde{x}_0\|^2 - \|x - \tilde{x}_0\|^2 &= \|\tilde{x} - x\|^2 + 2\langle \tilde{x} - x, x - \tilde{x}_0\rangle \\ &\leq \|\tilde{x} - x\|^2 + 2\|\tilde{x} - x\| \cdot \|x - \tilde{x}_0\| \\ &\leq \|\tilde{x} - x\|(\|\tilde{x} - x\| + 2\|x\| + 2\|\tilde{x}_0\|) \\ &\leq \hat{d\!l}_\rho(C_0, D_0)(2\rho + 2\rho + 2\rho) \leq 6\rho\hat{d\!l}_\rho(C_0, D_0),\end{aligned}$$

where we have used the inequalities $\|\tilde{x} - x\| \leq \hat{d\!l}_\rho(C_0, D_0) \leq 2\rho$. We can repeat the same argument, interchanging the roles of C_0 and D_0, to show that for every $x \in (D_0)_\rho$ with $\|x - \tilde{x}_0\|^2 - \|\tilde{x}_0\|^2 \leq \rho$, there exists $\tilde{x} \in C_0$ such that $\|x - \tilde{x}\| \leq 6\rho\hat{d\!l}_\rho(C_0, D_0)$ and $\|\tilde{x} - \tilde{x}_0\|^2 - \|\tilde{x}_0\|^2 \leq \|x - \tilde{x}_0\|^2 - \|\tilde{x}_0\|^2 + 6\rho\hat{d\!l}_\rho(C_0, D_0)$.

We can also give a more direct derivation of this bound for $\|p_C(x_0) - p_D(x_0)\|$ that does not (explicitly) rely on the general results of §3. It is included here because it yields a better "constant." We assume now that $\rho \geq \|x_0\| + d(x_0, C) + d(x_0, D)$. From

$$\|p_C(x_0)\| \leq \|x_0\| + d(x_0, C) \leq \rho, \qquad \|p_D(x_0)\| \leq \|x_0\| + d(x_0, D) \leq \rho,$$

follows the existence of $\tilde{z} \in C$ and $\tilde{y} \in D$ such that

$$\|p_D(x_0) - \tilde{z}\| = d(p_D(x_0), C), \qquad \|p_C(x_0) - \tilde{y}\| = d(p_C(x_0), D),$$

with

$$\|p_C(x_0) - \tilde{y}\| \leq \hat{d\!l}_\rho(C, D), \qquad \|p_D(x_0) - \tilde{z}\| \leq \hat{d\!l}_\rho(C, D).$$

The classical optimality conditions for $p_C(x_0)$ and $p_D(x_0)$ tell us that

$$\langle x_0 - p_C(x_0), \tilde{z} - p_C(x_0)\rangle \leq 0, \qquad \langle x_0 - p_D(x_0), \tilde{y} - p_D(x_0)\rangle \leq 0.$$

Adding these two inequalities and regrouping terms, we have

$$\begin{aligned}\|p_C(x_0) - p_D(x_0)\|^2 &\leq \langle p_C(x_0) - x_0,\ p_D(x_0) - \tilde{z}\rangle + \langle p_D(x_0) - x_0,\ p_C(x_0) - \tilde{y}\rangle \\ &\leq \|p_C(x_0) - x_0\|\ \|p_D(x_0) - \tilde{z}\| + \|p_D(x_0) - x_0\|\ \|p_C(x_0) - \tilde{y}\| \\ &\leq \max[\,\|p_D(x_0) - \tilde{z}\|,\ \|p_C(x_0) - \tilde{y}\|\,] \cdot (d(x_0, C) + d(x_0, D)) \\ &\leq \rho\hat{d\!l}_\rho(C, D),\end{aligned}$$

where the last inequality follows from the definition of ρ. In §6, we return to this example and show that this latter bound for $\|p_C(x_0) - p_D(x_0)\|$ is actually attained and is the best possible.

Example 4.2. *Penalization.* Let $f_0 : X \to \mathbb{R}$ be a locally Lipschitz function to be minimized on a set $C \subset X$. We approximate the problem

$$\text{minimize } f(x) := f_0(x) + \delta_C(x),$$

where δ_C is the indicator function of C, by a problem of the type

$$\text{minimize } f_\theta(x) := f_0(x) + \varphi_\theta(x),$$

where $\{\, \varphi_\theta : X \to \mathbb{R}_+, \theta \geq 0 \,\}$ is a parametrized family of functions such that

(i) $\varphi_\theta = 0$ on C;

(ii) for some $p \geq 1$ and $\alpha > 0$: $\varphi_\theta \geq \alpha\theta[d(\cdot, C)]^p$.

We are going to let θ tend to ∞. Fix $\rho > 0$ and define

$$\lambda_\rho := \sup\{\, |f_0(x) - f_0(y)|/(\|x-y\|) \mid \|x\| \leq \rho, \|y\| \leq \rho, x \neq y \,\},$$
$$\mu_\rho := \sup\{\, |f_0(x)| \mid \|x\| \leq \rho \,\}.$$

We will show that for some γ_ρ (calculated below), we have

$$\hat{dl}_\rho(f, f_\theta) \leq \gamma_\rho \theta^{-1/p}.$$

Moreover, with $\bar{x} \in \operatorname{argmin} f$ and $x_\theta \in \operatorname{argmin} f_\theta$, and assuming that there exists a finite-valued conditioning function ψ such that

$$f_0(x') - f_0(\bar{x}) \geq \psi(\|x' - \bar{x}\|) \quad \forall\, x' \in C,$$

we shall also prove that for θ sufficiently large and ρ_1 as defined below,

$$\psi^{\square}(\|x_\theta - \bar{x}\|) \leq 4\gamma_{\rho_1}\theta^{-1/p} \qquad \text{for all } x_\theta \in \operatorname{argmin} f_\theta,$$

i.e., x_θ converges to $\bar{x}$ at an *exponential rate*. Since $f_\theta \leq f$, $e((\operatorname{epi} f)_\rho, \operatorname{epi} f_\theta) = 0$ with e the excess as defined in §2. The Kenmochi conditions (Theorem 2.2) will provide us with an upper bound for $e((\operatorname{epi} f_\theta)_\rho, \operatorname{epi} f)$. We start with a point $\|\tilde{x}\| \leq \rho$ such that $|f_\theta(\tilde{x})| \leq \rho$. By the definition of f_θ and φ_θ, it follows that

$$f_0(\tilde{x}) + \alpha\theta d(\tilde{x}, C)^p \leq \rho.$$

Because $|f_0| \leq \mu_\rho$ on $\rho\mathbb{B}$, we have

$$d(\tilde{x}, C) \leq [(\alpha\theta)^{-1}(\rho + \mu_\rho)]^{1/p},$$

and for every $0 < \varepsilon < 1$, there exists $\tilde{x}_\varepsilon \in C$ such that

$$\|\tilde{x} - \tilde{x}_\varepsilon\| \leq \theta^{-1/p}[\alpha^{-1}(\rho + \mu_\rho)]^{1/p} + \varepsilon.$$

The upper bound for $f(\tilde{x}_\varepsilon)$ is obtained directly from the preceding inequality and the local Lipschitz property of f_0. Since $\tilde{x}_\varepsilon \in C$,

$$f(\tilde{x}_\varepsilon) = f_0(\tilde{x}_\varepsilon) \leq f_0(\tilde{x}) + \lambda_{\rho_1}\|\tilde{x} - \tilde{x}_\varepsilon\|,$$

where $\rho_1 = \rho + [(\alpha\theta)^{-1}(\rho + \mu_\rho)]^{1/p} + 1$. If we now take into account that $f_0 \le f_\theta$ and that $\varphi_\theta \ge 0$, we obtain

$$f(\tilde{x}_\varepsilon) \le f_\theta(\tilde{x}) + \theta^{-1/p}\lambda_{\rho_1}[\alpha^{-1}(\rho + \mu_\rho)]^{1/p} + \varepsilon\lambda_{\rho_1}.$$

This, and the bound on $\|\tilde{x} - \tilde{x}_\varepsilon\|$, show that the conditions in Theorem 2.2(b) are satisfied, and thus $\hat{d\!l}_\rho(f, f_\theta) \le \gamma_\rho\theta^{-1/p}$ with

$$\gamma_\rho := [\alpha^{-1}(\rho + \mu_\rho)]^{1/p}(\max[1, \lambda_{\rho_1}]).$$

The inequality $\psi^{\text{¤}}(\|x_\theta - \bar{x}\|) \le 4\gamma_{\rho_1}\theta^{-1/p}$ for some conditioning function ψ is obtained from the preceding result via Theorem 3.8; to apply Theorem 3.8 recall that we need to make the translation from f_0 to $f_0(\cdot + \bar{x}) - f_0(\bar{x})$, and thus λ_ρ needs to be replaced by $\lambda_{\rho+\|\bar{x}\|}$, and μ_ρ by $\mu_{\rho+\|\bar{x}\|+|f_0(\bar{x})|}$.

5. Conditioning functions. We introduced conditioning functions $\psi : \mathbb{R}_+ \to \overline{\mathbb{R}}_+$ (with $\psi(0) = 0$) in §3 to capture the (radial) shape of a function f in the neighborhood of a minimizer $\bar{x}$ and we defined, in terms of ψ, the generally stronger notion of a *ψ-minimizer*: $\bar{x} \in \operatorname{argmin} f$ and for some $\rho > 0$,

$$f(x') - f(\bar{x}) \ge \psi(\|\bar{x} - x'\|) \quad \forall x' \in \mathbb{B}(\bar{x}, \rho). \tag{5.1}$$

The basic inequalities in Theorems 3.5 and 3.8 rely on the conditioning function ψ, more precisely on $\psi^{\text{¤}}$, to obtain an upper bound on the distance between $\bar{x}$ and a minimizer $\hat{x}$ of a function g (in a certain neighborhood of f). Clearly, the best bounds will be obtained with the "largest" possible conditioning function for which the preceding relation holds.

PROPOSITION 5.1. *Suppose $(X, \|\cdot\|)$ is a normed linear space, $f : X \to \overline{\mathbb{R}}$, and $\bar{x} \in \operatorname{argmin} f$. Then, given $\rho > 0$, there exists a largest conditioning function $\psi_{f,\bar{x}}$ such that*

$$f(x') - f(\bar{x}) \ge \psi_{f,\bar{x}}(\|\bar{x} - x'\|) \quad \forall x' \in \mathbb{B}(\bar{x}, \rho).$$

In fact,

$$\psi_{f,\bar{x}}(\theta) = \begin{cases} \inf\{f(y) - f(x) \mid \|y - x\| = \theta\} & \textit{if } \theta \in [0, \rho], \\ \infty & \textit{for } \theta > \rho. \end{cases}$$

Proof. Simply notice that if $(\psi_i)_{i\in I}$ is the family of conditioning functions for which (5.1) holds, then $\sup_{i\in I}\psi_i$ is still a conditioning functions and thus $\psi_{f,\bar{x}} = \sup_{i\in I}\psi_i$. To see that the expression for $\psi_{f,\bar{x}}$ is valid, one verifies that the expression defines a conditioning function and that any function that is larger on $[0, \rho]$ will fail to satisfy (5.1) at some point $x' \in \mathbb{B}(\bar{x}, \rho)$. □

The function $\psi_{f,\bar{x}}$ is called the *radial regularization* of f at $\bar{x}$ because the function $y \mapsto \psi_{f,\bar{x}}(\|y\|)$ is the largest radial function that minorizes $y \mapsto f(x + y) - f(x)$ on $\mathbb{B}(\bar{x}, \rho)$; $\psi_{f,\bar{x}}$ plays an important role in the theory of Orlicz spaces; cf. Fougères [22]. In general, $\psi_{f,\bar{x}}$ is not an increasing function, but this is always the case if f is convex and $\bar{x}$ is unique. When f admits a unique minimizer, instead of $\psi_{f,\bar{x}}$, we simply write ψ_f for the radial regularization of f at $\bar{x}$.

PROPOSITION 5.2. *Suppose $(X, \|\cdot\|)$ is a normed linear space, $f : X \to \overline{\mathbb{R}}$ is a convex function, and $x_f = \operatorname{argmin} f$. For given $\rho > 0$, the radial regularization ψ_f of f at x_f is strictly increasing on $[0, \rho]$; in fact, $\theta \mapsto \theta^{-1}\psi_f(\theta)$ is an increasing function.*

Proof. The proof is based on an elementary property of convex functions. Let $0 \le \theta_1 \le \theta_2 \le \rho$, and y_2 be a point in X such that $\|y_2 - x\| = \theta_2$. Set $y_1 := (1 - \theta_1/\theta_2)x +$

$(\theta_1/\theta_2)y_2$. Then $\|y_1 - x\| = (\theta_1/\theta_2)\|y_2 - x\| = \theta_1$. Hence, $\psi_f(\theta_1) \le f(y_1) - f(x)$, which by convexity of f yields

$$\psi_f(\theta_1) \le \left(1 - \frac{\theta_1}{\theta_2}\right) f(x) + \frac{\theta_1}{\theta_2} f(y_2) - f(x) \le \frac{\theta_1}{\theta_2}(f(y_2) - f(x)).$$

This inequality holds for any y_2 such that $\|y_2 - x\| = \theta_2$; it follows that

$$\psi_f(\theta_1) \le \frac{\theta_1}{\theta_2}\psi_f(\theta_2).$$

Thus, $\theta \mapsto \theta^{-1}\psi_f(\theta)$ is an increasing function on $[0, \rho]$. We observe that $\psi_f(\theta) > 0$ when $\theta \neq 0$ (since x_f is a unique minimizer), from which it follows that $\theta \mapsto \psi_f(\theta)$ is strictly increasing on $[0, \rho]$. □

Remark 5.3. Even if f is convex and x_f is the unique minimizer of f, ψ_f may very well fail to be convex. If it is important to work with a convex conditioning function, one could choose for ψ the convex closure of ψ_f, i.e., the function whose epigraph is the convex hull of epi ψ_f.

The notion of a *forcing function*, which we encounter in the study of well-posedness as well as in the design of numerical procedures for nonlinear problems, is closely related to that of a conditioning function. A function $\varphi : \mathbb{R} \to \overline{\mathbb{R}}_+$ is a forcing function if $\varphi(0) = 0$ and $\varphi(\theta^\nu) \to 0$ implies $\theta^\nu \to 0$. Of course, a forcing function is a conditioning function. Moreover, if φ is a forcing function and x_f is a φ-minimizer of f, then x_f is the *unique* minimizer of f on $\mathbb{B}(x_f, \rho)$ for some $\rho > 0$ and every minimizing sequence that lies in $\mathbb{B}(x_f, \rho)$ converges strongly to x_f. This means that the minimization problem is *well posed* à la Tykhonov [48]; Zolezzi [52, Cor. 1] noted that a minimization problem is well posed if and only if there exists a forcing function φ so that x_f is a φ-minimizer of f. Although the basic results have been stated in terms of conditioning functions, the practical use of these results will almost always depend on being able to come up with a conditioning function that is in fact a forcing function.

When faced with a particular optimization problem with possibly a large number of variables, the construction of an appropriate conditioning function could be quite involved, the main reason being that the point x_f that actually minimizes f is a priori unknown. It is thus important to exploit the global properties of f that will guarantee the existence of a useful conditioning function. This brings us to examine the notion of *uniform convexity*. An abundant literature has been devoted to this subject; see, e.g., Zalinescu [51]; Vladimirov, Nestorov, and Chekanov [49]; Sonntag [47]; and Dontchev [18] for the role played by uniform convexity in the analysis of stability questions in optimization and optimal control; for a recent survey, including new results, consult Azé [12]. For simplicity's sake, we are going to limit our observations to the case when X is a Hilbert space.

For $\gamma \ge 0$, a function $f : X \to \overline{\mathbb{R}}$ is *γ-uniformly convex* if for all $x^0, x^1 \in \operatorname{dom} f$ and all $\lambda \in [0, 1]$,

$$f((1-\lambda)x^0 + \lambda x^1) \le (1-\lambda)f(x^0) + \lambda f(x^1) - \gamma\lambda(1-\lambda)\|x^0 - x^1\|^2.$$

PROPOSITION 5.4. *Let $\gamma > 0$, and $f : X \to \overline{\mathbb{R}}$ be a proper, lsc, γ-uniformly convex function, with X a Hilbert space. Then f reaches its minimum at a unique point x_f that satisfies*

$$f(x) \ge f(x_f) + \gamma\|x - x_f\|^2 \quad \forall x \in X;$$

i.e., x_f is a γ-$\|\cdot\|^2$-minimizer of f.

Proof. Let $\partial f(x)$ denote the set of subgradients of f at x. When f is γ-uniformly convex, we have that for all $x, y \in \operatorname{dom} f$ and $v \in \partial f(x)$:

$$f(y) \geq f(x) + \langle v, y - x \rangle + \gamma \|x - y\|^2;$$

see [12] for details. From this the assertion follows directly since $x_f \in \operatorname{argmin} f$ implies $0 \in \partial f(x_f)$. □

DEFINITION 5.5 [40]. For $f : X \to \overline{\mathbb{R}}$, $\bar{x}$ is a strong local minimizer of f if there exists $\gamma > 0$ and a neighborhood V of $\bar{x}$ such that for all $y \in V$, $f(y) \geq f(\bar{x}) + \gamma\|\bar{x} - x\|^2$.

A strong local minimizer is a ψ-minimizer of f with $\psi(r) = \gamma r^2$ for $\gamma > 0$. When analyzing sufficient conditions for optimality in terms of second-order derivatives, we are naturally led to the notion of a strong minimizer [41]. Recent results of Rockafellar [41], [43] allow us to characterize the largest value of γ in terms of a lower bound for the second-order derivatives for a quite large class of functions. A lsc function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is epi-differentiable at x if the functions $h \mapsto (1/t)[f(x+th) - f(x)]$ epi-converge as $t \downarrow 0$ [40]. This epi-limit is the epi-derivative of f at x and is denoted by f'_x (with $f'_x(0) > -\infty$). A vector v is an *epi-subgradient* at x if $f'_x(y) \geq \langle v, y \rangle$ for all $y \in \mathbb{R}^n$. A function f is *twice epi-differentiable at x relative to v* if it is epi-differentiable at x and the functions

$$\psi_{x,v;t}(u) = \frac{2}{t^2}[f(x+tu) - f(x) - t\langle u, v \rangle]$$

epi-converge as $t \downarrow 0$. The epi-limit is the *second-order epi-derivative* and is denoted by $f''_{x,v}$ (with $f''_{x,v}(0) > -\infty$). When f is epi-differentiable at x relative to every epi-subgradient v (at x), f is said to be *twice epi-differentiable* at x.

PROPOSITION 5.6 [44, Thm. 2.2]. *Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be lsc and $\bar{x}$ a point at which f is finite and twice epi-differentiable. Suppose zero is a pseudo-gradient of f at $\bar{x}$ and $f''_{\bar{x},0}(u) > 0$ for all $u \neq 0$. Then $\bar{x}$ is a strong local minimizer of f. Moreover, with $\gamma_0 := \min_{|u|=1} f''_{\bar{x},0}(u)$, we have*

$$f(y) \geq f(\bar{x}) + \tfrac{1}{2}\gamma_0\|y - \bar{x}\|^2 + o(\|y - \bar{x}\|^2).$$

Thus, for all $\gamma < \gamma_0/2$ there exists $\rho_\gamma > 0$ such that

$$f(y) \geq f(\bar{x}) + \gamma\|y - \bar{x}\|^2 \quad \forall y \in \mathbb{B}(\bar{x}, \rho_\gamma).$$

To conclude, let us also examine the relationship between uniform convexity and the conditioning number associated with a nonlinear optimization problem. For an introduction to nonlinear conditioning, cf. Lemaire-Misonne [32], for example.

Let X be a Hilbert space, $f : X \to \overline{\mathbb{R}}$, and $x_0 \in \operatorname{argmin}_{\mathbb{B}(x_0,\rho)} f$ for some $\rho > 0$. Let $\varepsilon > 0$ and assume that for all linear perturbations $f - \langle v, \cdot \rangle$ of f with $\|v\| \leq \varepsilon$ there exists x_v a unique local minimizer of $f - \langle v, \cdot \rangle$. The *conditioning number* associated with f at x_0 (relative to linear perturbations) is the (positive) number defined by

$$C_l(x_0; f) := \lim_{r \downarrow 0} \sup_{\|v\| \leq r} \frac{\|x_v - x_0\|}{\|v\|} = \limsup_{v \to 0} \frac{\|x_v - x_0\|}{\|v\|}.$$

If f is convex and f^* is the conjugate of f, the preceding assumptions are equivalent to: the mapping $\partial f^* : X \rightrightarrows X$ is single-valued for all $\|v\| \leq \varepsilon$, and

$$C_l(x_0; f) := \limsup_{v \to 0} \frac{\|\partial f^*(v) - \partial f^*(0)\|}{\|v\|}.$$

The following proposition is well known in nonlinear analysis; we sketch out a proof for the sake of completeness.

PROPOSITION 5.7. *Let X be a Hilbert space, $f : X \to \overline{\mathbb{R}}$. Suppose that f is a proper, lsc, γ-uniformly convex function for some $\gamma > 0$ and x_f the (unique) minimizer of f. Then $C_l(x_f; f) \le 1/2\gamma$.*

Proof. The optimality conditions tell us that $0 \in \partial f(x_f)$ and $v \in \partial f(x_v)$ for all $\|v\| \le \varepsilon$. And thus, from γ-uniform convexity, it follows that

$$\langle v, x_v - x_f \rangle \ge 2\gamma \|x_v - x_f\|^2.$$

Applying the Cauchy–Schwarz inequality yields $(\|x_v - x_f\|)/\|v\| \le 1/2\gamma$ for all $\|v\| \le \varepsilon$, from which the bound for $C_l(x_f; f)$ follows immediately. □

The upper bound for $C_l(x_f; f)$ is tight; simply consider the case when the Cauchy–Schwarz inequality turns out to be an equality. When ∂f is linear, note that the conditioning number does not depend on the point x_f, and the definition of the conditioning number is then consistent with that commonly used in numerical linear analysis.

6. Evaluating bounds. This last section is a grab bag of examples. They highlight various features of the basic results in §3. In particular, we shall again be concerned with Example 4.1—the projection of a point on a moving convex set—to underscore the fact that the upper bound for $\|\hat{x} - x_f\|$ ($\hat{x} \in \operatorname{argmin} g$) in Theorem 3.8 is a "best" possible bound.

Let us begin with a simple example where we utilize Theorem 3.5 to estimate $\widehat{dl}_\rho$.

Example 6.1. Let $X = \mathbb{R}$, $f(x) = x^2/2$, and

$$f_\theta^{(x)} = \begin{cases} \frac{1}{2}x^2 & \text{if } x \le -2\theta, \\ -\theta x & \text{if } -2\theta \le x \le 0, \\ \frac{1}{2}x^2 - \theta x & \text{if } 0 \le x \le 2\theta, \\ \frac{1}{2}x^2 - 2\theta^2 & \text{if } 2\theta \le x, \end{cases}$$

for θ a positive parameter that will be allowed to go to zero; see Fig. 4.

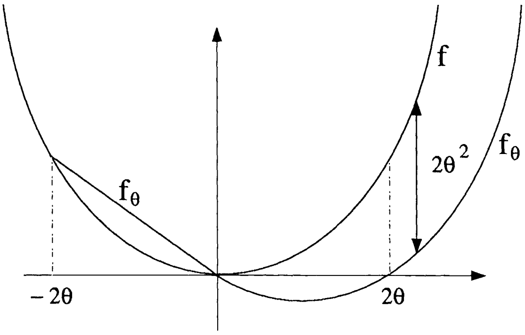


FIG. 4. *Estimating $\widehat{dl}_\rho(f, f_\theta)$.*

The functions f and f_θ are convex continuous functions that are piecewise C^∞. They achieve their minimum at $x_0 = 0$ and $x_\theta = \theta$ (with value $-\theta^2/2$). Thus $|\operatorname{argmin} f - \operatorname{argmin} f_\theta| = \theta$. We have that

$$\sup_{x \in \mathbb{R}} |f(x) - f_\theta(x)| = 2\theta^2.$$

From Corollary 3.7, it follows that

$$\theta^2 = \|x_0 - x_\theta\|^2 \le 8\widehat{dl}_\rho(f, f_\theta).$$

From the definition of $\widehat{dl}_\rho(f, f_\theta) \le \|f - f_\theta\|_\infty = 2\theta^2$. Thus $\widehat{dl}_\rho(f, f_\theta)$ is approximately $\kappa\theta^2$ with κ varying between $\frac{1}{8}$ and 2. □

This example also provides us with an illustration of the results of [7, §5] that relates the distance between subgradient mappings and the epi-distance between functions. Both f and f_θ are differentiable with

$$f_\theta'(x) = \begin{cases} x & \text{if } x \le -2\theta, \\ -\theta & \text{if } -2\theta \le x \le 0, \\ x - \theta & \text{if } 0 \le x \le 2\theta, \\ x & \text{if } 2\theta \le x, \end{cases}$$

and $f'(x) = x$. For ρ sufficiently large, $\widehat{dl}_\rho(\operatorname{gph} f', \operatorname{gph} f_\theta') = \theta$. Of course, this is in accordance with [7, Thm. 5.2], which asserts that when ρ is large enough, $\widehat{dl}_\rho(\operatorname{gph} \partial f, \operatorname{gph} \partial f_\theta) \le \kappa_\rho(\widehat{dl}_\rho(f, f_\theta))^{1/2}$ for some "constant" κ_ρ that depends on ρ.

The next example is also elementary but this time it involves nonsmooth convex functions (with nonsmoothness occurring precisely at the point at which f achieves its minimum).

Example 6.2. Let $X = \mathbb{R}$, θ a positive parameter (that will go to zero), and for some $p \in [1, \infty)$,

$$f_\theta(x) = \theta^{p-1}|x| + \frac{1}{p}|x|^p,$$
$$g_\theta(x) = \theta^{p-1}|x - \theta| + \frac{1}{p}|x|^p.$$

The functions f_θ and g_θ achieve their minimum only at $x_f(\theta) = 0$ and $x_g(\theta) = \theta$; see Fig. 5.

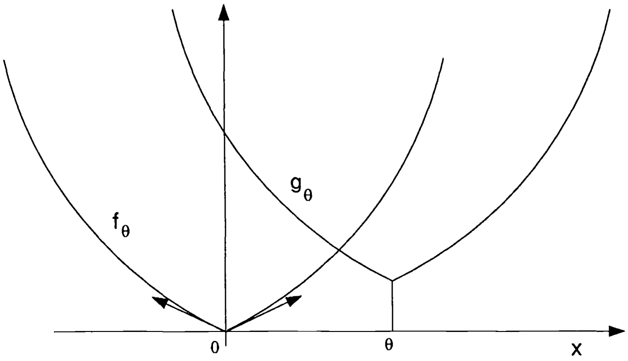


FIG. 5. *Calculating the ρ-Hausdorff distance.*

Hence $|x_f(\theta) - x_g(\theta)| = \theta$, while $d_\infty(f_\theta, g_\theta) := \sup_x |f_\theta(x) - g_\theta(x)| = \theta^p$. Thus, $|x_f(\theta) - x_g(\theta)| = d_\infty(f_\theta, g_\theta)^{1/p}$. From [7, §3], it follows that

$$\widehat{dl}_\rho(f_\theta, g_\theta) \le d_\infty(f, g) \le (1 + \theta^{p-1} + \rho^{p-1})\widehat{dl}_\rho(f_\theta, g_\theta)$$

for all $\theta \geq 0$ and $\rho \geq 0$. In terms of Theorem 3.5, we see that $x_f(\theta) = 0$ is a ψ-minimum of f_θ with $\psi(\eta) = (1/p)\eta^p$. And, indeed, the largest conditioning function ψ (independent of θ), such that for all $x, c, f_\theta(x) \geq f_\theta(0) + \psi(|x|)$, is $\psi = (1/p)|\cdot|^p$! □

The next example shows that the exponent $\frac{1}{2}$ obtained in Example 4.1 (the projection of a point on a moving set) is the best possible. We shall show later that the geometry of the space plays a determining role in this. This exponent $\frac{1}{2}$ comes up in a number of related, but more specialized, results: the sweeping problem (le problème de rafle) studied by Moreau [33], in the work on isometries for the Legendre–Fenchel transform by Attouch and Wets [5], in the work on approximation for the solutions of elliptic partial differential equations by Rabier and Thomas [35], in approximation and perturbation analysis of optimal control problems (Dontchev [18]), and when dealing with specific perturbations in nonlinear programming (Daniel [16] and Schultz [46]).

Example 6.3. Let $X = \mathbb{R}^2$ with the euclidean norm. Let $C_\theta = \overline{AE_\theta}$ and $D_\theta = \overline{AF_\theta}$, both depending on the angle (parameter) θ as in Fig. 6; $\overline{AF_\theta}$ is a chord, $\overline{AE_\theta}$ is a horizontal line segment, and E_θ lies on the same vertical axis as F_θ, both F_θ and E_θ converge to A as $\theta \to 0$.

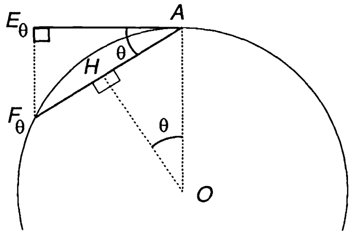


FIG. 6. *Projection on a convex set.*

The point $x_0 = (0,0)$ is projected on these convex sets: $p_{C_\theta}(x_0) = A$ and $p_{D_\theta}(x_0) = H_\theta$, i.e., $\|p_{C_\theta}(x_0) - p_{D_\theta}(x_0)\| = \sin\theta$, assuming that $\overline{0A}$ is of length 1. On the other hand, for ρ sufficiently large, $\hat{dl}_\rho(C_\theta, D_\theta) = d(E_\theta, F_\theta) = 2\sin^2\theta$, and thus $\|p_{C_\theta}(x_0) - p_{D_\theta}(x_0)\| = (1/\sqrt{2})\hat{dl}_\rho(C_\theta, D_\theta)^{1/2}$. Thus, Hölder continuity of order $\frac{1}{2}$, as obtained in Example 4.1, is the best possible.

To see how the exponent $\frac{1}{2}$ is related to the Hilbert structure of the space X, we make use of the fact that Clarkson's inequalities characterize the spaces of type p, like $(\mathbb{R}^n, \|\cdot\|_p)$ with $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$, or $l^p(\mathbb{N})$, $L^p(\Omega)$, $W^{m,p}(\Omega)$ (see [1, Thm. 2.28]):

$$\text{if } p \geq 2: \quad \left\|\frac{u+v}{2}\right\|_p^p \leq \tfrac{1}{2}\|u\|_p^p + \tfrac{1}{2}\|v\|_p^p - \frac{1}{2^p}\|v-u\|_p^p \quad \forall u, v \in X;$$

$$\text{if } p \leq 2: \quad \left\|\frac{u+v}{2}\right\|_p^{p'} \leq \tfrac{1}{2}\|u\|_p^{p'} + \tfrac{1}{2}\|v\|_p^{p'} - \frac{1}{2^p}\|v-u\|_p^{p'} \quad \forall u, v \in X,$$

where $p' = p/(p-1)$ is the conjugate exponent of p. The functions $f = \delta_C + \|\cdot - x_0\|_p^p$ when $p \geq 2$ and $f = \delta_C + \|\cdot - x_0\|_p^{p'}$ when $p \leq 2$ satisfy the same type of inequalities. Consequently, if X is a space of type p with $p \in (1, \infty)$, the functions f defined above are proper, lsc, convex functions that satisfy: for all $u_0, u_1 \in \operatorname{dom} f$ and all $\lambda \in (0,1)$,

$$f((1-\lambda)u_0 + \lambda u_1) \leq (1-\lambda)f(u_0) + \lambda f(u_1) - \lambda(1-\lambda)\psi(\|u_0 - u_1\|_p)$$

for $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ defined by

$$\psi(\eta) := \frac{1}{2^{p-2}}\eta^p \quad \text{if } p \geq 2 \quad \text{and} \quad \psi(\eta) := \frac{1}{2^{p'-2}}\eta^{p'} \quad \text{if } p < 2.$$

Note that $\psi(0) = 0$ and $\psi(\eta) > 0$ if $\eta > 0$. It is shown [12] then that for all $u_0 \in \operatorname{dom} f$ and $v_0 \in \partial f(u_0)$,

$$f(x) \geq f(u_0) + \langle v_0, x - u_0 \rangle + \psi(\|x - u_0\|_p) \quad \forall x \in X.$$

It follows that for $p \geq 2$, $p_C(x_0) \in \operatorname{argmin} \delta_C + \| \cdot - x_0\|_p^p$ is in fact a ψ-minimizer of f with $\psi(\eta) = 2^{2-p}\eta^p$. Similarly, if $p \leq 2$, then $p_C(x_0) \in \operatorname{argmin} \delta_C + \| \cdot - x_0\|_p^{p'}$ is a ψ-minimizer of f for $\psi(\eta) = 2^{2-p'}\eta^{p'}$. The next proposition then follows from Theorem 3.5.

PROPOSITION 6.4. *Let X be a Banach space of type p (say $L^p(\Omega), l^p(\mathbb{N}) \ldots$) with $1 < p < \infty$. For $x_0 \in X$ and C nonempty, closed, and convex, the mapping $C \mapsto p_C(x_0)$ is Hölder continuous with exponent $1/p$ if $p \geq 2$ and with exponent $1/p' = (p-1)/p$ if $p \leq 2$.*

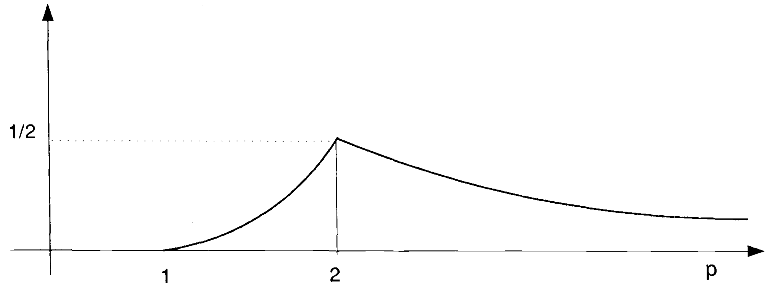


FIG. 7. *Variation of the Hölder exponent.*

Figure 7 shows the variation of the Hölder exponent as a function of p. It is in the Hilbert case that we are able to obtain the best stability result; the Hilbert metric is well suited to approximation theory. In contrast, when $p \to 1$ or $p \to \infty$ the Hölder exponent goes to zero. Indeed, it is well known that in a Banach space of type $l^1, L^1, \ldots$, the solution of a minimization problem involving the norm may not be unique because the norm is not uniformly convex. Let us consider the case $X = \mathbb{R}^n$ equipped with the l^1-norm. Then $p_C(x_0)$ is, in general, a nonempty convex set.

It may be tempting to conjecture that when $C_\theta \to C$ with respect to the Pompeiu–Hausdorff distance, then $p_{C_\theta}(x_0) \to p_C(x_0)$ for the Pompeiu–Hausdorff distance. But that is not the case, as can be seen from the following simple example: Let $X = \mathbb{R}^2$, $\|x\| = |x_1| + |x_2|$, $x_0 = (0,0)$, $C = \{\lambda(0,1) + (1-\lambda)(1,0) \mid \lambda \in [0,1]\}$, and $C_\theta = \{\lambda(0,1) + (1-\lambda)(1 + (1/\theta), 0) \mid \lambda \in [0,1]\}$; cf. Fig. 8.

When working in a nonreflexive Banach space, we show in [8] that the notion of approximate ε-solution still enjoys good stability properties.

Acknowledgment. We are particularly thankful to Dominique Azé (Perpignan), Alain Fougères (Perpignan), R. T. Rockafellar (Washington), and Pierre-Jean Laurent (Grenoble) for many comments and suggestions.

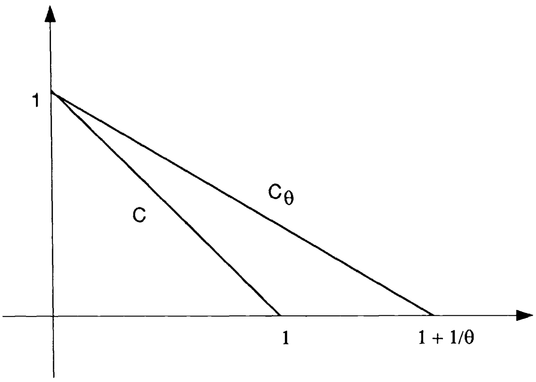


FIG. 8. *Projections and the Hausdorff metric.*

REFERENCES

[1] R. A. ADAMS, *Sobolev Spaces*, Academic Press, New York, 1976.

[2] H. ATTOUCH, *Variational Convergence for Functions and Operators*, Applicable Mathematics Series, Pitman, London, 1984.

[3] H. ATTOUCH, R. LUCCHETTI, AND R. J.-B. WETS, *The topology of the ρ-Hausdorff distance*, Ann. Mat. Pura Appl., CLX (1991), pp. 303–320.

[4] H. ATTOUCH AND R. J.-B. WETS, *Approximation and convergence in nonlinear optimization*, in Nonlinear Programming 4, O. Mangasarian , R. Meyer, and S. Robinson, eds., Academic Press, New York, 1981, pp. 367–394.

[5] ———, *Isometries for the Legendre–Fenchel transform*, Trans. Amer. Math. Soc., 296 (1986), pp. 33–60.

[6] ———, *Another isometry for the Legendre–Fenchel transform*, J. Math. Anal. Appl., 131 (1988), pp. 404 - 411.

[7] ———, *Quantitative stability of variational systems:* I. *The epigraphical distance*, Trans. Amer. Math. Soc., 3 (1991), pp. 695–729.

[8] ———, *Quantitative stability of variational systems:* III. *ε-approximate solutions*, Math. Programming, (1993), to appear.

[9] J.-P. AUBIN, *Lipschitz behavior of solutions to convex minimization problems*, Math. Oper. Res., 9 (1984), pp. 87–111.

[10] J.-P. AUBIN AND H. FRANKOWSKA, *On inverse function theorems for set-valued maps*, J. Math. Pures Appl., 66 (1987), pp. 71 - 89.

[11] J.-P. AUBIN AND R. J.-B. WETS, *Stable approximations of set-valued maps*, Ann. Ins. H. Poincaré, 5 (1988), pp. 519–535.

[12] D. AZÉ, *Characterization of uniform convexity of functionals*, Tech. Rep., AVAMAC, Université de Perpignan, Perpignan, France, 1986.

[13] D. AZÉ AND J.-P. PENOT, *Recent quantitative results about the convergence of convex sets and functions*, in Functional Analysis and Approximations. Proc. Internat. Conference Bagni di Lucca, May 1988, P. L. Papini, ed., Pitagora Editrice, Bologna, 1990, pp. 90–110.

[14] B. BANK, J. GUDDAT, D. KLATTE, B. KUMMER, AND K. TAMMER, *Non-linear Parametric Optimization*, Birkhäuser-Verlag, Basel, 1983.

[15] G. BEER AND R. LUCCHETTI, *The epi-distance topology: Continuity and stability results with applications to convex optimization*, Math. Oper. Res., 1992, to appear.

[16] J. W. DANIEL, *Stability of the solution of definite quadratic programs*, Math. Programming, 5 (1973), pp. 41–53.

[17] S. DOLECKI, *Convergence of minima in convergence spaces*, Optimization, 17 (1986), pp. 553–572.

[18] A. L. DONTCHEV, *Perturbations, Approximations and Sensitivity Analysis of Optimal Control System*, Lecture Notes in Control and Inform. Sci. 52, Springer-Verlag, Berlin, 1983.

[19] J.-P. Evans and F. J. Gould, *Stability in nonlinear programming*, Oper. Res., 18 (1970), pp. 107–118.

[20] A. V. Fiacco, *Convergence properties of local solutions of sequences of nonlinear programming problems in general spaces*, J. Optimiz. Theory Appl., 13 (1974), pp. 1–12.

[21] ———, *Optimization with data perturbations*, Ann. Oper. Res., 27 (1990).

[22] A. Fougères, *Convexité et coercivité: structure semi-normée intrinsique*, Séminaire d'Analyse Convexe de Montpellier-Perpignan, exposé no. 6, 1981.

[23] J. Gauvin, *The generalized gradient of a marginal function in mathematical programming*, Math. Oper. Res., 4 (1979), pp. 458–463.

[24] J. Gauvin and R. Janin, *Directional behaviour of optimal solutions in nonlinear mathematical programming*, Math. Oper. Res., 13 (1988), pp. 629–649.

[25] ———, *Directional derivative of the value function in parametric optimization*, Ann. Oper. Res., 27 (1990), pp. 237–252.

[26] J.-B. Hiriart-Urruty, *Lipschitz r-continuity of the approximate subdifferential of a convex function*, Mat. Scand., 47 (1980), pp. 123–134.

[27] W. W. Hogan, *Point-to-set maps in mathematical programming*, SIAM Rev., 15 (1973), pp. 591–603.

[28] P. Kall, *Approximation to optimization problems: An elementary review*, Math. Oper. Res., 11 (1986), pp. 9–18.

[29] Y. Kaniovski, A. King, and R. J.-B. Wets, *Probabilistic bounds (via large deviations) for the solutions of stochastic programming problems*, IIASA Working Paper #92-xx, 1992, in preparation.

[30] D. Klatte and B. Kummer, *On the (Lipschitz) continuity of solutions of parametric optimization problems*, in Proc. 16th Conference on Mathematical Optimization, Selin 1986, Seminarbericht no. 64, Humboldt Universität Berlin, (1986), pp. 50–61.

[31] B. Kummer, *Stability of generalized equations and Kuhn–Tucker points of perturbed convex programs*, in Proc. 11th TIMS Internat. Conference, Copenhagen 1983.

[32] C. Lemaire-Misonne, *Validation des résultats: conditionement de problèmes*, in Proc. 7ème Rencontre Franco-Belge de Statisticiens, Rouen, 1986.

[33] J.-J. Moreau, *Rafle par un convexe variable, 2ème partie*, Séminaire d'Analyse Convexe, Université de Montpellier, 2 (1972), pp. 3:1–3:47.

[34] U. Mosco, *Convergence of convex sets and of solutions of variational inequalities*, Adv. Math., 3 (1969), pp. 510–585.

[35] P. Rabier and J. M. Thomas, *Exercises d'Analyse Numérique des Equations aux Dérivèes Partielles*, Masson, Paris, 1985.

[36] S. M. Robinson, *Stability theory for systems of inequalities. Part* I: *Linear systems*, SIAM J. Numer. Anal., 12 (1975), pp. 754–769.

[37] ———, *Stability theory for systems of inequalities. Part* II: *Differentiable nonlinear systems*, SIAM J. Numer. Anal., 13 (1976), pp. 754–769.

[38] ———, *Local epi-continuity and local optimization*, Math. Programming, 37 (1987), pp. 208 – 222.

[39] ———, *An implicit-function theorem for a class of nonsmooth functions*, Math. Oper. Res., 16 (1991), pp. 292–309.

[40] R. T. Rockafellar, *Directional differentiability of the optimal value in a nonlinear programming problem*, Math. Programming Stud., 21 (1984), pp. 87–111.

[41] ———, *Maximal monotone relations and the second derivative of nonsmooth functions*, Ann. Inst. H. Poincaré: Anal. Non Linéaire, 2 (1985), pp. 167–184.

[42] R. T. Rockafellar, *Lipschitzian stability in optimization: The role of nonsmooth analysis*, in Nondifferentiable Optimization: Motivations and Applications, V. Demyanov, and D. Pallatschke, eds., Springer-Verlag Lecture Notes in Economics and Mathematical Systems, No. 255, 1985. pp. 55–73.

[43] ———, *First and second-order epi-differentiability in nonlinear programming*, Trans. Amer. Math. Soc., 307 (1988), pp. 75–108.

[44] ———, *Second-order optimality conditions in nonlinear programming obtained by way of epi-derivatives*, Math. Oper. Res., 14 (1989), pp. 462 – 484.

[45] R. T. Rockafellar and R. J.-B. Wets, *Variational systems, an introduction*, in Multifunctions and Integrands: Stochastic Analysis, Approximation and Optimization, G. Salinetti, ed., Lecture Notes in Math. 1091, Springer-Verlag, 1984, pp. 1–54.

[46] R. Schultz, *Estimates for Kuhn–Tucker points of perturbed convex programs*, Tech. Rep., Sektion Mathematik, Humbold Univ., Berlin, Germany, 1986.

[47] Y. Sonntag, *Approximation et pénalisation en optimisation*, Publications de Mathématiques Appliquées, n. 83-2, Université de Provence, France, 1983.

[48] A. N. Tykhonov, *On the stability of functional optimization problem*, Comput. Math. and Math. Phys., 6 (1966), pp. 28–33.

[49] A. B. Vladimirov, J. E. Nestorov, and J. N. Chekanov, *On uniformly convex functions*, manuscript, 1986.

[50] R. J.-B. Wets, *Convergence of convex functions, variational inequalities and convex optimization problems*, in Variational Inequalities and Complementarity Problems, R. Cottle , F. Giannessi, and J. L. Lions, eds., John Wiley, Chichester, 1980, pp. 405–419.

[51] C. Zalinescu, *On uniformly convex functions*, J. Math. Anal. Appl., 95 (1983), pp. 344–374.

[52] T. Zolezzi, *On equi-wellset minimum problems*, Optimization, 4 (1978), pp. 209–223.

[53] ———, *A characterization of well-posed optimal control systems*, SIAM J. Control Optim., 19 (1981), pp. 604–616.

SIAM J. OPTIMIZATION
Vol. 3, No. 2, pp. 382–397, May 1993

PARTIAL-UPDATE NEWTON METHODS FOR UNARY, FACTORABLE, AND PARTIALLY SEPARABLE OPTIMIZATION*

DONALD GOLDFARB† AND SIYUN WANG‡

Abstract. A modified Newton method for solving unary optimization problems that is based upon only partially updating an approximation to the Hessian matrix at each iteration is developed using rank-one updates. Two partial updating criteria are presented: the first enables the method to retain the quadratic convergence property of the classical Newton method, while the second enables it to achieve the superlinear convergence property of quasi-Newton methods. Globally convergent modifications of the partial-update Newton method are also given. Finally, the methods and proofs of their convergence are extended to partially separable and factorable optimization problems.

Key words. unary function, partially separable function, factorable function, inexact Newton method, partial-update Newton method, rank-one update

AMS subject classifications. 65K05, 65K10

1. Introduction.

Consider the unconstrained nonlinear optimization problem

$$\min_{x \in \mathbf{R}^n} f(x). \tag{1.1}$$

Following McCormick and Sofer [20] we call problem (1.1) a *unary optimization* problem if $f(x)$ takes the form

$$f(x) = \sum_{i=1}^{m} U_i\left(\alpha_i(x)\right), \tag{1.2}$$

where, for $i = 1, \ldots, m$, $\alpha_i(x) = a_i^T x$, a_i is a constant vector of size $n \times 1$, and $U_i(\cdot)$ is a *unary function*, i.e., $U_i(\cdot)$ is a function of a single argument. Note that a separable function $f(x) = \sum_{i=1}^{n} f_i(x_i)$, the objective function of the *linear robust regression* problem (e.g., see Byrd [1]), and the *dual* objective function of the *entropy* problem in information theory (e.g., see Eriksson [5]) are of the form (1.2).

Let us assume that the unary functions $U_i(\cdot)$, $i = 1, \ldots, m$, in (1.2) are all twice continuously differentiable. Then, using the chain rule of differentiation, the gradient vector and the Hessian matrix of function (1.2) are

$$\nabla f(x) = \sum_{i=1}^{m} \left(\frac{dU_i(\alpha_i(x))}{d\alpha_i} \right) a_i \quad \text{and} \quad \nabla^2 f(x) = \sum_{i=1}^{m} \left(\frac{d^2 U_i(\alpha_i(x))}{d\alpha_i^2} \right) a_i a_i^T, \tag{1.3}$$

respectively.

For problem (1.1), if we assume that $f(x)$ takes the form (1.2) and the Hessian matrix $\nabla^2 f(x)$ is nonsingular for all $x \in R^n$, then a classical algorithm for finding a solution to (1.1) is Newton's method:

*Received by the editors January 8, 1990; accepted for publication (in revised form) March 4, 1992. This research was supported in part by National Science Foundation grants DMS 85-12277, DMS 91-06195, and CDR 84-21402, and by Office of Naval Research contract N-00014-87-K0214.

†Department of Industrial Engineering and Operations Research, Columbia University, New York, New York 10027.

‡Chemical Banking Corporation, New York, New York 10017.

Given an initial point x_0, for $k = 0, 1, \ldots$, compute a sequence of steps $\{s_k\}$ and iterates $\{x_k\}$ as follows:

$$\text{(1.4)} \qquad \begin{aligned} \text{solve} \quad & \nabla^2 f(x_k) s_k = -\nabla f(x_k) \\ \text{and set} \quad & x_{k+1} = x_k + s_k. \end{aligned}$$

It is well known that, under suitable conditions, Newton's method has a local quadratic rate of convergence; i.e., there exists a positive constant γ such that

$$\|x_{k+1} - x^*\| \leq \gamma \|x_k - x^*\|^2$$

if x_0 is sufficiently close to x^*, where x^* is a stationary point of (1.1).

On each iteration of Newton's method (1.4), after forming the Hessian matrix and gradient, we need to solve a system of linear equations, which takes $O(n^3)$ operations. When the Hessian matrix has the special form (1.3), we can modify the above Newton method (1.4) to develop a more efficient algorithm for solving the unary optimization problem (1.1)–(1.2). To see this, let

$$\nabla^2 f(x) = \sum_{i=1}^{m} \phi_i(x) a_i a_i^T = A^T \Phi(x) A,$$

where $\phi_i(x) = d^2 U_i(\alpha_i(x))/d\alpha_i^2$, $i = 1, \ldots, m$, and $\Phi(x) = \operatorname{diag}\{\phi_1(x), \ldots, \phi_m(x)\}$, and $A^T = [a_1, \ldots, a_m]$. Clearly, as the Hessian matrix changes, from step to step, only the diagonal matrix Φ is affected. Suppose that only the jth diagonal elements of $\Phi(x_k)$ and $\Phi(x_{k-1})$ differ at iteration k, i.e.,

$$A^T \Phi(x_k) A = A^T \Phi(x_{k-1}) A + (\phi_j(x_k) - \phi_j(x_{k-1}))\, a_j a_j^T,$$

or, equivalently,

$$\nabla^2 f(x_k) = \nabla^2 f(x_{k-1}) + (\phi_j(x_k) - \phi_j(x_{k-1}))\, a_j a_j^T.$$

Consequently, $\nabla^2 f(x_k)^{-1}$ can be obtained from $\nabla^2 f(x_{k-1})^{-1}$ (assuming that both $\nabla^2 f(x_k)$ and $\nabla^2 f(x_{k-1})$ are invertible and $\nabla^2 f(x_{k-1})^{-1}$ is given) by the well-known Sherman–Morrison rank-one updating formula [24]:

$$\left(H + \omega c c^T\right)^{-1} = H^{-1} - H^{-1} c \left(\omega^{-1} + c^T H^{-1} c\right)^{-1} c^T H^{-1},$$

where H, c, and ω correspond to $\nabla^2 f(x_{k-1})$, a_j, and $\phi_j(x_k) - \phi_j(x_{k-1})$, respectively. Therefore, the next iterate

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

can be obtained in only $O(n^2)$ arithmetic operations after evaluating ∇f and ϕ_i, $i = 1, \ldots, m$. If l entries of $\Phi(x_k)$ and $\Phi(x_{k-1})$ differ, we can perform l rank-one updates and obtain the new inverse $\nabla^2 f(x_k)^{-1}$ in $O(ln^2)$ operations. A similar approach is used in polynomial interior point algorithms for linear and quadratic programming to reduce their complexity bounds (see, e.g., Karmarkar [18], Gonzaga [13], Goldfarb and Liu [12], and Ye [27]).

The foregoing observation motivates us to consider the following modified Newton method, which we refer to as a *partial-update* Newton method. This method takes advantage of the computation done in previous steps and only partially updates the diagonal matrix Φ, even though all of its diagonal elements might change at each iteration, using the rank-one updating formula given above to obtain the inverse of an approximation to the exact Hessian matrix at the current iterate. To be precise, we define a diagonal matrix $\Phi^k = \operatorname{diag}\{\phi_1^k, \dots, \phi_m^k\}$, a "working approximation" to $\Phi(x_k)$ in step k, by $\phi_i^0 = \phi_i(x_0)$, $i = 1, \dots, m$, and for $k \geq 1$,

$$\phi_i^k = \begin{cases} \phi_i^{k-1} & \text{if } \phi_i(x_k) \text{ is "replaceable" by } \phi_i^{k-1}, \\ \phi_i(x_k) & \text{otherwise,} \end{cases}$$

for $i = 1, \dots, m$. Then from Φ^k we obtain a "working approximation" $H^k = A^T \Phi^k A$ to $\nabla^2 f(x_k)$, or equivalently,

$$H^k = \nabla^2 f(x_k) + E_k,$$

where $E_k = \sum_{i=1}^m \left(\phi_i^k - \phi_i(x_k)\right) a_i a_i^T$, and compute

$$s_k = -\left(H^k\right)^{-1} \nabla f(x_k) = -\left(\nabla^2 f(x_k) + E_k\right)^{-1} \nabla f(x_k), \tag{1.5}$$

instead of computing s_k as in (1.4).

Since $\nabla^2 f(x_k) s_k = -\nabla f(x_k) + r_k$, where

$$r_k = E_k \left(\nabla^2 f(x_k) + E_k\right)^{-1} \nabla f(x_k), \tag{1.6}$$

the partial-update Newton method can be viewed as a special type of *inexact* Newton method. In [2], Dembo, Eisenstat, and Steinhaug analyzed the convergence properties of such methods under various assumptions on the forcing sequence $\{\eta_k\}$ of bounds on the relative residuals in (1.6), where $\|r_k\|/\|\nabla f(x_k)\| \leq \eta_k$. Also note that, if $\phi_i^l = \phi_i^0 = \phi_i(x_0)$, $i = 1, \dots, m$, for all $l \geq 1$, then the partial-update Newton method reduces to the simplified Newton method $x_{k+1} = x_k - \nabla^2 f(x_0)^{-1} \nabla f(x_k)$, for all $k \geq 1$.

In the first subsection of §2 we prove the local convergence of the conceptual partial-update Newton algorithm and in the second subsection we establish the rates of convergence for two variants of the method determined by different partial-update criteria. Our analysis is closely related to the analysis of Dembo, Eisenstat, and Steinhaug [2]. In §3, two globally convergent modifications of the partial-update Newton method are presented and some preliminary numerical results obtained using these methods are given. The final section is devoted to an extension of the partial-update Newton method to partially separable and factorable functions.

2. Local convergence results.

In this section we assume:

(A1) there exists a point $x^* \in \mathbf{R}^n$ with $\nabla f(x^*) = 0$;

(A2) $\nabla^2 f(x) = A^T \Phi(x) A$ and $\nabla^2 f(x^*)$ is nonsingular, where $A^T = [a_1, \dots, a_m]$ is an $n \times m$ matrix with rank n, a_i is the ith column of A^T, $i = 1, \dots, m$, and $\Phi(x) = \operatorname{diag}\{\phi_1(x), \dots, \phi_m(x)\}$;

(A3) $\Phi(x)$ is continuous in a neighborhood of x^*.

We use the Euclidean vector norm and the matrix norm induced from it, both of which we denote by $\|\cdot\|$, and we define $\beta = \|\nabla^2 f(x^*)^{-1}\|$.

2.1. Local convergence.

Here we show that, under assumptions (A1)–(A3), the partial-update Newton method is locally linearly convergent. First note that, under (A1)–(A3), for any $\epsilon > 0$ and $\tau > 0$ there exists a $\delta_1 > 0$ such that

$$\|\Phi(x) - \Phi(y)\| < \tau \tag{2.1}$$

and

$$\|\nabla f(x_k) - \nabla f(x^*) - \nabla^2 f(x^*)(x_k - x^*)\| \le \epsilon \|x_k - x^*\|, \tag{2.2}$$

provided that $\max\{\|x - x^*\|, \|y - x^*\|, \|x_k - x^*\|\} < \delta_1$.

The next result is an immediate consequence of the Perturbation Lemma in Ortega and Rheinboldt [22, Theorem 2.2.3].

LEMMA 2.1. *Under assumptions* (A1)–(A3), *for* $0 < \epsilon < \frac{1}{4\beta}$, *there exists a positive constant* δ_2 *such that*

$$\|\left(\nabla^2 f(x_k) + E_k\right) - \nabla^2 f(x^*)\| < \epsilon, \tag{2.3}$$

$\nabla^2 f(x_k) + E_k$ *is nonsingular, and*

$$\|\left(\nabla^2 f(x_k) + E_k\right)^{-1}\| < 2\beta, \tag{2.4}$$

provided that $\|x_k - x^*\| < \delta_2$ *and* $\|\Phi^k - \Phi(x_k)\| < \tau$, *where* $\tau = \epsilon/2\|A\|^2 < 1/8\beta\|A\|^2$.

THEOREM 2.2. *Let assumptions* (A1)–(A3) *hold. Then there exists* $\delta > 0$ *such that, if* $\|x_0 - x^*\| \le \delta$, *the sequence* $\{x_k\}$ *generated by the partial-update Newton method converges to* x^*. *Moreover, the convergence is linear, i.e.,*

$$\|x_{k+1} - x^*\| \le t\|x_k - x^*\| \quad \forall k, \tag{2.5}$$

where $0 < t < 1$.

Proof. Let $\epsilon > 0$ be such that $0 < t = 4\epsilon\beta < 1$, $\tau = \frac{\epsilon}{2}\|A\|^2$, and $\delta = \min\{\delta_1, \delta_2\}$ so that (2.1)–(2.4) hold. Assuming that $\|x_0 - x^*\| < \delta$, we prove (2.5) by induction.

Since E_0 is the zero matrix, it is easy to verify that (2.5) is true for $k = 0$. Now supposing that (2.5) is true for $k \le N - 1$, then $\|x_l - x^*\| \le t^l\|x_0 - x^*\| < \delta$, $0 \le l \le N$. Hence, since $|\phi_i^N - \phi_i(x_N)| = |\phi_i(x_{l_i}) - \phi_i(x_N)|$ for some l_i, where $1 \le l_i \le N$, it follows from (2.1) that

$$\|\Phi^N - \Phi(x_N)\| = \max_{1 \le i \le m} |\phi_i^N - \phi_i(x_N)| \le \max_{0 \le l \le N}\{\|\Phi(x_l) - \Phi(x_N)\|\} < \tau.$$

Therefore, (2.2)–(2.4) hold with $k = N$, and

$$\begin{aligned} x_{N+1} - x^* &= x_N - x^* - \left(\nabla^2 f(x_N) + E_N\right)^{-1} \nabla f(x_N) \\ &= \left(\nabla^2 f(x_N) + E_N\right)^{-1} \left(\nabla^2 f(x_N) + E_N - \nabla^2 f(x^*)\right)(x_N - x^*) \qquad (2.6) \\ &\quad - \left(\nabla^2 f(x_N) + E_N\right)^{-1} \left(\nabla f(x_N) - \nabla f(x^*) - \nabla^2 f(x^*)(x_N - x^*)\right). \end{aligned}$$

The result then follows by taking norms and using the triangle inequality. □

2.2. Rates of convergence. In this section we assume

(A4) $\Phi(x)$ is Lipschitz continuous at x^* with Lipschitz constant L.

It then follows that $\nabla^2 f(x)$ is Lipschitz continuous at x^* with Lipschitz constant $\|A\|^2 L$ and, from Ortega and Rheinboldt [22, Theorem 3.2.12], that (2.2) can be strengthened to

$$\|\nabla f(x_k) - \nabla f(x^*) - \nabla^2 f(x^*)(x_k - x^*)\| \le \frac{L}{2}\|A\|^2\|x_k - x^*\|^2 \tag{2.7}$$

for $\|x_k - x^*\|$ sufficiently small.

We now give two "replacement" criteria for the partial-update Newton method. The first one retains the local quadratic convergence property of the classical Newton method. Under the second criterion our partial-update Newton method converges superlinearly and there is trade-off between the rate of convergence and the number of rank-one updates.

Criterion 1. For $i = 1, \dots, m$,

$$\phi_i^k \text{ is replaceable by } \phi_i^{k-1} \text{ if } \frac{|\phi_i(x_k) - \phi_i^{k-1}|}{|\phi_i(x_k) - \phi_i^{k-1}| + \|\nabla f(x_k)\|} < \eta,$$

where $0 < \eta < 1$. Note that Criterion 1 essentially says to keep $\phi_i^k = \phi_i^{k-1}$ as long as $\|\nabla f(x_k)\|$ is not too small relative to $|\phi_i(x_k) - \phi_i^{k-1}|$. Therefore, only as $x_k \to x^*$, i.e., only as x_k becomes very close to x^*, is ϕ_i^k set equal to $\phi_i(x_k)$ if η is close to 1, say $\eta = 0.99$.

Criterion 2. For $i = 1, \dots, m$,

$$\phi_i^k \text{ is replaceable by } \phi_i^{k-1} \text{ if } k \le p \quad \text{or} \quad \frac{|\phi_i(x_k) - \phi_i^{k-1}|}{\max\limits_{k-p+1 \le j \le k} \{|\phi_i(x_k) - \phi_i(x_{j-1})|\}} \le 1,$$

where p is a given positive integer.

In order to characterize the rates of convergence for variants of the partial-update Newton method that use these two replacement criteria, we need the following lemma.

LEMMA 2.3 (Ortega and Rheinboldt [22, Theorems 9.2.8 and 9.2.9]). *Let the sequence $\{x_k\}$, which is generated by an iterative process, converge to a limit x^*. Furthermore, let $\gamma_0, \gamma_1, \dots, \gamma_l$ be nonnegative constants. If there is a $k_0 \ge l$ such that*

$$\|x_{k+1} - x^*\| \le \|x_k - x^*\| \left(\sum_{j=0}^{l} \gamma_j \|x_{k-j} - x^*\| \right) \quad \forall k \ge k_0,$$

then the iterates $\{x_k\}$ converge to x^* with R-order at least r_l, where r_l is the unique positive root of $t^{l+1} - t^l - 1 = 0$. Moreover, $r_l \in (1, 2)$, $r_{l+1} < r_l$, and $\lim_{l\to\infty} r_l = 1$.

THEOREM 2.4. *Let assumptions* (A1), (A2), *and* (A4) *hold and let $\{x_k\}$ be the sequence of iterates generated by the partial-update Newton method. Then*

(1) *$\{x_k\}$ is locally quadratically convergent to x^* if Criterion* 1 *is used and*

(2) *$\{x_k\}$ is locally superlinearly convergent to x^* with R-order at least r_p, where r_p is the unique positive root of $t^{p+1} - t^p - 1 = 0$, if Criterion* 2 *is used. Moreover, $1 < r_p < 2$, $r_{p+1} < r_p$, and $\lim_{p\to\infty} r_p = 1$.*

Proof. (1) Since, under Criterion 1, $\|\Phi^k - \Phi(x_k)\| \le \frac{\eta}{1-\eta}\|\nabla f(x_k)\|$, conclusion (1) follows from Theorem 3.4 in [2].

(2) Under Criterion 2, we have from (2.5) and (A4) that

$$\begin{aligned} \|\Phi^k - \Phi(x_k)\| &\le \max_{k-p+1 \le j \le k} \left\{ \|\Phi(x_k) - \Phi(x^*)\| + \|\Phi(x_{j-1}) - \Phi(x^*)\| \right\} \\ &\le 2L\|x_{k-p} - x^*\| \end{aligned} \tag{2.8}$$

if $k > p$. Let $\|x_0 - x^*\| \le \delta$ and δ be sufficiently small so that Theorem 2.2 holds and $\|\Phi(x) - \Phi(x^*)\| \le L\|x - x^*\|$ for $\|x - x^*\| \le \delta$. It then follows from (2.6)–(2.8) that

$$\|x_{k+1} - x^*\| \le \|x_k - x^*\| \left(3\beta L\|A\|^2 \cdot \|x_k - x^*\| + 4\beta L\|A\|^2 \cdot \|x_{k-p} - x^*\| \right) \quad \forall k > p.$$

Conclusion 2 then follows from Lemma 2.3. □

Under Criterion 2, the parameter p determines the average number of rank-one updates required at each iteration, as well as the R-order of the convergence of the iterates to x^*. This fact is established by the following proposition.

PROPOSITION 2.5. *On the average, the number of the rank-one updates at each iteration is at most $\frac{m}{p}$ if Criterion 2 is used.*

Proof. It follows from Criterion 2 that for all i, $i = 1, \ldots, m$, $\phi_i^p = \phi_i^{p-1} = \cdots = \phi_i^0 = \phi_i(x_0)$, and that after any iteration k in which $\phi_i(x_k)$ is *not* replaceable by ϕ_i^{k-1}, i.e., ϕ_i^k is set equal to $\phi_i(x_k)$, then $\phi_i(x_l)$ is replaceable by ϕ_i^{l-1} during at least the next p iterations $l = k+1, \ldots, k+p$. The proposition is an immediate consequence of this observation. □

If we use Criterion 2 and take $p = m$, then each iteration of the partial-update Newton method requires on the average at most one rank-one update, and hence just $O(n^2)$ operations, the same amount of work as in quasi-Newton methods. In [6] Gay proved that Broyden's so-called "good" and "bad" rank-one update quasi-Newton methods converge superlinearly to a stationary point x^* of $f(x)$ with order at least $2^{1/2n}$. When $p = m$, the order of convergence of our partial-update Newton method $r_m > 2^{1/2n}$, if $n \le m \le cn$ and $1 \le c \ll n$. Thus, in this special case, the lower bound on the efficiency of our method is better than the lower bound for either of Broyden's methods.

Note that, under Criterion 2, each ϕ_i, $i = 1, \ldots, m$, stays fixed for at least p iterations and all ϕ_i may not get updated at the same time. If all of the ϕ_i stay fixed for *exactly* p iterations and they are all updated at the *same* time, then the partial-update Newton method under Criterion 2 reduces to the p-step method

$$\begin{aligned} &x_{k+1} = x_{k,p}, \quad x_{k,i+1} = x_{k,i} - \nabla^2 f(x_k)^{-1} \nabla f(x_{k,i}), \\ &i = 0, 1, \ldots, p-1, \quad x_{k,0} = x_k, \end{aligned} \tag{2.9}$$

considered in Traub [26], in which each major iteration consists of p simplified Newton steps. Shamanskii [23] considered the p-step Newton-like method obtained by replacing the $\nabla^2 f(x_k)$ in (2.9) by the operator J_k whose jth column is $J_k e_j = (\nabla f(x_k + h_k e_j) - \nabla f(x_k)) / h_k$ for $j = 1, \ldots, n$. (Here e_j is the jth column of the identity matrix and h_k is of order $\|\nabla f(x_k)\|$.)

3. Globally convergent implementations. In the first part of this section we present two modifications of the partial-update Newton method to make it globally convergent. In the second part we give some preliminary numerical results.

3.1. Globally convergent modifications. From the local convergence analysis of §2, we know that the partial-update Newton methods considered there converge rapidly to a stationary point x^* of $f(x)$ once they get close enough to such a point. However, if these methods do not start near enough to x^*, they can fail to converge. Also, if the partially updated Hessian $A^T \Phi^k A$ is singular, these methods are not well defined. Therefore, as in the case of Newton's method, it is necessary to modify our partial-update Newton methods so that they converge globally. In this section we propose modifications that utilize the special structure of $A^T \Phi^k A$ to compute a positive definite approximate $A^T \hat{\Phi}^k A$ so that a descent direction is obtained.

Consider the following Wolfe-type linesearch algorithm.

ALGORITHM 3.1. For given α_1 and α_2, where $\alpha_1 \in (0, \frac{1}{2})$ and $\alpha_2 \in (\alpha_1, 1)$, and a given point x_0, determine x_{k+1}, $k = 0, 1, 2, \ldots$, as follows: If convergence stop. Otherwise, compute the descent direction $d_k = -(A^T \hat{\Phi}^k A)^{-1} \nabla f(x_k)$, where $A^T \hat{\Phi}^k A$ is nonsingular, and choose a steplength $\lambda_k > 0$, such that

$$f\,(x_k + \lambda_k d_k) \le f(x_k) + \alpha_1 \lambda_k \nabla f(x_k)^T d_k \tag{3.1a}$$

and

$$\nabla f\,(x_k + \lambda_k d_k)^T\, d_k \ge \alpha_2 \nabla f(x_k)^T d_k, \tag{3.1b}$$

and set $x_{k+1} = x_k + \lambda_k d_k$.

Let the smallest and largest eigenvalues of matrix H be denoted by $\lambda_{\min(H)}$ and $\lambda_{\max(H)}$, respectively.

THEOREM 3.1. *Let $\Phi(x)$ be continuous on an open set D and let the level set $S = \{x : f(x) \le f(x_0)\}$ be a compact subset of D for a given $x_0 \in D$, and assume that $\nabla f(x_k) \neq 0$ for all $k \ge 0$ and f has a finite number of stationary points in S. Then if there exist constants μ_1 and μ_2, where $0 < \mu_1 \le \mu_2$, such that $\mu_1 \le \lambda_{\min(A^T\hat{\Phi}^k A)} \le \lambda_{\max(A^T\hat{\Phi}^k A)} \le \mu_2$ for any $k \ge 0$, the sequence of iterates $\{x_k\}$ generated by Algorithm* 3.1 *converges to some $x^* \in S$ with $\nabla f(x^*) = 0$. Moreover, the rate of convergence is at least R-linear if $\nabla^2 f(x^*)$ is invertible.*

Proof. From the fairly standard argument (e.g., see Dennis and Schnabel [3], Goldfarb [11], or Moré and Sorensen [21]) $x_k \to x^* \in S$ with $\nabla f(x^*) = 0$.

Let $\delta_0 > 0$ and k_0 be such that $\Phi(x)$ is continuous on the closed ball $B = B(x^*, \delta_0) \subset S$ and $x_k \in B$ for all $k \ge k_0$. Then, since $\|\nabla f(x_k)\|^2 = d_k^T(A^T\hat{\Phi}^k A)^2 d_k \le \mu_2^2\|d_k\|^2$, $-d_k^T \nabla f(x_k) = d_k^T A^T \hat{\Phi}^k A d_k \ge \mu_1 d_k^T d_k$, and, from (3.1b),

$$-\nabla f(x_k)^T d_k \le \frac{1}{1-\alpha_2}\|\nabla f\,(x_k + \lambda_k d_k) - \nabla f(x_k)\| \cdot \|d_k\|,$$

we have from the mean-value theorem (e.g., see [22]) that

$$\frac{\mu_1}{\mu_2}\|\nabla f(x_k)\| \cdot \|d_k\| \le -\nabla f(x_k)^T d_k \le \frac{\gamma_0}{1-\alpha_2}\lambda_k \|d_k\|^2 \quad \forall k \ge k_0, \tag{3.2}$$

where $\gamma_0 = \max\{\|\nabla^2 f(x)\| \mid x \in B\}$. Hence, combining (3.1a) and (3.2), we see that

$$\begin{aligned} f(x_k + \lambda_k d_k) &\le f(x_k) - \frac{\alpha_1\mu_1}{\mu_2}\lambda_k \|\nabla f(x_k)\| \cdot \|d_k\| \\ &\le f(x_k) - \frac{\alpha_1(1-\alpha_2)\mu_1^2}{\gamma_0\mu_2^2}\|\nabla f(x_k)\|^2 \quad \forall k \ge k_0, \end{aligned}$$

and the result that the rate of convergence is at least R-linear follows from Theorem 14.1.6 in [22]. □

The simplest modification of Φ^k that ensures that the conditions on the eigenvalues of $A^T\Phi^k A$ required by Theorem 3.1 are satisfied, assuming that $\Phi(x)$ is continuous on S, is to define the modified "working approximation" $\hat{\Phi}^k = \text{diag}\{\hat{\phi}_1^k, \ldots, \hat{\phi}_m^k\}$ to $\Phi(x_k)$ by the following modifications.

Modification 1. For $i = 1, \ldots, m$, set $\hat{\phi}_i^0 = \max\{\theta, \phi_i(x_0)\}$, where θ is a prescribed small positive constant, and at step k, $k \ge 1$, define $\hat{\phi}_i^k$ by the following criteria.

Criterion 1′.

$$\hat{\phi}_i^k = \begin{cases} \hat{\phi}_i^{k-1} & \text{if } \dfrac{|\phi_i(x_k) - \hat{\phi}_i^{k-1}|}{|\phi_i(x_k) - \hat{\phi}_i^{k-1}| + \|\nabla f(x_k)\|} < \eta, \\ \max\{\theta, \phi_i(x_k)\} & \text{otherwise,} \end{cases}$$

where $0 < \eta < 1$.

Criterion 2′.

$$\hat{\phi}_i^k = \begin{cases} \hat{\phi}_i^{k-1} & \text{if } k \le p \text{ or } \dfrac{|\phi_i(x_k) - \hat{\phi}_i^{k-1}|}{\max\limits_{k-p+1 \le j \le k} \{|\phi_i(x_k) - \phi_i(x_{j-1})|\}} \le 1, \\ \max\{\theta, \phi_i(x_k)\} & \text{otherwise,} \end{cases}$$

where p is a given positive integer.

Modification 1 may be overly cautious since $A^T \Phi^k A$ can be positive definite even if some diagonal elements of Φ^k are negative. Consequently, we now propose an alternate modification which ensures that the modified "working approximation" $A^T \tilde{\Phi}^k A$ is exactly equal to the unmodified working approximation $A^T \Phi^k A$ for all k whenever $A^T \Phi^k A$ is positive definite for all k. The modification is based on a method of McCormick [19, §§7.3–7.4] for computing the positive part for a symmetric matrix given in dyadic form.

Modification 2. For $i = 1, \ldots, m$, determine σ_i^0 so $A^T \tilde{\Phi}^0 A = A^T \left(\Phi(x_0) + \Sigma^0\right) A$ is positive definite, where $\Sigma^0 = \text{diag}\,\{\sigma_1^0, \ldots, \sigma_m^0\}$, and at step k, $k \ge 1$, set

$$\tilde{\phi}_i^k = \bar{\phi}_i^k + \sigma_i^k,$$

where $\bar{\phi}_i^k$ is defined by the following.

Criterion 1″.

$$\bar{\phi}_i^k = \begin{cases} \tilde{\phi}_i^{k-1} & \text{if } \dfrac{|\phi_i(x_k) - \tilde{\phi}_i^{k-1}|}{|\phi_i(x_k) - \tilde{\phi}_i^{k-1}| + \|\nabla f(x_k)\|} < \eta, \\ \phi_i(x_k) & \text{otherwise,} \end{cases}$$

where $0 < \eta < 1$.

Criterion 2″.

$$\bar{\phi}_i^k = \begin{cases} \tilde{\phi}_i^{k-1} & \text{if } k \le p \quad \text{or } \dfrac{|\phi_i(x_k) - \tilde{\phi}_i^{k-1}|}{\max\limits_{k-p+1 \le j \le k} \{|\phi_i(x_k) - \phi_i(x_{j-1}) - \sigma_i^{j-1}|\}} \le 1, \\ \phi_i(x_k) & \text{otherwise,} \end{cases}$$

where p is a given positive integer, and $\Sigma^k = \text{diag}\,\{\sigma_1^k, \ldots, \sigma_m^k\}$ is determined as described below to guarantee that $A^T \tilde{\Phi}^k A$ is positive definite.

To specify how the σ_i^k are to be chosen in Modification 2, we define the sets $U^k = \{i \mid \bar{\phi}_i^k = \tilde{\phi}_i^{k-1}, i = 1, \ldots, m\}$, $V^k = \{i \mid \bar{\phi}_i^k > \tilde{\phi}_i^{k-1}, i = 1, \ldots, m\}$, and $W^k = \{i \mid \bar{\phi}_i^k < \tilde{\phi}_i^{k-1}, i = 1, \ldots, m\}$. If we set $\sigma_i^k = 0$ for $i \in U^k \cup V^k$ and define Ψ^k as $\Psi^k = A^T \tilde{\Phi}^{k-1} A + \sum_{i \in V^k} (\phi_i(x_k) - \tilde{\phi}_i^{k-1}) a_i a_i^T$, then it follows from the definition of $A^T \tilde{\Phi}^k A$ under Criteria 1″ or 2″ that

$$A^T \tilde{\Phi}^k A = \Psi^k + \sum_{i \in W^k} \hat{\sigma}_i^k a_i a_i^T, \tag{3.3}$$

where $\hat{\sigma}_i^k = \phi_i(x_k) + \sigma_i^k - \tilde{\phi}_i^{k-1}$. If $A^T \tilde{\Phi}^{k-1} A$ is positive definite then Ψ^k is. Moreover, if l is any index in W^k, $\Psi^k + \hat{\sigma}_l^k a_l a_l^T = \left(\Psi^k\right)^{1/2} \left(I + \hat{\sigma}_l^k \left(\Psi^k\right)^{-1/2} a_l a_l^T \left(\Psi^k\right)^{-1/2}\right) \left(\Psi^k\right)^{1/2}$ is positive definite if and only if $\hat{\sigma}_l^k > -(1/a_l^T \left(\Psi^k\right)^{-1} a_l)$ since $I + \hat{\sigma}_l^k \left(\Psi^k\right)^{-1/2} a_l a_l^T \left(\Psi^k\right)^{-1/2}$ has all unit eigenvalues except for one which equals $1 + \hat{\sigma}_l^k a_l^T \left(\Psi^k\right)^{-1} a_l$. Therefore, we can determine σ_i^k, $i \in W^k$, recursively as follows:

Let l be any index in W^k and consider $\Psi^k + \left(\phi_l(x_k) + \sigma_l^k - \tilde{\phi}_l^{k-1}\right) a_l a_l^T$. If we choose

$$\text{(3.4)} \qquad \begin{aligned} &\sigma_l^k = 0 \quad \text{if } \left(\phi_l(x_k) - \tilde{\phi}_l^{k-1}\right) a_l^T \left(\Psi^k\right)^{-1} a_l + 1 > 0, \\ &\sigma_l^k = \tilde{\phi}_l^{k-1} - \phi_l(x_k) - \gamma_l^k + \min\{\theta, \gamma_l^k\} \quad \text{otherwise,} \end{aligned}$$

where $\gamma_l^k = \left(1/a_l^T \left(\Psi^k\right)^{-1} a_l\right) > 0$ and θ is a prescribed small positive constant, and update

$$\text{(3.5)} \qquad \Psi^k := \Psi^k + \left(\phi_l(x_k) + \sigma_l^k - \tilde{\phi}_l^{k-1}\right) a_l a_l^T \quad \text{and} \quad W^k := W^k \setminus \{l\},$$

then the updated Ψ^k is positive definite, and we can repeat the above procedure until W^k is the empty set. Note that the term $\min\{\theta, \gamma_l\}$ in (3.4) not only ensures that Ψ^k remains positive definite during its recursive computation, but more important, that $\|A^T \tilde{\Phi}^k A\|$ is uniformly bounded above for all k.

Choosing Σ^k by the above procedure ensures that $A^T \tilde{\Phi}^k A$ is positive definite. Initially, we need to determine a Σ^0 so that $A^T \tilde{\Phi}^0 A = A^T \left(\Phi(x_0) + \Sigma^0\right) A$ is positive definite. If we write $A^T \left(\Phi(x_0) + \Sigma^0\right) A$ as

$$A^T \left(\Phi(x_0) + \Sigma^0\right) A = A^T A + \sum_{i=1}^{m} \left(\phi_i(x_0) + \sigma_i^0 - 1\right) a_i a_i^T,$$

this can be accomplished by applying the above procedure with $\bar{\phi}_i^k$ replaced by $\phi_i(x_0)$ and $\tilde{\phi}_i^{k-1}$ replaced by 1.

Modification 2 has several desirable properties. First, Algorithm 3.1 using Criterion $2''$ still needs, on the average, only at most $\frac{m}{p}$ rank-one updates at each iteration. Second, $\tilde{\Phi}^k = \bar{\Phi}^k$ if $A^T \bar{\Phi}^k A$ is positive definite, and hence $\tilde{\Phi}^k = \Phi^k$, where Φ^k is defined by Criterion 1 or 2, whenever $A^T \Phi^k A$ is positive definite for all k. To verify this, we just need to show that $\sigma_i^k = 0$, for $i = 1, \ldots, m$, under our selection rule, if $A^T \bar{\Phi}^k A$ is positive definite. Since for any $s \in W^k$

$$A^T \bar{\Phi}^k A = \Psi^k + \left(\phi_s(x_k) - \tilde{\phi}_s^{k-1}\right) a_s a_s^T + \sum_{i \in W^k \setminus \{s\}} \left(\phi_i(x_k) - \tilde{\phi}_i^{k-1}\right) a_i a_i^T,$$

it follows from the negative and positive definiteness of $\sum_{i \in W^k \setminus \{s\}} (\phi_i(x_k) - \tilde{\phi}_i^{k-1}) a_i a_i^T$ and $A^T \bar{\Phi}^k A$, respectively, that $\Psi^k + (\phi_s(x_k) - \tilde{\phi}_s^{k-1}) a_s a_s^T$ is positive definite, which implies that $\sigma_s^k = 0$. Updating Ψ^k and arguing inductively, one can conclude that $\sigma_i^k = 0$, $i = 1, \ldots, m$. Finally, as we point out in the next section, the extra computation required to implement Modification 2 is moderate.

Setting $\hat{\phi}_i^k = \tilde{\phi}_i^k + \theta$, for all i and k, so that $\lambda_{\min\left(A^T \hat{\Phi}^k A\right)} \geq \theta \lambda_{\min(A^T A)}$, we then have the following theorem, which is an immediate consequence of Theorem 3.1.

THEOREM 3.2. *Under the assumptions of Theorem* 3.1, *Algorithm* 3.1, *where* $\hat{\Phi}^k$ *is defined by either Modification* 1 *or* 2, *is globally and R-linearly convergent.*

3.2. Implementation. At the kth step of Algorithm 3.1 for $k \geq 1$, the main computational effort involves solving $(A^T \hat{\Phi}^k A) d_k = -\nabla f(x_k)$, where

$$\text{(3.6)} \qquad A^T \hat{\Phi}^k A = A^T \hat{\Phi}^{k-1} A + \sum_{j \in J_k} \hat{\sigma}_j^k a_j a_j^T,$$

$J_k = \{j \mid \phi_j(x_k)$ is not "replaceable" by $\hat{\phi}_j^{k-1}$ at iteration $k\}$, and $\hat{\sigma}_j^k = \max\{\phi_j(x_k), \theta\} - \hat{\phi}_j^{k-1}$ if Modification 1 is used, and $\hat{\sigma}_j^k = \phi_j(x_k) + \sigma_j^k + \theta - \hat{\phi}_j^{k-1}$ if Modification 2 is used. Assuming that the Cholesky factorization $L_{k-1}L_{k-1}^T$ of $A^T\hat{\Phi}^{k-1}A$ is available, the Cholesky factorization $L_kL_k^T$ of $A^T\hat{\Phi}^kA$ can be obtained by applying a numerically stable rank-one updating procedure, such as Method C2 in Gill et al. [8] or Method 2 in Goldfarb [10], $|J_k|$ times. If Modification 2 is used, the updates corresponding to indices $j \in V^k \subseteq J_k$ are performed first to give the Cholesky factors of the initial matrix Ψ^k. The remaining indices in J_k and the corresponding updates of the Cholesky factors are then computed using the recursive procedure (3.4), (3.5) to determine Σ^k. Note that the extra cost of computing Σ^k is just the cost of solving $|W^k|$ triangular systems of linear equations.

We now present some preliminary numerical test results. All algorithms were coded in FORTRAN and compiled by the F-77 SUN FORTRAN compiler, and the results were obtained using double precision arithmetic on a SUN SPARC. We used the termination condition $\|\nabla f(x_k)\| < 10^{-5}\max\{1, \|x_k\|\}$ and the linesearch algorithm proposed in Dennis and Schnabel [3] with the parameter settings $\alpha_1 = 10^{-4}$ and $\alpha_2 = 0.1$ in the linesearch conditions (3.1a) and (3.1b). The test functions that we used were

(1) the extended Powell singular (EPS) function in n variables:

$$f(x) = \sum_{i=1}^{n/4}\Big[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4\Big],$$

starting at $x_0 = (3, -1, 0, 1, 3, -1, 0, 1, \ldots)$;

(2) the extended Rosenbrock (ER) function in n variables:

$$\begin{aligned} f(x) &= \sum_{i=1}^{n/2}\left[100\left(x_{2i} - x_{2i-1}^2\right)^2 + (1 - x_{2i-1})^2\right] \\ &= \sum_{i=1}^{n/2}\Big[100x_{2i-1}^4 + \frac{200}{3}x_{2i}^3 - \frac{100}{3}\left(x_{2i-1} + x_{2i}\right)^3 \\ &\qquad - \frac{100}{3}\left(x_{2i} - x_{2i-1}\right)^3 + \left(1 - x_{2i-1}\right)^2 + 100x_{2i}^2\Big], \end{aligned}$$

starting at $x_0 = (-1.2, 1, -1.2, 1, \ldots)$;

(3) the extended Rosenbrock cliff (ERC) function in n variables:

$$f(x) = \sum_{i=1}^{n/2}\left[\left(\frac{x_{2i-1} - 3}{100}\right)^2 + (x_{2i} - x_{2i-1}) + \exp\left[20(x_{2i-1} - x_{2i})\right]\right],$$

starting at $x_0 = (0, -1, 0, -1, \ldots)$;

(4) the variably dimensional (VD) function in n variables:

$$f(x) = \sum_{i=1}^{n}(x_i - 1)^2 + \left[\sum_{i=1}^{n} i(x_i - 1)\right]^2 + \left[\sum_{i=1}^{n} i(x_i - 1)\right]^4,$$

starting at $x_0 = (\xi_i)$, where $\xi_i = 1 - \frac{i}{n}$, $i = 1, \ldots, n$; and

(5) the Broyden tridiagonal (BT) function in n variables:

$$
\begin{aligned}
f(x) &= \sum_{i=1}^{n} [(3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1]^2 \\
&= \sum_{i=1}^{n} \Bigg[4x_i^4 - \frac{2}{3}(4x_i - x_{i-1} - 2x_{i+1} + 1)^3 - \frac{2}{3}(2x_i - x_{i-1} - 2x_{i+1} + 1)^3 \\
&\qquad + \frac{4}{3}(3x_i - x_{i-1} - 2x_{i+1} + 1)^3 + (3x_i - x_{i-1} - 2x_{i+1} + 1)^2 \Bigg],
\end{aligned}
$$

where $x_0 = x_{n+1} = 0$, starting at $x_0 = (-1, -1, -1, -1, \ldots)$. Note that the second expressions for the ER and BT functions are in unary form.

The test results are summarized in Table 1. The quantities $N_i/N_f/N_{upd}$ in the first row of each cell of these tables are, respectively, the numbers of iterations, function evaluations, and rank-one updates performed by the algorithm. The number in parentheses in the second row of each cell is the CPU time in seconds. The table presents results for the extended Powell singular, the extended Rosenbrock, the extended Rosenbrock cliff, the variably dimensional, and the Broyden tridiagonal functions for n (the number of variables) equal to 40, 80, and 160. The column headings "PU1-1$'$" and "PU1-2$'$" refer, respectively, to the partial-update Newton method under Criteria 1$'$ and 2$'$ in Modification 1, while the headings "PU2-1$''$" and "PU2-2$''$" refer, respectively, to the method under Criteria 1$''$ and 2$''$ in Modification 2. In these methods we used $p = \lfloor \sqrt{n} \rfloor$, $\eta = 0.99$, and $\theta = 10^{-6}$. The last two columns, with the heading "Newton" and "p-Newton," give results for the modified Newton method of Gill and Murray [9] and the p-step modified Newton method of Traub [26] and Shamanskii [23], respectively, using the same termination criterion and linesearch as the other algorithms.

The test results for Modifications 1 and 2 were identical (except the CPU time) for problems EPS, ERC, and VD because they were convex. Also, due to the structure of the extended Powell singular function, starting at the chosen x_0, it is not difficult to see that, for $i = 2, \ldots, \frac{n}{4}$,

$$
\begin{aligned}
&\phi_{4i-3}(x_k) = \phi_1(x_k), \quad \phi_{4i-2}(x_k) = \phi_2(x_k), \\
&\phi_{4i-1}(x_k) = \phi_3(x_k), \quad \text{and} \quad \phi_{4i}(x_k) = \phi_4(x_k)
\end{aligned}
$$

at any iteration k, $k \geq 1$. Hence the number of rank-one updates in each iteration will be an integer multiple of $\frac{n}{4}$. The numbers N_{upd} associated with the extended Rosenbrock and the extended Rosenbrock cliff functions can be similarly explained.

These preliminary results show that although the partial update methods take more iterations and function evaluations than Gill and Murray's modified Newton method, partial update methods take less time to solve some types of problems than do modified Newton methods. In our test set this was true for the EPS, ERC, and VD sets of problems, all of which were convex. Also, method PU2-2$''$ took the least time to solve the largest incidence of problem BT.

4. Extension to partially separable and factorable optimization. The goal of this final section is to extend our partial-update Newton method to solve *partially separable* and *factorable* minimization problems. Partially separable problems are defined by Griewank and Toint [14], [15] as problems where the objective function has a decomposition of the form

$$
f(x) = \sum_{i=1}^{m} f_i(x), \qquad x \in \mathbf{R}^n, \tag{4.1}
$$

TABLE 1

Problem	n	PU1-1′	PU1-2′	PU2-1″	PU2-2″	Newton	p-Newton
(EPS)	40	6/17/40 (0.36)	7/36/20 (0.34)	6/17/40 (0.38)	7/36/20 (0.35)	7/14/0 (0.55)	9/44/0 (0.33)
	80	8/24/120 (2.35)	8/42/40 (1.68)	8/24/120 (2.84)	8/42/40 (1.83)	7/14/0 (2.99)	11/48/0 (1.89)
	160	8/24/240 (28.01)	10/46/80 (10.69)	8/24/240 (28.48)	10/46/80 (11.28)	7/14/0 (28.08)	16/58/0 (13.01)
(ER)	40	50/238/2540 (65.46)	78/288/880 (15.13)	20/40/1000 (11.43)	43/115/1000 (14.07)	11/27/0 (1.39)	61/134/0 (2.88)
	80	59/274/5640 (259.64)	94/350/2320 (109.09)	20/39/2080 (72.20)	50/118/2080 (79.98)	11/27/0 (13.49)	85/189/0 (20.65)
	160	53/244/10560 (1243.11)	98/342/4400 (590.62)	20/43/4800 (510.52)	52/124/4080 (472.33)	11/27/0 (69.27)	109/238/0 (82.13)
(ERC)	40	13/37/140 (1.10)	14/41/120 (0.99)	13/37/140 (1.12)	14/41/120 (1.10)	10/22/0 (1.12)	31/119/0 (1.79)
	80	13/43/240 (5.26)	15/44/240 (5.44)	13/43/240 (5.28)	15/44/240 (5.48)	10/22/0 (5.95)	34/135/0 (6.79)
	160	15/39/560 (40.15)	17/54/480 (36.36)	15/39/560 (41.26)	17/54/480 (37.42)	10/22/0 (47.68)	49/187/0 (43.36)
(VD)	40	11/33/4 (0.40)	16/47/4 (0.53)	11/33/4 (0.41)	16/47/4 (0.55)	11/19/0 (1.41)	13/67/0 (0.62)
	80	14/31/7 (1.81)	17/46/5 (1.93)	14/31/7 (1.85)	17/46/5 (2.00)	11/20/0 (18.24)	25/89/0 (3.59)
	160	20/37/9 (12.09)	24/48/8 (12.23)	20/37/9 (13.26)	24/48/8 (13.09)	12/22/0 (104.67)	62/169/0 (59.03)
(BT)	40	18/34/97 (1.71)	16/33/68 (1.49)	8/10/214 (3.84)	11/21/145 (2.06)	5/7/0 (0.36)	8/16/0 (0.31)
	80	15/29/176 (11.26)	18/32/110 (5.54)	11/20/327 (36.55)	14/26/280 (26.66)	5/7/0 (2.41)	10/21/0 (2.02)
	160	19/40/179 (46.35)	18/40/18 (19.65)	14/27/483 (98.13)	17/33/5 (16.51)	5/7/0 (27.81)	13/27/0 (17.48)

Numbers in cells are: $N_i/N_f/N_{upd}$—first row; (CPU secs.)—second row.

where each *element function* $f_i(\cdot)$ depends on only n_i variables, where n_i is small compared to n, the total number of variables of the problem. Partially separable problems arise naturally in many different fields, such as finite elements, variational calculations, and transportation networks (see [16] for more examples). Building approximations to the low-rank Hessian of each element function separately, Griewank and Toint [14], [15] developed partitioned variable metric update algorithms and obtained encouraging numerical results [16].

Assume that $f_i(x)$, $i = 1, \ldots, m$, in (4.1) are all twice continuously differentiable, and the gradient vector and the Hessian matrix of function (4.1) are

$$\nabla f(x) = \sum_{i=1}^{m} \nabla f_i(x) \quad \text{and} \quad \nabla^2 f(x) = \sum_{i=1}^{m} \nabla^2 f_i(x),$$

respectively. Note that each *element Hessian* $\nabla^2 f_i$ has nonzero entries in at most n_i rows and columns since element function f_i only depends on $n_i \ll n$ "internal" variables. We can rewrite $\nabla^2 f_i(x)$, which we shall also denote by $H_i(x)$, as

$$H_i(x) \equiv \nabla^2 f_i(x) = M_i G_i(x) M_i^T, \qquad i = 1, \ldots, m, \tag{4.2}$$

where $G_i(x)$ consists of the $n_i \times n_i$ nonzero submatrix of $\nabla^2 f_i(x)$ and M_i is an $n \times n_i$ matrix whose jth column is the qth column of the $n \times n$ identity matrix if x_q is the jth internal variable of f_i. For example, if $n = 4$ and $f_i(x)$ is a function of only x_1 and x_4, then

$$H_i(x) = \begin{pmatrix} h_{11}(x) & 0 & 0 & h_{14}(x) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ h_{41}(x) & 0 & 0 & h_{44}(x) \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} h_{11}(x) & h_{14}(x) \\ h_{41}(x) & h_{44}(x) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Factorable optimization problems are defined by McCormick [19] as problems where the objective function $f(x)$ is a *factorable function*, i.e., one that can be represented as the last in a finite sequence of functions $\{f_j(x)\}$ that are composed as follows:

(1°) for $j = 1, \ldots, n$, $f_j(x) = x_j$;

(2°) for $j > n$, $f_j(x)$ equals $f_k(x) + f_l(x)$, $f_k(x) \cdot f_l(x)$, or $T_j\left[f_k(x)\right]$, where $T_j(\cdot)$ is a function of a single variable, and $k, l < j$. It is quite easy to see that a unary function is a special case of a factorable function.

As pointed out by Jackson and McCormick [17], a factorable function possesses two properties that can be exploited to produce efficient algorithms: (1) its gradient and Hessian can be computed exactly (in terms of the derivatives of $T_j(\cdot)$), automatically, and efficiently if it is assumed to be twice continuously differentiable; (2) its Hessian is naturally given as a sum of *outer products* (*dyads*) of vectors, i.e.,

$$\nabla^2 f(x) = \sum_i \left[\alpha_i(x) u_i(x) v_i(x)^T + \alpha_i(x) v_i(x) u_i(x)^T\right], \tag{4.3}$$

where $\{u_i(x)\}$ and $\{v_i(x)\}$ are n vectors, and $\{\alpha_i(x)\}$ are scalars, which are all available, having been required for the computation of the gradient of $f(x)$. This dyadic structure of $\nabla^2 f(x)$ has been used by Emami [4] to obtain a factorization of a generalized inverse of the Hessian of a factorable function, by Ghotb [7] for computing the generalized inverse of a reduced Hessian, when it is given in dyadic form, and by Sofer [25] to obtain computationally efficient techniques for constructing the generalized inverse of such a reduced Hessian and updating it. Since

$$\alpha u v^T + \alpha v u^T = \frac{\alpha}{2}(u+v)(u+v)^T - \frac{\alpha}{2}(u-v)(u-v)^T,$$

(4.3) can be rewritten as

$$\nabla^2 f(x) = \sum_{i=1}^{m} H_i(x),$$

where $H_i(x) = \phi_i(x)a_i(x)a_i(x)^T$, $i = 1,\ldots,m$, are rank-one matrices, $a_i(x)$, $i = 1,\ldots,m$, are n vectors; and $\phi_i(x)$, $i = 1,\ldots,m$, are scalars, all of which are functions of x.

Thus, in both the partially separable and factorable cases, we can express the Hessian of $f(x)$ as

$$\nabla^2 f(x) = \sum_{i=1}^{m} H_i(x),$$

where each $H_i(x)$ has low rank. Because of this, it is possible to extend the partial-update Newton methods of the previous sections to solve partially separable and factorable optimization problems.

Such partial-update Newton methods for partially separable and factorable optimization compute a step direction by formula (1.5). But now the "working approximation" H^k to $\nabla^2 f(x_k)$ takes the form

$$H^k = \sum_{i=1}^{m} H_i^k,$$

where, for $i = 1,\ldots,m$, H_i^k is a "working approximation" to the element Hessian $H_i(x_k) \equiv \nabla^2 f_i(x_k)$ in the methods for partially separable optimization and to the rank-one matrix $H_i(x_k) \equiv \phi_i(x_k)a_i(x_k)a_i(x_k)^T$ in the methods for factorable optimization.

To be specific, unmodified versions of these methods initially set $H_i^0 = H_i(x_0)$, $i = 1,\ldots,m$, and at step k ($k \geq 1$) set

$$\text{(4.4)} \qquad H_i^k = \begin{cases} H_i^{k-1} & \text{if } H_i(x_k) \text{ is "replaceable" by } H_i^{k-1}, \\ H_i(x_k) & \text{otherwise,} \end{cases}$$

for $i = 1,\ldots,m$. Also, in analogy with the replacement criteria of §2, we have, substituting H for ϕ and the Frobenius norm $\|\cdot\|_F$ (or any other matrix norm) for the absolute value $|\cdot|$, the following.

Criterion 1*. For $i = 1,\ldots,m$,

$$H_i(x_k) \text{ is replaceable by } H_i^{k-1} \text{ if } \frac{\|H_i^{k-1} - H_i(x_k)\|_F}{\|H_i^{k-1} - H_i(x_k)\|_F + \|\nabla f(x_k)\|} < \eta,$$

where $0 < \eta < 1$.

Criterion 2*. For $i = 1,\ldots,m$,

$$H_i(x_k) \text{ is replceable by } H_i^{k-1} \text{ if } k \leq p \text{ or } \frac{\|H_i^{k-1} - H_i(x_k)\|_F}{\max_{k-p+1\leq j\leq k}\{\|H_i(x_k) - H_i(x_{j-1})\|_F\}} \leq 1,$$

where p is a given positive integer.

We note that because of the special form (4.2) of $H_i(x)$ in the partially separable case, H_i^k can be expressed as

$$H_i^k = M_i G_i^k M_i^T,$$

where G_i^k is an $n_i \times n_i$ dense matrix. Moreover, since $\|H_i^k - H_i(x_k)\|_F = \|G_i^{k-1} - G_i(x_k)\|_F$ and $\|H_i(x_k) - H_i(x_{j-1})\|_F = \|G_i(x_k) - G_i(x_{j-1})\|_F$, G's can be substituted for H's in the updating procedure (4.4) and in Criteria 1* and 2* in this case.

It is not very surprising that if we replace assumptions (A2)–(A4) by:

(A2′) $\nabla f_i(x)$, $i = 1, \ldots, m$, are all continuously differentiable in a neighborhood of x^* and $\nabla^2 f(x^*)$ is nonsingular; and

(A3′) $\nabla^2 f_i(x)$, $i = 1, \ldots, m$, are all Lipschitz continuous at x^*,

we can prove the following local convergence results using arguments analogous to those used in §2.

THEOREM 4.1. *Let $\{x_k\}$ be the sequence of iterates generated by the partial-update Newton method for partially separable or factorable optimization. Then*

(1) *$\{x_k\}$ is locally and linearly convergent to x^* under assumptions* (A1) *and* (A2′);

(2) *$\{x_k\}$ is locally quadratically convergent to x^*, if Criterion* 1^* *is used, under assumptions* (A1), (A2′), *and* (A3′);

(3) *$\{x_k\}$ is locally superlinearly convergent to x^*, if Criterion* 2^* *is used, under assumptions* (A1), (A2′), *and* (A3′). *Moreover, if p is finite, the rate of convergence is at least r_p, where r_p is the unique positive root of $t^{p+1} - t^p - 1 = 0$ and $1 < r_p < 2$, and on the average each iteration needs at most $\frac{m}{p}$ low-rank updates.*

As in the first part of §3 we can modify the partial-update Newton methods for partially separable and factorable problems to ensure global convergence to a stationary point of $f(x)$. These modifications, their implementation, and the results of numerical testing will be presented in a future report.

Acknowledgment. We thank two anonymous referees for their helpful suggestions.

REFERENCES

[1] R. H. BYRD, *Algorithms for robust regression*, in Nonlinear Optimization 1981, M. J. D. Powell, ed., Academic Press, New York, 1982, pp. 79–84.

[2] R. S. DEMBO, S. C. EISENSTAT, AND T. STEINHAUG, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[3] J. E. DENNIS AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, NJ, 1983.

[4] G. EMAMI, *Evaluating strategies for Newton's method using a numerically stable generalized inverse algorithm*, Ph.D. thesis, Dept. of Operations Research, George Washington Univ., Washington, DC, 1978.

[5] J. ERIKSSON, *A note on solution of large sparse maximum entropy problems with linear equality constraints*, Math. Programming, 18 (1980), pp. 146–154.

[6] D. M. GAY, *Some convergence properties of Broyden's method*, SIAM J. Numer. Anal., 16 (1979), pp. 623–630.

[7] F. GHOTB, *Newton's method for linearly constrained optimization problems*, Ph.D. thesis, Dept. of Operations Research, George Washington Univ., Washington, DC, 1980.

[8] P. E. GILL, G. H. GOLUB, W. MURRAY, AND M. A. SAUNDERS, *Methods for modifying matrix factorizations*, Math. Comp., 28 (1974), pp. 505–535.

[9] P. E. GILL, W. MURRAY, AND M. H. WRIGHT, *Practical Optimization*, Academic Press, London, 1981.

[10] D. GOLDFARB, *Factorized variable metric methods for unconstrained optimization*, Math. Comp., 30 (1976), pp. 796–811.

[11] ———, *Curvilinear path steplength algorithms for minimization which use directions of negative curvature*, Math. Programming, 18 (1980), pp. 31–40.

[12] D. GOLDFARB AND S. LIU, *An $O(n^3L)$ primal interior point algorithm for convex quadratic programming*, Math. Programming, 49 (1991), pp. 325–340.

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

[14] A. GRIEWANK AND PH. L. TOINT, *On the unconstrained optimization of partially separable functions*, in Nonlinear Optimization 1981, M. J. D. Powell, ed., Academic Press, New York, 1982, pp. 301–312.

[15] ———, *Partitioned variable metric updates for large structured optimization problems*, Numer. Math., 39 (1982), pp. 119–137.

[16] A. GRIEWANK AND PH. L. TOINT, *Numerical experiments with partially separable optimization problems*, in Numerical Analysis: Proceedings Dundee 1983, D. F. Griffiths, ed., Lecture Notes in Math. 1066, Springer-Verlag, Berlin, 1984, pp. 203–220.

[17] R. H. F. JACKSON AND G. P. MCCORMICK, *The polyadic structure of factorable function tensors with applications to high-order minimization techniques*, J. Optim. Theory Appl., 51 (1986), pp. 63–94.

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

[19] G. P. MCCORMICK, *Nonlinear Programming: Theory, Algorithm, and Applications*, John Wiley, New York, 1983.

[20] G. P. MCCORMICK AND A. SOFER, *Optimization with unary functions*, Math. Programming, 52 (1991), pp. 167–178.

[21] J. J. MORÉ AND D. C. SORENSEN, *Newton's method*, in Studies in Numerical Analysis, G. H. Golub, ed., MAA Studies in Math., 24 (1982), pp. 29–82.

[22] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[23] V. E. SHAMANSKII, *On a modification of the Newton method*, Ukrain. Mat. Zh., 19 (1967), pp. 133–138. (In Russian.)

[24] J. SHERMAN AND W. J. MORRISON, *Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix*, Ann. Math. Statist., 20 (1949), pp. 621.

[25] A. SOFER, *Computationally efficient techniques for generalized inversion in nonlinear programming*, Ph.D. thesis, Dept. of Operations Research, George Washington Univ., Washington, DC, 1983.

[26] J. TRAUB, *Iterative Methods for the Solution of Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1964.

[27] Y. YE, *A $O(n^3L)$ potential reduction algorithm for linear programming*, Math. Programming, 50 (1991), pp. 239–258.

SIAM J. OPTIMIZATION
Vol. 3, No. 2, pp. 398–412, May 199З

LARGE-STEP INTERIOR POINT ALGORITHMS FOR LINEAR COMPLEMENTARITY PROBLEMS*

MASAKAZU KOJIMA[†], YOSHIFUMI KURITA[‡], AND SHINJI MIZUNO[§]

Abstract. Recently Kojima, Megiddo, and Mizuno showed theoretical convergence of primal-dual interior point algorithms with the use of new step length rules for linear programs. Their rules, which only rely on the lengths of steps from the current iterates in the primal and dual spaces to the respective boundaries of the primal and dual feasible regions, allow taking large step lengths without performing any line search. This paper extends and modifies their analysis to interior point algorithms for positive semidefinite linear complementarity problems. Global convergence and polynomial-time convergence are presented under similar step length rules.

Key words. interior point algorithm, linear complementarity problem, linear programming, large step, long step, global convergence

AMS subject classification. 90C33

1. Introduction. Many interior point algorithms have been developed for the positive semidefinite linear complementarity problem (LCP). They work as a primal-dual interior point algorithm when they are applied to a linear program and a quadratic program. Among others, this paper is concerned with a class of interior point algorithms (see [3], [7], [8], [15], [22], [24], [27], etc.) characterized by

- extensions of the primal-dual interior point algorithm originating from a fundamental analysis by Megiddo [12] on the central trajectory, leading to optimal solutions of the standard-form linear program and its dual,
- moving in Newton direction towards the central trajectory at each iteration.

Kojima, Megiddo, Noma, and Yoshise [3] studied a unified approach to this class of algorithms, which was suggested by Kojima, Mizuno, and Yoshise [8]. See Kojima, Megiddo, and Ye [4]; Mizuno [14]; and Todd [21] for other types of interior point algorithms for the LCP. We remark that the projected scaled steepest descent algorithm given for the positive semidefinite LCP in [21] can also be regarded as an extension of the primal-dual interior point algorithm for linear programs.

The first polynomial-time primal-dual interior point algorithm was given by Kojima, Mizuno and Yoshise [6]. Their algorithm solves the standard-form linear program and its dual simultaneously in $O(nL)$ iterations. Soon after, the theoretical computational complexity $O(nL)$ was improved to $(\sqrt{n}L)$, and the primal-dual algorithm was extended to interior point algorithms for the positive semidefinite LCP and a convex quadratic program in [7], [17], and [18]. Independently, Tanabe [19], [20] proposed similar algorithms, which he called a centered Newton method.

In interior point algorithms, certain neighborhoods of the central trajectory and (primal-dual) potential functions (Todd and Ye [22]) have been major tools to determine step lengths ensuring global and/or polynomial-time convergence. This paper presents new step length rules which rely on neither of these tools but only the length of the step from the current iterate to the boundary of the feasible region of the LCP. The new

*Received by the editors July 1, 1991; accepted for publication (in revised form) March 5, 1992.

[†] Department of Information Sciences, Tokyo Institute of Technology, Oh-okayama, Meguro-ku, Tokyo 152, Japan.

[‡] Department of Systems Science, Tokyo Institute of Technology, Oh-okayama, Meguro-ku, Tokyo 152, Japan.

[§] Department of Prediction and Control, The Institute of Statistical Mathematics, 4-6-7 Minami-Azabu, Minato-ku, Tokyo 106, Japan.

rules are extensions and modifications of the step length rules that Kojima, Megiddo, and Mizuno [1] recently proposed for primal-dual interior point algorithms for linear programs.

2. Main results. The main results of this paper are founded on [1] where the original rules were given. We omit the proofs of some of the lemmas which we can derive by the same arguments as the ones used in [1]. But there are some substantial differences between interior point algorithms for linear programs and the positive semidefinite LCP, which require some additional analysis. We outline the interior point algorithms for the LCP, and then make the differences clear.

Let $\boldsymbol{M} \in R^{n\times n}$ and $\boldsymbol{q} \in R^n$. The linear complementarity problem (LCP) is the problem of finding a point $(\boldsymbol{x}, \boldsymbol{z}) \in R^{2n}$ such that

$$\boldsymbol{z} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q}, \quad \boldsymbol{X}\boldsymbol{z} = \boldsymbol{0}, \quad (\boldsymbol{x}, \boldsymbol{z}) \geq \boldsymbol{0},$$

where $\boldsymbol{X} = \operatorname{diag}(\boldsymbol{x}) \in R^{n\times n}$ denotes a diagonal matrix with the coordinates of a vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T \in R^n$. The equality $\boldsymbol{X}\boldsymbol{z} = \boldsymbol{0}$ is rewritten componentwise as $x_j z_j = 0$ for $j = 1, 2, \ldots, n$, which we call the complementarity condition. Define

$$\begin{aligned} S &= \{(\boldsymbol{x}, \boldsymbol{z}) \geq \boldsymbol{0} : \boldsymbol{z} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q}\}, \\ S_{++} &= \{(\boldsymbol{x}, \boldsymbol{z}) \in S : (\boldsymbol{x}, \boldsymbol{z}) > \boldsymbol{0}\}. \end{aligned}$$

We call S and S_{++} the feasible region of the LCP and its interior, respectively. We may state that the LCP is the problem of minimizing the total complementarity $\boldsymbol{x}^T\boldsymbol{z}$ over the feasible region S; if the minimum total complementarity attains zero, then the LCP has a solution. Throughout the paper we assume that

(a) a point $(\boldsymbol{x}^0, \boldsymbol{z}^0) \in S_{++}$ is known in advance,

(b) the matrix $\boldsymbol{M}$ is positive semidefinite, i.e., $\boldsymbol{x}^T\boldsymbol{M}\boldsymbol{x} \geq 0$ for every $\boldsymbol{x} \in R^n$.

We define the central trajectory S_{cen} for the LCP as the set of solutions $(\boldsymbol{x}(\mu), \boldsymbol{z}(\mu))$ to the system of equations with a parameter $\mu > 0$:

$$\boldsymbol{z} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q}, \quad \boldsymbol{X}\boldsymbol{z} = \mu\boldsymbol{e}, \quad \text{and} \quad (\boldsymbol{x}, \boldsymbol{z}) \geq \boldsymbol{0}. \tag{1}$$

Here $\boldsymbol{e}$ denotes the n-dimensional vector of ones. It is well known that the central trajectory S_{cen} converges to a solution of the LCP under the assumptions (a) and (b) as $\mu > 0$ approaches zero. See Theorem 4.1 of [5]. Assuming that we have obtained an interior point $(\boldsymbol{x}^k, \boldsymbol{z}^k)$ of the LCP at the beginning of the kth iteration, we now show how we compute a direction $(\boldsymbol{\Delta x}, \boldsymbol{\Delta z})$ to generate a new interior point $(\boldsymbol{x}^{k+1}, \boldsymbol{z}^{k+1})$ of the LCP. Define $f^k = (\boldsymbol{x}^k)^T\boldsymbol{z}^k/n$. Let $0 \leq \beta \leq 1$. We call β a search direction parameter. We compute a feasible direction $(\boldsymbol{\Delta x}, \boldsymbol{\Delta z})$ by applying Newton method at $(\boldsymbol{x}^k, \boldsymbol{z}^k)$ for finding a point $(\boldsymbol{x}(\mu), \boldsymbol{z}(\mu))$, with some $\mu = \beta f^k$, $\beta \in [0,\ 1]$, on the central trajectory S_{cen}. In other words, we solve the system of linear equations

$$\begin{aligned} \boldsymbol{Z}^k\boldsymbol{\Delta x} + \boldsymbol{X}^k\boldsymbol{\Delta z} &= \beta f^k\boldsymbol{e} - \boldsymbol{X}^k\boldsymbol{z}^k, \\ \boldsymbol{\Delta z} &= \boldsymbol{M}\boldsymbol{\Delta x} \end{aligned} \tag{2}$$

to get $(\boldsymbol{\Delta x}, \boldsymbol{\Delta z})$. The feasible direction satisfies

$$(\boldsymbol{z}^k)^T\boldsymbol{\Delta x} + (\boldsymbol{x}^k)^T\boldsymbol{\Delta z} = -(1-\beta)(\boldsymbol{x}^k)^T\boldsymbol{z}^k. \tag{3}$$

Now we consider the pair of the standard form LP and its dual:

$$\begin{aligned}
\text{(P)}\quad & \text{Minimize} && \boldsymbol{c}^T\boldsymbol{x}\\
& \text{subject to} && \boldsymbol{x} \in P = \{\boldsymbol{x} \;:\; \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq \boldsymbol{0}\}.\\
\text{(D)}\quad & \text{Maximize} && \boldsymbol{b}^T\boldsymbol{y}\\
& \text{subject to} && (\boldsymbol{y}, \boldsymbol{z}) \in D = \{(\boldsymbol{y}, \boldsymbol{z}) \;:\; \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{z} = \boldsymbol{c}, \boldsymbol{z} \geq \boldsymbol{0}\}.
\end{aligned}$$

Here $\boldsymbol{A} \in R^{m\times n}$, $\boldsymbol{c} \in R^n$, $\boldsymbol{b} \in R^m$, $\boldsymbol{x} \in R^n$, $\boldsymbol{y} \in R^m$, and $\boldsymbol{z} \in R^n$. In this case, we get a feasible direction $(\boldsymbol{\Delta x}, \boldsymbol{\Delta y}, \boldsymbol{\Delta z})$ at a kth iterate $(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k)$ by solving the system of linear equations

$$\begin{aligned}
\boldsymbol{Z}^k\boldsymbol{\Delta x} + \boldsymbol{X}^k\boldsymbol{\Delta z} &= \beta f^k\boldsymbol{e} - \boldsymbol{X}^k\boldsymbol{z}^k,\\
\boldsymbol{A}\boldsymbol{\Delta x} &= \boldsymbol{0},\\
\boldsymbol{A}^T\boldsymbol{\Delta y} + \boldsymbol{\Delta z} &= \boldsymbol{0}
\end{aligned}\tag{4}$$

(see [1]).

We now state some differences between interior point algorithms for the LCP and the LP.

(i) In the LCP case, we need to take a single step length $\alpha = \alpha^k$ over the entire space and a new iterate $(\boldsymbol{x}^{k+1}, \boldsymbol{z}^{k+1})$ such that

$$0 < \alpha < \alpha^k_{bd}, \qquad (\boldsymbol{x}^{k+1}, \boldsymbol{z}^{k+1}) = (\boldsymbol{x}^k, \boldsymbol{z}^k) + \alpha(\boldsymbol{\Delta x}, \boldsymbol{\Delta z}),\tag{5}$$

where

$$\alpha^k_{bd} = \max\{\alpha : \boldsymbol{x}^k + \alpha\boldsymbol{\Delta x} \geq \boldsymbol{0},\ \boldsymbol{z}^k + \alpha\boldsymbol{\Delta z} \geq \boldsymbol{0}\}.\tag{6}$$

In the LP case, we can take distinct step lengths $\alpha_p = \alpha^k_p$ in the primal space, $\alpha_d = \alpha^k_d$ in the dual space, and a new iterate $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{z}^{k+1})$ such that

$$\begin{aligned}
&0 \leq \alpha_p < \hat{\alpha}^k_p,\ 0 \leq \alpha_d < \hat{\alpha}^k_d,\\
&\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_p\boldsymbol{\Delta x}, \qquad (\boldsymbol{y}^{k+1}, \boldsymbol{z}^{k+1}) = (\boldsymbol{y}^k, \boldsymbol{z}^k) + \alpha_d(\boldsymbol{\Delta y}, \boldsymbol{\Delta z}),
\end{aligned}\tag{7}$$

where $\hat{\alpha}^k_p = \max\{\alpha : \boldsymbol{x}^k + \alpha\boldsymbol{\Delta x} \geq \boldsymbol{0}\}$ and $\hat{\alpha}^k_d = \max\{\alpha : \boldsymbol{z}^k + \alpha\boldsymbol{\Delta z} \geq \boldsymbol{0}\}$.

(ii) The directions $\boldsymbol{\Delta x}$ and $\boldsymbol{\Delta z}$ for the LCP form an acute angle, i.e.,

$$\boldsymbol{\Delta x}^T\boldsymbol{\Delta z} = \boldsymbol{\Delta x}^T\boldsymbol{M}\boldsymbol{\Delta x} \geq 0,$$

while the directions $\boldsymbol{\Delta x}$ and $\boldsymbol{\Delta z}$ for the LP are orthogonal, i.e.,

$$\boldsymbol{\Delta x}^T\boldsymbol{\Delta z} = -\boldsymbol{\Delta y}^T\boldsymbol{A}\boldsymbol{\Delta x} = 0.$$

Most of the existing theoretical primal-dual interior point algorithms for linear programs choose a single step length, which is usually much smaller than the distinct step lengths employed in the practical implementation in [10], [11], and [13], to ensure global and/or polynomial-time convergence. They often utilize certain neighborhoods of the central trajectory and/or primal-dual potential function when they determine a single step length. It was the aim of the paper [1] by Kojima, Megiddo, and Mizuno to fill these gaps between the practically efficient step lengths and the theoretical step lengths ensuring global and/or polynomial-time convergence. They proposed two sets of step length rules, Rule G and Rule P, which utilize no neighborhoods of the central trajectory, no potential function, and no line search, but only $\hat{\alpha}_p^k$ and $\hat{\alpha}_d^k$. Rule G ensures the global convergence, while Rule P ensures the $O(nL)$ iteration polynomial-time computational complexity. These two rules are extended and modified to Rules G$'$ and P$'$.

Now we focus on the second difference (ii). Let $0 \leq \beta = \beta^k < 1$. Suppose that we take a common step length $\alpha = \alpha^k = \alpha_p^k = \alpha_d^k$ in the primal and dual spaces of the LP. Then

$$(\boldsymbol{x}^{k+1})^T \boldsymbol{z}^{k+1} = (\boldsymbol{x}^k)^T \boldsymbol{z}^k - \alpha(1-\beta^k)(\boldsymbol{x}^k)^T \boldsymbol{z}^k. \tag{8}$$

In the LCP case, we observe that

$$(\boldsymbol{x}^{k+1})^T \boldsymbol{z}^{k+1} = (\boldsymbol{x}^k)^T \boldsymbol{z}^k - \alpha(1-\beta^k)(\boldsymbol{x}^k)^T \boldsymbol{z}^k + \alpha^2 \boldsymbol{\Delta x}^T \boldsymbol{\Delta z}. \tag{9}$$

In both cases, we may view the total complementarity $(\boldsymbol{x}^{k+1})^T \boldsymbol{z}^{k+1}$ at the new iterate $(\boldsymbol{x}^{k+1}, \boldsymbol{z}^{k+1})$ as a function of the step length α. In the LP case, it changes linearly with α while it changes quadratically in the LCP case. In both cases, the coefficient $-(1-\beta^k)(\boldsymbol{x}^k)^T \boldsymbol{z}^k$ of the linear term is strictly negative. The quadratic term in (9) makes an essential difference between the step length rules [1] for the LP and their extensions and modifications to the LCP, which we present in this paper. Under the assumption (b), the quadratic term $\alpha^2 \boldsymbol{\Delta x}^T \boldsymbol{\Delta z} = \alpha^2 \boldsymbol{\Delta x}^T \boldsymbol{M} \boldsymbol{\Delta x}$ is always nonnegative. Therefore, the total complementarity $(\boldsymbol{x}^{k+1})^T \boldsymbol{z}^{k+1}$ for the LCP is not monotone decreasing with respect to α if $\boldsymbol{\Delta x}^T \boldsymbol{\Delta z} > 0$. Let

$$\alpha_{\min}^k = \begin{cases} \dfrac{(1-\beta^k)(\boldsymbol{x}^k)^T \boldsymbol{z}^k}{2\boldsymbol{\Delta x}^T \boldsymbol{\Delta z}} & \text{if } \boldsymbol{\Delta x}^T \boldsymbol{\Delta z} > 0, \\ +\infty & \text{otherwise.} \end{cases} \tag{10}$$

Then the total complementarity attains its minimum at the step length $\alpha_{\min}^k$ when $\boldsymbol{\Delta x}^T \boldsymbol{\Delta z} > 0$. Thus, in the LCP case, it is reasonable to choose a step length $\alpha = \alpha^k$ satisfying $0 < \alpha < \alpha_{bd}^k$ and $\alpha \leq \alpha_{\min}^k$. It should be noted that α_{bd}^k is always finite under the assumptions (a) and (b) (since $(\boldsymbol{\Delta x}, \boldsymbol{\Delta z}) \not\geq \boldsymbol{0}$ by (3)) while $\alpha_{\min}^k$ can be $+\infty$.

A generic interior point algorithm is summarized as follows.

ALGORITHM. Let $(\boldsymbol{x}^0, \boldsymbol{z}^0) > 0$ be an initial feasible solution for the LCP, i.e., $\boldsymbol{z}^0 = \boldsymbol{M}\boldsymbol{x}^0 + \boldsymbol{q}$, $(\boldsymbol{x}^0, \boldsymbol{z}^0) > \boldsymbol{0}$.
Step 0. Let $k = 0$.
Step 1. Choose a $\beta = \beta^k \in [0, 1]$, and compute $(\boldsymbol{\Delta x}, \boldsymbol{\Delta z})$ by solving the system (2).
Step 2. Choose a step length $\alpha = \alpha^k$ and compute a new point $(\boldsymbol{x}^{k+1}, \boldsymbol{z}^{k+1})$ by (5).
Step 3. Let $k = k + 1$ and go to Step 1.

Now we are ready to describe two rules, Rules G$'$ and P$'$, which are extensions and modifications of Rules G and P in [1], respectively, for controlling the parameters α and β in the Algorithm.

Rule G$'$. Let $0 \leq \bar{\beta} < 1$, $0 < \bar{\theta} < 1$, and $0 < \alpha^*$ be fixed. At each iteration of the Algorithm, choose a search direction parameter $\beta = \beta^k \in [0, \bar{\beta}]$ and a step length $\alpha = \alpha^k = \min\{\alpha', \alpha_{\min}^k\}$, where

$$\begin{aligned} &\alpha' \in [\bar{\theta}\alpha^*, \alpha_{bd}^k) && \text{if } \alpha_{bd}^k \geq \alpha^*, \\ &\alpha' \in [\bar{\theta}{\alpha_{bd}^k}^2/\alpha^*, {\alpha_{bd}^k}^2/\alpha^*) && \text{if } \alpha_{bd}^k < \alpha^*. \end{aligned}$$

Rule P$'$. Let $0 < \beta^* \leq \bar{\beta} = \frac{1}{2}, \bar{\theta} = \frac{3}{4} \leq \theta^* < 1$, and $0 < \alpha^* \leq 1$ be fixed. At each iteration of the Algorithm, choose a search direction parameter $\beta = \beta^k \in [\beta^*, \bar{\beta}]$ and a step length $\alpha = \alpha^k = \min\{\alpha', \alpha^k_{\min}\}$, where

$$\begin{array}{ll} \alpha' \in [\bar{\theta}\alpha^*, \theta^*\alpha^k_{bd}] & \text{if } \alpha^k_{bd} \geq \alpha^*, \\ \alpha' \in [\bar{\theta}{\alpha^k_{bd}}^2/\alpha^*, \theta^*{\alpha^k_{bd}}^2/\alpha^*] & \text{if } \alpha^k_{bd} < \alpha^*. \end{array}$$

Obviously, Rule P$'$ is a special case of Rule G$'$. In either of the rules, the step length $\alpha = \alpha^k$ satisfies

$$(11) \qquad 0 < \min\left\{\bar{\theta}\alpha^*, \frac{\bar{\theta}{\alpha^k_{bd}}^2}{\alpha^*}, \alpha^k_{\min}\right\} \leq \alpha < \alpha^k_{bd} \quad \text{and} \quad \alpha \leq \alpha^k_{\min}.$$

Hence the Algorithm using either of them generates a sequence $\{(\boldsymbol{x}^k, \boldsymbol{z}^k)\}$ in the interior S_{++} of the feasible region S such that $(\boldsymbol{x}^{k+1})^T\boldsymbol{z}^{k+1} < (\boldsymbol{x}^k)^T\boldsymbol{z}^k$.

As in [1], we utilize a quantity π^k to measure a deviation from the central trajectory S_{cen} at the current iterate $(\boldsymbol{x}^k, \boldsymbol{z}^k) \in S_{++}$:

$$\pi^k = \min\left\{\frac{x^k_j z^k_j}{f^k} : j = 1, 2, \ldots, n\right\}.$$

Obviously, $0 < \pi^k \leq 1$.

We establish the following two theorems. Their proofs are given in §§5 and 6, respectively.

THEOREM 2.1. *The Algorithm using Rule G$'$ generates a bounded sequence* $\{(\boldsymbol{x}^k, \boldsymbol{z}^k)\}$ *such that* $\lim_{k\to\infty}(\boldsymbol{x}^k)^T\boldsymbol{z}^k = 0$.

THEOREM 2.2. *At each iteration of the Algorithm using Rule P$'$, we have*

$$(12) \qquad (\boldsymbol{x}^{k+1})^T\boldsymbol{z}^{k+1} \leq \left(1 - \frac{3\sigma^2}{8n\alpha^*}\right)(\boldsymbol{x}^k)^T\boldsymbol{z}^k,$$

where

$$(13) \qquad \sigma = \min\left\{\pi^0, \frac{(1-\theta^*)\beta^*\alpha^*}{2}, (1-\theta^*)^2\beta^*\right\}.$$

So far we have assumed that the LCP has a known initial interior point $(\boldsymbol{x}^0, \boldsymbol{z}^0) \in S_{++}$ from which the Algorithm starts. Theoretically, we can embed the LCP to be solved into an equivalent artificial linear complementarity problem having a known interior feasible solution with the total complementarity of order $2^{O(L)}$ from which the Algorithm starts, where L denotes the size of the original LCP to be solved. Theorem 2.2 guarantees that the Algorithm then generates in $O(nL)$ iterations an approximate solution, with the total complementarity less than 2^{-2L}, of the artificial problem, from which we can compute an exact solution of the LCP in $O(n^3)$ arithmetic operations or we can determine that the LCP has no solution. Thus the Algorithm using Rule P$'$ solves the LCP in $O(nL)$ iterations. See [3] and [7] for more details. Practically, however, this approach has the disadvantage that the artificial problem involves a very large number, called the big $\mathcal{M}$, which may cause numerical instability and/or computational inefficiency. To overcome such a difficulty, Kojima, Mizuno, and Yoshise [9] recently proposed a method according to which we can update $\mathcal{M}$ during the iterations of interior point algorithms even if we start with a relatively small $\mathcal{M}$. Their method can be easily incorporated into the Algorithm using Rule G′ without destroying the global convergence (Theorem 2.1).

The rest of the paper is devoted to the proofs of Theorems 2.1 and 2.2. In §3, we list mathematical symbols and notation which are used throughout the paper. Section 4 presents some basic lemmas. The proofs of Theorems 2.1 and 2.2 are given in §§5 and 6, respectively.

3. Notation.

$\boldsymbol{e} = (1, 1, \dots, 1)^T \in R^n$.
$S = \{(\boldsymbol{x}, \boldsymbol{z}) \geq \boldsymbol{0} : \boldsymbol{z} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q}\}$: the feasible region of the LCP.
$S_{++} = \{(\boldsymbol{x}, \boldsymbol{z}) > \boldsymbol{0} : \boldsymbol{z} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{q}\}$: the interior of S.
$S_{\text{cen}} = \{(\boldsymbol{x}, \boldsymbol{z}) \in S_{++} : \boldsymbol{X}\boldsymbol{z} = \mu\boldsymbol{e}$ for some $\mu > 0\}$: the central trajectory.
$(\boldsymbol{x}^k, \boldsymbol{z}^k)$: the kth iterate of the Algorithm.
$f^k = \dfrac{(\boldsymbol{x}^k)^T\boldsymbol{z}^k}{n}$.
$(\boldsymbol{\Delta x}, \boldsymbol{\Delta z})$: the search direction at the kth iterate.
$(\boldsymbol{x}^{k+1}, \boldsymbol{z}^{k+1}) = (\boldsymbol{x}^k, \boldsymbol{z}^k) + \alpha(\boldsymbol{\Delta x}, \boldsymbol{\Delta z})$: the $(k+1)$th iterate.
$\alpha = \alpha^k$: a step length at the kth iteration.
$\beta = \beta^k$: a direction parameter at the kth iteration.
$\alpha^k_{bd} = \max\{\alpha : (\boldsymbol{x}^k, \boldsymbol{z}^k) + \alpha(\boldsymbol{\Delta x}, \boldsymbol{\Delta z}) \geq \boldsymbol{0}\}$.

$$\alpha^k_{\min} = \begin{cases} \dfrac{(1-\beta^k)(\boldsymbol{x}^k)^T\boldsymbol{z}^k}{2\boldsymbol{\Delta x}^T\boldsymbol{\Delta z}} & \text{if } \boldsymbol{\Delta x}^T\boldsymbol{\Delta z} > 0, \\ +\infty & \text{otherwise.} \end{cases}$$

$\bar{\beta}$, $\bar{\theta}$, α^*, β^*, θ^* : constants fixed in Rules G′ and P′.

$$\pi^k = \min\left\{ \frac{x^k_j z^k_j}{f^k} : j = 1, 2, \dots, n \right\}.$$

The search direction $(\boldsymbol{\Delta x}, \boldsymbol{\Delta z})$ depends on the kth iterate $(\boldsymbol{x}^k, \boldsymbol{z}^k)$, but we omit its dependence on k.

4. Lemmas. In this section, we prove some lemmas which are used for the proofs of Theorems 2.1 and 2.2. Throughout this section, we assume that $0 \leq \beta = \beta^k < 1$ and $0 < \alpha = \alpha^k < \alpha^k_{bd}$. In the LP case, we have the equality $(\boldsymbol{x}^{k+1})^T\boldsymbol{z}^{k+1} = (1 - \alpha(1-\beta))(\boldsymbol{x}^k)^T\boldsymbol{z}^k$, as we have observed in (8). This equality plays an essential role in [1]. However, the equality is no longer valid for the LCP case because of the quadratic term $\alpha^2\boldsymbol{\Delta x}^T\boldsymbol{\Delta z}$ in the step length α appearing in (9). Instead, we have the following lemma.

LEMMA 4.1. *Assume that* $0 < \alpha = \alpha^k \leq \min\{\alpha^k_{\min}, \alpha^k_{bd}\}$. *Then*

$$0 \leq (1 - \alpha(1-\beta))(\boldsymbol{x}^k)^T\boldsymbol{z}^k \leq (\boldsymbol{x}^{k+1})^T\boldsymbol{z}^{k+1} \leq \left(1 - \frac{\alpha(1-\beta)}{2}\right)(\boldsymbol{x}^k)^T\boldsymbol{z}^k,$$
$$0 \leq f^{k+1} \leq \left(1 - \frac{\alpha(1-\beta)}{2}\right) f^k,$$
$$0 \leq 1 - \frac{\alpha(1-\beta)}{2} \leq 1.$$

Proof. It suffices to show the inequalities

$$(1 - \alpha(1-\beta))(\boldsymbol{x}^k)^T\boldsymbol{z}^k \leq (\boldsymbol{x}^{k+1})^T\boldsymbol{z}^{k+1} \leq \left(1 - \frac{\alpha(1-\beta)}{2}\right)(\boldsymbol{x}^k)^T\boldsymbol{z}^k,$$

because all other inequalities follow directly from the constructions of $(\boldsymbol{x}^{k+1}, \boldsymbol{z}^{k+1})$ and f^{k+1}. If $\boldsymbol{\Delta x}^T \boldsymbol{\Delta z} = 0$ then we easily obtain the desired inequalities from (9). Now suppose that $\boldsymbol{\Delta x}^T \boldsymbol{\Delta z} > 0$. Then $\alpha^k_{\min} = (1-\beta)(\boldsymbol{x}^k)^T \boldsymbol{z}^k / (2\boldsymbol{\Delta x}^T \boldsymbol{\Delta z})$. Hence we see from the equality (9) that

$$\begin{aligned}
(1-\alpha(1-\beta))(\boldsymbol{x}^k)^T \boldsymbol{z}^k &\leq (\boldsymbol{x}^{k+1})^T \boldsymbol{z}^{k+1} \\
&= (1-\alpha(1-\beta))(\boldsymbol{x}^k)^T \boldsymbol{z}^k + \alpha^2 \boldsymbol{\Delta x}^T \boldsymbol{\Delta z} \\
&\leq (1-\alpha(1-\beta))(\boldsymbol{x}^k)^T \boldsymbol{z}^k + \alpha \alpha^k_{\min} \boldsymbol{\Delta x}^T \boldsymbol{\Delta z} \\
&\leq (1-\alpha(1-\beta))(\boldsymbol{x}^k)^T \boldsymbol{z}^k + \alpha(1-\beta)(\boldsymbol{x}^k)^T \boldsymbol{z}^k / 2 \\
&= \left(1 - \frac{\alpha(1-\beta)}{2}\right)(\boldsymbol{x}^k)^T \boldsymbol{z}^k. \qquad \square
\end{aligned}$$

The lemma below can be proved in the same way as Lemma 3.3 of [1], and the proof is omitted.

LEMMA 4.2.

$$(\alpha^k_{bd})^2 \geq \min\left\{\frac{1}{4}, \frac{2(\pi^k)^2}{(\beta^2 - 2\pi^k\beta + \pi^k)n}\right\} \geq \min\left\{\frac{1}{4}, \frac{2(\pi^k)^2}{n}\right\}.$$

LEMMA 4.3.

$$\alpha^k_{\min} \geq \frac{2(1-\beta)\pi^k}{\beta^2 - 2\pi^k\beta + \pi^k} \geq 2(1-\bar{\beta})\pi^k.$$

Proof. By using the same argument as in the proof of Lemma 3.3 of [1], we have

$$\boldsymbol{\Delta x}^T \boldsymbol{\Delta z} \leq \frac{nf^k}{4\pi^k}(\beta^2 - 2\pi^k\beta + \pi^k).$$

Hence

$$\begin{aligned}
\alpha^k_{\min} &= \frac{(1-\beta)(\boldsymbol{x}^k)^T \boldsymbol{z}^k}{2\boldsymbol{\Delta x}^T \boldsymbol{\Delta z}} \\
&\geq \frac{(1-\beta)(\boldsymbol{x}^k)^T \boldsymbol{z}^k}{2} \cdot \frac{4\pi^k}{nf^k(\beta^2 - 2\pi^k\beta + \pi^k)} \\
&= \frac{2(1-\beta)\pi^k}{\beta^2 - 2\pi^k\beta + \pi^k} \quad (\text{since } (\boldsymbol{x}^k)^T \boldsymbol{z}^k = nf^k).
\end{aligned}$$

Thus the first inequality of the lemma has been shown. The second inequality follows from $0 < \beta^2 - 2\beta\pi^k + \pi^k < 1$, which is proved in [1]. □

LEMMA 4.4. *Let* $j \in \{1, 2, \ldots, n\}$. *If*

$$1 - \alpha - (1 - \alpha^k_{bd})\left(\frac{\alpha}{\alpha^k_{bd}}\right)^2 \geq 0, \tag{14}$$

then

$$x^{k+1}_j z^{k+1}_j \geq \left(1 - \alpha - (1-\alpha^k_{bd})\left(\frac{\alpha}{\alpha^k_{bd}}\right)^2\right)\pi^k f^k + \left(\alpha - \alpha^k_{bd}\left(\frac{\alpha}{\alpha^k_{bd}}\right)^2\right)\beta f^k.$$

Otherwise,

$$x_j^{k+1}z_j^{k+1} \geq \left(1 - \frac{\alpha}{\alpha_{bd}^k}\right)^2 \beta f^k.$$

Proof. By using the same argument as in the proof of Lemma 3.4 of [1], we have

$$x_j^{k+1}z_j^{k+1} \geq \left(1 - \alpha - (1-\alpha_{bd}^k)\left(\frac{\alpha}{\alpha_{bd}^k}\right)^2\right)x_j^kz_j^k + \left(\alpha - \alpha_{bd}^k\left(\frac{\alpha}{\alpha_{bd}^k}\right)^2\right)\beta f^k.$$

It follows from the definition of π^k that $x_j^kz_j^k \geq \pi^k f^k$. Replacing $x_j^kz_j^k$ by $\pi^k f^k$ in the above inequality, we obtain the former assertion of the lemma. Now we deal with the case

$$1 - \alpha - (1-\alpha_{bd}^k)\left(\frac{\alpha}{\alpha_{bd}^k}\right)^2 < 0.$$

If $x_j^kz_j^k \leq \beta f^k$, the inequalities above imply

$$\begin{aligned} x_j^{k+1}z_j^{k+1} &\geq \left(1 - \alpha - (1-\alpha_{bd}^k)\left(\frac{\alpha}{\alpha_{bd}^k}\right)^2\right)\beta f^k + \left(\alpha - \alpha_{bd}^k\left(\frac{\alpha}{\alpha_{bd}^k}\right)^2\right)\beta f^k \\ &= \left(1 - \left(\frac{\alpha}{\alpha_{bd}^k}\right)^2\right)\beta f^k \\ &\geq \left(1 - \frac{\alpha}{\alpha_{bd}^k}\right)^2 \beta f^k \quad (\text{since } 0 \leq \alpha/\alpha_{bd}^k < 1). \end{aligned}$$

If $x_j^kz_j^k \geq \beta f^k$, we have

$$\begin{aligned} x_j^{k+1}z_j^{k+1} &= (x_j^k + \alpha\Delta x_j)(z_j^k + \alpha\Delta z_j) \\ &\geq \left(x_j^k + \alpha\left(-\frac{x_j^k}{\alpha_{bd}^k}\right)\right)\left(z_j^k + \alpha\left(-\frac{z_j^k}{\alpha_{bd}^k}\right)\right) \\ &\qquad (\text{by the definition (6) of } \alpha_{bd}^k \text{ and } 0 \leq \alpha < \alpha_{bd}^k) \\ &= \left(1 - \frac{\alpha}{\alpha_{bd}^k}\right)^2 x_j^kz_j^k \\ &\geq \left(1 - \frac{\alpha}{\alpha_{bd}^k}\right)^2 \beta f^k. \end{aligned}$$

Thus we have shown the latter assertion of the lemma. □

5. Proof of Theorem 2.1. Throughout this section we assume Rule G′; hence $0 \leq \bar{\beta} < 1$, $0 < \bar{\theta} < 1$, and $0 < \alpha^*$. Define

$$\kappa = \min\left\{\frac{1}{4}, \frac{\alpha^*}{4}\right\} \quad \text{and} \quad \gamma = \frac{8(2+\alpha^*)}{7(1-\bar{\beta})\alpha^*}.$$

We need another lemma to prove Theorem 2.1.

LEMMA 5.1. *Assume that* $\pi^k \leq \frac{1}{2}$, $\alpha = \alpha^k \leq \kappa$, *and* $\beta = \beta^k \in [0, \bar{\beta}]$.

(i) $\alpha/\alpha^k_{bd} \leq \frac{1}{2}$.
(ii) $\alpha/(\alpha^k_{bd})^2 \leq 1/\alpha^*$.
(iii) $0 \leq 1 - \alpha - (1 - \alpha^k_{bd})(\alpha/\alpha^k_{bd})^2 \leq 1$.
(iv) $\pi^{k+1} \geq (1 - \alpha(1 - \beta)/2)^{2\gamma}\pi^k$. (*Recall that* $0 \leq (1 - \alpha(1 - \beta)/2) \leq 1$. *See Lemma* 4.1.)

Proof. (i) If $\alpha^k_{bd} \geq \alpha^*/2$ then the desired inequality follows from the assumption $\alpha \leq \alpha^*/4$. Otherwise, the step length α chosen by Rule G′ satisfies $\alpha \leq {\alpha^k_{bd}}^2/\alpha^*$. Hence $\alpha/\alpha^k_{bd} \leq \alpha^k_{bd}/\alpha^* \leq \frac{1}{2}$.

(ii) The step length α is less than α^k_{bd}; see (11). If $\alpha^k_{bd} \geq \alpha^*$ then $\alpha/{\alpha^k_{bd}}^2 \leq 1/\alpha^k_{bd} \leq 1/\alpha^*$. Otherwise, $\alpha \leq {\alpha^k_{bd}}^2/\alpha^*$, which implies $\alpha/{\alpha^k_{bd}}^2 \leq 1/\alpha^*$.

(iii) By the assumption, $\alpha \leq \frac{1}{4}$ and $\alpha < \alpha^k_{bd}$. Hence

$$\begin{aligned} 0 &\leq \left(1 - \frac{\alpha}{\alpha^k_{bd}}\right)\left(1 - \alpha + \frac{\alpha}{\alpha^k_{bd}}\right) \\ &= 1 - \alpha - (1 - \alpha^k_{bd})\left(\frac{\alpha}{\alpha^k_{bd}}\right)^2 \\ &= 1 - \left(\frac{\alpha}{\alpha^k_{bd}}\right)^2 - \alpha\left(1 - \frac{\alpha}{\alpha^k_{bd}}\right) \\ &\leq 1. \end{aligned}$$

(iv) First we observe that $1 - \alpha(1 - \beta)/2 > 0$, since $0 \leq \alpha \leq \kappa \leq \frac{1}{4}$ and $0 \leq \beta \leq \bar{\beta} < 1$. Let j be fixed. By (iii) and Lemma 4.4, we see that

$$x^{k+1}_j z^{k+1}_j \geq \left(1 - \alpha - (1 - \alpha^k_{bd})\left(\frac{\alpha}{\alpha^k_{bd}}\right)^2\right)\pi^k f^k + \left(\alpha - \alpha^k_{bd}\left(\frac{\alpha}{\alpha^k_{bd}}\right)^2\right)\beta f^k.$$

By Lemma 4.1, $f^{k+1} \leq (1 - \alpha(1 - \beta)/2)f^k$. Hence

$$\frac{x^{k+1}_j z^{k+1}_j}{f^{k+1}} \geq \varphi(\beta) = \frac{\psi(\beta)}{\chi(\beta)},$$

where $\psi : [0,1] \to R$ and $\chi : [0,1] \to R$ are functions such that

$$\begin{aligned} \psi(\xi) &= \left(1 - \alpha - (1 - \alpha^k_{bd})\left(\frac{\alpha}{\alpha^k_{bd}}\right)^2\right)\pi^k + \left(\alpha - \alpha^k_{bd}\left(\frac{\alpha}{\alpha^k_{bd}}\right)^2\right)\xi, \\ \chi(\xi) &= 1 - \frac{\alpha(1 - \xi)}{2}. \end{aligned}$$

We now prove that $\varphi(\beta) \geq \varphi(0)$ by showing that

$$\varphi'(\xi) = \frac{\psi'(\xi)\chi(\xi) - \chi'(\xi)\psi(\xi)}{\chi(\xi)^2} \geq 0 \quad \text{for every } \xi \in [0, \beta].$$

Evaluating the numerator $\psi'(\xi)\chi(\xi) - \chi'(\xi)\psi(\xi)$ for each $\xi \in [0, \beta]$, we have

$$\psi'(\xi)\chi(\xi) - \chi'(\xi)\psi(\xi) = \left(\alpha - \alpha^k_{bd}\left(\frac{\alpha}{\alpha^k_{bd}}\right)^2\right)\left(1 - \frac{\alpha(1 - \xi)}{2}\right)$$

$$
\begin{aligned}
&- \frac{\alpha}{2}\left\{\left(1-\alpha-(1-\alpha_{bd}^{k})\left(\frac{\alpha}{\alpha_{bd}^{k}}\right)^{2}\right)\pi^{k}+\left(\alpha-\alpha_{bd}^{k}\left(\frac{\alpha}{\alpha_{bd}^{k}}\right)^{2}\right)\xi\right\}\\
&= \alpha\left(1-\frac{\alpha}{\alpha_{bd}^{k}}\right)\left(1-\frac{\alpha}{2}\right)-\frac{\alpha}{2}\left(1-\alpha-(1-\alpha_{bd}^{k})\left(\frac{\alpha}{\alpha_{bd}^{k}}\right)^{2}\right)\pi^{k}\\
&\geq \alpha\left(1-\frac{\alpha}{\alpha_{bd}^{k}}\right)\left(1-\frac{\alpha}{2}\right)-\frac{\alpha}{2}\pi^{k} \quad \text{(by (iii))}\\
&\geq \alpha\left(1-\frac{\alpha}{\alpha_{bd}^{k}}\right)\frac{7}{8}-\frac{\alpha}{4} \quad \left(\text{since } \alpha \leq \tfrac{1}{4} \text{ and } \pi^{k} \leq \tfrac{1}{2}\right)\\
&\geq 3\alpha/16 \quad \text{(by (i))}.
\end{aligned}
$$

Thus we have shown $\varphi'(\xi) \geq 0$ for all $\xi \in [0, \beta]$. Hence

$$
\begin{aligned}
\frac{x_j^{k+1} z_j^{k+1}}{f^{k+1}} &\geq \varphi(\beta)\\
&\geq \varphi(0)\\
&= \frac{(1-\alpha-(1-\alpha_{bd}^{k})(\alpha/\alpha_{bd}^{k})^{2})\pi^{k}}{1-\alpha/2}\\
&= \left(1-\frac{\alpha/2+(1-\alpha_{bd}^{k})(\alpha/\alpha_{bd}^{k})^{2}}{1-\alpha/2}\right)\pi^{k}\\
&\geq \left(1-\frac{8}{7}\left(\frac{\alpha}{2}+\left(\frac{\alpha}{\alpha_{bd}^{k}}\right)^{2}\right)\right)\pi^{k}.
\end{aligned}
$$

This inequality holds for every $j = 1, 2, \ldots, n$, so we obtain from the definition of π^{k+1} that

$$
\pi^{k+1} \geq \left(1-\frac{8}{7}\left(\frac{\alpha}{2}+\left(\frac{\alpha}{\alpha_{bd}^{k}}\right)^{2}\right)\right)\pi^{k}.
$$

We now utilize the inequality $1-\gamma'\xi' \geq (1-\xi')^{2\gamma'}$ for every $\xi' \in [0,1]$ and $\gamma' \geq 0$ such that $\gamma'\xi' \leq \frac{1}{2}$. Let $\gamma' = 8(1+2\alpha/{\alpha_{bd}^{k}}^{2})/(7(1-\beta))$ and $\xi' = \alpha(1-\beta)/2$. Then we have $0 \leq \xi' \leq 1$, $\gamma' \geq 0$, and

$$
\xi'\gamma' = \frac{8}{7}\left(\frac{\alpha}{2}+\left(\frac{\alpha}{\alpha_{bd}^{k}}\right)^{2}\right) \leq \frac{8}{7}\left(\frac{1}{8}+\frac{1}{4}\right) \leq \frac{1}{2}.
$$

Hence $\pi^{k+1} \geq (1-\xi'\gamma')\pi^{k} \geq (1-\xi')^{2\gamma'}\pi^{k}$. By (ii) and $\beta \leq \bar{\beta}$, we finally observe that

$$
\gamma' = \frac{8(1+2\alpha/{\alpha_{bd}^{k}}^{2})}{7(1-\beta)} \leq \frac{8(1+2/\alpha^{*})}{7(1-\bar{\beta})} = \gamma. \qquad \square
$$

We are ready to prove Theorem 2.1 in the same way as Theorem 3.2 in [1]. First we show that the generated sequence $\{(\boldsymbol{x}^{k}, \boldsymbol{z}^{k})\}$ is bounded. It is well known that for every $t \geq 0$, the set $\{(\boldsymbol{x}, \boldsymbol{z}) \in S : \boldsymbol{x}^{T}\boldsymbol{z} \leq t\}$ is bounded under assumptions (a) and (b), stated in §2. In fact, if $(\boldsymbol{x}, \boldsymbol{z}) \in \{(\boldsymbol{x}, \boldsymbol{z}) \in S : \boldsymbol{x}^{T}\boldsymbol{z} \leq t\}$ for some $t \geq 0$, then

$$
\begin{aligned}
0 &\leq (\boldsymbol{x}-\boldsymbol{x}^{0})^{T}\boldsymbol{M}(\boldsymbol{x}-\boldsymbol{x}^{0})\\
&= (\boldsymbol{x}-\boldsymbol{x}^{0})^{T}(\boldsymbol{z}-\boldsymbol{z}^{0})\\
&\leq \boldsymbol{x}^{T}\boldsymbol{z}-(\boldsymbol{z}^{0})^{T}\boldsymbol{x}-(\boldsymbol{x}^{0})^{T}\boldsymbol{z}+(\boldsymbol{x}^{0})^{T}\boldsymbol{z}^{0};
\end{aligned}
$$

hence $(\boldsymbol{x}, \boldsymbol{z})$ belongs to a bounded set $\{(\boldsymbol{x}, \boldsymbol{z}) \geq \mathbf{0} : (\boldsymbol{z}^0)^T\boldsymbol{x} + (\boldsymbol{x}^0)^T\boldsymbol{z} \leq t + (\boldsymbol{x}^0)^T\boldsymbol{z}^0\}$. On the other hand, the inequality $(\boldsymbol{x}^{k+1})^T\boldsymbol{z}^{k+1} \leq (\boldsymbol{x}^k)^T\boldsymbol{z}^k$ holds for every $k = 0, 1, \ldots$. Therefore, the generated sequence $\{(\boldsymbol{x}^k, \boldsymbol{z}^k)\}$ is contained in a bounded set $\{(\boldsymbol{x}, \boldsymbol{z}) \geq \mathbf{0} : (\boldsymbol{z}^0)^T\boldsymbol{x} + (\boldsymbol{x}^0)^T\boldsymbol{z} \leq 2(\boldsymbol{x}^0)^T\boldsymbol{z}^0\}$.

Let $\bar{\pi}$ be an arbitrary positive constant not greater than $\frac{1}{4}$. If $\pi^k \geq \bar{\pi}$, we see $2\bar{\pi}^2/n \leq (\alpha^k_{bd})^2$ by Lemma 4.2, $2(1-\bar{\beta})\bar{\pi} \leq \alpha^k_{\min}$ by Lemma 4.3, and

$$0 < \kappa_1 \equiv \min\left\{\bar{\theta}\alpha^*, \frac{2\bar{\theta}\bar{\pi}^2}{n\alpha^*}, 2(1-\bar{\beta})\bar{\pi}\right\} \leq \min\left\{\bar{\theta}\alpha^*, \frac{\bar{\theta}(\alpha^k_{bd})^2}{\alpha^*}, \alpha^k_{\min}\right\} \leq \alpha^k$$

by the inequality (11). On the other hand, if $\alpha^k \geq \kappa_1$ or $\alpha^k \geq \kappa$, we have by Lemma 4.1 that

$$(\boldsymbol{x}^{k+1})^T\boldsymbol{z}^{k+1} \leq (1-\delta)(\boldsymbol{x}^k)^T\boldsymbol{z}^k \quad \text{with } \delta = \tfrac{1}{2}(1-\bar{\beta})\min\{\kappa_1, \kappa\}.$$

Hence $(\boldsymbol{x}^k)^T\boldsymbol{z}^k$ converges to zero as $k \to \infty$ if the inequality above holds for infinitely many k's. So we may restrict ourselves to the case where there exists an integer ℓ such that

$$\lim_{k\to\infty} \pi^k = 0, \quad \alpha^k \leq \kappa \quad \text{and} \quad \pi^k \leq \frac{1}{2} \quad \text{for every } k \geq \ell.$$

Applying (iv) of Lemma 5.1 and Lemma 4.1, we obtain

$$\pi^{k+1} \geq \left(1 - \frac{\alpha^k(1-\beta^k)}{2}\right)^{2\gamma} \pi^k \geq \left(\frac{(\boldsymbol{x}^{k+1})^T\boldsymbol{z}^{k+1}}{(\boldsymbol{x}^k)^T\boldsymbol{z}^k}\right)^{2\gamma} \pi^k \quad \text{for every } k \geq \ell.$$

It follows that

$$\pi^{\ell+r} \geq \left(\frac{(\boldsymbol{x}^{\ell+r})^T\boldsymbol{z}^{\ell+r}}{(\boldsymbol{x}^\ell)^T\boldsymbol{z}^\ell}\right)^{2\gamma} \pi^\ell \quad \text{for every } r = 1, 2, \ldots.$$

Since $\lim_{r\to\infty} \pi^{\ell+r} = 0$ by the assumption, we conclude that $(\boldsymbol{x}^{\ell+r})^T\boldsymbol{z}^{\ell+r} \to 0$ as $r \to \infty$. This completes the proof of Theorem 2.1.

6. Proof of Theorem 2.2. We prove a series of assertions that lead to the proof of Theorem 2.2:

(i) $\sigma \leq \min\{\alpha^*/16,\ 1/32\}$.
(ii) $\pi^k \geq \sigma$ for every $k = 0, 1, \ldots$.
(iii) At each iteration, the step length α^k satisfies $\alpha^k \geq 3\sigma^2/(2n\alpha^*)$.
(iv) $(\boldsymbol{x}^{k+1})^T\boldsymbol{z}^{k+1} \leq (1 - 3\sigma^2/(8n\alpha^*))(\boldsymbol{x}^k)^T\boldsymbol{z}^k$.

(i) By the assumption, we have $0 \leq 1 - \theta^* \leq \frac{1}{4}$ and $0 < \beta^* \leq \frac{1}{2}$. Hence

$$\begin{aligned}\sigma &\leq \min\left\{\frac{(1-\theta^*)\beta^*\alpha^*}{2},\ (1-\theta^*)^2\beta^*\right\}\\ &\leq \min\left\{\frac{(1/4)(1/2)\alpha^*}{2},\ (1/4)^2(1/2)\right\}\\ &\leq \min\{\alpha^*/16,\ 1/32\}.\end{aligned}$$

(ii) We prove the assertion by induction. By the definition, we have $\pi^0 \geq \sigma$. Assuming $\pi^k \geq \sigma$, we show $x_j^{k+1}z_j^{k+1} - \sigma f^{k+1} \geq 0$ for every $j = 1, 2, \ldots, n$. Then $\pi^{k+1} \geq \sigma$ follows from the definition of π^{k+1}. Let $\alpha = \alpha^k$ and $\beta = \beta^k$.

(ii.a) We first consider the case where the inequality (14) holds. From Lemma 4.4, $\pi^k \geq \sigma$, and Lemma 4.1, we have that

$$\begin{aligned} x_j^{k+1} z_j^{k+1} - \sigma f^{k+1} &\geq \left(1 - \alpha - (1 - \alpha_{bd}^k)\left(\frac{\alpha}{\alpha_{bd}^k}\right)^2\right)\pi^k f^k \\ &\quad + \left(\alpha - \alpha_{bd}^k\left(\frac{\alpha}{\alpha_{bd}^k}\right)^2\right)\beta f^k - \left(1 - \frac{\alpha(1-\beta)}{2}\right)\sigma f^k \\ &\geq g(\alpha), \end{aligned}$$

where

$$g(\xi) = \left(-\frac{\xi}{2} - \frac{\xi\beta}{2} - (1 - \alpha_{bd}^k)\left(\frac{\xi}{\alpha_{bd}^k}\right)^2\right)\sigma f^k + \left(\xi - \alpha_{bd}^k\left(\frac{\xi}{\alpha_{bd}^k}\right)^2\right)\beta f^k.$$

We observe that $g(\xi)$ is a quadratic function with the coefficient of the quadratic term

$$\begin{aligned} -(1 - \alpha_{bd}^k)\sigma f^k/{\alpha_{bd}^k}^2 - \alpha_{bd}^k\beta f^k/{\alpha_{bd}^k}^2 &= -(\sigma + (\beta - \sigma)\alpha_{bd}^k)f^k/{\alpha_{bd}^k}^2 \\ &\leq -(\sigma + (\beta^* - \sigma)\alpha_{bd}^k)f^k/{\alpha_{bd}^k}^2 \\ &< 0. \end{aligned}$$

Obviously, $g(0) = 0$. Hence if $g(\xi) \geq 0$ then $g(\xi') \geq 0$ for all $\xi' \in [0, \xi]$. Since

$$\begin{aligned} &\alpha \leq \theta^*\alpha_{bd}^k && \text{if } \alpha_{bd}^k \geq \alpha^*, \\ &\alpha \leq \theta^*(\alpha_{bd}^k)^2/\alpha^* && \text{otherwise,} \end{aligned}$$

it suffices to show that

$$\begin{aligned} &g(\theta^*\alpha_{bd}^k) \geq 0 && \text{if } \alpha_{bd}^k \geq \alpha^*, \\ &g(\theta^*(\alpha_{bd}^k)^2/\alpha^*) \geq 0 && \text{otherwise.} \end{aligned}$$

If $\alpha_{bd}^k \geq \alpha^*$ then

$$\begin{aligned} g(\theta^*\alpha_{bd}^k) &= \left(-\frac{\theta^*\alpha_{bd}^k}{2}(1+\beta) - (1 - \alpha_{bd}^k)(\theta^*)^2\right)\sigma f^k + \left(\theta^*\alpha_{bd}^k - \alpha_{bd}^k(\theta^*)^2\right)\beta f^k \\ &= \left(\frac{\theta^*\alpha_{bd}^k}{2}(2\theta^* - \beta - 1) - (\theta^*)^2\right)\sigma f^k + (1 - \theta^*)\theta^*\alpha_{bd}^k\beta f^k \\ &\geq -(\theta^*)^2\sigma f^k + (1 - \theta^*)\theta^*\alpha^*\beta^* f^k \\ &\qquad (\text{since } \beta^* \leq \beta \leq \tfrac{1}{2}, \tfrac{3}{4} \leq \theta^* < 1, \text{and } \alpha^* \leq \alpha_{bd}^k) \\ &\geq -(\theta^*)^2\frac{(1 - \theta^*)\alpha^*\beta^*}{2}f^k + (1 - \theta^*)\theta^*\alpha^*\beta^* f^k \\ &= (1 - \frac{\theta^*}{2})(1 - \theta^*)\theta^*\alpha^*\beta^* f^k \\ &\geq 0. \end{aligned}$$

Otherwise,

$$g(\theta^*{\alpha_{bd}^k}^2/\alpha^*) \geq \left(-\frac{\theta^*{\alpha_{bd}^k}^2}{2\alpha^*}(1+\beta) - \left(\frac{\theta^*\alpha_{bd}^k}{\alpha^*}\right)^2\right)\sigma f^k$$

$$
\begin{aligned}
&+\left(\frac{\theta^* {\alpha^k_{bd}}^2}{\alpha^*} - \alpha^k_{bd}\left(\frac{\theta^*\alpha^k_{bd}}{\alpha^*}\right)^2\right)\beta f^k \\
&= -\left(\frac{1+\beta}{2}\alpha^* + \theta^*\right)\theta^*\left(\frac{\alpha^k_{bd}}{\alpha^*}\right)^2 \sigma f^k + (\alpha^* - \theta^*\alpha^k_{bd})\theta^*\left(\frac{\alpha^k_{bd}}{\alpha^*}\right)^2 \beta f^k \\
&\geq -2\theta^*\left(\frac{\alpha^k_{bd}}{\alpha^*}\right)^2 \sigma f^k + (1-\theta^*)\alpha^*\theta^*\left(\frac{\alpha^k_{bd}}{\alpha^*}\right)^2 \beta^* f^k \\
&\qquad \text{(since } 0 \leq \beta,\ \theta^*,\ \alpha^* \leq 1,\ \alpha^k_{bd} < \alpha^*, \text{ and } \beta^* \leq \beta) \\
&= (-2\sigma + (1-\theta^*)\alpha^*\beta^*)\theta^*\left(\frac{\alpha^k_{bd}}{\alpha^*}\right)^2 f^k \\
&\geq 0.
\end{aligned}
$$

(ii.b) Now we consider the case where the inequality (14) does not hold. From Lemma 4.4, we have that

$$
\begin{aligned}
x^{k+1}_j z^{k+1}_j &\geq \left(1 - \frac{\alpha}{\alpha^k_{bd}}\right)^2 \beta f^k \\
&\geq (1-\theta^*)^2\beta^* f^{k+1} \quad (\text{since } \alpha/\alpha^k_{bd} \leq \theta^*,\ \beta^* \leq \beta, \text{ and } f^k \geq f^{k+1}) \\
&\geq \sigma f^{k+1} \quad (\text{since } \sigma \leq (1-\theta^*)^2\beta^*).
\end{aligned}
$$

(iii) At each iteration of the Algorithm using Rule P′, the step length α satisfies (11) with $\bar{\theta} = \frac{3}{4}$. First we see, by Lemma 4.2,

$$
\begin{aligned}
(\alpha^k_{bd})^2 &\geq \min\left\{\frac{1}{4},\ \frac{2(\pi^k)^2}{n}\right\} \\
&\geq \min\left\{\frac{1}{4},\ \frac{2\sigma^2}{n}\right\} \quad (\text{since } \pi^k \geq \sigma \text{ by (ii)}) \\
&\geq \frac{2\sigma^2}{n} \quad (\text{since } \sigma \leq 1/32 \text{ by (i)}),
\end{aligned}
$$

and, by Lemma 4.3,

$$\alpha^k_{\min} \geq 2(1-\bar{\beta})\pi^k \geq \sigma \quad (\text{since } \bar{\beta} \leq \tfrac{1}{2} \text{ and } \pi^k \geq \sigma).$$

Since the step length α satisfies (11) with $\bar{\theta} = \frac{3}{4}$ at each iteration of the Algorithm using Rule P′, we obtain

$$\alpha^k \geq \min\left\{\frac{3\alpha^*}{4}, \frac{3\sigma^2}{2n\alpha^*}, \sigma\right\} \geq \frac{3\sigma^2}{2n\alpha^*} \quad (\text{since } \sigma \leq \alpha^*/16 \text{ by (i)}).$$

(iv) By Lemma 4.1 and $\beta \leq \bar{\beta} = \frac{1}{2}$, $(\boldsymbol{x}^{k+1})^T\boldsymbol{z}^{k+1} \leq \left(1 - \frac{\alpha^k}{4}\right)(\boldsymbol{x}^k)^T\boldsymbol{z}^k$. Hence the desired inequality follows from (iii).

7. Concluding remarks on the local convergence. Recently many studies (see [2], [23], [25]–[27], etc.) have been done on the local convergence of interior point algorithms for linear programs and positive semidefinite linear complementarity problems. Among others, we refer to the work by Ye and Anstreicher [23], in which a version of the predictor-corrector interior point algorithm is proposed (see also Mizuno, Todd, and Ye

[16]). The algorithm of Ye and Anstreicher enjoys not only the $O(\sqrt{n}L)$ iteration global convergence, but also the Q-quadratic local convergence under the assumption that

(A) the LCP has a strict complementary solution.

It may be regarded as a special case of the Algorithm presented in §2. But their step length control of [23] elaborately utilizes a collection of prescribed neighborhoods of the form $\{(\boldsymbol{x},\boldsymbol{z}) \in S_{++} : \|\boldsymbol{X}\boldsymbol{z}/(\boldsymbol{x}^T\boldsymbol{z}/n) - \boldsymbol{e}\| \leq \alpha\}$ with $\alpha \in (0, 0.5]$, and is very different from our Rules G′ or P′. See also Ye, Güler, Tapia, and Zhang [25]. We can prove the following result, but the proof is omitted.

THEOREM 7.1. *Let* $\alpha^* \in (0, 1]$ *and* $\bar{\theta} \in (0, 1)$ *be fixed. At each iteration of the Algorithm, choose a direction parameter* $\beta = \beta^k$ *and a step length* $\alpha = \alpha^k$ *such that*

(B) $0 \leq \beta^k < 1$, $\beta^k \to 0$ *as* $k \to \infty$ *and* $\{\beta^k/\pi^k\}$ *is bounded; for example, let* $\beta^k = 0$ *throughout the iterations, or* $\beta^k = \min\{f^k, \pi^k, \bar{\beta}\}$ *adaptively;*

(C) $\alpha^k = \min\{\theta^k\alpha^k_{\mathrm{bd}}, \theta^k(\alpha^k_{\mathrm{bd}})^2/\alpha^*, \alpha^k_{\min}\}$, *where* $\{\theta^k \in [\bar{\theta}, 1) : k = 1, 2, \dots\}$ *is a sequence converging to* 1.

Then the generated sequence $\{(\boldsymbol{x}^k, \boldsymbol{z}^k)\}$ *is bounded and* $(\boldsymbol{x}^k)^T\boldsymbol{z}^k \to 0$ *as* $k \to \infty$. *If in addition*

(A)′ *the LCP has a unique solution* $(\boldsymbol{x}^*, \boldsymbol{z}^*)$, *which satisfies the strict complementarity* $\boldsymbol{x}^* + \boldsymbol{z}^* > \mathbf{0}$, *then* $(\boldsymbol{x}^k)^T\boldsymbol{z}^k$ *converges to* 0 *Q-superlinearly as* $k \to \infty$, *i.e.,*

$$\lim_{k\to\infty} \frac{(\boldsymbol{x}^{k+1})^T\boldsymbol{z}^{k+1}}{(\boldsymbol{x}^k)^T\boldsymbol{z}^k} = 0.$$

Acknowledgment. The authors thank two anonymous referees for useful comments and suggestions.

REFERENCES

[1] M. Kojima, N. Megiddo, and S. Mizuno, *Theoretical convergence of large-step primal-dual interior point algorithms for linear programming*, RJ 7872, IBM Almaden Research Center, San Jose, CA, 1990.

[2] M. Kojima, N. Megiddo, and T. Noma, *Homotopy continuation methods for nonlinear complementarity problems*, Math. Oper. Res., 16 (1991), pp. 754–774.

[3] 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, New York, 1991.

[4] M. Kojima, N. Megiddo, and Y. Ye, *An interior point potential reduction algorithm for the linear complementarity problem*, Math. Programming, to appear.

[5] M. Kojima, S. Mizuno, and T. Noma, *Limiting behavior of trajectories generated by a continuation method for monotone complementarity problems*, Math. Oper. Res. 15 (1990), pp. 662–675.

[6] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior point algorithm for linear programming*, in Progress in Mathematical Programming, Interior-Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 29–47.

[7] ———, *A polynomial-time algorithm for a class of linear complementarity problems*, Math. Programming, 44 (1989), pp. 1–26.

[8] ———, *An* $O(\sqrt{n}L)$ *iteration potential reduction algorithm for linear complementarity problems*, Math. Programming, 50 (1991), pp. 331–342.

[9] ———, *A little theorem of the big* $\mathcal{M}$ *in interior point algorithms*, Math. Programming, to appear.

[10] R. Marsten, R. Subramanian, M. Saltzman, I. Lustig, and D. Shanno, *Interior point methods for linear programming: Just call Newton, Lagrange and Fiacco and McCormick!*, Interfaces, 20 (1990), pp. 105–116.

[11] K. A. McShane, C. L. Monma, and D. F. Shanno, *An implementation of a primal-dual interior point method for linear programming*, ORSA J. Comput. 1 (1989), pp. 70–83.

[12] N. Megiddo, *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming, Interior-Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 131–158.

[13] S. MEHROTRA, *On the implementation of a (primal-dual) interior point method*, Tech. Rep. 90-03, Dept. of Industrial Engineering and Management Sciences, Northwestern Univ., Evanston, IL, 1990.

[14] S. MIZUNO, *A new polynomial time method for a linear complementarity problem*, Math. Programming, to appear.

[15] S. MIZUNO AND M. J. TODD, *An $O(n^3L)$ adaptive path following algorithm for a linear complementarity problem*, Math. Programming, to appear.

[16] S. MIZUNO, M. J. TODD, AND Y. YE, *On adaptive-step primal-dual interior-point algorithms for linear programming*, Math. Oper. Res., to appear.

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

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

[19] K. TANABE, *Complementarity-enforcing centered Newton method for mathematical programming*, in New Methods for Linear Programming, K. Tone, ed., The Institute of Statistical Mathematics, Minami-Azabu, Minato-ku, Tokyo, 1987, pp. 118–144.

[20] ———, *Centered Newton method for mathematical programming*, in Systems Modeling and Optimization, M. Iri and K. Yajima, eds., Springer-Verlag, New York, 1988, pp. 197–206.

[21] M. J. TODD, *Projected scaled steepest descent in Kojima-Mizuno-Yoshise potential reduction algorithm for the linear complementarity problem*, Tech. Rep. 950, School of Operations Research and Industrial Engineering, Cornell Univ., Ithaca, NY, 1990.

[22] M. J. TODD AND Y. YE, *A centered projective algorithm for linear programming*, Math. Oper. Res., 15 (1990), pp. 508–529.

[23] Y. YE AND K. ANSTREICHER, *On quadratic and $O(\sqrt{n}L)$ convergence of a predictor-corrector algorithm for* LCP, Working paper, Dept. of Management Science, The Univ. of Iowa, Iowa City, IA, 1990.

[24] Y. YE, K. O. KORTANEK, J. A. KALISKI, AND S. HUANG, *Near-boundary behavior of primal-dual potential reduction algorithms for linear programming*, Working Paper Series No. 90-9, College of Business Administration, The Univ. of Iowa, Iowa City, IA, 1990.

[25] Y. YE, O. GÜLER, R. A. TAPIA, AND Y. ZHANG, *A quadratically convergent $O(\sqrt{n}L)$-iteration algorithm for linear programming*, Math. Programming, to appear.

[26] Y. ZHANG AND R. A. TAPIA, *A quadratically convergent polynomial primal-dual interior-point algorithm for linear programming*, TR90-40, Dept. of Mathematical Sciences, Rice Univ., Houston, TX, 1990.

[27] Y. ZHANG, R. A. TAPIA, AND F. POTRA, *On the superlinear convergence of interior point algorithms for a general class of problems*, TR90-9, Dept. of Mathematical Sciences, Rice Univ., Houston, TX, 1990.

SIAM J. OPTIMIZATION
Vol. 3, No. 2, pp. 413–422, May 1993

ON THE SUPERLINEAR CONVERGENCE OF INTERIOR-POINT ALGORITHMS FOR A GENERAL CLASS OF PROBLEMS*

YIN ZHANG[†], RICHARD TAPIA[‡], AND FLORIAN POTRA[§]

Abstract. In this paper, the authors extend the Q-superlinear convergence theory recently developed by Zhang, Tapia, and Dennis for a class of interior-point linear programming algorithms to similar interior-point algorithms for quadratic programming and for linear complementarity problems. This unified approach consists of viewing all these algorithms as a damped Newton method applied to perturbations of a general problem. A set of sufficient conditions for these algorithms to achieve Q-superlinear convergence is established. The key ingredients consist of asymptotically taking the step to the boundary of the positive orthant and letting the centering parameter approach zero at a specific rate. The construction of algorithms that have both the global property of polynomiality and the local property of superlinear convergence will be the subject of further research.

Key words. interior-point algorithms, linear programming, quadratic programming, linear complementarity problems, Q-superlinear convergence

AMS subject classifications. 65K05, 90C05

1. Introduction.

Consider the general nonlinear system

$$F(x,y) = \begin{pmatrix} Mx + Ny - h \\ XYe \end{pmatrix} = 0, \qquad (x,y) \geq 0, \tag{1}$$

where $x, y, h, e \in \mathbf{R}^n$, $M, N \in \mathbf{R}^{n\times n}$, $X = \text{diag}(x)$, $Y = \text{diag}(y)$, and e has all components equal to one.

We call the following set the feasibility set of (1):

$$\Omega = \{(x,y) : x, y \in \mathbf{R}^n, Mx + Ny = h, (x,y) \geq 0\}.$$

A feasible pair $(x,y) \in \Omega$ is said to be strictly feasible if it is positive. In this work we tacitly assume that the relative interior of Ω is nonempty, i.e., strictly feasible points exist.

Problem (1) is sufficiently general to include linear and quadratic programming problems and linear complementarity problems. Observe that if $N = -I$, then this problem is the standard linear complementarity problem (LCP). Moreover, the assumption that M is positive semidefinite will be sufficient to guarantee that the algorithms under investigation produce well-defined iterates (Corollary 2.2).

It is well known that quadratic programs are special cases of LCPs. We now provide a somewhat different formulation of quadratic programs as special cases of (1) instead of those of the standard LCP. Consider the quadratic program (QP)

$$\begin{aligned} &\text{minimize} && c^Tx + \tfrac{1}{2}x^TQx \\ &\text{subject to} && Ax = b, \quad x \geq 0, \end{aligned} \tag{2}$$

*Received by the editors August 30, 1990; accepted for publication March 19, 1992. This research was supported in part by National Science Foundation cooperative agreement CCR-8809615, Air Force Office of Scientific Research grant 89-0363, Department of Energy grant DEFG05-86ER25017, and Army Research Office grant 9DAAL03-90-G-0093.

[†] Department of Mathematics and Statistics, University of Maryland Baltimore County, Baltimore, Maryland 21228-5398.

[‡] Department of Mathematical Sciences, Rice University, Houston, Texas 77251-1892.

[§] Department of Mathematics, The University of Iowa, Iowa City, Iowa 52242.

where $c, x \in \mathbf{R}^n$, $b \in \mathbf{R}^m$, $A \in \mathbf{R}^{m\times n}(m < n)$ and has full row rank, and $Q \in \mathbf{R}^{n\times n}$ is symmetric. In Corollary 2.2, we will demonstrate that iterates produced by the algorithms under investigation are well defined if Q is positive semidefinite on the null space of A. In this case, it is well known that the problem is convex and the first-order conditions are both necessary and sufficient for optimality. The first-order conditions for (2) can be transformed into the form of (1). To see this, let $B \in \mathbf{R}^{(n-m)\times n}$ be any matrix such that the columns of B^T form a basis for the null space of A. The first-order conditions for the quadratic program (2) are (see Dantzig [1])

$$(3) \qquad \begin{pmatrix} Ax - b \\ A^T\lambda - Qx + y - c \\ XYe \end{pmatrix} = 0, \qquad (x, y) \geq 0,$$

where λ and y are the dual variables. To eliminate the dual variables λ from the above system, we premultiply the second equation by the nonsingular matrix $[A^T \; B^T]^T$. Noticing that $BA^T = 0$, we obtain

$$0 = \begin{bmatrix} A \\ B \end{bmatrix} (A^T\lambda - Qx + y - c) = \begin{pmatrix} AA^T\lambda - A(Qx - y + c) \\ -BQx + By - Bc \end{pmatrix}.$$

Since AA^T is nonsingular, λ is uniquely determined once x and y are known. Removing the equation for λ, we arrive at the following $2n$-dimensional nonlinear system with nonnegativity constraints for (x, y)

$$(4) \qquad \begin{pmatrix} Ax - b \\ -BQx + By - Bc \\ XYe \end{pmatrix} = 0, \qquad (x, y) \geq 0.$$

Clearly, (4) is in the form of (1) with

$$(5) \qquad M = \begin{bmatrix} A \\ -BQ \end{bmatrix}, \quad N = \begin{bmatrix} 0 \\ B \end{bmatrix}, \quad \text{and} \quad h = \begin{bmatrix} b \\ Bc \end{bmatrix}.$$

When $Q = 0$, the quadratic program (2) reduces to a standard-form linear program (LP)

$$(6) \qquad \begin{array}{ll} \text{minimize} & c^Tx \\ \text{subject to} & Ax = b, \quad x \geq 0. \end{array}$$

Hence (2) also includes the linear program. However, because of the importance of linear programming in optimization, we will state results for linear programming separately, fully aware that they are special cases of quadratic programming. We have shown that the framework of problem (1) is quite general.

The objective of this work is to analyze the asymptotic behavior of a generic interior-point algorithm for solving (1). More specifically, we will study the Q-convergence rate of this general algorithm. The issues of global convergence and complexity are not of concern here.

Recently, Zhang, Tapia, and Dennis [18, Thm. 3.1] established a Q-superlinear convergence theory for a class of primal-dual interior-point algorithms for linear programming. In this paper, we extend their result to the general problem (1) and therefore extend the result to quadratic programming and LCPs. In spite of its close connection to [18], we have made this paper self-contained.

Given $u, v \in \mathbf{R}^n$ and $\eta \in \mathbf{R}$, we will use the notation

$$\min(u) = \min_{1 \le i \le n} [u]_i \quad \text{and} \quad \min(u, v, \eta) = \min\{\min(u), \min(v), \eta\},$$

where $[u]_i$ denotes the ith component of u.

The paper is organized as follows. In §2, we describe a general interior-point algorithmic framework for (1). Then in §3, we present our superlinear convergence rate result. Concluding remarks are given in §4.

2. Algorithm. It is now fairly well understood how a class of interior-point algorithms can be viewed as damped Newton methods and that the inclusion of the logarithmic barrier term (so-called centering) can be viewed as perturbing the right-hand side of the Newton system. Indeed, Zhang, Tapia, and Dennis [18] focused on issues concerning how fast the damped Newton method could approach the Newton method (i.e., step-length approaches one), and how fast the perturbation term (barrier parameter) should be phased out so that the fast convergence of Newton's method is not compromised. Their work covered linear programming applications. As previously mentioned, the objective of the present work is to extend a particularly nice part of their superlinear convergence theory to quadratic programming and LCPs. Our vehicle for accomplishing this objective is the use of the general problem (1). We assume that the reader is familiar with the above algorithmic considerations and we therefore present our algorithmic framework with no further motivation or explanation.

Recall that $F(x, y)$ is given by (1).

ALGORITHM 1. Given a pair $(x_0, y_0) > 0$. For $k = 0, 1, 2, \ldots,$ do

Step 1. Choose $\sigma_k \in [0, 1)$ and $\tau_k \in (0, 1)$. Set $\mu_k = \sigma_k x_k^T y_k / n$.

Step 2. Solve the following system for $(\Delta x_k, \Delta y_k)$:

$$F'(x_k, y_k) \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = -F(x_k, y_k) + \begin{pmatrix} 0 \\ \mu_k e \end{pmatrix}. \tag{7}$$

Step 3. Compute the step-length:

$$\alpha_k = \frac{-\tau_k}{\min(X_k^{-1}\Delta x_k, Y_k^{-1}\Delta y_k, -\tau_k)}. \tag{8}$$

Step 4. Update: $x_{k+1} = x_k + \alpha_k \Delta x_k$ and $y_{k+1} = y_k + \alpha_k \Delta y_k$.

Notice that in Algorithm 1, we do not require that the starting point (x_0, y_0) be feasible. Also notice that without the perturbation term $\mu_k e$ in the right-hand side of (7), the search direction $(\Delta x_k, \Delta y_k)$ is the Newton step. We always have $0 < \alpha_k \le 1$. Moreover, $\alpha_k = 1$ if and only if $\min(X_k^{-1}\Delta x_k, Y_k^{-1}\Delta y_k) \ge -\tau_k$. We should expect that only in rare cases would the full Newton step lead to a strictly positive iterate; hence we should expect in most cases to have $\alpha_k < 1$ where α_k is given by (8). The choice $\tau_k = 1$

corresponds to allowing steps to the boundary of the positive orthant and to a loss of strict feasibility. Therefore, it is natural to view Algorithm 1 as a perturbed and damped Newton's method. We see that if (x_0, y_0) is in Ω, then the iteration sequence $\{(x_k, y_k)\}$ will be strictly feasible. In the case of linear programming, there are no linear equations in $F(x, y)$ that involve both x and y. If $(x_k, y_k) \in \Omega$, then different step-lengths can be used to update x_k and y_k and still retain strictly feasible (x_{k+1}, y_{k+1}). This strategy has been shown to be more efficient in practice (see Lustig, Marsten, and Shanno [9], for example). However, it will not affect our results since our analysis will show that as long as $\tau_k \to 1$ both step-lengths will converge to one.

Algorithm 1 covers or is closely related to a wide range of existing interior-point algorithms for linear programming, quadratic programming, and LCPs. In particular, it covers most of the existing primal-dual interior-point algorithms for linear programming as well as quadratic programming, including Kojima, Mizuno, and Yoshise [7]; Todd and Ye [15]; Monteiro and Adler [12], [13]; Lustig [8]; Gonzaga and Todd [2]; Mizuno, Todd, and Ye [11]. Algorithms for LCPs that are covered by or closely related to Algorithm 1 include Kojima, Mizuno, and Yoshise [5], [6]; Kojima, Megiddo, and Noma [3]; and Kojima, Mizuno, and Noma [4].

Although these algorithms have been motivated and presented in various ways including path-following (homotopy or continuation), potential reduction, or affine scaling algorithms, most of them fit into the framework of the perturbed and damped Newton's method applied to the general problem (1). Due to the extensive activity in this area, our list of references is not complete. For a more complete list of references, especially in the cases of quadratic programming and LCPs, we refer the reader to two recent survey papers by Ye [16], [17].

The following proposition gives a condition which guarantees that the iterates produced by Algorithm 1 are well defined.

PROPOSITION 2.1. *The iterates produced by Algorithm* 1 *are well defined if for any positive diagonal matrix* $D \in \mathbf{R}^{n \times n}$, *the matrix* $N - MD$ *is nonsingular.*

Proof. Since $(x_0, y_0) > 0$ and

$$F'(x, y) = \begin{bmatrix} M & N \\ Y & X \end{bmatrix}, \tag{9}$$

the nonsingularity of $F'(x_0, y_0)$ is equivalent to that of

$$\begin{bmatrix} I & -MY_0^{-1} \\ 0 & I \end{bmatrix} F'(x_0, y_0) = \begin{bmatrix} 0 & N - MY_0^{-1}X_0 \\ Y_0 & X_0 \end{bmatrix}.$$

This latter matrix is nonsingular if and only if $N - MY_0^{-1}X_0$ is nonsingular. By our condition, (x_1, y_1) is well defined. An induction argument completes the proof. □

The following corollary is well known and one can easily verify that Proposition (2.1) is satisfied in the three cases of interest.

COROLLARY 2.2. *The iterates produced by Algorithm* 1 *are well defined for*

1. *the LCP* $(N = -I)$ *with* M *positive semidefinite,*

2. *the quadratic programming problem* (2) *with* Q *positive semidefinite on the null space of* A,

3. *the linear programming problem* (6).

We should mention that we have stated Algorithm 1 in the current form purely for the purposes of obtaining a unified theory and notational convenience. By directly applying the perturbed and damped Newton method to the first-order conditions for the quadratic program (2), it is not difficult to see that an identical iteration sequence $\{(x_k, y_k)\}$ will be generated without eliminating the dual variable λ and introducing the matrix B.

3. Superlinear convergence. The literature contains numerous studies directed at investigating the convergence properties of interior-point algorithms covered by or closely related to Algorithm 1. However, most of these studies were concerned only with the issues of global convergence and complexity. The issue of convergence rate, which is certainly important, has not been thoroughly studied for many interior-point algorithms. One of the few papers that studied asymptotic behavior (local convergence) of interior-point algorithms is Kojima, Megiddo, and Noma [3]. In their paper, Kojima, Megiddo, and Noma proved that for a class of complementarity problems, in addition to global convergence, superlinear and quadratic local convergence can be achieved by some interior-point algorithms in the form of Algorithm 1. However, all their convergence rate results were obtained under the restriction that the Jacobian matrix $F'(x, y)$ was nonsingular at the solution. In this section, we provide a set of sufficient conditions for superlinear convergence of Algorithm 1 applied to the general problem (1). These conditions do not require the nonsingularity of $F'(x, y)$ at solutions. How to apply these conditions to construct globally and superlinearly convergent algorithms is an interesting topic and the subject of further research.

It is satisfying that it is possible to obtain a superlinear convergence rate without the assumption of nonsingularity of the Jacobian matrix at the solution. In the case of linear programming, this allows one to avoid restrictive nondegeneracy assumptions. The motivation for this theory came from numerical experiments that demonstrated superlinear convergence even for highly degenerate linear programs.

At the kth iteration of Algorithm 1, let

$$\eta_k = \frac{x_k^T y_k / n}{\min(X_k Y_k e)}.$$

Since $x_k^T y_k / n$ is the average value of the elements of $X_k Y_k e$, it is clear that $\eta_k \geq 1$.

THEOREM 3.1. *Let $\{(x_k, y_k)\}$ be generated by Algorithm 1 with $\tau_k \to 1$ and $\sigma_k \to 0$, and let $(x_k, y_k) \to (x_*, y_*)$. Assume*

1. *strict complementarity at (x_*, y_*),*
2. *that the sequence $\{\eta_k\}$ is bounded,*
3. *that there exists $\rho \in [0, 1)$ such that for k sufficiently large*

$$\Delta x_k^T \Delta y_k \geq -\frac{\rho}{2}(\Delta x_k^T (X_k^{-1} Y_k) \Delta x_k + \Delta y_k^T (X_k Y_k^{-1}) \Delta y_k).$$

Then (x_*, y_*) solves problem (1) and the sequence $\{F(x_k, y_k)\}$ componentwise converges to zero Q-superlinearly. Furthermore, the sequence $\{F(x_k, y_k)\}$ itself is also Q-superlinearly convergent, i.e., for any norm

$$\lim_{k \to \infty} \sup \frac{\|F(x_{k+1}, y_{k+1})\|}{\|F(x_k, y_k)\|} = 0.$$

Before we prove Theorem 3.1, we would like to comment on the assumptions of Theorem 3.1. First, assumption 3 is not particularly restrictive since we will see later that in the context of linear programming, quadratic programming with Q positive semidefinite on the null space of A, and LCP with M positive semidefinite, we have the stronger result that $\Delta x_k^T \Delta y_k \geq 0$ for $(x_k, y_k) \in \Omega$. We used the more general assumption 3 instead of $\Delta x_k^T \Delta y_k \geq 0$ based on the consideration that the former could be useful in studying situations where (x_k, y_k) is not feasible. We stress that the algorithm designer is free to choose σ_k and τ_k, and the requirement that they be chosen so that $\sigma_k \to 0$ and $\tau_k \to 1$ is not particularly restrictive.

On the other hand, the compatibility of assumption 2 with the choices $\tau_k \to 1$ and $\sigma_k \to 0$ may be a cause for concern. It seems as if letting $\tau_k \to 1$ and $\sigma_k \to 0$ might force $\eta_k \to \infty$. However, our numerical experience has shown this not to be the case for linear programming. In our numerical studies with Netlib problems for linear programming, we let $\tau_k \to 1$ and $\sigma_k \to 0$ and always observed strict complementarity and bounded $\{\eta_k\}$. While on occasion we saw some rather large values for η_k's, they eventually leveled off or actually started to decrease as the iterates approached a solution. We did not observe continued growth in the values of η_k as our algorithm converged. Moreover, the observed convergence was clearly Q-superlinear and $\alpha_k \to 1$. Of course, the behavior of $\{\eta_k\}$ varies with several factors, including how fast $\{\tau_k\}$ converges to one and $\{\sigma_k\}$ to zero. We do not mean to imply that unbounded $\{\eta_k\}$ cannot occur. Instead, we feel that it appears to be more the exception than the rule in linear programming. It still remains to be seen whether or not this same phenomenon exists in quadratic programming and LCPs. There is no doubt that this topic merits further study.

To prove Theorem 3.1, we need the following lemma.

LEMMA 3.2. *Under the assumptions of Theorem* 3.1,

$$\lim_{k \to \infty} \alpha_k = 1. \tag{10}$$

Proof. Define at each iteration

$$p_k = X_k^{-1} \Delta x_k \quad \text{and} \quad q_k = Y_k^{-1} \Delta y_k. \tag{11}$$

At iteration k, from (7) and (9) we have

$$Y_k \Delta x_k + X_k \Delta y_k = -X_k Y_k e + \mu_k e,$$

or equivalently, recalling that $\mu_k = \sigma_k x_k^T y_k / n$ (see Step 1 of Algorithm 1)

$$p_k + q_k = -e + \mu_k (X_k Y_k)^{-1} e = -e + \sigma_k T_k e \tag{12}$$

where $T_k = (x_k^T y_k / n)(X_k Y_k)^{-1}$. Since $\eta_k = \|T_k e\|_\infty$, assumption 2 and $\sigma_k \to 0$ imply

$$\lim_{k \to \infty} (p_k + q_k) = -e. \tag{13}$$

Multiply both sides of (12) by $(X_k Y_k)^{\frac{1}{2}}$ and consider the square of the ℓ_2-norm. After dividing both sides by $x_k^T y_k$, we obtain the following equality:

$$\frac{\|(X_k Y_k)^{\frac{1}{2}} p_k\|_2^2 + \|(X_k Y_k)^{\frac{1}{2}} q_k\|_2^2 + 2\Delta x_k^T \Delta y_k}{x_k^T y_k} = \left(1 - 2\sigma_k + \sigma_k^2 \frac{x_k^T y_k}{n} \frac{e^T (X_k Y_k)^{-1} e}{n}\right).$$

Note that

$$\|(X_k Y_k)^{\frac{1}{2}} p_k\|_2^2 = \Delta x_k^T (X_k^{-1} Y_k) \Delta x_k \quad \text{and} \quad \|(X_k Y_k)^{\frac{1}{2}} q_k\|_2^2 = \Delta y_k^T (X_k Y_k^{-1}) \Delta y_k.$$

By assumption 3,

$$\frac{(1-\rho)(\|(X_kY_k)^{\frac{1}{2}}p_k\|_2^2+\|(X_kY_k)^{\frac{1}{2}}q_k\|_2^2)}{x_k^Ty_k} \le \left(1-2\sigma_k+\sigma_k^2\frac{x_k^Ty_k}{n}\frac{e^T(X_kY_k)^{-1}e}{n}\right).$$

Multiplying both sides of the above inequality by n, we obtain

$$(1-\rho)(\|T_k^{-\frac{1}{2}}p_k\|_2^2+\|T_k^{-\frac{1}{2}}q_k\|_2^2) \le n\left(1-2\sigma_k+\sigma_k^2\frac{e^TT_ke}{n}\right). \tag{14}$$

Assumption 2 implies that $\{\|T_k\|\}$ is bounded above and $\{\|T_k^{-\frac{1}{2}}\|\}$ is bounded away from zero. Therefore, from (14) both $\{p_k\}$ and $\{q_k\}$ are bounded. It now follows from (8) that $\{\alpha_k\}$ is bounded away from zero.

Now assume $[x_*]_i > 0$. Obviously,

$$1 = \lim_{k\to\infty}\frac{[x_{k+1}]_i}{[x_k]_i} = \lim_{k\to\infty}(1+\alpha_k[p_k]_i).$$

This implies $[p_k]_i \to 0$, because $\{\alpha_k\}$ is bounded away from zero. From (13) we have $[q_k]_i \to -1$. On the other hand, if $[x_*]_i = 0$, then $[y_*]_i > 0$ by strict complementarity. The same argument, interchanging the roles of p_k and q_k, gives $[q_k]_i \to 0$ and $[p_k]_i \to -1$. Therefore, the components of p_k and q_k converge to either 0 or -1. Consequently, from (11), (8), and $\tau_k \to 1$ it follows that $\alpha_k \to 1$. This completes the proof. □

Now we are ready to prove Theorem 3.1.

Proof of Theorem 3.1. Let

$$F_1(x,y) = Mx+Ny-h \quad \text{and} \quad F_2(x,y) = XYe.$$

We will prove that both $\{F_1(x_k,y_k)\}$ and $\{F_2(x_k,y_k)\}$ componentwise converge to zero Q-superlinearly. This will imply that $\{F(x_k,y_k)\}$ componentwise converges to zero Q-superlinearly. It is not difficult to see that componentwise Q-superlinear convergence of a vector sequence implies its Q-superlinear convergence.

First we show that the sequence $\{F_1(x_k,y_k)\}$ componentwise converges to zero Q-superlinearly. If $F_1(x_0,y_0) = 0$ (i.e., (x_0,y_0) is a feasible starting point), then it is easy to see that $F_1(x_k,y_k) = 0$ for all k. Therefore, we need only consider the case where $F_1(x_0,y_0) \ne 0$. Note that Newton's method solves linear equations in one step. If for some integer $p \ge 0$, $\alpha_p = 1$, then we have $F_1(x_k,y_k) = 0$ for all $k > p$. Therefore, we need only consider the case where $\alpha_k < 1$ for all k. It is easy to see from Steps 2 and 4 of Algorithm 1 that

$$F_1(x_{k+1},y_{k+1}) = (Mx_k+Ny_k-h)+\alpha_k(M\Delta x_k+N\Delta y_k) = (1-\alpha_k)F_1(x_k,y_k).$$

Since $\alpha_k \to 1$, $\{F_1(x_k,y_k)\}$ componentwise converges to zero Q-superlinearly.

We then show that the sequence $\{F_2(x_k,y_k)\}$ also componentwise converges to zero Q-superlinearly. From Step 4 of Algorithm 1,

$$X_k^{-1}x_{k+1} = e+\alpha_kp_k \quad \text{and} \quad Y_k^{-1}y_{k+1} = e+\alpha_kq_k.$$

Adding the above two equations, we have

$$X_k^{-1}x_{k+1}+Y_k^{-1}y_{k+1} = 2e+\alpha_k(p_k+q_k).$$

It follows from (13) and $\alpha_k \to 1$ that

$$\lim_{k\to\infty}(X_k^{-1}x_{k+1} + Y_k^{-1}y_{k+1}) = e. \tag{15}$$

If $[x_*]_i = 0$, then by strict complementarity, $[y_*]_i > 0$ and $[y_{k+1}]_i/[y_k]_i \to 1$. It follows from (15) that $[x_{k+1}]_i/[x_k]_i \to 0$. Therefore, $[x_k]_i \to 0$ Q-superlinearly. By the symmetry of the relation (15), we have $[y_k]_j \to 0$ Q-superlinearly if $[y_*]_j = 0$. Thus, all variables that converge to zero do so Q-superlinearly. That is, for each index i either

$$\lim_{k\to\infty}\frac{[x_{k+1}]_i}{[x_k]_i} = 0 \quad\text{and}\quad \lim_{k\to\infty}\frac{[y_{k+1}]_i}{[y_k]_i} = 1$$

or

$$\lim_{k\to\infty}\frac{[x_{k+1}]_i}{[x_k]_i} = 1 \quad\text{and}\quad \lim_{k\to\infty}\frac{[y_{k+1}]_i}{[y_k]_i} = 0.$$

In either case, for every index i,

$$\lim_{k\to\infty}\frac{[x_{k+1}]_i[y_{k+1}]_i}{[x_k]_i[y_k]_i} = \lim_{k\to\infty}\frac{[X_{k+1}Y_{k+1}e]_i}{[X_kY_ke]_i} = 0. \tag{16}$$

We have proved that $\{[X_kY_ke]_i\}$ converges to zero Q-superlinearly for every index i. As was mentioned above, the componentwise Q-superlinear convergence of $\{F(x_k, y_k)\}$ implies its Q-superlinear convergence. This completes the proof. □

A key idea in the proof of Theorem 3.1 can be traced back to a 1980 work by Tapia [14]. In Theorem 3 of that paper, Tapia pointed out that an algorithm which at each iteration satisfies the Taylor linearization of the complementarity equation has the property that the variables that converge to zero do so Q-superlinearly. This result assumed strict complementarity and step-length one. Observe that (15) is equivalent to

$$X_kY_ke + Y_k(x_{k+1} - x_k) + X_k(y_{k+1} - y_k) \to 0.$$

We see that the Taylor linearization of complementarity is satisfied asymptotically in our situation.

The following theorem deals with the Q-superlinear convergence of Algorithm 1 applied to LCPs, quadratic programming, and linear programming.

THEOREM 3.3. *Let $\{(x_k, y_k)\}$ be generated by Algorithm* 1 *with $\tau_k \to 1$ and $\sigma_k \to 0$, and let $(x_k, y_k) \to (x_*, y_*)$. Under assumptions* 1 *and* 2 *of Theorem* 3.1, *if $(x_p, y_p) \in \Omega$ for some p, then (x_*, y_*) solves problem* (1) *and the sequence $\{F(x_k, y_k)\}$ componentwise converges to zero Q-superlinearly for the following three cases:*

1. *the linear complementarity problem $(N = -I)$ with M positive semidefinite,*

2. *the quadratic programming problem* (2) *with Q positive semidefinite on the null space of A,*

3. *the linear programming problem* (6).

Proof. We need to prove that assumption 3 of Theorem 3.1 is satisfied for each of the above three cases. Observe that for all $k \geq p$ we have $(x_k, y_k) \in \Omega$ and $M\Delta x_k + N\Delta y_k = 0$ (see (7)). It suffices to prove that $u^Tv \geq 0$ for all $u, v \in \mathbf{R}^n$ satisfying $Mu + Nv = 0$.

In the first case $(N = -I)$, $Mu + Nv = 0$ is equivalent to $v = Mu$. Hence $u^Tv = u^TMu \geq 0$ because M is positive semidefinite.

In the second case (see (5)), $Mu + Nv = 0$ is equivalent to $Au = 0$ and $BQu = Bv$. Using the representations $u = B^Tu_2$ and $v = A^Tv_1 + B^Tv_2$, where $v_1 \in \mathbf{R}^m$ and

$u_2, v_2 \in \mathbf{R}^{n-m}$, and noticing that $A^T \perp B^T$, we have $u^T v = u_2^T BB^T v_2$. Moreover, $BQu = Bv$ is equivalent to $BQB^T u_2 = BB^T v_2$. Hence, if Q is positive semidefinite in the null space of A, then

$$u^T v = u_2^T BB^T v_2 = u_2^T (BQB^T) u_2 \geq 0.$$

The third case follows immediately from the fact that $Q = 0$ is positive semidefinite. □

It is worth noting that feasibility is assumed in Theorem 3.3 but not in Theorem 3.1. It is not clear if assumption 3 of Theorem 3.1 may be satisfied without feasibility. This topic perhaps deserves more study because infeasible starting points are used in most practical implementations.

4. Concluding remarks. The generality of (1) and the perturbed and damped Newton's method viewpoint have enabled us to analyze the local convergence behavior of a class of interior-point algorithms for linear programming, quadratic programming, and LCPs in a unified approach.

We developed a Q-superlinear convergence theory that does not assume any information on the Jacobian matrix at the solution. This theory was used to establish sufficient conditions for Q-superlinear convergence of a class of interior-point algorithms for linear programming, quadratic programming (with Q positive semidefinite on the null space of A), and positive semidefinite LCPs.

Acknowledgment. We thank two anonymous referees for their constructive comments.

REFERENCES

[1] G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963.

[2] C. C. Gonzaga and M. J. Todd, *An $O(\sqrt{n}L)$-iteration large-step primal-dual affine algorithm for linear programming*, SIAM J. Optimization, 2 (1992), pp. 349–359.

[3] M. Kojima, N. Megiddo, and T. Noma, *Homotopy continuation methods for complementarity problems*, Math. Oper. Res., 16 (1991), pp. 754–774.

[4] M. Kojima, S. Mizuno, and T. Noma, *A new continuation method for complementarity problems with uniform p-functions*, Math. Programming, 43 (1989), pp. 107–113.

[5] M. Kojima, S. Mizuno, and A. Yoshise, *An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems*, Math. Programming, 50 (1991), pp. 331–342.

[6] ———, *A primal-dual algorithm for a class of linear complementarity problems*, Math. Programming, 44 (1989), pp. 1-26.

[7] ———, *A primal-dual interior point method for linear programming*, in Progress in Mathematical Programming, Interior-Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 29–47.

[8] I. J. Lustig, *A generic primal-dual interior point algorithm*, Tech. Rep. SOR 88-3, Dept. of Civil Engineering and Operations Research, Princeton University, Princeton, NJ, 1988.

[9] I. J. Lustig, R. E. Marsten, and D. F. Shanno, *Computational experience with a primal-dual interior point method for linear programming*, J. Linear Algebra Appl., 152 (1991), pp. 191–222.

[10] N. Megiddo, *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming, Interior-Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 131–158.

[11] S. Mizuno, M. J. Todd, and Y. Ye, *On adaptive step primal-dual interior-point algorithms for linear programming*, Tech. Rep. 944, School of Operations Research and Industrial Engineering, Cornell Univ., 1989; Math. Oper. Res., to appear.

[12] R. C. Monteiro and I. Adler, *Interior path-following primal-dual algorithms. Part* I: *Linear programming*, Math. Programming, 44 (1989), pp. 27–41.

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

[14] R. A. Tapia, *On the role of slack variables in quasi-Newton methods for constrained optimization*, in Numerical Optimization of Dynamic Systems, L. C. W. Dixon and G. P. Szegö, eds., North-Holland, 1980, pp. 235–246.

[15] M. J. Todd and Y. Ye, *A centered projective algorithm for linear programming*, Math. Oper. Res., 15 (1990), pp. 508–529.

[16] Y. Ye, *Interior point algorithms for quadratic programming*, Working Paper Series No. 89-29, Dept. of Management Sciences, The Univ. of Iowa, Iowa City, IA, 1989; also in Recent Developments in Mathematical Programming, S. Kumar, ed., Gordon & Beach Scientific Publishers, New York, 1991, to appear.

[17] ———, *Interior point algorithms for global optimization*, Ann. Oper. Res., 25 (1990), pp. 59–74.

[18] Y. Zhang, R. A. Tapia, and J. E. Dennis, *On the superlinear and quadratic convergence of primal-dual interior point linear programming algorithms*, SIAM J. Optimization, 2 (1992), pp. 304–324.

SIAM J. OPTIMIZATION
Vol. 3, No. 2, pp. 423–441, May 1993

POINTWISE BROYDEN METHODS*

C. T. KELLEY† AND E. W. SACHS‡

Abstract. Pointwise quasi-Newton methods are designed for nonlinear equations and optimization problems in function spaces. They update coefficients of differential and integral operators and therefore take advantage of finer structure than conventional quasi-Newton methods. In this paper a general theory for those pointwise quasi-Newton methods that are based on Broyden's method are given. This paper unifies the theory of pointwise methods with that for Broyden's method in Hilbert space. A new superlinearly convergent method is introduced for elliptic boundary value problems and the new theory allows for a direct extension of a pointwise method for integral equations.

Key words. pointwise quasi-Newton method, superlinear convergence

AMS subject classifications. 45G10, 47H17, 65J15, 65K10, 65R20

1. Introduction. In this paper we put one class of pointwise quasi-Newton methods into a general framework and give a convergence proof sufficient to describe many of the applications in the literature and extend some of them. Pointwise quasi-Newton methods update coefficients of operators in function spaces. This is in contrast to methods such as Broyden's method, which update the operators themselves by adding low-rank terms. In many cases pointwise updates of this type give superlinear convergence whereas direct extensions of updates derived for finite-dimensional problems do not. The updates considered in this paper are pointwise extensions of Broyden's method [1] and we unify the theory of pointwise methods with that for Broyden's method itself. Pointwise methods that are extensions of rank-two updates were considered in [11] and [13] but are not considered in this paper.

Pointwise methods have been applied to boundary value problems [3], [6], [10], optimal control [11], [13], and integral equations [8], [9]. While the method in [11] is a pointwise extension of the BFGS method, the methods proposed in the remainder of the papers cited above are in one way or another based on pointwise variants of Broyden's method. The focus of this paper is on nonpartitioned forms of pointwise quasi-Newton methods. In [11] and [13] partitioned methods were used to take into account sparsity patterns of the coefficients of differential operators. These methods will be considered in a subsequent paper on partitioned pointwise methods.

In the body of the paper we give several examples, and in this introductory section we give only one to illustrate the idea. Consider the two-point boundary value problem

$$u'' + f(u, u') = 0, \qquad u(0) = u(1) = 0.$$

We write the problem as $F(u) = 0$ on the space

$$X = \{u \mid u \in C^2([0,1]), u(0) = u(1) = 0\}.$$

We assume that there is a solution u^* and that the Fréchet derivative of F is a nonsingular second-order differential operator at u^*.

*Received by the editors November 6, 1991; accepted for publication March 20, 1992.

†Department of Mathematics, North Carolina State University, Box 8205, Raleigh, North Carolina 27695-8205. The research of this author was supported by National Science Foundation grants INT-8800560, DMS-8900410, and DMS-9024622, and Air Force Office of Scientific Research grants AFOSR-ISSA-890044 and AFOSR-FQ8671-9101094.

‡Universität Trier, FB IV–Mathematik, Postfach 3825, 5500 Trier, Germany.

A standard finite element or finite difference discretization of this problem leads to a problem with a tridiagonal Jacobian matrix. As was pointed out in [6] the quasi-Newton iterates generated by use of the standard sparse Broyden or Schubert [15] algorithm do not converge rapidly to the solution. This observation was explained in [10], where it was shown that in the limit the Schubert update converges to an update that only modifies the zeroth-order term in the error in the Fréchet derivative. The update proposed in [6] for the discrete problem and analyzed in [10] for the continuous case updates both the first- and zeroth-order terms of the error in the Fréchet derivative. If the approximate derivative at the current iterate is

$$A_c = \frac{d^2}{dx^2} + a_1^c \frac{d}{dx} + a_0^c$$

and $u_+ = u_c - A_c^{-1} F(u_c)$ is the new quasi-Newton iterate, the update of the coefficients is given by

$$a_0^+ = a_0^c + (y - A_c s)(x) s(x) (s(x)^2 + s'(x)^2)^+$$

and

$$a_1^+ = a_1^c + (y - A_c s)'(x) s'(x) \frac{d}{dx} (s(x)^2 + s'(x)^2)^+ .$$

Here we have used the convention,

$$a^+ = \begin{cases} 1/a & \text{if } a \neq 0, \\ 0 & \text{if } a = 0 \end{cases}$$

for $a \in R$.

Note that this update modifies both unknown coefficients. In [10] it was shown that the iterates produced by this update converge q-superlinearly in C^1 provided the initial iterate is near the solution in the C^1 norm and the initial approximations for the coefficients are near those for $F'(u^*)$ in the uniform norm.

In this paper we put these pointwise updates in an abstract setting and unify several such methods. In §2 we formulate this abstraction. In §3 we state and prove the basic result on superlinear convergence in a weak sense. We apply this result, together with a compactness condition, to obtain q-superlinear convergence results in §4. The results in §4 extend those in [10]. When the compactness condition does not hold, a nonstandard notion of superlinear convergence often describes the performance of the algorithms, and we discuss this in §5. We use the ideas in §5 to extend the work in [8] and [9].

2. Notation and definitions. In this section we introduce pointwise inner product spaces. These are the fundamental objects in our study of pointwise quasi-Newton methods. With the definition of pointwise inner product space in hand we discuss the basic assumptions that relate the pointwise inner product with the nonlinear equation to be solved. All of this structure is used in §3 to define pointwise quasi-Newton updating in a general setting and to prove the basic convergence result.

DEFINITION 2.1. Let $\Omega \subset R^M$ for some $M < \infty$ be compact and let H be a Hilbert space. A Banach space X of H-valued functions on Ω is a *pointwise inner product space* over Ω if there is a map $\mu_X : X \times X \to L^\infty(\Omega)$ satisfying, for all $f, g, h \in X$, $x \in \Omega$, and $\alpha \in R$,

1. $\mu_X(f,g)(x) = \mu_X(g,f)(x)$,
2. $\mu_X(f+\alpha g,h)(x) = \mu_X(f,h)(x) + \alpha\mu_X(g,h)(x)$,
3. $\mu_X(f,f)(x) \geq 0$ and $= 0$ for almost all $x \in \Omega$ if and only if $f = 0$,
4. the norm on X is

$$\|f\|_X = \|\mu_X(f,f)^{1/2}\|_\infty. \tag{2.1}$$

We will call μ_X a pointwise inner product. It will be convenient to define

$$\|f\|_{\mu_X}(x) = \mu_X(f,f)^{1/2}(x)$$

for each $f \in X$. When the dependence of H on the space X is important we will refer to H_X.

An example of such a space is $X = L^\infty(\Omega; H)$, the space of functions on Ω valued in a Hilbert space H with norm

$$\|f\|_X = \text{ess-sup}_{x\in\Omega}(\|f(x)\|_H).$$

Here,

$$\mu_X(f,g)(x) = (f(x),g(x))_H,$$

where $(\cdot,\cdot)_H$ is the inner product on H. Only in simple cases like this is $\mu_H(\cdot,\cdot) = (\cdot,\cdot)_H$. Another choice for μ could be

$$\mu_X(f,g)(x) = (f(x),g(x))_H + \int_\Omega (f(y),g(y))_H \, dy,$$

which would give an equivalent norm. This choice was used in [9] in the context of integral equations. Similarly, the space of continuous H-valued functions could also have the structure above, as it is a closed subspace of $L^\infty(\Omega; H)$. For the Hart–Soul update considered in the introduction we formulate the problem/method pair so that $H = R^1$,

$$\mu_X(f,g)(x) = f(x)g(x) + f'(x)g'(x),$$

and $X = C^1$.

Spaces of differential or integral operators can be viewed as pointwise inner product spaces. This was done in [10]. Consider the space of operators on $C^k(\Omega; R^N)$ of the form

$$Lu = \sum_{|\alpha|\leq k} a_\alpha(x)D^\alpha$$

with continuous $N \times N$ matrix coefficients $\{a_\alpha\}$. This is a pointwise inner product space with

$$\mu(L_1,L_2)(x) = \sum_{|\alpha|\leq k} (a^1_\alpha(x), a^2_\alpha(x))_{R^{N\times N}},$$

and $H = R^{N\times N\times k}$. $C^k(\Omega; R^N)$ is also a pointwise inner product space with

$$\mu_X(u,v) = \sum_{|\alpha|\leq k} ((D^\alpha u),(D^\alpha v))_{R^N}.$$

The space of integral operators of the form

$$Lu(x) = m(x)u(x) + \int_\Omega k(x,y)u(y)\,dy$$

on $C(\Omega; R^N)$ with m continuous was made into a pointwise inner product space in [9] with

$$\mu_X(L_1, L_2) = (m_1(x), m_2(x))_{R^{N\times N}} + \int_\Omega (k_1(x,y), k_2(x,y))_{R^{N\times N}}\,dy.$$

Let X and Y be pointwise inner product spaces over $\Omega \subset R^N$ with corresponding pointwise inner products μ_X and $\mu_Y = (\cdot,\cdot)_{H_Y}$. In this paper Y is an intermediate space but its structure as a pointwise inner product space is crucial to the analysis. We consider nonlinear equations of the form

$$F(u) = 0, \tag{2.2}$$

where $F : X \to X$. We assume that F', the Fréchet derivative of F, has the form

$$F'(u) = J_C(u) + C^P J_A(u). \tag{2.3}$$

In (2.3) J_C is a Lipschitz continuously differentiable map from X to $\mathcal{L}(X)$, J_A a Lipschitz continuously differentiable map from X to $\mathcal{L}(X, Y)$, and C^P a bounded linear map from Y to X. The idea is that J_C and C^P will be computed and J_A will be approximated by a quasi-Newton method. The map C^P will be viewed as a preconditioner and is used for the most part as a theoretical artifice to make F a map from X to X. We discuss the reasons for this in the context of the applications in the following sections, but here we note that in the case of the Hart–Soul update,

$$C^P = \left(\frac{d^2}{dx^2}\right)^{-1},$$

$J_C = I$, and J_A is the first-order part of the Fréchet derivative. Multiplication of the equation by C^P on the left does not change the iterates and is not done in practice. It does, as we shall see, assist in the analysis.

The iteration will take the form

$$u_+ = u_c - B_c^{-1} F(u_c)$$

where $B_c = C_c + C_c^P A_c$. Here C_c and C_c^P are approximations to $J_C(u^*)$ and C^P that are computed by means other than quasi-Newton methods. As in [7] we do not explicitly address how C_c and C_c^P are computed, but if one uses $C_c^P = C^P$ and $C_c = J_C(u_c)$ then all the conditions we put on these maps to insure superlinear convergence are satisfied. We illustrate this through several examples in §§4 and 5.

This is an extension of the type of splitting used in [2]. The new feature is the inclusion of the preconditioning map C^P. The motivation for the imposition of this structure on F' will become clear in §4 when specific pointwise quasi-Newton methods are discussed.

We make the *standard assumptions* on F.

ASSUMPTION 2.1. There is $u^* \in X$ such that $F(u^*) = 0$, $F'(u^*)$ is nonsingular, and F' is Lipschitz continuous in a neighborhood $\mathcal{N}$ of u^* with Lipschitz constant γ.

Basic to all pointwise methods is the notion of a generalized rank-one operator. For any measurable H_Y-valued function u on Ω and functions $v, w \in X$ define

$$i(u,v)w = \mu_X(v,w)u. \tag{2.4}$$

Since v and w are in X and u is a measurable H_Y-valued function, $i(u,v)w$ is a measurable H_Y-valued function. Note that if ξ is a scalar-valued function on Ω,

$$(i(\xi u,v)w)(x) = \xi(x)(i(u,v)w)(x) \tag{2.5}$$

for all $x \in \Omega$.

We assume that our initial approximation A_0 to $J_A(u^*)$ differs from $J_A(u^*)$ by an error E^A which lies in a linear space of admissible error operators, $\mathcal{E} \subset \mathcal{L}(X,Y)$. We assume that $\mathcal{E}$ is a Banach space with norm $\|\cdot\|_{\mathcal{E}}$ and we make the following assumption.

ASSUMPTION 2.2. For all $x \in \Omega$, $u \in X$, and $E \in \mathcal{E}$,

$$\|(Eu)(x)\|_{H_Y} = \|Eu\|_{H_Y}(x) \leq \|E\|_{\mathcal{E}}\|u\|_{\mu_X}(x).$$

In the case of the Hart–Soul update $\mathcal{E}$ is the space of first-order differential operators with coefficients in L^∞. The norm is the sum of the norms of the two coefficients. In this case Assumption 2.2 is clearly true. A trivial but important consequence of Assumption 2.2 is that for all $E \in \mathcal{E}$, $x \in \Omega$, and $u \in X$,

$$\|E\|_{\mathcal{L}(X,Y)} \leq \|E\|_{\mathcal{E}}. \tag{2.6}$$

We apply (2.6) and estimates on the errors

$$E^C = C - J_C(u^*) \quad \text{and} \quad E^A = A - J_A(u^*)$$

to prove the following simple lemma on invertibility of operators B of the form

$$B = C + \tilde{C}^P A \tag{2.7}$$

where

$$\tilde{C}^P = C^P + E^P.$$

LEMMA 2.2. *There is $\epsilon_0 > 0$ and $\delta \in C[0,\epsilon_0)$ with $\delta(0) = 0$ such that if $\epsilon \in [0,\epsilon_0)$, $A \in \mathcal{L}(X,Y)$ with $E^A \in \mathcal{E}$,*

$$\|E^A\|_{\mathcal{E}} < \epsilon,$$

$\|E^C\|_{\mathcal{L}(X)} < \epsilon$, $\|E^P\|_{\mathcal{L}(Y,X)} < \epsilon$, *and B is given by* (2.7), *then B^{-1} exists and*

$$\|B^{-1} - F'(u^*)^{-1}\|_{\mathcal{L}(X)} < \delta(\epsilon).$$

Proof. By (2.6) the result is a consequence of the Banach lemma since

$$\begin{aligned} B - F'(u^*) &= E^C + \tilde{C}^P A - C^P F'_A(u^*) \\ &= E^C + (\tilde{C}^P - C^P)A + C^P(A - J_A(u^*)) \\ &= E^C + E^P A + C^P E^A, \end{aligned}$$

and so

$$\|F'(u^*) - B\|_{\mathcal{L}(X)} \leq q(\epsilon)$$

where

$$q(\epsilon) = \epsilon(1 + \epsilon + \|J_A(u^*)\|_{\mathcal{L}(X,Y)} + \|C^P\|_{\mathcal{L}(Y,X)}).$$

Therefore, the result holds with

$$\delta(\epsilon) = \frac{\|F'(u^*)^{-1}\|_{\mathcal{L}(X)}}{1 - \|F'(u^*)^{-1}\|_{\mathcal{L}(X)}q(\epsilon)},$$

as is standard. □

We make the following assumption to relate generalized rank-one operators to the error class.

ASSUMPTION 2.3. Let $s \in X$. For all $u \in X$ and $y \in Y$

$$i(y, u) \in \mathcal{E},$$

and for all $x \in \Omega$

$$\|i(y, u)\|_{\mathcal{E}}(x) \leq \|y\|_{H_Y}(x)\|u\|_{\mu_X}(x). \tag{2.8}$$

For all $E \in \mathcal{E}$

$$(P_s E) = \mu_X(s, s)^+ i(Es, s) \in \mathcal{E} \tag{2.9}$$

and

$$\|P_s\|_{\mathcal{L}(\mathcal{E})} \leq 1 \quad \text{and} \quad \|I - P_s\|_{\mathcal{L}(\mathcal{E})} \leq 1. \tag{2.10}$$

Note that (2.9) and the first half of (2.10) follow from (2.8). We include all of them in the assumption in order to collect all the facts on P_s in one place. At this point it becomes somewhat nontrivial to check the assumptions. The verification for the Hart–Soul update was done in [10] and will be placed in a more general context later in the present paper.

Assumption 2.3 is the critical coupling relation between the error class and the space on which the nonlinear equation is defined. In the following section we show how the assumptions in this section lead to superlinear convergence results and apply these to the Broyden method itself as well as to extensions of the method for elliptic boundary value problems proposed in [6] and analyzed in [10].

3. Basic results.

In this section we prove a weak superlinear convergence result from which the convergence results for specific methods in §4 will follow.

Pointwise Broyden updates take the form

$$A_+ = A_c + \mu_X(s, s)^+ i(y^\# - A_c s, s), \tag{3.1}$$

where $y^\#$ is selected to enforce various extensions of the secant condition $B_+ s = y = F(u_+) - F(u_c)$. Because the details of the formation of $y^\#$ are varied, we give an hypothesis on $y^\#$ that will be verified in the subsequent sections in the context of applications.

ASSUMPTION 3.1. There is a neighborhood $\mathcal{N}$ of u^* and $\delta > 0$ such that for all $u_c, u_+ = u_c + s \in \mathcal{N}$ there are $C_\Delta > 0$ and $y^\# \in Y$ such that for each $x \in \Omega$,

$$\|y^\# - J_A(u^*)s\|_{H_Y}(x) \leq C_\Delta(\|e_c\|_X + \|e_+\|_X)\|s\|_{\mu_X}(x).$$

Define an operator Δ_s by

$$\Delta_s = P_s\Delta_s = \mu_X(s,s)^+ i(y^\# - J_A(u^*)s, s). \tag{3.2}$$

Upon noting that Δ_s is in $\mathcal{E}$ by Assumption 2.3 we obtain the following trivial but important lemma, the proof of which is a direct analog of the finite-dimensional analysis in [2].

LEMMA 3.1. *Let Assumptions* 2.1, 2.2, 2.3, *and* 3.1 *hold, and let* $u_c, u_+ \in \mathcal{N}$. *If* $E_c^A \in \mathcal{E}$ *then* $E_+^A \in \mathcal{E}$ *and*

$$E_+^A = (I - P_s)E_c^A + \Delta_s \in \mathcal{E} \tag{3.3}$$

with

$$\|\Delta_s\|_{\mathcal{E}} \leq C_\Delta(\|e_+\|_X + \|e_c\|_X). \tag{3.4}$$

From this lemma we obtain a bounded deterioration inequality,

$$\|E_{n+1}^A\|_{\mathcal{E}} \leq \|E_n^A\|_{\mathcal{E}} + C_\Delta(\|e_{n+1}\|_X + \|e_n\|_X),$$

and therefore q-linear convergence in the X-norm in the standard way.

COROLLARY 3.2. *Assume that the assumptions of Lemmas* 3.1 *and* 2.2 *hold. Then for all* $\sigma \in (0,1)$ *there is* δ *such that if for all* n

$$\|E_n^C\|_{\mathcal{L}(X)} < \delta, \qquad \|C_n^P - C^P\|_{\mathcal{L}(Y,X)} < \delta,$$

and $\|e_0\|_X < \delta$, *and* $E_0^A \in \mathcal{E}$ *such that* $\|E_0^A\|_{\mathcal{E}} < \delta$, *then the pointwise Broyden iterates specified by the update formula* (3.1) *converge q-linearly to* u^* *in the norm of* X *with q-factor* σ *and the derivative errors* $\|E_n\|_{\mathcal{L}(X)}$ *and* $\|E_n^A\|_{\mathcal{E}}$ *are bounded.*

In the setting of traditional quasi-Newton methods one can obtain superlinear convergence by showing that the Dennis–Moré condition

$$\lim_{n\to\infty} \frac{\|E_n s_n\|_X}{\|s_n\|_X} = 0$$

holds. In the case of infinite-dimensional spaces verification of this condition requires additional assumptions and sometimes is not possible. One extension of this condition which can be verified in our context is given in the main result of this section.

This theorem requires a preliminary lemma and a compatibility assumption on the class $\mathcal{E}$ with an equivalence relation $\sim$. For $\bar{x} \in \Omega$ we equip X with the semi-inner product $\mu_X(\cdot,\cdot)(\bar{x})$. The space of equivalence classes under the relation

$$u \sim v \quad \text{if } \mu_X(u-v, u-v)(\bar{x}) = \|u - v\|_{\mu_X}^2(\bar{x}) = 0,$$

is a pre-Hilbert space under the inner product $\mu_X(\cdot,\cdot)(\bar{x})$, and we denote its completion by $\bar{X}$. In the discussion that follows we will identify an element of the class with a representor. We can relate $E \in \mathcal{E}$ to a map on $\bar{X}$ if we make the following assumption.

ASSUMPTION 3.2 For all $E \in \mathcal{E}$, $u, v \in X$, and $\bar{x} \in \Omega$ $\|u - v\|_{\mu_X}(\bar{x}) = 0$ implies $Eu(\bar{x}) = Ev(\bar{x})$.

In the simple example of the Hart–Soul update discussed in the introduction, $u \sim v$ means that u and v agree to first order at $\bar{x}$. The assumption asserts that if u and v agree to first order at $\bar{x}$ then any first-order operator applied to them will give the same value. We show how this assumption holds in the discussion of applications in §4.

We have the following lemma.

LEMMA 3.3. *Let Assumptions* 2.2 *and* 3.2 *hold and let* $\bar{x} \in \Omega$ *be given. Then the map from* $\bar{X}$ *to* H_Y *defined by*

$$\bar{E}f = (Ef)(\bar{x}) \quad \textit{for all } f \in X$$

is a bounded linear map from $\bar{X}$ *to* H_Y *with*

$$\|\bar{E}\|_{\mathcal{L}(\bar{X}, H_Y)} \leq \|E\|_{\mathcal{E}}. \tag{3.5}$$

Proof. The map $\bar{E}$ is well defined by Assumption 3.2. It is a continuous linear map by Assumption 2.2. In fact, Assumption 2.2 implies that

$$\|\bar{E}f\|_{H_Y} = \|(Ef)(\bar{x})\|_{H_Y} \leq \|E\|_{\mathcal{E}} \|f\|_{\mu_X}(\bar{x}) = \|E\|_{\mathcal{E}} \|f\|_{\bar{X}}.$$

This completes the proof. □

THEOREM 3.4. *Let the assumptions of Corollary* 3.2 *hold and let* $\delta > 0$ *correspond to some* $\sigma \in (0,1)$. *Then for each* $x \in \Omega$ *and* $\phi \in H_Y$,

$$\lim_{n \to \infty} \mu_X(s_n, s_n)^+(x)(\phi, (E_n^A s_n)(x))^2_{H_Y} = 0. \tag{3.6}$$

Moreover, there is M_A *such that*

$$\sup_{x, y \in \Omega} |\mu_X(s_n, s_n)^+(x)(\phi, E_n^A s_n)^2_{H_Y}(y)| < M_A \|\phi\|^2_Y \tag{3.7}$$

for all $n \geq 0$.

Proof. We fix $\bar{x} \in \Omega$ throughout the proof. Given a step $s = u_+ - u_c$, let

$$\bar{\mu} = \mu_X(s, s)^+(\bar{x}).$$

Hence from (3.3) and the fact that $P_s \Delta_s = \Delta_s$ we obtain for $f \in X$

$$\bar{E}_+^A f = (E_+^A f)(\bar{x}) = (E_c^A f)(\bar{x}) - (P_s E_c^A f)(\bar{x}) + (P_s \Delta_s f)(\bar{x}) = (\bar{E}_c^A) f - R_s f$$

where $R_s \in \mathcal{L}(\bar{X}, H_Y)$ is given by

$$R_s f = (P_s(E_c^A - \Delta_s) f)(\bar{x}) = \bar{\mu} \mu_X(s, f)(\bar{x})((E_c^A - \Delta_s)s)(\bar{x}).$$

We can compute the adjoint $R_s^* \in \mathcal{L}(H_Y, \bar{X}^*)$ with respect to the inner products $\mu_X(\cdot,\cdot)(\bar{x})$ on $\bar{X}$ and $(\cdot,\cdot)_{H_Y}$ on H_Y:

$$\begin{aligned} (\phi, R_s f)_{H_Y} &= \bar{\mu} \mu_X(s, f)(\bar{x})(\phi, ((E_c^A - \Delta_s)s)(\bar{x}))_{H_Y} \\ &= \mu_X(R_s^* \phi, f)(\bar{x}), \quad \phi \in H_Y, \quad f \in X, \end{aligned}$$

so that for $\phi \in H_Y$

$$R_s^* \phi = \bar{\mu}(\phi, ((E_c^A - \Delta_s)s)(\bar{x}))_{H_Y} s.$$

Similarly, the adjoint $(\bar{E}_c^A)^* \in \mathcal{L}(H_Y, \bar{X}^*)$ satisfies

$$\mu_X((\bar{E}_c^A)^*\phi, s)(\bar{x}) = (\phi, (\bar{E}_c^A)s)_{H_Y} = (\phi, (E_c^A s)(\bar{x}))_{H_Y}.$$

Hence we obtain the following important identity. For any $\phi \in H_Y$

$$\begin{aligned}\mu_X((\bar{E}_+^A)^*\phi, (\bar{E}_+^A)^*\phi)(\bar{x}) &= \mu_X((\bar{E}_c^A)^*\phi, (\bar{E}_c^A)^*\phi)(\bar{x}) \\ &\quad -2\mu_X((\bar{E}_c^A)^*\phi, R_s^*\phi)(\bar{x}) + \mu_X(R_s^*\phi, R_s^*\phi)(\bar{x}) \\ \\ &= \mu_X((\bar{E}_c^A)^*\phi, (\bar{E}_c^A)^*\phi)(\bar{x}) \\ &\quad -2\bar{\mu}(\phi, ((E_c^A - \Delta_s)s)(\bar{x}))_{H_Y}\mu_X((\bar{E}_c^A)^*\phi, s)(\bar{x}) \\ &\quad +\bar{\mu}^2(\phi, ((E_c^A - \Delta_s)s)(\bar{x}))_{H_Y}^2\mu_X(s, s)(\bar{x}) \\ \\ &= \mu_X((\bar{E}_c^A)^*\phi, (\bar{E}_c^A)^*\phi)(\bar{x}) - \bar{\mu}(\phi, (E_c^A s)(\bar{x}))_{H_Y}^2 \\ &\quad +\bar{\mu}(\phi, (\Delta_s s)(\bar{x}))_{H_Y}^2.\end{aligned}$$

So for $n = 0, 1, \ldots,$ if we let $P_n = P_{s_n}$ and $\Delta_n = \Delta_{s_n}$ we have
(3.8)

$$\begin{aligned}\textstyle\sum_{k=0}^n \mu_X(s_k, s_k)^+(\bar{x})(\phi, (E_k^A s)(\bar{x}))_{H_Y}^2 &\leq \mu_X((\bar{E}_0^A)^*\phi, (\bar{E}_0^A)^*\phi)(\bar{x}) \\ &\quad + \textstyle\sum_{k=0}^n \mu_X(s_k, s_k)^+(\bar{x})(\phi, (\Delta_k s_k)(\bar{x}))_{H_Y}^2.\end{aligned}$$

As $\bar{x} \in \Omega$ is arbitrary and Lemma 3.3 implies that

$$\mu_X((\bar{E}_0^A)^*\phi, (\bar{E}_0^A)^*\phi)(\bar{x}) \leq \|E\|_{\mathcal{E}}^2 \|\phi\|_{H_Y}^2$$

independently of $\bar{x}$, the proofs of (3.6) and (3.7) will be complete if we can estimate the sum on the right side of (3.8) independently of both $\bar{x}$ and k. To do this note that Assumption 2.2 and (3.4) imply that

$$\begin{aligned}(\mu_X(s_k, s_k)^+(\bar{x}))^{\frac{1}{2}}(\phi, (\Delta_k s_k)(\bar{x}))_{H_Y} &\leq (\mu_X(s_k, s_k)^+(\bar{x}))^{\frac{1}{2}} \|(\Delta_k s_k)(\bar{x})\|_{H_Y} \|\phi\|_{H_Y} \\ &\leq \|\Delta_k\|_{\mathcal{E}} \|\phi\|_{H_Y} \leq C_\Delta \|\phi\|_{H_Y} (\|e_k\|_X + \|e_{k+1}\|_X) \\ &\leq C_\Delta \|e_0\|_X \|\phi\|_{H_Y} (\sigma^k + \sigma^{k+1}) \leq 2C_\Delta \delta \|\phi\|_{H_Y} \sigma^k.\end{aligned}$$

Hence the series

$$\sum_{k=0}^{\infty} \mu_X(s_k, s_k)^+(\bar{x})(\phi, (\Delta_k s_k)(\bar{x}))_{H_Y}^2 \leq \frac{4(C_\Delta \|\phi\|_{H_Y} \delta)^2}{1 - \sigma^2}$$

converges. Setting

$$M_A = \frac{4(C_\Delta \delta)^2}{1 - \sigma^2} + \|E\|_{\mathcal{E}}^2$$

completes the proof. □

4. Superlinear convergence in the conventional sense. Consequences of Theorem 3.4 depend on properties of the nonlinear equation. In this section the problems satisfy a stronger assumption than Assumption 3.1.

ASSUMPTION 4.1. F can be written $F = F_C + C^P F_A$ with $J_C(u) = F'_C(u)$ and $J_A(u) = F'_A(u)$. There is a neighborhood $\mathcal{N}$ of u^* such that for all $u, v \in \mathcal{N}$

$$\begin{aligned} &\text{(a)} \quad F'_A(u) - F'_A(v) \in \mathcal{E} \quad \text{for all } u, v \in X \quad \text{and} \\ &\text{(b)} \quad \|F'_A(u) - F'_A(v)\|_{\mathcal{E}} \leq \gamma_{\mathcal{E}} \|u - v\|_X. \end{aligned} \tag{4.1}$$

We note that Assumption 4.1 implies Assumption 3.1 in the following lemma.

LEMMA 4.1. *If Assumption* 4.1 *holds and*

$$y^{\#} = F_A(u_+) - F_A(u_c),$$

then Assumption 3.1 *holds.*

The standard situation considered in the literature on pointwise quasi-Newton methods [3], [6], [8], [9], [10], [11], [13] is one for which H_Y is finite-dimensional. In that case (3.6) becomes

$$\lim_{n\to\infty} \mu_X(s_n, s_n)^+(x) \|E_n^A s_n(x)\|_{H_Y}^2 = 0. \tag{4.2}$$

In the case where $X = Y = H = R^N$, $F = F_A$, and Ω is a single point, (4.2) is the Dennis–Moré condition. In the case where $X = Y = H$ is infinite-dimensional, Ω is a single point, $F = F_A$, and $\mu_X(\cdot,\cdot) = (\cdot,\cdot)_H$, (3.6) is the weak superlinear convergence condition described in [5], [7], and [12]:

$$\lim_{n\to\infty} \left(\phi, \frac{E_n s_n}{\|s_n\|_H^2}\right) = 0. \tag{4.3}$$

Equation (4.3) is the basis for the convergence analysis of Broyden's method in infinite-dimensional spaces done in [14], [12], and [7]. We illustrate this with part of the next theorem.

The transition from weak to norm superlinear convergence depends on the structure of the particular problem. Broyden's method itself has been analyzed from this point of view in Hilbert space [12] and in Banach space [7]. In this section we prove a theorem on superlinear convergence of pointwise methods. We show how the conventional formulation of Broyden's method can be described in this setting. In the following section we discuss another notion of superlinear convergence that is relevant when the compactness conditions necessary for superlinear convergence in the traditional sense do not hold.

Let X be a pointwise inner product space and define X^p to be the completion of X in the norm

$$\|u\|_{X^p} = \left(\int_\Omega \|u\|_{\mu_X}^p \, dx\right)^{\frac{1}{p}}.$$

Note that $X = X^\infty$. Our first result is an extension of the result in [10] and a unification of that result with the superlinear convergence analysis for Broyden's method in Hilbert space given in [12].

THEOREM 4.2. *Assume that either*

- H_Y *is finite-dimensional, or*
- $X = Y = H$, *Ω is a single point, and* $\mu_X(\cdot,\cdot) = (\cdot,\cdot)_H$. *In addition, let the hypotheses for Theorem* 3.4 *hold, and let* $\delta > 0$ *correspond to some* $\sigma \in (0,1)$ *as in Corollary* 3.2.

Assume that the computed parts of F' satisfy

$$\lim_{n\to\infty} \frac{\|E_n^C s_n\|_X}{\|s_n\|_X} = 0 \tag{4.4}$$

and

$$\lim_{n\to\infty} \|E_n^P\|_{\mathcal{L}(Y,X)} = 0. \tag{4.5}$$

Also assume that $F'(u^*)^{-1}C^P$ can be extended to be a map in $\mathcal{L}(Y^{p_1}, X^{p_2})$ for some $1 \leq p_1 < \infty$ and $1 \leq p_2 \leq \infty$. Finally, assume that for every $M > 0$ the family

$$\{F'(u^*)^{-1}C^P E \,|\, E \in \mathcal{E},\ \|E\|_{\mathcal{E}} \leq M\} \subset \mathcal{L}(X)$$

is collectively compact. Then $u_n \to u^*$ q-superlinearly in the norm of X.

Proof. The goal in the proof will be to show that

$$\lim_{n\to\infty} \frac{\|E_n^A s_n\|_X}{\|s_n\|_X} = 0. \tag{4.6}$$

Equations (4.4), (4.5), and (4.6) together imply that the Dennis–Moré condition

$$\lim_{n\to\infty} \frac{\|E_n s_n\|_X}{\|s_n\|_X} = 0,$$

which implies q-superlinear convergence since the assumptions of this theorem already imply q-linear convergence.

Since $B_n s_n = -F(u_n)$, for each $x \in \Omega$ we have

$$E_n s_n = (B_n - F'(u^*))s_n = -F(u_n) - F'(u^*)(e_{n+1} - e_n).$$

Therefore,

$$\begin{aligned} E_n s_n &= -F'(u^*)e_{n+1} - F(u_n) + F'(u^*)e_n \\ &= -F'(u^*)e_{n+1} + \textstyle\int_0^1 (F'(u^*) - F'(u^* + te_n))e_n\, dt. \end{aligned}$$

Since

$$\begin{aligned} E_n &= E_n^C + (C_n^P A_n - C^P F_A'(u^*)) \\ &= E_n^C + (C_n^P - C^P)A_n + C^P(A_n - F_A'(u^*)) \\ &= E_n^C + E_n^P A_n + C^P E_n^A, \end{aligned}$$

we have that

$$\|C^P E_n^A s_n - E_n s_n\|_X \leq \tau_n \|s_n\|_X \tag{4.7}$$

where

$$\tau_n = M_B \|E_n^P\|_{\mathcal{L}(Y,X)} + \frac{\|E_n^C s_n\|}{\|s_n\|_X} \to 0.$$

Here M_B is a bound on the sequence $\{\|A_n\|_{\mathcal{L}(X,Y)}\}$, which exists by Corollary 3.2.

By Assumptions 2.2 and 2.1 for each $x \in \Omega$

$$\left\| \int_0^1 (F'(u^*) - F'(u^* + te_n))e_n \, dt \right\|_X \leq \frac{\gamma}{2} \|e_n\|_X^2.$$

Hence

$$\|C^P E_n^A s_n + F'(u^*)e_{n+1}\|_Y \leq \frac{\gamma \varepsilon}{2} \|e_n\|_X^2 + \tau_n \|s_n\|_X. \tag{4.8}$$

We now consider the case of finite-dimensional H_Y. Let $\alpha_n \in L^\infty(\Omega)$ be given by

$$\alpha_n = \sqrt{\mu_X(s_n, s_n) + (E_n^A s_n, E_n^A s_n)_{H_Y}}. \tag{4.9}$$

By Theorem 3.4 the finite-dimensionality of H_Y implies that (4.2) holds and hence $\alpha_n \to 0$ for each $x \in \Omega$; therefore, $\alpha_n \to 0$ in $L^p(\Omega)$ for all $1 \leq p < \infty$ by the dominated convergence theorem. In particular, $\alpha_n \to 0$ in $L^{p_1}(\Omega)$. So for each $x \in \Omega$,

$$\|E_n^A s_n\|_{H_Y}(x) \leq \alpha_n(x) \|s_n\|_{\mu_X}(x) \leq \alpha_n(x) \|s_n\|_X,$$

and therefore the sequence

$$\{\xi_n\} = \left\{ \frac{E_n s_n}{\|s_n\|_X} \right\}$$

is uniformly bounded (and hence is contained in Y) and converges both pointwise to 0 in Ω and also to 0 in the norm of Y^{p_1}.

Since $\|E_n^A\|_{\mathcal{E}}$ is uniformly bounded, the collective compactness assumption implies that the sequence $\{\zeta_n\}$ given by

$$\zeta_n = F'(u^*)^{-1} C^P E_n^A (s_n / \|s_n\|_X) \tag{4.10}$$

has an X-norm convergent subsequence. The limit of any such sequence must be zero as

$$\frac{E_n^A s_n}{\|s_n\|_X} \to 0$$

in the space Y^{p_1} and hence $\zeta_n \to 0$ in X^{p_2} by the hypothesis that $F'(u^*)^{-1}C^P$ can be extended to be a map in $\mathcal{L}(Y^{p_1}, X^{p_2})$. Therefore, $\zeta_n \to 0$ in X.

For the case $X = Y = H$, Ω a single point, and $\mu_X(\cdot,\cdot) = (\cdot,\cdot)_H$, we also seek to show $\zeta_n \to 0$. The approach is only a little different and the argument we give here is taken directly from [12]. We include it to show how the result given here unifies pointwise and conventional quasi-Newton methods. In the present case (4.3) holds, $\{\xi_n\}$ is bounded, and $\{\xi_n\}$ therefore converges weakly to 0. The collective compactness assumption implies that $\{\zeta_n\}$ has an X-norm convergent subsequence which must have limit zero by the weak convergence. Hence $\zeta_n \to 0$ in X.

In either case we can use (4.8) and obtain

$$\begin{aligned} \|e_{n+1}\|_{\mu_X} &\leq \|F'(u^*)^{-1} C^P E_n^A s_n\|_X + \|F'(u^*)^{-1}\|_{\mathcal{L}(X)} (\tfrac{\gamma}{2} \|e_n\|_X^2 + +2\tau_n \|e_n\|_X) \\ &\leq \|\zeta_n\|_X \|s_n\|_X + \|F'(u^*)^{-1}\|_{\mathcal{L}(X)} (\tfrac{\gamma}{2} \|e_n\|_X^2 + 2\tau_n \|e_n\|_X) \qquad (4.11) \\ &\leq \|\zeta_n\|_X (1 + \sigma) \|e_n\|_X + \|F'(u^*)^{-1}\|_{\mathcal{L}(X)} (\tfrac{\gamma}{2} \|e_n\|_X^2 + 2\tau_n \|e_n\|_X). \end{aligned}$$

Hence,

$$\|e_{n+1}\|_X \leq \chi_n \|e_n\|_X,$$

where

$$\chi_n = \|F'(u^*)^{-1}\|_{\mathcal{L}(X)} \left(\frac{\gamma\varepsilon}{2}\|e_n\|_X + 2\tau_n\right) + (1+\sigma)\|\zeta_n\|_X.$$

As $\chi_n \to 0$ as $n \to \infty$, the proof is complete. □

The second case of Theorem 4.2 is the main result of [12], which we state directly as a corollary. This is the special case $F_C = 0$, $C^P = I$, Ω a single point, and $X = Y = H$.

COROLLARY 4.3. *Let F be a Lipschitz continuously differentiable map from a Hilbert space H with $F(u^*) = 0$ and $F'(u^*)$ nonsingular. Then there is $\delta > 0$ such that if $\|u_0 - u^*\|_H < \delta$, $\|B_0 - F'(u^*)\|_{\mathcal{L}(H)} < \delta$, and $B_0 - F'(u^*) \in \mathcal{COM}(H)$, then the Broyden iterates converge superlinearly to u^*.*

The first case of Theorem 4.2 allows us to extend the results of [10]. In [10] systems of semilinear second-order elliptic partial differential equations were considered:

$$G(u) = \nabla^2 u + f(x, u(x), \nabla u(x)) = 0,$$

subject to linear Dirichlet boundary conditions on a set $\Omega \subset R^M$; an approximation to the Fréchet derivative of the form

$$B = \nabla^2 + \sum_{j=1}^{M} a_j(x)\frac{\partial}{\partial x_j} + a_0(x)$$

was maintained. The $N \times N$ matrices a_j were intended to approximate the first- and zeroth-order terms in the linear differential operator $G'(u^*)$. The update proposed for two-point boundary value problems in [6] and extended to elliptic systems in [10] can be expressed in terms of the pointwise inner product

$$\mu_X(u, v) = \sum_{j=0}^{M} \frac{\partial u}{\partial x_j}(x)\frac{\partial v}{\partial x_j}(x)$$

in exactly the form

$$B_+ = B_c + i((y - B_c s), s).$$

The main result in [10] was that the update was locally q-superlinearly convergent in the topology of C^1 if $u_0 \in C^2$ was close to u^* in the C^1 norm and the coefficients of B_0 were uniformly close to those of $G'(u^*)$. The update as expressed above does not fit into the precise scope of this paper, as ∇^2 is not defined on $X = C^1$. We obtain the same iterates, however, if we consider the map $F = \nabla^{-2}G$. We have

$$F(u) = u + \nabla^{-2} f(x, u, \nabla u).$$

This is of the form (2.3) with $X = C^1$, $F_C(u) = u$, $C^P = \nabla^{-2}$ with homogeneous Dirichlet boundary conditions, and $F_A(u)(x) = f(x, u(x), \nabla u(x))$. The equation $F(u) = 0$ may be viewed as a weak form of $G(u) = 0$.

We can apply Theorem 4.2 to extend the result in [10]. We consider equations of the form

$$Lu + f(x, u, D^{\alpha_1}u, D^{\alpha_2}u, \ldots, D^{\alpha_m}u) = 0 \tag{4.12}$$

on a bounded domain $\Omega \subset R^M$. Here L is a linear differential operator of order k, f is a smooth nonlinear function of $x \in R^M$ and at most $m \times \sum_{j=1}^{k-1} M^j$ vector variables in R^N. Here $\{\alpha_j\}$ are multi-indices of partial derivatives. We impose linear homogeneous boundary conditions which we write as $B(u) = 0$. We define

$$\mu_X(u,v) = \sum_{|\alpha|\leq k-1} ((D^\alpha u)(x), (D^\alpha v)(x))_{R^N} \tag{4.13}$$

and let $X = C^{k-1}$. We consider the equation in the form

$$F(u) = u + L^{-1} f(x, u, D^{\alpha_1}u, D^{\alpha_2}u, \ldots, D^{\alpha_M}u) = 0. \tag{4.14}$$

In (4.14) L^{-1} denotes the solution operator for $Lu = g$ with the boundary conditions $B(u) = 0$. We let the space $\mathcal{E}$ denote the space of linear differential operators of order $k-1$ with coefficients in L^∞ and write $E \in \mathcal{E}$ as

$$E = \sum_{|\alpha|\leq k-1} a_\alpha(x) D^\alpha \in \mathcal{E}. \tag{4.15}$$

We make $\mathcal{E}$ a pointwise inner product space by defining the pointwise inner product for $x \in \Omega$ by

$$(E_1, E_2)_{\mu_\mathcal{E}}(x) = \sum_{|\alpha|\leq k-1} (a_\alpha^1(x), a_\alpha^2(x))_F.$$

Here $\|\cdot\|_F$ denotes the Frobenius norm. So

$$\|E\|_\mathcal{E} = \sup_{x\in\Omega} \|E\|_{\mu_\mathcal{E}}(x). \tag{4.16}$$

We have the following theorem.

THEOREM 4.4. *Assume that f is Lipschitz continuously differentiable. Assume that a solution u^* exists and that $F'(u^*)$ is nonsingular. Assume that L^{-1} is a bounded operator from $L^\infty(\Omega; R^N)$ to $C^{k-1}(\Omega; R^N)$ and from $L^2(\Omega; R^N)$ to $H^k(\Omega; R^N)$. Let F be given by* (4.14) *with μ_X given by* (4.13)*, $X = C^{k-1}(\Omega; R^N)$, $\mathcal{E}$ the space of linear differential operators of order $k-1$ with coefficients in L^∞, and $\|E\|_\mathcal{E}$ given by* (4.16).

Then if $H_X = H_Y = R^N$, $C_n = I$, $C_n^P = L^{-1}$, and

$$F_A(u) = f(x, u, D^{\alpha_1}u, D^{\alpha_2}u, \ldots, D^{\alpha_M}u),$$

then the hypothesis of Theorem 3.4 *holds if u_0 is sufficiently near to u^* in the norm of X and the coefficients of A_0 are sufficiently near to those of $F'_A(u^*)$ in $L^\infty(\Omega; R^N)$. Therefore, the iterates given by the update* (3.1) *converge q-linearly in the norm of X.*

If in addition L^{-1} is a compact operator from $L^\infty(\Omega; R^N)$ to $C^{k-1}(\Omega; R^N)$ then the hypothesis of Theorem 4.2 *holds and the iterates converge q-superlinearly in the norm of X.*

Proof. The proof consists of verification of the assumption that $F'(u^*)^{-1}C^P \in \mathcal{L}(Y^p, X^{p_2})$ (Assumptions 2.1, 2.2, 2.3, 4.1, and 3.2). Of these assumptions, 2.1, 3.2, and 4.1 follow directly the assumptions on f and F and definitions of $\mathcal{E}$ and μ_X. The assumption that $F'(u^*)^{-1}C^P \in \mathcal{L}(Y^p, X^{p_2})$ is trivial taking $p_1 = p_2 = 2$ since then $Y^p = L^2$, $X^p = H^{k-1}$, and $F'(u^*)^{-1}C^P$ is a bounded map from L^2 to H^k by our assumption on L. We complete the proof by verifying Assumptions 2.2 and 2.3.

Let $u \in X$. Note that for all $x \in \Omega$, $u \in X$, and $E \in \mathcal{E}$ given by (4.15), we have by the Cauchy–Schwarz inequality that

$$\begin{aligned}\|Eu(x)\|_{R^N} &\le \textstyle\sum_{|\alpha|\le k-1} \|a_\alpha(x)(D^\alpha u)(x)\|_{R^N} \\ &\le \textstyle\sum_{|\alpha|\le k-1} \|a_\alpha(x)\|_{R^N} \|(D^\alpha u)(x)\|_{R^N} \\ &\le \left(\textstyle\sum_{|\alpha|\le k-1} \|a_\alpha(x)\|^2_{R^N}\right)^{1/2} \left(\textstyle\sum_{|\alpha|\le k-1} \|(D^\alpha u)(x)\|^2_{R^N}\right)^{1/2} \\ &\le \left(\textstyle\sum_{|\alpha|\le k-1} \|a_\alpha(x)\|^2_{F}\right)^{1/2} \left(\textstyle\sum_{|\alpha|\le k-1} \|(D^\alpha u)(x)\|^2_{R^N}\right)^{1/2} \\ &= \|E\|_{\mathcal{E}}(x)\|u\|_{\mu_X}(x),\end{aligned}$$

which is Assumption 2.2.

Now let $u, w \in X$ and $y \in Y$. By definition,

$$i(y,u)w(x) = \sum_{|\alpha|\le k-1} y(x)(D^\alpha u(x), D^\alpha w(x))_{R^N},$$

and so, by the Cauchy–Schwarz inequality,

$$\|i(y,u)w(x)\|_F \le \|y(x)\|_{R^N}\|u(x)\|_{\mu_X}(x)\|w(x)\|_{\mu_X}(x).$$

This is exactly (2.8).

To complete the verification of Assumption 2.3 it remains only to show that $\|I - P_s\|_{\mathcal{L}(\mathcal{E})} \le 1$. For $s \in X$ and $E \in \mathcal{E}$ given by (4.15) we have

$$\begin{aligned}i(Es,s)u &= (Es)(x)\mu_X(s,u) \\ &= \textstyle\sum_{|\tau|\le k-1}(a_\tau(x),(D^\tau s)(x))_{R^N}\sum_{|\alpha|\le k-1}(D^\alpha u(x))(D^\alpha s(x)).\end{aligned}$$

If we write

$$(I - P_s)E = \sum_{|\alpha|\le k-1} b_\alpha(x)D^\alpha,$$

we have that $b_\alpha = a_\alpha$ if $\mu_X(s,s)(x) = 0$. If $\mu_X(s,s)(x) \ne 0$ then P_s is an orthogonal projection on the vector space $R^{m\times N\times N}$ onto the one-dimensional subspace spanned by $\{D^\alpha u(x)\}_{\alpha\le k-1}$ and hence

$$\sum_{\alpha\le k-1} \|b_\alpha(x)\|^2_F \le \sum_{\alpha\le k-1} \|a_\alpha(x)\|^2_F.$$

This completes the verification of Assumption 2.3 and hence the proof. □

5. Nonstandard superlinear convergence. Problems and methods to which Theorem 4.2 can be applied are characterized by knowledge of the Fréchet derivative up to a compact error. This is the meaning of the requirement that the family

$$\{F'(u^*)^{-1}C^PE \,|\, E \in \mathcal{E},\ \|E\|_{\mathcal{E}} \le M\} \subset \mathcal{L}(X)$$

be collectively compact. When this compactness condition is violated a different form of superlinear convergence has been shown to hold in many situations [4], [8], [9], [11],

[13]. In the notation of §4 the local convergence is q-linear in X and superlinear in the sense that

$$\lim_{n\to\infty} \frac{\|e_{n+1}\|_{X^r}}{\|e_n\|_{X^s}} = 0 \tag{5.1}$$

for certain $1 \le r < s \le \infty$. The difference from the traditional notion of superlinear convergence is that the norms in the numerator and denominator are not the same.

We state and prove the general theorem on this type of convergence and then consider applications of that theory to the problems given in [4], [9], and [13]. This result gives (5.1) with $s = \infty$. A corollary of the proof will give the general form of (5.1) under an additional assumption.

THEOREM 5.1. *Assume that H_Y is finite-dimensional, the hypotheses for Theorem* 3.4 *hold, and that $\delta > 0$ corresponds to some $\sigma \in (0,1)$ as in Corollary* 3.2. *Assume that the computed parts of F' satisfy* (4.4) *and* (4.5). *Finally, assume that there are $r_0, r \in [1,\infty)$ such that $F'(u^*)^{-1}C^P$ can be extended to be a bounded operator from Y^{r_0} to X^r. Then*

$$\lim_{n\to\infty} \frac{\|e_{n+1}\|_{X^r}}{\|e_n\|_X} = 0. \tag{5.2}$$

Proof. The proof follows the lines of that of Theorem 4.2. The finite-dimensionality of H_Y implies that if α_n is given by (4.9), then $\alpha_n \to 0$ in $L^p(\Omega)$ for all $1 \le p < \infty$. Therefore,

$$\begin{aligned} \|E_n^A s_n\|_{Y^{r_0}}^{r_0} &= \textstyle\int_\Omega \|E_n^A s_n\|_{H_Y}^{r_0}(x)\,dx \\ &\le \textstyle\int_\Omega \alpha_n^{r_0}(x)\|s_n\|_{\mu X}^{r_0}(x)\,dx \\ &\le \|\alpha_n\|_{L^{r_0}}^{r_0}\|s_n\|_X^{r_0}. \end{aligned} \tag{5.3}$$

Equation (4.8) implies that

$$e_{n+1} = -F'(u^*)^{-1}C^P E_n^A s_n + \psi_n, \tag{5.4}$$

where

$$\psi_n = F'(u^*)^{-1}\left(\int_0^1 (F'(u^*) - F'(u^* + te_n))e_n\,dt - E_n^C s_n - E_n^P A_n s_n\right). \tag{5.5}$$

Our assumptions imply that

$$\frac{\|\psi_n\|_X}{\|e_n\|_X} \to 0.$$

Therefore,

$$\begin{aligned} \|e_{n+1}\|_{X^r} &\le \|F'(u^*)^{-1}C^P\|_{\mathcal{L}(Y^{r_0},X^r)}\|\alpha_n\|_{L^{r_0}}\|s_n\|_X + \|\psi_n\|_X \\ &\le \|F'(u^*)^{-1}C^P\|_{\mathcal{L}(Y^{r_0},X^r)}\|\alpha_n\|_{L^{r_0}}(1+\sigma)\|e_n\|_X + \|\psi_n\|_X, \end{aligned}$$

which completes the proof. □

In order to obtain the superlinear convergence rate given by (5.1) with values of s other than ∞, additional assumptions must be made on the convergence of the iterates.

Such assumptions are often trivial to verify [9], [11], [13]. We will state these assumptions in terms of the sequence ψ_n defined in (5.5).

THEOREM 5.2. *Let the assumptions for Theorem* 5.1 *hold. In addition, assume that* $r_0 \le r$ *and that for some* $s \in (r, \infty]$

$$\frac{\|\psi_n\|_{X^r}}{\|e_n\|_{X^s}} \to 0, \tag{5.6}$$

and that the convergence is q-linear in X^p *for all* $p \geq r$. *Then* (5.1) *holds.*

Proof. We reconsider (5.3). We note that for any $p > 1$ and $1/p + 1/q = 1$

$$\begin{aligned} \|E_n^A s_n\|_{Y^{r_0}}^{r_0} &= \textstyle\int_\Omega \|E_n^A s_n\|_{H_Y}^{r_0}(x)\,dx \\ &\le \textstyle\int_\Omega \alpha_n^{r_0}(x)\|s_n\|_{\mu_X}^{r_0}(x)\,dx \\ &\le \|\alpha_n^{r_0}\|_{L^q}\|\mu_X(s_n,s_n)^{r_0/2}\|_{L^p}. \end{aligned} \tag{5.7}$$

Since

$$\|\mu_X(s_n,s_n)^{r_0/2}\|_{L^p}^p = \int_\Omega \mu_X(s_n,s_n)^{pr_0/2}(x)\,dx$$

we may choose $p > 1$ such that $pr_0 = s$ and conclude that

$$\|E_n^A s_n\|_{Y^{r_0}} \le \|\alpha_n^{r_0}\|_{L^q}^{1/r_0}\|s\|_{X^s}. \tag{5.8}$$

The remainder of the proof is exactly the same as that of Theorem 5.1 up to the final estimate. We set

$$\chi_n = \frac{\|\psi_n\|_{X^r}}{\|e_n\|_{X^s}}$$

and obtain

$$\|e_{n+1}\|_{X^r} \le \|F'(u^*)^{-1}C^P\|_{\mathcal{L}(Y^{r_0},X^r)}\|\alpha_n^{r_0}\|_{L^q}^{1/r_0}\|s\|_{X^s} + \|\psi_n\|_{X^r}.$$

This is equivalent to (5.1) and completes the proof. □

Often, as in [9], [11], and [13], the assumptions for Theorem 5.2 are natural consequences of the structure. In the remainder of this section we consider several examples.

The simplest example is that of substitution operators (see [4]). Here

$$F(u)(x) = f(u(x)),$$

where f is a Lipschitz continuously differentiable map on R^N. The standard assumptions in this case are that $F'(u^*(x))$ is a nonsingular matrix-valued function with uniformly bounded inverse. We have $H_X = H_Y = R^N$, $\mu_X = (\cdot,\cdot)_{R^N}$, $X = L^\infty(\Omega : R^N)$, so $X^p = L^p(\Omega : R^N)$. We let $\mathcal{E}$ be the class of operators of multiplication by bounded $M \times M$ matrix-valued functions. If $E \in \mathcal{E}$ is the operator of multiplication by m_E we define

$$\|E\|_{\mathcal{E}} = \sup_{x\in\Omega}\|m_E(x)\|_{R^N}.$$

We take $F = F_A$. The assumptions required for Theorem 3.4 hold trivially. The pointwise update is Broyden's method itself applied at each $x \in \Omega$. Letting A_n be the operator of multiplication by m_n we have

$$m_+(x) = m_c(x) + \left(\|s(x)\|_{R^N}^2\right)^+ (y - A_c s)(x)s(x)^T.$$

The compactness conditions required in Theorem 4.2 do not hold because multiplication operators are not compact. However, the assumptions of Theorem 5.1 clearly do hold. In fact, since

$$\|E\|_{X^p} \le \|E\|_{\mathcal{E}}$$

for all $1 \le p \le \infty$ the iterates converge q-linearly in all the spaces X^p for $1 \le p \le \infty$. Since

$$\|\psi_n(x)\|_{R^N} \le \frac{\gamma}{2}\|e_n(x)\|_{R^N}^2$$

for all $x \in \Omega$, where γ is the Lipschitz constant of f', the assumptions of Theorem 5.2 hold as well.

In [9] fully nonlinear integral equations of the form

$$F(u)(x) = f(u(x)) + \int_\Omega k(x, y, u(y), u(x))\, dy = 0 \tag{5.9}$$

were considered. Here the unknown function $u \in L^\infty(\Omega : R^N)$. The Fréchet derivative is a sum of a multiplication operator and a compact integral operator. The requirement in [9] that all functions be continuous can be relaxed if we put the problem in the setting of this paper. We let

$$\mu_X(u, v) = u(x)^T v(x) + \int_\Omega u(y)^T v(y)\, dy.$$

$\mathcal{E}$ can be taken to be the space of operators of the form

$$Eu(x) = m_E(x)u(x) + \int_\Omega k_E(x, y)u(y)\, dy.$$

In [9] $\mathcal{E}$ was made into a pointwise inner product space, as was mentioned briefly in §2,

$$\mu_{\mathcal{E}}(A, B) = (m_A(x), m_B(x))_{R^F} + \int_\Omega (k_A(x, y), k_B(x, y))_{R^F}\, dy.$$

The norm on $\mathcal{E}$ is the pointwise inner product space norm

$$\|E\|_{\mathcal{E}} = \sup_{x\in\Omega} \mu_{\mathcal{E}}(E, E)(x)^{\frac{1}{2}}.$$

In this application $X = Y = L^\infty(\Omega : R^N)$, $F_A = F$. Verification of the hypotheses for Theorem 3.4 is direct and was explicitly carried out in [9]. The assumptions for Theorem 5.2 hold only for $r \ge 2$ and this is also described in [9].

The same update can be applied to the more general problem

$$F(u)(x) = f(u(x), \mathcal{K}(u)(x)) = 0,$$

where

$$\mathcal{K}(u)(x) = \int_\Omega k(x, y, u(y), u(x))\, dy,$$

$k \in L^\infty(\Omega \times \Omega \times R^N \times R^N : R^P)$, and $f \in L^\infty(R^N \times R^P : R^N)$ for some P. The pointwise inner product, the error class, and the convergence results are the same in this case.

REFERENCES

[1] C. G. BROYDEN, *A class of methods for solving simultaneous equations*, Math. Comp., 19 (1965), pp. 577–593.

[2] J. E. DENNIS AND H. F. WALKER, *Convergence theorems for least change secant update methods*, SIAM J. Numer. Anal., 18 (1981), pp. 949–987.

[3] A. GRIEWANK, *The solution of boundary value problems by Broyden based secant methods*, in Computational Techniques and Applications: CTAC 85, Proceedings of CTAC, Melbourne, August 1985, J. Noye and R. May, eds., North Holland, Amsterdam, 1986, pp. 309–321.

[4] ———, *On the iterative solution of differential and integral equations using secant updating techniques*, in The State of the Art in Numerical Analysis, A. Iserles and M. Powell, eds., 1987, Clarendon Press, Oxford, U. K., pp. 299–324.

[5] W. A. GRUVER AND E. SACHS, *Algorithmic Methods In Optimal Control*, Pitman, London, 1980.

[6] W. E. HART AND S. O. W. SOUL, *Quasi-Newton methods for discretized nonlinear boundary problems*, J. Inst. Appl. Math., 11 (1973), pp. 351–359.

[7] D. M. HWANG AND C. T. KELLEY, *Convergence of Broyden's method in Banach spaces*, SIAM J. Optimization, 2 (1992), pp. 505–532.

[8] C. T. KELLEY AND J. I. NORTHRUP, *Pointwise quasi-Newton methods and some applications*, in Distributed Parameter Systems, F. Kappel, K. Kunisch, and W. Schappacher, eds., Springer-Verlag, New York, 1987, pp. 167–180.

[9] ———, *A pointwise quasi-Newton method for integral equations*, SIAM J. Numer. Anal., 25 (1988), pp. 1138–1155.

[10] C. T. KELLEY AND E. W. SACHS, *A quasi-Newton method for elliptic boundary value problems*, SIAM J. Numer. Anal., 24 (1987), pp. 516–531.

[11] ———, *A pointwise quasi-Newton method for unconstrained optimal control problems*, Numer. Math., 55 (1989), pp. 159–176.

[12] ———, *A new proof of superlinear convergence for Broyden's method in Hilbert space*, SIAM J. Optimization, 1 (1991), pp. 146–150.

[13] C. T. KELLEY, E. W. SACHS, AND B. WATSON, *A pointwise quasi-Newton method for unconstrained optimal control problems*, II, J. Optim. Theory Appl., 71 (1991), pp. 535–547.

[14] E. SACHS, *Broyden's method in Hilbert space*, Math. Programming, 35 (1986), pp. 71–82.

[15] L. K. SCHUBERT, *Modification of a quasi-Newton method for nonlinear equations with sparse Jacobian*, Math. Comp., 24 (1970), pp. 27–30.

SIAM J. OPTIMIZATION
Vol. 3, No. 3, pp. 443–465, August 1993

NONSMOOTH EQUATIONS: MOTIVATION AND ALGORITHMS*

JONG-SHI PANG† AND LIQUN QI‡

Abstract. This paper reports on some recent developments in the area of solving of nonsmooth equations by generalized Newton methods. The emphasis is on three topics: motivation, characterization of superlinear convergence, and a new Gauss–Newton method for solving a certain class of nonsmooth equations. The characterization of superlinear convergence extends the classical result of Dennis and Moré for smooth equations and that of Ip and Kyparisis for B-differentiable equations. The Gauss–Newton method is different from that proposed recently by Han, Pang, and Rangaraj; it uses convex quadratic programs to generate descent directions for the least-squares merit function.

Key words. nonsmooth analysis, Newton methods, convergence theory, variational inequality, nonlinear programming, complementarity problems

AMS subject classifications. 90C30, 90C33

1. Introduction. In the past few years there has been a growing interest in the study of systems of nonsmooth equations; these are nonlinear equations that are defined by functions that are not differentiable in the traditional sense of Fréchet or Gâteaux. In particular, the numerical solution of these nonsmooth equations by some generalizations of the classical Newton methods for their smooth counterparts has received considerable attention. Two major factors have stimulated this growth of interest. The first factor is that nonsmooth equations provide a unified framework for the study of a number of important problems in mathematical and equilibrium programming. Within this framework these problems are brought one step closer to the classical problem of solving smooth equations for which there are rich theory and abundant solution methods that are very powerful [6], [21]. The second factor, which is a consequence of the first, is that on the basis of their nonsmooth equation formulation, some new solution methods can be developed for solving optimization and equilibrium problems; these methods are not only highly efficient but they also actually resolve the lack of robustness in many previous solution approaches (see [24]).

The present paper is intended to provide a unified treatment of the theory of solving nonsmooth equations by generalized Newton methods. This research emphasizes three major topics: motivation, characterization of superlinear convergence, and the design of a new Gauss–Newton method. To motivate the discussion we begin with a description of various sources of nonsmooth equations; these are drawn from complementarity, optimization, and several related problems. As evidenced from previous works on the subject [17], [22], [28], [30], [31], [34], results from nonsmooth analysis are important tools for the development of the Newton methods. For this reason we shall summarize the necessary background of nonsmooth analysis and shall define some new concepts that are useful for the Gauss–Newton method. The remaining part of the paper focuses on the study of iterative algorithms for solving the nonsmooth equations. Two general convergence results are derived; these extend some well-known characterizations of Q-superlinear convergence for smooth equations (due to Dennis and Moré [5])

* Received by the editors August 7, 1991; accepted for publication (in revised form) March 24, 1992.

† Department of Mathematical Sciences, Johns Hopkins University, Baltimore, Maryland 21218. The work of this author was supported by National Science Foundation grants ECS-8717968 and DDM-9104078; it was carried out while he was visiting the Department of Applied Mathematics at the University of New South Wales.

‡ Department of Applied Mathematics, University of New South Wales, Kensington, New South Wales 2033, Australia. The work of this author was supported by the Australian Research Council.

to the nonsmooth context. Applications of these results to some specific Newton methods are discussed. Section 5 describes a Gauss–Newton algorithm for a certain class of nonsmooth equations; this algorithm generalizes the NE/SQP method for the nonlinear complementary problem proposed in [24]. The global and Q-superlinear convergence of the Gauss–Newton method will be established.

2. Source of nonsmooth equations.

The focus of this paper is the numerical solution on nonsmooth equations

$$H(x) = 0, \tag{1}$$

where the mapping $H : R^n \to R^n$ is assumed to be locally Lipschitzian. Shapiro [41] has shown that when H is also directionally differentiable, then H must be B(ouligand)-differentiable in the sense of Robinson [33]. In this paper we take this to be the definition of a B-differentiable function; i.e., a function that is both locally Lipschitzian and directionally differentiable on an open set is said to be B-differentiable there.

The study of solving a system of B-differentiable equations was initiated in [22], in which a generalization of the classical Newton method for smooth equations was suggested as a solution method. That paper also contains a discussion of several mathematical/equilibrium programming problems to which the proposed methodology can be applied. In what follows we shall review these and several related problems and shall use them as the motivation for the study of the nonsmooth equation (1).

2.1. Nonlinear complementarity problem.

The nonlinear complementarity problem (NCP) provides the prime candidate for illustrating the methodology of nonsmooth equations. For this reason we start with it. Let $f : D \to R^n$ be a given function assumed to be continuously differentiable on the open set $D \subseteq R^n$ containing the nonnegative orthant R^n_+. This problem, denoted NCP (f), is to find a vector x such that

$$x \geq 0, \quad f(x) \geq 0, \quad x^T f(x) = 0.$$

There are two ways to formulate this problem as a system of nonsmooth equations; these are obtained through two functions $H : D \to R^n$ and $\tilde{H} : R^n \to R^n$ defined by

$$H(x) = \min(x, f(x)), \qquad \tilde{H}(z) = f(z^+) - z^-.$$

Here "min" denotes the componentwise minimum operator and z^+ and z^- are, respectively, the nonnegative part and the nonpositive part of the vector z. It is not difficult to verify that x is a zero of H if and only if x solves NCP (f) and that if z is a zero of $\tilde{H}$, then z^+ solves NCP(f) and, conversely, if x solves NCP(f), then $z = x - f(x)$ is a zero of $\tilde{H}$.

Both of these functions, H and $\tilde{H}$, are not F-differentiable, but they are B-differentiable. In general, each of these two functions has intrinsic properties, such that neither formulation dominates the other as far as the numerical methods are concerned.

2.2. NCP with upper bounds.

In mathematical/equilibrium programming it is common for the variables of a problem to have upper and lower bounds. When these are present, it would be desirable to deal with them simultaneously and not to treat one set of bounds implicitly and the other as explicit constraints. In the case of the NCP this is indeed possible with the nonsmooth equation approach.

Formally, the NCP with upper bounds is defined as follows. Let $a \in R^n$ be a positive vector and let $f : D \to R^n$ be a once continuously differentiable function defined on the same open set D. This problem is to find a vector pair $(x, y) \in R^n \times R^n$ such that

$$\begin{array}{llll} (2) & u = f(x) + y \geq 0, & x \geq 0, & u^T x = 0, \\ & u = a - x \geq 0, & y \geq 0, & u^T y = 0. \end{array}$$

Notice that in terms of the vectors x and y this is a standard NCP of order $2n$. Nevertheless, it is possible to turn this problem into a system of nonsmooth equations of order n. Again, there are two such formulations. One is defined by the function $\tilde{H} : R^n \to R^n$, with

$$\tilde{H}(z) = f(\Pi_{[0,a]}(z)) + (z - \Pi_{[0,a]}(z)),$$

where $\Pi_{[0,a]}(z)$ denotes the projection of the vector z onto the n-dimensional rectangle $[0, a]$, i.e.,

$$\Pi_{[0,a]}(z) = \min(a, \max(0, z)).$$

See §3 for a more general discussion of how a zero of this function $\tilde{H}$ corresponds to a solution of the given NCP with upper bounds.

The other formulation of the NCP with upper bounds as a system of nonsmooth equations is defined by the function $H : D \to R^n$, where

$$H(x) = \min(f(x)^+, x) + \min(f(x)^-, a - x).$$

With this function H it can be shown that a vector x solves the problem (2) if and only if x is a solution of the *constrained* equation.

$$(3) \qquad H(x) = 0, \qquad x \in [0, a].$$

The proof of this equivalence is fairly straightforward. As we shall see later, imposing the simple bound constraints on the variable x poses no difficulty for the global Newton method of solving this problem (3); in fact these constraints actually are beneficial for the numerical procedure.

2.3. Variational inequality problem over a convex set. The previous two examples are special cases of the variational inequality problem defined over a closed convex set. Let K be a closed convex subset of R^n, and let $f : D \to R^n$ be a once continuously differentiable function defined on the open set $D \subseteq R^n$ containing K. This problem, which we denote VI(K, f), is to find a vector $x^* \in K$ such that

$$(y - x^*)^T f(x^*) \geq 0 \quad \text{for all } y \in K.$$

When f is the gradient mapping of the real-valued function $\psi : R^n \to R$, the problem VI(K, f) becomes the stationary point problem of the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \psi(x) \\ \text{subject to} & x \in K. \end{array}$$

We refer the reader to [10] for a comprehensive review of the variational inequality problem.

Generalizing the functions H and $\tilde{H}$, we can derive two formulations of this problem as a system of nonsmooth equations. More specifically, define

$$H(x) = x - \Pi_K(x - f(x)) \quad \text{and} \quad \tilde{H}(z) = f(\Pi_K(z)) + (z - \Pi_K(z)).$$

The equivalence between the resulting systems of equations and the problem VI(K, f) is well known; see, e.g., [11, Chap. 4] for a proof. Robinson [36] calls the function $\tilde{H}$ a normal map. We point out that the convexity of the defining set K is important in these equivalent formulations.

The nonsmoothness of the functions H and $\tilde{H}$ is, of course, the consequence of the projection operator $\Pi_K(\cdot)$. When K is a polyhedral set, this operator possesses some differentiability properties that can be put to use algorithmically.

2.4. Karush–Kuhn–Tucker system. Consider the problem VI(K, f) in which the set K is represented by a system of differentiable inequalities and equalities:

$$K = \{x \in R^n : g(x) \leq 0, h(x) = 0\},$$

where $g : R^n \to R^p$ and $h : R^n \to R^q$ are twice continuously differentiable. In this case the aforementioned projection formulations fail to be well defined because of the possible nonconvexity of the set K. Nevertheless, under a standard constraint qualification, such as the polyhedrality of K or the well-known Mangasarian–Fromovitz condition, we may derive the Karush–Kuhn–Tucker system for the problem VI(K, f). The latter system is equivalent to a system of nonsmooth equations with the mapping $H : R^n \times R^p \times R^q \to R^n \times R^p \times R^q$ given by

$$(4) \qquad H(x, \lambda, \mu) = \begin{pmatrix} f(x) + \sum_{i=1}^{p} \lambda_i \nabla g_i(x) + \sum_{j=1}^{q} \mu_j \nabla h_j(x) \\ \min(\lambda, -g(x)) \\ h(x) \end{pmatrix},$$

where $\lambda \in R^p$ and $\mu \in R^q$ are the Lagrange multipliers associated with the inequality and equality constraints, respectively. Kojima [15] suggested an alternative formulation of the same Karush–Kuhn–Tucker system as a system of nonsmooth equations that involves the use of λ^+ and λ^-; cf. the function $\tilde{H}$ for the NCP.

2.5. Special system of piecewise-smooth equations. It is possible to generalize the NCP in a number of ways, resulting in various forms of the *generalized complementarity problem*. One such generalization leads to a system of nonsmooth equations defined by the function

$$H(x) = \min(f_1(x), \ldots, f_N(x)),$$

where each $f_j : R^n \to R^n$ is once continuously differentiable. A zero of this function H solves the following complementarity system:

$$f_j(x) \geq 0, \qquad j = 1, \ldots, N,$$

$$\prod_{j=1}^{N} f_{ij}(x) = 0, \qquad i = 1, \ldots, n,$$

where $f_{ij}(x)$ is the ith component of $f_j(x)$. A practical realization of this problem arises from a mechanical engineering application; see [20].

2.6. Inequality feasibility problem. This is the most fundamental problem in mathematical/equilibrium programming. Let $g : R^n \to R^n$ be a locally Lipschitzian function, and K be a polyhedral set in R^n. This problem is to find a vector $x \in R^n$ such that

$$g(x) \geq 0, \qquad x \in K.$$

Letting $H(x) = \min\,(0, g(x))$, we see that a constrained zero of H, i.e., a solution of the system

$$H(x) = 0, \qquad x \in K,$$

corresponds precisely to a solution of the feasibility problem.

2.7. Maximal monotone operator. Let $T : R^n \to R^n$ be a set-valued maximal monotone operator. An important problem is to find $x \in R^n$ such that

$$0 \in T(x). \tag{5}$$

The generalized equation [32] is a special case of this problem. According to the theory of the maximal monotone operator [3], [37], the resolvent of T, namely, $P_\lambda = (I+\lambda T)^{-1}$, where I is the identity operator and λ is a positive number, is always single valued and nonexpansive (hence globally Lipschitzian). Moreover, the solution of (5) is equivalent to that of the nonsmooth equation (1), where

$$H(x) = x - P_\lambda(x).$$

We should perhaps point out that, at present, properties for this function H that are useful for the development of a Newton method for solving the corresponding equation (1) are not well understood. Further research in exploring the nonsmooth nature of the resolvent is required for this purpose.

2.8. LC^1 optimization problem. For some optimization problem the objective function (which is real valued) is not a C^2 function but is an LC^1 function; i.e., it is once continuously differentiable and its derivative is locally Lipschitzian but not necessarily F-differentiable. For example, the extended linear-quadratic problem, which arises from stochastic programming and optimal control [39], [40], is such a problem in the fully quadratic case. The augmented Lagrangian of a C^2 nonlinear program is also an LC^1 function [30]. For more examples of LC^1 functions and the corresponding optimization problems see [29].

The problem of finding a stationary point of an unconstrained LC^1 optimization problem is equivalent to that of solving a system of locally Lipschitzian equations (1), where H is the gradient mapping of the objective function of the given optimization problem. For a constrained LC^1 optimization problem the Karush–Kuhn–Tucker system still leads to a system of nonsmooth equations.

3. Nonsmooth analysis. Nonsmooth analysis is an essential tool for the design of effective numerical methods for solving the nonsmooth equation (1) and for the development of the supporting convergence theory. Since Clarke introduced his generalized subdifferential theory [4] nonsmooth analysis has developed into a very fruitful discipline. However, a major portion of this analysis is associated with the optimization of a real-valued function; many concepts are thus defined only for functionals. The nonsmooth equation, on the other hand, involves a vector-valued function $H : R^n \to R^n$. Some existing concepts in nonsmooth analysis, therefore, become inadequate and need to be modified for adaptation to the vector-valued setting. An example of such a concept is that of semismoothness, originally introduced by Mifflin [19] for functionals. Qi and Sun [30] extended Mifflin's original definition to a vector-valued function and used the generalized notion to study a Newton method for solving (1).

In this section we review the notion of semismoothness for vector-valued functions and shall connect it with that of strong B-differentiability (introduced by Robinson [35]). We also define a new concept, called an upper subgradient, that we shall use in §5 for the development of the Gauss–Newton method.

3.1. Semismoothness. For all the nonsmooth equations presented in §2 the function $H : R^n \to R^n$ is locally Lipschitzian. For such a function Rademacher's theorem implies that H is almost everywhere F-differentiable. Let the set of points where H is F-differentiable be denoted D_H. Then for any $x \in R^n$ the generalized subdifferential of H at x in the sense of Clarke [4] is

$$\partial H(x) = \text{conv}\,\{\lim \nabla H(x^j) : x^j \to x, x^j \in D_H\},$$

which is a nonempty convex compact set. Considered as a set-valued mapping, ∂H is locally bounded and upper semicontinuous.

For $x, h \in R^n$ with $h \neq 0$ we say that *y tends to x in the direction h*, denoted by $y \to_h x$, if $y \to x$, $y \neq x$, and $(y-x)/\|y-x\| \to h/\|h\|$. We say that H is *semismooth* at x if H is locally Lipschitzian there and if for any $h \in R^n$ with $h \neq 0$

$$\lim_{y \to_h x} \{Vh : V \in \partial H(y)\}$$

exists. If H is semismooth at x, then H must be directionally differentiable (hence B-differentiable) at x and $H'(x;h)$ is equal to the above limit for any $h \neq 0$. If H is semismooth at all points in a given set, we say that H is semismooth in this set.

It was proved in [30] that H is semismooth at x if and only if all its component functions are the same. The class of semismooth functionals is very broad; indeed, according to [19], it includes the smooth functions, all convex functions, and the piecewise-smooth functions. Moreover, the sums, differences, products, and composites of semismooth functions are semismooth. In particular, if $f(x)$ is semismooth, then so is $\min(x, f(x))$. Furthermore, since the projection operator $\Pi_K(y)$ is a piecewise linear function of y when K is a polyhedron, it follows that Π_K is a semismooth operator. As a matter of fact, with the possible exceptions of the VI on a nonpolyhedral set and the example involving the resolvent of a maximal monotone operator, all equations encountered in §2 are of the piecewise-smooth, and hence semismooth, type.

In the following proposition we state a property of a semismooth function that will be used later.

PROPOSITION 1. *If $H : R^n \to R^n$ is semismooth at x, then*

$$\lim_{\substack{h \to 0 \\ V \in \partial H(x+h)}} \frac{\|H(x+y) - H(x) - Vh\|}{\|h\|} = 0.$$

Proof. Since semismooth implies B-differentiability, we have

$$\lim_{h \to 0} \frac{\|H(x+h) - H(x) - H'(x,h)\|}{\|h\|} = 0.$$

Moreover, by Theorem 2.3 of [30] we have

$$\lim_{\substack{h\to 9\\ V\in\partial H(x+h)}} \frac{\|H'(x,h)-Vh\|}{\|h\|}=0.$$

The desired conclusion follows easily from these two equalities. □

In [35] the concept of strong B-differentiability is defined. A function $H: R^n \to R^n$ is said to be *strongly B-differentiable* at $x \in R^n$ if H is B-differentiable at x and if

$$\lim_{h,h'\to 0} \frac{e_x(h')-e_x(h)}{\|h'-h\|}=0,$$

where $e_x(h) = H(x+h) - H(x) - H'(x,h)$ is the error of approximating $H(x+h)$ by the term $H(x) + H'(x,h)$. The following result shows that the strong B-differentiability property is stronger than that of semismoothness.

PROPOSITION 2. *If H is B-differentiable in a neighborhood of x and is strongly B-differentiable at x, then H is semismooth at x.*

Proof. Fix a scalar $t \in (0,1)$. In terms of the error function e_x we have

$$\begin{gathered} H(x+(1+t)h) = e_x((1+t)h) + H(x) + H'(x,(1+t)h),\\ H(x+h) = e_x(h) + H(x) + H'(x,h). \end{gathered}$$

Subtracting and rearranging terms, we obtain

$$H(x+(1+t)h) - H(x+h) - tH'(x,h) = e_x((1+t)h) - e_x(h).$$

The strong B-differentiability assumption implies

$$\lim_{\substack{h\to 0\\ t\downarrow 0}} \frac{e_x((1+t)h)-e_x(h)}{\|th\|}=0,$$

which yields

$$\lim_{h\to 0}\lim_{t\downarrow 0} \frac{H(x+(1+t)h)-H(x+h)-tH'(x,h)}{\|th\|}=0.$$

Hence we have

$$\lim_{h\to 0} \frac{H'(x+h,h)-H'(x,h)}{\|h\|}=0.$$

By Theorem 2.3 of [30] it follows that H is semismooth at x. □

3.2. BD-regularity. For a given $x \in R^n$ Clarke's generalized subdifferential $\partial H(x)$ is the convex hull of the following set:

$$\partial_B H(x) = \{\lim \nabla H(x^j) : x^j \to x, x^j \in D_H\}.$$

We call $\partial_B H(x)$ the B-*subdifferential* of H at x. This concept was introduced in [28], where an explanation was also given for its introduction. We say that H is BD-*regular* at x if all the elements in $\partial_B H(x)$, which themselves are $n \times n$ matrices, are nonsingular. This definition is slightly different from that in [28], where this condition was called strong BD-regularity and BD-regularity referred to a weaker condition. In proving the superlinear convergence results in §4 we need the following properties of BD-regularity.

PROPOSITION 3. *If H is BD-regular at x, then there is a neighborhood N of x and a constant c such that for any $y \in N$ and $V \in \partial_B H(y)$, V is nonsingular and $\|V^{-1}\| \geq c$. If, furthermore, $H(x) = 0$ and H is semismooth at x, then there is a neighborhood N' of x and a constant β such that for any $y \in N'$*

$$\|H(y)\| \geq \beta \|y - x\|.$$

Proof. The first part of this proposition is the first conclusion of Lemma 2.6 in [28]. If $H(x) = 0$ and H is semismooth at x, then

$$H(y) = H'(x, y - x) + o(\|y - x\|)$$

for each y there is a $V \in \partial_B H(x)$ such that $H'(x, y - x) = V(y - x)$. The second conclusion of the proposition now follows the first. □

3.3. Upper subdifferentiability. Motivated by the development in [24], we introduced the following concept. A real-valued function $\psi : R^n \to R$ is said to be *upper subdifferentiable* on a set $D \subseteq R^n$ if there exists a function $a : D \to R^n$ such that for all $x \in D$ and $h \in R^n$

$$\limsup_{\substack{y \to x, y \in D \\ t \downarrow 0}} \frac{\psi(y + th) - \psi(y) - ta(y)^T h}{t} \leq 0. \tag{6}$$

We call a an *upper subgradient function* of ψ on D, and we call $a(x)$ an *upper subgradient* of ψ at x. Notice that in this definition we have not imposed any property on the function a except for the above limit requirement.

As we shall see, the upper subgradient function plays an important part in the derivation of the Gauss–Newton method for solving (1). When D is an open set and ψ is continuously differentiable on D, then clearly ψ is upper subdifferentiable there. Moreover, if D is an open convex set and ψ is concave on D (not necessarily F-differentiable), then any subgradient of ψ is an upper subgradient. Hence a concave function is upper subdifferentiable. It turns out that by composing an F-differentiable function with a concave function in a proper order the resulting function is upper subdifferentiable (see Proposition 5 below). Moreover, the set of upper subdifferentiable functions forms a convex cone in the space of real-valued functions; that is, this set is closed under addition and positive scalar multiplication.

The upper subgradient is related to several directional derivatives. Indeed, if ψ is locally Lipschitzian at x, then putting $y = x$ in (6) immediately yields

$$a(x)^T h \geq \psi^D(x, h) \quad \text{for all } h \in R^n,$$

where ψ^D denotes the upper Dini directional derivative

$$\psi^D(x, h) = \limsup_{t \downarrow 0} \frac{\psi(x + th) - \psi(x)}{t}.$$

More generally, suppose that $\psi : R^n \to R$ is locally Lipschitzian on the set $D \subseteq R^n$. For any $x \in D$ the Clarke subdifferential [4] and the Michel–Penot subdifferential [1], [18] of ψ at x are defined, respectively, by

$$\partial\psi(x) = \{u \in R^n : \langle u, h \rangle \leq \psi^\circ(x, h) \quad \forall h \in R^n\}$$

and

$$\partial^\diamond\psi(x) = \{u \in R^n : \langle u, h\rangle \leq \psi^\diamond(x, h) \quad \forall h \in R^n\}$$

where $\psi^\circ(x, h)$ and $\psi^\diamond(x, h)$ are the Clarke and Michel–Penot directional derivatives of ψ at x in the direction h, respectively, i.e.,

$$\psi^\circ(x, h) = \limsup_{\substack{y \to x \\ t\downarrow 0}} \frac{\psi(y + th) - \psi(y)}{t}$$

and

$$\psi^\diamond(x, h) = \sup_{k \in R^n} \limsup_{t\downarrow 0} \frac{\psi(x + th + th) - \psi(x + tk)}{t}.$$

It is known that $\partial^\diamond\psi(x)$ and $\partial\psi(x)$ are nonempty compact convex sets and that

$$\partial^\diamond\psi(x) \subseteq \partial\psi(x), \qquad \psi^\diamond(x, h) \leq \psi^\circ(x, h).$$

Moreover, the above definition of $\partial\psi(x)$ coincides with the one given in §3.1 when H is a real-valued function. The following result summarizes the relationship between the upper subgradient and these various known concepts.

PROPOSITION 4. *Suppose that $\psi : R^n \to R$ is locally Lipschitzian on the set $D \subseteq R^n$. If ψ is upper subdifferentiable on D with an upper subgradient function $a(\cdot)$, then for each $x \in D$*

$$a(x) \in \partial^\diamond\psi(x) \subseteq \partial\psi(x). \tag{7}$$

Hence for any $h \in R^n$

$$\psi^D(x, h) \leq a(x)^T h \leq \psi^\diamond(x, h) \leq \psi^\circ(x, h).$$

Proof. Let $y = x$ in (6). Then we have for any $h \in R^n$

$$\begin{aligned} a(x)^T(-h) &\leq \liminf_{t\downarrow 0} \frac{\psi(x) - \psi(x + th)}{t} \\ &\leq \limsup_{t\downarrow 0} \frac{\psi(x + th - th) - \psi(x + th)}{t} \\ &\leq \psi^\diamond(x, -h). \end{aligned}$$

Since h is arbitrary, (7) follows. The last string of inequalities follows from this inclusion. □

An immediate consequence of Proposition 4 is the following.

COROLLARY 1. *Let ψ be as given in Proposition 4. Then for each $x \in D$ there exist a neighborhood N of x and a constant $c > 0$ such that for all $y \in D \cap N$, $\|a(y)\| \leq c$. Moreover, if $\{x^k\} \subseteq D$ converges to x and if $\{a(x^k)\}$ also converges, then the limit of $\{a(x^k)\}$ is an element of $\partial\psi(x)$.*

Proof. The first conclusion is a consequence of the local boundedness of the Clark generalized subdifferential $\partial\psi$. The second conclusion follows from the upper semicontinuity of the Clarke subdifferential. □

Using the above corollary, we may establish the following composite property of an upper subdifferential function.

PROPOSITION 5. *Let D be an open convex set in R^n, and let $\psi = \phi^\circ g$, where $g : D \to R^m$ is a continuously differentiable function, and $\phi : R^m \to R$ is locally Lipschitzian and upper subdifferentiable on $g(D)$. Then ψ is upper subdifferentiable on D.*

Proof. Let b be an upper subgradient function of ϕ on $g(D)$. Define for all $x \in D$, $a(x) = \nabla g(x)^T b(g(x))$. We claim that this a is a desired upper subgradient function of ψ on D. Let $h \in R^n$ be arbitrary, let $y \in D$ be sufficiently close to x, and let $t > 0$ be sufficiently small. By the continuous differentiability of g we may write

$$g(y + th) = g(y) + t\left(\nabla g(y)h + \frac{o(t)}{t}\right).$$

Let $r_h(t,h) = (g(y + th) - g(y))/t$. Then $r_h(t,y) \to \nabla g(x)h$ as $t \downarrow 0$ and $y \to x$. We have

$$\psi(y + th) - \psi(y) = \phi(g(y) + tr_h(t,y)) - \phi(g(y)),$$

which implies

$$\begin{aligned}\psi(y + th) - \psi(y) - ta(y)^T h = {}& \phi(g(y) + tr_h(t,y)) - \phi(g(y) + t\nabla g(x)h)\\ &+\phi(g(y) + t\nabla g(x) - \phi(g(y))\\ &-tb(g(y))^T \nabla g(x)h\\ &+tb(g(y))^T(\nabla g(x) - \nabla g(y))h.\end{aligned}$$

As $y \to x$, $g(y) \to g(x)$; hence Corollary 1 implies that $\|b(g(y))\|$ is bounded. Consequently, dividing by t and taking limit $t \downarrow 0$, $y \to x$, we deduce that the first right-hand difference in the above expression tends to zero (by the assumed local Lipschitzian property of ϕ and the fact that $r_h(t,y) \to \nabla g(x)h$); from this observation and the upper subdifferentiability of ϕ at $g(x)$, we easily establish the desired upper subdifferentiability of ψ at x. □

The upper subgradient is also related to another generalized gradient notion in the literature. Suppose that $\psi : R^n \to R \cup \{\infty, -\infty\}$ is an extended real-valued function. In [25], [27] a vector $u \in R^n$ is called a *lower semigradient* of ψ at x if

$$\liminf_{h\to 0} \frac{\psi(x + h) - \psi(x) - u^T h}{\|h\|} \geq 0. \tag{8}$$

Lower semigradients are referred to as *Dini subdifferentials* in [2], [13]. Comparing this definition with Proposition 2.5 of [38], we see that lower subgradients and epi-gradients are equivalent if ψ is epi-differentiable at the point in question. See [38] for the concepts of epi-gradients and epi-differentiability. By these definitions it is not difficult to prove the following proposition.

PROPOSITION 6. *Suppose that $\psi : R^n \to R$ is locally Lipschitzian on an open set $D \subseteq R^n$. If ψ is upper subdifferentiable on D with an upper subgradient function $a(\cdot)$, then for each $x \in D$, $-a(x)$ is a lower semigradient of $-\psi$ at x in the sense defined above.*

Proof. This is straightforward. □

Clearly, the definition of an upper subgradient function requires more than that of a lower semigradient in the case of locally Lipschitzian functions since in (6) y is introduced to approximate x, whereas in (8) only x is used.

Finally, we mention that Plastria [26] has used the term *lower subgradient* as a generalized concept of a subgradient in convex analysis. His definition is along the line of (8) but is different from the lower semigradient concept as well as from ours.

4. Characterization of superlinear convergence. In this section we consider the generalization of some well-known results due to Dennis and Moré [5] that characterize the Q-superlinear convergence of the family of quasi-Newton methods for solving a system of smooth equations. For ease of reference we quote their result as stated in [6, Thm. 8.2.4].

THEOREM 1 (Dennis–Moré). *Let $H : R^n \rightarrow R^n$ be F-differentiable in the open convex set in D in R^n. Assume that ∇H is continuous at some $x^* \in D$ and that $\nabla H(x^*)$ is nonsingular. Let $\{B_k\}$ be a sequence of nonsingular $n \times n$ matrices such that for some x^0 in D the sequence $\{x^k\}$ where*

$$x^{k+1} = x^k - B_k^{-1} H(x^k) \tag{9}$$

remains in D and converges to x^* and where $x^k \neq x^*$ for all k. Then $\{x^k\}$ converges Q-superlinearly to x^* and $H(x^*) = 0$ if and only if

$$\lim_{k\rightarrow\infty} \frac{\|B_k - \nabla H(x^*))d^k\|}{\|d^k\|} = 0, \tag{10}$$

where $d^k = x^{k+1} - x^k$.

There are two noteworthy points about this theorem. First, the function H is assumed to be continuously differentiable at the point x^*, and second, the result concerns a sequence of iterates of the form defined by the quasi-Newton formula (9). Recently, Ip and Kyparisis [14] extended the above theorem to the case of a B-differentiable function; they still require a strong F-differentiability condition of H at x^*, and they confine the discussion to iteration (9). Such a strong differentiability assumption hinders the application of the result to a more general nonsmooth setting; also, the confinement to the quasi-Newton iterates seems a bit too restrictive.

In what follows, we establish a generalized version of Theorem 1 that significantly relaxes the two confinements mentioned above.

THEOREM 2. *Let $H : R^n \rightarrow R^n$ be locally Lipschitzian in the open convex set $D \subseteq R^n$. Assume that H is semismooth and BD-regular at some $x^* \in D$. Let $\{x^k\} \subseteq D$ be any sequence that converges to x^* with $x^k \neq x^*$ for all k. Then $\{x^k\}$ converges Q-superlinearly to x^* and $H(x^*) = 0$ if and only if*

$$\lim_{k\rightarrow\infty} \frac{\|H(x^k) + V^k d^k\|}{\|d^k\|} = 0, \tag{11}$$

where $V^k \in \partial_B H(x^k)$ and $d^k = x^{k+1} - x^k$.

Proof. Write $e^k = x^k - x^*$. Then $d^k = e^{k+1} - e^k$, and both sequences $\{e^k\}$ and $\{d^k\}$ convergence to zero. We have

$$H(x^*) = [H(x^k) + V^k d^k] - [H(x^k) - H(x^*) - V^k e^k] - V^k e^{k+1}. \tag{12}$$

The semismoothness of H at x^* implies that the term in the second set of square brackets approaches zero as $k \rightarrow \infty$; moreover, since $\{V^k\}$ is bounded and $\{e^{k+1}\} \rightarrow 0$, the last term in (12) also approaches zero as $k \rightarrow \infty$. Hence if (11) holds, then $H(x^*) = 0$, furthermore, since each $(V^k)^{-1}$ exists and the sequence $\{\|(V^k)^{-1}\|\}$ is bounded (by the BD-regularity assumption and Proposition 3), if follows from (11), (12), and Proposition 1 that

$$\lim_{k\rightarrow\infty} \frac{\|e^{k+1}\|}{\|e^k\|} = 0,$$

which establishes the Q-superlinear convergence of the sequence $\{x^k\}$ to x^*.

Conversely, suppose $H(x^*) = 0$ and $\{x^k\}$ converges to x^* Q-superlinearly. Then reversing the above argument easily establishes condition (11). □

Notice that a presumption of the above theorem is that the sequence $\{x^k\}$ converges to x^*. Hence this result cannot be used to demonstrate the convergence of a sequence produced by a given method. Instead, the usefulness of the theorem is to provide a way to establish the rate of convergence. We now give several applications of Theorem 2 to some specific methods.

Example 1. Let d^k be a solution of the linear equation

$$H(x^k) + V^k d = 0,$$

where $V^k \in \partial_B H(x^k)$. This is the generalized Jacobian-based Newton method proposed in [28]. Condition (11) is clearly satisfied. Thus the Q-superlinear convergence of the sequence produced by this method follows easily from Theorem 2 under the stated assumptions of this result.

Example 2. Suppose that H is B-differentiable. Let d^k be a solution of the (nonlinear) equation

$$H(x^k) + H'(x^k, d) = 0. \tag{13}$$

This is the B-derivative-based Newton method proposed in [22]. When H is semismooth and (13) has a solution, this method becomes a special case of the one in the previous example; see [28], [30]. Hence the Q-superlinear convergence of this B-derivative-based method holds under the assumptions of Theorem 2 and the solvability of (13).

Example 3. Let d^k be a solution of the linear equation

$$H(x^k) + B_k d = 0,$$

where B_k is a member of a certain family of matrices. This is the quasi-Newton formula (9). In this case condition (11) becomes

$$\lim_{k \to \infty} \frac{\|(B_k - V^k)d^k\|}{\|d^k\|} = 0.$$

Hence under this limit property and the assumptions of Theorem 2 the Q-superlinear convergence of the sequence $\{x^*\}$ follows. This conclusion generalizes the result obtained by Ip and Kyparisis [14] under a more restrictive setting.

We consider a generalization of Theorem 1. Suppose that for any given iterate x^k there is a procedure for generating a direction d^k. If we define the next iterate $x^{k+1} = x^k + d^k$, then we have the situation as in Theorem 2. More generally, we may generate x^{k+1} by dampening the direction d^k, i.e.,

$$x^{k+1} = x^k + \lambda_k d^k, \tag{14}$$

where λ_k is a step length satisfying $0 < \lambda_k \leq 1$. (Do not confuse the d^k in (14) with that in Theorem 2; in particular, the former $d^k \neq x^{k+1} - x^k$ unless $\lambda_k = 1$.) It turns out that under the conditions of Theorem 2 if the sequence of directions $\{d^k\}$ satisfies the limit condition (11), then the sequence $\{x^k\}$ generated by (14) converges Q-superlinearly to x^* if and only if the steplength λ_k tends to 1. This result generalizes Corollary 2.3 of [5].

COROLLARY 2. *Let $H : R^n \to R^n$ satisfy the assumptions of Theorem* 2 *on the set D. Suppose that $\{x^k\}$, generated by* (14), *remains in D and converges to x^*. If* (11) *holds for* *the sequence of directions* $\{d^k\}$, *then* $H(x^*) = 0$ *and* $\{x^k\}$ *converges Q-superlinearly to* x^* *if and only if* $\{\lambda_k\}$ *converges to unity.*

Proof. Assume that $\{x^k\}$ converges to Q-superlinearly to x^* and $H(x^*) = 0$. By Theorem 2 we must have

$$\lim_{k\to\infty} \frac{\|\lambda_k^{-1}H(x^k) + V^k d^k\|}{\|d^k\|} = 0. \tag{15}$$

Since (11) holds for $\{d^k\}$, it follows that

$$\lim_{k\to\infty} \frac{\|(\lambda_k^{-1} - 1)H(x^k)\|}{\|d^k\|} = 0,$$

i.e.,

$$\lim_{k\to\infty} \frac{(1 - \lambda_k)\|H(x^k)\|}{\|s^k\|} = 0,$$

where $s^k = x^{k+1} - x^k$. Since $H(x) = 0$, Proposition 3 implies that there is a constant $\beta > 0$ such that $\|H(x^k)\| \geq \beta\|e^k\|$. Since $\{x^k\}$ converges Q-superlinearly to x^*, we have

$$\lim_{k\to\infty} \frac{\|s^k\|}{\|e^k\|} = 1.$$

Consequently, we obtain $\lambda_k \to 1$. The reverse direction of the corollary follows directly from Theorem 2. □

5. Gauss–Newton method. In the context of solving smooth equations the Gauss–Newton method [6] is a well-known numerical procedure that is often used as a globalization scheme of the basic Newton method. The Gauss–Newton method is generalized to a system of locally Lipschitzian equations in [9]. Nevertheless, the direction-generation step in this generalized Gauss–Newton method calls for the solution of a nonlinear program that in general is neither smooth nor convex. Because solving a nonsmooth nonconvex problem is generally quite difficult, the algorithm described in this reference is not likely to be an effective solution procedure in practice. In what follows we propose a variant of this method in which the direction-generation subproblems are convex quadratic programs that are always solvable. Since solving a convex quadratic program is nowadays relatively easy, the new algorithm is more promising.

In essence, the method developed below for the nonsmooth equation (1) is a generalization of the NE/SQP method for the NCP. The reader may want to consult reference [24] for some preliminary discussion of the principal ideas involved and for the omitted details in some of the proofs given here.

Before we begin we remind the reader that the locally Lipschitzian assumption of the function H is still in force. Consider the following constrained nonsmooth least-squares problem:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}H(x)^T H(x) \\ \text{subject to} & x \in X, \end{array} \tag{16}$$

where X is a certain polyhedral set in R^n. A question immediately arises: the original equation (1) is unconstrained, and so why do we suddenly introduce the set X, and how is it related to this question? As we shall see shortly, the presence of the set X actually facilitates the design of the desired algorithm; for several special problems, such as the standard NCP and the NCP with upper bounds, it is very natural to associate an appropriate set X with equation (1); see, e.g., (3). For our purpose here suffice it to consider X an abstract set useful for the construction of the algorithm. Let $\theta : X \to R$ denote the objective function of (16).

5.1. Special assumption. The function θ serves as a merit function for the Gauss–Newton method. Suppose we are given a vector that is not a zero of H; we wish to generate a descent direction at this point along which the value of θ can be decreased. In general, there are several ways to accomplish this; see [9], [12], [22], [23]. In the following we describe a general approach that relies on the following assumption.

Assumption 1. For each i the function $|H_i| : R^n \to R_+$ is upper subdifferentiable on the set

$$\{x \in R^n : H_i(x) \neq 0\} \cap X. \tag{17}$$

We call (17) the *nonzero set* of H_i. Similarly, we call

$$\{x \in R^n : H_i(x) > 0\} \cap X \tag{18}$$

the *positive set* of H_i, and we call

$$\{x \in R^n : H_i(x) < 0\} \cap X \tag{19}$$

the *negative set* of H_i.

It is useful to point out that if x belongs to the positive set (18), then there exists an open neighborhood V of x such that $V \cap X$ is a subset of this positive set. A similar statement holds for the negative set. Hence it follows that $|H_i|$ is upper subdifferentiable on the nonzero set if and only if H_i is upper subdifferentiable on the positive set and $-H_i$ is upper subdifferentiable on the negative set. The following proposition, which shows that Assumption 1 is satisfied by most of the functions H appearing in the problems presented in §2, makes use of this observation. In various parts of the proposition the continuous differentiability of certain functions should be interpreted as being valid on an open set containing the set X in question.

PROPOSITION 7. *All the functions H given below satisfy Assumption* 1 *with the set X as indicated*:

(i) $H(x) = \min(x, f(x)), X = R^n_+$, *provided that f is continuously differentiable*;

(ii) $H(x) = \min(x, f(x)^+) + \min(a - x, f(x)^-), X = [0, a]$, *provided that $a > 0$ and that f is continuously differentiable*;

(iii) *H is given by* (4) *and $X = R^n \times R^p_+ \times R^q$ provided that f is once continuously differentiable and g, h are twice continuously differentiable*;

(iv) $H(x) = \min(f_1(x), \dots, f_N(x))$, $X \subseteq R^n$, *provided that each $f_j : X \to R^n$ is continuously differentiable and that for each $i = 1, \dots, n$ and each $x \in X$ such that $H_i(x) < 0$ the minimum is attained at a unique index j*;

(v) $H(x) = \min(0, g(x)), X \subseteq R^n$, *provided that $g : X \to R^n$ is continuous and that each $-g_i$ is upper subdifferentiable on the set $\{x \in X : g_i(x) < 0\}$*.

Proof. Before giving the proof we observe that the function H in each part is locally Lipschitzian.

(i) The ith component function H_i is the composition of two functions $\phi : R^2 \to R$ and $g : X \to R^2$, where $\phi(a, b) = \min(a, b)$ for $(a, b) \in R^2$ and $g(x) = (x_i, f_i(x))$. Since the min function is concave in its two arguments, the upper subdifferentiability of $|H_i|$

on the positive set (18) follows from Proposition 5. It is easy to see that on the negative set (19), $H_i(x) = f_i(x) < x_i$, which implies $|H_i(x)| = -f_i(x)$. This observation and the assumed differentiability property of f_i easily establish the upper subdifferentiability of $|H_i|$ on the nonzero set.

(ii) Notice that on the set X both summands of H_i are nonnegative; hence the negative set of H_i is empty. Also, both f_i^- and f_i^+ are continuously differentiable on the open set $\{x : f_i(x) \neq 0\}$, which contains the positive set of H_i. Hence both summands of H_i are upper subdifferentiable on the positive set; thus so is $|H_i|$.

(iii) Each component of this function H is either itself continuously differentiable or is defined in terms of the min function. Hence the conclusion follows.

(iv) The assumption implies that $|H_i|$ is F-differentiable on its negative set; hence so is $|H_i|$. On the positive set $|H_i|$ is the composition of the min function (with N arguments) and a continuously differentiable vector-valued function; hence the upper subdifferentiability of $|H_i|$ again follows from Proposition 5. Notice that part (i) is a special case of this result.

(v) Clearly, the function H is nonpositive. Hence the positive set of H_i is empty. On the negative set we have $|H_i(x)| = -g_i(x) > 0$. Hence the upper subdifferentiability $|H_i|$ follows from the assumption of $-g_i$. □

Regrettably, Assumption 1 does not appear to hold for the functions H and $\tilde{H}$ given in §2.3. Although in Proposition 7 we have not explicitly exhibited the upper subdifferential function for each $|H_i|$ on the indicated set, from the proof we can easily construct the required subdifferential function; see also Proposition 5.

5.2. Regularity condition. Because we have associated the minimization problem (16) with (1), it is natural to ask when a stationary point of the former problem is a solution of the latter. To answer this question we recall that the cone of feasible directions of a set X at a point $x \in X$ is defined to be the set

$$\mathcal{F}_X(x) = \{d : x + \varepsilon d \in X \text{ for all sufficiently small } \varepsilon > 0\}.$$

When X is polyhedral (as in our analysis), $\mathcal{F}_X(x)$ is easily identified. The following result provides a necessary and sufficient condition for a stationary point of (16) to solve (1).

PROPOSITION 8. *Let $H : R^n \to R^n$ be B-differentiable and satisfy Assumption* 1. *If x is a stationary point of* (16), *i.e., if $x^* \in X$ and*

$$\theta'(x^*, y - x^*) \geq 0 \quad \text{for all } y \in X,$$

then $H(x^*) = 0$ if and only if for every $a(x^*) = (a_i(x^*)) \in \Pi_{i=1}^n \partial|H_i|(x^*)$ there exists a vector $d \in \mathcal{F}_X(x^*)$ such that for each i such that $H_i(x^*) \neq 0$

$$|H_i(x^*)| + a_i(x^*)^T d \leq 0. \tag{20}$$

Proof. Clearly, if $H(x^*) = 0$, then there is nothing to prove. For the converse we first note that we have

$$\theta'(x, d) = \sum_{i=1}^{n} |H_i(x^*)||H_i|'(x^*, d).$$

Let $a(x^*) \in \Pi_{i=1}^n \partial|H_i|(x^*)$ be such that for each i with $H_i(x^*) \neq 0, a_i(x^*)$ is an upper subgradient of $|H_i|$ at x^*; with this $a(x)$ let $d \in \mathcal{F}_X(x^*)$ satisfy condition (20) for each i such that $H_i(x^*) \neq 0$. Then we have for all i

$$-|H_i(x^*)|^2 \geq |H_i(x^*)|a_i(x^*)^T d \geq |H_i(x^*)||H_i|'(x^*, d).$$

With this vector d, substituting $y = x^* + \varepsilon d \in X$, where $\varepsilon > 0$ is a small enough scalar, into the stationarity condition for θ yields

$$0 \leq \varepsilon\theta'(x^*, d) \leq -\varepsilon H(x^*)^T H(x^*),$$

which implies $H(x^*) = 0$. □

A slightly different version of the above necessary and sufficient condition was introduced in [24] for the NCP. Following the terminology used there, we say that a given vector $x^* \in X$ is *s-regular* if this condition holds at x^*. Note that inequality (20) in this condition is required to hold only for those indices i satisfying $H_i(x^*) \neq 0$. Hence the vectors $a_i(x^*)$ corresponding to $H_i(x^*) = 0$ actually have no role in this regularity property.

5.3. Generation of descent direction. In the rest of this section we further assume that H is B-differentiable and satisfies Assumption 1. For each i let a_i be an upper subgradient function of $|H_i|$ on the nonzero set of H_i. Suppose we have a vector $x \in X$ such that $\theta(x) > 0$. Let $a_i(x)$ be an arbitrary vector in the B-subdifferential $\partial_B H_i(x)$ of H_i at x if $H_i(x) = 0$. Then we have

$$|H_i(x)||H_i|'(x, d) \leq |H_i(x)|a_i(x)^T d$$

for all $x, d \in R^n$ and all i. Define the functions $f\colon X \times R^n \to R_+$ and $z : X \times R^n \to R_+$ by

$$\phi(x, d) = \frac{1}{g}\sum_{i=1}^{n}(|H_i(x)| + a_i(x)^T d)^2 \quad \text{and} \quad z(x, d) = \frac{1}{2}\sum_{i=1}^{n}(a_i(x)^T d)^2.$$

The proposition below summarizes three important properties of these two functions.

PROPOSITION 9. *The following properties hold*:

(a) $\phi(x, 0) = \theta(x)$ *for all* $x \in X$,

(b) $\phi(x, d) - \phi(x, 0) - z(x, d) \geq \theta'(x, d)$ *for all* $(x, d) \in X \times R^n$,

(c) $\lim_{(u,d)\to(x,0)} \phi(u, d) = \phi(x, 0)$.

Proof. The first two assertions are fairly straightforward to prove. We prove only part (c). In turn, it suffices to show that all vectors in the collection $\{a_i(u)\}$ are bounded in norm by a constant for all u sufficiently close to x. But this follows from Corollary 1. □

For a given vector $x \in X$ consider the following convex quadratic programming problem in d, which we denote by (QP$_x$):

$$\begin{array}{ll} \text{minimize} & \phi(x, d) \\ \text{subject to} & x + d \in X. \end{array}$$

The proposition below summarizes the main properties of this quadratic program.

PROPOSITION 10. *Let $x \in X$ be given. The following properties hold for the problem* (QP_x):

(a) $d = 0$ *is feasible and an optimal solution, say,* $\tilde{d}$, *that satisfies* $\phi(x, \tilde{d}) \leq \theta(x)$, *always exists*,

(b) $z(x, \tilde{d}) \leq \theta(x)$;

(c) $\phi(x, \tilde{d}) = \theta(x)$ *if and only if* $z(x, \tilde{d}) = 0$; *if* $\phi(x, \tilde{d}) = \theta(x)$ *and if x is an s-regular vector, then* $\theta(x) = 0$;

(d) *if* $\phi\,(x, \tilde{d}) < \theta(x)$, *then for any* $\sigma \in (0, 1)$ *there exists a scalar* $\bar{\tau} > 0$ *such that for all* $\tau \in [0, \bar{\tau}]$

$$\theta(x + \tau\tilde{d}) - \theta(x) \leq -\sigma\tau z(x, \tilde{d}).$$

Proof. That $d = 0$ is feasible for (QP$_x$) is obvious. The existence of an optimal solution is a consequence of the well-known Frank–Wolfe theorem for quadratic programming [7]. The last conclusion of this part is trivial. To prove part (b), we note that by the minimum principle

$$0 \le \nabla_d \phi(x, \tilde{d})^T (d - \tilde{d}) = \sum_{i=1}^{n} (|H_i(x)| + a_i(x)^T \tilde{d}) a_i(x)^T (d - \tilde{d})$$

for every d feasible to (QP$_x$). In particular, substituting $d = 0$ and rearranging terms, we obtain

$$\begin{aligned} 2z(x, \tilde{d}) &\le -\sum_{i=1}^{n} |H_i(x)| a_i(x)^T \tilde{d} \\ &\le \frac{1}{2} \sum_{i=1}^{n} (H_i(x)^2 + (a_i(x)^T \tilde{d})^2), \end{aligned} \tag{21}$$

which easily yields part (b).

To prove part (c) suppose $\phi(x, \tilde{d}) = \theta(x)$. Then we have

$$z(x, \tilde{d}) = -\sum_{i=1}^{n} |H_i(x)| a_i(x)^T \tilde{d} \ge 2z(x, \tilde{d}),$$

where the last inequality has just been proved. Since $z(x, \tilde{d})$ is nonnegative, we must have $z(x, \tilde{d}) = 0$. Conversely, if $z(x, \tilde{d}) = 0$, then we have $a_i(x)^T \tilde{d} = 0$ for all i, which clearly implies $\phi(x, \tilde{d}) = \theta(x)$. Now suppose that $\phi(x, \tilde{d}) = \theta(x)$ and that x is s-regular. Let $d \in \mathcal{F}_X(x)$ be such that (20) holds for each i with $H_i(x) \ne 0$. Then for all $\varepsilon > 0$ sufficiently small we have $x + \epsilon d \in X$. Hence εd is feasible for (QP$_x$) and

$$\theta(x) \le \phi(x, \varepsilon d) = \frac{1}{2} \left[\sum_{i : H_i(x) \ne 0} (|H_i(x)| + \varepsilon a_i(x)^T d)^2 + \sum_{i : H_i(x) = 0} (\varepsilon a_i(x)^T d)^2 \right].$$

For an index i such that $H_i(x) \ne 0$ we have for $\varepsilon > 0$ small enough

$$0 \le |H_i(x)| + \varepsilon a_i(x)^T d \le (1 - \varepsilon) |H_i(x)|.$$

Consequently, it follows that

$$\theta(x) \le (1 - \varepsilon)^2 \theta(x) + \frac{\varepsilon^2}{2} \sum_{i : H_i(x) = 0} (a_i(x)^T d)^2.$$

Since this inequality must hold for all $\varepsilon > 0$ sufficiently small, it follows that $\theta(x) = 0$, as desired.

For part (d) it follows from Assumption 1 and part (b) of Proposition 9 that $\theta'(x, \tilde{d}) < -z(x, \tilde{d}) \le 0$. From this the desired conclusion is immediate. □

5.4. The method and its convergence. We now describe the Gauss–Newton method for solving the nonsmooth equation (1) under the setting given above.

Let $\rho, \sigma \in (0,1)$ be given scalars. Let $x^0 \in X$ be arbitrary. In general, given $x^k \in X$, solve the quadratic program (QP$_{x^k}$) and let d^k be any optimal solution. If $\phi(x^k, d^k) = \theta(x^k)$, terminate. Otherwise, let the step length $\tau_k = \rho^{m_k}$, where m_k is the smallest nonnegative integer m for which

$$\theta(x^k + \rho^m d^k) - \theta(x^k) \leq -\sigma \rho^m z(x^k, d^k).$$

Set $x^{k+1} = x^k + \tau_k d^k$, and repeat the general step.

We refer the reader to [24] for a more detailed explanation of the individual steps of the algorithm. In the following we assume that the algorithm generates an infinite sequence of iterates $\{x^k\}$ and a corresponding sequence of descent directions $\{d^k\}$. We wish to investigate the limiting behavior of $\{x^k\}$. For this purpose we assume that this sequence is bounded; thus it has at least one accumulation point, which we denote x^*. Clearly, $x^* \in X$. Our goal is to show that if certain regularity conditions hold at x^*, then $H(x^*) = 0$. Let $\{x^k : k \in \kappa\}$ be the subsequence whose limit is x^*.

Our first step in the convergence analysis to derive some properties of the sequence $\{d^k : k \in \kappa\}$. We say that x^* satisfies the *generalized b-regularity property* if there exist a neighborhood V of x^* and a positive scalar $c > 0$ such that for any upper subgradient function $a_i(\cdot)$ of $|H_i|$ on the nonzero set of H_i and for any vector $a_j(z) \in \partial_B H_j(z)$ if $H_j(z) = 0$, the matrix $a(x)$ whose rows are the vectors $(a_i(x)^T)$ is nonsingular and satisfies $\|a(x)^{-1}\| \leq c$ for all $x \in V$.

In principle, the results to be derived below will all remain valid if we restrict the vectors $a_i(x)$ in the generalized b-regularity property to the particular one used in the above-formulated Gauss–Newton method. With such a restriction this regularity property reduces essentially to the b-regularity property defined in [24] for the NCP.

With the generalized b-regularity condition we prove the following result.

LEMMA 1. *Suppose that x^* is the limit of the subsequence $\{x^k : k \in \kappa\}$ and that x^* satisfies the generalized b-regularity property. Then*

(i) *there exists a constant $\lambda > 0$ such that for all $k \in \kappa$ sufficiently large*

$$\lambda \|d^k\|^2 \leq z(x^k, d^k) \leq \theta(x^k);$$

(ii) $\lim_{k \in \kappa, k \to \infty} z(x^k, d^k) = 0 = \lim_{k \in \kappa, k \to \infty} \|d^k\|$.

Proof. The generalized b-regularity property implies that for some constant $c > 0$ and all $k \in \kappa$ sufficiently large we have $\|a(x^k)^{-1}\| \leq c$. Consequently, the left-hand inequality in part (i) follows easily with $\lambda = c^{-2}$. The right-hand inequality is part (b) of Proposition 10.

To prove part (ii) it suffices to show the first equality. We follow the argument used in Lemmas 4 and 5 in [24]. We note that part (i) implies that the sequences $\{d^k : k \in \kappa\}$ and $\{z(x^k, d^k) : k \in \kappa\}$ are bounded. Without loss of generality, we may assume that the latter sequence converges. As in [24, Lemma 4], we show that for any sequence of positive scalars $\{\lambda_k : k \in \kappa\}$ converging to zero

$$\limsup_{k \to \infty, k \in \kappa} \frac{\theta(x^k + \lambda_k d^k) - \theta(x^k)}{\lambda_k} \leq - \lim_{k \to \infty, k \in \kappa} z(x^k, d^k). \tag{22}$$

If i is an index such that $H_i(x^*) = 0$, then we have

$$\lim_{k \in \kappa, k \to \infty} H_i(x^k) = \lim_{k \in \kappa, k \to \infty} H_i(x^k + \lambda_k d^k) = 0.$$

Since

$$|H_i|^2(x^k+\lambda_k d^k)-|H_i|^2(x^k)=(|H_i|(x^k+\lambda_k d^k)-|H_i|(x^k))\\ \cdot(|H_i|(x^k+\lambda_k d^k)+|H_i|(x^k)),$$

by the local Lipschitz continuity of H_i we deduce

$$\limsup_{k\to\infty,k\in\kappa}\frac{|H_i|^2(x^k+\lambda_k d^k)-|H_i|^2(x^k)}{\lambda_k}=0.$$

On the other hand, if the index i is such that $H_i(x^*)\neq 0$, then $H_i(x^k)\neq 0$ for all $k\in\kappa$ sufficiently large. For such an index i we may write

$$|H_i|^2(x^k+\lambda_k d^k)-|H_i|^2(x^k)=T_{1,i}+T_{2,i}+T_{3,i},$$

where

$$\begin{aligned}T_{1,i}&=(|H_i|(x^k+\lambda_k d^k)-|H_i|(x^k)-\lambda_k a_i(x^k)^T d^k)\\ &\quad\cdot(|H_i|(x^k+\lambda_k d^k)+|H_i|(x^k)),\\ T_{2,i}&=2\lambda_k|H_i|(x^k)a_i(x^k)^T d^k,\\ T_{3,i}&=\lambda_k(|H_i|(x^k+\lambda_k d^k)-|H_i|(x^k))a_i(x^k)^T d^k.\end{aligned}$$

By the boundedness of $\{a_i(x^k):k\in\kappa\}$ and $\{d^k:k\in\kappa\}$ and the local Lipschitz continuity of H it follows that

$$\limsup_{k\to\infty,k\in\kappa}\frac{T_{3,i}}{\lambda_k}\leq 0.$$

In addition, by the upper subdifferentiability of $|H_i|$ on the nonzero set of H_i we deduce

$$\limsup_{k\to\infty,k\in\kappa}\frac{T_{1,i}}{\lambda_k}\leq 0.$$

By the first inequality in (21) we deduce

$$\limsup_{k\to\infty,k\in\kappa}\frac{\sum_{i=1}^n T_{2,i}}{\lambda_k}\leq -4\lim_{k\to\infty,\kappa\in\kappa}z(x^k,d^k)\leq -2\lim_{k\to\infty,\kappa\in\kappa}z(x^k,d^k),$$

where the last inequality follows since $z(x^k,d^k)$ is nonnegative. Consequently, we obtain the desired inequality (22) readily.

To complete the proof of part (ii) we use a standard argument related to the Armijo step-length procedure. Since this is rather routine, we omit it and refer the reader to [24, Lemma 5] for more details. (Inequality (22) is key to the omitted argument.) □

The next result asserts a technical property of the cone of feasible directions associated with a polyhedron.

LEMMA 2. *Let X be a polyhedron, and let $x\in X$. If $d\in\mathcal{F}_X(x)$, then there exist positive scalars $\bar{\varepsilon},\delta$ such that $y+\varepsilon d\in X$ for every $\varepsilon\in[0,\bar{\varepsilon}]$ and every $y\in X$ such that $\|y-x\|\leq\delta$.*

Proof. Write $X=\{x:Ax\geq b,Cx=d\}$. Then

$$\mathcal{F}_X(x)=\bigcap_{i\in I(x)}\{d:A_i d\geq 0,Cd=0\},$$

where $I(x)$ is the index set of the binding (inequality) constraints at x. For a vector $y \in X$ that is sufficiently close to x we must have $I(y) \subseteq I(x)$. Using this fact, we can easily deduce the existence of the positive scalars $\bar{\varepsilon}$ and δ with the desired properties. □

Remark. The proof of Lemma 2 shows that $\mathcal{F}_X(x) \subseteq \mathcal{F}_X(y)$ for all $y \in X$ sufficiently close to x. The important point of this lemma is that for each direction $d \in \mathcal{F}_X(x)$ there exists a constant $\bar{\varepsilon} > 0$ that applies uniformly to all such y.

Combining the above results, we may establish the desired zero property of the limit point x^*.

THEOREM 3. *Suppose that x^* is the limit of the subsequence $\{x^k : k \in \kappa\}$ and that x^* is s-regular and satisfies the generalized b-regularity property. Then $H(x^*) = 0$.*

Proof. The sequence of matrices $\{a(x^k) : k \in \kappa\}$ is bounded. Without loss of generality, we may assume that it converges to some matrix A. By Corollary 1, $A_i \in \partial|H_i|(x^*)$ for each i with $H_i(x^*) \neq 0$. Associated with this matrix A let $d \in \mathcal{F}_X(x^*)$ be a vector involved in the s-regularity of x^*. By Lemma 2 there exists an $\bar{\varepsilon} > 0$ such that for all $\varepsilon \in [0, \bar{\varepsilon}]$ and all $k \in \kappa$ sufficiently large we have $x^k + \varepsilon d \in X$; hence $\phi(x^k, d^k) \leq \phi(x^k, \varepsilon d)$. By following the proof of Theorem 1 in [24] it suffices to prove the inequality

$$(23) \qquad \limsup_{k \in \kappa, k \to \infty} \phi(x^k, \varepsilon d) \leq (1-\varepsilon)^2 \phi(x^*, 0) + O(\varepsilon^2)$$

for all $\varepsilon > 0$ sufficiently small. We may write

$$\phi(x^k, \varepsilon d) = \frac{1}{2}\left[\sum_{i: H_i(x^*) \neq 0} (|H_i(x^k)| + \varepsilon a_i(x^k)^T d)^2 + \sum_{i: H_i(x^*) = 0} (|H_i(x^k)| + \varepsilon a_i(x^k)^T d)^2\right].$$

Notice that if $H_i(x^*) = 0$, then $\lim_{k \in \kappa, k \to \infty} H_i(x^k) = 0$. Hence in the limit the second summand in the square brackets becomes $O(\varepsilon^2)$. For the first summand note that we have

$$\lim_{k \to \infty, k \in \kappa} (|H_i(x^k)| + \varepsilon a_i(x^k)^T d) = |H_i(x^*)| + \varepsilon A_i^T d \leq (1-\varepsilon)|H_i(x^*)|.$$

So if $H_i(x^*) \neq 0$, then for all $\varepsilon > 0$ sufficiently small

$$0 < |H_i(x^*)| + \varepsilon A_i^T d,$$

which implies

$$\lim_{k \to \infty, k \in \kappa} (|H_i(x^k)| + \varepsilon a_i(x^k)^T d)^2 \leq (1-\varepsilon)^2 |H_i(x^*)|^2.$$

The desired inequality (23) now follows easily. To complete the proof we note that since $\{d^k : k \in \kappa\} \to 0$ as $k \to \infty$, parts (a) and (c) of Proposition 9 imply that

$$\theta(x^*) = \limsup_{k \to \infty, k \in \kappa} \phi(x^k, d^k) \leq (1-\varepsilon)^2 \theta(x^*) + O(\varepsilon^2),$$

which easily yields $\theta(x^*) = 0$ since $\varepsilon > 0$ is arbitrary. □

Remark. It is worthwhile to mention that the limit point x^* is actually a constrained zero of H; that is, we also have $x^* \in X$. For several problems presented in §2 this is quite natural because by its construction the set X must contain all zeroes of H (see parts (i), (ii), and (iii) in Proposition 7).

Having established the zero property of x^*, we proceed to establish a stronger convergence property of the sequence $\{x^*\}$. For this purpose we need to make the additional assumption that H is semismooth at x^*. Again, our strategy is to extend the argument in [24] to the present more general framework. We first establish a lemma that generalizes Lemma 8 in this reference.

LEMMA 3. *Let $\bar{x} \in X$ be a solution of* (1). *Suppose that $\bar{x}$ satisfies the generalized b-regularity property and that H is semismooth at $\bar{x}$. Then for every $\varepsilon > 0$ there exists a $\delta > 0$ such that whenever $z \in X$ satisfies $\|z - \bar{x}\| \leq \delta$,*

$$\|z + \tilde{d} - \bar{x}\| \leq \varepsilon \|z - \bar{x}\|, \tag{24}$$

where $\tilde{d}$ is any optimal solution of the problem (QP_z).

Proof. By proceeding as in Lemma 8 of [24] and by letting $v = \bar{x} - z - \tilde{d}$, we may deduce

$$\|a(z)v\| \leq \|\, |H|(\bar{x}) - |H|(z) - a(z)(\bar{x} - z)\|$$

for every $z \in X$. By assumption, H is semismooth at $\bar{x}$; hence so are $|H|$ and every component function $|H_i|$. Hence by a variant of Proposition 1 we have

$$\lim_{z \in X, z \to \bar{x}} \frac{\|\, |H_i|(\bar{x}) - |H_i|(z) - a_i(z)^T(\bar{x} - z)\|}{\|\bar{x} - z\|} = 0.$$

By the generalized b-regularity of $\bar{x}$ there exists a constant $c > 0$ such that for every vector $z \in X$ sufficiently close to $\bar{x}$ we have

$$\|a(z)v\| \geq c\|v\|.$$

Combining the last three expressions, we easily derive the desired conclusion of the lemma. □

By using Lemma 3 and part (a) of Proposition 9, one can show (cf. the proof of Lemma 9 in [24]) that under the stated assumptions of the lemma, there exists a constant $c > 0$ such that if $L > 0$ is the Lipschitzian modulus of H at $\bar{x}$, then for $\varepsilon \in (0, 1)$, if z and $\tilde{d}$ are as stated in this lemma,

$$\theta(z + \tilde{d}) \leq \left(\frac{\varepsilon c L}{1 - \varepsilon}\right)^2 \theta(z).$$

Using this inequality and the subsequential convergence of $\{x^k\}$, one can easily establish the following additional convergence properties of the Gauss–Newton method.

THEOREM 4. *Let $H : R^n \to R^n$ be B-differentiable and satisfy Assumption* 1. *Suppose that x^* is a limit point of a sequence $\{x^k\}$ produced by the Gauss–Newton method. If x^* is s-regular and satisfies the generalized b-regularity property, then $H(x^*) = 0$. Moreover, if in addition H is semismooth at x^*, then*

(i) *there exists an integer $K > 0$ such that for all $k \geq K$ the step length $\tau_k = 1$; hence $x^{k+1} = x^k + d^k$;*

(ii) *the sequence $\{x^k\}$ converges to x^* Q-superlinearly.*

Proof. Part (i) follows from the argument sketched above with a choice of $\varepsilon \in (0, 1)$ satisfying

$$1 - \left(\frac{\varepsilon c L}{1 - \varepsilon}\right)^2 > \sigma.$$

Part (ii) follows from the proof of Lemma 3 and part (i). □

It is natural to wonder how Corollary 2 is related to Theorem 4. In essence, the proof of the theorem consists of a direct verification of the conditions stipulated by this corollary applied to the absolute value function $|H|$. Leaving out the details, we mention that by using the semismoothness property of H at x^* and the fact that

$$\|x^k + d^k - x^*\| = \mathrm{O}(\| \, |H|(x^*) - |H|(x^k) - a(x^k)(x^* - x^k)\|)$$

(as established in the proof of Lemma 3), one can verify that condition (11), with H replaced by $|H|$, holds for the sequence of directions $\{d^k\}$ generated by the Gauss–Newton method. Moreover, that the entire sequence $\{x^k\}$ converges to x^* and that the sequence of step lengths $\{\tau_k\} \rightarrow 1$ also follow from the above expression.

In conclusion, we mention that computational results for the method described herein can be found in [8] and [24]. In particular, [8] reports some computational experience with the Gauss–Newton method applied to solve the NCP with upper bounds by using the function H described in §2.2.

Acknowledgments. The authors are grateful to the Editor, Michael L. Overton, for his efficient handling of this paper and to the referees for their incisive comments, which helped to improve the presentation of the paper and to simplify some proofs. One referee recommended the use of the word upper to describe the subdifferentiability concept defined in §3.3.

REFERENCES

[1] J. R. Birge and L. Qi, *Semiregularity and Generalized Subdifferentials with Applications to Stochastic Programming*, Applied Mathematics Preprint 89/12, School of Mathematics, University of New South Wales, Sydney, Australia, revised April 1991.

[2] J. V. Burke and L. Qi, *Weak directional closedness and generalized subdifferentials*, J. Math. Anal. Appl., 159 (1991), pp. 485–499.

[3] J. Brézis, *Opérateurs Maximaux Monotones*, North-Holland, Amsterdam, 1973.

[4] F. H. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley, New York, 1983.

[5] J. E. Dennis and J. J. Moré, *A characterization of the superlinear convergence and its application to quasi-Newton methods*, Math. Comp., 28 (1974), pp. 549–560.

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

[7] M. Frank and P. Wolfe, *An algorithm for quadratic programming*, Naval Res. Logist., 3 (1956), pp. 95–110.

[8] S. A. Gabriel, *Algorithms for the Nonlinear Complementarity Problem: The* NE/SQP *Method and Extensions*, Ph.D. thesis, Department of Mathematical Sciences, Johns Hopkins University, Baltimore, MD, 1992.

[9] S. P. Han, J. S. Pang, and N. Rangaraj, *Globally convergent Newton methods for nonsmooth equations*, Math. Oper. Res., 17 (1992), pp. 586–607.

[10] P. T. Harker and J. S. Pang, *Finite-dimensional variational inequality and nonlinear complementarity problem: A survey of theory, algorithms and applications*, Math. Programming, 48 (1990), pp. 161–220.

[11] ———, *Modelling and Computation of Equilibria: A Variational Inequality Approach*, Academic Press, New York, to appear.

[12] P. T. Harker and B. Xiao, *Newton's method for the nonlinear complementarity problem: A B-differentiable equation approach*, Math. Programming, 48 (1990), pp. 339–357.

[13] A. D. Ioffe, *Calculus of Dini subdifferentials of functions and contingent coderivatives of set-valued maps*, Nonlinear Anal., 8 (1984), pp. 517–539.

[14] C. M. Ip and J. Kyparisis, *Local convergence of quasi-Newton methods for B-differentiable equations*, Math. Programming, 56 (1992), pp. 71–90.

[15] M. Kojima, *Strongly stable stationary solutions in nonlinear programming*, in Analysis and Computation of Fixed Points, S. M. Robinson, ed., Academic Press, New York, 1980, pp. 93–138.

[16] M. Kojima and S. Shindo, *Extensions of Newton and quasi-Newton methods to systems of PC^1 equations*, J. Oper. Res. Soc. Japan, 29 (1986), pp. 352–374.

[17] B. Kummer, *Newton's method for non-differentiable functions*, in Advances in Mathematical Optimization, J. Guddat, B. Bank, H. Hollatz, P. Kall, D. Karte, B. Kummer, K. Lommatzsch, L. Tammer, M. Vlach, and K. Zimmermann, eds., Akademi-Verlag, Berlin, 1988, pp. 114–125.

[18] P. Michel and J. P. Penot, *Calcul sous-differentiel pour des fonctions lipschitziennes et non lipschitziennes*, C. R. Acad. Sci. Paris, 298 (1984), pp. 269–272.

[19] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control Optim., 15 (1977), pp. 957–972.

[20] K. P. Oh, *The formulation of the mixed lubrication problem as a generalized nonlinear complementarity problem*, Trans. ASME, 108 (1986), pp. 598–604.

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

[22] J. S. Pang, *Newton's method for B-differentiable equations*, Math. Oper. Res., 15 (1990), pp. 311–341.

[23] ———, *A B-differentiable equation based, globally, and locally quadratically convergent algorithm for nonlinear programs, complementarity and variational inequality problems*, Math. Programming, 51 (1991), pp. 101–131.

[24] J. S. Pang and S. A. Gabriel, NE/SQP: *A robust algorithm for the nonlinear complementarity problem*, Math. Programming, to appear.

[25] J. P. Penot, *Calcul sous-différentiel et optimisation*, J. Funct. Anal., 27 (1978), pp. 248–276.

[26] F. Plastria, *Lower subdifferentiable functions and their minimization by cutting planes*, J. Optim. Theory Appl., 46 (1985), pp. 37–53.

[27] R. Poliquin, *Subgradient monotonicity and convex functions*, Nonlinear Anal., 14 (1990), pp. 305–317.

[28] L. Qi, *Convergence analysis of some algorithms for solving nonsmooth equations*, Math. Oper. Res., 18 (1993), pp. 227–244.

[29] ———, *LC^1 Functions and LC^1 Optimization Problems*, Applied Mathematics Preprint 91/21, School of Mathematics, University of New South Wales, Sydney, Australia, 1991.

[30] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Programming, to appear.

[31] D. Ralph, *Global convergence of damped Newton's method for nonsmooth equations via the path search*, Math. Oper. Res., to appear.

[32] S. M. Robinson, *Generalized equations*, in Mathematical Programming: The State of the Art, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 346–367.

[33] ———, *Local structure of feasible sets in nonlinear programming, part* III: *Stability and sensitivity*, Math. Programming Stud., 30 (1987), pp. 45–66.

[34] ———, *Newton's method for a Class of Nonsmooth Functions*, Industrial Engineering Working Paper, University of Wisconsin, Madison, WI, 1988.

[35] ———, *An implicit function theorem for a class of nonsmooth functions*, Math. Oper. Res., 16 (1991), pp. 292–309.

[36] ———, *Normal maps induced by linear transformations*, Math. Oper. Res., 17 (1992), pp. 691–714.

[37] R. T. Rockafellar, *Maximal monotone relations and the second derivatives of nonsmooth functions*, An. Inst. H. Poincaré Anal. Non Linear, 2 (1985), pp. 167–184.

[38] ———, *First- and second-order epi-differentiability in nonlinear programming*, Trans. Amer. Math. Soc., 307 (1988), pp. 75–108.

[39] ———, *Computational schemes for solving large-scale problems in extended linear-quadratic programming*, Math. Programming, 48 (1990), pp. 447–474.

[40] R. T. Rockafellar and R. J. B. Wets, *Generalized linear-quadratic problems of deterministic and stochastic optimal control in discrete time*, SIAM J. Control Optim., 28 (1990), pp. 810–822.

[41] A. Shapiro, *On concepts of directional differentiability*, J. Optim. Theory Appl., 66 (1990), pp. 477–487.

A NEWTON METHOD FOR CONVEX REGRESSION, DATA SMOOTHING, AND QUADRATIC PROGRAMMING WITH BOUNDED CONSTRAINTS*

WU LI† AND JOHN SWETITS†

Abstract. This paper formulates systems of piecewise linear equations, derived from the Karush–Kuhn–Tucker conditions for constrained convex optimization problems, as unconstrained minimization problems in which the objective function is a multivariate quadratic spline. Such formulations provide new ways of developing efficient algorithms for many optimization problems, such as the convex regression problem, the least-distance problem, the symmetric monotone linear complementarity problem, and the convex quadratic programming problem with bounded constraints. Theoretical results, a description of an algorithm and its implementation, and numerical results are presented along with a stability analysis.

Key words. data smoothing, Newton methods, convex regression, convex quadratic programs, unconstrained minimization of convex quadratic spline function

AMS subject classifications. primary 90C20, 90C33, 49M15; secondary 90C90, 90C31, 41A29I, 15A12, 62F30

1. Introduction. In this paper we reformulate systems of piecewise linear equations, derived from the Karush–Kuhn–Tucker conditions of constrained convex optimization problems, as unconstrained minimization problems. Such reformulations provide new ways of developing efficient algorithms for many optimization problems, such as the convex regression problem, the least-distance problem, the symmetric monotone linear complementarity problem, and the convex quadratic programming problem with bounded constraints. Our computational effort is focused on the least-distance problem with k-convex constraints:

$$\frac{1}{2}\|c-\hat{x}\|^2 = \min_{\nabla_k x \geq 0} \frac{1}{2}\|c-x\|^2, \tag{1.1}$$

where $c, x, \hat{x}$ are vectors in the $(n+k)$-dimensional Euclidean space $\mathbb{R}^{n+k}$, ∇_k is the kth order divided difference matrix defined by

$$(\nabla_k x)_j = \sum_{i=0}^{k} \binom{k}{i} (-1)^{k-i} x_{j+i} \quad \text{for } j = 1, \ldots, n,$$

$\nabla_k \hat{x} \geq 0$, and $\|\cdot\|$ denotes the 2-norm or the Euclidean norm on $\mathbb{R}^{n+k}$.

Some special cases of (1.1) are the monotone regression problem (for $k = 1$) [2], [38] and the convex regression problem (for $k = 2$) in statistics [38]. Equation (1.1) can also be considered as a data smoothing problem [5], [6] and is known as the best k-convex approximation problem in approximation theory (see [40], [45], [46], [49]).

For $k = 1$, (1.1) has been extensively studied by statisticians interested in statistical inferences under order restrictions [2], [38], a research area that has exploded with new developments and was included as a new AMS classification recently. There are many special algorithms that take advantage of the simple structure of (1.1) for $k = 1$. The most widely used algorithm is the pool-adjacent-violators algorithm by Ayer, Brunk, Ewing, and Reid [1]. The procedure first finds the greatest convex minorant of c and then computes the solution. There are only a few papers on the convex regression problem [38]. The motivation for studying the convex regression problem was to model utility

*Received by the editors May 28, 1991; accepted for publication (in revised form) April 6, 1992.

†Department of Mathematics and Statistics, Old Dominion University, Norfolk, Virginia 23529.

functions and functions representing productivity (etc.) in economics [13], [14]. The well-known Hildreth algorithm was originally invented to solve such a problem [15]. (Now the Hildreth algorithm and its variations are called row-action algorithms [3], [17], [18], which are special cases of the matrix splitting methods (see [22]).) In general, one could treat (1.1) as a special case of separable strictly convex quadratic programming problems and could solve the problem by using algorithms for convex quadratic programming problems and the associated linear complementary problem. For relevant results the reader is referred to [22], [30], [35].

Cullinan has used (1.1) as a model for data smoothing problems in applied numerical analysis [5]. The computation is done by an active set method, which takes advantage of the banded structure of ∇_k in its implementation. Numerical results were presented for various $k \leq 16$ and $n = 51, 101, 400$. The c are values of a function with small perturbations by a noisy random vector that satisfies a normal distribution. A detailed analysis of Cullinan's experiments is given in §8. The evaluation of the performance of his method is not conclusive. The method worked in some cases and failed in other cases. To our knowledge there have been no satisfactory algorithms for solving (1.1) with $k \geq 2$.

In general, the Karush–Kuhn–Tucker conditions can be reformulated as a system of piecewise linear equations. For many (strictly) convex optimization problems we shall show that the solution(s) of such a system are the solution(s) of the unconstrained minimization of a (strictly) convex multivariate quadratic spline function. Such a reformulation of (1.1) allows one to develop accurate and efficient algorithms for finding a solution of (1.1). We shall use the classical Newton method with exact line search for solving (1.1). Partially because of the special structure of (1.1), there are three features of our algorithm and its implementation:

(1) It is a descent method and is able to solve (1.1) with large n (e.g., $n = 2000$).

(2) It finds the solution in a finite number of iterations in exact arithmetic.

(3) An *a priori* estimate of the error between the approximate solution and the exact solution is given as a stopping criterion for the algorithm.

The method is also applicable to strictly convex quadratic programming problems with bounded constraints and the classical linear complementarity problem with a symmetric positive definite matrix. The efficiency of our algorithm also makes it useful as a tool for solving symmetric subproblems of the matrix splitting method [4], [20], [22]–[27].

Numerical experiments are done for various k and n: (1) $1 \leq k \leq 6$, $n = 50, 100$; (2) $1 \leq k \leq 3$, $n = 200, 400$; (3) $k = 1, 2$, $n = 1000, 2000$. The vector c is generated by one of the following elementary functions:

$$\begin{array}{cccccc} \sqrt{t}, & t^2, & \exp(t), & \sin(\pi t), & \sin(2\pi t), & \sin(4\pi t), \\ t, & t^3, & \exp(-t), & \cos(\pi t), & \cos(2\pi t), & \cos(4\pi t), \end{array}$$

perturbed by a random vector of magnitude 0.1. In the cases for which $n^k \leq 10^9$ the algorithm performs very well and we have a very small *a priori* error estimate for all approximate solutions. The algorithm's capability of handling a large amount of data for $k = 1, 2$ makes it computationally feasible to compute the best approximation of a given function by the monotone (or convex) functions. This has been studied for shape-preserving (or constrained) approximation problems [41]–[44], [49].

However, when n^k is large the algorithm fails to find a good approximate solution. We have to be very cautious in our interpretation of numerical results. Lin and Pang [22] recognized that the equivalent linear complementarity problem of (1.1) for $k = 2$ is an ill-conditioned problem and reported an unsuccessful attempt to apply matrix splitting methods to solve it (with $n = 50$). We shall give a rigorous stability analysis of (1.1) that explains mathematically why (1.1) itself is also an ill-conditioned problem. Despite the ill conditioning, we are convinced that the k-convex approximation is a good data smoothing technique.

The contents of this paper are organized as follows. In §2 we reformulate as unconstrained minimization problems the least-distance problem, the symmetric monotone linear complementarity problem, and the convex quadratic programming problem with bounded constraints. In §3 we outline the Newton method with exact line search for the reformulated unconstrained minimization problems. In §4 we estimate the condition numbers of kth divided difference matrices ∇_k. Section 5 is devoted to a stability analysis of the least-distance problem, including (1.1), and §6 contains implementation details of the algorithm for solving (1.1). In §7 we evaluate the performance of our algorithm, and in §8 we present our argument that (1.1) is an effective data smoothing technique. A summary is given in §9. Extensive numerical results are included in the Appendix.

2. Equivalent unconstrained minimization problem. In this section we show that systems of piecewise linear equations, derived from the Karush–Kuhn–Tucker conditions for constrained optimization problems, can be reformulated as an unconstrained minimization problem. In particular, we can transform the convex quadratic programming problem with bounded constraints, the symmetric monotone linear complementarity problem, and the least-distance problem to equivalent unconstrained minimization problems.

Consider the convex quadratic programming problem with bounded constraints (see [29] and the references therein):

$$\min_{l\le x\le u} \tfrac{1}{2}x^TMx+q^Tx, \tag{2.1}$$

where M is a symmetric positive semidefinite $n\times n$ matrix, $q\in\mathbb{R}^n$, vectors l, u specify bounds on x, and some components of l, u can be $\pm\infty$. The Karush–Kuhn–Tucker conditions of (2.1) form the following special affine variational problem:

$$\text{find a } w \text{ satisfying } l\le w\le u \text{ and } (x-w)^T(Mw+q)\ge 0 \text{ for } l\le x\le u. \tag{2.2}$$

It is not difficult to verify that w is a solution of (2.2) if and only if w satisfies the following system of piecewise linear equations:

$$w=(w-\alpha(Mw+q))_l^u, \tag{2.3}$$

where $\alpha>0$ is any constant and $(x)_l$ (or $(x)_u$) is the lower (or upper) truncation of x by l (or u) whose ith component is $\max\{l_i,x_i\}$ (or $\min\{u_i,x_i\}$). Define the multivariate quadratic spline function

$$\begin{aligned} f(w) := {} & \tfrac{1}{2}w^T(I-\alpha M)w-\tfrac{1}{2}\|(w-\alpha(Mw+q))_l^u\|^2 \\ & -l^T(w-\alpha(Mw+q))^l-u^T(w-\alpha(Mw+q))_u, \end{aligned} \tag{2.4}$$

with the convention $-\infty(\cdot)^{-\infty}=\infty(\cdot)_\infty=0$. Then, by straightforward computation one can verify that the gradient f' has the following form:

$$f'(w)=(I-\alpha M)(w-(w-\alpha(Mw+q))_l^u). \tag{2.5}$$

Before the proof of the equivalence of (2.3) and the unconstrained minimization of f we need a few results about monotone mappings. A mapping φ from $\mathbb{R}^n$ to $\mathbb{R}^n$ is called a monotone mapping if

$$(x-y)^T(\varphi(x)-\varphi(y))\ge 0 \quad \text{for } x,y\in\mathbb{R}^n.$$

The mapping φ is said to be uniformly monotone if there exists a constant $\gamma > 0$ such that

$$(x-y)^T(\varphi(x)-\varphi(y)) \geq \gamma \cdot \|x-y\|^2 \quad \text{for } x, y \in \mathbb{R}^n.$$

We say that φ is a piecewise linear mapping if $\varphi(x) = Q^T((Ax+b)_+ + (Cx+d))$, where Q is an $n \times m$ matrix, A, C are $m \times n$ matrices, $b, d \in \mathbb{R}^m$, and y_+ denotes a vector whose ith component is $\max\{y_i, 0\}$. Then we can identify the monotonicity of a piecewise linear mapping by its gradient.

LEMMA 2.1. *Suppose that φ is a piecewise linear mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$ and the gradient φ' (if it exists) is a positive semidefinite (or positive definite) matrix. Then φ is a monotone (or uniformly monotone) mapping.*

Proof. Since φ is piecewise linear, there are finitely many polyhedral sets D_i such that $\mathbb{R}^n = \bigcup_{i=1}^r D_i$ and φ is an affine mapping on each D_i (i.e., there exist a matrix Q_i and a vector q_i such that $\varphi(x) = Q_i x + q_i$ for $x \in D_i$). For any $x, y \in \mathbb{R}^n$ there exist $0 = \theta_0 < \theta_1 < \cdots < \theta_s = 1$ and indices $\{\alpha_j\}_{j=1}^s \subset \{i\}_{i=1}^r$ such that $w^{j-1}, w^j \in D_{\alpha_j}$ for $j = 1, \ldots, s$, where $w^j := y + \theta_j(x-y)$. Let $\varphi'(w^j)$ be the gradient of φ on D_{α_j}. Then

$$\begin{aligned}(x-y)^T(\varphi(x)-\varphi(y)) &= \sum_{j=1}^s (x-y)^T(\varphi(w^j)-\varphi(w^{j-1}))\\ &= \sum_{j=1}^s (x-y)^T\varphi'(w^j)(w^j - w^{j-1})\\ &= \sum_{j=1}^s (\theta_j - \theta_{j-1})(x-y)^T\varphi'(w^j)(x-y).\end{aligned} \tag{2.6}$$

Obviously, if φ' is positive semidefinite (or positive definite), then φ is monotone (or uniformly monotone). □

Remark. The proof is included for easy reference. We assume the result is well known. Fujisawa and Kuh [11] and Rheinboldt and Vandergraft [37, p. 685] also had the same proof for the positive definite case. Note that if φ is monotone (or uniformly monotone), then the gradient of φ is positive semidefinite (or positive definite) [32, p. 72].

LEMMA 2.2 [34] . *Let Φ be a differentiable function defined on $\mathbb{R}^n$. Then the gradient of Φ is a monotone mapping if and only if Φ is a convex function. If the gradient of Φ is a uniformly monotone mapping, then Φ is a strictly convex function.*

LEMMA 2.3. *Suppose that* $0 < \alpha < 1/\|M\|$. *Then the function f defined by* (2.4) *is a convex function. If M is symmetric positive definite, then f is a strictly convex function.*

Proof. Let $\varphi(w) := (I-\alpha M)(w - (w - \alpha(Mw+q))_l^u)$. Then it is not difficult to verify that, if $\varphi'(w)$ exists,

$$\varphi'(w) = (I-\alpha M) - (I-\alpha M)\sigma(I-\alpha M),$$

where σ is a diagonal matrix $\operatorname{diag}(\sigma_{11}, \sigma_{22}, \ldots, \sigma_{nn})$ and $\sigma_{ii} = 0$ or 1. If M is positive semidefinite, then $I - \alpha M$ is positive semidefinite with eigenvalues between 0 and 1. (Note that $\|M\|$ is the 2-norm of the matrix M, which is the largest eigenvalue of M.) Therefore, $(I-\alpha M) - (I-\alpha M)(I-\alpha M)$ is also positive semidefinite. We have

$$\begin{aligned}x^T\varphi'(w)x &= x^T(I-\alpha M)x - x^T(I-\alpha M)\sigma(I-\alpha M)x\\ &\geq x^T(I-\alpha M)x - x^T(I-\alpha M)(I-\alpha M)x \geq 0.\end{aligned}$$

Thus φ' is positive semidefinite. It follows from Lemmas 2.1 and 2.2 that f is a convex function. The proof for the positive definite case is the same. $\square$

It follows from Lemmas 2.1, 2.2, and 2.3, and from equation (2.5) that w is a minimizer of $f(w)$ if and only if w is a solution of (2.3). This proves the following theorem.

THEOREM 2.4. *Suppose that* $0 < \alpha < 1/\|m\|$. *Then* w *is a solution of* (2.3) *if and only if* w *is a solution of the unconstrained problem* $\min_{w \in \mathbb{R}^n} f(w)$.

Consider the classical symmetric positive semidefinite linear complementarity problem [22], [30]

$$Mx + q \geq 0, \quad x \geq 0, \quad x^T(Mx + q) = 0, \tag{2.7}$$

which is a special case of (2.3) with $l_i = 0$ and $u_i = +\infty$ for $i = 1, \dots, n$. Thus as an immediate consequence of Theorem 2.4 we have the following corollary.

COROLLARY 2.5. *Suppose that* $0 < \alpha < 1/\|M\|$. *Then* y *is a solution* (2.7) *if and only if* y *is a solution of the unconstrained minimization problem*

$$\min_{w \in \mathbb{R}^n} \tfrac{1}{2} w^T(I - \alpha M)w - \tfrac{1}{2}\|w - \alpha(Mw + q))_+\|^2, \tag{2.8}$$

where $(x_+)_i := \max\{0, x_i\}$.

Now consider the least-distance problem [7], [21], [28], [47], [48]

$$\min_{Ax \geq b} \tfrac{1}{2}\|x - c\|^2, \tag{2.9}$$

where A is an $m \times n$ matrix, $b \in \mathbb{R}^m$, and $x, c \in \mathbb{R}^n$. The Karush–Kuhn–Tucker conditions for (2.9) are the following [30]:

$$\begin{aligned} x = c + A^T w, \qquad & w \geq 0, \\ Ax \geq b, \qquad & w^T(Ax - b) = 0. \end{aligned} \tag{2.10}$$

It is not difficult to verify that x, w satisfy (2.10) if and only if $x = c + A^T w$ and w solves the following linear complementarity problem:

$$(AA^T)w + (Ac - b) \geq 0, \quad w \geq 0, \quad w^T((AA^T)w + (Ac - b)) = 0. \tag{2.11}$$

Therefore, we have the following corollary of Corollary 2.5.

COROLLARY 2.6. *Suppose that* $0 < \alpha < 1/\|A\|^2$. *Then* x *is a solution of* (2.9) *if and only if* $x = c + A^T w$, *where* w *is the solution of the unconstrained minimization problem*

$$\min_{w \in \mathbb{R}^m} \tfrac{1}{2} w^T(I - \alpha AA)w - \tfrac{1}{2}\|(w - \alpha(AA^T w + Ac - b))_+\|^2. \tag{2.12}$$

Remark. Li, Pardalos, and Han [21] give a different but similar reformulation of (2.9) as an unconstrained minimization problem with a convex quadratic spline as the objective function, when the constraints are $Ax = b$, $x \geq 0$. A very simple linear Gauss–Seidel algorithm with linear convergence rate is proposed and tested there.

3. Newton method for piecewise linear equations. Consider a piecewise linear mapping $\varphi : \mathbb{R}^n \to \mathbb{R}^n$; i.e., $\varphi(x) = Q^T((Ax + b)_+ + (Cx + d))$, where Q, A, C are $m \times n$ matrices and b, $d \in \mathbb{R}^m$. Suppose that the gradient φ' of φ is nonsingular (if it exists) and that there exists a nonsingular matrix B such that $B \cdot \varphi(x)$ is the gradient of a strictly convex function f on $\mathbb{R}^n$. Since φ is piecewise linear, there are finitely many polyhedral sets $\{D_i\}_{i=1}^r$ such that the interiors (intD_i) of D_i are mutually disjoint, φ is an affine mapping on D_i (i.e., $\varphi'(x)$ is a constant matrix for $x \in \text{int } D_i$), and $\bigcup_{i=1}^r D_i = \mathbb{R}^n$. We use $\varphi'(x)$ to denote one of the $\varphi'(x)|_{D_i}$ if x is in more that one D_i. The following algorithm shows how to apply the Newton method to solve the system of piecewise linear equations $\varphi(x) = 0$.

ALGORITHM 3.1
Newton method with line minimization

Given an initial point $x^0 \in \mathbb{R}^n$, generate a sequence x^{i+1}, $i = 0, 1, \ldots$, by the following iterative scheme:

```
Step 1. If $\varphi(x^i) = 0$, then stop.
Step 2. Compute $p^i = (\varphi'(x^i))^{-1} \cdot \varphi(x^i)$.
Step 3. Find $t_i$ such that $(p^i)^T \cdot B \cdot \varphi(x^i - t_i p^i) = 0$.
Step 4. Set $x^{i+1} = x^i = t_i p^i$.
```

We can solve three classes of problems by the Newton method with exact line search:

(1) the strictly convex quadratic programming problem with bounded constraints, i.e., (2.1) with M symmetric positive definite;

(2) the linear complementarity problem associated with a symmetric positive definite matrix M, i.e., (2.7) with a symmetric positive definite matrix M;

(3) the least-distance problem associated with a polyhedral set generated by linearly independent inequalities, i.e., (2.9) with a matrix A with full row rank.

Note that $(\varphi'(x^i))^{-1} \cdot \varphi(x^i) = (f''(x^i))^{-1} \cdot f'(x^i)$ and $f'(x^i - tp^i) = -(p^i)^T \cdot B \cdot \varphi(x^i - tp^i)$. Thus the iterative scheme is almost the same as the standard Newton method with line minimization for a strictly convex function with nonsingular Hessian matrix [10]. The difference is that $f'(x) = B \cdot \varphi(x)$ is a piecewise linear mapping that has only Gâteaux derivatives but no Fréchet derivatives at some points.

The behavior of the above algorithm is very similar to that of the standard Newton method for solving the unconstrained minimization of a strictly convex quadratic function. Instead of finding the solution in one iteration, the above algorithm terminates (theoretically) in a finite number of steps. The key idea is the following. If x^* is the unique solution of $\phi(x) = 0$ and if x^i is sufficiently close to x^*, then both x^* and x^i are in the same polyhedral set where φ is an affine mapping. Then the above algorithm is actually the standard Newton method, and one more iteration of the algorithm produces x^*.

THEOREM 3.2. *The Newton method with line minimization terminates in a finite number of iterations in exact arithmetic.*

Proof. Since $f(x^i) \le f(x^0)$ for $i \ge 0$ and f is strictly convex, $\{x^i\}$ is a bounded sequence. If the algorithm does not terminate in a finite number of iterations, let $\{x^{i_s}\}$ be a subsequence of $\{x^i\}$ that converges to a point x^*. Since there are only finitely many polyhedral sets $\{D_j\}_1^r$, at least one D_j contains infinitely many x^{i_s} such that $\varphi'(x^{i_s})$ is the gradient of $\varphi(x)$ on D_j. So we may assume $\{x^{i_s}\} \subset D_j$ such that $\varphi'(x^{i_s})$ is $\varphi_j := \varphi'(x)|_{D_j}$. Let $p^* := (\varphi_j)^{-1}\varphi(x^*)$. Then

$$
\begin{aligned}
\frac{d}{dt} f(x^* - tp^*)\Big|_{t=0} &= -\{f'(x^* - tp^*)\}^T p^* \big|_{t=0} \\
&= -(f'(x^*))^T p^* \\
&= -(f'(x^*))^T ((B\varphi_j))^{-1} (f'(x^*)).
\end{aligned}
\tag{3.1}
$$

Since $B\varphi_j$ is the Hessian of f on D_j and f is a strictly convex quadratic function on D_i, $(B\varphi_j)^{-1}$ is symmetric positive definite, as well as $B\varphi_j$. Thus there exists a scalar $\gamma > 0$ such that

$$x^T(B\varphi_j)^{-1}x \geq \gamma \cdot \|x\|^2 \quad \text{for } x \in \mathbb{R}^n. \tag{3.2}$$

If $f'(x^*) \neq 0$, then it follows from (3.1) and (3.2) that

$$\frac{d}{dt}f(x^* - tp^*)\bigg|_{t=0} \leq -\gamma \cdot \|f'(x^*)\|^2 < 0.$$

By the continuity of $f'(x)$ and $(\varphi_j)^{-1}\varphi(x)$ there exist $\epsilon > 0$ and $s_0 > 0$ such that

$$\frac{d}{dt}f(x^{i_s} - tp^{i_s}) \leq -\frac{\gamma}{2} \cdot \|f'(x^*)\|^2 < 0 \quad \text{for } s \geq s_0, \quad 0 \leq t \leq \epsilon.$$

By the mean value theorem there exist $0 < t_s < \epsilon$ such that

$$f(x^{i_s} - \epsilon p^{i_s}) - f(x^{i_s}) = \epsilon \cdot \frac{d}{dt}f(x^{i_s} - tp^{i_s})\bigg|_{t=t_s} \leq -\frac{\gamma\epsilon}{2} \cdot \|f'(x^*)\|^2 < 0 \quad \text{for } s \geq s_0,$$

which implies

$$f(x^{i_s+1}) - f(x^{i_s}) \leq -\frac{\gamma\epsilon}{2} \cdot \|f'(x^*)\|^2 < 0 \quad \text{for } s \geq s_0. \tag{3.3}$$

Since $f(x^i)$ is a monotonically decreasing sequence, $\lim_{i\to\infty} f(x^i) = \lim_{s\to\infty} f(x^{i_s}) = f(x^*)$, which contradicts (3.3). The contradiction proves $f'(x^*) = 0$ (i.e., $\varphi(x^*) = 0$).

Now let f_j be the strictly convex quadratic function such that $f_j(x) = f(x)$ for $x \in D_j$. Then $f_j'(x^*) = 0$; i.e.,

$$f_j(x^*) = \min_{x\in\mathbb{R}^n} f_j(x).$$

It is well known that the Newton method (with or without line minimization) produces the solution in one iteration for a strictly convex quadratic function. For $x^{i_s} \in D_j$ we have

$$x^* = x^{i_s} - (f_j''(x^{i_s}))^{-1}f_j'(x^{i_s}) = x^{i_s} - (B\varphi_j)^{-1}\varphi(x^{i_s}) = x^{i_s} - p^{i_s}.$$

Therefore,

$$f(x^{i_s+1}) \leq f(x^{i_s} - p^{i_s}) = f(x^*) = \min_{x\in\mathbb{R}^n} f(x),$$

which implies $x^i = x^*$ and $\varphi(x^i) = 0$ for $i > i_1$. So the algorithm should stop when $i = i_1 + 1$. This proves the finite termination of the Newton method with line minimization. □

Remark. The convergence of the above algorithm actually follows from more general results in [34, Chap. 14]. In particular, the reader is referred to [34, Problem 1, p. 507]. For convenience, we include the complete proof here.

Even though we cannot use the Newton Method to solve the unconstrained minimization of convex functions with singular Hessians, there are many so-called quasi-Newton methods that can handle singular Hessians [10].

Systems of piecewise linear equations have been the object of extensive research (see [8] and the references therein). In particular, Katzenelson's algorithm [16] and its generalizations are the favored methods for solving piecewise linear equations associated with resistor networks in electrical engineering (see [32] and the references therein), whereas the fixed point and complementarity pivoting algorithms are widely used methods for solving piecewise linear equations associated with optimization problems (see [9] and the references therein).

4. Condition numbers of the *k*th divided difference matrices. In this section we give estimates of the condition numbers [12] of the kth divided difference matrix and show that the order of $\|\nabla_k\| \cdot \|\nabla_k^+\|$ increases to at least n^k as $n \to \infty$. Here A^+ denotes the pseudoinverse of A [12], [31]. This makes the k-convex approximation problem a very difficult computational problem even if n is moderate. We will discuss the computational aspect of the k-convex in §5. The ill conditioning of ∇_k is crucial to understanding some numerical phenomena in §§7 and 8.

First note that $\|\nabla_k\|^2 = \|\nabla_k^T\|^2 = \|\nabla_k \nabla_k^T\|$ and $\nabla_k^+ = \nabla_k^T(\nabla_k\nabla_k^T)^{-1}$. Therefore,

$$\|\nabla_k^+\|^2 = \|(\nabla_k\nabla_k^T)^{-1}\|.$$

Thus the condition number of ∇_k is the square root of the condition number of $\Delta_k := \nabla_k\nabla_k^T$. It is not difficult to see that Δ_k is $(2k+1)$-banded symmetric positive definite Toeplitz matrix. We use band$(\mu_1, \mu_2, \ldots, \mu_{2k+1})$ to denote such an $n \times n$ matrix. It is not difficult to verify that

$$\Delta_k = \text{band}(\mu_0^k, \mu_1^k, \ldots, \mu_{2k}^k),$$

where

$$\mu_i^k = (-1)^{k-1}\binom{2k}{i}, \qquad i = 0, 1, \ldots, 2k. \tag{4.1}$$

It is well known that Δ_1 = band(−1,2,−1) is the so-called stiffness matrix derived from a difference scheme for the second-order ordinary differential equation with boundary conditions:

$$y'' = g(t), \qquad a \le t \le b, \quad y(a) = \alpha, \quad y(b) = \beta,$$

where g is a function defined on the interval $[a, b]$ and α, β are given real numbers. The eigenvalues of Δ_1 are $\lambda_i := 4 \cdot \sin^2(i\pi/2(n+1))$, $i = 1, \ldots, n$, and the corresponding eigenvectors are [33]

$$v^i := \left(\sin\frac{i\pi}{n+1}, \sin\frac{2i\pi}{n+1}, \ldots, \sin\frac{ni\pi}{n+1}\right)^T \quad \text{for } i = 1, \ldots, n.$$

LEMMA 4.1. *$\Delta_k x = \Delta_1^k x$ if $x_i = 0$ for $1 \le i \le 2k$ and $n - 2k + 1 \le i \le n$.*

Proof. Obviously, if $x_i = 0$ for $1 \le i \le 2k$, $n - 2k + 1 \le i \le n$, then

$$(\nabla_k^T x)_i = \sum_{j=0}^{k}\binom{k}{j}(-1)^j x_{i-k+j} \quad \text{for } k+1 \le i \le n-k,$$

$$(\nabla_k^T x)_i = 0 \quad \text{for } 1 \le i \le 2k,\ n - 2k + 1 \le i \le n.$$

Thus

$$(\nabla_k \nabla_k^T x)_i = 0 \quad \text{for } 1 \le i \le k,\ n-k+1 \le i \le n,$$

and for $k+1 \le i \le n-k$

$$\begin{aligned}(\nabla_k \nabla_k^T x)_i &= \sum_{s=0}^{k} \binom{k}{s} (-1)^{k-s} (\nabla_k^T x)_{i+s} \\ &= \sum_{s=0}^{k} \binom{k}{s} (-1)^{k-s} \sum_{j=0}^{k} \binom{k}{j} (-1)^j x_{i+s-k+j} \\ &= \sum_{s=0}^{k} \sum_{j=0}^{k} \binom{k}{s} \binom{k}{j} (-1)^{k-s-j} x_{i+s-k+j} \\ &= \sum_{j=0}^{2k} \binom{2k}{j} (-1)^{k-j} x_{i-k+j}.\end{aligned}$$

Also, we have

$$\begin{aligned}(\Delta_1^k x)_i &= \sum_{j=0}^{2k} \binom{2k}{j} (-1)^{k-j} x_{i-k+j} \quad \text{for } k+1 \le i \le n-k, \\ (\Delta_1^k x)_i &= 0 \quad \text{for } 1 \le i \le k,\ n-k \le i \le n.\end{aligned}$$

In fact, we can use induction to prove that

$$\begin{aligned}(\Delta_1^s x)_i &= \sum_{j=0}^{2s} \binom{2s}{j} (-1)^{s-j} x_{i-s+j} \quad \text{for } s+1 \le i \le n-s, \\ (\Delta_1^s x)_i &= 0 \quad \text{for } 1 \le i \le 2k-s,\ n-2k+s+1 \le i \le n.\end{aligned}$$

This completes the proof of Lemma 4.1. □

LEMMA 4.2. *For $n \ge 4k+1$ and $k = 1, 2, \ldots,$*

$$4^k \sin^{2k} \frac{(n-4k)\pi}{2(n+1)} \le \|\Delta_k\| \le 4^k.$$

Proof. Since $\|\Delta_k\|$ is symmetric and positive definite, it is well known that

$$\|\Delta_k\| = \lambda_{\max} \le \|\Delta_k\|_\infty,$$

where $\lambda_{\max}$ is the largest eigenvalue of Δ_k and $\|\Delta_k\|_\infty$ denotes the l_∞-norm of Δ_k [12]. From (4.1) we know that

$$\|\Delta_k\|_\infty = \sum_{i=0}^{2k} \binom{2k}{i} = 2^{2k} = 4^k.$$

Therefore, $\|\Delta_k\| \le 4^k$.

On the other hand, let $G :=\ \operatorname{span}\{v^n, v^{n-1}, \ldots, v^{n-4k}\}$, where v^i is the ith eigenvector of Δ_1 corresponding to λ_i. Then $\dim G = 4k + 1$ and

$$\|\Delta_1^k v\| \geq \left(4\sin^2 \frac{(n-4k)\pi}{2(n+1)}\right)^k \|v\| \quad \text{for } v \in G.$$

It follows from Lemma 4.1 that the range of $(\Delta_k - \Delta_i^k)$ is at most dimension $4k$. Therefore, there is a vector $v \in G$, $v \neq 0$, such that $\Delta_k v = \Delta_1^k v$. Hence

$$\|\Delta_k v\| = \|\Delta_1^k v\| \geq \left(4\sin^2 \frac{(n-4k)\pi}{2(n+1)}\right)^k \|v\|,$$

i.e.,

$$\|\Delta_k\| \geq \left(4\sin^2 \frac{(n-4k)\pi}{2(n+1)}\right)^k. \qquad \square$$

Remark. If we take $G :=\ \operatorname{span}\{v^1, v^2, \ldots, v^{4k+1}\}$, then $\dim G = 4k + 1$ and

$$\|\Delta_1^k v\| \leq \left(4\sin^2 \frac{(4k+1)\pi}{2(n+1)}\right)^k \|v\| \quad \text{for } v \in G.$$

Also, there is $v \in G$, $v \neq 0$, such that $\Delta_k v = \Delta_1^k v$. Hence

$$\|\Delta_k v\| = \|\Delta_1^k v\| \leq \left(4\sin^2 \frac{(4k+1)\pi}{2(n+1)}\right)^k \|v\|,$$

which implies

$$\|\Delta_k^{-1}\| \geq \left(4\sin^2 \frac{(4k+1)\pi}{2(n+1)}\right)^{-k}.$$

This proves the following lemma.

LEMMA 4.3. *For* $n \geq 4k + 1$ *and* $k = 1, 2, \ldots,$

$$\|\Delta_k^{-1}\| \geq 4^{-k} \sin^{-2k} \frac{(4k+1)\pi}{2(n+1)} \geq \left(\frac{n+1}{(4k+1)\pi}\right)^{2k}.$$

As a consequence of Lemmas 4.2 and 4.3 we have the following estimates of $\|\nabla_k\|$ and $\|\nabla_k^+\|$.

COROLLARY 4.4. *For* $n \geq 4k + 1$ *and* $k = 1, 2, \ldots,$

$$\|\nabla_k^+\| \geq \left(\frac{n+1}{(4k+1)\pi}\right)^k$$

and

$$2^k \left(\frac{n-4k}{n+1}\right)^k \leq \|\nabla_k\| \leq 2^k.$$

It is not difficult to verify that

$$\Delta_2 = \Delta_1^2 + \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{pmatrix}.$$

Thus

$$x^T\Delta_2 x = x^T\Delta_1^2 x + x_1^2 + x_2^2 \geq x^T\Delta_1^2 x \geq 16\sin^4\frac{\pi}{2(n+1)}\|x\|^2;$$

i.e., the minimum eigenvalue $\lambda_{\min}$ of Δ_2 is at least $16\sin^4(\pi/2(n+1)) \geq 16(n+1)^{-4}$. Since $\|\Delta_2^{-1}\| = 1/\lambda_{\min}$ [12], we have the following corollary.

COROLLARY 4.5. *For* $n \geq 9, ((n+1)/9\pi)^2 \leq \|\nabla_2^+\| \leq (n+1)^2/4$.

Remark. It follows from Lemma 4.2 and Corollary 4.4 that

$$\lim_{n\to\infty}\frac{\|\Delta_k\|}{4^k} = 1$$

and

$$\lim_{n\to\infty}\frac{\|\nabla_k\|}{2^k} = 1.$$

Unfortunately, we do not have an upper bound for $\|\nabla_k^+\|$ with $k \geq 3$. Such an upper bound is very important for obtaining a stable estimate (see §8) of the accuracy of approximate solutions generated by the Newton method for solving (1.1) implemented in §6.

5. Stability of the least-distance problem. Consider the constrained least-solved problem

$$\min_{Ax\geq b}\tfrac{1}{2}\|x-c\|^2, \tag{5.1}$$

where A is an $m \times n$ matrix with rank $m, b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. We have the following perturbed version of (5.1):

$$\min_{Ax\geq \hat{b}}\tfrac{1}{2}\|x-\hat{c}\|^2. \tag{5.2}$$

Let x^* and $\hat{x}$ be the solutions of (5.1) and (5.2), respectively. Since a metric projection from a Hilbert space to its closed convex set is nonexpansive [39], we have the following result.

LEMMA 5.1. *If* $b = \hat{b}$, *then* $\|x^* = \hat{x}\| \leq \|c - \hat{c}\|$.

LEMMA 5.2. $A(\hat{x} + A^+(b - \hat{b})) \geq b$ *and*

$$\|x^* - (\hat{x} + A^+(b-\hat{b}))\| \leq \|A^+(b-\hat{b})\| + \|c - \hat{c}\|.$$

Proof. Let $x = y + A^+(\hat{b} - b)$. Since $AA^+ = I$ is the identity matrix, $Ax \geq \hat{b}$ if and only if $Ay \geq b$. Thus $\hat{x}$ is a solution of (5.2) if and only if $\hat{x} = \hat{y} + A^+(\hat{b} - b)$, where $\hat{y}$ is a solution of the following least-distance problem:

$$\min_{Ay \geq b} \tfrac{1}{2}\|y - (\hat{c} + A^+(b - \hat{b}))\|^2. \tag{5.3}$$

So $\hat{y} := \hat{x} + A^+(b - \hat{b})$ is the solution of (5.3). It follows from Lemma 5.1 that

$$\|x^* - \hat{y}\| \leq \|c - (\hat{c} + A^+(b - \hat{b}))\| \leq \|c - \hat{c}\| + \|A^+(b - \hat{b})\|.$$

Obviously, $A\hat{y} \geq b$. This completes the proof of Lemma 5.2 □

Remark. Note that Lemma 5.2 implies that

$$\|x^* - \hat{x}\| \leq 2\|A^+\| \cdot \|b - \hat{b}\| + \|c - \hat{c}\|. \tag{5.4}$$

Let $F(b) := \{x \in \mathbb{R}^n : Ax \geq b\}$. It was proved by Li [19] that $\|A^+\|$ is a Lipschitz constant of F, i.e.,

$$H(F(b), F(\hat{b})) \leq \|A^+\| \cdot \|b - \hat{b}\|,$$

where $H(\cdot, \cdot)$ denotes the Hausdorff metric defined as

$$H(X, Y) := \max\left\{\sup_{x \in X} \inf_{y \in Y} \|x - y\|, \sup_{y \in Y} \inf_{x \in X} \|x - y\|\right\} \quad \text{for } X, Y \subset \mathbb{R}^n.$$

In general, if A is not of full row rank, then [19]

$$H(F(b), F(\hat{b})) \leq \left(\max_{A_0 \in \mathcal{A}} \|A_0^+\|\right) \cdot \|b - \hat{b}\|,$$

where $\mathcal{A}$ is the collection of all matrices consisting of rank(A) linearly independent of rows of A. We conjecture that one could replace A^+ by $\max_{A_0 \in \mathcal{A}} \|A_0^+\|$ for general A in (5.4).

Lemma 5.1 tells us that (5.1) is a stable problem with respect to perturbations of the data c. However, (5.1) is not stable with respect to perturbations of constraints if $\|A^+\|$ is large. For example, if $A = \nabla_k$, then $\|\nabla_k^+\|$ is at least of order n^k. This makes the well-posed problem (1.1) difficult to handle computationally, as can be seen from the following a priori error estimate of the approximate solution generated by the Newton method outlined in §3.

Consider the system of piecewise linear equations associated with the Lagrange multiplier w with respect to (5.1):

$$w = (w - \alpha(AA^T w + Ac - b))_+. \tag{5.5}$$

Suppose that we use the Newton method proposed in §3 to solve the above equation and that $w \geq 0$ is an approximate solution of (5.5). Then

$$w = (w - \alpha(AA^T w + Ac - b))_+ + \delta, \tag{5.6}$$

where δ is the error vector. Let

$$\hat{\delta}_i := \begin{cases} 0, & \text{if } (w - \alpha(AA^T w + Ac - b))_i \geq 0, \\ w_i, & \text{if } (w - \alpha(AA^T w + Ac - b))_i < 0, \end{cases} \tag{5.7}$$

and $\hat{w} := w - \hat{\delta}$. Then one can verify that

$$\begin{aligned} \hat{w} &= (w - \alpha(AA^Tw + Ac - b))_+ + \delta - \hat{\delta} \\ &= (w - \alpha(AA^Tw + Ac - b) + \delta - \hat{\delta})_+ \\ &= (\hat{w} - \alpha(AA^T\hat{w} + A\hat{c} - \hat{b}))_+, \end{aligned} \tag{5.8}$$

where $\hat{c} = c + A^T\hat{\delta}$ and $\hat{b} = b + (1/\alpha)\delta$. Therefore, (5.8) is the system of piecewise linear equations associated with the Lagrange multiplier w with respect to (5.2). Let

$$\hat{x} := A^Tw + c = A^T\hat{w} + \hat{c}. \tag{5.9}$$

Then $\hat{x}$ is the solution of (5.2). By Lemma 5.2 we have the following a priori error estimate for $\hat{x}$.

THEOREM 5.3. *Let w, δ, and $\hat{x}$ be given by* (5.6) *and* (5.9). *Then*

$$\|x^* - \hat{x}\| \le \|A^T\| \cdot \|\delta\| + \frac{2}{\alpha}\|A^+\delta\|,$$

where x^* is the solution of (5.1).

Remark. Errors caused by approximate solutions of (5.5) actually have the same effect on the solution as perturbations of c and b. Thus if A is ill conditioned, we have to be very careful in claiming how good the approximate solution is.

Consider the k-convex approximation problem now. Suppose that

$$w = (w - \alpha(\Delta_k w + \nabla_k c))_+ + \delta, \quad w \ge 0, \tag{5.10}$$

and that $\hat{x} := \nabla_k^T w + c$. Then we have the following a priori error estimate.

COROLLARY 5.4. *Let x^* be the solution of* (1.1). *Then*

$$\|x^* - \hat{x}\| \le \frac{2}{\alpha} \cdot (\delta^T \Delta_k^{-1} \delta)^{1/2} + \|\nabla_k\| \cdot \|\delta\|.$$

6. Implementation of the Newton method for *k*-convex approximation. We use the Newton method with exact line search to solve the following system of piecewise linear equations associated with the Lagrange multiplier w of the solution of (1.1):

$$\varphi(w) := w - (w - \alpha(\Delta_k w + \nabla_k c))_+ = 0. \tag{6.1}$$

We recover the solution of (1.1) by the formula $x^* = \nabla_k^T w + c$. There are many ways to implement the four steps of the Newton method with exact line search outlined in §3. Here is our implementation.

First, we can afford to do an exact line search because of the simple structure of the function $g(t) := -(p^i)^T(I - \alpha\Delta_k)\varphi(x^i - tp^i)$. The function $g(t)$ is a monotone linear spline function with n nodes!

Suppose $g(t) = \gamma + \beta \cdot t + v^T(d - tv)_+$ is a monotone nondecreasing linear spline function with $d, v \in \mathbb{R}^n$ and $\gamma, \beta \in \mathbb{R}$. Then we have the following simple algorithm to solve $g(t) = 0$. (Here we implicitly assume a certain data structure on d, c that facilitates the deletion of components of d, v).

ALGORITHM 6.1

Algorithm for finding a zero of monotone linear spline functions

```
Step 1. Predetermine a set $T := \{t_1, \dots, t_m\}$ of estimates of the zero(s) of $g(t)$, or simply
        let $m = 0$. Set $l = 1$.
Step 2. Delete all zero components of $v$ and the corresponding components of $d$.
Step 3. If $l \le m$, then $t := t_l$; otherwise, $t := d_j/v_j$ (one of the undeleted nodes).
Step 4. Consider four cases:
            case (1) $g(t) > 0, v_i > 0$, and $d_i/v_i \ge t$;
            case (2) $g(t) < 0, v_i < 0$, and $d_i/v_i \le t$;
            case (3) $g(t) > 0, v_i < 0$, and $d_i/v_i \ge t$;
            case (4) $g(t) < 0, v_i > 0$, and $d_i/v_i \le t$;
            case (5) $g(t) = 0$.
        Do the following for all remaining components of $d, v$: if case (1) or case (2)
        holds, then $\gamma := \gamma + v_i d_i$, $\beta := \beta - v_i^2$, and delete $d_i, v_i$ from $d, v$, respectively;
        if case (3) or case (4) holds, then delete $d_i, v_i$ from $d, v$, respectively; if case (5)
        holds, then $t$ is a zero of $g$ and stop.
Step 5. If $v$(or $d$) has some undeleted components, set $l := l + 1$ and return to Step 3.
Step 6. If $\beta \ne 0$, then $t := -\gamma/\beta$ is a zero of $g$; otherwise, $g(t)$ has no zero.
```

Remark. If we choose t to be the median of $\{d_i/v_i\}$, then at least half of the components of d, v will be deleted after Step 4. Since there is a linear time algorithm for finding the median, the above algorithm could be implemented as a linear time algorithm. For implementation details see [36]. In general, if one knows that $g(t)$ has a zero between a and b, then, by setting $T = \{a, b\}$, all nodes $\{d_i/v_i\}$ lying outside the interval (a, b) will be eliminated after the first two iterations.

THEOREM 6.2. *If $g(t) = \gamma + \beta \cdot t + v^T(d - tv)_+$ is a monotone linear spline function with $d, v \in \mathbb{R}^n$, then $g(t) = 0$ can be solved by $\mathcal{O}(n)$ flops.*

ALGORITHM 6.3

Algorithm for k-convex approximations

```
Step 1. $B = I - \alpha\Delta_k, b = -\alpha\Delta_k c$, and $w = 0$.
Step 2. Compute the residual $e := \varphi(w)$ of (6.1).
Step 3. If $\|e\|$ is less than a given error tolerance $\epsilon > 0$, then compute $z = \varphi(w_+)$ and
        a priori error estimate $\mu = (2/\alpha)(z^T\Delta_k^{-1}z)^{1/2} + \|\nabla_k\| \cdot \|z\|$; if $\mu$ is less than a
        given error tolerance $\rho > 0$ or if the number of iterations is larger than a given
        limit, then output $x = c + \nabla_k^T w_+$ and stop.
Step 4. Compute the Jacobian $\varphi'(w)$ as follows: if the $j$th component of $w - \varphi(w)$ is 0,
        the $j$th row of $\varphi'(w)$ is the $j$th row of the identity matrix; otherwise, the $j$th row
        of $\varphi'(w)$ is the $j$th row of $\alpha\Delta_k$.
Step 5. Compute the descent direction $p := (\varphi'(w))^{-1}e$ by band Gauss elimination
        without pivoting.
Step 6. Express the function $g(t) := -p^T B\varphi(w - tp)$ as

            $$g(t) = \gamma + \beta \cdot t + v^T(d - tv)_+,$$

        where $\gamma, \beta, d$, and $v$ can be computed as follows:

            $$v := B_p, \quad \gamma := -v^T w, \quad \beta := v^T p, \quad d = Bw + b.$$
```

Step 7. *Find the solution t of the equation* $g(t) = 0$ *by Algorithm* 6.1 *with* $T = \{0, 1, 2\}$.
Step 8. *If* $t \leq 0$ *or* $t \geq 2$, *replace t by* 1.
Step 9. *Set* $w := w - tp$, *and return to Step* 2.

We regard the line minimization procedure as a means of finding the locally optimal overrelaxation parameter. That is the reason that we modify the step size when it is not in the open interval (0,2). Note that Step 8 is very important since the ill conditioning of Δ_k may cause some computational problems here. For example, in exact arithmetic p should be a descent direction, but in practice we are not sure whether this is the case since implemented algorithm sometimes produces a step size $t \leq 0$. There are two explanations for this numerical phenomenon: (1) the inaccurate solution of p might be an ascent direction; (2) the effect of ill conditioning of Δ_k on g is that the graph of g might be very flat, which would cause the solution of the equation $g(t) = 0$ to be highly unstable. A step size $t \leq 0$ indicates the failure of the exact line search. Therefore, a modification is necessary. On the other hand, in our numerical experiments it seems that the correct step size should be less than two, but we cannot say that $t \geq 2$ is also an indication of the failure of the exact line search. Nevertheless, our modification should not hurt the performance of the algorithm since the next iterate moves along p at least one unit, which we believe is sufficient. Since the zero of $g(t)$ is positive, we include zero in T. The other two estimates $t = 1, 2$ are heuristic. Our choice of the set T dramatically reduces the number of iterations in Algorithm 6.1 for our numerical experiments.

To justify the stability of Gauss elimination without pivoting to compute the solution $p = (\varphi'(w))^{-1}e$ in Step 5 of Algorithm 6.3, we need the following standard notation for submatrices. For any two index sets J and K, $(\varphi'(w))_{J,K}$ denotes the matrix obtained by deleting the rows and columns of $\varphi'(w)$ whose indices are not in J and K, respectively. Let I be the set of indices j such that the jth row of $\varphi'(w)$ is the jth row of the identity matrix, and let $J := \{j : 1 \leq j \leq n, j \notin I\}$ be the complement of I. Then $\varphi'(w)p = e$ is equivalent to the following system:

$$p_I = e_I \quad \text{and} \quad (\varphi'(w))_{J,J}p_J = e_J - (\varphi'(w))_{J,I}e_I.$$

Such $\alpha\Delta_k$ is positive definite and $(\varphi'(w))_{J,J}$ is a principal submatrix of $\alpha\Delta_k$, $(\varphi'(w))_{J,J}$ is positive definite. Therefore, Gauss elimination without pivoting is stable.

7. Evaluation of Newton method for *k*-convex approximation problems. We have performed numerical tests for various k and n : (1) $1 \leq k \leq 6, n = 50, 100$; (2) $1 \leq k \leq 3$, $n = 200, 400$; (3) $k = 1, 2, n = 1000, 2000$. The vector c is generated according to k and n by one of the following elementary functions:

$$\begin{array}{cccccc} \sqrt{t}, & t^2, & \exp(t), & \sin(\pi t), & \sin(2\pi t), & \sin(4\pi t), \\ t, & t^3, & \exp(-t), & \cos(\pi t), & \cos(2\pi t), & \cos(4\pi t), \end{array} \tag{7.1}$$

perturbed by a random vector of magnitude 0.1. First we define the random generator as follows:

$$\text{random} := 0.1 \cdot (-1)^{\mathbf{irand}(0)} \cdot \mathbf{drand}(0), \tag{7.2}$$

where **irand**, which generates a positive integer, and **drand**, which generates a real number between 0 and 1, are the standard random number generators for FORTRAN 77.

The expression in (7.2) randomly generates a real number between -0.1 and 0.1. For given k and n let $t_i := (i-1)/(n+k-1), i = 1, \ldots, n+k$. (Note that $t_1 = 0$ and $t_{n+1} = 1$.) Then we generate the data c as follows:

$$c_i = g(t_i) + \text{ random} \quad \text{for } i = 1, \ldots, n+k, \tag{7.3}$$

where $g(t)$ is one of the functions listed in (7.1).

Suppose that $\hat{w} \geq 0$ is the approximate solution generated by our algorithm for k-convex approximation and that $\hat{x} := \nabla_k^T \hat{w} + c$. Then

$$\|x^* - \hat{x}\| \leq \frac{2}{\alpha} \cdot (\delta^T \Delta_k^{-1} \delta)^{1/2} + \|\nabla_k\| \cdot \|\delta\|,$$

where

$$\delta := \hat{w} - (\hat{w} - \alpha(\Delta_k \hat{w} + \nabla_k c))_+.$$

The parameters used in the tables in the appendix have the following meaning:

$g(t)$ is the function used in (7.3);
$N :=$ the total number of Newton iterations involved;
err $:= 2/\alpha \cdot (\delta^T \Delta_k^{-1} \delta)^{1/2} + \|\nabla_k\| \cdot \|\delta\|$;
$\mathbf{err}_0 := \|\hat{w} - (\hat{w} - \alpha(\Delta_k \hat{w} + \nabla_k c))_+\|$;
$N_0 :=$ the average number of iterations involved in exact line search;
CPU denotes the CPU time (in seconds) used by the algorithm to find $\hat{x}$;
$\mathbf{err}_1 :=$ the maximum negative components of $\nabla_k \hat{x}$;
$\mathbf{err}_2 := \max\{|\hat{x}_i - g(t_i)| : 1 \leq i \leq n+k\}$;
M denotes the number of failures of exact line search;
M_0 denotes the first index of indicators with failed exact line search.

Note that the key parameters are N, N_0, $\mathbf{err}_0$, and **err**, which are indicators of the efficiency and accuracy of the algorithm. From the eight tables included in the Appendix we can see that if $n^k \leq 10^9$, our algorithm is very efficient and produces a very accurate solution. But if n^k is too large, then the algorithm deteriorates. One can see this in Table 4 for $n = 100$, $k = 6$. Also notice that our algorithm is superb when $k = 1$ (see Table 7). This suggests an alternative way of computing solutions of monotone regression problems when n is large.

It is very important to pay attention to M and M_0. Usually, false step size occurs only when $\mathbf{err}_0$ is very small; i.e., when $\hat{w}$ is very close to the solution. However, we might have a poor a priori estimate due to the ill conditioning of Δ_k^{-1} while $\hat{w}$ is a good approximate solution. So false step size occurs frequently in the following iterations (see Table 4 in the Appendix). The first occurrence of false step size is an indication of a good approximate solution $\hat{w}$. When M is large, N and CPU are misleading since we might well find a good approximate solution with smaller N and CPU if we do not insist on a small a priori error estimate. Also, if $\|\Delta_k^{-1}\|$ is too large, then the a priori error estimate is not reliable.

The error tolerance $\epsilon = 10^{-9}$, and $\rho = 10^{-3}$. The parameter $\alpha = 4^{-k}$ (see Lemma 4.2). Only about one third of our numerical results (for $n = 100, 3 \leq k \leq 6$; $n = 400, 2 \leq k \leq 3$; $n = 2000, 1 \leq k \leq 2$) are included in the Appendix because of limited space. In general, for fixed k and g our numerical tests indicate a consistently better performance of the algorithm for smaller n. The experiments were performed on a SPARC station in double precision.

8. *k*-convex approximation as a data smoothing technique. In this section we give our opinion as to why the k-convex approximation is a good method for data smoothing despite its ill-conditioned nature. First, we give a mathematical explanation of some performance parameters used by Cullinan for evaluation of k-convex approximation as a data smoothing technique [5].

Cullinan used an active set method to compute the Karush–Kuhn–Tucker points x, w satisfying the following equation:

$$b = x - c - \nabla_k^T w. \tag{8.1}$$

Let I be the set of indices of active constraints of $\nabla_k x \geq 0$. Then, of 13 performance parameters, he used the following three quantities:

$$\|b\|, \quad \beta := \max_{i \in I} |(\nabla_k x)_i|, \quad \gamma := \max_{i \notin I} |w_i|. \tag{8.2}$$

Obviously,

$$w = (w - \nabla_k x)_+ + \delta, \tag{8.3}$$

where $\|\delta\|_\infty := \max_{1 \leq i \leq n} |\delta_i| \leq \max\{\beta, \gamma\}$. Substituting (8.1) into (8.3), we have

$$w = (w - (\Delta_k w + \nabla_k(c + b)))_+ + \delta. \tag{8.4}$$

Let x^* be the best k-convex approximation of c, and let $\hat{x}$ be the best k-convex approximation of $c + b$. By our a priori estimate (see Lemma 5.2),

$$\|x - x^*\| \leq \|x - \hat{x}\| + \|\hat{x} - x^*\| \leq 2(\delta^T \Delta_k^{-1} \delta)^{1/2} + \|\nabla_k\| \cdot \|\delta\| + \|b\|.$$

Therefore, if $\|\Delta_k^{-1}\|$ is not too large, small $\|b\|, \beta, \gamma$ imply that the approximate solution is quite accurate. In his numerical reports, for $n = 51$ and $k \leq 6$, $\|b\|, \beta, \gamma$ are at most 10^{-6}, where c is generated by $\exp(x)$ with small perturbations. For $n = 51$ and $k \leq 6$ we contend that $\|\Delta_k^{-1}\|$ might be not very large. Therefore, Cullinan's method produces quite satisfactory approximate solutions in these cases. However, for $n = 51$ and $k = 7$, $8, 9$, $\max\{\|b\|, \beta, \gamma\}$ are about $10^{-4.8}$, $10^{-4.6}$, $10^{-2.7}$, respectively. If $\|\Delta_k^{-1}\|$ is taken into consideration, the approximate solutions are not so satisfactory. For $n = 101$, $\|\Delta_k^{-1}\|$ might be significant and it is difficult to tell how good the approximate solutions are, even though $\max\{\|b\|, \beta, \gamma\}$ are quite small for $k \leq 4$ in Cullinan's reports.

An interesting case is when $k = 2, 3$, $n = 101$, and $c_i = \sin((i - 1)\pi/50)$ with small random perturbations, $i = 1, \ldots, 101$. The approximate solutions seem quite satisfactory since $\max\{\|b\|, \beta, \gamma\} \leq 10^{-10.6}$. More interesting is that for $k = 3$ the approximate solution is very close to the graph of the function $g(t) = \sin(2\pi t)$ for t in the interval [0,1], which is confirmed by our numerical experiments (see Table 1 in the Appendix). Since $c_i \approx g((i-1)/50)$, the approximate solution serves as a smoothing of $g(t)$ contaminated by noisy data.

Because of the ill conditioning of ∇_k, we must exercise caution in drawing conclusions from numerical experiments. There are some traps in the interpretation of numerical results. Here are some observations.

Given any smooth function $g(t)$ for $0 \leq t \leq 1$, let $c_i := g((i-1)/(n-1))$, $i = 1, \ldots, n$. Then

$$(\nabla_k c)_i = \frac{g^{(k)}(\xi_i)}{(n-1)^k} \sim n^{-k}.$$

For $n = 100$ and $k \geq 5, \nabla_k c$ is almost a zero vector. For $w = 0$ the error of (6.1) is very small and the approximate solution $x = c$, but x might be far away from the actual best k-convex approximation.

Now consider $k = 1$, the monotone regression problem. Let w be an eigenvector of Δ_1 corresponding to the smallest eigenvalue:

$$w := \left(\sin \frac{\pi}{n+1}, \sin \frac{2\pi}{n+1}, \ldots, \sin \frac{n\pi}{n+1} \right)^T .$$

Then $w \geq 0$ and $\Delta_1 w = (4 \sin^2(\pi/2(n+1)))w$. Suppose that n is very large. Then $\|\Delta_1 w\|$, $\|\nabla_1 c\|$, and $\|\nabla_1^T w\|$ are very small. Therefore, w is a very good approximate solution of (6.1) for $k = 1$, and the corresponding approximate solution of (1.1) is $c + \nabla_1^T w$, which is a very good approximation of $g(t)$. But, again, x might be far away from the actual best monotone approximation.

The above observations tell us that it is extremely important to have the a priori error estimate of the approximate solution since other information about the accuracy of the approximate solution might be false.

Now suppose that $c = g + d$, where g is a smooth vector and d is a small noisy vector. Cullinan [5] observed from his numerical experiments that, whereas an approximate solution may not be the k-convex best approximation of c, it could be smooth and sufficiently close to g to give an acceptable result for recovering g from c. Here is an intuitive way to view (1.1) as a reasonable scheme for data smoothing.

Since $\nabla_k g$ is very small, (6.1) is computationally equivalent to

$$w - (w - \alpha(\Delta_k w + \nabla_k d))_+ = 0. \tag{8.5}$$

Thus $d + \nabla_k^T w$ is the best k-convex approximation of d and is a smooth vector. Therefore, the approximate solution $x = g + d + \nabla_k^T W$ is a smooth vector that approximates g well. Note that x might not be k-convex and might have nothing to do with the best k-convex approximation of c, but it achieves the purpose of smoothing $c = g + d$. From this point of view, (1.1) is a reasonable device for filtering out noisy data. For example, consider Table 4 for $k = 6$ and $n = 100$ in the Appendix. Even though we have very poor a priori estimate, which means that the approximation solution may be far away from the exact solution, the approximate solution is very close to the original unperturbed smooth data g and we achieve the objective of smoothing c. This supports the point of view that (1.1) is a good data smoothing technique.

9. Summary. In this paper we reformulate the Karush–Kuhn–Tucker conditions for convex optimization problems as unconstrained convex minimization problems. Such reformulations provide new ways to develop efficient algorithms for solving many convex optimization problems, such as the least-distance problem, the symmetric monotone linear complementarity problem, and the convex quadratic programming problem with bounded constraints. The Newton method with exact line search is used to solve such a reformulation of the so-called k-convex approximation problem, the least-distance problem with k-convex constraints. The problem itself is ill conditioned, but our numerical results are very promising. We test the algorithm with respect to 12 standard elementary functions and various k, n. The performance of our algorithm is quite satisfactory. Because of the variety of the data tested, we believe that our algorithm should perform very well in general cases. However, the true challenge to our algorithm is to efficiently solve more general problems, such as the symmetric subproblems of the matrix splitting method [4], [20], [22]–[27] and the strictly convex quadratic programming problem with bounded constraints (see [29] and the references therein). It would be interesting to see how well our algorithm performs when applied to the following four classes of practical problems (mentioned in [29]): contact and friction in rigid-body mechanics, journal bearing lubrication, flow through a porous medium, and elastic torsion.

Some ad hoc tests have also been performed, and the algorithm has never failed if $n^k \leq 10^9$. There are some interesting phenomena in our ad hoc tests: if c is generated by a convex function with small random perturbations, the algorithm finds a very accurate solution in an extremely short time; if c is generated by $\sin(j\pi t)$ with small random perturbations, then the algorithm is slow to find the best convex approximation (i.e., the solution of (1.1) for $k = 2$) in general. A possible explanation for the difficulty of finding the best convex approximation of $\sin(j\pi t), j = 2, 3, \ldots,$ is that it generates the eigenvectors of Δ_1 corresponding to λ_j (see §4); thus the ill conditioning affects the accuracy of the descent direction.

As noted by Cullinan [5], it is very difficult to predict which K to use to filter out the noisy data from a set of contaminated smooth data (see Tables 1–8 in the Appendix). We also do not know whether the step size t in our algorithm should be in the open interval (0,2) (with exact arithmetic) or not.

Our algorithm seems very attractive if the matrices involved in the system of piecewise linear equations are banded. Otherwise, approximations to the descent direction might be necessary, by using the conjugate gradient method, for example. Further research in this direction is needed to determine that the reformulation does provide ways to develop efficient algorithms for general problems without the banded structure.

Appendix.

Group 1: $n = 100, 3 \leq k \leq 6$.

TABLE 1
$n = 100, k = 3$.

$g(t)$	N	err	err$_0$	N_0	CPU	err$_1$	err$_2$	M	M_0
$\sqrt{t}$	64	0.55E−07	0.16E−11	6.03	1.47	0.39E−12	0.31E−01	0	0
t	76	0.19E−06	0.21E−11	6.68	1.76	0.14E−09	0.51E−01	0	0
t^2	61	0.27E−08	0.68E−13	5.95	1.39	0.70E−12	0.10E−01	0	0
t^3	37	0.88E−06	0.65E−11	6.41	0.84	0.14E−12	0.10E+00	0	0
$\exp(t)$	47	0.19E−07	0.15E−10	6.21	1.07	0.55E−12	0.50E−01	0	0
$\exp(-t)$	59	0.37E−07	0.15E−11	6.37	1.36	0.98E−10	0.23E−01	0	0
$\sin(\pi t)$	91	0.31E−05	0.39E−10	5.79	2.09	0.23E−10	0.80E−01	0	0
$\sin(2\pi t)$	46	0.21E−06	0.17E−11	7.02	1.06	0.28E−12	0.73E−01	0	0
$\sin(4\pi t)$	129	0.20E−06	0.25E−11	6.79	2.98	0.54E−10	0.12E+01	1	128
$\cos(\pi t)$	44	0.37E−08	0.45E−13	7.05	1.03	0.29E−11	0.48E−01	0	0
$\cos(2\pi t)$	116	0.54E−05	0.29E−09	6.84	2.67	0.41E−09	0.58E+00	0	0
$\cos(4\pi t)$	133	0.20E−04	0.46E−09	6.43	3.03	0.40E−09	0.11E+01	0	0

TABLE 2
$n = 100, k = 4.$

$g(t)$	N	**err**	$\mathbf{err}_0$	N_0	CPU	$\mathbf{err}_1$	$\mathbf{err}_2$	M	M_0
$\sqrt{t}$	166	0.19E−04	0.26E−09	6.83	4.93	0.88E−09	0.13E+00	1	134
t	114	0.59E−03	0.32E−09	6.52	3.39	0.12E−10	0.36E−01	0	0
t^2	116	0.49E−05	0.59E−11	7.16	3.46	0.75E−11	0.33E−01	0	0
t^3	110	0.21E−04	0.77E−11	6.77	3.24	0.20E−08	0.35E−01	0	0
$\exp(t)$	119	0.64E−05	0.22E−09	6.86	3.50	0.21E−09	0.47E−01	0	0
$\exp(-t)$	100	0.28E−04	0.14E−10	6.75	2.96	0.19E−11	0.80E−01	0	0
$\sin(\pi t)$	83	0.21E−03	0.70E−10	6.42	2.46	0.88E−11	0.36E−01	0	0
$\sin(2\pi t)$	160	0.72E−06	0.16E−11	6.70	4.68	0.11E−09	0.20E+00	2	153
$\sin(4\pi t)$	220	0.37E−03	0.87E−09	6.57	6.50	0.10E−08	0.14E+01	0	0
$\cos(\pi t)$	78	0.86E−05	0.27E−10	6.32	2.30	0.17E−10	0.25E−01	0	0
$\cos(2\pi t)$	202	0.68E−05	0.91E−11	6.34	5.92	0.54E−09	0.53E+00	1	201
$\cos(4\pi t)$	300	0.41E−02	0.10E−08	3.05	8.07	0.13E−09	0.87E+00	0	0

TABLE 3
$n = 100, k = 5.$

$g(t)$	N	**err**	$\mathbf{err}_0$	N_0	CPU	$\mathbf{err}_1$	$\mathbf{err}_2$	M	M_0
$\sqrt{t}$	114	0.99E−04	0.27E−11	6.34	4.19	0.66E−09	0.80E−01	1	113
t	172	0.23E−04	0.14E−11	7.01	6.37	0.29E−09	0.39E−01	1	171
t^2	233	0.13E−03	0.18E−11	7.23	8.69	0.40E−09	0.36E−01	1	232
t^3	198	0.16E−04	0.10E−11	6.61	7.31	0.34E−09	0.52E−01	1	197
$\exp(t)$	223	0.12E−03	0.21E−11	7.04	8.25	0.31E−09	0.96E−01	1	222
$\exp(-t)$	300	0.23E−02	0.71E−10	3.73	10.39	0.68E−07	0.21E−01	0	0
$\sin(\pi t)$	189	0.11E−03	0.17E−11	6.65	6.96	0.72E−09	0.28E−01	1	188
$\sin(2\pi t)$	288	0.42E−03	0.26E−10	7.40	10.61	0.61E−08	0.20E+00	1	287
$\sin(4\pi t)$	201	0.92E−04	0.58E−11	6.75	7.41	0.93E−09	0.44E+00	1	200
$\cos(\pi t)$	158	0.65E−03	0.45E−11	6.43	5.83	0.83E−09	0.27E−01	1	157
$\cos(2\pi t)$	170	0.10E−03	0.30E−11	6.51	6.24	0.93E−09	0.86E−01	1	169
$\cos(4\pi t)$	300	0.17E−02	0.47E−10	6.08	10.83	0.16E−07	0.11E+01	1	263

TABLE 4
$n = 100, k = 6.$

$g(t)$	N	**err**	$\mathbf{err}_0$	N_0	CPU	$\mathbf{err}_1$	$\mathbf{err}_2$	M	M_0
$\sqrt{t}$	300	0.48E+01	0.70E−08	6.88	13.55	0.28E−04	0.42E−01	15	187
t	300	0.91E+00	0.57E−09	5.43	13.09	0.19E−05	0.71E−01	1	209
t^2	300	0.48E+00	0.44E−09	4.53	12.96	0.18E−05	0.88E−01	2	184
t^3	300	0.23E+00	0.20E−09	4.24	12.83	0.81E−06	0.61E−01	0	0
$\exp(t)$	300	0.36E−02	0.53E−11	6.22	13.40	0.55E−08	0.62E−01	10	231
$\exp(-t)$	300	0.97E+00	0.71E−09	6.17	13.33	0.22E−05	0.36E−01	7	220
$\sin(\pi t)$	300	0.17E−01	0.31E−10	6.21	13.27	0.13E−06	0.53E−01	1	245
$\sin(2\pi t)$	300	0.19E+00	0.56E−09	4.91	13.03	0.23E−05	0.73E−01	8	174
$\sin(4\pi t)$	300	0.66E+04	0.27E−03	7.17	13.50	0.86E+00	0.57E+00	1	293
$\cos(\pi t)$	300	0.50E−02	0.20E−10	3.81	12.79	0.74E−07	0.48E−01	0	0
$\cos(2\pi t)$	300	0.27E−01	0.17E−09	5.72	13.20	0.17E−06	0.56E−01	0	0
$\cos(4\pi t)$	300	0.19E+00	0.71E−09	6.63	13.47	0.68E−06	0.82E+00	5	250

Group 2: $n = 400, 2 \leq k \leq 3$.

TABLE 5
$n = 400, k = 2$.

$g(t)$	N	err	err_0	N_0	CPU	err_1	err_2	M	M_0
$\sqrt{t}$	417	0.44E−07	0.15E−09	6.35	27.92	0.51E−10	0.27E+00	0	0
t	78	0.42E−07	0.36E−11	6.85	5.19	0.58E−10	0.42E−01	0	0
t^2	36	0.18E−10	0.37E−14	8.14	2.43	0.98E−14	0.83E−01	1	35
t^3	29	0.12E−07	0.70E−12	7.41	1.94	0.34E−13	0.67E−01	0	0
$\exp(t)$	26	0.69E−09	0.61E−12	7.12	1.74	0.21E−13	0.17E−01	0	0
$\exp(-t)$	29	0.70E−08	0.58E−12	7.45	1.95	0.62E−11	0.72E−01	0	0
$\sin(\pi t)$	423	0.93E−07	0.53E−11	6.73	28.17	0.10E−10	0.63E+00	1	422
$\sin(2\pi t)$	304	0.57E−06	0.10E−09	6.64	20.11	0.17E−08	0.11E+01	0	0
$\sin(4\pi t)$	428	0.92E−08	0.37E−11	6.83	28.85	0.10E−10	0.12E+01	1	427
$\cos(\pi t)$	356	0.39E−05	0.38E−09	6.20	23.80	0.38E−10	0.25E+00	0	0
$\cos(2\pi t)$	92	0.26E−08	0.20E−12	6.84	6.09	0.45E−12	0.27E+00	1	91
$\cos(4\pi t)$	276	0.80E−07	0.40E−11	7.12	18.40	0.95E−11	0.12E+01	1	275

TABLE 6
$n = 400, k = 3$.

$g(t)$	N	err	err_0	N_0	CPU	err_1	err_2	M	M_0
$\sqrt{t}$	192	0.98E−04	0.64E−10	7.81	17.21	0.16E−11	0.57E−01	0	0
t	412	0.13E−04	0.55E−11	7.04	36.84	0.52E−10	0.60E−01	2	409
t^2	302	0.13E−05	0.18E−11	8.44	27.30	0.18E−10	0.19E−01	1	301
t^3	206	0.41E−04	0.25E−10	8.54	18.52	0.16E−08	0.81E−01	0	0
$\exp(t)$	309	0.36E−06	0.56E−12	7.94	27.84	0.44E−11	0.96E−01	1	308
$\exp(-t)$	370	0.47E−03	0.86E−09	7.59	33.19	0.53E−07	0.31E−01	0	0
$\sin(\pi t)$	630	0.18E−04	0.56E−11	8.69	56.66	0.50E−10	0.57E−01	1	629
$\sin(2\pi t)$	1200	0.16E−02	0.60E−09	2.88	96.46	0.38E−07	0.85E−01	0	0
$\sin(4\pi t)$	756	0.82E−03	0.29E−09	7.72	68.21	0.29E−08	0.11E+01	2	753
$\cos(\pi t)$	234	0.31E−04	0.10E−09	8.40	21.04	0.65E−08	0.24E−01	0	0
$\cos(2\pi t)$	973	0.34E−03	0.69E−10	8.57	87.88	0.66E−09	0.62E+00	1	972
$\cos(4\pi t)$	758	0.36E−03	0.21E−09	8.10	68.36	0.30E−08	0.11E+01	1	757

Group 3: $n = 2000, 1 \leq k \leq 2$.

TABLE 7
$n = 2000, k = 1$.

$g(t)$	N	err	err_0	N_0	CPU	err_1	err_2	M	M_0
$\sqrt{t}$	10	0.13E−12	0.83E−15	6.30	2.36	0.44E−15	0.44E−01	1	9
t	9	0.10E−12	0.78E−15	7.22	2.13	0.22E−15	0.57E−01	1	8
t^2	8	0.21E−06	0.81E−09	7.00	1.92	0.23E−08	0.87E−01	0	0
t^3	11	0.74E−13	0.11E−14	6.45	2.61	0.44E−15	0.96E−01	1	10
$\exp(t)$	9	0.50E−13	0.73E−15	6.78	2.11	0.89E−15	0.35E−01	1	8
$\exp(-t)$	10	0.14E−07	0.83E−10	5.90	2.29	0.41E−10	0.37E+00	0	0
$\sin(\pi t)$	37	0.10E−07	0.64E−10	3.11	7.99	0.40E−11	0.72E+00	0	0
$\sin(2\pi t)$	74	0.40E−08	0.15E−09	2.72	16.28	0.23E−10	0.10E+01	0	0
$\sin(4\pi t)$	49	0.75E−07	0.49E−09	3.00	10.84	0.18E−08	0.10E+01	0	0
$\cos(\pi t)$	10	0.37E−08	0.23E−09	4.90	2.33	0.20E−11	0.10E+01	0	0
$\cos(2\pi t)$	50	0.18E−07	0.11E−09	2.74	10.68	0.46E−09	0.12E+01	0	0
$\cos(4\pi t)$	54	0.51E−10	0.66E−12	3.19	11.91	0.17E−12	0.12E+01	1	53

TABLE 8
$n = 2000, k = 2.$

$g(t)$	N	err	err_0	N_0	CPU	err_1	err_2	M	M_0
$\sqrt{t}$	2436	0.94E−05	0.42E−10	8.15	828.61	0.49E−10	0.26E+00	2	1762
t	222	0.22E−04	0.12E−09	8.84	75.05	0.68E−12	0.11E+00	0	0
t^2	149	0.22E−07	0.92E−13	9.39	50.41	0.20E−12	0.99E−01	1	148
t^3	120	0.41E−06	0.46E−11	9.23	40.65	0.78E−13	0.34E−01	0	0
$\exp(t)$	118	0.18E−04	0.18E−09	9.92	39.80	0.33E−11	0.32E−01	0	0
$\exp(-t)$	135	0.11E−07	0.18E−12	8.27	45.75	0.33E−12	0.96E−01	1	134
$\sin(\pi t)$	2579	0.40E−04	0.34E−09	7.48	859.29	0.36E−09	0.63E+00	1	2578
$\sin(2\pi t)$	2756	0.35E−04	0.21E−09	8.83	929.21	0.33E−09	0.11E+01	1	2755
$\sin(4\pi t)$	1951	0.11E−04	0.22E−09	9.10	651.84	0.33E−09	0.12E+01	1	1950
$\cos(\pi t)$	1920	0.44E−05	0.39E−10	7.30	642.33	0.40E−10	0.25E+00	1	1919
$\cos(2\pi t)$	1100	0.10E−05	0.96E−11	8.78	367.35	0.12E−10	0.26E+00	1	1099
$\cos(4\pi t)$	1672	0.55E−04	0.25E−09	8.04	562.73	0.38E−09	0.12E+01	1	1671

Acknowledgments. The authors thank the associate editor and the two referees for many helpful comments.

REFERENCES

[1] M. AYER, H. D. BRUNK, G. M. EWING, AND W. T. REID, *An empirical distribution function for sampling with incomplete information*, Ann. Math. Statist., 26 (1955), pp. 641–647.

[2] R. E. BARLOW, D. J. BARTHOLOMEW, J. M. BREMNER, AND H. D. BRUNK, *Statistical Inference Under Order Restriction—The Theory and Application of Isotone Regression*, John Wiley & Sons, New York, 1972.

[3] Y. CENSOR, *Row-action methods for huge and sparse systems and their applications*, SIAM Rev., 23 (1981), pp. 444–466.

[4] C. W. CRYER, *The solution of a quadratic programming problem using systematic overrelaxation*, SIAM J. Comput., 9 (1971) pp. 385–392.

[5] M. P. CULLINAN, *Data smoothing using non-negative divided differences and ℓ_2 approximation*, IMA J. Numer. Anal., 10 (1990), pp. 583–608.

[6] M. P. CULLINAN AND M. J. D. POWELL, *Data smoothing by divided differences*, in Numerical Analysis Proceedings, Dundee 1981, G. A. Watson, ed., LNIM 912, Springer-Verlag, Berlin, 1981, pp. 26–37.

[7] A. DAX, *The smallest point of a polytope*, J. Optim. Theory Appl., 64 (1990), pp. 429–432.

[8] B. C. EAVES AND U. G. ROTHBLUM, *Relationships of properties of piecewise affine maps over ordered fields*, Linear Algebra Appl., 132 (1990), pp. 1–63.

[9] B. C. EAVES AND H. SCARF, *The solution of systems of piecewise linear equations*, Math. Oper. Res., 1 (1976), pp. 1–27.

[10] R. FLETCHER, *Practical Methods of Optimization*, John Wiley & Sons, New York, 1981.

[11] T. FUJISAWA AND E. S. KUH, *Piecewise-linear theory of nonlinear networks*, SIAM J. Appl. Math., 22 (1972), pp. 307–328.

[12] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983.

[13] D. L. HANSON AND G. PLEDGER, *Consistency in concave regression*, Ann. Statist., 4 (1976), pp. 1038–1050.

[14] C. HILDRETH, *Point estimates of ordinates of concave functions*, Ann. Math. Statist., 29 (1954), pp. 598–619.

[15] ———, *A quadratic programming procedure*, Naval Res. Quart., 4 (1957), pp. 79–85; erratum, p. 361.

[16] S. KATZENELSON, *An algorithm for solving nonlinear resistor networks*, Bell Syst. Tech. J., 44 (1965), pp. 1605–1620.

[17] A. N. IUSEM AND A. R. DE PIERRO, *On the convergence properties of Hildreth's quadratic programming algorithm*, Math. Programming, 47 (1990), pp. 37–51.

[18] A. LENT AND Y. CENSOR, *Extensions of Hildreth's row-action method for quadratic programming*, SIAM J. Control Optim., 18 (1980), pp. 444–454.

[19] W. Li, *The Best Error Bounds for Perturbed Feasible and Optimal Solutions of a Linear Program*, Tech. Report TR91-14, Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA, 1991.

[20] ———, *Remarks on Convergence of Matrix Splitting Algorithm for the Symmetric Linear Complementarity Problem*, Tech. Report TR91-8, Department of Mathematics and Statistics , Old Dominion University, Norfolk, VA, 1991; SIAM J. Optim., to appear.

[21] W. Li, P. Pardalos, and C. G. Han, *Gauss–Seidel method for least distance problems*, J. Optim. Theory Appl., to appear.

[22] Y. Y. Lin and J.-S. Pang, *Iterative methods for large quadratic programs: A survey*, SIAM J. Control Optim., 25 (1987), pp. 383–411.

[23] Z. Q. Luo and P. Tseng, *On the linear convergence of descent methods for convex essentially smooth minimization*, J. Optim. Theory Appl., to appear.

[24] ———, *On the convergence of a matrix splitting algorithm for the symmetric monotone linear complementarity problem*, SIAM J. Control Optim., 29 (1991), pp. 1037–1060.

[25] ———, *Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem*, SIAM J. Optim., to appear.

[26] O. L. Mangasarian, *Solution of symmetric linear complementarity problems by iterative methods*, J. Optim. Theory Appl., 22 (1977), pp. 465–485.

[27] ———, *Convergence of iterates of an inexact matrix splitting algorithm for the symmetric monotone linear complementarity problem*, SIAM J. Optim., 1 (1991), pp. 114–122.

[28] B. F. Mitchell, V. F. Demyyanov, and V. N. Malozemov, *Finding the point of a polyhedron closest to the origin*, SIAM J. Control Optim., 12 (1974), pp. 19–26.

[29] J. J. Moré and G. Toraldo, *On the solution of large quadratic programming problems with bounded constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.

[30] K. G. Murty, *Linear Complementarity, Linear and Nonlinear Programming*, Helderman-Verlag, Berlin, 1988.

[31] M. Z. Nashed, *Generalized Inverses and Applications*, Academic Press, New York, 1976.

[32] T. Ohtsuki, T. Fujisawa, and S. Kumagai, *Existence theorems and a solution algorithm for piecewise-linear resistor networks*, SIAM J. Math. Anal., 8 (1977), pp. 69–99.

[33] J. M. Ortega, *Numerical Analysis: A Second Course*, Academic Press, New York, 1972.

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

[35] J.-S Pang, *Methods for quadratic programming: A survey*, Comput. Chem. Engrg., 7 (1983), pp. 583–594.

[36] P. M. Pardalos and N. Kovoor, *An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds*, Math. Programmming, 46 (1990), pp. 321–328.

[37] W. C. Rheinboldt and J. S. Vandergraft, *On piecewise affine mappings in* $\mathbb{R}^n$, SIAM J. Appl. Math., 29 (1975), pp. 680–689.

[38] T. Robertson, F. T. Wright, and R. L. Dykstra, *Order Restricted Statistical Inference*, John Wiley & Sons, New York, 1988.

[39] I. Singer, *The Theory of Best Approximation and Functional Analysis*, in Regional Conference Series, No. 13, Society for Industrial and Applied Mathematics, Philadelphia, 1974.

[40] J. J. Swetits and S. E. Weinstein, *Construction of a best monotone approximation in* ℓ_p *for* $1 < p < \infty$, Approx. Theory Appl., 5 (1989), pp. 69–77.

[41] ———, *Construction of the best monotone approximation in* $L_p[0,1]$, J. Approx. Theory, 61 (1990), pp. 118–130.

[42] ———, *The computation of a best monotone* L_p *approximation* $1 \leq p < \infty$, Numer. Func. Anal. Optim., 11 (1990), pp. 811–822.

[43] J. J. Swetits, S. E. Weinstein, and Y. Xu, *On the characterization and computation of the best monotone approximation in* $L_p[0,1]$ *for* $1 \leq p < \infty$, J. Approx. Theory, 60 (1990), pp. 58–69.

[44] ———, *Approximation in* $L_p[0,1]$ *by n-convex functions*, Numer. Func. Anal. Optim., 11 (1990), pp. 167–179.

[45] V. A. Ubhaya, *An* $\mathcal{O}(n)$ *algorithm for discrete n-point convex approximation with applications to continuous case*, J. Math. Anal. Appl., 72 (1979), pp. 338–354.

[46] ———, *A linear time algorithm for convex and monotone approximation*, Comput. Math. Appl., 9 (1983), pp. 326–336.

[47] P. Wolfe, *Algorithm for a least-distance programming problem*, Math. Programming Stud., 1 (1974), pp. 190–205.

[48] ———, *Find the nearest point in a polytope*, Math. Programming, 11 (1976), pp. 128–149.

[49] Y. Xu, *Best Approximation with Geometric Constraints*, Ph. D. thesis, Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA, 1990.

SIAM J. OPTIMIZATION
Vol. 3, No. 3, pp. 489–502, August 1993

SECOND-ORDER MULTIPLIER UPDATE CALCULATIONS FOR OPTIMAL CONTROL PROBLEMS AND RELATED LARGE SCALE NONLINEAR PROGRAMS[*]

J. C. DUNN[†]

Abstract. A second-order multiplier update rule is applied to K-stage discrete-time optimal control problems with control and state variable constraints. Each update entails the assembly and solution of sparse block-banded equilibrium equations. Several direct elimination solution techniques are considered, and it is shown that the updates can always be calculated in $O(K)$ flops. Part of the analysis applies not only to control problems, but also to other similarly structured large scale nonlinear programs with equality and inequality constraints.

Key words. Newtonian multiplier updates, Newtonian projection methods, banded equilibrium equations, efficient solution methods

AMS subject classifications. 49M29, 65K10, 90C06

1. Introduction.

This note examines the cost of implementing a superlinearly convergent Newtonian multiplier update rule for K-stage discrete-time optimal control problems

$$\text{(1A)} \qquad \min J(x,u) = \sum_{i=1}^{K+1} f_i^\circ(x_i, u_i)$$

subject to state and control variable constraints

$$\text{(1B)} \qquad \theta_i(x_i,u_i) \le 0, \qquad i \in \{1,\ldots,K+1\},$$

dynamic equations

$$\text{(1C)} \qquad x_{i+1} = f_i(x_i,u_i), \qquad i = 1,\ldots,K,$$

and separated end conditions

$$\text{(1D)} \qquad \phi_1(x_1) = 0, \qquad \phi_{K+1}(x_{K+1}) = 0.$$

Our objective is to show that the update rule in question can be calculated for (1) by solving any of four different systems of sparse block-banded equilibrium equations, and that each of these sparse linear systems can in turn be assembled and solved in $O(K)$ flops by one or more direct elimination algorithms. Finer numerical distinctions among the various $O(K)$ computational approaches are likely to be problem-dependent within the class (1), and are not attempted here.

The $O(K)$ cost estimates obtained in the present study actually apply to the somewhat larger class of structured nonlinear programs

$$\text{(2A)} \qquad \min J(z) = \sum_{i=1}^{K+1} f_i^\circ(z_i)$$

[*] Received by the editors July 22, 1991; accepted for publication (in revised form) March 30, 1992. This investigation was supported by National Science Foundation grant DMS-9002848.

[†] Mathematics Department, Box 8205, North Carolina State University, Raleigh, North Carolina 27695-8205.

subject to

$$g(z) \leq 0 \tag{2B}$$

and

$$h(z) = 0 \tag{2C}$$

with

$$z = (z_1, \dots, z_{K+1}), \tag{2D}$$

$$g(z) = (g_1(z), \dots, g_{K+1}(z)), \tag{2E}$$

$$h(z) = (h_1(z), \dots, h_K(z)), \tag{2F}$$

$$z_i \in \mathbb{R}^{d_i}, \tag{2G}$$

$$g_i(z) = \theta_i(z_i) \in \mathbb{R}^{q_i}, \tag{2H}$$

$$h_i(z) = \pi_i(z_i, z_{i+1}) \in \mathbb{R}^{p_i}, \tag{2I}$$

and

$$\sum_{i=1}^{K+1} d_i = d, \quad \sum_{i=1}^{K+1} q_i = q, \quad \sum_{i=1}^{K} p_i = p, \tag{2J}$$

where d_i, p_i, and q_i are positive integers, and f_i° and the scalar components $\theta_{i,j}$ and $\pi_{i,j}$ are twice continuously differentiable real-valued functions on $\mathbb{R}^d$. For such problems, J, g, and h are twice continuously differentiable functions from $\mathbb{R}^d$ to $\mathbb{R}^1$, $\mathbb{R}^q$, and $\mathbb{R}^p$; the differentials g', $g_{i,j}''$, and J'' have block-diagonal matrix representations; and h' and $h_{i,j}''$ have block-echelon and block-tridiagonal matrix representations. In particular, the constraints (1C)–(1D) can be expressed in (2F) with separable functions

$$\pi_i(z_i, z_{i+1}) = \pi_i^\circ(z_i) + \pi_i^1(z_{i+1}), \tag{3}$$

in which case $h_{i,j}''$ has a block-diagonal matrix representor. These special features can be exploited in the numerical solution of (2), and hence (1).

When the scalar components of θ_i are simple affine or convex functions (a common occurrence in the control problem setting), it may be easier to deal with (2) indirectly by embedding J and h in an augmented Lagrangian

$$\mathcal{L}(\mu, z) = J(z) + \langle \mu, h(z) \rangle + \tfrac{1}{2} c \|h(z)\|^2,$$

and addressing the relaxed problems,

$$\min_{g(z) \leq 0} \mathcal{L}(\mu, z) \tag{4}$$

for a sequence of multiplier vectors

$$\mu = (\mu_1, \dots, \mu_K), \qquad \mu_i \in \mathbb{R}^{p_i},$$

generated by update rules

$$\mu \to \mu + \Delta\mu, \tag{5A}$$

$$B(\mu, z(\mu))\Delta\mu = h(z(\mu)), \tag{5B}$$

where $z(\mu)$ is a solution of (4), $B(\mu, z)$ is a suitably constructed $p \times p$ matrix, and μ, $\Delta\mu$, and $h(z)$ are written in column matrix form (see [1] for an exposition of the general multiplier method). In practice, $z(\mu)$ is replaced by an approximate stationary point $\hat{z}(\mu)$ for (4), obtained with a truncated iteration of some algorithm that capitalizes on the special structure of g and $\mathcal{L}$ for (2). If each iteration of this inner algorithm for (4) can be done cheaply, then the overall effectiveness of the computational scheme will depend on the convergence properties of the inner solver for (4) and the outer update loop (5), and on the cost of solving (5B). The present investigation deals with the last question for problem (2) and the second-order Newtonian update scheme described in §2. Local superlinear convergence theorems are proved in [2] for multiplier methods that employ this scheme in conjunction with a Newtonian projection method for general finite-dimensional nonlinear programs with equality and inequality constraints. Analogous convergence results are also proved in [2] for asymptotically exact multiplier methods based on a related modified second-order update rule that incorporates an additional term on the right side of (5); the new term is generated by the inner Newtonian projection algorithm for (4), and computational cost estimates qualitatively similar to those obtained in §2 can also be established for the modified second-order rule.

The update rule in §2 has been applied previously in a different way to control problems with no intermediate state variable restrictions [3], [4]. In such cases, it is possible to treat x as a function of u determined by the dynamic equations, compute multiplier updates for the remaining end condition in $O(K)$ flops with dynamic programming [3]–[6] or other comparably efficient methods [7], [8], and still retain simplicity in the inequality constraints for the counterpart of the relaxed problem (4); however, this approach seems ill suited to problems with general intermediate state/control variable constraints, where the elimination of x converts the simple separable conditions (1B) into nonseparable inequality constraints on u, and makes the inner relaxed problem correspondingly more difficult (particularly for the gradient projection algorithms used in [2]–[4]). On the other hand, while the formulation investigated here preserves simplicity in (1B) at the cost of dealing with a potentially large number of new multipliers for the dynamic equations, the resulting linear system (5B) can be rearranged and solved in $O(K)$ flops, provided that d_i, p_i, and q_i remain bounded as $i \to \infty$ (this provision is tacitly enforced from here onward).

Alternative differential dynamic programming, sequential quadratic programming, and augmented Lagrangian algorithms for control problems and other large sparse nonlinear programs are described in [9]–[14]. The slack variable multiplier schemes for nonlinear programs described in [1] and [21] are also potentially valuable for optimal control problems, and lead to similar second-order update calculations. For example, in Chap. 3 of [1], inequality constraints are converted to equality constraints with squared slack variables ζ, all of the constraints are incorporated in an augmented

Lagrangian, and (4) is replaced by an *unconstrained* minimization problem in the expanded independent variable vector (z, ζ); moreover, as shown in [1], [22], and [23], the inner minimization with respect to ζ can be done explicitly for each fixed z, leaving a reduced unconstrained minimization problem of the form

$$\min_{z\in\mathbb{R}^n}\left\{J(z)+\langle\mu,h(z)\rangle+\frac{1}{2}c\|h(z)\|^2+\frac{1}{2c}\sum_{i=1}^{K+1}\sum_{j=1}^{q_i}\phi\left(\lambda_{i,j},g_{i,j}(z)\right)\right\},$$

where

$$\phi(s,t)=[\max(0,s+ct)]^2-s^2.$$

For problem (2), the corresponding second-order update rule for (μ,λ) in [1] entails the solution of sparse block-banded equilibrium equations like those considered here, with attendant $O(K)$ cost estimates. Finally, it should be noted that the slack variable approach and the dynamic programming technique in [5] provide another alternative multiplier method for control problems with intermediate stage-wise inequality constraints on control variables only; however, complications arise here once again when intermediate constraints are imposed on control *and* state variables.

2. The second-order multiplier update rule. In principle, the update scheme described in this section determines $\Delta\mu$ by solving the $p\times p$ linear system (5B) with coefficient matrix

$$B(\mu,z)=(Q_T^t\nabla h)^t(Q_T^tLQ_T)^{-1}(Q_T^t\nabla h), \tag{6A}$$

where the columns of the $d\times\tau$ matrix Q_T supply an orthonormal basis for the subspace T orthogonal to selected "ϵ-active" (i.e., almost active) g-constraint gradients at (μ,z), the columns of the $d\times p$ matrix ∇h are the gradients of the h-constraints at z, and the $d\times d$ matrix L is formed from the z-Hessians of $\mathcal{L}$ and the ϵ-active g-constraints, and from a related "least squares" estimate $\lambda(\mu,z)$ of the g-multiplier for the relaxed problem (4). More precisely

$$L=\nabla_{zz}^2\mathcal{L}(\mu,z)+\sum_{i=1}^{K+1}\sum_{j=1}^{q_i}\lambda_{i,j}(\mu,z)\nabla^2g_{i,j}(z), \tag{6B}$$

where $\lambda\in\mathbb{R}^q$ satisfies the linear equations

$$\sum_{i=1}^{K+1}\sum_{j=1}^{q_i}\lambda_{i,j}\nabla g_{i,j}=-(I-Q_TQ_T^t)\nabla_z\mathcal{L}, \tag{6C}$$

$$\lambda_{i,j}=0,\qquad j\notin\alpha_i(\mu,z),\qquad i=1,\dots,K+1, \tag{6D}$$

with

$$\alpha_i(\mu,z)=\{j\in\{1,\dots,q_i\}:g_{i,j}(z)\geq-\|\nabla g_{i,j}(z)\|\epsilon(\mu,z)\}. \tag{6E}$$

With some minor abuse of notation, let ∇g denote the $d\times\nu$ matrix with columns formed from the gradients of the ϵ-active constraints at (μ,z). The general analysis in [2] shows that if $\epsilon(\mu,\cdot)$ is a suitably defined nonnegative continuous measure of nonstationarity for problem (4), and if the penalty constant c in $\mathcal{L}$ is sufficiently large, then null $[\nabla h \ \nabla g] = \{0\}$, and L is positive-definite on null $\nabla g^t = T$ uniformly in (μ, z) near (μ^*, z^*), provided that

I. The gradients of the h-constraints and the *active* g-constraints at z^* are linearly independent.

II. The Kuhn–Tucker first-order necessary conditions and the strict complementarity condition for problem (2) hold at z^*, with h-multiplier μ^* (and g-multiplier λ^*).

III. The second-order Kuhn–Tucker sufficient condition for (2) holds at z^*.

Under these circumstances, it is not difficult to see that null $Q_T^t \nabla h = \{0\}$, and the matrices $Q_T^t L Q_T$ and B in (6) are positive-definite uniformly in (μ, z) near (μ^*, z^*); moreover, near (μ^*, z^*) each of the following linear systems has an invertible coefficient matrix and $\Delta\mu$ may be obtained by solving either set of equations:

$$\text{(7)} \qquad \begin{bmatrix} Q_T^t L Q_T & Q_T^t \nabla h \\ (Q_T^t \nabla h)^t & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \Delta\mu \end{bmatrix} = \begin{bmatrix} 0 \\ -h \end{bmatrix}$$

or

$$\text{(8)} \qquad \begin{bmatrix} L & \nabla h & \nabla g \\ \nabla h^t & 0 & 0 \\ \nabla g^t & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi' \\ \Delta\mu \\ \eta \end{bmatrix} = \begin{bmatrix} 0 \\ -h \\ 0 \end{bmatrix}$$

(see [1] and [2], and also the proof of Theorem 2.1 in this section). These systems belong to the class of equilibrium equations associated with Lagrangian formulations for other constrained optimization algorithms (cf. [11], [13], and [15]), and with variational principles for structural mechanics, electrical networks, and fluid dynamics [16], [17].

In the present context, some of the claims made above for (7) and (8) continue to hold if L is replaced by

$$\text{(9)} \qquad \begin{aligned} D = L - c\nabla h \nabla h^t &= \nabla^2 J + \sum_{i=1}^{K} \sum_{j=1}^{q_i} (\mu_{i,j} + c h_{i,j}) \nabla^2 h_{i,j} \\ &+ \sum_{i=1}^{K+1} \sum_{j=1}^{q_i} \lambda_{i,j} \nabla^2 g_{i,j}, \end{aligned}$$

and $\Delta\mu$ is replaced by $\Delta\mu - ch$ (cf. [1, pp. 134–135]). There are immediate advantages to be gained by doing this since the computation of $\nabla h \ \nabla h^t$ is thereby avoided, and since D is block-diagonal when condition (3) holds (as it does for control problems). On the other hand, conditions I–III do not insure that D is positive-definite on null $[\nabla g^t]$ or even that $Q_T^t D Q_T$ is invertible near (μ^*, z^*). Nevertheless, the following theorem shows that the resulting modifications of (7) and (8) are worth considering.

THEOREM 2.1. *Suppose that* null $[\nabla h \nabla g] = \{0\}$ *and that* $L = D + c\nabla h \nabla h^t$ *is positive-definite on* null ∇g^t. *Then:*

(i) *The coefficient matrix in each of the systems below is invertible:*

$$\text{(10)} \qquad \begin{bmatrix} Q_T^t D Q_T & Q_T^t \nabla h \\ (Q_T^t \nabla h)^t & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} 0 \\ -h \end{bmatrix},$$

$$\begin{bmatrix} D & \nabla h & \nabla g \\ \nabla h^t & 0 & 0 \\ \nabla g^t & 0 & 0 \end{bmatrix} \begin{bmatrix} \xi' \\ \eta \\ \zeta \end{bmatrix} = \begin{bmatrix} 0 \\ -h \\ 0 \end{bmatrix}. \tag{11}$$

(ii) *The following statements are equivalent:*

a. *$\Delta\mu$ is the unique solution of* (5B)–(6);

b. *For some ξ, $(\xi, \Delta\mu - ch)$ is the unique solution of* (10);

c. *For some ξ' and ζ, $(\xi', \Delta\mu - ch, \zeta)$ is the unique solution of* (11).

(iii) *If the columns of H supply a basis for* null $(Q_T^t \nabla h)^t$ *then $H^t Q_T^t D Q_T H$ is positive-definite.*

Proof. First note that if $Q_T^t \nabla h \eta = 0$ then $\nabla h \eta \in (\text{null } \nabla g^t)^\perp = \text{co}\ell \nabla g$ and therefore $\nabla h \eta + \nabla g \zeta = 0$ for some ζ. By hypothesis, this implies that $\eta = 0$ and hence

$$\text{null } Q_T^t \nabla h = \{0\}.$$

Second, note that if $(Q_T^t \nabla h)^t \xi = 0$ then $Q_T^t D Q_T \xi = Q_T^t L Q_T \xi$. Consequently, if (ξ, η) is in the null space of the coefficient matrix in (11) then

$$Q_T^t D Q_T \xi + Q_T^t \nabla h \eta = 0$$

and

$$\xi^T Q_T^t L Q_T \xi = 0.$$

By hypothesis, the last equation implies that $\xi = 0$, and the previous two equations then imply that $\eta = 0$. This proves that the coefficient matrix in (10) is invertible. With a similar argument, it can be shown that the coefficient matrix in (11) is also nonsingular.

Now observe that $Q_T^t L Q_T$ is positive-definite by hypothesis, and therefore $\Delta\mu$ satisfies (5B)–(6) if and only if for some ξ,

$$(Q_T^t \nabla h)^t \xi = -h, \qquad Q_T^t L Q_T \xi = -Q_T^t \nabla h \, \Delta\mu.$$

Furthermore, the latter equations are satisfied if and only if

$$(Q_T^t \nabla h)^t \xi = -h, \qquad Q_T^t D Q_T \xi = -Q_T^t \nabla h \, (\Delta\mu - ch).$$

Hence (iia) and (iib) are equivalent statements. To prove that (iib) and (iic) are also equivalent, let $\xi' = Q_T \xi$ and note that

$$\nabla g^t \xi' = 0$$

and that (ξ, η) solves (10) if and only if

$$\nabla h^t \xi' = -h, \qquad Q_T^t (D\xi' + \nabla h \, \eta) = 0.$$

Moreover, the last equation holds if and only if $D\xi' + \nabla h \eta \in \text{co}\ell \nabla g$, i.e., if for some ζ,

$$D\xi' + \nabla h \eta + \nabla g \zeta = 0.$$

Finally, observe that $H^tQ_TLQ_TH$ is positive-definite and that

$$\begin{aligned} H^tQ_TLQ_TH &= H^tQ_T^tDQ_TH + cH^t(Q_T^t\nabla h)(Q_T^t\nabla h)^tH \\ &= H^tQ_T^tDQ_TH. \qquad \square \end{aligned}$$

When no g-constraints are ϵ-active, the matrix ∇g is absent in (8) and (11), the $d \times d$ identity matrix will serve for Q_T in (6), (7), and (10), and the resulting equations characterize the standard second-order update rule for equality-constrained minimization [1].

3. Implementation costs. Equations (5B)–(8), (10), and (11) supply alternative descriptions of a second-order multiplier update rule that actually applies to a general class of nonlinear programs with equality and inequality constraints in $\mathbb{R}^d$. The costs associated with assembling and solving these equations for problem (2) are now considered in more detail. In this analysis, it is assumed that null $[\nabla h\ \nabla g] = \{0\}$ and that $L = D + c\nabla h\nabla h^t$ is positive-definite on $T =$ null ∇g^t (see conditions I–III and the related discussion in §2).

For problem (2), ∇h is a $d \times p$-sparse "block-echelon form" matrix

$$\begin{bmatrix} (\nabla h)_{1,1} & \cdots & (\nabla h)_{1,K} \\ \vdots & & \\ (\nabla h)_{K+1,1} & & (\nabla h)_{K+1,K} \end{bmatrix}$$

with $d_i \times p_j$-dimensional submatrices

$$(\nabla h)_{i,j} = \begin{cases} 0, & i \notin \{j, j+1\}, \\ \nabla_{z_i}\pi_j(z_j, z_{j+1}), & i \in \{j, j+1\} \end{cases} \tag{12}$$

for $1 \leq i \leq K+1$ and $1 \leq j \leq K$. Furthermore, if there are ν_i ϵ-active g-constraint indices in the set $\alpha_i(\mu, z)$ and if

$$\nu = \nu_1 + \cdots + \nu_{K+1}$$

with

$$\nu_i > 0 \Longleftrightarrow i \in \{\rho_1 < \cdots < \rho_k\}$$

for all i, then ∇g is a sparse $d \times \nu$ matrix with $d_i \times \nu_{\rho_i}$-dimensional submatrices

$$(\nabla g)_{i,j} = \begin{cases} 0, & i \neq \rho_j, \\ \nabla\theta_i(z_i), & i = \rho_j \end{cases} \tag{13}$$

for $1 \leq i \leq K+1$ and $1 \leq j \leq K$. Accordingly, the subspace T is a direct sum

$$T_1 = T_1 \oplus \cdots \oplus T_{K+1} \tag{14A}$$

with

$$T_i = \begin{cases} \text{null}\,\nabla\theta_i^t \subset \mathbb{R}^{d_i}, & i \in \{\rho_1, \ldots, \rho_k\}, \\ \mathbb{R}^{d_i}, & i \notin \{\rho_1, \ldots, \rho_k\}, \end{cases} \tag{14B}$$

(14C) $$\tau_i = \dim T_i = d_i - \nu_i,$$

(14D) $$\tau = \dim T = \tau_1 + \cdots + \tau_{K+1}.$$

Since null $[\nabla h\ \nabla g] = \{0\}$ it follows that $\tau_i > 0$ for some $i \in \{1, \ldots, K+1\}$. Assume that

$$\tau_i > 0, \qquad i \in \{\sigma_1 < \cdots < \sigma_\ell\} \neq \emptyset.$$

Then Q_T is a sparse $d \times \tau$ matrix with $d_i \times \tau_{\sigma_i}$-dimensional submatrices

(15) $$(Q_T)_{i,j} = \begin{cases} 0, & i \neq \sigma_j, \\ Q_{T_i}, & i = \sigma_j \end{cases}$$

for $1 \leq i \leq K+1$ and $1 \leq j \leq \ell$, where the columns of $Q_{T_{\sigma_j}}$ provide an orthonormal basis for T_{σ_j}. It can now be seen that $\nabla h\, \nabla h^t$, L, and D are $d \times d$-dimensional block-tridiagonal matrices with $d_i \times d_j$-dimensional blocks, D is block-diagonal if condition (3) holds, $Q_T^t L Q_T$ and $Q_T^t D Q_T$ are $\tau \times \tau$-dimensional block-tridiagonal matrices with $\tau_{\sigma_i} \times \tau_{\sigma_j}$-dimensional blocks, $Q_T^t D Q_T$ is block-diagonal if (3) holds, and $Q_T^t \nabla h$ is a $\tau \times p$ block-echelon form matrix with $\tau_{\sigma_i} \times p_j$-dimensional blocks

(16) $$(Q_T^t \nabla h)_{i,j} = \begin{cases} 0, & j \notin \{\sigma_i - 1, \sigma_i\}, \\ Q_{T_{\sigma_i}} \nabla_{z_{\sigma_i}} \pi_j, & j \in \{\sigma_i - 1, \sigma_i\} \end{cases}$$

for $1 \leq i \leq \ell$ and $1 \leq j \leq K$. In the proof of Theorem 2.1, it was shown that null $Q_T^t \nabla h = \{0\}$, hence $Q_T^t \nabla h$ cannot have a column of zero blocks and so

(17A) $$K \leq 2\ell \leq 2(K+1)$$

and

(17B) $$\sigma_{i-1} + 1 \leq \sigma_i \leq \sigma_{i-1} + 2$$

for $1 \leq i \leq \ell$, with

(17C) $$\sigma_o = 0.$$

Finally, for problem (2), equations (6D) decompose into $K+1$ uncoupled linear systems

(18A) $$\lambda_{i,j} = 0, \qquad j \notin \alpha_i(\mu, z),$$

(18B) $$\sum_{j=1}^{q_i} \lambda_{i,j} \nabla \theta_{i,j} = -Q_{N_i} \nabla_{z_i} \mathcal{L},$$

with

(18C) $$Q_{N_i} = \begin{cases} I - Q_{T_i} Q_{T_i}^t, & \text{if } \nu_i < d_i, \\ I, & \text{if } \nu_i = d_i. \end{cases}$$

In general, the nonzero blocks in Q_T can be obtained from k QR decompositions of the $d_{\rho_i} \times d_{\nu_i}$-dimensional matrices $\nabla\theta_{\rho_i}(z_{\rho_i})$, and the nonzero components of λ can be formed by solving k associated $\nu_{\rho_i} \times \nu_{\rho_i}$-dimensional upper triangular linear systems in place of (18). If derivative evaluation costs for f_i^o, θ_i, and π_i remain bounded as $i \to \infty$, it then follows that the overall cost of assembling the coefficient matrix in each of the equations (7), (8), (10), and (11) is $O(K)$. Further significant simplifications are possible in important special cases. For example, if g is affine the Hessians $\nabla^2 g_{i,j}$ vanish and the multipliers λ are not required in L and D. More specifically, if g is affine and the constraint $g \leq 0$ expresses upper and lower bounds on the components of z, then the columns of $\nabla\theta_i$ are a subset of the columns of the $d_i \times d_i$ identity matrix I, and the remaining columns of I produce the nonzero block Q_{T_i} in Q_T; under these circumstances, $Q_T^t \nabla h$ is obtained by merely deleting selected rows of ∇h, and $Q_T^t L Q_T$ and $Q_T^t D Q_T$ are obtained by deleting rows and columns of L and D. As noted earlier, the matrix D (and hence $Q_T^t D Q_T$) is block-diagonal if condition (3) holds. In particular, if h is affine then (3) holds, D is block-diagonal, and the Hessians $\nabla^2 h_{i,j}$ vanish in L and D. Finally, since $\lambda(\mu, z)$ is nonnegative near (μ^*, z^*) satisfying conditions I–III in §2 [2], it follows that D and $Q_T^t D Q_T$ are block-diagonal and positive-definite near (μ^*, z^*) when h is affine, $g_{i,j}$ is convex, and $\nabla^2 f_i^o(z_i^*)$ is positive-definite.

With reference to §2, the systems (7), (8), (10), and (11) can now be seen as equilibrium equations

$$(19) \qquad \begin{bmatrix} A & E \\ E^t & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$$

with nonsingular sparse block-banded coefficient matrices. In principle, (19) can be solved by Gaussian elimination with partial pivoting; however, the systems of interest here have block dimensions and bandwidths of order $O(K)$, and in such cases pivoting can produce extensive fill-in in the lower right coefficient block with potentially prohibitive attendant computational costs [18], [19]. For instance, even if $Q_T^t D Q_T$ is diagonal in (10), the initial pivoting operations can place a lower Hessenberg matrix in the lower right block, and the overall cost of solving (19) for y is then $O(K^2)$. The situation is still less favorable for (7), (8), and (11), where operation counts for standard elimination algorithms may reach $O(K^3)$. Fortunately, it is possible to achieve much better results with other methods that exploit the structure in (7), (8), (10), and (11).

When A is invertible and null $E = \{0\}$, the so-called "displacement method" for nonsingular systems (19) uses block-elimination to replace the lower left block with 0 and the lower right block with $-E^t A^{-1} E$. Since (19) is nonsingular, the matrix $E^t A^{-1} E$ must be invertible and the next stages of block elimination will produce y as the unique solution of

$$(20) \qquad E^t A^{-1} E y = A^{-1} a - b$$

(cf. [13], [16], and [17]). This method is applicable and efficient for (10) in those cases where $Q_T^t D Q_T$ is invertible and block-diagonal, since the corresponding system (20) reduces to

$$(21\text{A}) \qquad C \Delta\mu = (I + cC)h$$

with a block-tridiagonal invertible $p \times p$ coefficient matrix

$$(21\text{B}) \qquad C = (Q_T^t \nabla h)^t (Q_T^t D Q_T)^{-1} (Q_T \nabla h),$$

and $\Delta\mu$ can be computed at a total cost of $O(K)$ flops; however, the invertibility of $Q_T^t D Q_T$ near (μ^*, z^*) does not follow from the standard regularity conditions I–III in §2. On the other hand, the displacement method is universally applicable but inefficient for (7) near (μ^*, z^*) (where (20) is precisely (5B)–(6)), since A is now block-tridiagonal, $E^t A^{-1} E$ is dense, and the cost of computing $\Delta\mu$ increases to $O(K^3)$ (this is also true for (10) when D is invertible but tridiagonal). Similarly, when L and D are invertible in (8) and (11), the bandwidth of the coefficient matrix in (20) is $O(K)$ and the cost of solving for $\Delta\mu$ by elimination may again increase to $O(K^3)$. For present purposes, the displacement method is therefore limited to systems (10) with block-diagonal invertible $Q_T^t D Q_T$.

A second approach, known as the "force method," may be useful for (10) when $Q_T^t D Q_T$ is not invertible or block-diagonal. This scheme first computes a particular solution x_p for

$$E^t x = b \tag{22}$$

and a matrix H whose columns supply a basis for null E^t. The associated formula

$$x = x_p + H\beta \tag{23}$$

then yields the complete solution of the lower half of (19) with free parameters β, and when this expression is substituted in the upper half of (19) it is seen that β must satisfy

$$AH\beta + Ey = a - Ax_p \tag{24}$$

and therefore

$$H^t A H\beta = H^t(a - Ax_p) \tag{25}$$

(cf. [13], [16], and [17]). For the system (10), A is $Q_T^t D Q_T$, E is $Q_T^t \nabla h$, and $H^t A H$ is positive-definite according to Theorem 2.1. In this case, (25) can be solved for β, (23) produces x, and y can be obtained by solving a corresponding block-tridiagonal positive-definite system of normal equations.

$$E^t E\, y = E^t(a - Ax). \tag{26}$$

The effectiveness of this approach turns on the existence of sparse bases for null E^t that are easily computed and yield block-banded matrices $H^t A H$ with fixed small block-bandwidths. Such bases are readily constructed for special instances of (10); however, the "force method" will not be pursued further here.

A third approach to (7), (8), (10), and (11) rests on the following basic observation: by permuting the equations and unknowns in a sparse system, it may be possible to concentrate all nonzero coefficients nearer the main diagonal; if this is so, the transformed system can be solved more efficiently by Gaussian elimination because of reduced fill-in [18], [19]. This technique has been used in the treatment of linear systems related to (8) [11], [13], [20], and Chapter 5 in [18] supplies another illustration for equilibrium systems (19) with block-diagonal A and block-tridiagonal E. The permutation schemes described in [11], [18] are modified here to suit the special structure of the coefficient matrix in (10), i.e.,

$$M = \begin{bmatrix} A & E \\ E^t & 0 \end{bmatrix}, \tag{27A}$$

where

$$A = Q_T^t D Q_T \tag{27B}$$

and

$$E = Q_T^t \nabla h. \tag{27C}$$

With reference to (15)–(17), the matrix A has ℓ block-rows and ℓ block-columns, E has ℓ block-rows and K block-columns, and thus M has $\ell+K$ block-rows and $\ell+K$ block-columns, with $K+1 \le 2\ell \le 2(K+1)$; moreover, A is either block-tridiagonal or block-diagonal, and

$$E_{i,j} \neq 0 \Rightarrow \max\{1, \sigma_i - 1\} \le j \le \min\{K, \sigma_i\} \tag{28A}$$

for $1 \le i \le \ell$ and $1 \le j \le K$, with

$$0 = \sigma_o < \cdots < \sigma_\ell \le K + 1 \tag{28B}$$

and

$$\sigma_{i-1} + 1 \le \sigma_i \le \sigma_{i-1} + 2. \tag{28C}$$

Note that the integers $\ell + \sigma_1 < \cdots < \ell + \sigma_{\ell-1}$ appear in sequence in the ordered K-tuple

$$(\ell + 1, \ldots, \ell + K), \tag{29}$$

that $\ell + \sigma_\ell$ also appears in (29) if $\sigma_\ell = K$, and that all remaining entries in (29) are integers of the form $\ell + \sigma_i - 1$ with $1 \le i \le \ell$ and $\sigma_i = \sigma_{i-1} + 2$. Now construct the ordered $(\ell + K)$-tuple

$$\phi = \{\phi_1, \ldots, \phi_{\ell+K}\} \tag{30A}$$

by inserting the integers $1, \ldots, \ell - 1$ immediately before $\ell + \sigma_1, \ldots, \ell + \sigma_{\ell-1}$ in (29), and inserting ℓ either before or after $\ell + K$ according to whether $\sigma_\ell = K$ or $K + 1$; equivalently, construct the entries in (30A) with the following recursion:

(30B)

```
set $i = 1$, $j = 1$, $k = 1$
do while $i \in \{1, \ldots, \ell + K\}$
    if $i = \sigma_j + j - 1$ then
        $\phi_i = j$
        $j = j + 1$
    else
        $\phi_i = \ell + k$
        $k = k + 1$
    end if
$i = i + 1$
```

In the special case $\ell = K + 1$, $\sigma_i = i$, the rule (30) produces the permutation

$$\phi = (1,\ \ell + 1,\ 2,\ \ell + 2, \ldots, \ell + K,\ \ell) \tag{31}$$

employed in Chapter 5 of [18]. In general, the $O(K)$ block-bandwidth of M is compressed to a fixed block-bandwidth between 3 and 7 when the block-rows and block-columns of M are permuted in accordance with (30).

THEOREM 3.1. *Let M and ϕ be defined by* (27) *and* (30). *Construct the corresponding partitioned matrix*

$$\widehat{M} = \begin{bmatrix} (\widehat{M})_{1,1} & \cdots & (\widehat{M})_{1,\ell+K} \\ \vdots & & \vdots \\ (\widehat{M})_{\ell+K,1} & \cdots & (\widehat{M})_{\ell+K,\ell+K} \end{bmatrix}$$

with

$$(\widehat{M})_{i,j} = (\widehat{M})_{\phi_i,\phi_j}$$

for $1 \leq i \leq \ell + K$ *and* $1 \leq j \leq \ell + K$. *Then:*

(i) $\widehat{M}$ *is block-tridiagonal if* $Q_T^t D Q_T$ *is block-diagonal.*

(ii) $\widehat{M}$ *is block-septadiagonal if* $Q_T^t D Q_T$ *is block-tridiagonal.*

(iii) $\widehat{M}$ *is block-pentadiagonal if* $Q_T^t D Q_T$ *is block-tridiagonal and* ϕ *is given by* (31).

Proof. By definition, $\widehat{M}$ is block-tridiagonal if and only if for $1 \leq i \leq \ell + K$, $1 \leq j \leq \ell + K$,

$$(M)_{\phi_i,\phi_j} \neq 0 \Rightarrow |j - i| \leq 1. \tag{32}$$

Since M is symmetric, it suffices to prove (32) for all i, j such that $\phi_i \leq \phi_j$. Suppose that $Q_T^t D Q_T$ is block-diagonal and $(M)_{\phi_i,\phi_j} \neq 0$ with $\phi_i \leq \phi_j$. Then by (27), (28) and (30),

$$\begin{aligned} &\phi_i \in \{1, \ldots, \ell\} \quad \text{and} \\ &(\phi_j = \phi_i \text{ or } \max\{\ell + 1,\ \ell + \sigma_{\phi_i} - 1\} \leq \phi \leq \min\{\ell + K,\ \ell + \sigma_{\phi_i}\}). \end{aligned} \tag{33}$$

If $\phi_i = 1$ and $\sigma_{\phi_i} = 1$, then (30) and (33) imply that $i = 1$, $\phi_j \in \{\phi_i, \ell + 1\} = \{\phi_i, \phi_{i+1}\}$, and therefore $|j - i| \leq 1$. If $1 \leq \phi_i \leq \ell$ and $2 \leq \sigma_{\phi_i} \leq K$, then (30) and (33) imply that $1 < i < \ell + K$, $\phi_j \in \{\phi_i, \ell + \sigma_{\phi_i} - 1, \ell + \sigma_{\phi_i}\} = \{\phi_i, \phi_{i-1}, \phi_{i+1}\}$, and therefore $|j - i| \leq 1$. Finally, if $\phi_i = \ell$ and $\sigma_{\phi_i} = K + 1$ then (30) and (33) imply that $i = \ell + K$, $\phi_j \in \{\phi_i, \ell + K\} = \{\phi_i, \phi_{i-1}\}$, and therefore $|j - i| \leq 1$. This proves assertion (i).

When $Q_T^t D Q_T$ is block-tridiagonal and $(M)_{\phi_i,\phi_j} \neq 0$ with $\phi_i \leq \phi_j$, condition (33) is replaced by

$$\phi_i \in \{1, \ldots, \ell\} \quad \text{and}$$

$$(\phi_i \leq \phi_j \leq \min\{\phi_i + 1, \ell\} \text{ or } \max\{\ell + 1,\ \ell + \sigma_{\phi_i} - 1\} \leq \phi_j \leq \min\{\ell + K,\ \ell + \sigma_{\phi_i}\}).$$

Moreover, consecutive integers in the sequence $1, \ldots, \ell$ are spaced no more than three entries apart in ϕ, in general, and no more than two entries apart when (31) holds. Assertions (ii) and (iii) follow at once from these observations and minor adjustments of the proof for (i). □

When the permutation (30) is applied to the coefficient matrix in (7), parts (ii) and (iii) of Theorem 3.1 hold with D replaced by L. Bandwidth compression schemes similar to (30) can also be formulated for the coefficient matrices in (8) and (11).

Thus, each of the $O(K)$ bandwidth systems (7), (8), (10), and (11) is similar under permutations to a fixed bandwidth system

$$\widehat{M}\hat{v} = \hat{w} \tag{34}$$

that can be solved in $O(K)$ flops by Gaussian elimination, and the required permutations and their inverses are readily assembled at the cost of $O(K)$ comparisons. More precise solution cost estimates can be obtained from a closer inspection of $\widehat{M}$. For example, when $Q_T^t D Q_T$ is block-diagonal, it can be seen that the matrix $\widehat{M}$ in Theorem 3.1 is a sum,

$$\widehat{M} = \widehat{M}^{(1)} + \cdots + \widehat{M}^{(\ell)}, \tag{35A}$$

of ℓ symmetric block-tridiagonal matrices with

$$(\widehat{M}^{(m)})_{i,j} = (M^{(m)})_{\phi_i,\phi_j}, \tag{35B}$$

$$(M^{(m)})_{i,j} = \begin{cases} (M)_{i,j}, & i = m, \quad i \le j \le \ell + K, \\ 0, & i \ne m, \quad i \le j \le \ell + K, \end{cases} \tag{35C}$$

and ϕ specified by (30). For $1 \le m \le \ell$ and $2 \le \sigma_m \le K$, the nonzero blocks in $\widehat{M}^{(m)}$ are confined to a cruciform pattern

$$\begin{array}{ccc} & \vdots & \\ 0 & (E)_{m,\sigma_m-1}{}^t & 0 \\ \cdots\ (E)_{m,\sigma_m-1} & (A)_{m,m} & (E)_{m,\sigma_m}\ \cdots \\ 0 & (E)_{m,\sigma_m}{}^t & 0 \\ & \vdots & \end{array} \tag{36}$$

with center in block-row ϕ_m^{-1} and block-column ϕ_m^{-1}, and $2 \le \phi_m^{-1} \le \ell + K - 1$. For $m = 1$, $\sigma_1 = 1$ (respectively, $m = \ell$, $\sigma_\ell = K + 1$), the matrix $(A)_{m,m}$ appears as the first (respectively, last) diagonal block in $\widehat{M}^{(m)}$ and the rest of the pattern in (36) is truncated in the obvious way. Moreover, for $2 \le m \le \ell$,

$$\phi_m^{-1} - \phi_{m-1}^{-1} = \sigma_m - \sigma_{m-1} + 1. \tag{37}$$

It is now evident that the half-bandwidth of the symmetric matrix $\widehat{M}$ does not exceed

$$\omega = \max_{1 \le m \le \ell} (\tau_m + \max\{p_{\sigma_m - 1}, p_{\sigma_m}\}) - 1,$$

and hence the flop count for solving the corresponding permutation (34) of (10) by row elimination with pivots on a sequential machine does not exceed

$$\begin{array}{ccc} 2\omega^2(\tau + p) & + & 3\omega(\tau + p) \\ \text{(factor)} & & \text{(solve)} \end{array} \tag{38}$$

(cf. [19]). In view of (35)–(37), it is also apparent that if $\sigma_m - \sigma_{m-1} = 2$ for one or more values of m, then (34) decomposes into two or more block-tridiagonal subsystems that can be solved in parallel; however, it is less obvious and more significant that in all cases, block-tridiagonal systems (34) can be solved on multiprocessor machines with significant speedups [8]. Similar gains have been achieved in parallel implementations of the displacement and force methods for equilibrium equations as well [17].

REFERENCES

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

[2] M. Aljazzaf, *Multiplier Methods with Partial Elimination of Constraints for Nonlinear Programming*, Ph.D. thesis, North Carolina State University, Raleigh, NC, 1990.

[3] J. C. Dunn, *Formal augmented Newtonian projection methods for continuous-time optimal control problems*, in Proc. 28th IEEE Conf. on Decision and Control, Tampa, FL, 1989.

[4] ———, *Scaled gradient projection methods for optimal control problems and other structured nonlinear programs*, in Proc. New Trends in Systems Theory Conference, Geneva, Switzerland, 1990.

[5] J. C. Dunn and D. P. Bertsekas, *Efficient dynamic programming implementations of Newton's method for unconstrained optimal control problems*, J. Optim. Theory Appl., 63 (1989), pp. 23–38.

[6] S. J. Wright, *Partitioned dynamic programming for optimal control*, SIAM J. Optim., 2 (1991), pp. 620–642. (See also preprint MCS-P173-0890, Argonne National Laboratory, Argonne, IL.)

[7] ———, *Solution of discrete-time optimal control problems on parallel computers*, Parallel Comput., 16 (1990), pp. 221–238.

[8] ———, *Parallel algorithms for banded linear systems*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 824–842. (See also preprint MCS-P64-0289, Argonne National Laboratory, Argonne, IL.)

[9] K. Ohno, *A new approach to differential dynamic programming for discrete-time systems*, IEEE Trans. Automat. Control, AC-23 (1978), pp. 37–47.

[10] J. F. A. de O. Pantoja and D.-Q. Mayne, *Sequential quadratic programming algorithm for discrete-time optimal control problems with control inequality constraints*, Internat. J. Control, 53 (1991), pp. 823–836.

[11] S. J. Wright, *Interior point methods for optimal control of discrete-time systems*, Preprint MCS-P266-0491, Argonne National Laboratory, Argonne, IL, 1991.

[12] R. H. Nickel, *A Sequential Quadratic Programming Algorithm for Solving Large Sparse Nonlinear Programs*, Ph.D. thesis, University of North Carolina at Chapel Hill, 1984.

[13] K. C. P. Machielsen, *Numerical Solution of Optimal Control Problems with State Constraints by Sequential Quadratic Programming in Function Space*, CWI Tract No. 53, Centrum voor Wiskunde en Informatica, Amsterdam, 1988.

[14] W. W. Hager, *Multiplier methods for nonlinear optimal control*, SIAM J. Numer. Anal., 27 (1990), pp. 1–20.

[15] R. Fletcher, *Practical Methods of Optimization, Vol.* 2, John Wiley and Sons, New York, 1981.

[16] G. Strang, *A framework for equilibrium equations*, SIAM Rev., 30 (1988), pp. 283–297.

[17] R. J. Plemmons and R. E. White, *Substructuring methods for computing the null space of equilibrium matrices*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 1–22.

[18] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.

[19] W. W. Hager, *Applied Numerical Linear Algebra*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[20] S. J. Wright, Private communication, May 1991.

[21] R. A. Tapia, *Diagonalized multiplier methods and quasi-Newton methods for constrained minimization*, J. Optim. Theory Appl., (1977), pp. 135–194.

[22] R. T. Rockafellar, *New applications of duality in convex programming*, in Proc. Fourth Conf. Probability, Brasov, Romania, 1971, pp. 73–81.

[23] ———, *The multiplier method of Hestenes and Powell applied to convex programming*, J. Optim. Theory Appl., 12 (1973), pp. 555–562.

SIAM J. OPTIMIZATION
Vol. 3, No. 3, pp. 503–515, August 1993

004

A NONINTERIOR CONTINUATION METHOD FOR QUADRATIC AND LINEAR PROGRAMMING*

BINTONG CHEN† AND PATRICK T. HARKER‡

Abstract. The noninterior point path-following algorithm presented by the authors in 1990 is specialized to the mixed linear complementarity problem and its special cases (quadratic and linear programming). The new algorithm is related to, but has several advantages over, the interior point path-following algorithms.

Key words. linear complementarity, quadratic programming, linear programming, continuation, interior point algorithms

AMS subject classifications. 90C05, 90C20

1. Introduction. In a series of papers [1], [2] Chen and Harker have developed a new continuation method for monotone variational inequality, linear, and nonlinear complementarity problems. The new algorithm is closely related to, but has several advantages over, the existing interior point path-following algorithms. Although it follows the same interior path, called the *path of centers* in the literature, the new algorithm possesses the following advantages:

- it can start from an infeasible point, and each of the intermediate iterates does not have to remain interior;
- it generates more efficient Newton directions at each iteration;
- it can reduce the continuation parameter with more flexibility;
- at each iteration the resulting equation can be solved inexactly, and line search procedures can be easily incorporated.

Preliminary numerical experiments [2] for the linear complementarity problem (LCP) demonstrate that the new algorithm is more efficient than the interior point algorithm of Koijma, Mizuno, and Yoshie [11] and is competitive with Lemke's method. The purpose of this paper is to extend the new continuation method to the mixed LCP (MLCP) and to quadratic and linear programs.

Many interior point algorithms have been developed following the revolutionary paper by Karmarkar [8]. Depending on the main mathematical tools used, these interior point algorithms are often called

- potential reduction algorithms,
- path-following algorithms,
- affine scaling algorithms,
- projective scaling algorithms.

A comprehensive survey of the interior point algorithms was given by Todd [24]. The algorithms that are most closely related to this paper are the path-following methods.

The interior point path-following algorithms for linear programming (LP) have been studied by Gonzaga [5]; Kojima, Mizuno, and Yoshie [10]; Monterio and Adler [19]; Nazareth [21], [22]; Renegar [23]; Vaidya [26]; and Ye [27]. The interior point

* Received by the editors December 13, 1990; accepted for publication March 25, 1992. This work was supported by the National Science Foundation Presidential Young Investigator Award ECE-8552773 and by a grant from the Burlington Northern Railroad.

† Department of Management and Systems, College of Business and Economics, Washington State University, Pullman, Washington 99164-4726.

‡ Decision Sciences Department, Wharton School, University of Pennsylvania, Philadelphia, Pennsylvania 19104-6366.

path-following algorithms for convex quadratic programs (QPs) have been studied by Goldfarb and Liu [4], Mehrotra and Sun [17], Monterio and Adler [20], and Ye [27]. These path-following algorithms were extended to solve LCPs by Kojima et al. [9]; Kojima, Mizuno, and Yoshie [11]; and Tseng [25], and to quadratic programming with quadratic constraints by Mehrotra and Sun [18].

Although they are theoretically very attractive, most path-following algorithms require each intermediate iterate to follow the path of centers very closely and, therefore, to take small step lengths. The success of the practical versions of the path-following algorithms [15], [12] relies on longer step sizes to reduce the number of iterations. However, all the interior point algorithms, as implied by their name, require that all the intermediate iterates stay interior, which sometimes restricts the choice of a longer step length. It is precisely this need to stay interior that the method proposed herein will overcome.

The paper is organized as follows. In §2 the continuation method developed in [1], [2] is extended to the MLCP defined by a P_0-matrix. Sections 3 and 4 specialize the MLCP algorithm for solving QP and LP, respectively. Several formulations of QP and LP as an MLCP or LCP are considered, and the advantages and disadvantages of each formulation are discussed; conclusions are drawn in §5.

2. Mixed linear complementarity problems.

Let $\mathbf{M}^1 \in \Re^{n\times n}$, $\mathbf{M}^2 \in \Re^{n\times m}$, $\mathbf{M}^3 \in \Re^{m\times n}$, $\mathbf{M}^4 \in \Re^{m\times m}$, $\mathbf{q}^1 \in \Re^n$, and $\mathbf{q}^2 \in \Re^m$ be matrices and vectors of appropriate dimensions. Define $\mathbf{q} = (\mathbf{q}^1, \mathbf{q}^2)^T$ and

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}^1 & \mathbf{M}^2 \\ \mathbf{M}^3 & \mathbf{M}^4 \end{pmatrix}. \tag{1}$$

Consider the MLCP, denoted by MLCP$(\mathbf{M}, \mathbf{q})$, which is the problem of finding an $(\mathbf{x}, \mathbf{y}) \in \Re^n \times \Re^m$ such that

$$\begin{gathered} \mathbf{w} = \mathbf{M}^1\mathbf{x} + \mathbf{M}^2\mathbf{y} + \mathbf{q}^1 \geq \mathbf{0}, \qquad \mathbf{x} \geq \mathbf{0}, \quad \mathbf{w}^T\mathbf{x} = 0, \\ \mathbf{M}^3\mathbf{x} + \mathbf{M}^4\mathbf{y} + \mathbf{q}^2 = \mathbf{0}. \end{gathered}$$

It is well known that both QP and LP are special cases of the above MLCP. Given a $\mu > 0$, define the perturbed MLCP (PMLCP), denoted by PMLCP$(\mathbf{M}, \mathbf{q}, \mu)$, as that of finding an $(\mathbf{x}, \mathbf{y}) \in \Re^n \times \Re^m$ such that

$$\mathbf{w} = \mathbf{M}^1\mathbf{x} + \mathbf{M}^2\mathbf{y} + \mathbf{q}^1 > \mathbf{0}, \qquad \mathbf{x} > \mathbf{0}, \quad w_i x_i = \mu, \quad i = 1, \ldots, n, \tag{2}$$

$$\mathbf{M}^3\mathbf{x} + \mathbf{M}^4\mathbf{y} + \mathbf{q}^2 = \mathbf{0}. \tag{3}$$

The following result is a straightforward extension of Theorem 1 in [1]:

THEOREM 2.1. *Assume that* $\mathbf{M}^1$ *has a positive diagonal. Then* $(\mathbf{x}, \mathbf{y})$ *is a solution of* MLCP$(\mathbf{M}, \mathbf{q})$ *if and only if it solves the following system of nonlinear equations, denoted by* $\mathbf{J}(\mathbf{x}, \mathbf{y}, \mu) = \mathbf{0}$:

$$\begin{gathered} (\mathbf{M}^1\mathbf{x} + \mathbf{M}^2\mathbf{y} + \mathbf{q})_i + m^1_{ii}x_i - \sqrt{[(\mathbf{M}^1\mathbf{x} + \mathbf{M}^2\mathbf{y} + \mathbf{q})_i - m^1_{ii}x_i]^2 + 4m^1_{ii}\mu} = 0 \quad \forall i, \\ \mathbf{M}^3\mathbf{x} + \mathbf{M}^4\mathbf{y} + \mathbf{q}^2 = \mathbf{0}. \end{gathered}$$

Let $(\mathbf{x}(\mu), \mathbf{y}(\mu))$ be a solution of $\mathbf{J}(\mathbf{x}, \mathbf{y}, \mu) = \mathbf{0}$. Define the set of paths or trajectories generated by the continuation method as

$$\mathbf{T} = \{(\mathbf{x}(\mu), \mathbf{y}(\mu), \mu) : 0 < \mu \leq \bar{\mu}\},$$

where $\bar{\mu} > 0$ is some positive number. For notational simplicity, if $\mathbf{T}$ consists of a single path, we call $\mathbf{T}$ the path or trajectory. The following theorem characterizes the set $\mathbf{T}$.

THEOREM 2.2. *If $\nabla\mathbf{J}(\mathbf{x}(\mu),\mathbf{y}(\mu),\mu)$ is nonsingular for all $0 < \mu < \bar{\mu}$, then $\mathbf{T}$ consists solely of continuously differentiable paths. If, in addition, $\mathbf{T}$ is bounded, then $(\mathbf{x}(\mu),\mathbf{y}(\mu))$ approaches a limit point $(\mathbf{x}(0),\mathbf{y}(0)) \in \mathbf{S}$ as μ approaches zero, where $\mathbf{S}$ is the solution set of the original MLCP.*

Proof. The first part of the theorem is a direct application of the Path Theorem in [3]; the proof of the second part of the theorem is similar to that of Theorem 5 in [1]. □

Therefore, if the conditions of Theorem 2.2 are satisfied and $\mathbf{T}$ is nonempty, various path-following algorithms (see [3] for details) can be used to trace the path to a solution of the original MLCP.

To explore the implications of the above theorem in more detail, let us define

$$r_i(\mathbf{x},\mathbf{y},\mu) = 1 - \frac{(\mathbf{M}^1\mathbf{x}+\mathbf{M}^2\mathbf{y}+\mathbf{q})_i - m^1_{ii}x_i}{\sqrt{[(\mathbf{M}^1\mathbf{x}+\mathbf{M}^2\mathbf{y}+\mathbf{q})_i - m^1_{ii}x_i]^2 + 4m^1_{ii}\mu}}$$

and let $\mathbf{D}^1 = \operatorname{diag}\{m^1_{ii}\}$ and $\mathbf{R}(\mathbf{x},\mathbf{y},\mu) = \operatorname{diag}\{r_i(\mathbf{x},\mathbf{y},\mu)\}$. Then, after some algebraic manipulation, we obtain

$$\nabla\mathbf{J}(\mathbf{x},\mathbf{y},\mu) = \begin{pmatrix} \mathbf{R}(\mathbf{x},\mathbf{y},\mu) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{M}^{1r}(\mathbf{x},\mathbf{y},\mu) & \mathbf{M}^2 \\ \mathbf{M}^3 & \mathbf{M}^4 \end{pmatrix}, \tag{4}$$

where

$$\mathbf{M}^{1r}(\mathbf{x},\mathbf{y},\mu) = \mathbf{M}^1 + \mathbf{D}^1(2\mathbf{R}^{-1}(\mathbf{x},\mathbf{y},\mu) - \mathbf{I}).$$

We now establish the condition for $\nabla\mathbf{J}(\mathbf{x},\mathbf{y},\mu)$ to be nonsingular. Let

$$\bar{\mathbf{M}} = \mathbf{M}^1 - \mathbf{M}^2(\mathbf{M}^4)^{-1}\mathbf{M}^3 \quad \text{and} \quad \bar{\mathbf{q}} = \mathbf{q}^1 - (\mathbf{M}^4)^{-1}\mathbf{q}^2.$$

To establish the nonsingularity result, we need the following lemma.

LEMMA 2.3. *Suppose that $\mathbf{M}$ defined in (1) is a positive semidefinite (P_0-) matrix and that $\mathbf{M}^4$ is nonsingular. Then $\bar{\mathbf{M}}$ is a positive semidefinite (P_0-) matrix.*

Proof. We shall first prove this result for the case in which $\mathbf{M}$ is positive semidefinite. Let $\mathbf{x}$ be an arbitrary vector in $\Re^n$, and let $\mathbf{y}$ be a vector in $\Re^m$ such that

$$\mathbf{M}^3\mathbf{x} + \mathbf{M}^4\mathbf{y} = \mathbf{0}$$

or, equivalently,

$$\mathbf{y} = -(\mathbf{M}^4)^{-1}\mathbf{M}^3\mathbf{x}.$$

Then,

$$\begin{aligned} \mathbf{x}^T\bar{\mathbf{M}}\mathbf{x} &= \mathbf{x}^T(\mathbf{M}^1 - \mathbf{M}^2(\mathbf{M}^4)^{-1}\mathbf{M}^3)\mathbf{x} \\ &= \mathbf{x}^T\mathbf{M}^1\mathbf{x} + \mathbf{x}^T\mathbf{M}^2\mathbf{y} \\ &= \mathbf{x}^T\mathbf{M}^1\mathbf{x} + \mathbf{x}^T\mathbf{M}^2\mathbf{y} + \mathbf{y}^T\mathbf{M}^3\mathbf{x} + \mathbf{y}^T\mathbf{M}^4\mathbf{y} \\ &= (\mathbf{x}^T,\mathbf{y}^T)\mathbf{M}(\mathbf{x}^T,\mathbf{y}^T)^T \\ &\geq 0. \end{aligned}$$

The first equality is from the definition of $\bar{\mathbf{M}}$. The second and third equalities are from the definition of $\mathbf{y}$ and the nonsingularity of $\mathbf{M}^4$. The last inequality is true because $\mathbf{M}$ is positive semidefinite by assumption.

To establish this result for the case in which $\mathbf{M}$ is a P_0-matrix, we only have to show that the determinants of all the principal minors of $\bar{\mathbf{M}}$ are nonnegative. Denote

$$N = \{1, \ldots, n\} \quad \text{and} \quad M = \{1, \ldots, m\}.$$

Let $I \subset N$, and let $\bar{\mathbf{M}}_{II}$ be the associated principal minor of $\bar{\mathbf{M}}$. Let $J = I \cup M$, and let $\mathbf{M}_{JJ}$ be the associated principal minor of $\mathbf{M}$. Then

$$\begin{aligned} |\bar{\mathbf{M}}_{II}| &= |\mathbf{M}^1_{II} - \mathbf{M}^2_{I\cdot}(\mathbf{M}^4)^{-1}\mathbf{M}^3_{\cdot I}| \\ &= \frac{|\mathbf{M}_{JJ}|}{|\mathbf{M}^4|} \geq 0. \end{aligned}$$

The first equality is true by definition. The second equality is true by matrix identity. The inequality holds because by assumption $\mathbf{M}^4$ is nonsingular and $\mathbf{M}$ is a P_0-matrix and thus so are $\mathbf{M}_{JJ}$ and $\mathbf{M}^4$. □

On the basis of the above lemma we have the following theorem.

THEOREM 2.4. *Suppose that* $\mathbf{M}$ *defined in* (1) *is a* P_0*-matrix,* $m^1_{ii} > 0$, $i = 1, \ldots, n$, *and that* $\mathbf{M}^4$ *is nonsingular. Then* $\nabla\mathbf{J}(\mathbf{x}, \mathbf{y}, \mu)$ *is nonsingular for all* $\mu > 0$ *and* $(\mathbf{x}, \mathbf{y}) \in \Re^n \times \Re^m$.

Proof. One can verify that $0 < r_i(\mathbf{x}, \mathbf{y}, \mu) < 2$ for all $\mu > 0$ and $i = 1, \ldots, n$. As a result, both $\mathbf{R}(\mathbf{x}, \mathbf{y}, \mu)$ and $(2\mathbf{R}^{-1}(\mathbf{x}, \mathbf{y}, \mu) - \mathbf{I})\mathbf{D}^1$ are positive diagonal matrices since $\mathbf{D}^1 > 0$ by assumption. Thus it suffices to show that the matrix on the right-hand side of (4) is nonsingular. Let $(\mathbf{x}, \mathbf{y})$ be a vector such that

$$\begin{pmatrix} \mathbf{M}^{1r}(\mathbf{x}, \mathbf{y}, \mu) & \mathbf{M}^2 \\ \mathbf{M}^3 & \mathbf{M}^4 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{0}.$$

Using the assumption that $\mathbf{M}^4$ is nonsingular and eliminating the variable $\mathbf{y}$ from the above equation, we obtain

$$[\mathbf{M}^{1r}(\mathbf{x}, \mathbf{y}, \mu) - \mathbf{M}^2(\mathbf{M}^4)^{-1}\mathbf{M}^3]\mathbf{x} = \mathbf{0},$$

or

$$[\bar{\mathbf{M}} + \mathbf{D}^1(2\mathbf{R}^{-1}(\mathbf{x}, \mathbf{y}, \mu) - \mathbf{I})]\mathbf{x} = \mathbf{0}.$$

Since $\bar{\mathbf{M}}$ is a P_0-matrix by Lemma 2.3, the matrix in front of $\mathbf{x}$ is a P-matrix and therefore is nonsingular. Hence $\mathbf{x} = \mathbf{0}$ and $\mathbf{y} = -(\mathbf{M}^4)^{-1}\mathbf{M}^3\mathbf{x} = \mathbf{0}$. This implies that the matrix on the right-hand side of (4) and thus the Jacobian $\nabla\mathbf{J}(\mathbf{x}, \mathbf{y}, \mu)$ are nonsingular. □

For the monotone LCP the existence and other properties of $\mathbf{T}$ have been shown by Megiddo [16]. The corollary below is a simple extension of Megiddo's result.

COROLLARY 2.5. *Suppose that* $\mathbf{M}$ *is positive semidefinite and that* $\mathbf{M}^4$ *is nonsingular. Then* PMLCP(μ) *has a unique and uniformly bounded solution for all* $0 < \mu \leq \bar{\mu} < \infty$ *if there exists an* $(\mathbf{x}, \mathbf{y})$ *such that*

$$\begin{aligned} &\mathbf{M}^1\mathbf{x} + \mathbf{M}^2\mathbf{y} + \mathbf{q}^1 > \mathbf{0}, \qquad \mathbf{x} > \mathbf{0}, \\ &\mathbf{M}^3\mathbf{x} + \mathbf{M}^4\mathbf{y} + \mathbf{q}^2 = \mathbf{0}. \end{aligned}$$

Corollary 2.5 provides a condition under which the path $\mathbf{T}$ exists and satisfies the condition of Theorem 2.2. Therefore, a solution of the MLCP can be obtained by a path-following algorithm. When $\mathbf{M}$ is a P_0-matrix, we can establish a similar result, which is an extension of Theorem 9 in [2]:

COROLLARY 2.6. *Suppose that the assumptions of Theorem* 2.4 *are satisfied and that, in addition, there exists an* $(\mathbf{x}^0, \mathbf{y}^0)$ *such that the set*

$$X_J = \{(\mathbf{x}, \mathbf{y}) : \|\mathbf{J}(\mathbf{x}, \mathbf{y}, \mu)\| \leq \|\mathbf{J}(\mathbf{x}^0, \mathbf{y}^0, \mu)\|\}$$

is uniformly bounded for all $0 < \mu \leq \bar{\mu}$ *and some* $\bar{\mu} > 0$. *Then* PMLCP(μ) *and thus* $\mathbf{J}(\mathbf{x}, \mathbf{y}, \mu) = \mathbf{0}$ *have unique and uniformly bounded solutions for all* $0 < \mu \leq \bar{\mu}$.

Proof. The proof is similar to that of Theorem 9 in [2]. □

The above two corollaries provide the conditions under which $\mathbf{T}$ is well behaved and leads to a solution of the MLCP. We now present the continuation method for the MLCP with a P_0-matrix.

Initiation Step Let ε be a given stopping tolerance. Choose an initial point $(\mathbf{x}^0, \mathbf{y}^0) \in \Re^n \times \Re^m$ and sequences $\mu^k > 0$ and $\epsilon^k > 0$, $k = 1, 2, \ldots$, such that

$$\mu^k \leq \mu^{k-1} \quad \text{and} \quad \lim_{k \to \infty} \mu^k = 0,$$
$$\epsilon^k \leq \epsilon^{k-1} \quad \text{and} \quad \lim_{k \to \infty} \epsilon^k = 0.$$

Set $k = 1$ and go to the Main Step.

Main Step

1. Starting with $(\mathbf{x}^{k-1}, \mathbf{y}^{k-1})$, find $(\mathbf{x}^k, \mathbf{y}^k)$ by solving $\mathbf{J}(\mathbf{x}, \mathbf{y}, \mu^k) = \mathbf{0}$ such that $\|\mathbf{J}(\mathbf{x}^k, \mathbf{y}^k, \mu^k)\| \leq \epsilon^k$.
2. If $\|\min\{\mathbf{x}^k, \mathbf{w}^k\}\| \leq \varepsilon$, where $\mathbf{w} = \mathbf{M}^1\mathbf{x} + \mathbf{M}^2\mathbf{y} + \mathbf{q}$ and "min" is taken componentwise, terminate; otherwise, go to Step 1.

3. Convex quadratic programming. This section specializes the algorithm for the MLCP to the case of convex QPs. Two types of QPs are considered: the strictly convex QP with general constraints and the convex QP in standard form.

Consider the following strictly convex QP with general constraints:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} \geq \mathbf{b}, \qquad \mathbf{E}\mathbf{x} = \mathbf{d}, \end{array}$$

where $\mathbf{Q} \in \Re^{n \times n}$ is a symmetric positive definite matrix and $\mathbf{A}, \mathbf{E}, \mathbf{c}, \mathbf{b}, \mathbf{d}$ are matrices and vectors of proper dimensions. Let $\mathbf{y}$ and $\mathbf{z}$ be the dual variables associated with constraints $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ and $\mathbf{E}\mathbf{x} = \mathbf{d}$, respectively. The Karush–Kuhn–Tucker (K–K–T) conditions for the QP are

$$\mathbf{Q}\mathbf{x} - \mathbf{A}^T\mathbf{y} - \mathbf{E}^T\mathbf{z} + \mathbf{c} = \mathbf{0}, \tag{5}$$
$$\mathbf{w} = \mathbf{A}\mathbf{x} - \mathbf{b} \geq \mathbf{0}, \qquad \mathbf{y} \geq \mathbf{0}, \quad \mathbf{w}^T\mathbf{y} = 0, \tag{6}$$
$$\mathbf{E}\mathbf{x} - \mathbf{d} = \mathbf{0}. \tag{7}$$

Notice that the above MLCP does not satisfy the assumptions of Theorem 2.1 since the diagonal of the corresponding matrix is not strictly positive. To avoid this difficulty we solve for $\mathbf{x}$ from (5) and substitute it into (6) and (7) to obtain

$$\mathbf{w} = \mathbf{A}\mathbf{Q}^{-1}\mathbf{A}^T\mathbf{y} + \mathbf{A}\mathbf{Q}^{-1}\mathbf{E}^T\mathbf{z} - \mathbf{A}\mathbf{Q}^{-1}\mathbf{c} - \mathbf{b} \geq \mathbf{0}, \qquad \mathbf{y} \geq \mathbf{0}, \quad \mathbf{w}^T\mathbf{y} = \mathbf{0},$$
$$\mathbf{E}\mathbf{Q}^{-1}\mathbf{A}^T\mathbf{y} + \mathbf{E}\mathbf{Q}^{-1}\mathbf{E}^T\mathbf{z} - \mathbf{E}\mathbf{Q}^{-1}\mathbf{c} - \mathbf{d} = \mathbf{0},$$

which is the MLCP defined by

$$\mathbf{M} = \begin{pmatrix} \mathbf{A}\mathbf{Q}^{-1}\mathbf{A}^T & \mathbf{A}\mathbf{Q}^{-1}\mathbf{E}^T \\ \mathbf{E}\mathbf{Q}^{-1}\mathbf{A}^T & \mathbf{E}\mathbf{Q}^{-1}\mathbf{E}^T \end{pmatrix}, \qquad \mathbf{q} = \begin{pmatrix} -\mathbf{A}\mathbf{Q}^{-1}\mathbf{c} - \mathbf{b} \\ -\mathbf{E}\mathbf{Q}^{-1}\mathbf{c} - \mathbf{d} \end{pmatrix}.$$

The above transformation is described in [7]. The following proposition shows that the above MLCP satisfies the requirements of the continuation method for the MLCP described in the previous section.

PROPOSITION 3.1. *Suppose that* $\mathbf{Q}$ *is a symmetric and positive definite matrix. Then* $\mathbf{M}$ *is positive semidefinite. If, in addition,* $\mathbf{A}$ *has no row identically equal to zero and* $\mathbf{E}$ *has full row rank, then* $\mathbf{M}$ *has a positive diagonal and* $\mathbf{E}\mathbf{Q}^{-1}\mathbf{E}^T$ *is positive definite.*

Proof. Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)^T$ be any vector. Then

$$\mathbf{x}^T\mathbf{M}\mathbf{x} = (\mathbf{A}\mathbf{x}_1 + \mathbf{E}\mathbf{x}_2)^T\mathbf{Q}^{-1}(\mathbf{A}\mathbf{x}_1 + \mathbf{E}\mathbf{x}_2) \geq 0.$$

The inequality is true because the inverse of a symmetric positive definite matrix is also a positive definite matrix. Therefore, $\mathbf{M}$ is positive semidefinite. Now, suppose $\mathbf{x}_2 \neq \mathbf{0}$. Then $\mathbf{E}^T\mathbf{x}_2 \neq \mathbf{0}$ since $\mathbf{E}$ has full row rank. Therefore,

$$\mathbf{x}_2^T\mathbf{E}\mathbf{Q}^{-1}\mathbf{E}^T\mathbf{x}_2 = (\mathbf{E}^T\mathbf{x}_2)^T\mathbf{Q}^{-1}(\mathbf{E}^T\mathbf{x}_2) > 0,$$

which implies that $\mathbf{E}\mathbf{Q}^{-1}\mathbf{E}^T$ is positive definite. It remains to be shown that $\mathbf{A}\mathbf{Q}^{-1}\mathbf{A}^T$ has a positive diagonal. Notice that the iith element of $\mathbf{A}\mathbf{Q}^{-1}\mathbf{A}^T$ equals $\mathbf{A}_{i.}\mathbf{Q}^{-1}\mathbf{A}_{.i}^T$, which is positive since $\mathbf{A}$ has no row identically equal to zero. □

Therefore, the algorithm for the MLCP can be applied to the above MLCP formulation of the QP immediately. However, the above formulation has a major disadvantage, namely, the matrix $\mathbf{Q}$ needs to be inverted. For large-scale problems with sparse data, the inversion not only is time consuming but also destroys the sparse structure of the problem. Therefore, the above formulation is more suitable for a QP with a separable objective function. We now describe an alternative formulation suitable for a nonseparable QP, proposed by Han and Mangasarian [6].

Let γ be a number such that $\gamma\mathbf{Q} - \mathbf{I}$ is positive definite. One can verify that the K–K–T conditions of the QP are equivalent to

$$\mathbf{w} = \gamma\mathbf{A}\mathbf{A}^T\mathbf{y} - \mathbf{A}(\gamma\mathbf{Q} - \mathbf{I})\mathbf{x} + \gamma\mathbf{A}\mathbf{E}^T\mathbf{z} - \gamma\mathbf{A}\mathbf{c} - \mathbf{b} \geq \mathbf{0}, \qquad \mathbf{y} \geq \mathbf{0}, \quad \mathbf{w}^T\mathbf{y} = 0,$$
$$(\gamma\mathbf{Q} - \mathbf{I})\mathbf{Q}\mathbf{x} - (\gamma\mathbf{Q} - \mathbf{I})\mathbf{A}^T\mathbf{y} - (\gamma\mathbf{Q} - \mathbf{I})\mathbf{E}^T\mathbf{z} + (\gamma\mathbf{Q} - \mathbf{I})\mathbf{c} = \mathbf{0},$$
$$\gamma\mathbf{E}\mathbf{A}^T\mathbf{y} - \mathbf{E}(\gamma\mathbf{Q} - \mathbf{I})\mathbf{x} + \gamma\mathbf{E}\mathbf{E}^T\mathbf{z} - \gamma\mathbf{E}\mathbf{c} - \mathbf{d} = \mathbf{0},$$

which are an MLCP defined by

$$\mathbf{M} = \begin{pmatrix} \gamma\mathbf{A}\mathbf{A}^T & -\mathbf{A}(\gamma\mathbf{Q} - \mathbf{I}) & \gamma\mathbf{A}\mathbf{E}^T \\ -(\gamma\mathbf{Q} - \mathbf{I})\mathbf{A}^T & (\gamma\mathbf{Q} - \mathbf{I})\mathbf{Q} & -(\gamma\mathbf{Q} - \mathbf{I})\mathbf{E}^T \\ \gamma\mathbf{E}\mathbf{A}^T & -\mathbf{E}(\gamma\mathbf{Q} - \mathbf{I}) & \gamma\mathbf{E}\mathbf{E}^T \end{pmatrix}, \qquad \mathbf{q} = \begin{pmatrix} -\gamma\mathbf{A}\mathbf{c} - \mathbf{b} \\ (\gamma\mathbf{Q} - \mathbf{I})\mathbf{c} \\ -\gamma\mathbf{E}\mathbf{c} - \mathbf{d} \end{pmatrix}.$$

The above MLCP also satisfies the requirements of the continuation method for the MLCP, as shown by the following proposition.

PROPOSITION 3.2. *Suppose that* $\mathbf{Q}$ *is symmetric and positive definite. Then* $\mathbf{M}$ *is positive semidefinite. If, in addition,* $\mathbf{A}$ *has no row identically equal to zero and* $\mathbf{E}$ *has full row rank, then* $\mathbf{M}$ *has a positive diagonal and the following submatrix of* $\mathbf{M}$ *is positive definite*:

$$\mathbf{M}_s = \begin{pmatrix} (\gamma\mathbf{Q}-\mathbf{I})\mathbf{Q} & -(\gamma\mathbf{Q}-\mathbf{I})\mathbf{E}^T \\ -\mathbf{E}(\gamma\mathbf{Q}-\mathbf{I}) & \gamma\mathbf{E}\mathbf{E}^T \end{pmatrix}.$$

Proof. That $\mathbf{M}$ is positive semidefinite and has a positive diagonal can be proved in a manner similar to the proof of Proposition 3.1. It remains to be shown that $\mathbf{M}_s$ is positive definite. Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)^T \neq \mathbf{0}$ be any vector. Then

$$\begin{aligned} \mathbf{x}^T\mathbf{M}_s\mathbf{x} &= \mathbf{x}_1^T(\gamma\mathbf{Q}-\mathbf{I})\mathbf{Q}\mathbf{x}_1 - 2\mathbf{x}_1^T(\gamma\mathbf{Q}-\mathbf{I})\mathbf{E}^T\mathbf{x}_2 + \mathbf{x}_2^T(\gamma\mathbf{E}\mathbf{E}^T)\mathbf{x}_2 \\ &= \frac{1}{\gamma}\|(\gamma\mathbf{Q}-\mathbf{I})\mathbf{x}_1 - \gamma\mathbf{E}^T\mathbf{x}_2\|^2 + \mathbf{x}_1^T(\gamma\mathbf{Q}-\mathbf{I})\mathbf{Q}\mathbf{x}_1 - \frac{1}{\gamma}\mathbf{x}_1^T(\gamma\mathbf{Q}-\mathbf{I})^2\mathbf{x}_1 \\ &\geq \frac{1}{\gamma}\mathbf{x}_1^T(\gamma\mathbf{Q}-\mathbf{I})\mathbf{x}_1 \\ &> 0. \end{aligned}$$

The last inequality follows from the definition of γ. □

Therefore, the algorithm for the MLCP can also be readily applied to this formulation of the QP. However, both MLCP formulations of the QP have a common disadvantage, namely, the dimensions of the resulting MLCPs are larger than that of the QP. As a result, a matrix of larger dimension must be inverted at each iteration of the continuation method. To avoid this difficulty we now restrict our study to the convex QP in standard form, where the algorithm is simplified by taking advantage of the special structure of this problem. The resulting algorithm is closely related to the interior point path-following algorithm for the QP [20]. However, it possesses all the advantages mentioned in §1.

Consider the following convex QP in standard form:

$$\begin{aligned} &\text{minimize} \quad \mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ &\text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{Q} \in \Re^{n\times n}$ is a symmetric and positive semidefinite matrix, $\mathbf{A} \in \Re^{m\times n}$, $\mathbf{c} \in \Re^n$, and $\mathbf{b} \in \Re^m$ are the matrix and vectors of appropriate dimensions. Let $\mathbf{y} \in \Re^m$ be the dual variable of constraint $\mathbf{A}\mathbf{x} = \mathbf{b}$. The K–K–T conditions of the above QP are

$$\mathbf{w} = \mathbf{Q}\mathbf{x} - \mathbf{A}^T\mathbf{y} + \mathbf{c} \geq \mathbf{0}, \qquad \mathbf{x} \geq \mathbf{0}, \quad \mathbf{w}^T\mathbf{x} = 0,$$
$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

which are an MLCP defined by

$$\mathbf{M} = \begin{pmatrix} \mathbf{Q} & -\mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{pmatrix} \quad \text{and} \quad \mathbf{q} = \begin{pmatrix} \mathbf{c} \\ -\mathbf{b} \end{pmatrix}.$$

Assume that $\mathbf{Q}$ has no row identically equal to zero and that $\mathbf{A}$ has full row rank. Then $Q_{ii} > 0$ for all i since $\mathbf{Q}$ is symmetric and positive semidefinite. By Theorem 2.1 the corresponding PMLCP(μ) is equivalent to the following system of nonlinear equations, denoted by $\mathbf{J}_Q(\mathbf{x}, \mathbf{y}, \mu) = \mathbf{0}$:

$$(\mathbf{Q}\mathbf{x} - \mathbf{A}^T\mathbf{y} + \mathbf{c})_i + Q_{ii}x_i - \sqrt{[(\mathbf{Q}\mathbf{x} - \mathbf{A}^T\mathbf{y} + \mathbf{c})_i - Q_{ii}x_i]^2 + 4Q_{ii}\mu} = 0 \quad \forall i,$$
$$\mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}.$$

Denote

$$r_i(\mathbf{x},\mathbf{y},\mu) = 1 - \frac{(\mathbf{Qx}-\mathbf{A}^T\mathbf{y}+\mathbf{c})_i - Q_{ii}x_i}{\sqrt{[(\mathbf{Qx}-\mathbf{A}^T\mathbf{y}+\mathbf{c})_i - Q_{ii}x_i]^2 + 4Q_{ii}\mu}}, \tag{8}$$

and let $\mathbf{D} = \text{diag}\{Q_{ii}\}$ and $\mathbf{R} = \text{diag}\{r_i(\mathbf{x},\mathbf{y},\mu)\}$. Then, after some algebraic manipulation, we obtain

$$\nabla \mathbf{J}_Q(\mathbf{x},\mathbf{y},\mu) = \begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{Q}^r & -\mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{pmatrix}, \tag{9}$$

where

$$\mathbf{Q}^r = \mathbf{Q} + \mathbf{D}(2\mathbf{R}^{-1} - \mathbf{I}).$$

We have the following theorem, which is similar to Theorem 2.4.

THEOREM 3.3. *Suppose that* $\mathbf{Q}$ *is symmetric and positive semidefinite and that* $\mathbf{A}$ *has full row rank. Then* $\nabla\mathbf{J}_Q(\mathbf{x},\mathbf{y},\mu)$ *is nonsingular for all* $\mu > 0$ *and* $(\mathbf{x},\mathbf{y}) \in \Re^n \times \Re^m$.

Proof. One can verify that $\mathbf{R}$ is a positive diagonal matrix and that $\mathbf{Q}^r$ is a positive definite matrix for all $\mathbf{x} \in \Re^n$, $\mathbf{y} \in \Re^m$, and $\mu > 0$. Thus the matrix on the right-hand side of (9) is positive definite because of its special skew symmetric structure and the assumption that $\mathbf{A}$ has full row rank. □

Define the strictly feasible primal and dual region of the QP by

$$\Omega_+ = \{(\mathbf{x},\mathbf{y}) \in \Re^n \times \Re^m : \mathbf{x} \geq \mathbf{0},\ \mathbf{w} \geq \mathbf{0},\ \mathbf{w} = \mathbf{Qx} - \mathbf{A}^T\mathbf{y} + \mathbf{c},\ \mathbf{Ax} = \mathbf{b}\}.$$

It has been shown [16], [19] that PMLCP(μ) associated with the QP or $\mathbf{J}_Q(\mathbf{x},\mathbf{y},\mu) = \mathbf{0}$ has a unique and uniformly bounded solution for all $0 < \mu \leq \bar{\mu}$ if Ω_+ is nonempty. This property together with Theorem 3.3 assures that the continuation method for the MLCP is well defined when applied to the above MLCP formulation of the QP.

The most time-consuming step in the continuation method for the QP is to solve the system of nonlinear equations $\mathbf{J}_Q(\mathbf{x},\mathbf{y},\mu) = \mathbf{0}$ at each iteration. Let $(\Delta\mathbf{x},\Delta\mathbf{y})$ be the Newton direction associated with the current point $(\mathbf{x},\mathbf{y})$ such that $\mathbf{Ax} = \mathbf{b}$. Straightforward algebraic calculation yields

$$\Delta\mathbf{x} = (\mathbf{Q}^r)^{-1}\{\mathbf{A}^T[\mathbf{A}(\mathbf{Q}^r)^{-1}\mathbf{A}^T]^{-1}\mathbf{A}(\mathbf{Q}^r)^{-1} - \mathbf{I}\}(\mathbf{R})^{-1}\tilde{\mathbf{J}_Q}, \tag{10}$$

$$\Delta\mathbf{y} = [\mathbf{A}(\mathbf{Q}^r)^{-1}\mathbf{A}^T]^{-1}\mathbf{A}(\mathbf{Q}^r)^{-1}(\mathbf{R})^{-1}\tilde{\mathbf{J}_Q}, \tag{11}$$

where $\tilde{\mathbf{J}_Q} \in \Re^n$ is a vector consisting of $J_{Qi}(\mathbf{x},\mathbf{y},\mu)$, $i = 1,\ldots,n$. Therefore, the matrix to be inverted at each iteration is

$$\mathbf{A}[\mathbf{Q} + \mathbf{D}(2\mathbf{R}^{-1} - \mathbf{I})]^{-1}\mathbf{A}^T, \tag{12}$$

which is positive definite for all $\mu > 0$. We now present the continuation method for the convex QP in standard form:

Initiation Step Let ε be a given stopping tolerance. Choose an initial continuation parameter $\mu^1 > 0$ and an $(\mathbf{x}^0,\mathbf{y}^0) \in \Re^n \times \Re^m$ such that $\mathbf{Ax}^0 = \mathbf{b}$. Set $k = 1$, and go to the Main Step.

Main Step

1. Starting with $(\mathbf{x}^{k-1},\mathbf{y}^{k-1})$, calculate the Newton direction $(\Delta\mathbf{x}^k,\Delta\mathbf{y}^k)$ by (10) and (11).

2. $\mathbf{x}^k = \mathbf{x}^{k-1} + \alpha^k \Delta \mathbf{x}^k$; $\mathbf{y}^k = \mathbf{y}^{k-1} + \alpha^k \Delta \mathbf{y}^k$, where α^k is a scalar obtained by some line search procedure based on $\|\mathbf{J}_Q(\mathbf{x}, \mathbf{y}, \mu^k)\|$.
3. If $\|\min\{\mathbf{x}^k, \mathbf{w}^k\}\| \le \varepsilon$, where $\mathbf{w} = \mathbf{Q}\mathbf{x} - \mathbf{A}^T\mathbf{y} + \mathbf{c}$ and "min" is taken componentwise, terminate; otherwise, choose an $\mu^{k+1} \le \mu^k$ based on the scale of $\|\mathbf{J}_Q(\mathbf{x}^k, \mathbf{y}^k, \mu^k)\|$ and $\|\min\{\mathbf{x}^k, \mathbf{w}^k\}\|$. Set $k = k + 1$ and go to Step 1.

Notice that the new continuation method is very flexible in choosing the parameters of the algorithm. In particular,

- $(\mathbf{x}^0, \mathbf{y}^0)$ need not be either primal or dual feasible—in practice, it could be any good approximate solution;
- μ^1 can be very small (except for numerical considerations), and μ^k can be reduced faster than it can in interior point algorithms;
- α^k is selected by a line search procedure, which will tend to accelerate convergence.

Both the interior point algorithm and the continuation method introduced above are penalty function methods by nature. As μ approaches zero, the matrices to be inverted in both methods become more and more ill conditioned. Recall that the matrix inverted at each iteration in the primal and dual interior point algorithm [20] is

$$\mathbf{A}(\mathbf{Q} + \mathbf{X}^{-1}\mathbf{W})^{-1}\mathbf{A}^T, \tag{13}$$

where $\mathbf{X}$, $\mathbf{W}$ are positive diagonal matrices with entries x_i and w_i, respectively. It is clear that (12) and (13) have the same structure. Actually, the following result shows that these two matrices are identical on the path of centers.

PROPOSITION 3.4. *Let* $(\mathbf{x}(\mu), \mathbf{y}(\mu))$ *be the solution of* PMLCP(μ) *or, equivalently, of equation* $\mathbf{J}_Q(\mathbf{x}, \mathbf{y}, \mu) = \mathbf{0}$. *Then*

$$\mathbf{Q} + \mathbf{D}(2\mathbf{R}^{-1}(\mathbf{x}(\mu), \mathbf{y}(\mu), \mu) - \mathbf{I}) = \mathbf{Q} + \mathbf{X}^{-1}(\mu)\mathbf{W}(\mu).$$

Proof. The proof is similar to that of Proposition 2 in [2]. □

Proposition 3.4 roughly demonstrates that the continuation method proposed here has the same degree of ill conditioning as does the interior point path-following algorithm as μ approaches zero. However, the continuation method possesses all the advantages mentioned in §1.

4. Linear program. This section specializes the algorithms discussed in the previous two sections to the case of LP. Two types of LPs are considered: the LP with inequality constraints only and the LP in standard form.

Consider the LP with general constraints, called LP$_g$:

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} \ge \mathbf{b}, \qquad \mathbf{E}\mathbf{x} = \mathbf{d}. \end{array}$$

Also consider the associated QP denoted by QP(ε):

$$\begin{array}{ll} \text{minimize} & \frac{\varepsilon}{2}\mathbf{x}^T\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} \ge \mathbf{b}, \qquad \mathbf{E}\mathbf{x} = \mathbf{d}, \end{array}$$

where $\varepsilon > 0$. The following theorem due to Mangasarian and Meyer [14] enables us to transform the LP to a QP so that the continuation method developed for the QP can be readily applied.

THEOREM 4.1. *Suppose* LP_g *has a solution. Then there exists a real positive number* $\bar{\varepsilon} > 0$ *such that for each* ε *in the interval* $(0, \bar{\varepsilon}]$ *the unique solution of* $\mathrm{QP}(\varepsilon)$ *is independent of* ε *and is also a solution of* LP_g.

Although the value of $\bar{\varepsilon}$ cannot be predetermined, numerical experiments [13] show that it has a magnitude of $O(m)$ or $O(n)$, where m and n are number of constraints and number of variables, respectively.

Consider the LP with only inequality constraints, called LP_1:

$$\begin{array}{ll} \text{minimize} & \mathbf{b}^T\mathbf{y} \\ \text{subject to} & \mathbf{A}^T\mathbf{y} \geq \mathbf{c}, \end{array}$$

where $\mathbf{A} \in \Re^{m \times n}$, $\mathbf{b} \in \Re^m$, and $\mathbf{c} \in \Re^n$ are the matrix and vectors of appropriate dimensions. This LP formulation is not very restrictive since, by setting $\mathbf{y}_1 = -\mathbf{y}$, we have

$$\begin{array}{ll} \text{maximize} & \mathbf{b}^T\mathbf{y}_1 \\ \text{subject to} & \mathbf{A}^T\mathbf{y}_1 \leq -\mathbf{c}, \end{array}$$

which is the dual of the following LP in standard form:

$$\begin{array}{ll} \text{minimize} & (-\mathbf{c}^T\mathbf{x}) \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b}, \qquad \mathbf{x} \geq \mathbf{0}. \end{array}$$

By Theorem 4.1, if LP_1 has a solution, there exists an $\bar{\varepsilon}$ such that the solution of the following QP, denoted by $\mathrm{QP}_1(\varepsilon)$, is a solution of LP_1 for all $0 \leq \varepsilon \leq \bar{\varepsilon}$:

$$\begin{array}{ll} \text{minimize} & \frac{\varepsilon}{2}\mathbf{y}^T\mathbf{y} + \mathbf{b}^T\mathbf{y} \\ \text{subject to} & \mathbf{A}^T\mathbf{y} \geq \mathbf{c}. \end{array}$$

Let $\mathbf{z} \in \Re^n$ be the dual variables of the constraint. The K–K–T conditions of $\mathrm{QP}_1(\varepsilon)$ are

$$\varepsilon\mathbf{y} - \mathbf{A}\mathbf{z} + \mathbf{b} = \mathbf{0} \tag{14}$$

$$\mathbf{w} = \mathbf{A}^T\mathbf{y} - \mathbf{c} \geq \mathbf{0}, \qquad \mathbf{z} \geq \mathbf{0}, \quad \mathbf{w}^T\mathbf{z} = 0. \tag{15}$$

Solving for $\mathbf{y}$ in (14) and substituting into (15), we obtain the following LCP:

$$\mathbf{w} = \mathbf{A}^T\mathbf{A}\mathbf{z} - \mathbf{A}^T\mathbf{b} - \varepsilon\mathbf{c} > \mathbf{0}, \qquad \mathbf{z} > \mathbf{0}, \quad \mathbf{w}^T\mathbf{x} = 0. \tag{16}$$

PROPOSITION 4.2. *If* $\mathbf{A}^T$ *has no row identically equal to zero, then* $\mathbf{A}^T\mathbf{A}$ *is a positive semidefinite matrix with a positive diagonal.*

Therefore, the continuation algorithm for the monotone LCP [2] applies here. Let $\mathbf{z}(\varepsilon)$ be the solution of (16). The primal solution of $\mathrm{QP}_1(\varepsilon)$ is calculated by

$$\mathbf{y}(\varepsilon) = \frac{1}{\varepsilon}(\mathbf{A}\mathbf{z}(\varepsilon) - \mathbf{b}),$$

which is also a solution of LP_1 if $0 < \varepsilon \leq \bar{\varepsilon}$.

Now let us consider the LP in standard form, denoted by LP_2:

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \end{array}$$

where $\mathbf{A} \in \Re^{m\times n}$, $\mathbf{b} \in \Re^m$, and $\mathbf{c} \in \Re^n$ are the matrix and vectors of appropriate dimensions. By Theorem 4.1, if LP_2 has a solution, then the solution of the following QP, denoted by $\mathrm{QP}_2(\varepsilon)$, is a solution of LP_2 for all $0 \le \varepsilon \le \bar{\varepsilon}$:

$$\begin{array}{ll} \text{minimize} & \frac{\varepsilon}{2}\mathbf{x}^T\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \ge \mathbf{0}. \end{array}$$

Notice that $\mathrm{QP}_2(\varepsilon)$ satisfies all the requirements of the continuation method for the QP described in §3 and therefore can be solved directly by the algorithm. Let $\mathbf{y}$ be the dual variable of constraint $\mathbf{Ax} = \mathbf{b}$. The equation $\mathbf{J}_Q(\mathbf{x},\mathbf{y},\mu) = \mathbf{0}$ is now simplified to $\mathbf{J}_L(\mathbf{x},\mathbf{y},\varepsilon,\mu) = \mathbf{0}$, defined as follows:

$$\varepsilon x_i + (\mathbf{c} - \mathbf{A}^T\mathbf{y})_i - \sqrt{(\mathbf{c} - \mathbf{A}^T\mathbf{y})_i^2 + \varepsilon\mu} = 0, \quad i = 1,\ldots,n,$$
$$\mathbf{Ax} - \mathbf{b} = \mathbf{0}.$$

Denote

$$r_i(\mathbf{y},\varepsilon,\mu) = 1 - \frac{(\mathbf{c} - \mathbf{A}^T\mathbf{y})_i}{\sqrt{(\mathbf{c} - \mathbf{A}^T\mathbf{y})_i^2 + 2\varepsilon\mu}},$$

and denote $\mathbf{R} = \mathrm{diag}\{r_i(\mathbf{y},\varepsilon,\mu)\}$. Then, the corresponding Newton direction at $(\mathbf{x},\mathbf{y})$ becomes

$$\Delta\mathbf{x} = \frac{1}{\varepsilon}\left\{\frac{1}{\varepsilon^2}\mathbf{R}\mathbf{A}^T[\mathbf{A}\mathbf{R}\mathbf{A}^T]^{-1}\mathbf{A} - \mathbf{I}\right\}\tilde{\mathbf{J}_L},$$
$$\Delta\mathbf{y} = \frac{1}{\varepsilon^2}[\mathbf{A}\mathbf{R}\mathbf{A}^T]^{-1}\mathbf{A}\tilde{\mathbf{J}_L},$$

where $\tilde{\mathbf{J}_L} \in \Re^n$ is a vector consisting of $J_{Li}(\mathbf{x},\mathbf{y},\mu)$, $i = 1,\ldots,n$. The matrix inverted at each iteration is $\mathbf{ARA}^T$, which is positive definite if $\mathbf{A}$ has full row rank since $0 < r_i(\mathbf{y},\varepsilon,\mu) < 2$ for all $\mathbf{y} \in \Re^m$. Notice that this matrix has the same dimension and structure as that in the primal–dual interior point algorithm [19]. Therefore, the computation time at each iteration of the new continuation method should be similar to that of the interior point algorithm. However, the new method possesses all the advantages discussed in §§1 and 3.

We now compare the continuation methods for the LP based on the above two different formulations. The computationally intensive part of both formulations is the inversion of matrices at each iteration. For the LP with only inequality constraints, the matrix inverted in the LCP formulation is of the following form (see [2]):

$$(\mathbf{A}^T\mathbf{A} + \mathbf{D}),$$

where $\mathbf{D} \in \Re^{n\times n}$ is a positive diagonal matrix that is updated at each iteration. If $\mathbf{A}$ is dense, it takes only n arithmetic operations at each iteration to form the above matrix since $\mathbf{A}^T\mathbf{A}$ must be calculated only once. It takes an additional $\frac{1}{6}n^3$ to invert the matrix. Therefore, the total number of arithmetic operations at each iteration for the first formulation is $\frac{1}{6}n^3$ if the lower-order operations are ignored. In the second formulation the matrix to be inverted is

$$\mathbf{ARA}^T.$$

It takes m^2n arithmetic operations to form the matrix and an additional $\frac{1}{6}m^3$ arithmetic operations to invert the matrix. Therefore, the total number of operations is $m^2n + \frac{1}{6}m^3$ at each iteration if the lower-order operations are ignored. As a consequence, if $n \leq 2.5m$, the first formulation requires fewer operations at each iteration. Otherwise, the second formulation takes fewer operations. Both formulations discussed here have advantages, as discussed in §1, over the primal–dual interior point algorithm. However, they also share a common disadvantage, namely, the parameter $\bar{\varepsilon}$ must be estimated beforehand.

5. Conclusion and future research. The paper extends the continuation method for the LCP to the MLCP and then specializes the algorithm to both the QP and LP. Various formulations of a QP and LP as an MLCP are considered, and advantages and disadvantages of each formulation are discussed. The new continuation methods are similar to the interior point algorithms in terms of arithmetic operations at each iteration. However, they are more flexible in choosing the initial point and in reducing the continuation parameters.

Future research will be directed toward establishing the computational complexity of the new methods and toward finding an optimal way of reducing the continuation parameters. Extensive computational tests of the LP against the primal–dual interior point algorithm will be performed in the future.

REFERENCES

[1] B. Chen and P. T. Harker, *A Non-Interior-Point Continuation Method for Monotone Variational Inequalities*, Working Paper 90-10-02, Decision Sciences Department, Wharton School, University of Pennsylvania, Philadelphia, 1990; Math. Programming, to appear.

[2] ———, *A Non-Interior-Point Continuation Method for Linear Complementarity Problems*, Working Paper 90-10-03, Decision Sciences Department, Wharton School, University of Pennsylvania, Philadelphia, 1990; SIAM J. Matrix Anal. Appl., to appear.

[3] C. B. Garcia and W. I. Zangwill, *Pathways to Solutions, Fixed Points and Equilibria*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[4] D. Goldfarb and S. Liu, *An $O(n^3L)$ Primal Interior Point Algorithm for Convex Quadratic Programming*, Report, Department of Industrial Engineering and Operations Research, Columbia University, New York, 1988.

[5] C. C. Gonzaga, *An Algorithm for Solving Linear Programming in $O(n^3L)$ Operations*, Tech. Report UCB/ERL M87/10, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA, 1987.

[6] S. P. Han and O. L. Mangasarian, *A dual differentiable exact penalty function*, Math. Programming, 25 (1983), pp. 293–306.

[7] 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.

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

[9] M. Kojima, N. Megiddo, T. Noma, and A. Yoshie, *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problem*, Tech. Report, Department of Information Sciences, Tokyo Institute of Technology, Tokyo, Japan, 1990.

[10] M. Kojima, S. Mizuno, and A. Yoshie, *A primal-dual interior point algorithm for linear programming*, in Progress in Mathematical Programming: Interior-Point and Related Methods, N. Megiddo, ed., Springer-Verlag, Berlin, 1989, pp. 29–47.

[11] ———, *A polynomial-time algorithm for a class of linear complementarity problems*, Math. Programming, 44 (1989), pp. 1–26.

[12] I. J. Lustig, R. E. Marsten, and D. F. Shanno, *Computational Experience with a Primal–Dual Interior Point Method for Linear Programming*, Tech. Report J-89-11, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1989.

[13] O. L. Mangasarian, *Iterative solution of linear programs*, SIAM J. Numer. Anal., 18 (1981), pp. 606–614.

[14] O. L. Mangasarian and R. R. Meyer, *Nonlinear perturbation of linear programs*, SIAM J. Control Optim., 17 (1979), pp. 745–752.

[15] K. A. McShane, C. L. Monma, and D. Shanno, *An implementation of a primal–dual interior point method for linear programming*, ORSA J. Comput., 1 (1989), pp. 70–83.

[16] N. Megiddo, ed., *Progress in Mathematical Programming: Interior-Point and Related Methods*, Springer-Verlag, Berlin, 1989.

[17] S. Mehrotra and J. Sun, *An Algorithm for Convex Quadratic Programming that Requires $O(n^{3.5}L)$ Arithmetic Operations*, Tech. Report 87-24, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL, 1987.

[18] ———, *A Method of Analytic Centers for Quadratically Constrained Convex Quadratic Programs*, Tech. Report 88-01, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL, 1988.

[19] R. D. C. Monterio and I. Adler, *Interior path following primal-dual algorithms. Part* I: *Linear programming*, Math. Programming, 44 (1989), pp. 27–43.

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

[21] J. L. Nazareth, *Homotopy techniques in linear programming*, Algorithmica, 1 (1986), pp. 529–535.

[22] ———, *The Homotopy Principle and Algorithms for Linear Programming*, Report, Department of Pure and Applied Mathematics, Washington State University, Pullman, WA, 1988.

[23] J. Renegar, *A polynomial-time algorithm, based on Newton's method, for linear programming*, Math. Programming, 40 (1988), pp. 59–93.

[24] M. J. Todd, *Recent Developments and New Directions in Linear Programming*, Tech. Report 827, School of Operations Research and Industrial Engineering, College of Engineering, Cornell University, Ithaca, NY, 1988.

[25] P. Tseng, *Complexity Analysis of a Linear Complementarity Algorithm Based on a Lyapunov Function*, Report, Center for Intelligent Control Systems, Massachusetts Institute of Technology, Cambridge, MA, 1989.

[26] P. M. Vaidya, *An Algorithm for Linear Programming Which Requires $O(((m+n)n^2+(m+n)^{1.5})L)$ Arithmetic Operations*, Tech. Report, AT&T Bell Laboratories, Murray Hill, NJ, 1987.

[27] Y. Ye, *Interior Algorithms for Linear, Quadratic, and Linearly Constrained Convex Programming*, Ph.D. thesis, Department of Engineering-Economics Systems, Stanford University, Stanford, CA, 1987.

SIAM J. OPTIMIZATION
Vol. 3, No. 3, pp. 516–537, August 1993

005

AN IMPLEMENTATION OF THE DUAL AFFINE SCALING ALGORITHM FOR MINIMUM-COST FLOW ON BIPARTITE UNCAPACITATED NETWORKS*

MAURICIO G. C. RESENDE† AND GERALDO VEIGA‡

Abstract. This paper describes an implementation of the dual affine scaling algorithm for linear programming specialized to solve minimum-cost flow problems on bipartite uncapacitated networks. This implementation uses a preconditioned conjugate gradient algorithm to solve the system of linear equations that determines the search direction at each iteration of the interior point algorithm. Two preconditioners are considered: a diagonal preconditioner and a preconditioner based on the incidence matrix of an approximate maximum weighted spanning tree of the network. Under dual nondegeneracy this spanning tree allows for early identification of the optimal solution. By applying an ϵ-perturbation to the cost vector, an optimal extreme-point primal solution is produced in the presence of dual degeneracy. The implementation is tested by solving several large instances of randomly generated assignment problems, comparing solution times with the network simplex code NETFLO and the relaxation algorithm code RELAX. This interior point algorithm greatly benefits from implementation in a parallel architecture. For the largest instances tested the interior point code was competitive with both the simplex and relaxation codes.

Key words. linear programming, interior point algorithm, network flows, assignment problem, conjugate gradient, preconditioning

AMS subject classifications. 65-05, 65F10, 65K05, 65Y05, 90C05, 90C06, 90C35

1. Introduction.

Consider a directed graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, with (i, j) denoting a directed edge from vertex i to vertex j. On the basis of this underlying graph we define a network by attaching certain numerical quantities to the vertices and edges. Let $b_i \geq 0$ represent the units of flow produced or consumed at each vertex $i \in V$. Associated with each edge $(i, j) \in E$ we define the quantities c_{ij}, l_{ij}, and u_{ij} representing, respectively, the unit cost, lower bound, and upper bound. The solution of a network flow problem, often referred to as the *flow*, is represented by the $|E|$-dimensional vector x, where each component x_{ij} stands for the flow in each edge. The minimum-cost network flow (MCNF) problem consists of finding a flow of minimum cost, subject to flow conservation constraints for all vertices, bounding for the flow on all edges. Often it is required that the optimal flow consist of only integer quantities.

The implementation we describe is restricted to problems for which G is bipartite with a vertex partition S, T and $l_{ij} = 0$ and $u_{ij} = \infty$. An edge (i, j) has vertex $i \in S$ and vertex $j \in T$. All the techniques described in this paper can be applied to the more general MCNF problem described in the previous paragraph. We limit our focus to this subclass because of implementation considerations. The following is an integer

* Received by the editors February 11, 1991; accepted for publication (in revised form) March 19, 1992.

† AT&T Bell Laboratories, Murray Hill, New Jersey 07974.

‡ Department of Industrial Engineering and Operations Research, University of California, Berkeley, California 94720.

programming formulation for this MCNF problem:

$$
\begin{aligned}
&\text{minimize} && \sum_{ij\in E} c_{ij}x_{ij} && \\
&\text{subject to} && \sum_{ik\in E} x_{ik} = b_i, && i \in S, \\
(1.1)\quad & && \sum_{kj\in E} x_{kj} = b_j, && j \in T, \\
& && x_{ij} \geq 0 \ \text{integer}, && (i,j) \in E.
\end{aligned}
$$

We assume that

$$\sum_{i\in S} b_i = \sum_{j\in T} b_j.$$

Furthermore, if we assume that G is connected, (1.1) has a single redundant constraint that we remove from the formulation. In the resulting constraint matrix there is one-to-one correspondence between basic sequences and spanning trees of G. Finally, we assume that the data in b and c is integer. Since the constraint matrix in (1.1) is totally unimodular, all basic solutions of (1.1) are integer, the integrality constraint in (1.1) can be dropped, and the problem can be solved with a linear programming algorithm that produces a vertex solution.

Variations of the simplex method [11] can be customized to solve the MCNF problem. Mature implementations of these algorithms are widely used to solve large-scale problems. However, the combinatorial nature of the simplex method variants results in a rapid growth of the number of iterations as the problem dimensions grow. Instances with a few thousand vertices often require several million simplex iterations. Furthermore, primal degeneracy is often present in certain classes of network flow problems, causing most simplex iterations to be degenerate.

The main motivation of this study is that in practice the number of iterations in interior point algorithms for linear programming appears to grow slowly with problem size. Most direct comparisons between interior point algorithms and the simplex method (e.g., [1], [27]) conclude that as problem size increases the advantage increasingly tilts toward interior point methods. One may conjecture, then, that for large enough problems, interior point algorithms will also outperform simplex-based methods for network flow problems despite the fact that the interior point algorithms are implemented in double-precision arithmetic whereas network simplex codes are implemented in integer arithmetic. Furthermore, interior point methods do not appear to be affected by degeneracy as much as is the simplex method [29].

The dual affine scaling (DAS) algorithm [12] was among the first of the interior point methods to be shown to be a competitive alternative to the simplex method. Adler, Karmarkar, Resende, and Veiga [1] described an implementation of the DAS algorithm and compared their implementation with the simplex code MINOS 4.1 [28]. Data structures and programming techniques used in that implementation are described in [2]. Let A be an $m \times n$ matrix, let c and x be n-dimensional vectors, and let b be an m-dimensional vector. The DAS algorithm solves the linear program

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

indirectly by solving its dual

$$\begin{aligned} &\text{maximize} && b^\top y \\ &\text{subject to} && A^\top y + s = c, \qquad s \geq 0, \end{aligned} \tag{1.3}$$

where s is an n-dimensional vector of slacks and y is an m-dimensional vector. The algorithm starts with an initial interior solution

$$y^0 \in \{y \mid s = c - A^\top y > 0\}$$

and obtains iterate y^{k+1} from y^k according to

$$y^{k+1} = y^k + \alpha\,(AD_k^2A^\top)^{-1}b,$$

where

$$D_k = \text{diag}(1/s_1, \ldots, 1/s_n)$$

and α is such that $s^{k+1} = c - A^\top y^{k+1} > 0$. At each iteration a tentative primal solution [30] is given by

$$x^k = D_k^2A^\top(AD_k^2A^\top)^{-1}b. \tag{1.4}$$

It is easy to verify that $Ax^k = b$. However, x^k can only be guaranteed feasible (i.e., $x^k \geq 0$) at an optimal dual solution if dual nondegeneracy is assumed [12], [15]. Adler and Monteiro [3] have shown that for the continuous version of the DAS algorithm no nondegeneracy assumption is needed for the convergence of both the iterates and the primal estimates.

The bulk of the work in the DAS algorithm is related to building and updating the matrix $AD_k^2A^\top$ and solving the system of linear equations

$$(AD_k^2A^\top)d_y = b \tag{1.5}$$

that determines the ascent direction at each iteration of the algorithm. Adler et al. consider two approaches for solving (1.5). The first approach is Cholesky factorization, in which the matrix $AD_k^2A^\top$ is factored into an upper triangular matrix $L^\top$ and a lower triangular matrix L and the system

$$LL^\top d_y = b$$

is solved by first applying forward substitution to

$$Lz = b$$

and then applying back substitution to

$$L^\top d_y = z.$$

This approach is considered satisfactory when the number of nonzero elements in the factors is small. This may not be the case when there is large fill-in (fill-in is the difference in number of nonzeros between $AD_k^2A^\top$ and the $LL^\top$ factors) or when the problem is large and even without much fill-in when the number of nonzeros in the factors is large. Even though the constraint matrix of the MCNF problem is sparse, the factorization of $AD^2A^\top$ can produce considerable fill-in.

In [1], [2] Adler et al. use the preconditioned conjugate gradient algorithm [16], [25] when there are one or more dense columns in the A matrix, consequently making L and $L^\top$ dense. In this approach the dense columns of the matrix A are dropped, resulting in $\tilde{A}$, and the incomplete Cholesky factors of $\tilde{A}D^2\tilde{A}^\top$ are used as preconditioners. Karmarkar and Ramakrishnan [18] also use the conjugate gradient algorithm with the DAS algorithm for solving large linear programs. In these large linear programs the number of nonzero elements in the factors is large regardless of fill-in and direct factorization methods are too slow. They compare their implementation with MINOS 5.1 on a class of randomly generated minimum-cost network flow problems and suggest that it would be interesting to compare a special-purpose interior point implementation with a special purpose network simplex code. Mehrotra [23] has developed a code based on [18] and has solved numerous *netlib* [13] problems.

Several studies have concluded that interior point methods are not competitive with network simplex codes for solving network flow problems (e.g., [4], [5], [9]). In this paper we show several examples of large-scale network flow problems for which an interior point method is competitive with mature network flow codes. We describe and test a special-purpose implementation of the DAS algorithm for MCNF problems built on the general-purpose implementation of the DAS algorithm described by Adler et al. [1], [2].

The outline of the paper is as follows. In §2 we describe our implementation of the preconditioned conjugate gradient algorithm, including conjugate gradient stopping criteria. In §3 we consider preconditioners used for the MCNF problem. We describe a diagonal preconditioner and a spanning tree preconditioner. In §4 we consider early stopping of the DAS algorithm, identifying a primal optimal basis. Early stopping often avoids problems experienced by the conjugate gradient algorithm when the iterate of DAS algorithm is close to a face. In §5 we discuss another way to avoid numerical problems by dropping dual constraints. In §6 we consider dual degeneracy and present a perturbation scheme that allows the primal iterates of the DAS algorithm to converge to a vertex. In §7 we consider a parallel implementation of our algorithm. In §8 we test our implementation on large randomly generated assignment problems and compare our results with the network simplex code NETFLO [19] and the relaxation method code RELAX [8]. Concluding remarks are in §9.

2. Computing the ascent direction. The implementation of the DAS algorithm for network flow problems described in this paper is based on a preconditioned conjugate gradient algorithm for solving the direction-finding system at each iteration. Our algorithm differs slightly from the preconditioned conjugate gradient algorithm described by Adler et al. [1], [2]. Here, the preconditioned conjugate gradient algorithm consists of solving

$$M^{-1}(AD_k^2A^\top)d_y = M^{-1}b, \tag{2.1}$$

where M is a positive definite matrix. The objective is to make the preconditioned matrix

$$M^{-1}(AD_k^2A^\top)$$

less ill conditioned than $AD_k^2A^\top$, improving the convergence of the conjugate gradient algorithm.

```
procedure pcg($A, D_k, b, \epsilon_{cg}$)
1   $d_y^0 := b$;
2   $r_0 := (AD_k^2A^\top)d_y^0 - b$;
3   $z_0 := M^{-1}r_0$;
4   $p_0 := z_0$;
5   $i := 0$;
6   do $z_i^\top r_i \geq \epsilon_{cg} \rightarrow$
7       $q_i := (AD_k^2A^\top)p_i$;
8       $\alpha_i := z_i^\top r_i / p_i^\top q_i$;
9       $d_y^{i+1} := d_y^i + \alpha_i p_i$;
10      $r_{i+1} := r_i - \alpha_i q_i$;
11      $z_{i+1} := M^{-1}r_{i+1}$;
12      $\beta_i := z_{i+1}^\top r_{i+1} / z_i^\top r_i$;
13      $p_{i+1} := z_{i+1} + \beta_i p_i$;
14      $i := i + 1$
15  od
end pcg;
```

FIG. 2.1. *Preconditioned conjugate gradient algorithm.*

Let $\langle x, y \rangle_M = x^\top My$ be the inner product of x and y with respect to M, a positive definite matrix. Since $M^{-1}(AD_k^2A^\top)$ is symmetric with respect to M, i.e.,

$$\langle x, M^{-1}(AD_k^2A^\top)y \rangle_M = \langle M^{-1}(AD_k^2A^\top)x, y \rangle_M,$$

we can solve (2.1) with the standard conjugate gradient algorithm (see [14]) with all inner products (and norms) replaced by

$$\langle x, y \rangle_M = x^\top My,$$

resulting in the algorithm described by the pseudocode in Fig. 2.1.

The computationally intensive steps in the preconditioned conjugate gradient algorithm are lines 2, 3, 7, and 11 in Fig. 2.1. These lines correspond to matrix–vector multiplications (2 and 7) and systems of linear equations (3 and 11). Lines 2 and 3 are computed once, and lines 7 and 11 are computed once every conjugate gradient iteration. The multiplications carried out are of the form

$$(AD_k^2A^\top)z_0,$$

where z_0 is the vector d_y^0 in line 2 and p_i in line 7. There is some computational advantage in decomposing this matrix–vector multiplication into three steps:

(i) $z_1 = A^\top z_0$,
(ii) $z_2 = D_k^2 z_1$,
(iii) $z_3 = Az_2$.

Forming $AD_k^2A^\top$ explicitly requires $3|E|$ multiplications and $2|E|$ additions. Performing the matrix–vector multiplication requires $|V| + 2|E|$ multiplications and $|V| + 2|E|$ additions. If decomposition is performed, the matrix–vector multiplication requires $|E|$ multiplications and $3|E|$ additions. A further enhancement to this computation is that it can be carried out in parallel, which is the subject of §7.

The preconditioned residual computed in lines 3 and 11 of Fig. 2.1 amounts to solving the system of linear equations

$$Mz_{i+1} = r_{i+1},$$

where M is such that the system can be easily solved.

The usual stopping criterion for the conjugate gradient algorithm is to terminate when the 2-norm of the residue $\|r_i\|_2^2 = \|(AD_k^2A^\top)d_y^i - b\|_2^2$ is less than a given tolerance ϵ_{cg}. In our implementation we use the suggestion made in [18] and compute the angle θ between $(AD_k^2A^\top)d_y^i$ and b and stop when $|1-\cos\theta| < \epsilon_{\cos}$, where $\epsilon_{\cos}$ is some small tolerance (typically, $\epsilon_{\cos} = 10^{-8}$). The computation of

$$\cos\theta = \frac{|b^\top(AD_k^2A^\top)d_y^i|}{\|b\| \cdot \|(AD_k^2A^\top)d_y^i\|}$$

has the complexity of one conjugate gradient iteration and therefore is not carried out at each conjugate gradient iteration. In this implementation it is computed every 20 iterations of the conjugate gradient algorithm. This stopping criterion effectively halts the conjugate gradient algorithm when a good enough direction is on hand.

3. Preconditioners. Diagonal preconditioners were perhaps the first preconditioners used with the conjugate gradient algorithm [14]. They are simple to compute and lead to $\mathcal{O}(|E|)$ multiplications in steps 3 and 11 of the preconditioned conjugate gradient algorithm of Fig. 2.1. In practice they are effective on MCNF problems during the initial iterations of the DAS algorithm. However, in some instances of MCNF problems, as the DAS iterations progress they tend to lose their effectiveness. Yeh [34] used diagonal preconditioning exclusively in her implementation of the DAS algorithm with conjugate gradient for the assignment problem.

The diagonal preconditioner used in our implementation is

$$M = \text{diag}(AD_k^2A^\top).$$

This preconditioner can be computed in $\mathcal{O}(|E|)$ additions and multiplications. The preconditioned residue systems

$$Mz_{i+1} = r_{i+1}, \qquad i = 0, 1, \ldots,$$

of lines 3 and 11 of the pseudocode of Fig. 2.1 can each be solved in $\mathcal{O}(|V|)$ divisions.

Karmarkar and Ramakrishnan [17] and Vaidya [32] have suggested using a minimum weighted spanning tree preconditioner for network flow problems. Our spanning tree preconditioner is based on those suggestions.

Let $\mathcal{S}_k$ be the submatrix of A corresponding to a maximum weighted spanning tree of G with appropriately defined weights. The spanning tree preconditioner is

$$M = \mathcal{S}_k\mathcal{D}_k^2\mathcal{S}_k^\top,$$

where

$$\mathcal{D}_k = \text{diag}(1/s_{T_1}, \ldots, 1/s_{T_m})$$

and $T_1, \ldots, T_m$ are the edge indices of the spanning tree. The edge weight vector can be the primal estimates [30]

$$w = D_k^2A^\top(AD_k^2A^\top)^{-1}b$$

or the reciprocal estimates [18]

$$w = D_k^2e, \tag{3.1}$$

where e is an $|E|$-vector of all ones. It is well known that in the absence of degeneracy both of the above weight vectors can be used to identify a primal basic sequence of the MCNF problem. For primal degenerate but dual nondegenerate problems only weights (3.1) can be used to identify a primal basic sequence. As the reader will see in §6 we use a random perturbation scheme to avoid dual degeneracy and therefore use the reciprocal estimates (3.1) as weights for the tree preconditioner.

Instead of computing a maximum weighted spanning tree, we find an approximate maximum weighted spanning tree. To do so we use a variant of Kruskal's greedy algorithm [21], in which we carry out an approximate bucket sort in place of the usual exact sorting of the weights. Let $w_{\min}$ and $w_{\max}$ be the values of the minimum and maximum weights, respectively. We partition the $[w_{\min}, w_{\max}]$ interval into a given number of subintervals and classify each edge into a bucket. With this approximate scheme the time to construct a preconditioner is small compared to the time required by the iterations of the conjugate gradient algorithm.

At each conjugate gradient iteration the preconditioned residue system

$$M_k z_{i+1} = r_{i+1} \tag{3.2}$$

must be solved. Substitution of the spanning tree preconditioner into (3.2) yields

$$(\mathcal{S}_k \mathcal{D}_k^2 \mathcal{S}_k^\top) z_{i+1} = r_{i+1}. \tag{3.3}$$

The matrix $\mathcal{S}_k$ of a spanning tree can be permuted to a triangular form $\tilde{\mathcal{S}}_k$ in $\mathcal{O}(|V|)$ time. Consequently, (3.3) becomes

$$(\tilde{\mathcal{S}}_k \mathcal{D}_k^2 \tilde{\mathcal{S}}_k^\top) z_{i+1} = r_{i+1}, \tag{3.4}$$

which can be solved in $\mathcal{O}(|V|)$ time, first by forward substitution,

$$\tilde{\mathcal{S}}_k t_1 = r_{i+1},$$

then by solving the diagonal system

$$\mathcal{D}_k^2 t_2 = t_1,$$

and finally by back substitution,

$$\tilde{\mathcal{S}}_k^\top z_{i+1} = t_2.$$

Since $\mathcal{S}_k$ is made up of only $+1$ entries, back and forward substitutions involve only additions. In the general MCNF problem $\mathcal{S}_k$ corresponds to the incidence matrix of a directed spanning tree.

We have observed that in many instances the diagonal preconditioner is effective during the initial iterations of the DAS algorithm but that as the DAS iterations progress it often tends to lose its effectiveness. On the other hand, during the initial iterations of the DAS algorithm the dual slacks provide little information and consequently the spanning tree preconditioner is not as effective as the diagonal preconditioner. As the DAS iterations progress the spanning tree preconditioner becomes increasingly effective. In this implementation we use both preconditioners. We begin with the diagonal and monitor the number of iterations required by the conjugate gradient algorithm. When the number of conjugate gradient iterations surpasses some specified limit, the code switches over to the spanning tree preconditioner. The spanning tree preconditioner is used from that point on and the code never returns to

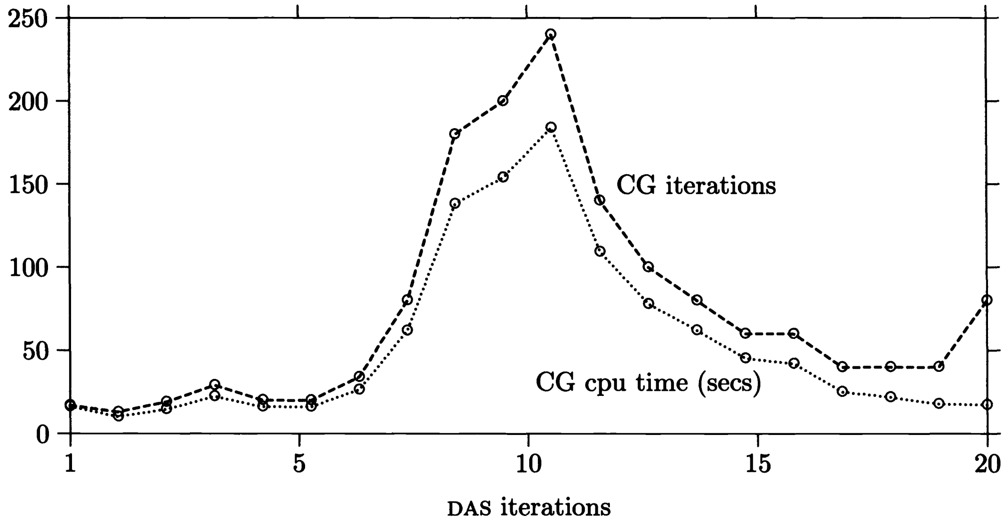


FIG. 3.1. *Conjugate gradient* (CG) *iterations and* CPU *time for a* 1000 × 1000 *assignment problem with* 95, 612 *edges.*

diagonal preconditioning (even though diagonal preconditioning may become effective again near the final iterations of the DAS algorithm). It may also be the case that we never switch from diagonal to spanning tree preconditioner. This has been observed in many solutions. Figure 3.1 illustrates this switch on a 1000 × 1000 assignment problem with $|E| = 95,612$. In this example the change of preconditioner is triggered when the conjugate gradient algorithm exceeds 200 iterations for the first time.

4. Identifying an optimal primal basis. Practical experience with direct factorization implementations of interior point methods has shown that these implementations are numerically quite stable, even toward the final iterations, when the condition number of $AD_k^2A^\top$ may become very large. Iterative methods, like the conjugate gradient method, are sensitive to ill conditioning of the matrix. The effect observed on the conjugate gradient algorithm is a substantial increase in the number of iterations.

One remedy for this problem is not to allow the DAS iterations to proceed to the point where ill conditioning occurs. To do this, one must be able to identify a primal optimal basis early on in the DAS iterations to stop the iterations of the interior point algorithm with an optimal solution at hand. In the following discussion let us assume that there exists a unique optimal primal solution. In §6 we consider the case of dual degenerate MCNF problems.

Under dual nondegeneracy the DAS algorithm converges to an optimal dual solution with the primal solution estimate converging to the unique optimal primal solution [15]. For early detection of the optimal primal solution, at each iteration we attempt to build a primal basis that includes all edges with nonzero flow in the primal optimal solution. One approach is to identify a basis from the dual solution at each iteration. We compute an approximate maximum weight spanning tree based on the reciprocals of the dual slacks. The reciprocals of the dual slacks are an alternative method of estimating the primal solution in dual interior point methods.

Let $\mathcal{S}_k$ be the spanning tree selected at iteration k. Let x^* be the solution to

$$\mathcal{S}_k x = b. \tag{4.1}$$

Linear system (4.1) can be solved with integer arithmetic in $\mathcal{O}(|V|)$ operations. If $x^* \geq 0$, then x^* is a primal feasible (integer) solution to the MCNF problem. If $x^* \ngeq 0$, then the spanning tree has not correctly picked out a feasible basis and additional DAS iterations are required.

If an initial interior dual feasible solution is assumed to be available, optimality of a feasible flow x^* can be confirmed by simply checking the dual gap between the current interior dual iterate y^k and the tentative primal basic feasible solution x^*. If

$$c^\top x^* - b^\top y^k < 1,$$

then x^* is an optimal flow. One can do somewhat better by taking as the bound $b^\top y^{\max}$, where $y^{\max}$ is obtained by following the line segment that passes through y^{k-1} and y^k all the way to the boundary of the polytope, i.e., by taking

$$y^{\max} = y^k + \alpha d_y,$$

where

$$\alpha = \min\{-s_i^k/(d_s)_i \mid (d_s)_i < 0,\ i = 1, \ldots, |E|\}.$$

If $x^* \geq 0$ and $c^\top x^* - b^\top y^{\max} < 1$, then x^* is an optimal (integer) primal basic solution and DAS is halted. If an initial interior dual solution is not readily available, a big-M scheme as described in [1] can be used.

Optimality testing is not carried out until the final few iterations of the DAS algorithm. In this implementation we compute the primal flow x only when the dual convergence criterion

$$|b^\top y^{k+1} - b^\top y^k|/|b^\top y^{k+1}| < 10^{-2}$$

is satisfied.

Let $c_{\mathcal{S}_k}$ be the cost subvector corresponding to the spanning tree basis $\mathcal{S}_k$. No guarantee can be made as to whether $y^* = c_{\mathcal{S}_k}\mathcal{S}_k^{-1}$ is a feasible dual solution. Hence our implementation does not provide an optimal primal–dual pair, but rather it provides an optimal integer primal solution. One way to proceed if a dual optimal vertex is needed is to jump to the network simplex algorithm, starting with the current optimal primal vertex. Yeh [34] has reported that for the few instances tested, few network simplex iterations were needed to find a primal–dual optimal pair by using this post processing idea.

Another approach for obtaining an optimal vertex from an interior solution is described in [26]. For a linear programming problem with a unimodular coefficient matrix, under a dual nondegeneracy assumption a fractional solution x' can be rounded off to the optimal integer solution if $c^\top x' - v^* < \frac{1}{2}$, where v^* is the optimal objective value. An algorithm that generates feasible primal solutions and bounds on the optimal objective value can use this property for early termination. For example, primal-dual variants of interior point methods can use this result.

5. Dropping dual constraints. As the DAS algorithm converges, the nonbinding dual constraints in the optimal solution have little influence in the computation of the search direction. Since the elements in the scaling diagonal matrix D corresponding to nonbinding constraints converge to zero, the coefficients in the matrix $AD_k^2A^\top$ are dominated by inner products involving columns corresponding to binding constraints in the optimal solution. Dropping these columns from the computation of the coefficient matrix has two advantages. First, there is an immediate reduction in the computational effort in matrix–vector multiplications carried out in the conjugate gradient algorithm. Second, because of primal degeneracy, matrix $AD_k^2A^\top$ becomes ill conditioned as the algorithm converges. Under this circumstance the rounding errors associated with the matrix–vector multiplications involving very small coefficients induce a perturbation that may slow the convergence of the conjugate gradient algorithm. In this section we describe the dual-constraint-dropping strategy used in this implementation.

Let A_d and D_d be submatrices of A and D corresponding to the dropped columns, and let $A_{\bar{d}}$ and $D_{\bar{d}}$ be the submatrices corresponding to the remaining columns. The coefficient matrix in the system of linear equations can be expressed as

$$AD^2A^\top = A_{\bar{d}}D_{\bar{d}}^2A_{\bar{d}}^\top + A_dD_d^2A_d^\top. \tag{5.1}$$

Since the dropping strategy is such that all coefficients of $A_dD_d^2A_d^\top$ have small values, the coefficient matrix for the linear system can be approximated by the first term of the right-hand side of (5.1).

Under primal degeneracy, as the DAS algorithm converges, the approximate matrix may be rank deficient. The conjugate gradient algorithm can still be used in the case of singular matrices as long as the system of linear equations has a solution. However, the computation of preconditioners requires special attention. In this implementation the computation of the diagonal preconditioner is unchanged. We allow any number of columns to be dropped, which may result in an overdetermined system. When the spanning tree preconditioner is used, we prevent columns corresponding to the spanning tree basis from being dropped.

At each iteration of the DAS algorithm our implementation tries to identify the nonbinding dual constraints in the optimal solution. On the basis of the current dual slack iterate, we partition the dual slacks into two sets: one of small slacks and another of large slacks. To do this we follow the strategy used by Karmarkar and Ramakrishnan in their conjugate gradient implementation [18]. We compute the arithmetic mean of the elements of the scaling matrix

$$A_{|E|}(D^2) = \frac{1}{|E|}\sum_{i=1}^{|E|}\frac{1}{s_i^2}$$

and their harmonic mean

$$H_{|E|}(D^2) = |E|/\sum_{i=1}^{|E|} s_i^2$$

and then take the geometric mean of those two means:

$$\sigma_s = \sqrt{A_{|E|}(D^2)\cdot H_{|E|}(D^2)}.$$

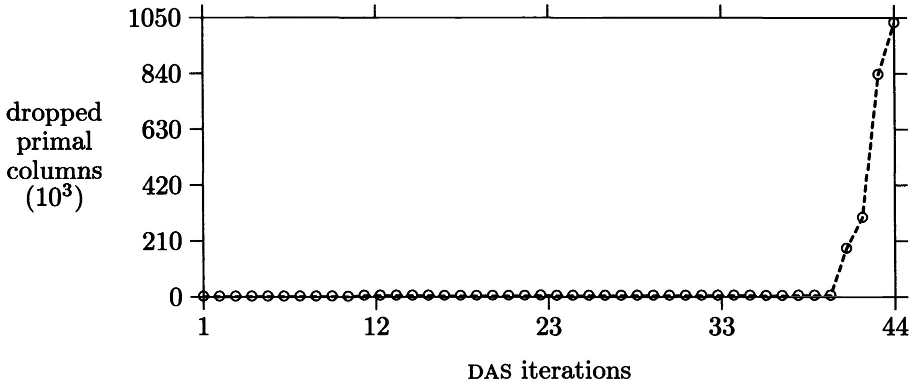


FIG. 5.1. *Primal column dropping on a* 35,000 × 35,000 *assignment problem with* 2,000,000 *edges.*

Constraint i is dropped if $1/s_i^2 < \epsilon_{\text{drop}}\sigma_s$ and, in the case of the spanning tree preconditioner, if edge e_i is not in the current spanning tree basis. ϵ_{drop} is a small tolerance (typically, $\epsilon_{\text{drop}} = 10^{-3}$).

The above scheme allows dropped constraints to be reconsidered in later iterations. In practice, we have observed that often columns that are dropped during a given iteration of DAS are not dropped in a later iteration.

Figure 5.1 illustrates column dropping on a 35,000 × 35,000 assignment problem with 2 million edges. In this problem, a few primal columns are dropped after 10 iterations and a significant number of columns begin being excluded at iteration 41. Just before the process is halted at iteration 44, over 1 million columns are dropped.

6. Avoiding degeneracy. Dual degeneracy, i.e., the existence of multiple primal optimal solutions, is often present in MCNF problems. Affine scaling algorithms have been shown to converge to the relative interior of the optimal face [3], [31], and consequently the primal estimates in the DAS algorithm converge to noninteger solutions in the presence of dual degeneracy. Given the integer nature of the MCNF formulation, we would like an extreme-point optimal solution. In some classes of MCNF problems, such as the assignment problem, the integer requirement cannot be relaxed. The early stopping scheme described in §4 will not be effective in the presence of dual degeneracy. Even when the DAS algorithm correctly identifies the optimal face, the maximum spanning tree procedure can potentially produce an infeasible primal basis.

To circumvent this problem we use a classical idea known as *ϵ-perturbation*, originally due to Charnes [10]. For general-purpose linear programs, by perturbing the primal cost vector c, Megiddo and Chandrasekaran [22] show a polynomial time algorithm for finding ϵ_0 such that for any $0 < \epsilon < \epsilon_0$ the perturbed problem is nondegenerate. However, even this result is specialized for MCNF problems, their theoretical value of ϵ_0 is too small to be used in practice. Instead of using the classical perturbation vector

$$\epsilon = (\epsilon, \epsilon^2, \ldots, \epsilon^n)^\top$$

TABLE 6.1
Effect of degeneracy on assignment problems.

dual degenerate			
problem		iterations	
$\lvert V\rvert$	$\lvert E\rvert$	to opt	total
200	2694	13	15
200	5041	10	12
200	10000	9	13
400	4408	11	15
400	10190	13	17
400	40000	28	28
800	16655	12	16
800	40420	14	18
800	80081	20	23
1000	13380	10	15
1000	37658	13	18
1000	74521	22	26
1500	9350	11	16
1500	27574	11	16

dual nondegenerate			
problem		iterations	
$\lvert V\rvert$	$\lvert E\rvert$	to opt	total
200	2694	13	15
200	5041	8	12
200	10000	8	11
400	4408	8	13
400	10190	11	15
400	40000	16	19
800	8871	9	14
800	16655	10	16
800	80081	20	23
1000	13366	10	15
1000	25523	12	17
1000	74432	22	26
1500	9330	11	16
1500	27502	11	16

in this implementation, we perturb each cost entry randomly,

$$c_i = c_i + \delta_i \epsilon,$$

where δ_i are independently and identically distributed uniform random variables in the interval $(-1, 1)$ and ϵ is a small given real parameter. In this implementation we use $\epsilon = (2M|V|)^{-1}$, where $M = \max\{|u_1|, \ldots, |u_{|E|}|\}$. Our heuristic may still lead to suboptimal solutions, although this did not occur in the experiments described in this paper.

Table 6.1 shows iteration counts for dual degenerate and dual nondegenerate problems. The table identifies problems by number of vertices and edges and shows the number of iterations required to find an optimal basic primal sequence (to opt) and also shows the total number of iterations. The perturbation scheme was successful in avoiding dual degeneracy in all test problems considered. The total numbers of iterations were similar for the two problem classes. This phenomenon (i.e., that in practice interior point methods are not sensitive to degeneracy) has been observed in other studies [29]. However, as expected, for dual nondegenerate problems the primal basis finding scheme of §4 requires fewer iterations than are required for dual degenerate problems.

Recently, Mehrotra [24] described a similar approach and reported the success of his scheme in finding vertex solutions for problems in the *netlib* suite.

7. Parallel implementation. The most computationally intensive steps of the conjugate gradient algorithm are the matrix–vector multiplications (steps 2 and 7 of Fig. 2.1). Because they are matrix–vector multiplications, they are natural candidates for parallel implementation. We have implemented these matrix–vector multiplications in parallel on an eight-processor Alliant FX/80 parallel computer.

To accomplish this implementation let us assume that data structures similar to those described by Adler et al. [2] are used for representing the A and $A^\top$ matrices. Let arrays {`ia`, `ja`, `iat`, `jat`} store the $A^\top$ matrix. Consider the matrix–vector multiplication $y = A^\top x$. It can be carried out with the FORTRAN code in Fig. 7.1. Compiler directives are given for executing the outer loop concurrently (in parallel) and not in vector mode and for executing the inner loops with the vector processor but not concurrently.

```
cvd$l  concur
cvd$l  novector
       do i = 1,n
          y(i) = 0.0
cvd$l     noconcur
cvd$l     vector
          do k = ia(i), ia(i+1)-1
             y(i) = y(i) + x(ja(k))
          enddo
       enddo
```

FIG. 7.1. *Parallel matrix–vector multiplication with Alliant* FORTRAN.

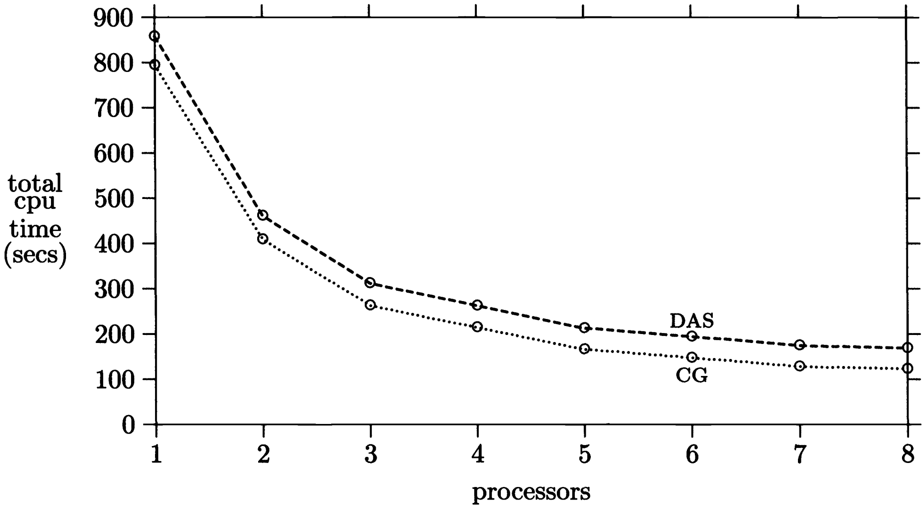


FIG. 7.2. CPU *times for parallel implementation of the conjugate gradient* (CG) *and dual affine scaling* (DAS) *algorithms on a* 500 × 500 *assignment problem with* 37,501 *edges.*

Figures 7.2 and 7.3 illustrate the effect of using 1 through 8 parallel processors on a 500 × 500 assignment problem with 37,501 edges. Figure 7.2 gives total CPU times (in Alliant FX/80 seconds) for both the conjugate gradient and the DAS algorithms. The matrix–vector multiplication in the conjugate gradient algorithm is the only computation implemented in parallel. Consequently, whereas with a single processor the conjugate gradient is responsible for over 90% of the total CPU time, with eight processors it takes only about 75%. Figure 7.3 shows the speedup attained for both the DAS and conjugate gradient algorithms. When eight processors were used, a speedup of about 5 was observed for the interior point algorithm, whereas a speedup factor of approximately 6.5 was observed for the conjugate gradient algorithm.

8. Computational results. In this section we present experimental results for the implementation of the DAS algorithm for the bipartite uncapacitated MCNF problems described in this paper. We report tests on randomly generated assignment problems for which the right-hand side of the constraints in (1.1) is a unit vector. The DAS code used here is a general-purpose solver for bipartite uncapacitated MCNF

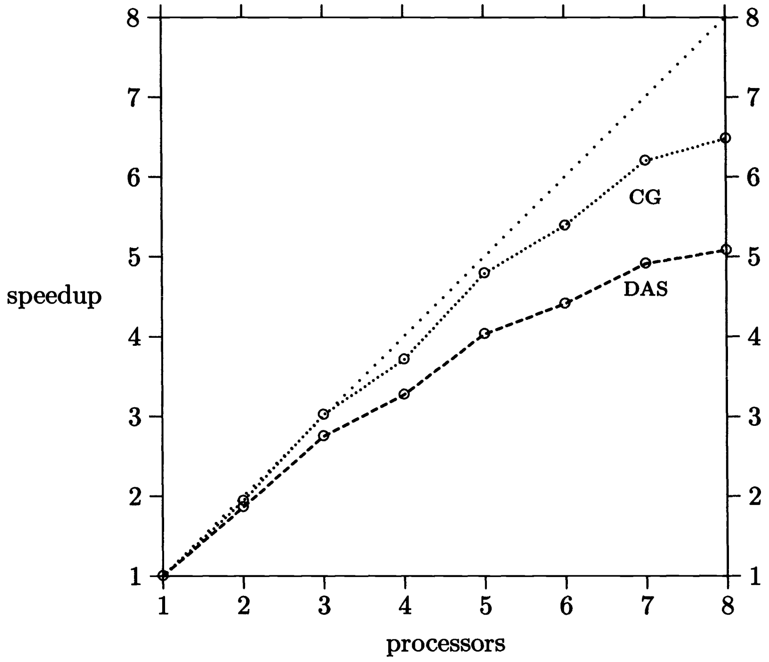


FIG. 7.3. *Speedup for parallel implementation of the conjugate gradinet* (CG) *and dual affine scaling* (DAS) *algorithms on a* 500 × 500 *assignment problem with* 37, 501 *edges.*

problems, with no specific features tailored to the solution of assignment problems. Furthermore, with little modification it can be fitted to handle general uncapacitated problems.

All problems were generated with the random network generator NETRAND. We used NETRAND because it allows control of dual degeneracy in the generation process and provides, a priori, the optimal value of the objective function. We briefly describe NETRAND in §8.1.

We compare the conjugate-gradient-based implementation of DAS with both the network simplex code NETFLO [19] and the relaxation code RELAX [8] on larger assignment problems. Since the DAS code is not tailored to handle assignment problems, we do not compare it with any implementation of the auction algorithm [6], a variation of the relaxation method specific to the solution of assignment problems. Computational results [7] indicate that the auction algorithm is substantially faster than RELAX in the solution of randomly generated assignment problems. Recently, the auction algorithm has been implemented on a massively parallel computer [33].

We have run the codes on problems having linear programming formulations with up to 70,000 constraints and 2,000,000 variables. These results are described in §8.2.

All runs were carried out on an Alliant FX/80 parallel-vector computer. It is configured with 8 parallel processors used for numerically intensive computations and 6 microprocessors used for less intensive tasks, 256 Mbytes of main memory, 512 Kbytes of cache memory, and 3.1 Gbytes of disk storage. Each parallel processor has 8 vector registers, each capable of operating on 32 double-precision numbers simultaneously. In our experiments all code was written in FORTRAN and was compiled on the Alliant FORTRAN compiler with flags `-O -DAS`. No special care was taken to vectorize the code or to implement it in parallel, except for the matrix–vector multiplication in the conjugate gradient algorithm. All times reported are user times given by the system call `times()`.

8.1. Random network generator. NETRAND is a generator of random minimum-cost network flow problems. Unlike other generators used for the same purpose, NETRAND generates problems with a known optimal solution and offers control over the degree of degeneracy present at optimality. This preliminary version of NETRAND, intended as a replacement for NETGEN [20], attempts, with some limitations, to provide the user with the same controls over the problem structure and parameters. Currently, NETRAND can generate uncapacitated MCNF problems, including special subclasses based on bipartite graphs, such as the assignment and transportation problems.

The user can control the topology of the underlying graph by setting the numbers of vertices, edges, sources, and sinks. In the context of NETRAND, sources and sinks are nodes with zero in-degree and zero out-degree, respectively. Consistent with this information, NETRAND generates a random spanning tree, which determines the optimal basis for the MCNF problem. Next, the optimal flow is generated by randomly distributing the total flow supplied by the user in such a way that at least one unit of flow is produced or consumed by each source or sink. By supplying appropriate values for the numbers of vertices, sources, sinks, and total supply, the user can generate common special subclasses of the minimum-cost network flow problem such as the assignment problem and the classical transportation problem.

Before generating the remaining edges for the underlying graph, we generate cost coefficients for the basic edges sampled from a uniform distribution. Basic cost coefficients are sampled from a uniform distribution over a range of values supplied by the user. Lastly, the remaining edges are sampled from the set of all possible edges in such a way that the necessary numbers of degenerate and nondegenerate edges requested by the user are satisfied. For each nondegenerate edge the cost coefficient is sampled from a uniform distribution over the range of values requested by the user intersected with the range values preserving dual feasibility. The numbers of potential degenerate and nondegenerate edges can be inconsistent with the user's request, resulting in a major limitation of the current version of NETRAND.

8.2. Large assignment problems. In this section we report testing the parallel implementation of the network DAS algorithm on large dual nondegenerate assignment problems generated with NETRAND. These problems have a cost structure such that edges corresponding to optimal primal basic variables have costs uniformly distributed between 0 and 10, whereas nonbasic variables have costs uniformly distributed between 0 and 100. We compare the DAS code with two mature network optimization codes, NETFLO [19] and RELAX [8]. Forty-seven problems were generated, ranging from 1,000 to 70,000 vertices and from 20,000 to 2 million edges. The interior point code was run on all problems. In some instances we did not run the other codes. Many of the instances generated have the same dimensions ($|V|$ and $|E|$) but were generated by using different random number generator seeds.

Tables 8.1, 8.2, and 8.3 summarize the runs. In these tables the abbreviations DNR and NA mean "did not record" and "not applicable," respectively.

Table 8.1 shows performance results for the interior point code. Problems are identified by their names and dimensions ($|V|$ and $|E|$). The average number of conjugate gradient iterations and conjugate gradient CPU times per interior point

TABLE 8.1
DAS *runs.*

Problem			Conjugate gradient		DAS iterations		DAS
name	$\vert V\vert$	$\vert E\vert$	ave iters	ave time (s)	total	to opt	time (s)
p001	1000	20000	59.6	3.7	16	14	84.3
p002	1000	20000	83.0	4.8	25	25	153.9
p003	1000	20000	99.2	6.1	17	14	130.5
p004	1000	20000	98.5	6.0	17	14	129.0
p005	1000	20000	109.2	7.2	19	16	166.0
p006	2000	30000	182.2	20.4	17	14	391.5
p007	2000	30000	190.6	21.3	17	14	405.1
p008	2000	30000	179.8	18.9	18	13	386.1
p008	2000	30000	172.5	17.5	19	13	379.0
p010	2000	30000	181.1	19.3	18	13	392.7
p011	2500	78126	DNR	DNR	23	18	589.7
p012	5000	135000	187.7	76.3	24	19	2090.3
p013	5000	135000	103.2	48.6	33	33	1907.3
p014	5000	135000	110.7	53.6	27	23	1718.4
p015	5000	135000	209.9	84.0	23	18	2175.9
p016	5000	135000	103.2	50.5	24	22	1460.6
p017	5000	187500	DNR	DNR	26	20	1866.0
p018	10000	272000	88.7	88.3	25	21	2731.6
p019	10000	272000	120.6	121.2	23	20	3282.0
p020	10000	272000	93.8	90.2	31	31	3392.9
p021	10000	272000	94.6	93.9	25	19	2866.7
p022	10000	272000	257.4	206.4	27	22	6152.3
p023	10000	300000	DNR	DNR	29	22	4003.1
p024	15000	412000	171.5	264.2	39	32	11535.8
p025	15000	412000	81.4	124.5	25	20	3912.3
p026	15000	412000	92.7	140.6	28	22	4788.9
p027	15000	412000	88.3	135.1	27	21	4339.3
p028	15000	412000	83.3	127.2	24	19	3818.9
p029	20000	500000	DNR	DNR	28	20	5441.8
p030	20000	552000	100.6	207.8	28	23	7011.5
p031	20000	552000	499.0	596.8	33	26	21069.8
p032	20000	552000	579.1	957.5	54	48	54224.9
p033	20000	552000	115.6	235.5	37	37	10132.6
p034	20000	552000	99.5	199.7	31	31	7409.2
p035	25000	692000	93.2	242.3	32	32	9319.6
p036	25000	692000	130.5	338.4	34	27	13232.3
p037	25000	692000	102.8	271.7	33	33	10634.0
p038	25000	692000	80.6	214.9	28	24	7449.6
p039	25000	692000	158.2	419.5	34	29	16053.2
p040	30000	832000	227.7	743.7	47	47	37912.6
p041	30000	832000	85.0	264.3	32	32	10341.3
p042	30000	832000	171.7	559.7	47	47	29070.9
p043	30000	832000	114.8	361.1	31	26	13141.7
p044	50000	1000000	DNR	DNR	35	27	15919.3
p045	50000	2000000	146.0	1071.4	36	34	44020.3
p046	60000	2000000	105.5	803.7	36	30	34563.8
p047	70000	2000000	128.7	1009.0	44	35	51419.1

iteration are given, as are the total number of DAS iterations, the number of DAS iterations needed to identify the optimal primal basic sequence (to opt), and the total DAS CPU time. All times exclude problem input for all three codes. Table 8.2 presents results for the network simplex code NETFLO and for the code RELAX. For each instance the table gives the number of simplex iterations, the total simplex CPU time, and the CPU times for the code RELAX.

TABLE 8.2
NETFLO *and* RELAX *runs.*

Problem			NETFLO		RELAX
name	$\|V\|$	$\|E\|$	iterations	time (s)	time (s)
p001	1000	20000	17982	16.4	8.4
p002	1000	20000	18845	18.0	14.6
p003	1000	20000	16906	15.7	13.5
p004	1000	20000	18210	17.3	13.7
p005	1000	20000	15364	15.0	15.6
p006	2000	30000	49739	48.5	23.3
p007	2000	30000	47144	47.1	18.4
p008	2000	30000	54810	51.5	24.8
p009	2000	30000	47557	45.4	21.1
p010	2000	30000	55897	51.8	16.7
p011	2500	78126	76437	324.8	NA
p012	5000	135000	239770	331.9	305.8
p013	5000	135000	205890	296.6	339.9
p014	5000	135000	196564	284.7	481.2
p015	5000	135000	193572	282.2	305.3
p016	5000	135000	196937	284.5	495.2
p017	5000	187500	228316	913.0	NA
p018	10000	272000	624348	1039.3	1081.0
p019	10000	272000	629039	1015.2	1477.0
p020	10000	272000	702648	1105.1	1259.0
p021	10000	272000	702655	1110.8	1081.2
p022	10000	272000	808731	1232.3	920.3
p023	10000	300000	713472	1996.8	NA
p024	15000	412000	1622872	2630.6	2340.0
p025	15000	412000	1306598	2245.4	2298.8
p026	15000	412000	1235798	2097.0	2897.8
p027	15000	412000	1569061	2544.6	2595.8
p028	15000	412000	1377862	2333.9	2362.7
p029	20000	500000	2335066	5449.6	NA
p030	20000	552000	2067875	3868.0	4015.5
p031	20000	552000	2536126	4164.5	2388.3
p032	20000	552000	67294561	105950.6	4054.1
p033	20000	552000	1936375	3547.4	5213.2
p035	25000	692000	3528587	6075.5	4733.8
p036	25000	692000	2920310	5668.5	5260.1
p037	25000	692000	3382243	6222.9	5344.3
p038	25000	692000	2962294	5741.8	5194.3
p039	25000	692000	2651862	4969.7	7911.1
p040	30000	832000	4384954	8197.1	5638.0
p041	30000	832000	7134236	11744.5	6697.8
p042	30000	832000	46117227	76462.9	10761.0
p043	30000	832000	3894180	7581.7	12111.0
p044	50000	1000000	13705215	26174.2	NA
p045	50000	2000000	12053117	27398.3	38845.0
p046	60000	2000000	14261143	30387.1	45476.0
p047	70000	2000000	NA	NA	31986.0

Table 8.3 summarizes the runs by grouping instances with identical dimensions and gives averages for DAS iterations (to identify a primal optimal basic sequence and total), DAS CPU time, NETFLO iterations and CPU times, and RELAX CPU times. NETFLO-to-DAS and RELAX-to-DAS CPU time ratios are also given.

Figures 8.1 and 8.2 illustrate the numerical results. We make the following observations regarding these results:

- There exists a trend in the relative performance of the interior point code and

TABLE 8.3
Summary of runs.

Problem Type		DAS			NETFLO		RELAX	CPU RATIOS	
		average iters		average	average	average	average	NTF ÷	RLX ÷
$\|V\|$	$\|E\|$	to opt	total	time (s)	iters	time (s)	time (s)	DAS	DAS
1000	20000	16.6	18.8	132.7	17461.4	16.5	13.1	0.124	0.099
2000	30000	13.4	17.8	390.9	51029.4	48.9	20.9	0.125	0.053
2500	78126	18.0	23.0	589.7	76437.0	324.8	DNR	0.551	NA
5000	135000	23.0	26.2	1870.5	206546.6	296.0	385.5	0.158	0.206
5000	187500	20.0	26.0	1866.0	228316.0	913.0	DNR	0.489	NA
10000	272000	22.6	26.2	3685.1	693484.2	1100.5	1163.7	0.299	0.316
10000	300000	22.0	29.0	4003.1	713472.0	1996.8	DNR	0.499	NA
15000	412000	22.8	28.6	5679.0	1422438.2	2370.3	2499.0	0.417	0.440
20000	500000	20.0	28.0	5441.8	2335066.0	5449.6	DNR	1.001	NA
20000	552000	33.0	36.6	19969.6	18458734.3	29382.6	3917.8	1.471	0.196
25000	692000	29.0	32.2	11337.7	3089059.2	5735.7	5688.7	0.506	0.502
30000	832000	38.0	39.3	22616.7	15382649.3	25996.5	8802.0	1.149	0.389
50000	1000000	27.0	35.0	15919.3	13705215.0	26174.2	DNR	1.644	NA
50000	2000000	34.0	36.0	44020.3	12053117.0	27398.3	38845.0	0.622	0.882
60000	2000000	30.0	36.0	34563.8	14261143.0	30387.1	45476.0	0.879	1.316
70000	2000000	35.0	44.0	51419.1	DNR	DNR	31986.0	NA	0.622

the other two codes. As problem size increases the relative performance of the interior point code improves. This behavior is similar to what has been observed in comparisons of interior point methods and the simplex method for general linear programming.

- On problems with at least 30,000 vertices, the interior point code was faster than NETFLO in 3 of 7 cases and was faster than RELAX in 1 of 7 cases. On problems with at least 20,000 vertices, DAS was within a factor of two of the solution time of NETFLO in 15 of 22 cases and of RELAX in 11 of 22 cases.
- In all 47 instances the interior-point code found an optimal integer solution. NETFLO failed to find an optimal solution after over 160 million iterations on problem p047.
- The matrix–vector multiplication of the conjugate gradient algorithm in the DAS code was implemented in parallel. The other two codes were not implemented in parallel. However, it is fair to say that the steepest edge variant of the simplex method may also benefit from a parallel architecture. Removing the parallel matrix–vector multiplication should slow down the interior point code by a factor of about five. This will not affect the main observation of this section, i.e., that the relative performance of the DAS code improves with problem size, changing only the break-even point. With eight processors this break-even point appears to be in the vicinity of 20,000 vertices for NETFLO and 50,000 vertices for RELAX.
- On these problems the interior point algorithm spent on average 14% of its iterations to prove optimality, after finding the first feasible basic primal sequence. In all instances the first feasible sequence found always turned out to be optimal. This suggests that there may be some potential for jumping to the network simplex algorithm with this feasible sequence as an initial solution, as suggested by Yeh [34].
- The diagonal preconditioner performed quite well on many of the problems tested. In fact, on several of the largest problems tested there was no need to activate the tree preconditioner. A commonly accepted criterion for goodness of a preconditioner is that it should result in the conjugate gradient algorithm

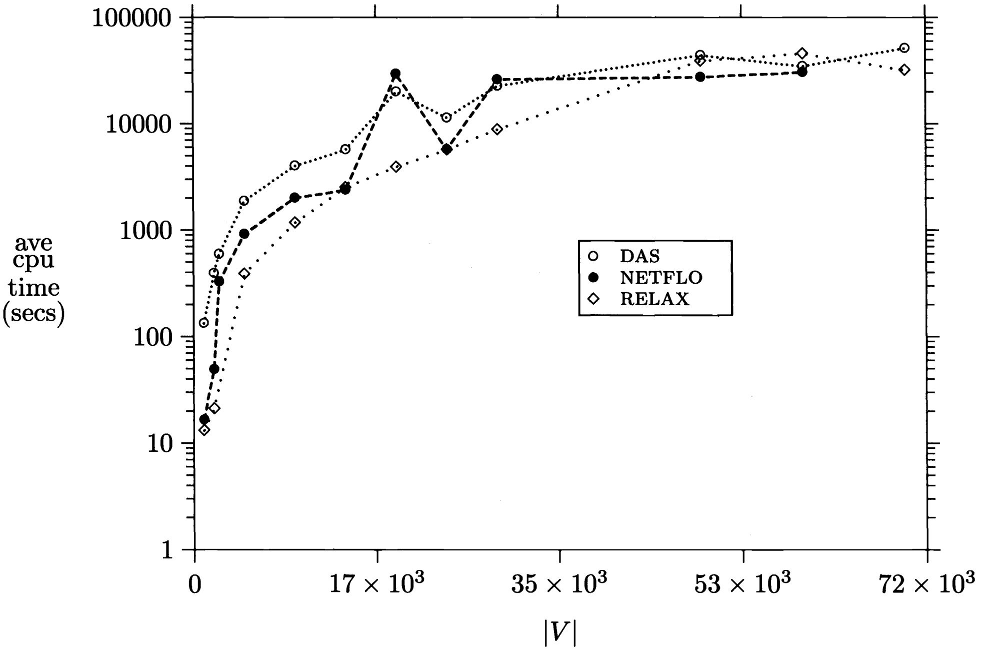


FIG. 8.1. *Running times of* DAS, NETFLO, *and* RELAX.

taking $\sqrt{n}$ iterations to solve an $n \times n$ linear system. In many instances the average number of conjugate gradient iterations was below this threshold. For example, the 70,000 × 70,000 system solved at each DAS iteration of problem p047 took on average only 128.7 conjugate gradient iterations.

- The number of iterations of the network simplex algorithm grows quickly with problem size. For example, for the class of 15,000 × 412,000 problems the simplex method needed, on average, over 1.4 million iterations to find the optimal solution. Even though no factorization is needed by the network simplex method and all computations are carried out in integer arithmetic, as the problems grow these special characteristics of the network simplex method are not sufficient to offset the large number of simplex iterations. On the other hand, the number of affine scaling iterations grows slowly with problem size (see Fig. 8.2). Furthermore, the number of conjugate gradient iterations appears to level off at a low value. Consequently, the ratio of CPU times between NETFLO and the DAS code increases with problem size, in spite of the fact that the interior point code carries out most of its computation in double-precision arithmetic. The determining factor with respect to performance in the interior point implementation is the matrix–vector multiplication. Since this can be very efficiently implemented in parallel, we believe that much performance enhancement should be expected for DAS on multiprocessor computer architectures.
- Figure 8.1 shows that for problems with fewer than 15,000 vertices the code RELAX is the fastest, followed by NETFLO and DAS. However, as the problems increase in size this ranking is no longer valid.

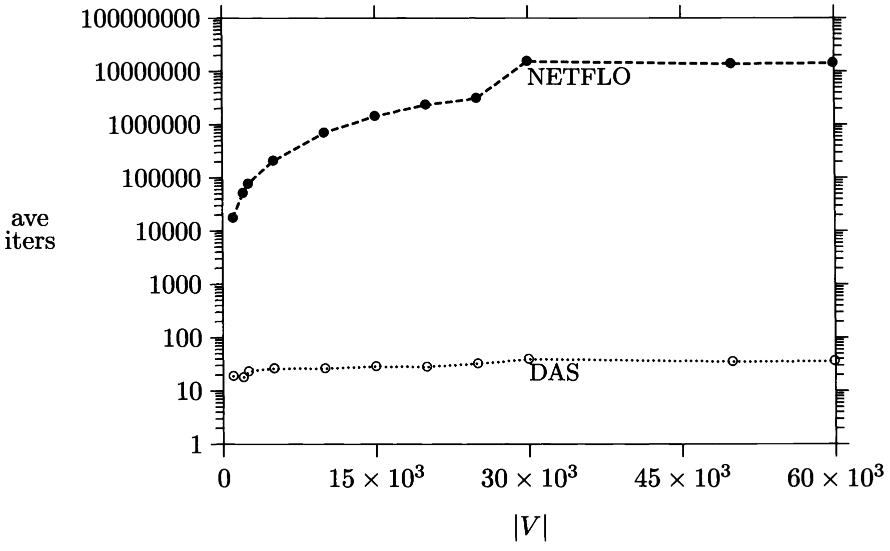


FIG. 8.2. NETFLO *and* DAS *iterations.*

- A feasible dual interior solution was found by DAS in a single iteration on all problems tested.
- In 21 of the 47 problems an optimal primal basic sequence was found the first time a spanning tree was built.

9. Concluding remarks. In this paper we described an implementation of the DAS algorithm for linear programming for solving bipartite uncapacitated minimum-cost network flow problems. Because of the excessive computational demand of direct factorization, interior point methods were previously thought not to be competitive with other methods for solving problems in this class. Our implementation makes use of a preconditioned conjugate gradient algorithm to compute the ascent direction. Besides being much more efficient than direct factorization, the conjugate gradient algorithm depends heavily on matrix–vector multiplication, which can be implemented in parallel. We implemented a parallel conjugate gradient algorithm and observed a speedup of over a factor of 5 in the interior point code on an eight-processor parallel computer.

We limited our experimental study to a special class of minimum-cost network flow problems: assignment problems. For problems in this class we performed extensive computational experiments, concluding that as problem sizes increase the interior point method's performance relative to other commonly used MCNF algorithms improves. For the largest problems tested our code was competitive with both the network simplex code NETFLO and the relaxation method code RELAX. If the observed trend continues with larger problems, one should expect the interior point method to be the method of choice for solving large-scale network flow problems.

To test our code on general minimum-cost network flow problems, we must first implement lower and upper bounds on the flow variables and data structures to handle directed edges in general networks.

Acknowledgments. M. G. C. Resende acknowledges several insightful discussions with N. Karmarkar, K. G. Ramakrishnan, and P. Vaidya. Both authors acknowledge an anonymous referee whose suggestions greatly improved the paper.

REFERENCES

[1] I. Adler, N. Karmarkar, M. Resende, and G. Veiga, *An implementation of Karmarkar's algorithm for linear programming*, Math. Programming, 44 (1989), pp. 297–335.

[2] ———, *Data structures and programming techniques for the implementation of Karmarkar's algorithm*, ORSA J. Comput., 1 (1989), pp. 84–106.

[3] I. Adler and R. Monteiro, *Limiting behaviour of the affine scaling continuous trajectories for linear programming problems*, Math. Programming, 50 (1991), pp. 29–51.

[4] A. Armacost and S. Mehrotra, *Computational comparison of the network simplex method with the affine scaling method*, Opsearch, 28 (1991), pp. 26–43.

[5] J. Aronson, R. Barr, R. Helgason, J. Kennington, A. Loh, and H. Zaki, *The Projective Transformation Algorithm of Karmarkar: A Computational Experiment with Assignment Problems*, Tech. Report 85-OR-3, Department of Operations Research, Southern Methodist University, Dallas, TX, 1985.

[6] D. Bertsekas, *A new algorithm for the assignment problem*, Math. Programming, 21 (1981), pp. 152–171.

[7] ———, *The auction algorithm for assignment and other network problems: A tutorial*, Interfaces, 20 (1990), pp. 133–149.

[8] D. Bertsekas and P. Tseng, *Relaxation methods for minimum cost ordinary and generalized network flow problems*, Oper. Res., 36 (1988), pp. 93–114.

[9] W. Carolan, J. Hill, J. Kennington, S.Niemi, and S. Wichmann, *An empirical evaluation of the Korbx algorithms for military airlift applications*, Oper. Res., 38 (1990), pp. 240–248.

[10] A. Charnes, *Optimality and degeneracy in linear programming*, Econometrica, 20 (1952), pp. 160–170.

[11] G. Dantzig, *Maximization of a linear function of variables subject to linear inequalities*, in Activity Analysis of Production and Allocation, T. Koopsmans, ed., John Wiley, New York, 1951, pp. 339–347.

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

[13] D. Gay, *Electronic mail distribution of linear programming test problems*, Math. Programming Committee Algorith. Newsletter, 13 (1985), pp. 10–12.

[14] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983.

[15] C. Gonzaga, *Convergence of the Large Step Primal Affine-Scaling Algorithm for Primal Nondegenerate Linear Programs*, Tech. Report ES-230/90, Department of Systems Engineering and Computer Science, COPPE/Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, 1990.

[16] M. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand. Sec. B, 49 (1952), pp. 409–436.

[17] N. Karmarkar and K. Ramakrishnan, private communication, 1988.

[18] ———, *Computational results of an interior point algorithm for large scale linear programming*, Math. Programming, 52 (1991), pp. 555–586.

[19] J. Kennington and R. Helgason, *Algorithms for Network Programming*, John Wiley, New York, 1980.

[20] D. Klingman, A. Napier, and J. Stutz, NETGEN: *A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems*, Management Sci., 20 (1974), pp. 814–821.

[21] J. Kruskal, Jr., *On the shortest spanning subtree of a graph and the traveling salesman problem*, Proc. Amer. Math. Soc., 7 (1956), pp. 48–50.

[22] N. Megiddo and R. Chandrasekaran, *On the ϵ-Perturbation Method for Avoiding Degeneracy*, Tech. Report, IBM Almaden Research Center, San Jose, CA, 1988.

[23] S. Mehrotra, *Implementation of Affine Scaling Methods: Approximate Solution of System of Linear Equations Using Preconditioned Conjugate Gradient Methods*, Tech. Report 89-04, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 1989.

[24] ———, *On finding a vertex solution using interior point methods*, Linear Algebra Appl., 152 (1991), pp. 233–253.

[25] J. Meijerink and H. van der Vorst, *An iterative method for linear equation systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comput., 31 (1977), pp. 148–162.

[26] S. Mizuno, S. Saigal, and J. Orlin, *Determination of Optimal Vertices from Feasible Solutions in Unimodular Linear Programming*, Tech. Report, Department of Prediction and Control, Institute of Statistical Mathematics, Tokyo, 1989.

[27] C. Monma and A. Morton, *Computational experimental with a dual affine variant of Karmarkar's method for linear programming*, Oper. Res. Let., 6 (1987), pp. 261–267.

[28] B. Murtagh and M. Saunders, MINOS 5.0 *User's Guide*, Tech. Report SOL 77-9, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA, 1977.

[29] M. Todd, *The effects of degeneracy and unbounded variables on variants of Karmarkar's linear programming algorithm*, in Large-Scale Numerical Optimization, T. Coleman and Y. Li, eds., Society for Industrial and Applied Mathematics, Philadelphia, 1990, pp. 81–91.

[30] M. Todd and B. Burrell, *An extension to Karmarkar's algorithm for linear programming using dual variables*, Algorithmica, 1 (1986), pp. 409–424.

[31] T. Tsuchiya, *Global convergence of the affine scaling methods for degenerate linear programming problems*, Math. Programming, 52 (1991), pp. 377–404.

[32] P. Vaidya, *Solving Linear Equations with Symmetric Diagonally Dominant Matrices by Constructing Good Preconditioners*, Tech. Report, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 1990.

[33] J. Wein and S. Zenios, *On the Massively Parallel Solutions of the Assignment Problem*, Tech. Report, Department of Decision Sciences, Wharton School, University of Pennsylvania, Philadelphia, 1990.

[34] Q.-J. Yeh, *A Reduced Dual Affine Scaling Algorithm for Solving Assignment and Transportation Problems*, Ph.D. thesis, Department of Industrial Engineering and Operations Research, Columbia University, New York, 1989.

SIAM J. OPTIMIZATION
Vol. 3, No. 3, pp. 538–543, August 1993

CONVERGENCE ANALYSIS OF A PROXIMAL-LIKE MINIMIZATION ALGORITHM USING BREGMAN FUNCTIONS*

GONG CHEN† AND MARC TEBOULLE†

Abstract. An alternative convergence proof of a proximal-like minimization algorithm using Bregman functions, recently proposed by Censor and Zenios, is presented. The analysis allows the establishment of a global convergence rate of the algorithm expressed in terms of function values.

Key words. Bregman functions, proximal methods, convex programming

AMS subject classification. 90C25

1. Introduction.

Consider the convex optimization problem

$$(1) \qquad (P) \qquad \min\{f(x) : x \in \mathbb{R}^n\},$$

where $f : \mathbb{R}^n \mapsto (-\infty, +\infty]$ is a proper, lower semicontinuous convex function. One method of solving (P) is to regularize the objective function by using the proximal mapping as introduced by Moreau [12]. Given a real positive number λ, a proximal approximation of f is defined by

$$(2) \qquad f_\lambda(x) = \inf_u\{f(u) + 1/2\lambda\|x-u\|^2\}.$$

As proved by Moreau [12], the function f_λ is convex and differentiable, and when it is minimized it possesses the same set of minimizers and the same optimal value as problem (P). Using these properties, Martinet [11] introduced the proximal minimization algorithm for solving problem (P). The method is as follows: given an initial point $x_0 \in \mathbb{R}^n$, a sequence $\{x_k\}$ is generated by solving

$$(3) \qquad x^k = \operatorname{argmin}\{f(x) + (1/2\lambda_k)\|x - x^{k-1}\|^2\},$$

where $\{\lambda_k\}_{k=1}^{\infty}$ is a sequence of positive numbers. A major contribution to proximal methods has been developed by Rockafellar [15], who proved the convergence of the proximal point algorithm for finding the zero of an arbitrary maximal monotone operator and who gave applications to convex programming in [14]. For further details and references on proximal methods we refer the reader to the excellent survey paper of Lemaire [10].

Several researchers have considered the possibility of replacing the quadratic kernel in (2)–(3) by entropylike distances; see, e.g., [5], [7], [8], [16], and [17]. In [5] Censor and Zenios replaced method (3) by a method of the form

$$(4) \qquad x^k = \operatorname{argmin}\{f(x) + \lambda_k^{-1}D(x, x^{k-1})\},$$

with D being a Bregman's distance or D-function (see §2 for a definition), and that is accordingly called the *proximal minimization with D-functions* (PMD).

This paper presents an alternative proof of the PMD algorithm proposed in [5]. Our analysis is motivated by the elegant new convergence proof results developed by

* Received by the editors January 21, 1991; accepted for publication (in revised form) April 9, 1992. This research was partially supported by U.S. Air Force Office of Scientific Research grants 0218-88 and 91-0008 and National Science Foundation grants ECS-8802239 and DMS-92012297.

† Department of Mathematics and Statistics, University of Maryland, Baltimore County Campus, Baltimore, Maryland 21228.

Guler [9] for the classical proximal minimization algorithm (3). It allows us to obtain a global convergence rate estimate for the residual $f(x_k) - f(u)$, where $u \in \mathbb{R}^n$ is arbitrary (Theorem 3.4 below), thus revealing that the lines of analysis of [3] and [9] also apply to the PMD algorithm.

For a generalization of the PMD algorithm for finding the zero of a monotone operator see [7], and for applications of nonquadratic proximal methods to convex programming see, e.g., [7], [16], and [17]; in the latter convergence results are derived for the exponential multiplier method [2].

2. Proximal minimization algorithm with Bregman functions. The notations and definitions used in the following are as in the book by Rockafellar [13]. In particular, $\operatorname{dom} f$, $\operatorname{ran} f, \operatorname{ri} C$, and $\bar{C}$ denote the domain and range of f, the relative interior of the set C, and the closure of the set C, respectively.

Given a differentiable function ψ, a measure of distance based on Bregman's distance [1] is defined by

$$D_\psi(x,y) = \psi(x) - \psi(y) - \langle x-y, \nabla\psi(y)\rangle, \tag{5}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product in $\mathbb{R}^n$ and where $\nabla\psi$ is the gradient of ψ. The function ψ is called a *Bregman function* if it satisfies the properties given in the definition below; see, e.g., [4] and [6].

DEFINITION 2.1. Let $S \in \mathbb{R}^n$ be an open set. Then $\psi : \bar{S} \mapsto \mathbb{R}$ is called a Bregman function with zone S if the following hold:

(i) ψ is continuously differentiable on S.

(ii) ψ is strictly convex and continuous on $\bar{S}$.

(iii) For every $\alpha \in \mathbb{R}$ the partial level sets $L_1(y,\alpha) = \{x \in \bar{S} : D_\psi(x,y) \leq \alpha\}$ and $L_2(x,\alpha) = \{y \in S : D_\psi(x,y) \leq \alpha\}$ are bounded for every $y \in S$ and $x \in \bar{S}$.

(iv) If $\{y^k\} \in S$ converges to y^*, then $D_\psi(y^*, y^k) \to 0$.

(v) If $\{x^k\}$ and $\{y^k\}$ are sequences such that $y^k \to y^* \in \bar{S}$, $\{x^k\}$ is bounded, and if $D_\psi(x^k, y^k) \to 0$, then $x^k \to y^*$.

The class of functions satisfying the conditions of Definition 2.1 is denoted by $\mathcal{B}$. Note that assumptions (iv) and (v) are necessary only for computational purposes. $D_\psi(x,y)$ is not a distance (it might not be symmetric and might not satisfy the triangle inequality), but by the strict convexity of ψ it follows immediately that $D_\psi(x,y) \geq 0$ and is equal to zero if and only if $x = y$. With the special choice $S = \mathbb{R}^n$ and $\psi(x) = \frac{1}{2}||x||^2$ one obtains $D_\psi(x,y) = \frac{1}{2}||x-y||^2$. Another prominent example useful in applications (see, e.g., [4], [17]) is obtained by choosing the entropy kernel.

Example 2.1. With $S = \mathbb{R}^n_{++} := \{x \in \mathbb{R}^n : \ x_i > 0,\ i = 1,\dots,m\}$ and $\psi(x) = \sum_{i=1}^n x_i \log x_i - x_i$ (with the convention $0 \log 0 = 0$) we obtain the Kullback–Liebler relative entropy distance

$$D_\psi(x,y) = \sum_{i=1}^n x_i \log\frac{x_i}{y_i} + y_i - x_i \quad \forall\, (x,y) \in \mathbb{R}^n_+ \times \mathbb{R}^n_{++}. \tag{6}$$

Note that all the assumptions of Definition 2.1 are met. Other characterizations of Bregman functions and examples can be found in [6], [7], and [16].

The PMD algorithm is as follows. Given $\psi \in \mathcal{B}$ and a sequence of positive $\{\lambda_k\}$ satisfying

$$\lim_{n\to\infty} \sum_{k=1}^n \lambda_k = +\infty \tag{7}$$

and starting with an initial point $x_0 \in S$, one generates a sequence $\{x_k\}$ by the iterative scheme

$$x^k = \operatorname*{argmin}_{x \in \mathbb{R}^n} \{f(x) + \lambda_k^{-1} D_\psi(x, x^{k-1})\}. \tag{8}$$

The convergence of this algorithm has been proved in [5] under the following assumptions:

(A) $\liminf_{k\to\infty}\{\lambda_k : k \geq 0\} > 0$;

(B) f is bounded below, and the PMD generates a sequence $\{x_k\}$ such that $x_k \in S$ for all k;

(C) $\psi \in \mathcal{B}$ is twice continuously differentiable with positive definite Hessian, and $D_\psi(\cdot, \cdot)$ is jointly convex.

Observe that instead of (A), which asked the sequence $\{\lambda_k\}$ to be bounded away from zero, we request the weaker assumption (7). However, it should be noted that in practice (A) is not restrictive. As an alternative to (B), which is an assumption on the objective function f, we make an assumption on the Bregman function ψ. Following [7], we assume that $\operatorname{ran} \nabla\psi = \mathbb{R}^n$. This assumption needed to guarantee that the existence of the sequence $\{x^k\}$ produced by (8) is a relatively mild one and is satisfied for the entropy case of Example 2.1 and for many other interesting examples; see, e.g., [6], [7], and [16] for other assumptions that guarantee the existence of x^k. Finally, we note that (C) seems to be a redundant assumption.

3. Convergence analysis of the PMD method.

In this section we derive a global convergence rate estimate for the PMD algorithm, from which its convergence follows. Our analysis is different from that proposed in [5] and [7]; it follows the more direct approach of [9]. One important element in the convergence proof is a rather simple property satisfied by the Bregman distance that apparently has not been observed before. This property, which we call a *three-points identity*, appears to be a natural generalization of the quadratic identity valid for the Euclidean norm.

LEMMA 3.1. *Let $\psi \in \mathcal{B}$. Then for any three points $a, b \in S$ and $c \in \bar{S}$ the following identity holds:*

$$D_\psi(c, a) + D_\psi(a, b) - D_\psi(c, b) = \langle \nabla\psi(b) - \nabla\psi(a), c - a\rangle. \tag{9}$$

Proof. Using the definition of D_ψ, we have

$$\langle \nabla\psi(a), c - a\rangle = \psi(c) - \psi(a) - D_\psi(c, a), \tag{10}$$

$$\langle \nabla\psi(b), a - b\rangle = \psi(a) - \psi(b) - D_\psi(a, b), \tag{11}$$

$$\langle \nabla\psi(b), c - b\rangle = \psi(c) - \psi(b) - D_\psi(c, b). \tag{12}$$

Subtracting (10) and (11) from (12) gives the result. □

The next result, which is the key ingredient for the proof of our theorem, generalizes for Bregman distances the result in [9, Lem. 2.2].

LEMMA 3.2. *Let $f(x)$ be a closed proper convex function on $\mathbb{R}^n$. Given $\psi \in \mathcal{B}$ with* $\operatorname{ri}(\operatorname{dom} f) \subseteq S$, *let $\{\lambda_k\}$ be an arbitrary sequence of positive numbers and let $\{x_k\}$ be the sequence generated by the PMD given in (8). Then for any $u \in \bar{S}$*

$$\lambda_k(f(x^k) - f(u)) \leq D_\psi(u, x^{k-1}) - D_\psi(u, x^k) - D_\psi(x^k, x^{k-1}). \tag{13}$$

Proof. From (8), x_k is the minimum point of $f(x) + D_\psi(x, x^{k-1})$. Then, since $\mathrm{ri}(\mathrm{dom}\, f) \subset S$ from [13, Thm. 27.4], this is equivalent to

$$\langle u - x^k, \nabla_x D_\psi(x, x^{k-1}) + \lambda_k y^k \rangle \geq 0$$

for all $u \in \bar{S}$ and some $y^k \in \partial f(x^k)$, the subdifferential of f at x^k. From the definition of D_ψ the above inequality is equivalent to

$$\lambda_k \langle x^k - u, y^k \rangle \leq \langle u - x^k, \nabla\psi(x^k) - \nabla\psi(x^{k-1}) \rangle. \tag{14}$$

Applying Lemma 3.1 at the points $c = u$, $a = x^k$, $b = x^{k-1}$, we obtain

$$\langle u - x^k, \nabla\psi(x^k) - \nabla\psi(x^{k-1}) \rangle = D_\psi(u, x^{k-1}) - D_\psi(u, x^k) - D_\psi(x^k, x^{k-1}). \tag{15}$$

But since f is convex and $y^k \in \partial f(x^k)$, we also have

$$\lambda_k (f(x^k) - f(u)) \leq \lambda_k \langle x^k - u, y^k \rangle. \tag{16}$$

If (14), (15), and (16) are combined, the result follows. □

In the following the infimum of f is denoted by $f_* = \inf_{x \in \mathbb{R}^n} f(x)$ and the set of minimizers of f (possibly empty) is denoted by $X_* = \{x \in \mathbb{R}^n : f(x) = f_*\}$. We define $\sigma_n = \sum_{k=1}^n \lambda_k$.

LEMMA 3.3. *Suppose the conditions of Lemma 3.2 are met. Then*

(i) *$f(x^k)$ is nonincreasing;*

(ii) *$D_\psi(u, x^k)$ is nonincreasing whenever $u \in X_*$;*

(iii) *$\sigma_n(f(x^n) - f(u)) \leq D_\psi(u, x^0) - D_\psi(u, x^n) - \sum_{k=1}^n \lambda_k^{-1} \sigma_k D_\psi(x^k, x^{k-1})$, $u \in \bar{S}$.*

Proof. (i) Since x^k satisfies (8), for all $x \in \mathbb{R}^n$

$$f(x^k) + \lambda_k^{-1} D_\psi(x^k, x^{k-1}) \leq f(x) + \lambda_k^{-1} D_\psi(x, x^{k-1}),$$

and thus, in particular, with $x = x^{k-1}$ it follows that

$$\lambda_k (f(x^{k-1}) - f(x^k)) \geq D_\psi(x^k, x^{k-1}) \geq 0 \tag{17}$$

since $D_\psi(x^{k-1}, x^{k-1}) = 0$.

(ii) For all $u \in X_*$, $f(x^k) - f(u) \geq 0$. Then, by using Lemma 3.2

$$0 \leq \lambda_k (f(x^k) - f(u)) \leq D_\psi(u, x^{k-1}) - D_\psi(u, x^k) - D_\psi(x^k, x^{k-1}),$$

from which we obtain

$$D_\psi(u, x^k) \leq D_\psi(u, x^{k-1}) - D_\psi(x^k, x^{k-1}) \leq D_\psi(u, x^{k-1})$$

since $D_\psi(x^k, x^{k-1}) \geq 0$.

(iii) Using $\sigma_k = \lambda_k + \sigma_{k-1}$, with $\sigma_0 = 0$, multiplying (17) by σ_{k-1}, and summing over $k = 1, ..n$, one has

$$\begin{gathered} \sigma_{k-1} f(x^{k-1}) - (\sigma_k - \lambda_k) f(x^k) \geq \sigma_{k-1} \lambda_k^{-1} D_\psi(x^k, x^{k-1}), \\ -\sigma_n f(x^n) + \sum_{k=1}^n \lambda_k f(x^k) \geq \sum_{k=1}^n \sigma_{k-1} \lambda_k^{-1} D_\psi(x^k, x^{k-1}). \end{gathered} \tag{18}$$

Once again using Lemma 3.2 by summing (13) over $k = 1, ..n$, we obtain

$$(19) \qquad -\sigma_n f(u) + \sum_{k=1}^{n} \lambda_k f(x^k) \leq D_\psi(u, x^0) - D_\psi(u, x^n) - \sum_{k=1}^{n} D_\psi(x^k, x^{k-1}).$$

Then subtracting (18) from (19) gives (iii). □

We are now in position to prove our main result for the PMD algorithm.

THEOREM 3.4. *Let the sequence $\{x_k\}$ be generated by the PMD algorithm. For any $u \in \bar{S}$*

$$(20) \qquad f(x^n) - f(u) \leq \sigma_n^{-1} D_\psi(u, x^0).$$

Therefore, if $\sigma_n \to \infty$, then $f(x^n) \to f_* = \inf f(x)$. Moreover, if $X_* \neq \emptyset$, then x^n converges to a minimizer of f and satisfies

$$(21) \qquad f(x^n) - f^* \leq \sigma_n^{-1} D_\psi(x^*, x^0) \ \forall x^* \in X_*.$$

Proof. From Lemma 3.3 (iii) we immediately obtain (20). Consider the case $f_* > -\infty$. From Lemma 3.3 (i), $f(x^k)$ is nonincreasing. By the definition of f_* there exists a v such that $f(v) < f_* + \epsilon$ for any $\epsilon > 0$. Let $u = v \in \bar{S}$ in (20), and take the limit with $\sigma_n \to +\infty$; we obtain $\lim_{k\to\infty} f(x^k) < f_* + \epsilon$. Hence since ϵ is arbitrary, $\lim_{k\to\infty} f(x^k) = f_*$. Similarly, one can prove the convergence in the case $f_* = -\infty$. If $X_* \neq \emptyset$, there exists a $x^* \in X_*$. Let $u = x^*$ in (20); then

$$(22) \qquad f(x^n) - f(x^*) \leq \sigma_n^{-1} D_\psi(x^*, x^0),$$

which proves (21). Now from Lemma 3.3 (ii), $D_\psi(x^*, x^k)$ is nonincreasing; hence $D_\psi(x^*, x^k) \leq D_\psi(x^*, x^0)$. Since $\psi \in \mathcal{B}$, from Definition 2.1 (iii) it follows that $\{x^k\}$ is bounded. Let z be a limit point of $\{x^k\}$ with subsequence $\{x^{k_j}\} \to z$. Since $f(x^k) \to f_*$, $f(x^{k_j}) \to f_*$, and so $f(z) \leq f_*$ because f is lower semicontinuous. It follows that $z \in X_*$. Now from Lemma 3.3(ii), $D_\psi(z, x^k)$ is nonincreasing; hence for any $k \geq k_j$

$$(23) \qquad D_\psi(z, x^k) \leq D_\psi(z, x^{k_j}).$$

Since $x^{k_j} \to z$, for any positive δ, starting from some k_{j_0},

$$(24) \qquad D_\psi(z, x^{k_{j0}}) \leq \delta.$$

Therefore $D_\psi(z, x^k) \to 0$. To prove that $\{x_k\}$ has only one limit point let $\bar{x} \in \bar{S}$ be another limit point of $\{x^k\}$. Then $D(z, x^{k_l}) \to 0$ with $x^{k_l} \to \bar{x}$. So from Definition 2.1 (v), $z = \bar{x}$, and therefore $x^k \to z \in X_*$. □

REFERENCES

[1] L. BREGMAN, *The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming*, U.S.S.R. Comput. Math. and Math. Phys., 7 (1967), pp. 200–217.

[2] D. P. BERTSEKAS, *Constrained Optimization and Lagrangian Multipliers*, Academic Press, New York, 1981.

[3] H. BREZIS AND P. L. LIONS, *Produits infinis de resolvantes*, Israel J. Math., 29 (1978), pp. 329–345.

[4] Y. Censor and A. Lent, *An interval row action method for interval convex programming*, J. Optim. Theory Appl., 34 (1981), pp. 321–353.

[5] Y. Censor and S. A. Zenios, *The proximal minimization algorithm with D-functions*, J. Optim. Theory Appl., 73 (1992), pp. 451–464.

[6] A. DePierro and A. Iusem, *A relaxed version of Bregman's method for convex programming*, J. Optim. Theory Appl., 5 (1986), pp. 421–440.

[7] J. Eckstein, *Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming*, Math. Oper. Res., to appear.

[8] P. P. B. Eggermont, *Multiplicative iterative algorithms for convex programming*, Linear Algebra Appl., 130 (1990), pp. 25–42.

[9] O. Guler, *On the convergence of the proximal point algorithm for convex minimization*, SIAM J. Control Optim., 29 (1991), pp. 403–419.

[10] B. Lemaire, *The proximal algorithm*, in International Series of Numerical Mathematics, Vol. 87, J. P. Penot, ed., Birkhäuser-Verlag, Basel, Switzerland, 1989, pp. 73–87.

[11] B. Martinet, *Pertubation des methodes d'optimisation: Application*, RAIRO Anal. Numer., 12 (1978), pp. 153–171.

[12] J. J. Moreau, *Proximité et dualite dans un espace Hilbertien*, Bull. Soc. Math. France, 93 (1965), pp. 273–299.

[13] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[14] ———, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.

[15] ———, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[16] M. Teboulle, *Entropic proximal mappings with applications to nonlinear programming*, Math. Oper. Res., 17 (1992), pp. 670–690.

[17] P. Tseng and D. P. Bertsekas, *On the convergence of the exponential multiplier method for convex programming*, Math. Programming, to appear.

SIAM J. OPTIMIZATION
Vol. 3, No. 3, pp. 544-563, August 1993

007

A LAGRANGIAN RELAXATION ALGORITHM FOR MULTIDIMENSIONAL ASSIGNMENT PROBLEMS ARISING FROM MULTITARGET TRACKING*

AUBREY B. POORE† AND NENAD RIJAVEC†

Abstract. The central problem in multitarget tracking is the data association problem of partitioning the observations into tracks in some optimal way so that an accurate estimate of the true tracks can be recovered. This work considers what is perhaps the simplest multitarget tracking problem in a setting where the issues are easily delineated, i.e., straight lines in two-dimensional space-time with an error component introduced into the observations. A multidimensional assignment problem is formulated using gating techniques to introduce sparsity into the problem and filtering techniques to generate tracks which are then used to score each assignment of a collection of observations to a filtered track. Problem complexity is further reduced by decomposing the problem into disjoint components, which can then be solved independently. A recursive Lagrangian relaxation algorithm is developed to obtain high quality suboptimal solutions in real-time. The algorithms are, however, applicable to a large class of sparse multidimensional assignment problems arising in general multitarget and multisensor tracking. Results of extensive numerical testing are presented for a case study to demonstrate the speed, robustness, and exceptional quality of the solutions.

Key words. multitarget tracking, multidimensional assignment problems, data association, Lagrangian relaxation

AMS subject classification. 90C08

1. Introduction and overview. The problem of taking pictures at a discrete set of times of a large number of objects moving in space, and then from these observations determining the past or present and predicting the future states of these objects is fundamental to multitarget and multisensor tracking [4]–[6], [13], [26], [28], [29], [32], [33], [39], [41]. Central to this process is the data association problem, in which the observations are partitioned into tracks in such a way that the tracks of the objects can be identified [5], [6], [13], [28], [32], [33], [39], [41]. (Smoothing, filtering, and prediction techniques [2] can be used to obtain further information about past, present, and future states.) Although combinatorial optimization is the natural framework for the formulation of these problems, the corresponding techniques have long been considered computationally too intensive for real-time applications, and for good reason. The corresponding optimization algorithms for these problems, which are formulated here as multidimensional assignment problems, are claimed to be NP-hard [31]. To further appreciate the difficulties, one only has to examine the tradeoffs between two current methods in multitarget tracking [13]: track while scan and batch. For the former, one essentially extends tracks a scan at a time using, for example, a two-dimensional assignment [12], [24], [27] or a greedy algorithm [13]. This methodology is real-time but results in a large number of partial and incorrect assignments, and thus incorrect track identification. The fundamental difficulty with this approach is the lack of information in single scan processing to partition the observations into tracks. To obtain the required information, one needs to consider several scans all at

*Received by the editors October 31, 1990; accepted for publication (in revised form) April 15, 1992. This work was partially supported by the Federal Systems Corporation of the IBM Corporation in Boulder, Colorado and Owego, New York, and by the Air Force Office of Scientific Research through AFOSR grants AFOSR-88-0059 and AFOSR-91-0138.

†Department of Mathematics, Colorado State University, Fort Collins, Colorado 80523.

once, i.e., the batch approach, but this approach is considered computationally too intensive for practical real-time applications. Given the ever-present need to identify *all* tracks in *real-time*, the challenge to combinatorial optimization is to design fast algorithms for advanced computer architectures that will solve the underlying data association problem, and thus the track identification, in real-time.

The use of combinatorial optimization in multitarget tracking is not new and dates back to the mid-sixties and the pioneering work of Sittler [39], who used maximum likelihood estimation to evaluate all possible track updates and employed track splitting (several hypotheses were maintained for each track) and pruning (when their probabilities fell below a certain threshold). Maximum likelihood estimation was further investigated by Stein and Blackman [41], who developed a comprehensive probability for track initiation, track length expectancy, missed detections, and false alarms. Morefield [28] pioneered the use of integer programming to solve a set packing problem arising from a data association problem. These works are further discussed in the books of Blackman [13], Bar-Shalom and Fortmann [6], and Bar-Shalom [5], which also serve as excellent introductions to the field of multitarget tracking. More recently, Pattipati, Somnath, Bar-Shalom, and Washburn [32], [33] have formulated multidimensional assignment problems for the passive multisensor data association problem. For the three-dimensional problem they present a Lagrangian relaxation algorithm and impressive numerical results, and then discuss extensions to the multidimensional problem. Their three-dimensional algorithm, like ours, is essentially that of Frieze and Yadegar [17].

The objective in this work is to begin an exploration of the data association problem using combinatorial optimization techniques. We purposely consider what is perhaps the simplest multitarget tracking problem in a setting where the issues are easily delineated; however, the multidimensional assignment problems are quite close in complexity to those for very general models [35].

The formulation of the data association problem as a K-dimensional assignment problem (K corresponds to the number of scans) is achieved in four stages: *gating*, *filtering*, *scoring*, and *assignment formulation.* The purpose of *gating* is to rule out the most unlikely combinations of observations and thereby to introduce sparsity into the problem. Given a combination of observations (one from each scan) from gating, the *filtering* problem is to generate a track which might have produced these observations. Given a feasible combination of observations and a filtered track, one next assigns a *score* or price to the assignment of this combination of observations to its filtered track. Finally, the K-dimensional *assignment problem* is formulated so that each observation is required to belong to exactly one track.

In designing algorithms, one must be guided by the differences between the tracking problem and the assignment problem. These multidimensional assignment problems are NP-hard, but the tracking problem must be solved in real-time. The errors in the observations are transferred via filtering and scoring to a certain level of noise in the objective function, so that optimization below this noise level is meaningless. Given these differences, the strategy employed in this work is to construct high quality, feasible, suboptimal solutions of the assignment problems in real-time. The basic scheme is a recursive Lagrangian relaxation similar to the one developed by Frieze and Yadegar [16], [17] for three-dimensional assignment problems. This algorithm is recursive in that a K-dimensional assignment problem is relaxed to a $(K-1)$-dimensional one by incorporating one set of constraints into the objective function using a Lagrangian relaxation of this set. Given a solution of the $(K-1)$-dimensional problem, a feasible solution of the K-dimensional problem is then reconstructed. The

$(K-1)$-dimensional problem is solved in a similar manner and the process is repeated until one reaches the two-dimensional problem, which is solved exactly. The speed and robustness of this scheme are partially due to the decomposition of the large problem into a number of smaller disjoint components, which can be solved independently.

The remainder of the paper is organized as follows. The model problems are explained in §2, followed by the formulation of the multidimensional assignment problem in §3. An overview of the class of recursive Lagrangian relaxation algorithms, along with refinements, is given in §§4 and 5. Results of extensive numerical testing are presented as a case study to demonstrate the speed, robustness, and exceptional quality of the solutions in §6.

2. Model problems and data association. The tracks are assumed to be straight lines in two-dimensional space-time, so that the target velocity is constant. If N denotes the number of targets and $x(j,t)$ is the spatial coordinate of the jth target at time t, the *true tracks* are given by

$$x(j,t) = b_j + m_j t \quad \text{for } j = 1, \ldots, N. \tag{2.1}$$

The observed positions of these targets are given by

$$z(j,t) = x(j,t) + e(j,t) \quad \text{for } j = 1, \ldots, N, \tag{2.2}$$

where the errors $e(j,t)$ are assumed to be independent identically distributed, zero-mean, Gaussian random variables. At a discrete set of *scan times* $\{t_k\}_{k=1}^{K}$ ($t_1 \leq t_2 \leq \cdots \leq t_K$) pictures of the objects are taken, and the spatial positions are recorded as $\{z_{i_k}^k\}_{i_k=1}^{N}$ for each scan time t_k. (The relation between the observation $z_{i_k}^k$ and the particular target generating it is now assumed to be unknown.) The remaining assumptions are that the probability of detection is one, i.e., all the targets are detected at every scan; the probability of false alarm is zero, i.e., all observations belong to true targets; all objects are assumed to initiate at time $t = 0$ and to have no termination at a later time; and there are always as many observations as targets.

Since the noise $e(j,t_k)$ is assumed to be Gaussian with zero mean and standard deviation σ, 99.7% of the observations will lie within 3σ of the true tracks. Any observation error lying outside the interval $[-3\sigma, 3\sigma]$ is returned to the nearest end point, i.e., if $e(j,t_k) > 3\sigma$ ($< -3\sigma$), $e(j,t_k)$ is reset to 3σ (-3σ, respectively). The reason is that in the current model, the probability of detection is assumed to be one and the probability of false alarm is zero, so there is no method for dealing with missed detections in the gating procedure. These restrictions are removed in the more general model [35].

In what follows, the term *track of observations* [6], [28] is used to denote a sequence of observations $\{z_{i_1}^1, \ldots, z_{i_K}^K\}$, one from each scan. Assuming that the parameters N, $b_{\min}$, $b_{\max}$, $m_{\min}$, $m_{\max}$, $\{t_k\}_{k=1}^{K}$, and σ are known, the *data association problem* is to partition the $N \times K$ observations into N tracks of observations, so that the corresponding N filtered tracks represent the N true tracks.

3. Problem formulation. Given K scans of observations, the data association problem is next formulated as a K-dimensional assignment problem using the following four stages: *gating*, *filtering*, *scoring*, and *assignment formulation*. Each of these stages is considered in the following four subsections.

3.1. Gating. For N tracks and K scans, the potential number of tracks of observations is N^K. To substantially reduce this number, we use two gating procedures:

a coarse and computationally cheap phase, followed by a finer but more expensive one. Both use a maximum error r, which is generally set to 3σ in all the computations. The following definition is needed for the description of the first gating procedure.

DEFINITION. FORWARD CONE. Let r be the maximum error, $k \in \{1, \dots, K\}$, and let $z_{i_k}^k$ be an observation at time t_k. Define b_1 and b_2 by $z_{i_k}^k = m_{\min} t_k + b_1 = m_{\max} t_k + b_2$. Then, a *forward cone* $C(z_{i_k}^k, k)$ from the observation $z_{i_k}^k$ is defined by $C(z_{i_k}^k, k) = \emptyset$ when $k = K$ and by $C(z_{i_k}^k, k) = \{z_{i_l}^l \mid k < l \le K,\ m_{\min} t_l + b_1 - 2r \le z_{i_l}^l \le m_{\max} t_l + b_2 + 2r\}$ for $1 \le k \le K - 1$.

The first gating procedure is recursive, and proceeds by constructing a series of forward cones. It is assumed that the scan times are globally available, i.e., the value k determines t_k.

Phase One of Gating

Let **list** *be the variable that will contain all the feasible tracks of observations. The recursive procedure* **Gate** *is defined below.*

```
set list to empty
for all $i = 1, \dots, N$ do
    $\hat{C} = C(z_{i_1}^1, 1)$
    track = $\{z_{i_1}^1\}$
    gate($z_{i_1}^1, \hat{C}, 1, K$, track, list)
gate($z, C, k, K$, track, list)
    if $(k = K)$
        add track to list
    else
        $I = \{i \mid z_i^{k+1} \in C\}$
        for all $i \in I$ do
            $\hat{C} = C \cap C(z_i^{k+1}, k+1)$
            gate($z_i^{k+1}, \hat{C}, k+1, K$, track $\cup \{z_i^{k+1}\}$, list)
```

Given a track of observations from the first phase, say $\{z_{i_k}^k\}_{k=1}^K$, the second phase of gating is to determine whether or not there is a slope m and an intercept b that satisfy the various constraints.

Phase Two of Gating

A given track of observations $\{z_{i_k}^k\}_{k=1}^K$ *passes gating*
if there is a feasible solution to the following inequalities:

$m_{\min} \le m \le m_{\max}$, $b_{\min} \le b \le b_{\max}$,
and
$-r \le mt_k + b - z_{i_k}^k \le r$ *for* $k = 1, \dots, K$.

Otherwise, the track of observations fails gating.

Any track of observations $\{z_{i_k}^k\}_{k=1}^K$ that fails gating is not allowed to appear in the final partition of the observations into tracks.

3.2. Filtering. Given a track of observations $\{z_{i_k}^k\}_{k=1}^K$ that has passed gating, the next problem is to determine a corresponding *filtered track.* The slope $m = m_{i_1,\dots,i_K}$ and intercept $b = b_{i_1,\dots,i_K}$ of the filtered track $x = mt + b$ are the minimizers of the least squares problem

$$\text{(3.1)} \qquad \text{minimize} \ \sum_{k=1}^{K} [z_{i_k}^k - mt_k - b]^2.$$

3.3. Scoring. The score or price associated with assigning the track of observations $\{z_{i_k}^k\}_{k=1}^K$ to the filtered track $x = m_{i_1,\ldots,i_K}t + b_{i_1,\ldots,i_K}$ is defined to be

(3.2)
$$c_{i_1,i_2,\ldots,i_K} = \begin{cases} \sum_{k=1}^K (z_{i_k}^k - m_{i_1,\ldots,i_K}t_k - b_{i_1,\ldots,i_K})^2 & \text{if } (z_{i_1}^1,\ldots,z_{i_K}^K) \text{ passes gating,} \\ \infty & \text{if } (z_{i_1}^1,\ldots,z_{i_K}^K) \text{ fails gating.} \end{cases}$$

Since the errors in the observations are assumed to be independent identically distributed Gaussian random variables with zero mean, this sum of distances is equivalent to using the a priori negative log likelihood estimation technique. The resulting solution has the same maximum likelihood in both a priori and a posteriori estimates [13, Chap. 9].

3.4. The assignment problem formulation. The data association problem of partitioning the observations into tracks can now be formulated as a multidimensional assignment problem. Define a 0–1 variable $z_{i_1,\ldots,i_K}$ for a track of observations $(z_{i_1}^1,\ldots, z_{i_K}^K)$ by

$$(3.3) \qquad z_{i_1,\ldots,i_K} = \begin{cases} 1 & \text{if } (z_{i_1}^1,\ldots,z_{i_K}^K) \text{ is assigned to its filtered track,} \\ 0 & \text{otherwise.} \end{cases}$$

One may preassign $z_{i_1,\ldots,i_K}$ to 0 if $(z_{i_1}^1,\ldots,z_{i_K}^K)$ fails gating. Then the constraints in the problem arise from the requirement that the observation $z_{i_k}^k$ on scan k belongs to exactly one track, which can be stated mathematically as

$$(3.4) \qquad \sum_{i_1=1}^N \sum_{i_2=1}^N \cdots \sum_{i_{k-1}=1}^N \sum_{i_{k+1}=1}^N \cdots \sum_{i_{K-1}=1}^N \sum_{i_K=1}^N z_{i_1,\ldots,i_k,\ldots,i_K} = 1$$

for $i_k = 1,\ldots,N$ and $k = 1,\ldots,K$. Then the problem of assigning tracks of observations to tracks in such a way that each observation is assigned to exactly one track and the overall score (cost) is minimized can be formulated as the following *multidimensional assignment problem*:

$$(3.5) \qquad \begin{aligned} \text{minimize} \quad & \sum_{i_1=1}^N \cdots \sum_{i_K=1}^N c_{i_1,\ldots,i_K} z_{i_1,\ldots,i_K} \\ \text{subject to} \quad & \sum_{i_2=1}^N \cdots \sum_{i_K=1}^N z_{i_1,\ldots,i_K} = 1, \quad i_1 = 1,\ldots,N, \\ & \sum_{i_1=1}^N \cdots \sum_{i_{k-1}=1}^N \sum_{i_{k+1}=1}^N \cdots \sum_{i_K=1}^N z_{i_1,\ldots,i_K} = 1, \\ & \text{for } i_k = 1,\ldots,N, \quad k = 2,\ldots,K-1, \\ & \sum_{i_1=1}^N \cdots \sum_{i_{K-1}=1}^N z_{i_1,\ldots,i_K} = 1, \quad i_K = 1,\ldots,N \\ & z_{i_1,\ldots,i_K} \in \{0,1\}, \quad i_1,\ldots,i_K = 1,\ldots,N. \end{aligned}$$

4. A recursive Lagrangian relaxation algorithm.

Lagrangian relaxation originally gained prominence as a method for obtaining tight bounds for a branch and bound algorithm in Held and Karp's highly successful work on the travelling salesman problem [22], [23]. Overviews of this methodology can be found in the works of Geoffrion [19], Fisher [14], [15], Shapiro [40], Gavish [18], Nemhauser and Wolsey [30], and the references therein. Our motivation for using this approach comes from the the computational experience of several investigators [15], who have found that the duality gap is particularly small and the bounds particularly good when one only uses equality constraints in the relaxation. Other than the integrality constraints, the multidimensional assignment problem (3.5) is exactly of this form. The particular Lagrangian relaxation scheme developed in this work is motivated by the relaxation scheme of Frieze and Yadegar [16], [17] for three-dimensional assignment problems and incorporates the conjugate subgradient algorithms of Wolfe [42], [43] for nonsmooth optimization. We also use an adaptation of the reverse auction algorithm of Bertsekas, Castañon, and Tsaknakis [11], [12] for the two-dimensional assignment problems. As stated in the introduction, the relaxation algorithm is recursive in that a K-dimensional assignment problem is relaxed to a $(K-1)$-dimensional one by incorporating one set of constraints into the objective function using a Lagrangian relaxation of this set. Given a solution of the $(K-1)$-dimensional problem, a feasible solution of the K-dimensional problem is then reconstructed. The $(K-1)$-dimensional problem is solved in a similar manner, and the process is repeated until it reaches the two-dimensional problem, which is solved by the reverse auction algorithm [11], [12].

Consider again the multidimensional assignment problem (3.5). Although any set of the constraints can be used, the description here will be based on the relaxation of the last set of constraints in (3.5), which is given by

$$
\begin{aligned}
\phi(u) = \text{minimize}\quad & \sum_{i_1=1}^{N} \cdots \sum_{i_K=1}^{N} c_{i_1,\ldots,i_K} z_{i_1,\ldots,i_K} \\
& - \sum_{i_K=1}^{N} u_{i_K} \left(\sum_{i_1=1}^{N} \cdots \sum_{i_{K-1}=1}^{N} z_{i_1,\ldots,i_K} - 1 \right) \\
\text{subject to}\quad & \sum_{i_2=1}^{N} \cdots \sum_{i_K=1}^{N} z_{i_1,\ldots,i_K} = 1, \quad i_1 = 1, \ldots, N, \\
& \sum_{i_1=1}^{N} \cdots \sum_{i_{k-1}=1}^{N} \sum_{i_{k+1}=1}^{N} \cdots \sum_{i_K=1}^{N} z_{i_1,\ldots,i_K} = 1, \\
& \text{for } i_k = 1, \ldots, N, \quad k = 2, \ldots, K-2, \\
& \sum_{i_1=1}^{N} \cdots \sum_{i_{K-2}=1}^{N} \sum_{i_K=1}^{N} z_{i_1,\ldots,i_K} = 1, \quad i_{K-1} = 1, \ldots, N \\
& z_{i_1,\ldots,i_K} \in \{0,1\}, \quad i_1, \ldots, i_K = 1, \ldots, N,
\end{aligned}
\tag{4.1}
$$

where u is the multiplier vector associated with the last set of the constraints. This problem is easily converted to a more obvious $(K-1)$-dimensional assignment problem by first defining, for each $(i_1, \ldots, i_{K-1})$, an index $m = m(i_1, \ldots, i_{K-1})$ and a new cost function $d_{i_1,\ldots,i_{K-1}}$ by

$$
\begin{aligned}
m &= \arg\min\{c_{i_1,\ldots,i_{K-1},i_K} - u_{i_K} \mid i_K = 1, \ldots, N\}, \\
d_{i_1,\ldots,i_{K-1}} &= c_{i_1,\ldots,i_{K-1},m} - u_m.
\end{aligned}
\tag{4.2}
$$

An equivalent problem is

$$
(4.3)\qquad
\begin{aligned}
\hat{\phi}(u) = \text{minimize}\quad & \sum_{i_1=1}^{N}\cdots\sum_{i_{K-1}=1}^{N} d_{i_1,\dots,i_{K-1}} y_{i_1,\dots,i_{K-1}} \\
\text{subject to}\quad & \sum_{i_2=1}^{N}\cdots\sum_{i_{K-1}=1}^{N} y_{i_1,\dots,i_{K-1}} = 1, \quad i_1 = 1,\dots,N, \\
& \sum_{i_1=1}^{N}\cdots\sum_{i_{k-1}=1}^{N}\sum_{i_{k+1}=1}^{N}\cdots\sum_{i_{K-1}=1}^{N} y_{i_1,\dots,i_{K-1}} = 1, \\
& \text{for } i_k = 1,\dots,N, \quad k = 2,\dots,K-2, \\
& \sum_{i_1=1}^{N}\cdots\sum_{i_{K-2}=1}^{N} y_{i_1,\dots,i_{K-1}} = 1, \quad i_{K-1} = 1,\dots,N \\
& y_{i_1,\dots,i_{K-1}} \in \{0,1\}, \quad i_1,\dots,i_{K-1} = 1,\dots,N.
\end{aligned}
$$

Once (4.3) is solved, the solution of (4.1) is easily recovered via

$$
(4.4)\qquad z_{i_1,\dots,i_K} = \begin{cases} y_{i_1,\dots,i_{K-1}} & \text{if } i_K = m(i_1,\dots,i_{K-1}), \\ 0 & \text{otherwise.} \end{cases}
$$

The numerical algorithm begins with the construction of a sequence of multipliers $\{u^n\}_{n=1}^{\infty}$ such that $\{\phi(u^n)\}_{n=1}^{\infty}$ is monotone increasing and $\lim_{n\to\infty}\phi(u^n) = \bar{\phi} \equiv \sup\{\phi(u) \ : \ u \in \mathbb{R}^n\}$. (The initial approximation is $u^1 = 0$, as suggested in the work of Bazaraa and Goode [7].) For each multiplier u^n, a feasible solution z^n of the original problem is recovered by a scheme described below, and satisfies

$$
(4.5)\qquad \phi(u^n) \le \bar{\phi} \le v(\bar{z}) \le v(z^n),
$$

where $\bar{z}$ is the optimal solution of (3.5) and $v(z)$ is the value of the objective function for a feasible solution z [19].

The function $\phi(u)$ is a concave, piecewise linear, continuous function, so that maximizing $\phi(u)$ is a problem of nonsmooth optimization. One of the most widely used methods is the subgradient algorithm [9], [20], [21], [30]. We have, however, found that *our* implementation of the conjugate subgradient method of Wolfe [42], [43] is computationally superior to *our* implementation of the subgradient algorithm, and thus we use the former. (For completeness, a brief description of Wolfe's algorithm is included in Appendix A.) Note that each time $\phi(u)$ is evaluated, a $(K-1)$-dimensional assignment problem must be solved. If $K > 3$, this problem is solved by relaxation or by techniques discussed in §5, but if $K = 3$, the reduced assignment problem is two-dimensional and is solved by the reverse auction algorithm [11], [12].

Given a multiplier u^n generated in the course of maximizing $\phi(u)$ in (4.1), the next problem is to recover a feasible solution z^n of (3.5). For notational convenience, the superscript n will be dropped. Thus, let u (u^n) be a multiplier as in (4.1), and let $\hat{z} = \hat{z}(u)$ be a corresponding feasible solution of the minimization problem (4.1), which is obtained in the course of evaluating $\phi(u)$. The objective is to recover a feasible solution Z (z^n) of (3.5). Since $\hat{z}$ satisfies the first $K-1$ sets of constraints in the original problem (3.5), it is feasible for (3.5) if the Kth set of constraints is satisfied. In this case, the recovered solution is taken to be $Z = \hat{z}$. If $\hat{z}$ does not satisfy the last set of constraints, a procedure is needed to recover a feasible solution Z. Motivated by the work of Frieze and Yadegar [16], [17], we reconstruct this feasible solution of (3.5) so that it agrees with $\hat{z}$ in the first $K-1$ dimensions. Since there are generally many options, we choose the one that minimizes the objective function in (3.5) over all such choices.

Let $\{(j, i_2(j), \ldots, i_{K-1}(j))\}_{j=1}^{N}$ be an enumeration of those first $K-1$ indices of $\hat{z}$ for which $\hat{z}_{j,i_2(j),\ldots,i_{K-1}(j),k} = 1$ for some $k = 1, \ldots, K$. Define

$$d_{jk} = c_{j,i_2(j),\ldots,i_{K-1}(j),k} \quad \text{for } k = 1, \ldots, N$$

and let W denote a solution of the two-dimensional assignment problem

$$\text{(4.6)} \qquad \begin{aligned} &\text{minimize} && \sum_{j=1}^{N}\sum_{k=1}^{N} d_{jk} w_{jk} \\ &\text{subject to} && \sum_{k=1}^{N} w_{jk} = 1, \quad j = 1, \ldots, N, \\ & && \sum_{j=1}^{N} w_{jk} = 1, \quad k = 1, \ldots, N, \\ & && w_{jk} \in \{0,1\}, \quad j,k = 1, \ldots, N. \end{aligned}$$

The recovered feasible solution Z is defined by

$$\text{(4.7)} \qquad Z_{i_1,\ldots,i_K} = \begin{cases} 1 & \text{if } \hat{z}_{i_1,\ldots,i_{K-1},k} = 1 \quad \text{for some } k \text{ and } W_{i_1 i_K} = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The feasible solution Z is optimal in that it is the minimizer of the objective function in (3.5) over all feasible solutions of (3.5) that make the same assignments as $\hat{z}$ in the first $K-1$ coordinates.

This recovery procedure has been formulated for the assignment problem with no variables preassigned to zero. If some of the variables are preassigned to zero, some variables in (4.6) could also be assigned the value zero. This might cause (4.6) to have no feasible solution, in which case the recovery algorithm will treat preassigned variables as free variables to be assigned a zero or one, but with infinite cost coefficients. As a result, as many assignments are made as possible.

Three *termination* criteria are used for the algorithm: cumulative step length, duality gap, and the number of steps taken. The first criterion, the cumulative step length, is described by Wolfe [42]. This criterion does not depend on having an estimate of the value of the optimal solution and is an integral part of the conjugate subgradient algorithm; the algorithm is stopped if several very small steps are taken in succession.

As discussed earlier, it is the duality gap $[\bar{\phi}, v(\bar{z})]$ that appears to be particularly small for relaxations of equality constraints [15]. Since $\phi(u^n) \leq \bar{\phi} \leq v(\bar{z}) \leq v(z^n)$, one can use the distance $v(z^n) - \phi(u^n)$ to estimate the closeness of $v(z^n)$ to the optimum $v(\bar{z})$. Thus the second termination criterion is to terminate when the distance $v(z^n) - \phi(u^n)$ is sufficiently small [20]. If the algorithm has not yet succeeded in computing a feasible solution of the assignment problem, the value of the objective function is assumed to be infinite, so this criterion cannot be satisfied.

The third termination criterion is the maximum number of steps the main relaxation algorithm is allowed to take. Two limits for the number of steps are used. If a feasible solution for the assignment problem has been obtained, only a small number of steps is taken. If no feasible solution has yet been found, the iterations are continued until the second limit is reached or a feasible solution is computed. This third termination criterion is usually the one satisfied.

5. Preprocessing and algorithm refinements.

The relaxation scheme is slow at times due to local effects, such as regions of high contention where many tracks cross one another. A partial resolution to this difficulty is to decompose these multidimensional assignment problems into disjoint components that can then be solved independently. After the decomposition, a branch and bound technique is used to solve small components for the following reason. The overhead required in setting up the relaxation algorithm makes branch and bound more efficient on small components. Furthermore, relaxations of these sparse K-dimensional assignment problems introduces additional sparsity, and frequently the problems decompose into a number of small components that are solved optimally without having to relax all the way back to the two-dimensional problem.

5.1. Decomposition into disjoint components.

Consider a graph with the vertices being the observations. Two vertices will be connected by an edge if they belong to successive scans and are part of the same feasible track of observations. Connected components of the graph are then easily found by constructing a spanning forest via a depth-first search. A detailed algorithm can be found in the book by Aho, Hopcroft, and Ullman [1, §5.2]. The following modification of this algorithm uses the additional information that all the observations in a track of observations belong to the same component, thereby increasing the speed of the algorithm.

Decomposition algorithm

```
Given the observations in the list LO and the tracks of observations in the
list LFT, the algorithm proceeds as follows:
Set all the observations and tracks to unmarked
while there are unmarked observations do
    pick any unmarked observation and mark it
    repeat
        mark all the unmarked tracks that pass through a marked observation
        mark all the unmarked observations that belong to a newly marked track
    until no marked/unmarked combinations are left
    identify all the marked observations and tracks as a new component
        and remove them from the problem
```

Decomposition of a different type might be based on the identification of bi- and triconnected components [1] of a graph, if they exist. For example, an algorithm such as that given by Aho, Hopcroft, and Ullman [1, §5.2] might be used to identify any biconnected components, and then an enumeration or branch and bound scheme could then be based on an enumeration of the connections between larger components.

5.2. Branch and bound algorithm.

Although general descriptions of branch and bound techniques can be found in several references [3], [10], [30], [38], a brief description of our algorithm is given in this subsection for completeness.

Let `costs` $= \{c_i \mid c_i = c_{i_1,\ldots,i_K}, i_k = 1,\ldots,N, k = 1,\ldots,K\}$ be a list of sorted cost coefficients, and let `list` $= \{x_i \mid x_i = x_{i_1,\ldots,i_K} \textit{ if } c_i = c_{i_1,\ldots,i_K}\}$ be the corresponding list of variables. If the problem is sparse, the variables preassigned to zero are taken to have infinite costs and are not to be included in `list` and `costs`. Each variable in list can have one of the three values : zero, one, or free. After completion, the variable solution will contain the optimal solution, while the variable bound will contain its value. If the solution = $\emptyset$, then every feasible solution requires that some variables that have been preassigned to zero be changed to one. The algorithm then proceeds as follows by recursively calling the procedure **BranchBound**.

```
Branch and Bound Algorithm
set inList to |list|
for all $x_i \in$ list do
    set $x_i$ to free
set bound to $\infty$
set value to 0
set solution to $\emptyset$
set trySolution to $\emptyset$
set inSolution to 0
BranchBound(list, costs, inList, 0, $K$,
                solution, trySolution, inSolution, value, bound)
BranchBound(list, costs, n, current, $K$,sol, try, m, value, bound)
if (m = $K$)
    set sol = try
    set bound = value
else
    set $I = \{i \mid$ current $< i \leq$ n and $x_i$ is free$\}$
    for all $i \in I$ do
        if value + $c_i \geq$ bound
            return
        set $x_i = 1$
        let $R(x_i) = \{x_j \mid x_j$ is free and $\exists\, k$ s.t. $i_k = j_k$
                where $x_i = x_{i_1,\ldots,i_K}$ and $x_j = x_{j_1,\ldots,j_K}\}$
        for all $x \in R(x_i)$ do
            set $x$ to 0
        set locValue = value
        let $\hat{I} = \{i \in I \mid x_i$ is free$\}$
        if $\hat{I} \neq \emptyset$
            $j = \min\{j \in \hat{I}\}$
            let locValue = value + $(K -$ m$)c_j$
        if (locValue < bound)
            BranchBound(list, costs, n, $i$, $K$,
                sol, try $\cup \{x_i\}$, m + 1, value + $c_i$, bound)
        set $x_i = 0$
        for all $x \in R(x_i)$ do
            set $x$ to free
```

6. Numerical experiments: A case study. In this section we present some representative numerical results from our extensive testing of the problems and algorithms developed in the earlier sections. Twenty test problems were randomly generated, as discussed in §2, with the following parameters: the number of targets $N = 100$, the initial x-intercepts $\{b_j\}_{j=1}^{N}$ and slopes $\{m_j\}_{j=1}^{N}$ are chosen randomly using a uniform distribution over $[b_{\min}, b_{\max}] = [0, 1000]$ and $[m_{\min}, m_{\max}] = [-0.2, 0.2]$, respectively, the *maximum error* of 3σ ranged from 0.1 to 5.0, and the time interval between observations is 40 seconds. These problems are scale-invariant for $m_{\max}\Delta t = 0.2 \times 40$, where Δt denotes the time between scans. Thus if the observations are taken every five seconds, the slopes can range between -1.6 and 1.6. All computations were performed on an IBM RS/6000-320. Tables 1 through 6 describe the complexity of the tracking and assignment problems, and Tables 7 through 11 present the results and performance measurements of the algorithms. The "—" in the bottom right-hand corner of these tables indicates that a number of problems failed to run due to memory limitations, so that the averages were omitted.

Tables 1 and 2 give the number of variables after the first and second phases of gating. The number of variables given in Table 1 is also the total number of variables that are ever treated explicitly. Table 1 shows that the first gating procedure significantly reduces the number of tracks of observations that have to be considered by the second gating procedure. For example, for eight scans, less than 10^6 tracks of observations, which is a tiny fraction of the 10^{16} possible tracks, are submitted to the second gating procedure.

TABLE 1
Problem size after phase one of gating.

Max. error	3 scans	4 scans	5 scans	6 scans	7 scans	8 scans
0.1	1050	2768	8014	25350	94882	339047
0.5	1103	3013	9154	29828	113698	401806
1.0	1165	3309	10239	35767	142817	523148
2.0	1300	4025	13694	51877	215143	893451
3.0	1428	4697	17802	72595	272382	—
4.0	1564	5559	22620	99524	265042	—
5.0	1729	6536	28734	103955	—	—

TABLE 2
Problem size after phase two of gating.

Max. error	3 scans	4 scans	5 scans	6 scans	7 scans	8 scans
0.1	430	516	618	717	817	915
0.5	553	649	774	912	1073	1250
1.0	703	916	1176	1498	1953	2663
2.0	986	1748	2992	4982	8448	15185
3.0	1239	2852	6408	14225	27792	—
4.0	1463	4139	11617	32217	55523	—
5.0	1671	5497	18359	47660	—	—

Tables 3 and 4 give the average numbers of small and large components. A small component is one that has less than 19 variables and no more than 3 observations per scan; otherwise, a component is classified as large. (This empirical distinction is based on whether relaxation or branch and bound will solve the problem faster.)

Note that as the measurement errors increase, the number of small components drops. The reason is that as the error size increases, more tracks interact, i.e., share

TABLE 3
Number of small components.

Max. error	3 scans	4 scans	5 scans	6 scans	7 scans	8 scans
0.1	67	87	89	90	90	92
0.5	30	39	41	43	42	43
1.0	20	20	19	21	20	22
2.0	12	10	9	9	10	13
3.0	8	6	5	6	8	—
4.0	6	4	3	3	5	—
5.0	4	3	3	3	—	—

TABLE 4
Number of large components.

Max. error	3 scans	4 scans	5 scans	6 scans	7 scans	8 scans
0.1	7	2	2	2	3	3
0.5	12	10	11	10	10	10
1.0	13	12	12	11	10	8
2.0	14	12	11	9	7	6
3.0	13	11	9	8	6	—
4.0	13	10	9	7	5	—
5.0	12	9	8	6	—	—

observations, and thus small components merge into large ones. Large component behavior is more interesting. As the errors increase, the number of large components increases at first and then decreases. This is due to component merging. For smaller errors, small components merge to make new large components. For larger errors, large components are lost by merging, and there are few small components remaining that can merge to form new large components. Finally, as the observation error continues to increase the problems eventually become dense and do not decompose. Group tracking techniques [13] then become applicable.

If the measurement error is kept constant and the number of scans is increased, both the number of small components and the number of variables increase, but the number of large components decreases. This behavior can be explained with the aid of Tables 5 and 6.

As more scans are added, the average and largest component sizes increase. Also, if the error is kept constant, additional scans will mean tighter gates. Thus, most of the tracks will interact with fewer neighbors and components will be slower to merge. However, in areas where tracks lie very close to one another, the number of feasible tracks in a component will tend to grow exponentially as the scans are added. The evidence for this can be found by comparing the maximum component sizes with the total sizes of the assignment problems: for larger errors and more scans, more than half of the variables belong to a single component. This leads to the conclusion that there is a "best" level of information, i.e., a best number of scans for a given noise level. Too many scans might not improve the solution quality, but could lead to the exponential increase of the number of variables in the largest components. This suggests that

TABLE 5
Average large component size.

Max. error	3 scans	4 scans	5 scans	6 scans	7 scans	8 scans
0.1	20	19	23	26	31	36
0.5	33	39	47	63	83	102
1.0	46	64	90	135	200	367
2.0	69	144	290	599	1178	2739
3.0	95	263	733	1936	4521	—
4.0	116	415	1405	4845	12103	—
5.0	147	617	2501	8582	—	—

TABLE 6
Largest component size.

Max. error	3 scans	4 scans	5 scans	6 scans	7 scans	8 scans
0.1	33	21	26	29	38	45
0.5	77	88	106	154	212	277
1.0	121	177	282	418	662	1160
2.0	214	502	1080	2199	4199	8500
3.0	303	980	2877	7202	16960	—
4.0	405	1520	5753	18395	35622	—
5.0	492	2190	9790	27713	—	—

additional scans might more effectively be addressed within the context of a sliding window of observations [5] and track extension, as opposed to track initiation.

To assess the quality of the solutions, one first needs to distinguish between two issues: the performance of the algorithms in obtaining high-quality suboptimal solutions to the multidimensional assignment problems, and the quality of the identified tracks. We discuss these issues separately, present tables for each, and then present the timings.

The objective function in the assignment problem contains noise due to the observation error, which is transferred to the cost coefficients via filtering. Thus the *first* test is to determine whether the algorithms compute the solutions to or below this noise level. Since the true tracks and corresponding true tracks of observations used to generate the problem are available, one can compute an objective function value for the true tracks of observations scored against the true tracks as well as the true filtered tracks, which are obtained from the true track of observations by filtering. If the algorithms have computed a solution to or below the noise level, the score of the computed solution should be at or below that of the generating solution. Indeed, the following three tables show this to be the case. Note that the true filtered tracks fit the true tracks of observations *much* better than the true tracks, as can be seen from Tables 7 and 8.

For lower observation errors the algorithms yield solutions with a score as good as or slightly better than those of the true tracks of observations scored against either the true tracks or the true filtered tracks, while at higher observation errors the scores are significantly lower. On the average the algorithm is computing at or below the noise

TABLE 7
Average score for true tracks.

Max. error	3 scans	4 scans	5 scans	6 scans	7 scans	8 scans
0.1	≤ 0.4	≤ 0.4	1	1	1	1
0.5	9	11	14	16	19	22
1.0	34	45	55	66	77	87
2.0	138	179	222	265	309	348
3.0	310	403	500	596	695	—
4.0	550	716	890	1064	1246	—
5.0	862	1120	1390	1642	—	—

TABLE 8
Average score for true filtered tracks.

Max. error	3 scans	4 scans	5 scans	6 scans	7 scans	8 scans
0.1	≤ 0.4	≤ 0.4	≤ 0.4	≤ 0.4	1	1
0.5	3	6	8	11	14	16
1.0	12	22	33	44	55	65
2.0	48	91	133	177	221	262
3.0	108	206	302	401	498	—
4.0	191	367	532	702	888	—
5.0	297	570	825	1082	—	—

TABLE 9
Average score for the computed tracks.

Max. error	3 scans	4 scans	5 scans	6 scans	7 scans	8 scans
0.1	≤ 0.4	≤ 0.4	≤ 0.4	≤ 0.4	1	1
0.5	3	5	8	11	13	16
1.0	10	21	31	41	52	62
2.0	37	80	120	164	206	244
3.0	76	169	262	356	443	—
4.0	120	284	445	593	773	—
5.0	171	413	657	914	—	—

level in the problem. Although the optimal solution is surely being computed in many cases, we give an example in Appendix B which shows that the relaxation algorithm and recovery procedure need not always produce the optimal solution, regardless of the choice of the multipliers in the relaxed problem.

The *second* criterion is to judge how well the tracks of observations from the solution of the multidimensional assignment problem identify the true tracks. One way of measuring this identification is to determine how well the tracks of observations are able to predict the location of the true tracks on the next scan. Although this process of extrapolation is not always reliable, it is part of predicting the future location of the objects. Let $\{z^1_{i_1}(i), \ldots, z^K_{i_K}(i)\}_{i=1}^N$ be an enumeration of the N tracks of observations computed from the solution of the multidimensional assignment problem. Then, given the N true tracks, and these N computed tracks of observations, the first task is to assign the computed tracks of observations to the true tracks, and this is accomplished by using the following two-dimensional assignment problem.

Let $\{m_j t + b_j\}_{j=1}^{N}$ be an enumeration of the N true tracks. For i and $j = 1, \ldots, N$ define

$$c_{ij} = \sum_{k=1}^{K} |z_{i_k}^{k}(i) - m_j t_k - b_j|^2$$

and

$$x_{ij} = \begin{cases} 1 & \text{if the } i\text{th track of observations is assigned to true track } j, \\ 0 & \text{if the } i\text{th track of observations is not assigned to true track } j. \end{cases}$$

With this definition of the 0–1 variables x_{ij} and cost coefficients c_{ij}, one then solves the corresponding two-dimensional assignment problem to determine the best assignment. (This assignment problem is the same as the one formulated in (4.6) with d_{ij} and w_{ij} replaced by c_{ij} and x_{ij}, respectively.) Next, let $\{x_{i(j)j}\}_{j=1}^{N}$ be an enumeration of all those 0–1 variables x_{ij} for which $x_{ij} = 1$, and let $\{z_{i_1}^{1}(i(j)), \ldots, z_{i_K}^{K}(i(j))\}$ be a track of observations assigned to true track j. Now, determine the set of all slopes m and intercepts b satisfying $m_{\min} \leq m \leq m_{\max}$, $b_{\min} \leq b \leq b_{\max}$, and $-r \leq mt_k + b - z_{i_k}^{k}(i(j)) \leq r$ for $k = 1, \ldots, K$. The range of the straight lines $x = mt + b$ as m and b vary over this set determines a gate at time $t = t_{K+1}$. The jth true track has been identified if the true target position is within this gate.

Since the following tables represent an average over 20 test problems and there are 100 tracks per problem, an identification of 99.9% implies that all but one track out of 1000 was correctly identified.

TABLE 10
Percentage of correctly identified tracks.

Max. error	3 scans	4 scans	5 scans	6 scans	7 scans	8 scans
0.1	99.9	99.9	99.9	99.9	99.9	99.9
0.5	99.6	99.9	99.8	99.8	99.8	99.8
1.0	98.8	99.9	99.6	99.7	99.7	99.7
2.0	98.6	99.6	99.3	99.5	99.7	99.4
3.0	98.3	98.5	98.8	99.2	99.1	—
4.0	98.0	97.6	97.9	97.9	98.6	—
5.0	97.5	97.2	97.2	97.2	—	—

One would expect that as more scans are added, the identification should improve. Table 3 shows this to be the case, but not always. The reason for this phenomena is quite simple and generic. Regions of high contention cause the identification to degrade. Although additional scans of observations generally resolve this local difficulty, additional regions of high contention appear in different areas of space on subsequent scans. Also, for a fixed maximum error, the gates used in the identification become smaller as more scans are added, so that the identification criterion becomes tighter as the number of scans increases. We have also investigated those tracks that have not been identified through the use of computer visualization. In almost every instance, the tracks lie just outside of our criteria. In the next table we present the timings for the above problems.

We have placed much emphasis on real-time identification; let us now be specific. Suppose a radar sweep takes 5 to 10 seconds. The objective then is to process as many scans as possible between sweeps to improve identification, and to solve the problem in the allotted time. The above table gives some idea of the present capability for 100 targets. Possible improvements in the algorithms include the use of "hot starts" and a sliding window implementation for track extension. The reason for the former is that one has a high-quality suboptimal or optimal solution of a closely related K- or $K-1$-dimensional problem, and such a solution should be used as a "hot start" for the given problem. As an example of this latter possibility, consider the case of maximum error of 2.0. At four scans we have identified 99.6% of the targets, but this information is not used in solving the subsequent assignment problems for five through eight scans. Once a relatively high percentage of the tracks have been identified, one might use a sliding window of observations to process the incoming scans, thereby reducing the dimension of the assignment problems and thus improving the timings. Each of these algorithmic enhancements and massive parallelizations of the algorithms is currently under investigation for more general tracking problems [35].

TABLE 11

Solution times in seconds.

Max. error	3 scans	4 scans	5 scans	6 scans	7 scans	8 scans
0.1	0.02	0.03	0.04	0.04	0.04	0.05
0.5	0.03	0.04	0.05	0.07	0.09	0.11
1.0	0.04	0.07	0.10	0.17	0.22	0.39
2.0	0.06	0.13	0.28	0.49	0.91	3.00
3.0	0.08	0.22	0.50	1.40	3.27	—
4.0	0.09	0.30	0.87	3.77	8.12	—
5.0	0.09	0.46	1.37	5.72	—	—

Appendix A. The conjugate subgradient algorithm.

A key step in the Lagrangian relaxation is the maximization of the concave, piecewise linear, and continuous functions $\phi(u)$ in (4.1). For completeness, a brief outline of our implementation of Wolfe's conjugate subgradient algorithm [42], [43] is given here.

Some definitions are necessary for the description of the algorithm. Let G be a finite set of vectors, and define $d = \text{Nr } G$ to be a convex combination of the vectors in G that has the minimum 2-norm. The algorithm for computing Nr G is given in the work of Wolfe [42], [43]. The set of all subgradients of ϕ at u will be denoted by $\partial\phi(u)$; however, only a single element of $\partial\phi(u)$ is computed at each stage. Finally, define $g(t) \in \partial\phi(u+td)$ for fixed u and d and any $t \geq 0$,

$$M(t) = \frac{\langle g(t), d\rangle}{|\, d\,|^2}, \quad \text{and} \quad Q(t) = \frac{\phi(u+td) - \phi(u)}{t|\, d\,|^2} \quad \text{for } t > 0.$$

The conjugate subgradient algorithm is as follows.

Let ε, δ, b, $m_2 < m_1 < \frac{1}{2}$, all positive, be given and fixed throughout the algorithm. Let an initial approximation of the multiplier u be given. Initialize the line search parameter t, compute $g \in \partial\phi(u)$, and set $G = \{g\}$ and $a = 0$.

```
loop
    compute $d = Nr\ G$
    if $\mid d \mid < \varepsilon$ then
        if $a \leq \delta$ then
            exit loop (algorithm finished)
        else (reset) set $G = g$ and $a = 0$
    else compute $u_+ = u + td$, $g_+ = g(t)$
        using $m_1$, $m_2$, $M$, and $Q$ accept, double, or halve $t$
        if $t$ was accepted or doubled then
            set $u = u_+$, $g = g_+$, add $g$ and $d$ to $G$, set $a = a + \mid td \mid$
        else if $t \mid d \mid \leq b$ then
                discard $u_+$ and $g_+$, add $g$ and $d$ to $G$
end loop
```

If u is a multiplier such that (4.2) has more than one optimal solution, $d = \text{Nr } G$ is the best choice for the step direction. The conjugate subgradient algorithm belongs to a broader class of algorithms called "bundle methods" [25]. (The set G is called a "bundle.") Methods in this class substitute several recent directions in the bundle for $\partial\phi(u)$. Each time a step is taken, the direction is added to the bundle in some manner. To prevent the bundle from growing too large, oldest directions are discarded as new vectors are added to the bundle. If the bundle size is set to one, bundle methods are equivalent to the subgradient algorithm. In our computations the best bundle size was determined to be 5, but it can be easily changed to suit a particular problem.

Appendix B. An example. This example illustrates the failure of the relaxation algorithm and recovery procedure developed in §4 to produce an optimal solution, regardless of the multipliers chosen. Let $\bar{x}$ be an optimal solution of the K-dimensional assignment problem (3.5), $\hat{x}(u)$ be the optimal solution of the relaxed problem (4.1) for a given multiplier vector u, $\phi(u)$ be the value of the minimum of the relaxed problem (4.1), and $x(u)$ be the recovered feasible solution of (3.5).

Let $K = 4$ and $N = 3$ and define the cost coefficients as follows:

$$\begin{aligned}
&c_{1111} = c_{2222} = c_{3333} = 5,\\
&c_{111k} = c_{222l} = c_{333m} = 10 \quad k = 2,3, \ l = 1,3, \ m = 1,2,\\
&c_{2131} = c_{2132} = c_{1321} = c_{1322} = c_{3211} = c_{3212} = c_{2313} = c_{3123} = c_{1233} = 1,\\
&c_{2133} = c_{1323} = c_{3213} = c_{2311} = c_{2312} = c_{3121} = c_{3122} = c_{1231} = c_{1232} = 15;
\end{aligned}$$

the remaining cost coefficients are set to 50. The optimal solution is $x_{1111} = 1$, $x_{2222} = 1$, $x_{3333} = 1$ with the remaining variables set to zero, and the corresponding objective function value is 15. The objective function values of the remaining feasible solutions are greater than or equal to 17. Consider the relaxation of the last set of constraints, and let u_1, u_2, u_3 be the corresponding multipliers. To recover the optimal solution of the original problem, the solution of the relaxed problem must have $x_{111k} = x_{222l} = x_{333m} = 1$ for some k, l, m with the remaining variables being zero. Notice that the particular choice of k, l, and m is dictated by the multiplier values. Let A denote the relaxed objective function value associated with this set of solutions (of the relaxed problem). Since the cost coefficients are known, A can be expressed as a function of multipliers by the following expression:

$$\begin{aligned}
A = &\min\{5 - u_1, 10 - u_2, 10 - u_3\} + \min\{10 - u_1, 5 - u_2, 10 - u_3\}\\
&+ \min\{10 - u_1, 10 - u_2, 5 - u_3\}.
\end{aligned}$$

It will be shown that, regardless of the choice of multipliers, no solution in this solution set can be an optimal solution of the relaxed problem. To see this, define two other solution sets of the relaxed problem by $x_{213k} = x_{132l} = x_{321m} = 1$ and $x_{231k} = x_{312l} = x_{123m} = 1$, where again the remaining variables are set to zero and the particular choices of k, l, and m depend on the multiplier values. Let B and C denote the relaxed objective function values for these two solution sets. B and C can again be expressed as the functions of multipliers

$$\begin{aligned} B =& 3\min\{1-u_1, 1-u_2, 15-u_3\}, \\ C =& 3\min\{15-u_1, 15-u_2, 1-u_3\}. \end{aligned}$$

We now claim that for any choice of the multipliers, $A > \min\{B, C\}$, i.e., none of the solutions from the first solution set can ever be an optimal solution of the relaxed problem. To see this, consider all the possible multiplier choices. They yield 10 possible values for A. In all cases, either B or C is smaller than A:

$$\begin{aligned}
A =& 5-u_1+10-u_1+10-u_1 = 25-3u_1 > 3-3u_1 \geq B, \\
A =& 5-u_1+10-u_1+5-u_3 = 20-2u_1-u_3 > 17-2u_1-u_3 \geq B, \\
A =& 5-u_1+5-u_2+10-u_1 = 20-2u_1-u_2 > 3-2u_1-u_2 \geq B, \\
A =& 5-u_1+5-u_2+10-u_2 = 20-u_1-2u_2 > 3-u_1-2u_2 \geq B, \\
A =& 5-u_1+5-u_2+5-u_3 = 15-u_1-u_2-u_3 > 3-2u_1-u_2 \geq B, \\
A =& 5-u_1+10-u_3+5-u_3 = 20-u_1-2u_3 > 17-u_1-2u_3 \geq C, \\
A =& 10-u_2+5-u_2+10-u_2 = 25-3u_2 > 3-3u_2 \geq B, \\
A =& 10-u_2+5-u_2+5-u_3 = 20-2u_2-u_3 > 17-2u_2-u_3 \geq B, \\
A =& 10-u_3+5-u_2+5-u_3 = 20-u_2-2u_3 > 17-u_2-2u_3 \geq C, \\
A =& 10-u_3+10-u_3+5-u_3 = 25-3u_3 > 3-3u_3 \geq C.
\end{aligned}$$

This argument shows that, regardless of the multiplier choice, none of the solutions from the first solution set can be an optimal solution of the relaxed problem. Thus the optimal solution to the original problem can never be obtained through the relaxation of the last constraint set.

Acknowledgments. We wish to thank T. Barker, R. E. Blahut, M. Munger, and J. Persichetti of IBM for the many stimulating discussions that led to the problem and algorithm developments presented in this work.

REFERENCES

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, 1974.

[2] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice-Hall, Englewood Cliffs, NJ, 1979.

[3] M. L. Balinski, ed., *Mathematical Programming Study* 2: *Approaches to Integer Programming*, North Holland, Amsterdam, 1974.

[4] Y. Bar-Shalom, *Tracking methods in a multitarget environment*, IEEE Trans. Automat. Control, AC-23 (1978), pp. 618–626.

[5] Y. Bar-Shalom, ed., *Multitarget-Multisensor Tracking: Advanced Applications*, Artech House, Dedham, MA, 1990.

[6] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, Boston, MA, 1988.

[7] M. S. BAZARAA AND J. J. GOODE, *A survey of various tactics for generating Lagrangian multipliers in the context of Lagrangian duality*, European J. Oper. Res., 3 (1979), pp. 322–338.

[8] M. S. BAZARAA AND J. J. JARVIS, *Linear Programming and Network Flows*, 2nd ed., John Wiley and Sons, New York, 1989.

[9] M. S. BAZARAA AND H. D. SHERALI, *On the choice of step size in subgradient optimization*, European J. Oper. Res., 7 (1981), pp. 380–388.

[10] E. M. L. BEALE, *Integer programming*, in The State of the Art in Numerical Analysis, E. Jacobs, ed., Academic Press, London, 1977.

[11] D. P. BERTSEKAS, *Linear Network Optimization*, MIT Press, Cambridge, MA, 1991.

[12] D. P. BERTSEKAS, D. A. CASTAÑON, AND H. TSAKNAKIS, *Reverse auction and the solution of inequality constrained assignment problems*, preprint, March 1991.

[13] S. S. BLACKMAN, *Multiple Target Tracking with Radar Applications*, Artech House, Dedham, MA, 1986.

[14] M. L. FISHER, W. D. NORTHUP, AND J. F. SHAPIRO, *Using duality to solve discrete optimization problems: Theory and computational experience*, Math. Programming Stud., 3 (1975), pp. 56–94.

[15] M. L. FISHER, *The Lagrangian relaxation method for solving integer programming problems*, Management Sci., 27 (1981), pp. 1–18.

[16] A. M. FRIEZE, *A bilinear programming formulation of the* 3*-dimensional assignment problem*, Math. Programming, 7 (1974), pp. 376–379.

[17] A. M. FRIEZE AND J. YADEGAR, *An algorithm for solving* 3*-dimensional assignment problems with application to scheduling a teaching practice*, J. Oper. Res. Soc., 32 (1981), pp. 989–995.

[18] B. GAVISH, *On obtaining the "best" multipliers for a Lagrangean relaxation for integer programming*, Comput. Oper. Res., 5 (1978), pp. 55–71.

[19] A. M. GEOFFRION, *Lagrangean relaxation for integer programming*, in Mathematical Programming Study 2: Approaches to Integer Programming, M. L. Balinski, ed., North Holland, Amsterdam, 1974.

[20] J. L. GOFFIN, *On convergence rates of subgradient optimization methods*, Math. Programming, 13 (1977) pp. 329–347.

[21] M. HELD, P. WOLFE, AND H. P. CROWDER, *Validation of subgradient optimization*, Math. Programming, 6 (1974), pp. 62–88.

[22] M. HELD AND R. M. KARP, *The traveling salesman problem and minimal spanning trees*, Oper. Res., 18 (1970), pp. 1138–1162.

[23] ———, *The traveling-salesman problem and minimum spanning trees: Part* II, Math. Programming, 1 (1971), pp. 6–25.

[24] R. JONKER AND A. VOLGENANT, *A shortest augmenting path algorithm for dense and sparse linear assignment problems*, Computing, 38 (1987), pp. 325–340.

[25] C. LEMARECHAL, *Bundle methods in nonsmooth optimization*, in Nonsmooth Optimization IIASA Proceedings 3, C. Lemarechal and R. Mifflin, eds., Pergamon, Oxford, 1978, pp. 79–102.

[26] P. S. MAYBECK, J. E. NEGRO, S. J. CUSMANO, AND M. DE PONTE, *A new tracker for air-to-air missile targets*, Trans. Automat. Control, AC-24 (1979), pp. 900–905.

[27] L. F. MCGINNIS, *Implementation and testing of a primal-dual algorithm for the assignment problem*, Oper. Res., 31 (1983), pp. 277–291.

[28] C. L. MOREFIELD, *Application of* 0-1 *integer programming to multitarget tracking problems*, IEEE Trans. Automat. Control, AC-22 (1977), pp. 302–312.

[29] S. MORI, C. CHONG, E. TSE, AND R. P. WISHNER, *Tracking and classifying multiple targets without a priori identification*, IEEE Trans. Automat. Control, AC-31 (1986), pp. 401–409.

[30] G. L. NEMHAUSER AND L. A. WOLSEY, *Integer and Combinatorial Optimization*, John Wiley and Sons, New York, 1988.

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

[32] K. R. PATTIPATI, D. SOMNATH, AND Y. BAR-SHALOM, *A relaxation algorithm for the passive sensor data association problem*, Proc. 1989 Amer. Control Conf., 3 (1989), pp. 2617–2624.

[33] K. R. PATTIPATI, D. SOMNATH, Y. BAR-SHALOM, AND R. B. WASHBURN, *Passive multisensor data association using a new relaxation algorithm*, in Multitarget-Multisensor Tracking: Advanced Applications, Y. Bar-Shalom, ed., Artech House, Dedham, MA, 1991.

[34] W. P. PIERSKALLA, *The multi-dimensional assignment problem*, Oper. Res., 16 (1968), pp. 422–431.

[35] A. B. POORE AND N. RIJAVEC, *Multitarget tracking and multidimensional assignment problems*, in Proc. 1991 SPIE Conf. on Signal and Data Processing of Small Targets, Vol 1481, O. E. Drummond, ed., 1991, pp. 345–356.

[36] D. B. REID, *An algorithm for tracking multiple targets*, IEEE Trans. Automat. Control, AC-24 (1979), pp. 843–854.

[37] R. T. ROCKAFELLAR, *Network Flows and Monotropic Optimization*, John Wiley and Sons, New York, 1984.

[38] A. SCHRIJVER, *Theory of Linear and Integer Programming*, John Wiley and Sons, New York, 1986.

[39] R. W. SITTLER, *An optimal data association problem in surveillance theory*, IEEE Trans. Military Electron., AES-11 (1981), pp. 122–130.

[40] J. F. SHAPIRO, *A survey of Lagrangian techniques for discrete optimization*, Ann. Discrete Math., 5 (1979), pp. 113–138.

[41] J. J. STEIN AND S. S. BLACKMAN, *Generalized correlation of multi-target data*, IEEE Trans. Aerospace Electron. Systems, AES-11 (1975), pp. 1207–1217.

[42] P. WOLFE, *A method of conjugate subgradients for minimizing nondifferentiable functions*, Math. Programming Stud., 3 (1975), pp. 147–173.

[43] P. WOLFE, *Finding the nearest point in a polytope*, Math. Programming, 11 (1976), pp. 128–149.

[44] M. R. ZUNIGA, J. M. PICONE, AND J. K. UHLMANN, *An algorithm for improved gating combinatorics in multiple-target tracking*, NRL Memorandum Report 6691, Naval Research Laboratory, Washington, DC, August 1990.

SIAM J. OPTIMIZATION
Vol. 3, No. 3, pp. 564–581, August 1993

008

MANIFOLD STRUCTURE OF THE KARUSH–KUHN–TUCKER STATIONARY SOLUTION SET WITH TWO PARAMETERS*

RYUICHI HIRABAYASHI†, MASAYUKI SHIDA‡, AND SUSUMU SHINDOH§

Abstract. This paper deals with a 2-dimensional parameter family of nonlinear programs: minimize $h_0(x,t)$ subject to the equality constraints $h_i(x,t) = 0$ $(i = 1,\ldots,l)$ and the inequality constraints $h_j(x,t) \leq 0$ $(j = l+1,\ldots,m)$. Each h_i $(i = 0,1,\ldots,m)$ is a twice continuously differentiable real-valued map defined on the $(n+2)$-dimensional Euclidean space R^{n+2}, where $x \in R^n$ denotes a variable vector and $t \in R^2$ denotes a 2-dimensional parameter vector. The local properties of the Karush–Kuhn–Tucker stationary solution set, the set Σ consisting of all (x,t) such that x is a stationary solution of the program for some t, are studied. In fact, it is shown that if the Mangasarian–Fromovitz constraint qualification and a regular value condition are satisfied, (i) the set Σ is a 2-dimensional topological manifold without a boundary, and (ii) the set Σ is a generalized creased manifold if, in addition, a constant rank condition holds.

Key words. multiparametric nonlinear programs, generalized creased manifold, topological manifold, piecewise differentiable manifold, Karush–Kuhn–Tucker set

AMS subject classifications. 90C30, 90C31

1. Introduction.

The main purpose of this paper is to investigate some properties of the set of Karush–Kuhn–Tucker stationary solutions (see, for example, Luenberger [13]) of a nonlinear program under continuous deformations. We shall shortly speak about a stationary solution instead of a Karush–Kuhn–Tucker stationary solution. To be concrete, we deal with a d-dimensional parameter family of nonlinear programs

$$P(t): \begin{array}{ll} \text{minimize} & h_0(x,t) \\ \text{subject to} & x \in X(t), \end{array}$$

where

$$X(t) = \{x \in R^n \; : \; h_i(x,t) = 0 \; (i \in L), h_j(x,t) \leq 0 \; (j \in M)\},$$

$$L = \{1,2,\ldots,l\}, \quad M = \{l+1, l+2, \ldots, m\}, \quad h = (h_0, h_1, \ldots, h_m)^\top$$

is a twice continuously differentiable (i.e., C^2) map from the $(n+d)$-dimensional Euclidean space R^{n+d} into R^{1+m}. Here, $x = (x_1, x_2, \ldots, x_n)^\top \in R^n$ denotes a variable vector, $t \in R^d$ denotes a d-dimensional parameter vector, and $\top$ stands for transposition. Later on, certain regularity assumptions will be imposed on the parametric constraint set $X(t)$. We study local properties of the stationary solution set Σ (i.e., the set of all (x,t) such that x is a stationary solution of $P(t)$ for some t) and the stationary point set Π (i.e., the set of all (x,y,t) such that x is a stationary solution of $P(t)$ for some t and y is a Lagrange multiplier vector associated with x).

There are at least three approaches to the 1-dimensional parameter family $P(t)$ $(t \in R)$ of nonlinear programs. The first approach is due to Kojima and Hirabayashi

* Received by the editors June 18, 1991; accepted for publication (in revised form) April 16, 1992.

† Department of Management Sciences, Science University of Tokyo, Kagurazaka, Shinjuku-ku, Tokyo 162, Japan.

‡ Department of Systems Science, Tokyo Institute of Technology, Oh-okayama, Meguro-ku, Tokyo 152, Japan.

§ Department of Mathematics and Physics, The National Defense Academy, Hashirimizu, Yokosuka, Kanagawa 239, Japan.

[12], who showed that the stationary solution set Σ is the disjoint union of paths (without a boundary) and closed loops. The second approach is due to Jongen, Jonker, and Twilt [4]–[9], who introduced the generalized critical points. They showed that the generalized critical points can generically be classified into five types. The third approach, by Poore and Tiahrt, is a direct application of bifurcation theory [15], [19]. For more details see an excellent survey paper by Jongen and Weber [10].

If the dimension d of the parameter space exceeds 1, the situation becomes more complicated. In [12] Kojima and Hirabayashi showed, under a suitable regularity assumption, that the stationary point set Π is a piecewise continuously differentiable manifold of dimension d. Moreover, if, in addition, a stationary solution $x \in X(t)$ is strongly stable (in the sense of Kojima [11]), then the stationary solution set Σ around x is parametrizable by means of the parameter t. Within the context of Pareto theory, Schecter [17] investigated the structure of the set Σ for special types of multiparametric programs under the following rank conditions:

(i) the gradients corresponding to the equality constraints are linearly independent in the feasible region;

(ii) the gradients corresponding to the active inequality constraints are linearly independent in the feasible region;

(iii) the rank of the set of the gradients corresponding to active constraints (including the equality constraints) is locally constant.

Under these rank conditions Schecter [17] proved that the set Σ is a creased manifold with a boundary. At the boundary points of this manifold the Mangasarian–Fromovitz constraint qualification (MFCQ) is violated. Jongen, Jonker, and Twilt [9] showed, under the linear independence constraint qualification (LICQ), that the set Σ of a general $P(t)$ $(t \in R^d)$ is generically a creased manifold (in the sense of Schecter [17]). Shindoh, Hirabayashi, and Matsumoto [18] dealt with 2-parameter cases and proved that, in general, the stationary index (a natural generalization of the Morse index) of a stationary point (x, y, t) can locally change at most by two on the stationary point set Π.

In this paper we investigate the structure of the stationary solution set Σ of $P(t)$ $(t \in R^d)$ for the case $d = 2$. The paper is organized as follows. In §2 we define a class of generalized creased manifolds that are topological manifolds and that form a natural extension of creased manifolds, and we present some preliminary results. In §3 some fundamental results necessary in the subsequent sections will be derived. In §4 we show that under MFCQ, a certain regularity assumption, and a certain constant rank assumption, the set Σ is a 2-dimensional generalized creased manifold without a boundary. In §5 we show that, under MFCQ and a certain regularity assumption only, the set Σ is a 2-dimensional topological manifold without a boundary.

In this paper we make great use of the structure of the Lagrange multiplier vector set that forms a polytope with dimension less than or equal to $d = 2$. For this case the adjacency of its faces is very simple. However, if $d \geq 3$, the polytope may have dimension d and the adjacency of its faces is combinatorially complicated. Hence it is rather difficult to treat the case of $d \geq 3$.

2. Preliminaries.

2.1. PC^1-maps. For every nonempty convex subset C of R^p the dimension of C is defined to be the dimension of affine hull $\operatorname{aff} C$, and is denoted by $\dim C$. By a *cell* we mean a closed convex polyhedral set (i.e., the intersection of a finite number of closed half-spaces) in R^p. By a *k-cell* we mean a cell of dimension k. If a cell B is a face of a cell C, we write $B \prec C$.

Let C be a k-cell in R^p, let $\mathcal{S}$ be a finite or countable collection of k-cells in R^p, and let $\overline{\mathcal{S}}$ be the collection of all faces of any $\sigma \in \mathcal{S}$, i.e., $\overline{\mathcal{S}} = \{\tau \subset R^p : \tau \prec \sigma$ for some $\sigma \in \mathcal{S}\}$. $\mathcal{S}$ is said to be a subdivision of C and we write $C = |\mathcal{S}|$ if the following three conditions are satisfied:

(i) $C = \bigcup\{\sigma : \sigma \in \mathcal{S}\}$;

(ii) for each $\tau, \sigma \in \mathcal{S}$ with $\tau \neq \sigma$, $\operatorname{Rel\,Int} \tau \cap \operatorname{Rel\,Int} \sigma = \emptyset$;

(iii) $\mathcal{S}$ is locally finite, i.e., each point $x \in C$ has a neighborhood that intersects with only a finite number of k-cells of $\mathcal{S}$.

Let $\mathcal{S}$ be a subdivision of a cell C in R^p. A PC^1 *(piecewise continuously differentiable) map* $G : |\mathcal{S}| \rightarrow R^q$ is a continuous map from C into R^q such that for each cell σ of $\mathcal{S}$ there exists an open set $U \supset \sigma$ and a C^1 map $G' : U \rightarrow R^q$ that satisfies $G'|\sigma = G|\sigma$, where $G|\sigma$ (or $G'|\sigma$) denotes the restriction of the map G (or G') to the cell σ. We shall use the notation $DG(z|\sigma)$ for the Jacobian matrix of the restriction of a PC^1 map $G : |\mathcal{S}| \rightarrow R^q$ to a cell $\sigma \in \overline{\mathcal{S}}$ at z.

2.2. Manifolds. A *d-dimensional topological manifold* in R^p is a subset Q of R^p such that for each $z \in Q$ there exists an open set $V \subset R^d$ and a homeomorphism $\phi : V \rightarrow R^p$ with $\phi(V)$ a neighborhood of z in Q. We call ϕ a parametrization around z.

A *d-dimensional* C^1*-manifold* (respectively, *manifold with a boundary*) in R^p is a subset Q of R^p such that for each $z \in Q$ there exists an open set $V \subset R^d$ and a C^1-embedding $\phi : V \rightarrow R^p$ with $\phi(V)$ a neighborhood of z in Q (respectively, $\phi(V \cap R^q_+ \times R^{d-q})$ a neighborhood of z in Q for some integer $q \in \{0, 1, \ldots, d\}$, where $R_+ = \{r \in R | r \geq 0\}$). We also call ϕ a parametrization around z.

A *d-dimensional* PC^1*-manifold with respect to a subdivision* $\mathcal{S}$ *of* R^p is a subset Q in R^p such that Q is a d-dimensional topological manifold, and for each cell $\sigma \in \mathcal{S}$ there exists a d-dimensional C^1-manifold Q' such that $Q \cap \sigma = Q' \cap \sigma$.

We call $\mathcal{S}$ a *creased subdivision* of R^d at the origin if $\mathcal{S}$ is a subdivision of R^d (i.e., $|\mathcal{S}| = R^d$) such that it is a finite number of collection of d-cells, each of which contains the origin. By a *section* we mean one of cells $(0, 1, \ldots, d$ cells) of $\overline{\mathcal{S}}$ induced by a creased subdivision $\mathcal{S}$ of R^d.

A *d-dimensional generalized creased manifold* is a subset Q in R^p such that there exist $V_\alpha, \phi_\alpha, \mathcal{S}_\alpha$; each V_α is an open neighborhood of the origin in R^d, each $\phi_\alpha : V_\alpha \rightarrow Q$ is a map (called a parametrization), and each $\mathcal{S}_\alpha$ is a creased subdivision of R^d that satisfies the following:

(i) $Q = \bigcup_\alpha \phi_\alpha(V_\alpha)$.

(ii) Each ϕ_α is a homeomorphism onto an open subset of Q. (Hence, Q is a topological manifold.)

(iii) For each α the restriction of ϕ_α to any section of $\mathcal{S}_\alpha$ is a C^1-embedding.

(iv) If ϕ_α, ϕ_β are two parametrizations, let $V_{\alpha\beta} = \phi_\alpha^{-1} \circ \phi_\beta(V_\beta) \subset V_\alpha$, and let $V_{\beta\alpha} = \phi_\beta^{-1} \circ \phi_\alpha(V_\alpha) \subset V_\beta$, so that $\phi_\beta^{-1} \circ \phi_\alpha$ is a homeomorphism of $V_{\alpha\beta}$ onto $V_{\beta\alpha}$. Then $\phi_\beta^{-1} \circ \phi_\alpha(V_{\alpha\beta} \cap$ any section of $\mathcal{S}_\alpha) = V_{\beta\alpha} \cap$ some section of $\mathcal{S}_\beta$.

A generalized creased manifold is a natural generalization of a creased manifold (see [17]). In particular, a 1-dimensional generalized creased manifold is a creased manifold.

Let Q be a d-dimensional generalized creased manifold. Then for each point $z \in Q$ there exist a neighborhood of the origin in R^d, a parametrization ϕ with $\phi(0) = z$, and a creased subdivision $\mathcal{S}$ (see Fig. 2.1). In this case, $\phi(V \cap$ a section of $\mathcal{S})$ is said to be a *section* of a generalized creased manifold around z.

Under this notation any d-dimensional PC^1-manifold for which extended pieces

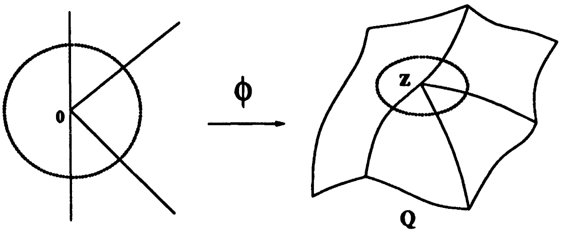


FIG. 2.1. *Generalized creased manifold.*

intersect each cell of a subdivision transversally is a generalized creased manifold. Here, a piece is a set of the type $Q \cap \sigma$, as introduced above.

2.3. Regular values of PC^1-maps. Let G be a C^1-map from an open subset U of R^p into R^q, and let V be a subset of U. A point $c \in R^q$ is said to be a *regular value* of $G : U \to R^q$ on V if $\operatorname{rank} DG(z) = q$ for every $z \in V$ such that $G(z) = c$. Obviously, if $p < q$, then no $c \in G(V)$ can be a regular value.

We shall define a *regular value* of a PC^1-map. Let $\mathcal{S}$ be a subdivision of a p-cell of C in R^p, and let $G : |\mathcal{S}| \to R^q$ be a PC^1-map. A point $c \in R^q$ is a regular value of the PC^1-map $G : |\mathcal{S}| \to R^q$ if $c \in R^q$ is a regular value of $G|\sigma$ for every face of any cell σ of $\mathcal{S}$ (for details, see [12]). Subsequent sections will be concerned with a special case for which $p = q + 2$. In this case, if c is a regular value of G, then $G^{-1}(c) = \{z \in |\mathcal{S}| : G(z) = c\}$ does not intersect any cell of $\overline{\mathcal{S}}$ with dimension less than q. Furthermore, using the well-known implicit function theorem [13] and some other elementary results in differential topology, we can prove the theorem below. The proof is omitted here.

THEOREM 2.1. *Let $\mathcal{S}$ be a subdivision of R^{q+2}, and let $G : |\mathcal{S}| \to R^q$ be a PC^1-map. Suppose that $c \in R^q$ is a regular value of the PC^1-map G. Then each connected component on $G^{-1}(c)$ is a 2-dimensional PC^1-manifold that intersects every face of any cell in $\mathcal{S}$ transversally.*

Remark 2.2. In Theorem 2.1, for each k-cell $\sigma \in \overline{\mathcal{S}}$ we have that if $G^{-1}(c) \cap \sigma \neq \emptyset$, then $G^{-1}(c) \cap \operatorname{Rel Int} \sigma$ is nonempty and, moreover, is a $(k-q)$-dimensional manifold. In particular, if $k \leq q - 1$, then $G^{-1}(c) \cap \sigma = \emptyset$.

2.4. Stationary solutions. As is stated in [12], it is convenient to formulate the (Karush–Kuhn–Tucker) stationary condition by means of a system of equations (the so-called Kojima function). We denote by $D_x h_i(x,t)$ the partial derivative of the map h_i with respect to x, i.e.,

$$D_x h_i(x,t) = \left(\frac{\partial h_i(x,t)}{\partial x_1}, \ldots, \frac{\partial h_i(x,t)}{\partial x_n} \right).$$

For every $\alpha \in R$ let

$$\alpha^+ = \max\{0, \alpha\} \quad \text{and} \quad \alpha^- = \min\{0, \alpha\}.$$

Then we obviously have

$$\alpha^+ \geq 0, \qquad \alpha^- \leq 0 \quad \text{for every } \alpha \in R$$

and $\alpha^+ = 0$ or $\alpha^- = 0$ (complementarity). Then the Kojima function $H : R^{n+m+2} \to R^{n+m}$ is defined as follows:

$$H(x,y,t) = \begin{bmatrix} D_x h_0(x,t)^\top + \sum_{i\in L} y_i D_x h_i(x,t)^\top + \sum_{j\in M} y_j^+ D_x h_j(x,t)^\top \\ -h_1(x,t) \\ \vdots \\ -h_l(x,t) \\ y_{l+1}^- - h_{l+1}(x,t) \\ \vdots \\ y_m^- - h_m(x,t) \end{bmatrix}$$

The stationary condition now becomes

$$H(x,y,t) = 0.$$

Let $t \in R^2$ be fixed. If $x \in R^n$ satisfies the above stationary condition for some $y \in R^m$, we call x a stationary solution of $P(t)$, we call y a Lagrange multiplier vector associated with x, and we call the pair (x, y) a stationary point of $P(t)$. Define Σ to be the set of all (x,t)'s such that x is a stationary solution of $P(t)$:

$$\Sigma = \{(x,t) \in R^{n+2} : H(x,y,t) = 0 \text{ for some } y \in R^m\}.$$

The set Σ is our target whose local properties will be studied. For convenience, we also define Π to be the set of all (x, y, t) such that (x, y) is a stationary point of $P(t)$:

$$\Pi = \{(x,y,t) \in R^{n+m+2} : H(x,y,t) = 0\}.$$

Note that the set Σ is the natural projection of Π under the map $(x, y, t) \mapsto (x,t)$. Let

$$J_0(x,t) = \{j \in M : h_j(x,t) = 0\}.$$

Condition 0 (linear independence constraint qualification). For every (x,t) in Σ, $\{D_x h_i(x,t) : i \in L \cup J_0(x,t)\}$ is linearly independent.

Condition 1 (Mangasarian–Fromovitz constraint qualification [14]). For every (x,t) in Σ

(i) $\{D_x h_i(x,t) : i \in L\}$ is linearly independent;

(ii) there exists a $w \in R^n$ such that

$$D_x h_i(x,t)w = 0 \quad \text{for every } i \in L,$$

$$D_x h_j(x,t)w < 0 \quad \text{for every } j \in J_0(x,t).$$

As is well known, Condition 0 implies Condition 1. Moreover, Condition 0 ensures the uniqueness of the Lagrange multiplier vector associated with a stationary solution (x,t), whereas Condition 1 ensures the boundedness of the set of the Lagrange multiplier vectors [3].

Condition 2 (regularity condition). $0 \in R^{n+m}$ is a regular value of the PC^1-map H.

For the more precise definition, see [11] and [12]. Under Condition 2, Π is a 2-dimensional PC^1-manifold without a boundary.

Throughout the paper we use the following notations and symbols:

$$H(x,y,t)=\begin{pmatrix} D_xh_0(x,t)^\top+\sum_{i\in L}y_iD_xh_i(x,t)^\top+\sum_{j\in M}y_j^+D_xh_j(x,t)^\top \\ -h_i(x,t) \qquad (i\in L) \\ y_j^- - h_j(x,t) \qquad (j\in M) \end{pmatrix},$$

$$\Pi=\{(x,y,t)\in R^{n+m+2}:\ H(x,y,t)=0\}\quad \text{(stationary point set)},$$

$$\Sigma=\{(x,t)\in R^{n+2}:\ H(x,y,t)=0 \text{ for some } y\in R^m\}\quad \text{(stationary solution set)}.$$

For each I,J with $J\subseteq I\subseteq M$

$$\tau_{IJ}=R^n\times\{y\in R^m:\ y_i<0\Leftrightarrow i\in M\backslash I,\ y_j>0\Leftrightarrow j\in J\}\times R^2,$$

$$\mathcal{K}^*=\{\overline{\tau}_{JJ}:J\subseteq M\}$$

is a subdivision of R^{n+m+2},

$$\overline{\mathcal{K}}^*=\{\sigma:\sigma \text{ is a face for some } \overline{\tau}_{JJ}(J\subseteq M)\},$$

$$\Pi_{IJ}=\{(x,y,t)\in\Pi:(x,y,t)\in\tau_{IJ}\},$$

$$\Sigma_J=\{(x,t)\in\Sigma:h_i(x,t)=0\Leftrightarrow i\in L\cup J\},$$

Z_J (respectively, P_{IJ}) is a connected component of Σ_J (respectively, Π_{IJ}), ρ is a natural projection map from Π to Σ ($(x,y,t)\mapsto(x,t)$),

$$Y(x,t)=\{y\in R^m:(x,y,t)\in\Pi\}$$

for a fixed $(x,t)\in\Sigma$,

$$N(x,y,t)=D_x^2h_0(x,t)+\sum_{i\in L}y_iD_x^2h_i(x,t)+\sum_{j\in M}y_j^+D_x^2h_j(x,t),$$

$$A_J(x,t)=[D_xh_j(x,t)^\top\ (j\in L\cup J)],$$

$$M_J(x,y,t)=\begin{pmatrix} N(x,y,t) & A_J(x,t) \\ -A_J(x,t)^\top & 0 \end{pmatrix},$$

$$W_J(x,t)=\{w\in R^n:D_xh_i(x,t)w=0\ (i\in L\cup J)\},$$

$$J_0(x,t)=\{j\in M:h_j(x,t)=0\},$$

$$J_+(y)=\{j\in M:y_j>0\}\quad \text{for } y\in R^m,$$

$$J_n(y) = \{j \in M : y_j \geq 0\} \quad \text{for } y \in R^m,$$

and $|J|$ is the cardinality of J.

Under Conditions 1 and 2 and the above notations, we can easily see the following properties:

$$\overline{\mathcal{K}}^* = \{\overline{\tau}_{IJ} : J \subseteq I \subseteq M\},$$

$$\overline{\Pi}_{IJ} = \bigcup_{K_2 \subseteq J \subseteq I \subseteq K_1} \Pi_{K_1 K_2},$$

$$\overline{\Sigma}_J = \bigcup_{J' \supseteq J} \Sigma_{J'} \qquad (\overline{\Sigma}_J \text{ is a closure of } \Sigma_J \text{ in } \Sigma),$$

$$\Sigma_{J'} \subseteq \overline{\Sigma}_J \quad \text{for each } J' \supseteq J,$$

$$\rho(\overline{\Pi}_{JJ}) = \overline{\Sigma}_J.$$

Suppose that $z = (x, y, t)$ is a stationary point of $P(t)$. Then we see that

$$J_+(y) \subset J_n(y) = J_0(x,t) \quad \text{and} \quad z = (x,y,t) \in \overline{\tau}_{JJ}$$

if $J_+(y) \subset J \subset J_n(y)$.

Then, up to a row permutation, we can represent the $(n+m) \times (n+m)$ Jacobian matrix $D_{(x,y)}H(z|\overline{\tau}_{JJ})$ as

$$D_{(x,y)}H(z|\overline{\tau}_{JJ}) = \left(\begin{array}{cc|c} \multicolumn{2}{c|}{M_J(z)} & 0 \\ \hline -A_J^c(z)^\top & 0 & E \end{array}\right),$$

where $A_J^c(z) = [D_x h_j(x,t)^\top \ (j \in M \backslash J)]$ and E denotes the $(m - |J|) \times (m - |J|)$ identity matrix. Hence in this case we have

$$\det\, D_{(x,y)}H(z|\overline{\tau}_{JJ}) = \det M_J(z).$$

3. Basic results. In this section we introduce some basic properties of the stationary solutions.

LEMMA 3.1. *Suppose that Conditions* 1 *and* 2 *hold. Suppose that* $(x,t) \in \Sigma$ *and* $J_0(x,t) = \{l+1, l+2, \ldots, k\}$ *for some* $k \in \{l, l+1, \ldots, m\}$. *In case* $k = l$ *we set* $J_0(x,t) = \emptyset$. *Then we have either*

(i) $\operatorname{rank} A_{J_0(x,t)}(x,t) = k \geq l$ *or*

(ii) $\operatorname{rank} A_{J_0(x,t)}(x,t) = k-2$ *or* $k-1 \ \geq l+1$.

Proof. If $\operatorname{rank} A_{J_0(x,t)}(x,t) = k - d$ $(d \geq 1)$, then from Condition 1 we see that $k - d \geq l + 1$. From Condition 2, $\operatorname{rank} A_{J_0(x,t)}(x,t) \geq k - 2$. □

Lemmas 3.2 and 3.3 below are independent of Condition 2.

LEMMA 3.2. *Suppose that Condition* 1 *holds. Let* Z *be a subset of* Σ, *and let* $P = \rho^{-1}(Z) = \{(x,y,t) \in \Pi : (x,t) \in Z\}$. *Then the following hold:*

(i) *If* Z *is compact, so is* P. *In particular, if* Z *consists of a single point, then* P *is a compact cell.*

(ii) *If* Z *is connected, then so is* P.

(iii) *If Z is a connected component of Σ, then P is a connected component of Π.*

LEMMA 3.3. *Let P be a connected component of Π, and let Z be the natural projection of P, i.e., $Z = \rho(P)$. Then Z is a connected component of Σ, and $P = \rho^{-1}(Z)$ holds.*

In view of Lemmas 3.2 and 3.3, we see that there is a one-to-one correspondence between the connected components of Π (respectively, $\overline{\Pi}_{JJ}$) and those of Σ (respectively, $\overline{\Sigma}_J$).

The next lemma, which asserts the upper semicontinuity of ρ^{-1}, is well known and is fundamental to this paper.

LEMMA 3.4. *Under Condition 1, ρ^{-1} is upper semicontinuous in the sense of Berge (see* [1], [2]).

Proof. See [16, Thm. 2.3]. □

Remark 3.5. If Condition 0 (LICQ) holds at (x,t), then there exists one and only one Lagrange multiplier $y \in R^m$ for (x,t) and ρ is a homeomorphism from some neighborhood of $(x,y,t) \in \Pi$ to some neighborhood of $(x,t) \in \Sigma$.

If Conditions 1 and 2 hold, then $\operatorname{corank} A_{J_0(x,t)}(x,t) \leq 2$, where $\operatorname{corank} A = \min\{m,n\} - \operatorname{rank} A$ for any $n \times m$ matrix A.

LEMMA 3.6. *Under Conditions 1 and 2, for each $(x,t) \in \Sigma$*

(i) *if $\operatorname{corank} A_J(x,t) = 0$, then $\rho^{-1}(x,t)$ is a singleton;*

(ii) *if $\operatorname{corank} A_J(x,t) = 1$, then $\rho^{-1}(x,t)$ is a singleton or a line segment;*

(iii) *if $\operatorname{corank} A_J(x,t) = 2$, then $\rho^{-1}(x,t)$ is a 2-dimensional polytope, where $J = J_0(x,t)$.*

Proof. (i) This case is clear.

(ii) Since $\operatorname{corank} A_J(x,t) = 1$, the set $Y(x,t)$ is an intersection of a straight line

$$\left\{y \in R^m : \; y_j = h_j(x,t) \; (j \in M\backslash J), \; D_x h_0(x,t)^\top + \sum_{i \in L \cup J} y_i D_x h_i(x,t)^\top = 0\right\}$$

and the y-component of an orthant $\overline{\tau}_{JJ}$. Note that $(x,t) \in Z_J$ and Condition 1 (MFCQ) is satisfied at this point, and so $\rho^{-1}(x,t)$ is nonempty and compact. Therefore, $\rho^{-1}(x,t)$ is a singleton or a line segment.

(iii) In this case $\operatorname{corank} A_J(x,t) = 2$. By an argument similar to that in (ii), $Y(x,t)$ is a (nonempty and compact) intersection of a 2-dimensional plane

$$\left\{y \in R^m : \; y_j = h_j(x,t) \; (j \in M\backslash J), \; D_x h_0(x,t)^\top + \sum_{i \in L \cup J} y_i D_x h_i(x,t)^\top = 0\right\}$$

and the y-component of an orthant $\overline{\tau}_{JJ}$, so that the set $Y(x,t)$ is a singleton, a line segment, or a 2-dimensional polytope. However, if it is 1-dimensional, then it is not bounded, and this contradicts Condition 1 (MFCQ), whereas Condition 2 (regularity condition) and Theorem 2.1 yield that it is not a singleton. Hence $\rho^{-1}(x,t)$ is a two-dimensional polytope. □

4. Structure of the stationary solution set: I.

In this section we show that the stationary solution set is a 2-dimensional generalized creased manifold under the assumption of Conditions 1 and 2 and under the additional Condition 3 below, which was introduced by Schecter [17].

For $k \geq 0$ let $\mathcal{J}_k$ denote the collection of Z_J (connected component of Σ_J) such that $\operatorname{corank} A_J(x,t) = k$ at every $(x,t) \in \overline{Z}_J$ ($\overline{Z}_J$ is a closure of Z_J in Σ).

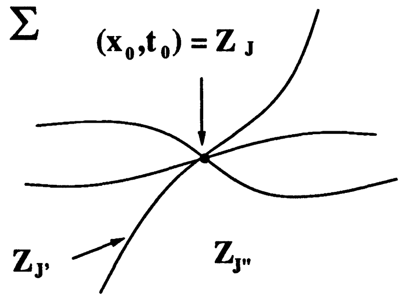


FIG. 4.1.

Condition 3 (constant rank condition [17]). For each $J \subseteq M$ every connected component Z_J of Σ_J belongs to some $\mathcal{J}_k$. Assuming Condition 2, we have $k = 0, 1$, or 2.

Throughout this section we assume that Conditions 1, 2, and 3 hold.

The next two results are key to proving the manifold theorems (Theorems 4.3 and 4.5) below. We understand a ruled surface to be the union of a one-parameter family of straight lines.

LEMMA 4.1. *For each* $(x,t) \in \Sigma$

(i) *if* $Z_J \in \mathcal{J}_0$ *and* $(x,t) \in Z_J$, *then* $\rho^{-1}(x,t)$ *is a singleton;*

(ii) *if* $Z_J \in \mathcal{J}_1$ *and* $(x,t) \in Z_J$, *then* $\rho^{-1}(x,t)$ *is a singleton or a line segment and* $\rho^{-1}(Z_J)$ *is a ruled surface with a boundary;*

(iii) *if* $Z_J \in \mathcal{J}_2$ *and* $(x,t) \in Z_J$, *then* $\rho^{-1}(x,t)$ *is a* 2-*dimensional polytope and* $Z_J = \{(x,t)\}$.

Proof. (i) Since $Z_J \in \mathcal{J}_0$ (i.e., $\{D_x h_j(x,t)\ (j \in L \cup J)\}$ is linearly independent), $\rho^{-1}(x,t)$ is a singleton from Lemma 3.6(i).

(ii) Since $Z_J \in \mathcal{J}_1$, $\operatorname{corank} A_J(x,t) = 1$. From Lemma 3.6(ii), $\rho^{-1}(x,t)$ is a singleton or a line segment. From Condition 3 (constant-rank condition) and Lemma 3.4, $\rho^{-1}(\overline{Z}_J)$ is a ruled surface with a boundary.

(iii) Since $Z_J \in \mathcal{J}_2$, $\operatorname{corank} A_J(x,t) = 2$. From Lemma 3.6(iii), $\rho^{-1}(x,t)$ is a 2-dimensional polytope. It is clear that $Z_J = \{(x,t)\}$. □

In the next lemma, the (constant rank) Condition 3 is heavily used.

LEMMA 4.2. *Let* $Z_J \in \mathcal{J}_2$. (*Hence from Lemma* 4.1(iii) *there exists some* (x_0,t_0) *such that* $Z_J = \{(x_0,t_0)\}$.) *Then there is at least one* $Z_{J''} \in \mathcal{J}_0$ *with* (i) $J'' \subset J$, (ii) $|J''| = |J| - 2$, *and* (iii) $(x_0,t_0) \in \overline{Z}_{J''}$. *Moreover, if* $Z_{J''}$ *satisfies properties* (i), (ii), *and* (iii), *then* $Z_{J''} \in \mathcal{J}_0$. *There is at least one* $Z_{J'} \in \mathcal{J}_1$ *with* (iv), $J'' \subset J' \subset J$, (v) $(x_0,t_0) \in \overline{Z}_{J'}$, *and* (vi) $|J'| = |J| - 1$. *Moreover, if* $Z_{J'}$ *satisfies properties* (iv), (v), *and* (vi), *then* $Z_{J'} \in \mathcal{J}_1$ (*see Fig.* 4.1). *Also, for any* J' *such that* $J'' \subset J' \subset J$ *there exists* $Z_{J'}$ *such that* $(x_0,t_0) \in Z_{J'}$ *and* $Z_{J'} \in \mathcal{J}_1$.

Proof. Since $\rho^{-1}(x_0,t_0)$ is a 2-dimensional polytope, each vertex of this polytope is on an intersection of four orthants. Denote one of those vertices by (x_0,y_0,t_0), which is in $\overline{\Pi}_{JJ} \cap \overline{\Pi}_{J''J''}$ with $|J''| = |J| - 2$. Therefore, there exists a $Z_{J''}$ such that $(x_0,t_0) \in \overline{Z}_{J''}$. Without loss of generality, we may assume that $J = \{l+1,\ldots,k\}$ and $J'' = \{l+1,\ldots,k-2\}$. Let $J_1' = \{l+1,\ldots,k-1\}$, and let $J_2' = \{l+1,\ldots,k-2,k\}$ (in this case $(x_0,y_0,t_0) \in \overline{\Pi}_{J_i'J_i'}$ $(i = 1,2)$). From Lemma A.1 in the Appendix there exists at least one nonzero element in $\det M_J$, $\det M_{J_1'}$, $\det M_{J_2'}$ and $\det M_{J''}$.

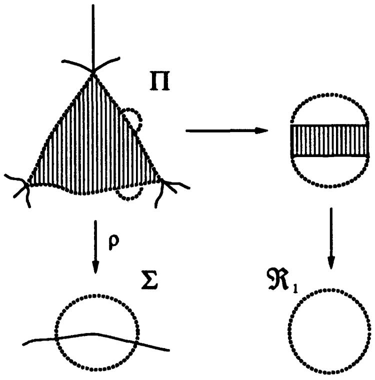


FIG. 4.2.

For the sake of simplicity, set $A_J := A_J(x_0, t_0)$ and set $M_J := M_J(x_0, y_0, t_0)$. Note that $\operatorname{corank} M_J = 2$ since $\operatorname{corank} A_J = 2$ ($Z_J \in \mathcal{J}_2$). Thus $1 \leq \operatorname{corank} M_{J_1'}$, $\operatorname{corank} M_{J_2'} \leq 2$, and hence $\det M_{J_1'} = \det M_{J_2'} = 0$. Therefore, $\det M_{J''} \neq 0$, and it implies $\operatorname{corank} A_{J''} = 0$. Therefore, $Z_{J''} \in \mathcal{J}_0$.

Since $\operatorname{rank} A_{J''} \leq \operatorname{rank} A_{J_i'} \leq \operatorname{rank} A_J$ $(i = 1, 2)$ and $\operatorname{rank} A_{J''} = \operatorname{rank} A_J = k - 2$, $\operatorname{rank} A_{J_i'} = k - 2$ $(= |L \cup J_i'| - 1)$ $(i = 1, 2)$. Therefore, $Z_{J_i'} \in \mathcal{J}_1$ $(i = 1, 2)$.

The other cases are easily shown. □

Now we are in a position to state one of the main results of this paper. Set

$$\Pi_0 = \bigcup_{Z_J \in \mathcal{J}_0} \overline{\rho^{-1}(Z_J)}.$$

THEOREM 4.3. *Suppose that Conditions* 1, 2, *and* 3 *hold. Then the set* Σ *is a* 2-*dimensional topological manifold without a boundary.*

Proof. Let any $(x_0, t_0) \in \Sigma$ be fixed. We will prove that some neighborhood of (x_0, t_0) in Σ is homeomorphic to R^2.

Case 1 ($Z_J \in \mathcal{J}_0$ and $(x_0, t_0) \in Z_J$). Since $(x_0, t_0) \in Z_J \in \mathcal{J}_0$ from Lemma 4.1(i), the set $\rho^{-1}(x_0, t_0)$ is a singleton denoted by (x_0, y_0, t_0). Let N be a neighborhood of (x_0, y_0, t_0) in Π such that (x, t) lies in Z_J for each $(x, y, t) \in N$. Since $Z_J \in \mathcal{J}_0$, the mapping ρ is a homeomorphism from N to $\rho(N)$ (from Remark 3.5). Hence Π is homeomorphic to Σ on the neighborhood of (x_0, y_0, t_0), i.e., in the neighborhood of (x_0, t_0), Σ is a 2-dimensional topological manifold (without a boundary).

Case 2 ($Z_J \in \mathcal{J}_1$ and $(x_0, t_0) \in Z_J$). In this case, $\rho^{-1}(x_0, t_0)$ is a singleton or a line segment (from Lemma 4.1(ii)).

Subcase 2.1 ($\rho^{-1}(x_0, t_0)$ is a line segment; see Fig. 4.2). This case has been already proved by Schecter [17], and we briefly trace his proof since it makes the rest of the proof more understandable.

For $i = 1, 2$ let (x_0, y_0^i, t_0) be end points of $\rho^{-1}(x_0, t_0)$, and let N_i be a neighborhood of (x_0, y_0^i, t_0) in Π. We shall show that $\rho|_{N_i \cap \Pi_0}$ is one-to-one. If $(x, y, t) \in \operatorname{Rel\,Int}(N_i \cap \Pi_0)$, then there exists $Z_{J'} \in \mathcal{J}_0$ with $(x, t) \in Z_{J'}$ and hence ρ is one-to-one. If $(x, y, t) \in N_i \cap \partial\Pi_0$, then $\rho^{-1}(x, t)$ is a line segment in τ_{JJ} and hence $\{(x, y, t)\} = \rho^{-1}(x, t) \cap N_i \cap \Pi_0$. Thus ρ is one-to-one in this case.

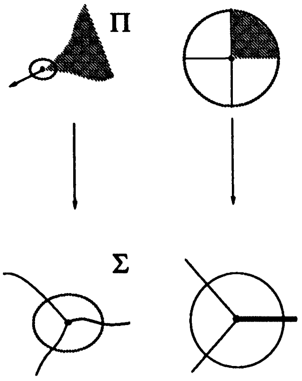


FIG. 4.3.

Next, it may be easily seen that a neighborhood of (x_0, t_0) in Σ is homeomorphic to the quotient space

$$\mathcal{R}_1 = (N_1 \cap \Pi_0) \cup (N_2 \cap \Pi_0)/(x,y,t) \sim (x',y',t') \Leftrightarrow (x,t) = (x',t').$$

If $(x, y_i, t) \in N_i \cap \operatorname{Rel\,Int} \Pi_0$, then it is not identified with any other point in $\mathcal{R}_1$. If $(x, y_1, t) \in N_1 \cap \partial\Pi_0$ (respectively, $(x, y_2, t) \in N_2 \cap \partial\Pi_0$), then $\rho^{-1}(x,t)$ is a line segment open in $\Pi \backslash \Pi_0$. Hence (x, y_1, t) (respectively, (x, y_2, t)) can be identified with some $(x, y_2, t) \in N_2 \cap \partial\Pi_0$ (respectively, $(x, y_1, t) \in N_1 \cap \partial\Pi_0$) from Lemma 4.1(ii).

Note that $N_i \cap \Pi_0$ is homeomorphic to $R \times R_+$ (for $i = 1, 2$). Now those boundaries $N_i \cap \partial\Pi_0$ $(i = 1, 2)$ are identified in $\mathcal{R}_1$ in the sense above. Hence Σ is a 2-dimensional topological manifold (without a boundary) in a neighborhood of (x_0, t_0).

Subcase 2.2 ($\rho^{-1}(x_0, t_0)$ is a singleton denoted by (x_0, y_0, t_0); see Fig. 4.3). In this case (x_0, y_0, t_0) is on an intersection of four orthants. Therefore, there are three index sets J'', J_1', and J_2' with $|J''| = |J| - 2$ and $J'' \subset J_i' \subset J$ $(i = 1, 2)$ such that $(x_0, t_0) \in Z_J \cap \overline{Z}_{J''} \cap \overline{Z}_{J_1'} \cap \overline{Z}_{J_2'}$ (in this case $Z_{J''}$ and $Z_{J_i'}$ $(i = 1, 2)$ are elements of $\mathcal{J}_0$). Let N be a neighborhood of (x_0, y_0, t_0) in Π. Then $N \cap \Pi_0 = N \backslash \Pi_{JJ}$ and $N \cap \partial\Pi_0 = N \cap (\Pi_{JJ_1'} \cup \Pi_{JJ_2'} \cup \Pi_{JJ''})$.

A neighborhood of (x_0, t_0) in Σ is homeomorphic to the quotient space

$$\mathcal{R}_2 = N \cap \Pi_0/(x,y,t) \sim (x',y',t') \Leftrightarrow (x,t) = (x',t').$$

It is clear that $\rho|_{N \cap \operatorname{RelInt} \Pi_0}$ is one-to-one. For each $(x, y, t) \in N \cap \Pi_{JJ_1'}$ (respectively, $N \cap \Pi_{JJ_2'}$) the set $\rho^{-1}(x,t)$ is a line segment such that one end point is (x, y, t). From Lemma 4.1(ii) another end point is in $N \cap \Pi_{JJ_2'}$ (respectively, $N \cap \Pi_{JJ_1'}$). Thus in $\mathcal{R}_2$ these two points are identified and no other points are identified. If $(x, y, t) \in N \cap \Pi_{JJ''}$, then $(x, y, t) = (x_0, y_0, t_0)$ and it is not identified with any other points in $\mathcal{R}_2$. Since $N \cap \Pi_0$ is homeomorphic to $R \times R_+$ and its boundary is identified in $\mathcal{R}_2$, the set Σ is a 2-dimensional topological manifold (without a boundary) in the neighborhood of (x_0, t_0).

Case 3 ($Z_J \in \mathcal{J}_2$ and $(x_0, t_0) \in Z_J$; see Fig. 4.4). In this case $\rho^{-1}(x_0, t_0)$ is a 2-dimensional polytope (from Lemma 4.1(iii)), and so its boundary is orientable. Let $z_0^i = (x_0, y_0^i, t_0)$ be its vertices $(i = 1, \ldots, r, \bmod r)$, whose numbering is determined along its boundary. For each i define

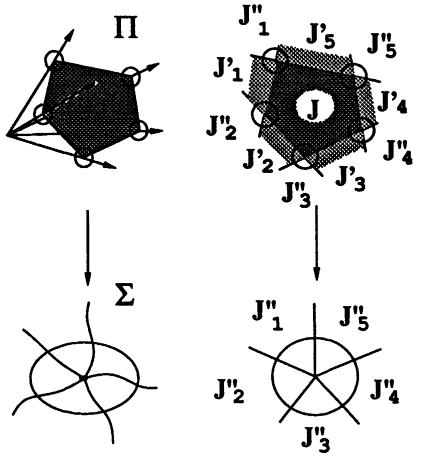


FIG. 4.4.

(i) J_i'' as an index set with $|J_i''| = |J| - 2$ such that $z_0^i \in \Pi_{JJ_i''}$;

(ii) J_i' as an index with $|J_i'| = |J| - 1$ such that $J_i'', J_{i+1}'' \subset J_i' \subset J$ and the edge $\overline{z_0^i z_0^{i+1}}$ is in $\Pi_{JJ_i'}$.

In this case $J_i'' \subset J_i', J_{i-1}' \subset J$, $Z_{J_i''} \in \mathcal{J}_0, Z_{J_i'} \in \mathcal{J}_1$, and $Z_J \in \mathcal{J}_2$ (from Lemma 4.2). Let N_i be a neighborhood of z_0^i in Π. From Lemma 4.2, again, $N_i \cap \Pi_0 = N_i \cap \overline{\Pi}_{J_i''J_i''}$.

We shall show that $\rho|_{N_i \cap \Pi_0}$ is one-to-one. If $(x, y, t) \in N_i \cap \operatorname{Rel\,Int} \Pi_0$, then $(x, t) \in Z_{J_i''}$; hence, ρ is one-to-one. If $(x, y, t) \in N_i \cap \partial\Pi_0$, then two cases occur. The first is that (x, y, t) is in $\Pi_{JJ_i''}$; in this case (x, t) is (x_0, t_0). The other case is that (x, y, t) is in $\Pi_{J_i'J_i''}$ (respectively, $\Pi_{J_{i-1}'J_i''}$); in this case $\rho^{-1}(x, t)$ is a line segment in $\overline{\Pi}_{J_i'J_i'}$ (respectively, $\overline{\Pi}_{J_{i-1}'J_{i-1}'}$), and hence $\{(x, y, t)\} = \rho^{-1}(x, t) \cap N_i \cap \Pi_0$. Thus $\rho|_{N_i \cap \Pi_0}$ is one-to-one.

A neighborhood of (x_0, t_0) in Σ is homeomorphic to the quotient space

$$\mathcal{R}_3 = \bigcup_{i=1}^{r} (N_i \cap \Pi_0)/(x, y, t) \sim (x', y', t') \Leftrightarrow (x, t) = (x', t').$$

If $(x, y, t) \in N_i \cap \operatorname{Rel\,Int} \Pi_0$, (x, y, t) cannot be identified with any other points in $\mathcal{R}_3$. If $(x, y, t) \in N_i \cap \partial\Pi_0$, then the following two cases occur:

(i) If $(x, y, t) \in \Pi_{JJ_i''}$, then (x, y, t) is (x_0, y_0^i, t_0) and hence it is identified with all other vertices of $\rho^{-1}(x, t)$ in $\mathcal{R}_3$.

(ii) If $(x, y, t) \in N_i \cap \Pi_0 \cap \Pi_{J_{i-1}'J_i''}$ (respectively, $\Pi_{J_i'J_i''}$), then $\rho^{-1}(x, t)$ is a line segment in $\overline{\Pi}_{J_{i-1}'J_{i-1}'}$ (respectively, $\overline{\Pi}_{J_i'J_i'}$). Hence it can be identified with some $(x', y', t') \in N_{i-1} \cap \Pi_0 \cap \Pi_{J_{i-1}'J_{i-1}''}$ (respectively, $N_{i+1} \cap \Pi_0 \cap \Pi_{J_i'J_{i+1}''}$) in $\mathcal{R}_3$.

Note that for each $i \in \{1, \ldots, r\}$, $N_i \cap \Pi_0$ is homeomorphic to $R \times R_+$. Now those boundaries are identified in $\mathcal{R}_3$ in the sense above. Hence Σ is a 2-dimensional topological manifold (without a boundary) in the neighborhood of (x_0, t_0). Hence each connected component of Σ is a 2-dimensional topological manifold without a boundary. □

Remark 4.4. Schecter [17] shows and proves only Case 1 and Subcase 2.1 in Case 2.

THEOREM 4.5. *Under Conditions* 1, 2, *and* 3 *the set* Σ *is a* 2*-dimensional generalized creased manifold without a boundary.*

Proof. From Theorem 4.3 we know that Σ is a topological manifold without a boundary and that it is obtained by all $\overline{Z}_J$ with $Z_J \in \mathcal{J}_0$. Note that the set Π is a 2-dimensional PC^1-manifold transversally intersecting each face of every cell of the subdivision $\mathcal{K}^*$. Also note that $\rho^{-1}(\overline{Z}_J)$ is a connected component of $\overline{\Pi}_{JJ}$ and a 2-dimensional C^1-manifold with a boundary. Hence from the proof of Theorem 4.3 it suffices to show that $\rho^{-1}|_{\overline{Z}_J}$ is a diffeomorphism for each $Z_J \in \mathcal{J}_0$.

We have already seen that ρ is a homeomorphism from Lemma 4.1(i) in this case.

Since $\rho^{-1}(\overline{Z}_J)$ is a 2-dimensional C^1-manifold with a boundary, there exist two linearly independent tangent directions for each point in $\rho^{-1}(\overline{Z}_J)$. Now it suffices to show that projections of these tangent directions to (x,t)-space ($R^n \times R^2$) are still linearly independent.

Let T_z denote a tangent space of $\rho^{-1}(\overline{Z}_J)$ at z (it is 2-dimensional). Then it is clear that $DH(z|\overline{\tau}_{JJ})w = 0$ for each $w \in T_z$. Now suppose that there exist $w_1 = (\dot{x}_1, \dot{y}_1, \dot{t}_1)$ and $w_2 = (\dot{x}_2, \dot{y}_2, \dot{t}_2)$ in T_z such that w_1 and w_2 are linearly independent but the projections of these directions to (x,t)-space are linear dependent. Without loss of generality, we may assume that $w_2 = (0, \dot{y}_2, 0)$ (note that $w_2 \neq 0$), and $DH(z|\overline{\tau}_{JJ})$ is of the form

$$DH(z|\overline{\tau}_{JJ}) = \left(\begin{array}{cc|c|c} N & A_J & 0 & \\ -A_J^\top & 0 & & \partial H/\partial t \\ \cline{1-3} * & 0 & E & \end{array}\right) =: \left(\begin{array}{c|c|c} N & & \\ -A_J^\top & K & \partial H/\partial t \\ \cline{1-1} * & & \end{array}\right).$$

Note that $Z_J \in \mathcal{J}_0$, so that the $(n+m) \times m$ submatrix K has full column rank. Therefore, $\dot{y}_2 = 0$ since $DH(z|\overline{\tau}_{JJ})w_2 = K\dot{y}_2 = 0$. Hence $w_2 = 0$, which contradicts the fact that $w_2 \neq 0$. Thus, the projection of the basis of T_z spans R^2. Hence $\rho^{-1}|_{\overline{Z}_J}$ is diffeomorphic.

Consequently, the set Σ is a 2-dimensional generalized creased manifold without a boundary. □

5. Structure of the stationary solution set: II. In this section we show that merely Conditions 1 and 2 together already guarantee that the stationary solution set Σ is a 2-dimensional topological manifold.

THEOREM 5.1. *Under Conditions* 1 *and* 2 *the set* Σ *is a* 2*-dimensional topological manifold without a boundary.*

Proof. If Condition 3 holds at all points (x,t) in Σ, the assertion is proved in Theorem 4.3. So we assume that there exists some point (x,t) in Σ at which Condition 3 is violated. Let $\Omega := \{(x,t) \in \Sigma : \operatorname{corank} A_{J_0(x,t)}(x,t) > 0\}$ (we assume that Ω is not empty). Note that Ω is a closed subset of Σ. Let (x,t) be in Ω. If for any Z_J ($J \subseteq M$) with $(x,t) \in \overline{Z}_J$ there exists some neighborhood U of (x,t) such that $\operatorname{rank} A_J(x',t') = \operatorname{rank} A_J(x,t)$ for each point (x',t') of $\overline{Z}_J \cap U$ (we call this point (x,t) an element of $\operatorname{Rel\,Int} \Omega$), then the assertion of this theorem is clear from Theorem 4.3 at point (x,t) (at each point of $\operatorname{Rel\,Int} \Omega$).

We have only to show the case for which (x,t) is in $\partial\Omega$ ($= \Omega \backslash \operatorname{Rel\,Int} \Omega$) (i.e., for some Z_J ($J \subseteq M$) with $(x,t) \in \overline{Z}_J$ and for any small neighborhood U of (x,t) there exists some point $(x',t') \in \overline{Z}_J \cap U$ such that $\operatorname{rank} A_J(x,t) < \operatorname{rank} A_J(x',t')$). First, assume that $\operatorname{corank} A_J(x,t) = 1$ and that $\rho^{-1}(x,t)$ is a line segment (it may be a singleton, but it is clear for that case).

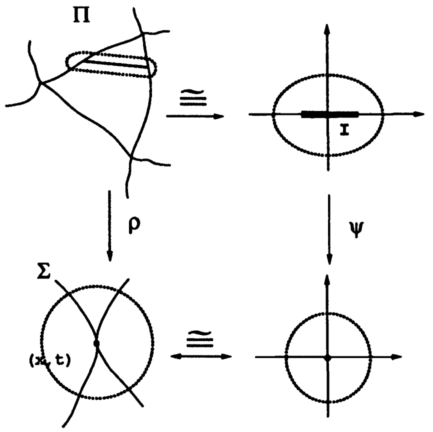


FIG. 5.1.

(i) *Case* 1 ($(x_0, t_0) \in \partial\Omega$ is an isolated point in Ω; see Fig. 5.1). In this case we may assume that $\rho^{-1}(x_0, t_0)$ is a line segment. Thus we may choose some neighborhood U of $\rho^{-1}(x_0, t_0)$ in Π and some chart from U to some neighborhood V of an interval $I = [-1, 1] \times \{0\}$ in R^2 that maps the line segment $\rho^{-1}(x_0, t_0)$ onto I.

Let ψ be a continuous map from V onto some neighborhood of the origin in R^2 such that $\psi(x) = (d(x, I)\cos(\arg\ x), d(x, I)\sin(\arg\ x))$, where $d(x, I) = \min\{\|x-x'\| : x' \in I\}$ and arg x is an argument of x, i.e., $x/\|x\| = (\cos(\arg\ x), \sin(\arg\ x))$. This maps I to the origin and maps $R^2\backslash I$ to $R^2\backslash\{0\}$ homeomorphically. Note that ρ is a continuous map from $\rho^{-1}(x_0, t_0)$ to (x_0, t_0) and that it locally maps $\Pi\backslash\rho^{-1}(x_0, t_0)$ to $\Sigma\backslash(x_0, t_0)$ homeomorphically. Thus, we conclude that Σ is a 2-dimensional topological manifold in the neighborhood of (x_0, t_0). (Note that even if there exist an infinite number of points in $\partial\Omega$ that are isolated points in Ω, this assertion remains true.)

(ii) *Case* 2 ($(x_0, t_0) \in \partial\Omega$ is an accumulation point in Ω). In this case the following two cases may occur: either $\{(x_0, t_0)\}$ is a path-connected component in Ω, or it is not.

Subcase 2.1 ($\{(x_0, t_0)\}$ is not a path-connected component of Ω; see Fig. 5.2). In this case we almost can use the same argument that we used in Case 1. We may choose some neighborhood U of $\rho^{-1}(x_0, t_0)$ in Π and some chart from U to some neighborhood V of an interval $I = [-1, 1] \times \{0\}$ in R^2 that maps the line segment $\rho^{-1}(x_0, t_0)$ onto I. In this case $\rho^{-1}(x, t)$ is a line segment for all points (x, t) of Ω, so that $\rho^{-1}(\Omega)$ is a ruled surface with a boundary. We may assume that the chart maps $\rho^{-1}(\Omega) \cap U$ to $I' = [-1, 1] \times (-\infty, 0] \cap V$. Choose a continuous map ψ given in Case 1. In this case we can identify two points of $\partial\psi(I')$. Thus we conclude that Σ is a 2-dimensional topological manifold in the neighborhood of (x_0, t_0). (Note that, even if there exists an infinite number of points in $\partial\Omega$ that are path-connected with some other points in Ω, this assertion is still true.)

Subcase 2.2 ($\{(x_0, t_0)\}$ is a path-connected component of Ω; see Fig. 5.3). We may assume that Ω is a sequence $\{(x_i, t_i)\}_{i=1,2,\ldots}$ of the points that accumulates to (x_0, t_0). Let l_i be the line segments $\rho^{-1}(x_i, t_i)$ $(i = 1, 2, \ldots)$, and let l_0 be the line segment $\rho^{-1}(x_0, t_0)$. As in Case 1, we may choose some neighborhood U of l_0 and some chart from U to some neighborhood V of an interval $I = [-1, 1] \times \{0\}$ in R^2

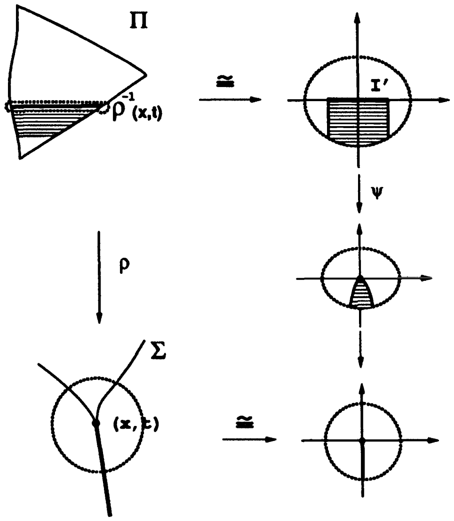


FIG. 5.2.

that maps the line segment l_0 onto I since Π is locally flat in U. We may assume that line segments l_i $(i = 1, 2, \dots)$ are in U and that they converge to l_0.

Note that V is homeomorphic to the interior of a disc D^2. By identifying its boundary with the boundary of another disc D^2 we obtain a sphere S^2. Since l_0 is a 1-dimensional manifold (with two end points) in the compact 2-dimensional manifold S^2, if we delete l_0, then we obtain R^2 and l_i's are 1-dimensional manifolds (with two end points) that are discrete from each other in this space (divergent to $\{\infty\}$). Therefore, each l_i $(i = 1, 2, \dots)$ is reduced to Case 1, and we may choose a continuous map ψ from R^2 to R^2 that maps l_i to some point $\{x_i\}$ for each $i = 1, 2, \dots$ that diverges to $\{\infty\}$ and maps $R^2 \setminus \cup_{i=1}^{\infty} l_i$ to $R^2 \setminus \cup_{i=1}^{\infty} \{x_i\}$ homeomorphically. Next, we identify l_0 with $\{\infty\}$ and we obtain a sphere S^2, again. Last, we separate a closed disc from S^2, and we obtain an open disc (homeomorphic to V).

Now ρ maps line segment $l_i = \rho^{-1}(x_i, t_i)$ to (x_i, t_i) for $i = 0, 1, 2, \dots$ and maps $\Pi \setminus \cup_{i=0}^{\infty} \rho^{-1}(x_i, t_i)$ to $\Sigma \setminus \cup_{i=0}^{\infty} \{(x_i, t_i)\}$ homeomorphically. Thus we conclude that Σ is a 2-dimensional topological manifold in the neighborhood of (x_0, t_0).

The case $\operatorname{corank} A_J(x, t) = 2$ is clear from an argument similar to the one above. Thus each connected component of Σ is a 2-dimensional topological manifold without boundary. □

Appendix. LEMMA A.1 (for the proof of Lemma 4.2). *Let N be an $n \times n$ symmetric matrix, and let A^k be an $n \times k$ matrix $(n \geq k)$. Let M be an $(n + k) \times (n + k - 2)$ matrix of the form*

$$M = \left(\begin{array}{c|c} N & A^{k-2} \\ \hline A^{k\mathrm{T}} & 0 \end{array} \right),$$

where A^{k-2} is an $n \times (k - 2)$ matrix with the first $(k - 2)$ columns of A^k $(k \geq 2)$. Suppose that M is of full rank (i.e., $\operatorname{rank} M = n + k - 2$). Then there exists an index

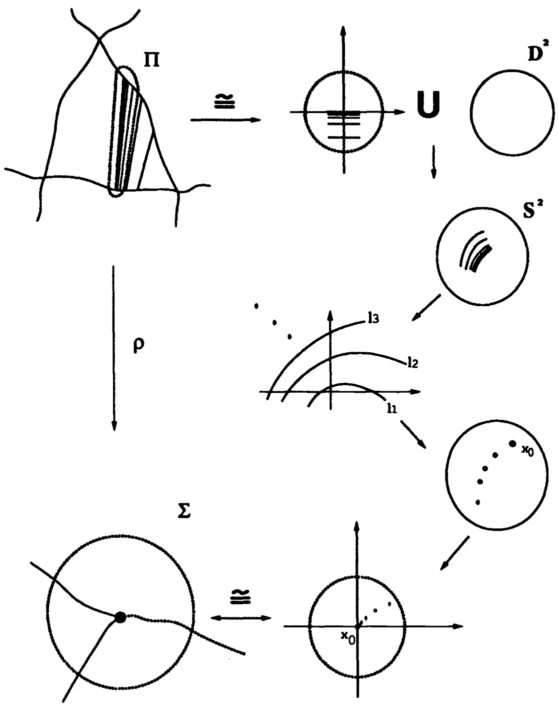


FIG. 5.3.

set J with $\{1,\ldots,k-2\} \subseteq J \subseteq \{1,\ldots,k\}$ such that

$$M^J = \left(\begin{array}{c|c} N & A^J \\ \hline A^{J\top} & 0 \end{array}\right)$$

is nonsingular, where A^J is an $n \times |J|$ submatrix of A^k that contains A^{k-2} as a submatrix.

Proof. We are interested only in the rank condition, and so we can use an elementary transformation freely. Let $\operatorname{rank} A^k = k - r(r = 0, 1 \text{ or } 2)$. Without loss of generality, we may assume that

$$A^k = \left(\begin{array}{c|c} E_{k-r} & 0 \\ \cline{1-1} 0 & \end{array}\right)$$

is an $n \times k$ matrix and that

$$A^{k-2} = \begin{pmatrix} E_{k-2} \\ 0 \end{pmatrix}$$

is an $n \times (k-2)$ matrix (since M is of full rank, so is A^{k-2}).

Thus, we obtain

$$\operatorname{rank} M = \operatorname{rank} \left(\begin{array}{cc|c} N & & E_{k-2} \\ & & 0 \\ \hline E_{k-r} & & 0 \\ & 0 & \end{array}\right) = \operatorname{rank} \left(\begin{array}{cc|c} 0 & & E_{k-2} \\ \cline{2-3} & C & 0 \\ \hline E_{k-r} & & 0 \\ & 0 & \end{array}\right),$$

where C is an $(n-k+2)\times(n-k+r)$ matrix. Note that C is a lower-right submatrix of N, and hence the $(n-k+r)\times(n-k+r)$ lower (square) submatrix D of C is symmetric. Let $\operatorname{rank} D = s$. Since M is of full rank, so is C. We may assume

$$C = \left(\frac{*}{D}\right) = \left(\begin{array}{c|c} 0 & 0 \\ \cline{2-2} & E_{n-k+r-s} \\ \cline{1-2} E_s & 0 \\ \cline{1-1} 0 & \end{array}\right).$$

Remember that the lower square submatrix of N is symmetric.

Let A^J be the first $(2k-n-2r+s)$ columns of A^k. Set $u = 2k-n-2r+s$. Then we have the form

$$M^J = \left(\begin{array}{c|c} N & A^J \\ \hline A^{J\top} & 0 \end{array}\right) = \left(\begin{array}{c|c|c} 0 & 0 & E_u \\ \hline 0 & \begin{array}{ccc} & & E_{n-k+r-s} \\ & E_s & \\ E_{n-k+r-s} & & \end{array} & 0 \\ \hline E_u & 0 & 0 \end{array}\right).$$

Therefore, the assertion is clear. □

Acknowledgments. The authors thank two anonymous referees for useful comments and suggestions.

REFERENCES

[1] K. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer, *Nonlinear Parametric Optimization*, Akademie-Verlag, Berlin, 1982.

[2] C. Berge, *Topological Spaces—including a Treatment of Multi-Valued Functions, Vector Spaces and Convexity*, Macmillan, New York, 1963.

[3] C. G. Gauvin, *A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming*, Math. Programming, 12 (1977), pp. 136–138.

[4] H. Th. Jongen, P. Jonker, and F. Twilt, *On one-parameter families of sets defined by (in)equality constraints*, Nieuw Arch. Wisk. (3), 30 (1982), pp. 307–322.

[5] ———, *Nonlinear Optimization in R^n*, I: *Morse Theory, Chebychev Approximation*, Peter Lang Verlag, Frankfurt, 1983.

[6] ———, *Critical sets in parametric optimization*, Math. Programming, 34 (1986), pp. 333–353.

[7] ———, *Nonlinear Optimization in R^n*, II: *Transversality, Flows, Parametric Aspects*, Peter Lang Verlag, Frankfurt, 1986.

[8] ———, *One-parameter families of optimization problems: Equality constraints*, J. Optim. Theory Appl., 48 (1986), pp. 141–161.

[9] ———, *Parametric optimization: The Kuhn–Tucker set*, in Parametric Optimization and Related Topics, Vol. 35, J. Guddat, H. Th. Jongen, B. Kummer, and F. Nožička, eds., Akademie-Verlag, Berlin, 1987, pp. 196–208.

[10] H. Th. Jongen and G. W. Weber, *On parametric nonlinear programming*, Ann. Oper. Res., 27 (1990), pp. 253–284.

[11] M. Kojima, *Strongly stable stationary solutions in nonlinear programs*, in Analysis and Computation of Fixed Points, S. M. Robinson, ed., Academic Press, New York, 1980, pp. 93–138.

[12] M. Kojima and R. Hirabayashi, *Continuous deformations of nonlinear programs*, Math. Programming Stud., 21 (1984), pp. 150–198.

[13] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1973; 2nd ed., 1984.

[14] 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.

[15] A. B. Poore and C. A. Tiahrt, *Bifurcation problems in nonlinear parametric programming*, Math. Programming, 39 (1987), pp. 189–205.

[16] S. M. Robinson, *Generalized equations and their solutions, part* II: *Applications to nonlinear programming*, Math. Programming Stud., 19 (1982), pp. 200–221.

[17] S. SCHECTER, *Structure of the first-order solution set for a class of nonlinear programs with parameters*, Math. Programming, 34 (1986), pp. 84–110.

[18] S. SHINDOH, R. HIRABAYASHI, AND T. MATSUMOTO, *Structure of solution set to nonlinear programs with two parameters,* I: *Change of stationary indices*, in Parametric Optimization and Related Topics II, J. Guddat, H. Th. Jongen, B. Kummer, and F. Nožička, eds., Academie-Verlag, Berlin, 1989, pp. 168–175.

[19] C. A. TIAHRT AND A. B. POORE, *A bifurcation analysis of the nonlinear parametric programming problem*, Math. Programming, 47 (1990), pp. 117–141.

SIAM J. OPTIMIZATION
Vol. 3, No. 3, pp. 582–608, August 1993

NUMERICAL EXPERIENCE WITH LIMITED-MEMORY QUASI-NEWTON AND TRUNCATED NEWTON METHODS*

X. ZOU†, I. M. NAVON‡, M. BERGER§, K. H. PHUA¶,
T. SCHLICK‖, AND F. X. LE DIMET**

Abstract. Computational experience with several limited-memory quasi-Newton and truncated Newton methods for unconstrained nonlinear optimization is described. Comparative tests were conducted on a well-known test library [J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *ACM Trans. Math. Software*, 7 (1981), pp. 17–41], on several synthetic problems allowing control of the clustering of eigenvalues in the Hessian spectrum, and on some large-scale problems in oceanography and meteorology. The results indicate that among the tested limited-memory quasi-Newton methods, the L-BFGS method [D. C. Liu and J. Nocedal, *Math. Programming*, 45 (1989), pp. 503–528] has the best overall performance for the problems examined. The numerical performance of two truncated Newton methods, differing in the inner-loop solution for the search vector, is competitive with that of L-BFGS.

Key words. limited-memory quasi-Newton methods, truncated Newton methods, synthetic cluster functions, large-scale unconstrained minimization

AMS subject classifications. 90C30, 93C20, 93C75, 65K10, 76C20

1. Introduction.

Limited-memory quasi-Newton (LMQN) and truncated Newton (TN) methods represent two classes of algorithms that are attractive for large-scale problems because of their modest storage requirements. They use a low and adjustable amount of storage and require the function and gradient values at each iteration. Preconditioning of the Newton equations may be used for both algorithms. In this case, additional function information (e.g., a sparse approximation to the Hessian) may also be required at each iteration. LMQN methods can be viewed as extensions of conjugate-gradient (CG) methods in which the addition of some modest storage serves to accelerate the convergence rate. TN methods attempt to retain the rapid convergence rate of classical Newton methods while economizing storage and computational requirements so as to become feasible for large-scale applications. They can be particularly powerful when structure information of the objective function is exploited [29].

LMQN originated from the works of Nazareth [21] and Perry [25], [26] and were further extended by Shanno [31], [32] resulting in the CONMIN code of Shanno and

*Received by the editors March 29, 1991; accepted for publication (in revised form) April 22, 1992. This work was supported by National Science Foundation grants ATM-8806553, CHE-9002146, and ASC-9157582 (Presidential Young Investigator Award), U.S. Department of Energy contract DE-FC0583ER250000, and the Searle Scholar program.

†Supercomputer Computations Research Institute, Florida State University, Tallahassee, Florida 32306.

‡Department of Mathematics and Supercomputer Computations Research Institute, Florida State University, Tallahassee, Florida 32306.

§Division of Applied Mathematics, Tel-Aviv University, Ramat-Aviv, Tel-Aviv 69978, Israel.

¶Department of Information and Computer Science, National University of Singapore, Kent Ridge, Singapore 1026, Singapore.

‖Courant Institute of Mathematical Sciences and Department of Chemistry, New York University, 251 Mercer Street, New York, New York 10012.

**Grenoble Laboratoire de Modelization et Calcul, B.P. 53X, 38041 Grenoble Cedex, France.

Phua [33]. Many researchers, including Buckley [1], Nazareth [22], Nocedal [23], Gill and Murray [8], and Nash [15]–[17], studied these methods. Gill and Murray proposed an LMQN method with preconditioning whose code has recently been implemented in routine E04DGF of the NAG library [14]. Buckley and Lenir [3], [4] proposed a variable-storage CG algorithm. The method becomes the usual Shanno–Phua LMQN when the available storage is minimal. Their method was implemented in code BBVSCG, recently updated and improved by Buckley [2]. More recently, the L-BFGS method of Liu and Nocedal [12] based on the limited-memory BFGS method described by Nocedal [23] was developed. L-BFGS is available as routine VA05AD of the Harwell software library. Two TN methods proposed by Nash [15]–[17] and by Schlick and Fogelson [29] have also been made available by the authors for distribution. Here the codes were tested on the variational data assimilation problems in meteorology.

Several large-scale unconstrained minimization algorithms have been previously compared. Navon and Legler [19] compared a number of different CG methods for problems in meteorology and concluded that the Shanno–Phua [33] LMQN algorithm was the most adequate for their test problems. The studies of Gilbert and Lemaréchal [7] and of Liu and Nocedal [12] indicated that the L-BFGS method is among the best LMQN methods available to date. Nash and Nocedal [18] compared the L-BFGS method with the TN method of Nash [15]–[17] on 53 problems of dimensions 10^2 to 10^4. Their results suggested that performance is correlated with the degree of nonlinearity of the objective function: for quadratic and approximately quadratic problems the TN algorithm outperformed L-BFGS, whereas for most of the highly nonlinear problems L-BFGS performed better.

The aim of this paper is to compare and analyze the performance of several LMQN methods. The most representative LMQN method is then compared with TN methods for large-scale problems in meteorology. We focus on various implementation details, such as step-size searches, stopping criteria, and other practical computational features. In §2 we briefly review the tested LMQN methods. The relationships of the different methods to one another are discussed along with practical implementation details. TN methods are briefly described in §3. In §4 we describe the various test problems used in the Moré, Garbow, and Hillstrom [13] package, the synthetic cluster problem, and some real-life large-scale problems ($\sim 10^4$ variables) from oceanography and meteorology. Discussion of the performance of the different LMQN methods and some general observations are presented in §5. In §6 the performance of TN methods for the optimal control problems in meteorology is presented. Summary and conclusions are presented in §7.

2. LMQN algorithms. The behavior of CG algorithms with inexact line searches may depart considerably from theoretical expectations. For this reason, methods such as LMQN compute a descent direction but impose much milder restrictions on the accepted step length.

LMQN algorithms have the following basic structure for minimizing $J(\mathbf{x})$, $\mathbf{x} \in \mathcal{R}^N$:

1) Choose an initial guess $\mathbf{x}_0$ and a positive definite initial approximation to the inverse Hessian matrix $\mathbf{H}_0$ (which may be chosen as the identity matrix).

2) Compute

$$\mathbf{g}_0 = \mathbf{g}(\mathbf{x}_0) = \nabla J(\mathbf{x}_0), \tag{2.1}$$

and set

$$\mathbf{d}_0 = -\mathbf{H}_0\mathbf{g}_0. \tag{2.2}$$

3) For $k = 0, 1, \ldots$, set

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \tag{2.3}$$

where α_k is the step size (see below).

4) Compute

$$\mathbf{g}_{k+1} = \nabla J(\mathbf{x}_{k+1}). \tag{2.4}$$

5) Check for restarts (discussed below).

6) Generate a new search direction $\mathbf{d}_{k+1}$ by setting

$$\mathbf{d}_{k+1} = -\mathbf{H}_{k+1}\mathbf{g}_{k+1}. \tag{2.5}$$

7) Check for convergence: If

$$\|\mathbf{g}_{k+1}\| \leq \epsilon \ \max\{1, \|\mathbf{x}_{k+1}\|\}, \tag{2.6}$$

stop, where $\epsilon = 10^{-5}$. Otherwise, continue from step 3.

LMQN methods combine the advantages of the CG low storage requirement with the computational efficiency of the quasi-Newton (Q-N) method. They avoid storage of the approximate Hessian matrix by building several rank-one or rank-two matrix updates. In practice, the BFGS update formula [12], [24] forms an approximate inverse Hessian from $\mathbf{H}_0$ and k pairs of vectors $(\mathbf{q}_i, \mathbf{p}_i)$, where $\mathbf{q}_i = \mathbf{g}_{i+1} - \mathbf{g}_i$ and $\mathbf{p}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$ for $i \geq 0$. Since $\mathbf{H}_0$ is generally taken to be the identity matrix or some other diagonal matrix, the pairs $(\mathbf{q}_i, \mathbf{p}_i)$ are stored instead of $\mathbf{H}_k$, and $\mathbf{H}_k\mathbf{g}_k$ is computed by a recursive algorithm. All the LMQN methods presented below fit into this conceptual framework. They differ only in the selection of the vector couples $(\mathbf{q}_i, \mathbf{p}_i)$, the choice of $\mathbf{H}_0$, the method for computing $\mathbf{H}_k\mathbf{g}_k$, the line-search implementation, and the handling of restarts.

2.1. CONMIN.

The LMQN method of Shanno and Phua [33] is a two-step LMQN-like CG method that incorporates Beale restarts. Only seven vectors of storage are necessary.

Step sizes are obtained by using Davidon's [5] cubic interpolation method to satisfy the following Wolfe [34] conditions:

$$J(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \leq J(\mathbf{x}_k) + \beta' \alpha_k \mathbf{g}_k^T \mathbf{d}_k, \tag{2.7}$$

$$\left| \frac{\nabla J(\mathbf{x}_k + \alpha_k \mathbf{d}_k)^T \mathbf{d}_k}{\mathbf{g}_k^T \mathbf{d}_k} \right| \leq \beta, \tag{2.8}$$

where $\beta' = 0.0001$ and $\beta = 0.9$.

The following restart criterion is used:

$$|\mathbf{g}_{k+1}^T \mathbf{g}_k| \geq 0.2 \|\mathbf{g}_{k+1}\|^2. \tag{2.9}$$

The new search direction $\mathbf{d}_{k+1}$, defined by (2.5), is obtained by setting

$$\mathbf{H}_{k+1} = \hat{\mathbf{H}}_k - \frac{\mathbf{p}_k \mathbf{q}_k^T \hat{\mathbf{H}}_k + \hat{\mathbf{H}}_k \mathbf{q}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k} + \left(1 + \frac{\mathbf{q}_k^T \hat{\mathbf{H}}_k \mathbf{q}_k}{\mathbf{p}_k^T \mathbf{q}_k}\right) \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k}. \tag{2.10}$$

If a restart is satisfied, (2.5) is changed to

$$\mathbf{d}_{k+1} = -\hat{\mathbf{H}}_k \mathbf{g}_{k+1}, \tag{2.11}$$

where

$$\hat{\mathbf{H}}_k = \gamma_t \left(\mathbf{I} - \frac{\mathbf{p}_t \mathbf{q}_t^T + \mathbf{q}_t \mathbf{p}_t^T}{\mathbf{p}_t^T \mathbf{q}_t} + \frac{\mathbf{q}_t^T \mathbf{q}_t}{\mathbf{p}_t^T \mathbf{q}_t} \frac{\mathbf{p}_t \mathbf{p}_t^T}{\mathbf{p}_t^T \mathbf{q}_t} \right) + \frac{\mathbf{p}_t \mathbf{p}_t^T}{\mathbf{p}_t^T \mathbf{q}_t}. \tag{2.12}$$

Here the subscript t represents the last step of the previous cycle for which a line search was made. The parameter $\gamma_t = \mathbf{p}_t^T \mathbf{q}_t / \mathbf{q}_t^T \mathbf{q}_t$ is obtained by minimizing the condition number $\mathbf{H}_t^{-1} \mathbf{H}_{t+1}$ [33].

The Shanno and Phua method implemented in CONMIN uses two couples of vectors $\mathbf{q}$ and $\mathbf{p}$ to build its current approximation of the Hessian matrix. The advantage of CONMIN is that it generates descent directions automatically without requiring exact line searches as long as $(\mathbf{q}_k, \mathbf{p}_k)$ are positive at each iteration. This can be ensured by satisfying the second Wolfe condition (2.8) in the line search. However, CONMIN cannot take advantage of additional storage that might be available.

2.2. E04DGF. The Gill and Murray nonlinear unconstrained minimization algorithm is a two-step LMQN method with preconditioning and restarts. The amount of working storage required by this method is $12N$ real words of working space.

The step size is determined as follows. Let $\{\alpha^j, j = 1, 2, \cdots, \}$ define a sequence of points that tend in the limit to a local minimizer of the cost function along the direction $\mathbf{d}_k$. This sequence may be computed by means of a safeguarded polynomial interpolation algorithm. A choice of the initial step length is the one suggested by Davidon [5]:

$$\alpha^0 = \begin{cases} -2(J_k - J_{\text{est}})/\mathbf{g}_k^T \mathbf{d}_k & \text{if } -2(J_k - J_{\text{est}})/\mathbf{g}_k^T \mathbf{d}_k \leq 1, \\ 1 & \text{if } -2(J_k - J_{\text{est}})/\mathbf{g}_k^T \mathbf{d}_k > 1. \end{cases} \tag{2.13}$$

Here J_{est} represents an estimate of the cost function at the solution point. Let t be the first index of this sequence that satisfies

$$|\nabla J(\mathbf{x}_k + \alpha^t \mathbf{d}_k)^T \mathbf{d}_k| \leq -\eta \mathbf{g}_k^T \mathbf{d}_k, \qquad 0 \leq \eta \leq 1. \tag{2.14}$$

The method finds the smallest nonnegative integer r such that

$$J_k - J(\mathbf{x}_k + 2^{-r} \alpha^t \mathbf{d}_k) \geq -2^{-r} \alpha^t \mu \mathbf{g}_k^T \mathbf{d}_k, \quad 0 \leq \mu \leq \frac{1}{2}, \tag{2.15}$$

and then sets $\alpha_k = s^{-r} \alpha^t$.

A restart is required if one of the Powell restart criteria (2.9) or the condition

$$-1.2 \|\mathbf{g}_{k+1}\|_2^2 \leq \mathbf{g}_{k+1}^T \mathbf{d}_{k+1} \leq -0.8 \|\mathbf{g}_{k+1}\|_2^2 \tag{2.16}$$

is satisfied [27].

The new search direction is generated by (2.5), where $\mathbf{H}_{k+1}$ is calculated by the following two-step BFGS formula:

$$\mathbf{U}_2 = \mathbf{U}_1 - \frac{1}{\mathbf{q}_k^T \mathbf{p}_k} (\mathbf{U}_1 \mathbf{q}_k \mathbf{p}_k^T + \mathbf{p}_k \mathbf{q}_k^T \mathbf{U}_1) + \frac{1}{\mathbf{q}_k^T \mathbf{p}_k} \left(1 + \frac{\mathbf{q}_k^T \mathbf{U}_1 \mathbf{q}_k}{\mathbf{q}_k^T \mathbf{p}_k} \mathbf{p}_k \mathbf{p}_k^T \right), \tag{2.17}$$

$$\mathbf{H}_{k+1} = \mathbf{U}_2 - \frac{1}{\mathbf{q}_k^T\mathbf{p}_k}(\mathbf{U}_2\mathbf{q}_k\mathbf{p}_k^T + \mathbf{p}_k\mathbf{q}_k^T\mathbf{U}_2) + \frac{1}{\mathbf{q}_k^T\mathbf{p}_k}\left(1 + \frac{\mathbf{q}_k^T\mathbf{U}_2\mathbf{q}_k}{\mathbf{q}_k^T\mathbf{p}_k}\mathbf{p}_k\mathbf{p}_k^T\right). \tag{2.18}$$

If a restart is indicated, the following self-scaling update method [31], [32] is used instead of $\mathbf{U}_2$:

$$\hat{\mathbf{U}}_2 = \gamma\mathbf{U}_1 - \gamma\frac{1}{\mathbf{q}_t^T\mathbf{p}_t}(\mathbf{U}_1\mathbf{q}_t\mathbf{p}_t^T + \mathbf{p}_t\mathbf{q}_t^T\mathbf{U}_1) + \frac{1}{\mathbf{q}_t^T\mathbf{p}_t}\left(1 + \gamma\frac{\mathbf{q}_t^T\mathbf{U}_1\mathbf{q}_t}{\mathbf{q}_t^T\mathbf{p}_t}\mathbf{p}_t\mathbf{p}_t^T\right), \tag{2.19}$$

where $\gamma = \mathbf{q}_t^T\mathbf{p}_t/\mathbf{q}_t^T\mathbf{U}_1\mathbf{q}_t$ and $\mathbf{U}_1$ is a diagonal preconditioning matrix rather than the identity matrix.

2.3. L-BFGS. The LMQN algorithm L-BFGS [12] was chosen as one of the candidate minimization techniques to be tested since it accommodates variable storage, which is crucial in practice for large-scale minimization problems. The method abandons the restart procedure. The update formula generates matrices by using information from the last m Q-N iterations, where m is the number of Q-N updates supplied by the user (generally, $3 \leq m \leq 7$). After $2Nm$ storage locations are exhausted, the Q-N matrix is updated by replacing the oldest information by the newest information. Thus the Q-N approximation of the inverse Hessian matrix is continuously updated.

In the line search a unit step length is always tried first, and only if it does not satisfy the Wolfe condition is a cubic interpolation performed. This ensures that L-BFGS resembles the (full-memory) BFGS method as much as possible while being as economical as possible for large-scale problems, for which the quadratic termination properties are generally not very meaningful.

$\mathbf{H}_{k+1}$ of (2.5) is obtained by the following procedure. Let $\hat{m} = \min\{k, m-1\}$. Then update $\mathbf{H}_0$ $\hat{m}+1$ times by using the vector pairs $(\mathbf{q}_j, \mathbf{p}_j)_{j=k-\hat{m}}^{k}$, where $\mathbf{p}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\mathbf{q}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$, and

$$\begin{aligned}\mathbf{H}_{k+1} =& (\mathbf{v}_k^T \cdots \mathbf{v}_{k-\hat{m}}^T)\mathbf{H}_0(\mathbf{v}_{k-\hat{m}} \cdots \mathbf{v}_k)\\ &+ \rho_{k-\hat{m}}(\mathbf{v}_k^T \cdots \mathbf{v}_{k-\hat{m}+1}^T)\mathbf{p}_{k-\hat{m}}\mathbf{p}_{k-\hat{m}}^T(\mathbf{v}_{k-\hat{m}+1} \cdots \mathbf{v}_k)\\ &+ \rho_{k-\hat{m}+1}(\mathbf{v}_k^T \cdots \mathbf{v}_{k-\hat{m}+2}^T)\mathbf{p}_{k-\hat{m}+1}\mathbf{p}_{k-\hat{m}+1}^T(\mathbf{v}_{k-\hat{m}+2} \cdots \mathbf{v}_k)\\ &\vdots\\ &+ \rho_k\mathbf{p}_k\mathbf{p}_k^T.\end{aligned} \tag{2.20}$$

Here $\rho_k = 1/(\mathbf{q}_k^T\mathbf{p}_k)$, $\mathbf{v}_k = \mathbf{I} - \rho_k\mathbf{q}_k\mathbf{p}_k^T$, and $\mathbf{I}$ is the identity matrix.

Two options for the above procedure are offered in the code. One performs a more accurate line search by using a small value for β in (2.8) (e.g., $\beta = 10^{-2}$ or $\beta = 10^{-3}$); this is advantageous when the function and gradient evaluations are inexpensive. The other uses simple scaling to reduce the number of iterations. In general it is preferable to replace $\mathbf{H}_0$ of (2.20) by $\mathbf{H}_k^0$ as one proceeds, so that $\mathbf{H}_0$ incorporates more up-to-date information according to one of the following:

M1: $\mathbf{H}_k^0 = \mathbf{H}_0$ (no scaling).

M2: $\mathbf{H}_k^0 = \gamma_0\mathbf{H}_0$, $\gamma_0 = \mathbf{q}_0^T\mathbf{p}_0/\|\mathbf{q}_0\|^2$ (only initial scaling).

M3: $\mathbf{H}_k^0 = \gamma_k\mathbf{H}_0$, $\gamma_k = \mathbf{q}_k^T\mathbf{p}_k/\|\mathbf{q}_k\|^2$.

Since Liu and Nocedal [12] reported that M3 is the most effective scaling, we use it in all our numerical experiments.

2.4. BBVSCG. BBVSCG implements the LMQN method of Buckley and Lenir and may be viewed as an extension of the Shanno and Phua method. Extra storage space can be accommodated.

The method begins by performing the BFGS Q-N update algorithm. When all available storage is exhausted, the current BFGS approximation to the inverse Hessian matrix is retained as a preconditioning matrix. The method then continues by performing preconditioned memoryless Q-N steps, equivalent to the preconditioned CG method with exact line searches. The memoryless Q-N steps are then repeated until the criterion of Powell [27] indicates that a restart is required. At that time all the BFGS corrections are discarded and a new approximation to the preconditioning matrix begins.

For the line search, when $k \leq m$, a step size of $\alpha = 1$ is tried. A line search using cubic interpolation is applied only if the new point does not satisfy $\mathbf{p}_k^T \mathbf{q}_k > 0$. For $k > m$, $\alpha_k = -\mathbf{g}_k^T \mathbf{d}_k / \mathbf{d}_k^T \mathbf{H}_k \mathbf{d}_k$. At least one quadratic interpolation is performed before α_k is accepted.

The search direction is calculated by $\mathbf{d}_{k+1} = -\mathbf{H}_k \mathbf{g}_k$ instead of by (2.5), where $\mathbf{H}_k$ is obtained as follows.

(i) If $k = 1$, use a scaled Q-N BFGS formula:

$$\mathbf{H}_1 = \mathbf{\Phi}_0 - \frac{\mathbf{\Phi}_0 \mathbf{q}_k \mathbf{p}_k^T + \mathbf{p}_k \mathbf{q}_k^T \mathbf{\Phi}_0}{\mathbf{p}_k^T \mathbf{q}_k} + \left(1 + \frac{\mathbf{q}_k^T \mathbf{\Phi}_0 \mathbf{q}_k}{\mathbf{p}_k^T \mathbf{q}_k}\right) \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k}, \tag{2.21}$$

where $\mathbf{\Phi}_0$ is defined as $\mathbf{\Phi}_0 = (\omega_0 / v_0) \mathbf{H}_0$, $\omega_0 = \mathbf{p}_0^T \mathbf{q}_0$, and $v_0 = \mathbf{q}_0^T \mathbf{H}_0 \mathbf{q}_0$.

(ii) If $1 < k \leq m$, use the Q-N BFGS formula:

$$\mathbf{H}_k = \mathbf{H}_{k-1} - \frac{\mathbf{H}_{k-1} \mathbf{q}_k \mathbf{p}_k^T + \mathbf{p}_k \mathbf{q}_k^T \mathbf{H}_{k-1}}{\mathbf{p}_k^T \mathbf{q}_k} + \left(1 + \frac{\mathbf{q}_k^T \mathbf{H}_{k-1} \mathbf{q}_k}{\mathbf{p}_k^T \mathbf{q}_k}\right) \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k}. \tag{2.22}$$

(iii) If $k > m$, use the preconditioned memoryless Q-N formula:

$$\mathbf{H}_k = \mathbf{H}_m - \frac{\mathbf{p}_k \mathbf{q}_k^T \mathbf{H}_m + \mathbf{H}_m \mathbf{q}_k \mathbf{p}_k^T}{\mathbf{p}_k \mathbf{q}_k^T} + \left(1 + \frac{\mathbf{q}_k^T \mathbf{H}_m \mathbf{q}_k}{\mathbf{p}_k \mathbf{q}_k^T}\right) \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k \mathbf{q}_k^T}, \tag{2.23}$$

where $\mathbf{H}_m$ is used as a preconditioner.

The matrix $\mathbf{H}_k$ need not be stored since only matrix–vector products $(\mathbf{H}_k \mathbf{v})$ are required. These are calculated from

$$\mathbf{H}_k \mathbf{v} = \mathbf{H}_q \mathbf{v} - \left[\frac{\mathbf{u}_k^T \mathbf{v}}{\omega_k} - \left(1 + \frac{v_k}{\omega_k}\right) \frac{\mathbf{p}_k^T \mathbf{v}}{\omega_k}\right] - \frac{\mathbf{p}_k^T \mathbf{v}}{\omega_k} \mathbf{u}_k, \tag{2.24}$$

where $v_k = \mathbf{q}_k^T \mathbf{H}_q \mathbf{q}_k$, $\omega_k = \mathbf{p}_k^T \mathbf{q}_k$, and $\mathbf{u}_k = \mathbf{H}_q \mathbf{q}_k$. The subscript q is either $k - 1$ or m, depending on whether $k \leq m$ or $k > m$. If one applies (2.24) recursively, the following formula is obtained:

$$\mathbf{H}_q \mathbf{v} = \mathbf{H}_0 \mathbf{v} - \sum_{j=1}^{q} \left\{ \left[\frac{\mathbf{u}_j^T \mathbf{v}}{\omega_j} - \left(1 + \frac{v_j}{\omega_j}\right) \frac{\mathbf{p}_j^T \mathbf{v}}{\omega_j}\right] - \frac{\mathbf{p}_j^T \mathbf{v}}{\omega_j} \mathbf{u}_j \right\}. \tag{2.25}$$

The total storage required for the matrices $\mathbf{H}_1, \ldots, \mathbf{H}_m$ consists of $m(2N + 2)$ locations.

If $k > m$, a restart test is implemented. Restarts will take place if (2.9) and (2.16) are satisfied. In that case $\mathbf{H}_m$ is discarded, k is set to 1, and the algorithm continues from step 1.

Both the L-BFGS and Buckley–Lenir methods allow the user to specify the number of Q-N updates m. When $m = 1$, BBVSCG reduces to CONMIN, whereas when $m = \infty$, both L-BFGS and the Buckley–Lenir methods are identical to the Q-N BFGS method (implemented in the CONMIN-BFGS code).

3. TN methods. Just as LMQN methods attempt to combine modest storage and computational requirements of CG methods with the convergence properties of the standard Q-N methods, TN methods attempt to retain the rapid (quadratic) convergence rate of classic Newton methods while making storage and computational requirements feasible for large-scale applications [6]. Recall that Newton methods for minimizing a multivariate function $J(\mathbf{x}_k)$ are iterative techniques based on minimizing a local quadratic approximation to J at every step. The quadratic model of J at a point $\mathbf{x}_k$ along the direction of a vector $\mathbf{d}_k$ can be written as

$$J(\mathbf{x}_k + \mathbf{d}_k) \approx J(\mathbf{x}_k) + \mathbf{g}_k^T\mathbf{d}_k + \frac{1}{2}\mathbf{d}_k^T\mathbf{H}_k\mathbf{d}_k, \tag{3.1}$$

where $\mathbf{g}_k$ and $\mathbf{H}_k$ denote the gradient and Hessian, respectively, of J at $\mathbf{x}_k$. Minimization of this quadratic approximation produces a linear system of equations for the search vector $\mathbf{d}_k$ that are known as the Newton equations:

$$\mathbf{H}_k\mathbf{d}_k = -\mathbf{g}_k. \tag{3.2}$$

In the modified Newton framework a sequence of iterates is generated from $\mathbf{x}_0$ by the rule $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{d}_k$. The vector $\mathbf{d}_k$ is obtained as the solution (or approximate solution) of the system (3.2) or, possibly, a modified version of it, where some positive definite approximation to $\mathbf{H}_k$, $\tilde{\mathbf{H}}_k$, replaces $\mathbf{H}_k$.

When an approximate solution is used, the method is referred to as a *truncated* Newton method because the solution process of (3.2) is not carried to completion. In this case $\mathbf{d}_k$ may be considered satisfactory when the residual vector $\mathbf{r}_k = \mathbf{H}_k\mathbf{d}_k + \mathbf{g}_k$ is sufficiently small. Truncation may be justified since accurate search directions are not essential in regions far away from local minima. For such regions any descent direction suffices, and so the effort expended in solving the system accurately is often unwarranted. However, as a solution of the optimization problem is approached, the quadratic approximation of (3.1) is likely to become more accurate and a smaller residual may be more important. Thus the truncation criterion should be chosen to enforce a smaller residual systematically as minimization proceeds. One such effective strategy requires

$$\|\mathbf{r}_k\| \leq \eta_k\|\mathbf{g}_k\|, \tag{3.3}$$

where

$$\eta_k = \min\left\{\frac{c}{k}, \|\mathbf{g}_k\|\right\}, \quad c \leq 1. \tag{3.4}$$

Indeed, it can be shown that quadratic convergence can still be maintained [6]. Other truncation criteria have also been discussed [15], [16], [29].

The quadratic subproblem of computing an approximate search direction at each step is accomplished through some iterative scheme. This produces a nested iteration structure: an outer loop for updating $\mathbf{x}_k$ and an inner loop for computing $\mathbf{d}_k$.

The *linear* CG method is attractive for large-scale problems because of its modest computational requirements and theoretical convergence in at most N iterations [9]. However, since CG methods were developed for positive definite systems, adaptations must be made in the present context where the Hessian may be indefinite. Typically, this is handled by terminating the inner loop (at iteration q) when a direction of negative curvature is detected ($\mathbf{d}_q^T \mathbf{H}_k \mathbf{d}_q < \xi$, where ξ is a small positive tolerance such as 10^{-10}); an exit direction that is guaranteed to be a descent direction is then chosen [6], [29]. An alternative procedure to the linear CG for the inner loop is based on the Lanczos factorization [9], which works for symmetric but not necessarily positive definite systems. It is important to note that different procedures for the inner loop can lead to a very different overall performance in the minimization [28].

Implementations of two TN packages are examined in this work: TN1, developed by Nash [15]–[17], which uses a modified Lanczos algorithm with an automatically supplied diagonal preconditioner, and TN2 (TNPACK) developed by Schlick and Fogelson [29] (see also [30]), designed for structured separable problems for which the user provides a sparse preconditioner for the inner loop. In TN2 a sparse modified Cholesky factorization based on the Yale Sparse Matrix Package is used to factor the preconditioner, which need not be positive definite (computational chemistry problems, where such situations occur, provided motivation for the method). Two modified Cholesky factorizations have been implemented in TN2 [28]. Although we have not yet formulated a preconditioner for our meteorology application, we intend to focus future efforts on formulating an efficient preconditioner for this package. Here we report only results for which no preconditioning is used in TN2. Although it is clear that performance must suffer, our results provide further perspective. Full algorithmic descriptions of the TN codes can be found in the original cited works.

4. Testing problems.

Moré, Garbow, and Hillstrom [13] developed a relatively large collection of carefully coded test functions of different degrees of difficulty and designed very simple procedures for testing the reliability and robustness of the optimization software. We used these problems to test the different LMQN methods.

The test problems of [13] involve Hessians of varying spectral condition numbers and eigenvalues, and the eigenvalues are generally of unknown and uncontrollable dispersion. A synthetic test function with a controllable spectrum of clustered eigenvalues was thus also tested.

Two representative real-life large-scale unconstrained minimization applications from meteorology and oceanography were also examined to compare the performances of the LMQN and TN methods. The number of variables for these large-scale problems ranges from 7330 to 14,763.

4.1. Standard library test problems.

All of the 18 test problems of Moré, Garbow, and Hillstrom for unconstrained minimization have the following composition:

$$J(\mathbf{x}) = \sum_{i=1}^{m} f_i^2(\mathbf{x}), \quad m \leq N, \quad \mathbf{x} \in \mathcal{R}^N. \tag{4.1}$$

These problems were all minimized by using both the recommended standard starting points $\mathbf{x}_0$ as well as by using nonstandard starting points, taken as $10\mathbf{x}_0$ and $100\mathbf{x}_0$. The vectors $\mathbf{x}_0$ and $100\mathbf{x}_0$ are regarded as being close to and far away from the solution, respectively; it is not unusual for unconstrained minimization algorithms to succeed with an initial guess of $\mathbf{x}_0$ but fail with an initial guess of either $10\mathbf{x}_0$ or $100\mathbf{x}_0$.

4.2. Synthetic cluster function problems. Consider the quadratic objective function

$$J(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x}, \tag{4.2}$$

where $\mathbf{x}$ is a vector of N variables and $\mathbf{H}$ is an $N \times N$ positive definite matrix of real entries. There exists then a real orthogonal matrix $\mathbf{Q}$ such that

$$\mathbf{Q}^T\mathbf{H}\mathbf{Q} = \operatorname{diag}(\lambda_1, \ldots, \lambda_N), \tag{4.3}$$

where the ith column of $\mathbf{Q}$ is the ith eigenvector corresponding to the ith eigenvalue λ_i. The objective function can be written as

$$J(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{D}\mathbf{Q}^T\mathbf{x}. \tag{4.4}$$

The orientation and shape of this N-dimensional quadratic surface is a function of $\mathbf{Q}$ and $\mathbf{D}$: the directions of the principal axes of this hyperellipsoid are determined by the directions of the eigenvectors, and the lengths of the axes are determined by the eigenvalues. The axes' lengths are inversely proportional to the square root of the corresponding eigenvalues.

Consider a quadratic objective function defined by

$$J(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{N}(D_{ii}x_i)^2, \tag{4.5}$$

where

$$D_{ii} = \left(1 + \frac{i - M_{k-1} - \lfloor\frac{N_k}{2}\rfloor - 1}{\lfloor\frac{N_k}{2}\rfloor + 1}D_k\right)c_k, \tag{4.6}$$

$\lfloor\ \rfloor$ represents the floor function, N_k, M_k, and K are some positive integers, and c_k and D_k are some real values satisfying the following restrictions:

$$\begin{gathered}\sum_{k=1}^{K} N_k = N, \quad 1 \le K \le N, \\ c_1 < c_2 < \cdots < c_K, \quad 0 \le D_k < 1, \\ M_k = \sum_{m=1}^{k} N_m, \quad M_0 = 0.\end{gathered} \tag{4.7}$$

By comparing (4.7) with (4.4) we see that the function (4.5) has the standard basis eigenvectors (since $\mathbf{Q}$ in this case was taken to be equal to the identity matrix $\mathbf{I}$) and K clusters of eigenvalues with N_k eigenvalues in the kth cluster, respectively. The kth cluster is located around the position c_k with interval width D_k, which is defined in a fractional form in terms of c_k ($0.0 \le D_k < 1.0$). For example, $D_k = 0.5$ implies an interval width of $[0.5c_k, 1.5c_k]$.

This function yields an eigenvalue system of homogeneous dispersion within each cluster, i.e., each cluster consists of equally spaced eigenvalues. The choice $\mathbf{Q} = \mathbf{I}$ determines an orientation of the hyperellipsoid in which the principal axes are aligned parallel to the $\mathbf{x}$ coordinates. The advantage of this choice is that, without loss of generality, the objective function and its gradient vector are computationally very simple even for large N, which permits testing on very large problems at a relatively low cost. The gradient components for this choice are given by

$$G_i = (D_{ii})^2 x_i, \quad i = 1, \ldots, N, \tag{4.8}$$

and the condition number of the Hessian is given by

$$c_n = \left(\frac{D_{NN}}{D_{11}}\right)^2. \tag{4.9}$$

As shown below, we can test the LMQN methods with a variety of setup values for the various parameters $(N; K; N_k, k = 1, \ldots, K; c_k, k = 1, \ldots, K; D_k, k = 1, \ldots, K)$.

Example 1. Let $N = 21; K = 1; N_1 = N; D_1 = A; C_1 = 1.0$. These parameters yield N-dimensional hyperellipsoidal contours. The condition number of this system is $c_n = ((11 + 10A)/(11 - 10A))^2$.

Example 2. Let $N = 21; K = 2, N_1 = 11, N_2 = 10; C_1 = 1.0, C_2 = A; D_1 = 0.2, D_2 = 0.3$. These parameters yield a bicluster problem, the condition number being controlled by $c_n = ((6 + 4D_2)c_2/(6 - 5D_1)c_1)^2$.

4.3. Oceanography problem. This problem is derived from an analysis of the monthly averaged pseudo-wind-stress components over the Indian Ocean. We attempt to analyze the wind over a region Ω by using the following available information: (a) ship-reported averages on a 1° resolution mesh and (b) a 60-yr pseudostress climatology. The objective function is a measure of discrepancy in the data according to certain prescribed conditions, which may be dynamically or statistically motivated. According to climatological observations, the wind pattern should be smooth. Some measure of roughness and some measure of lack of fit to climatology should also be included in the objective function [11].

To formulate the problem, we used the following objective function:

$$\begin{aligned} J(\tau_x, \tau_y) = {} & \frac{1}{L^2} \sum_x \sum_y [(\tau_x - \tau_{x_0})^2 + (\tau_y - \tau_{y_0})^2] \\ & + \frac{\gamma}{L^2} \sum_x \sum_y [(\tau_x - \tau_{x_c})^2 + (\tau_y - \tau_{y_c})^2] \\ & + L^2 \lambda \sum_x \sum_y \left[(\nabla^2(\tau_x - \tau_{x_c}))^2 + (\nabla^2(\tau_y - \tau_{y_c}))^2\right] \\ & + \beta \sum_x \sum_y [\nabla \cdot (\tau_{\mathbf{x}} - \tau_{\mathbf{x}_c})]^2 + \alpha \sum_x \sum_y [\mathbf{k} \cdot \nabla \times (\tau_{\mathbf{x}} - \tau_{\mathbf{x}_c})]^2, \end{aligned} \tag{4.10}$$

where τ_{x_0} and τ_{y_0} are the components of the 1° mean values determined by the ship wind reports; τ_{x_c} and τ_{y_c} are climatology pseudostress vectors, respectively; $\tau_x = u \cdot (u^2 + v^2)^{1/2}$ and $\tau_y = v \cdot (u^2 + v^2)^{1/2}$ are the resultant eastward and northward pseudostress components, respectively; $\mathbf{v}$ represents the wind vector; and L is a length scale (chosen to be 1° latitude), which makes all the terms in the objective cost function dimensionally uniform and scales them to the same order of magnitude. The coefficients (actually weights) γ, λ, β, and α control how closely the direct minimization fits each constraint. The first term in J expresses the closeness to the input data. The second measures the fit to the climatology data values for that month. The third is a smoothing term for data roughness and controls the radius of influence of an anomaly in the input data. The fourth and the fifth terms are boundary-layer kinematic terms that force the results to be comparable to the climatology.

A discretization of the domain Ω of 3665 mesh points produces $2 \times 3665 = 7330$ variables.

4.4. Meteorology problems. Combining in an optimal way new information (in the form of measurements) with a priori knowledge (in the form of a forecast) is a key idea of variational data assimilation in numerical weather prediction. The object is to produce a regular, physically consistent two- or three-dimensional representation of the state of the atmosphere from a series of measurements (called observations) that are heterogeneous in both space and time. This approach is implemented by minimizing a cost function measuring the misfit between the observations and the solution predicted by the model.

Below, a two-dimensional limited-area shallow-water-equations model is used to evaluate a quadratic objective function. The equations may be written as the following (see [20] for details):

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} - fv + \frac{\partial \phi}{\partial x} = 0, \tag{4.11a}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + fu + \frac{\partial \phi}{\partial y} = 0, \tag{4.11b}$$

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} + v\frac{\partial \phi}{\partial y} + \phi\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) = 0, \tag{4.11c}$$

where f is the Coriolis parameter u, v are the two components of the velocity field, and ϕ is the geopotential field; both fields are spatially discretized with a centered-difference scheme in space and an explicit leap-frog integration scheme in time. A rectangular domain of size $L = 6000\,\text{km}$, $D = 4400\,\text{km}$ is used along with discretization parameters $\Delta x = 300\,\text{km}$, $\Delta y = 220\,\text{km}$, and $\Delta t = 600\,\text{s}$.

This model is widely used in meteorology and oceanography since it contains most of the physical degrees of freedom (including gravity waves) present in the more sophisticated three-dimensional primitive-equation models. It is computationally less expensive to implement, and results with this model can be expected to be similar to those obtained from a more complicated primitive-equation model. The gradient of the objective function with respect to the control variables is calculated by the adjoint technique [20].

Two experiments are conducted here. The first involves a model in which only the initial conditions serve as the control variables. The second includes both the initial and boundary conditions as control variables.

The objective function is defined as a simple weighted sum of squared differences between the observations and the corresponding prediction model values:

$$J = W_\phi \sum_{n=1}^{N_\phi} (\phi_n - \phi_n^{\text{obs}})^2 + W_V \sum_{n=1}^{N_V} [(u_n - u_n^{\text{obs}})^2 + (v_n - v_n^{\text{obs}})^2], \tag{4.12}$$

where u, v are the two components of the velocity field, ϕ is the geopotential field, N_ϕ is the total number of geopotential observations available over the assimilation window (t_0, t_R), and N_V is the total number of wind vector observations. The quantities $u_n^{\mathrm{obs}}, v_n^{\mathrm{obs}}$, and ϕ_n^{obs} are the observed values for the northward wind component, the eastward wind component, and the geopotential field, respectively, and the quantities u_n, v_n, and ϕ_n are the corresponding computed model values. W_ϕ and W_V are weighting factors, taken to be the inverse of estimates of the statistical root-mean-square observational errors on geopotential and wind components, respectively. Values of $W_\phi = 10^{-4}\,\mathrm{m}^{-4}\mathrm{s}^4$ and $W_V = 10^{-2}\,\mathrm{m}^{-2}\mathrm{s}^2$ are used. In the first problem the objective function J is viewed as a function of $\mathbf{x}_0 = (u(t_0), v(t_0), \phi(t_0))^T$, whereas in the second J is a function of $(\mathbf{x}_0, \mathbf{v})$, where $\mathbf{v}$ represents a function of time defined on the boundary.

For the experiments the observational data consist of the model-integrated values for wind and geopotential at each time step starting from the Grammeltvedt initial conditions [10] (see Fig. 1). Random perturbations of these fields, performed by using a standard library randomizer RANF on the CRAY-YMP (shown in Fig. 2), are then used as the initial guess for the solution. A grid of 21×21 points in space and 60 time steps in the assimilation window (10 hrs) results in a dimension of the vector of control variables of 1323 for the initial control problem. Controlling the boundary conditions of a limited-area model implies storing in memory as control variables all of the three field variables on the boundary perimeter for all the time steps. The dimension of the vector of control variables thus becomes 14,763.

Two different scaling procedures were considered: gradient and consistent. The first scales the gradient of the objective function. The second makes the shallow-water-equations model nondimensional.

5. Numerical results for LMQN methods. In most of our test problems (those in [13] and synthetic problems) the computational cost of the function is low and the computational effort of the minimization iteration sometimes dominates the cost of evaluating the function and gradient. However, there are also several practical large-scale problems (for example, the variational data assimilation in meteorology) for which the functional computation is expensive. We report, therefore, both the number of function and gradient evaluations and the time required for minimization of some problems.

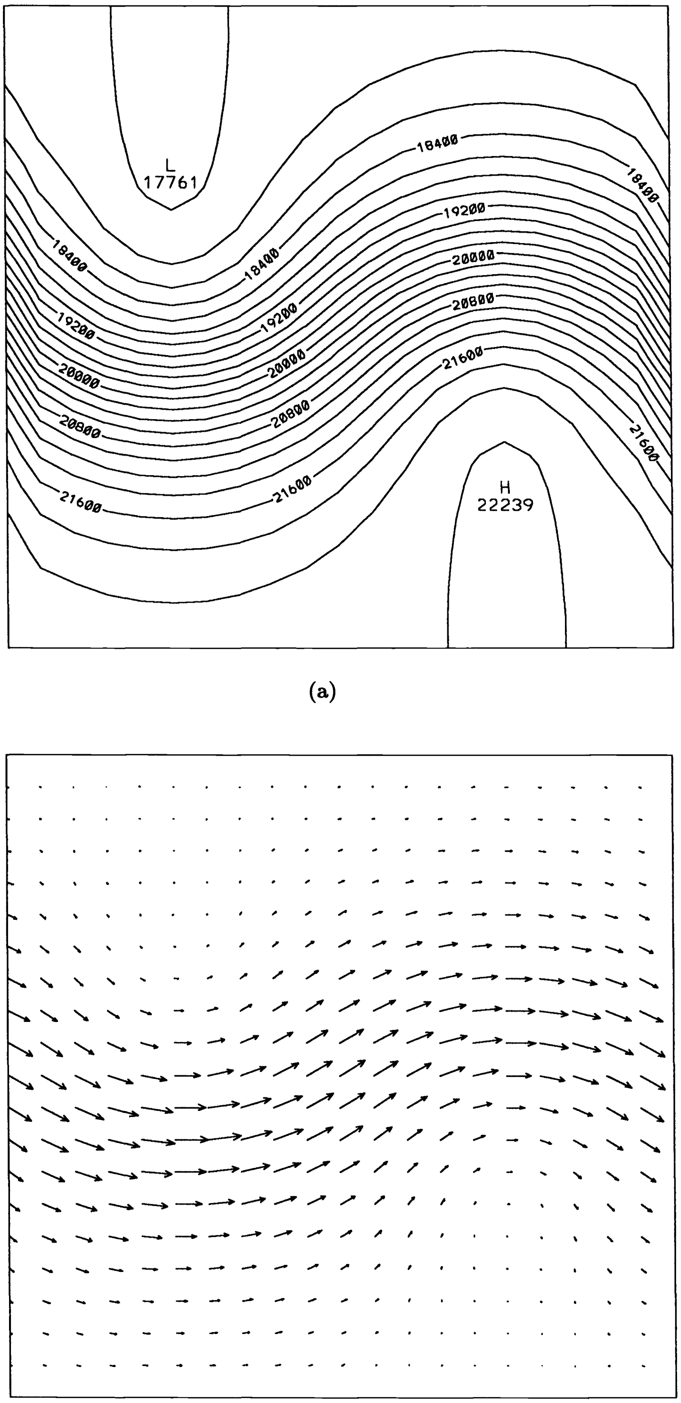
Table 1 shows the amount of storage required by the different LMQN methods for various values of m, the number of Q-N updates, and the dimension N.

The runs below were performed on a CRAY-YMP, for which the unit roundoff is approximately 10^{-14}. In all tables "Iter" represents the total number of iterations, "Nfun" represents the total number of function calls, "MTM" represents the total CPU time spent in minimization, and "FTM" represents the CPU time spent in function and gradient evaluations.

5.1. Results for the standard library test problems. For the 18 test problems, the number of variables ranges from 2 to 100. All the runs reported in this section and §5.2 were terminated when the stopping criterion (2.6) was satisfied. Low accuracy in the solution is adequate in practice.

In the corresponding tables P denotes the problem number, and the results are reported in the form

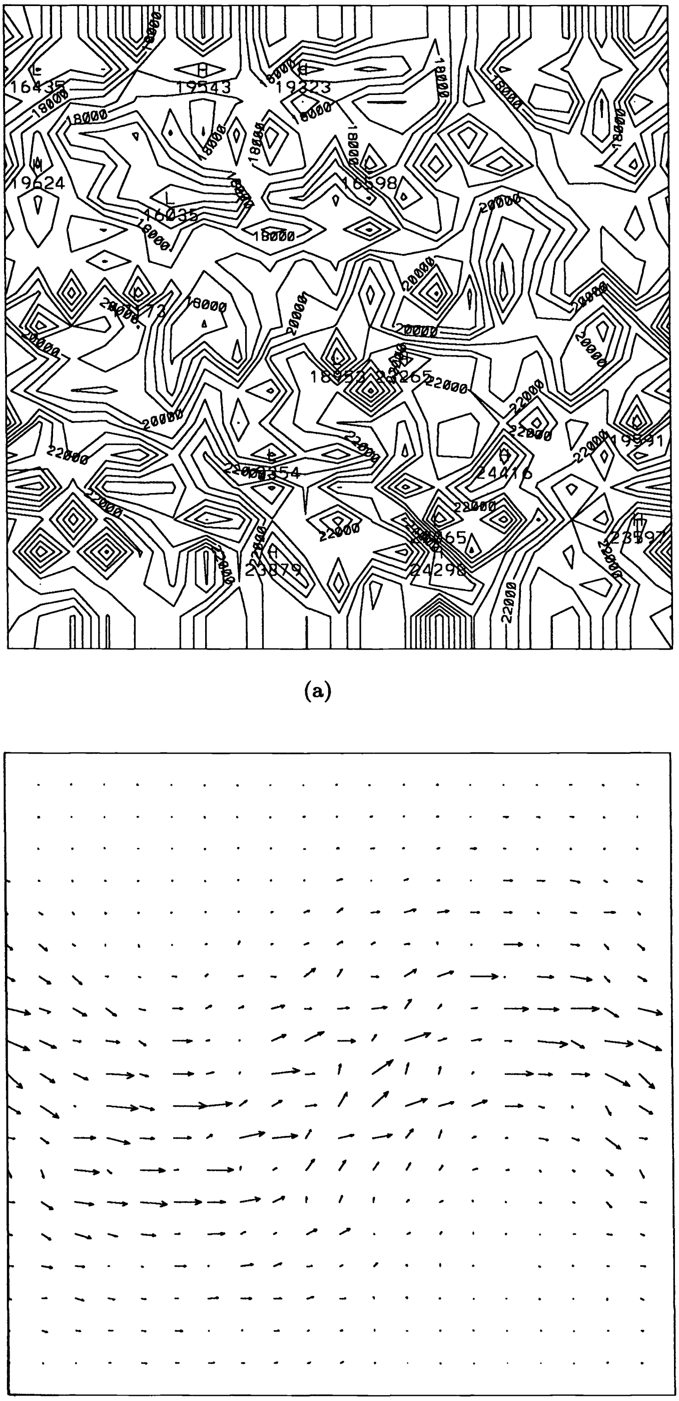
CONMIN-CG/CONMIN-BFGS/E04DGF,
L-BFGS ($m = 3$)/L-BFGS ($m = 5$)/L-BFGS ($m = 7$),
BBVSCG ($m = 3$)/BBVSCG ($m = 5$)/BBVSCG ($m = 7$).



(a)

(b)

FIG. 1 *Geopotential field* (a) *based on the Grammeltvedt initial condition and the wind field* (b), *calculated from the geopotential fields in Fig.* 1(a) *by the geostrophic approximation at the same time levels. Contour interval is* 200 m^2s^{-2} *and the value of maximum vector is* 29.9 ms^{-1}.



(a)

(b)

FIG. 2 *Random perturbation of the geopotential* (a) *and the wind* (b), *fields in Fig.* 1. *Contour interval is* 500 m^2s^{-2} *and the value of maximum vector is* 54.4 ms^{-1}.

TABLE 1

Storage locations (N, dimension of the control variable; m, number of quasi-Newton updates).

CONMIN-CG	CONMIN-BFGS	E04DGF	L-BFGS	BBVSCG
5N+2	N(N+7)/2	14N	(2m+2)N+2m	(2m+3)N+2m

Table 2 compares the performance of the four LMQN methods from the standard starting points, with $m = 3, 5, 7$ updates for both L-BFGS and the Buckley–Lenir method. An "F" indicates failure when the maximum number of function calls (3000) is exceeded. An "S" indicates failure in the line search. The latter may occur from roundoff, and a solution may be obtained nonetheless.

The results show that for some problems in which the objective function depends on no more than three or four variables (such as problems 4, 10, 12, and 16) the full-memory Q-N BFGS method is clearly superior to the LMQN methods. For other problems the LMQN methods display better performance.

For most problems the number of iterations and function calls required decreases as the number of Q-N updates m is increased in L-BFGS. A dramatic case illustrating this is the extended Powell singular function (problem 15). The variation of the value of the objective function and the norm of the gradient with the number of iterations is shown in Fig. 3. For $m = 3$ the number of iterations and function calls required to reach the same convergence criteria is (65, 76); for $m = 5$ it is (56, 66), and for $m = 7$ it is (39, 45). The difference between different values of m becomes obvious only after 18 iterations.

BBVSCG usually uses the fewest function calls when $m = 7$. Either $m = 7$ or $m = 3$ performs best in terms of the number of iterations. Figures 4 and 5 present two illustrative examples. Figure 4 presents the variation of the value of J and the norm of ∇J for the Wood function (problem 17), and Fig. 5 presents the same variation for the variable-dimensioned function (problem 6 ($N = 100$)). The differences between the cases $m = 5$ and $m = 7$ for the two problems are smaller than the corresponding differences between the $m = 3$ and $m = 5$ cases.

Table 2 also shows that L-BFGS usually requires fewer function calls than does BBVSCG. This agrees with the experience of Liu and Nocedal [12], who suggested that BBVSCG gives little or no "speed-up" from additional storage. To investigate this further, we measure in Figs. 6 and 7 the effect of increasing the storage. We define the speed-up by using the same definition as did Liu and Nocedal, i.e., the ratio of function calls required when $m = 3$ and $m = 7$.

We see from these figures that the speed-up of BBVSCG is not smaller than that of L-BFGS. There are cases for which L-BFGS gains more speed-up than does BBVSCG (i.e., problems 2, 4, 5, 7a, 9b, 11, 15, 18). However, there are also cases for which BBVSCG has larger speed-up than does L-BFGS (i.e., problems 7b, 8, 9a, 12, 13, 16, 17). Therefore, the reason that L-BFGS requires fewer function calls cannot be the difference in speed-up between the two codes.

For problems for which the function and gradient evaluations are inexpensive, we also examine the number of iterations and the total time required by the two methods. From Table 2 we see that BBVSCG usually requires fewer iterations and less total CPU time than does L-BFGS. The more accurate line search in BBVSCG may provide an explanation. Will a more accurate line search in L-BFGS decrease the number of iterations? In Table 3 we present the results for L-BFGS ($m = 7$) when the line search is forced to satisfy (2.8) with $\beta = 0.01$ rather than 0.9.

For most problems (18 out of 21) the number of iterations when L-BFGS is used

TABLE 2

Eighteen standard library test problems with standard starting points.

P	N	Iter	Nfun	MTM	FTM
				(total CPU time)	(function calls' CPU)
1	3	22/28/37	49/31/81	0.0223/0.0238/0.0211	0.0008/0.0005/0.0014
		28/27/28	35/31/34	0.0278/0.0316/0.0383	0.0006/0.0005/0.0006
		25/31/26	42/44/39	0.0229/0.0326/0.0315	0.0007/0.0007/0.0007
2	6	24/41/48	50/45/100	0.0260/0.0557/0.0301	0.0026/0.0023/0.0051
		52/42/35	66/46/42	0.0556/0.0523/0.0513	0.0034/0.0024/0.0021
		45/52/38	80/86/55	0.0456/0.0619/0.0516	0.0041/0.0044/0.0029
3	3	3/7/4	7/9/11	0.0028/0.0058/0.0070	0.0001/0.0002/0.0002
		6/7/7	9/9/9	0.0059/0.0073/0.0074	0.0002/0.0002/0.0002
		4/5/4	9/9/8	0.0032/0.0042/0.0031	0.0002/0.0002/0.0001
4	2	99/140/167	257/187/433	0.0912/0.1042/0.0770	0.0033/0.0024/0.0051
		173/167/169	229/208/215	0.1752/0.2045/0.2461	0.0030/0.0027/0.0028
		114/123/136	232/235/231	0.1066/0.1359/0.1723	0.0030/0.0031/0.0030
5	3	11/32/47	31/40/113	0.0109/0.0285/0.0278	0.0010/0.0013/0.0037
		30/29/27	40/40/37	0.0310/0.0361/0.0383	0.0013/0.0013/0.0012
		19/22/22	36/35/36	0.0175/0.0230/0.0279	0.0012/0.0011/0.0012
6	10	6/19/19	13/20/41	0.0052/0.0373/0.0148	0.0002/0.0003/0.0006
		19/19/19	20/20/20	0.0187/0.0224/0.0256	0.0003/0.0003/0.0003
		14/13/17	26/22/27	0.0129/0.0142/0.0218	0.0004/0.0003/0.0004
	100	10/36/35	21/37/73	0.0224/1.7774/0.0464	0.0004/0.0007/0.0014
		36/36/36	37/37/37	0.0733/0.0936/0.1125	0.0007/0.0007/0.0007
		15/26/29	37/49/45	0.0288/0.0586/0.0707	0.0007/0.0010/0.0009
7	12	215/97/F	432/100/F	0.4707/0.2870/2.1152	0.2152/0.0498/1.4839
		2202/396/222	2511/458/255	3.5291/0.7419/0.4794	1.2506/0.2282/0.1276
		239/234/201	433/373/289	0.4597/0.4616/0.4111	0.2158/0.1859/0.1440
	30	600/135/F	1208/140/F	2.1548/1.1544/3.8358	1.2577/0.1457/3.1262
		F/2049/106	F/2400/1240	6.2462/5.7458/3.2839	3.1237/2.4986/1.2903
		572/470/223	1108/800/355	1.8518/1.4923/0.7219	1.1542/0.8334/0.3697
8	100	24/71/F	59/80/F	0.0604/3.5806/0.7649	0.0010/0.0014/0.0512
		60/59/56	73/69/67	0.1266/0.1590/0.1824	0.0012/0.0012/0.0011
		67/48/49	132/87/80	0.1276/0.1066/0.1271	0.0023/0.0015/0.0014
9	10	125/17/332	306/26/863	0.1437/0.0355/0.1997	0.0173/0.0015/0.0484
		17/21/17	19/23/19	0.0182/0.0261/0.0236	0.0011/0.0013/0.0011
		18/17/18	32/25/24	0.0186/0.0190/0.0228	0.0018/0.0014/0.0014
	50	127/F/141	281/F/309	0.2719/9.3038/0.1596	0.0650/0.6949/0.0712
		250/250/223	331/308/272	0.4425/0.5301/0.5577	0.0766/0.0712/0.0628
		136/146/148	254/256/215	0.2381/0.2883/0.3168	0.0589/0.0593/0.0498
10	2	8/10/11	18/15/28	0.0063/0.0073/0.0097	0.0001/0.0001/0.0002
		13/13/13	25/25/25	0.0142/0.0162/0.0177	0.0002/0.0002/0.0002
		11/11/16	26/22/30	0.0098/0.0108/0.0191	0.0002/0.0002/0.0002
11	4	26/35/36	F/36/79	0.4817/0.0356/0.0231	0.1299/0.0016/0.0034
		38/22/27	63/43/37	0.0432/0.0305/0.0387	0.0027/0.0019/0.0016
		25/24/23	S/48/52	0.0344/0.0262/0.0279	0.0051/0.0016/0.0014
12	3	15/16/29	36/21/71	0.0207/0.0183/0.0330	0.0081/0.0048/0.0161
		26/24/24	35/32/33	0.0337/0.0355/0.0402	0.0079/0.0072/0.0075
		13/13/15	30/27/24	0.0187/0.0199/0.0222	0.0068/0.0061/0.0054
13	100	46/42/53	98/52/122	0.1457/2.1017/0.0749	0.0142/0.0075/0.0176
		49/48/47	56/54/53	0.1094/0.1345/0.1574	0.0081/0.0078/0.0077
		43/54/51	78/91/48	0.0968/0.1431/0.1422	0.0113/0.0132/0.0107
14	100	18/36/24	49/50/69	0.04571.7946/0.0378	0.0006/0.0006/0.0009
		33/35/38	45/46/50	0.0706/0.0941/0.1227	0.0005/0.0006/0.0006
		31/30/34	56/48/52	0.0576/0.0645/0.0854	0.0007/0.0006/0.0006
15	100	47/42/46	95/43/108	0.1247/2.0963/0.0536	0.0014/0.0006/0.0016
		65/56/39	76/66/45	0.1367/0.1512/0.1243	0.0011/0.0010/0.0007
		44/49/58	86/79/82	0.0854/0.1087/0.1494	0.0013/0.0012/0.0012
16	2	10/15/16	22/16/35	0.0078/0.0103/0.0114	0.0002/0.0001/0.0003
		16/14/14	18/15/15	0.0150/0.0151/0.0169	0.0001/0.0001/0.0001
		14/16/15	30/25/19	0.0130/0.0161/0.0163	0.0002/0.0002/0.0002
17	4	48/36/200	106/43/418	0.0497/0.0362/0.0871	0.0009/0.0004/0.0035
		106/93/86	137/119/113	0.1073/0.1152/0.1260	0.0011/0.00100.0009
		30/24/22	53/42/34	0.0272/0.0253/0.0262	0.0004/0.0003/0.0003
18	100	686/465/832	1384/491/1724	8.8249/26.57/9.0296	6.7316/2.39398.4209
		1137/853/781	1213/895/821	8.2528/6.6732/6.6005	5.8978/4.3587/3.9912
		709/591/698	1393/1110/1318	8.5565/7.2290/9.2365	6.7916/5.4006/6.5045

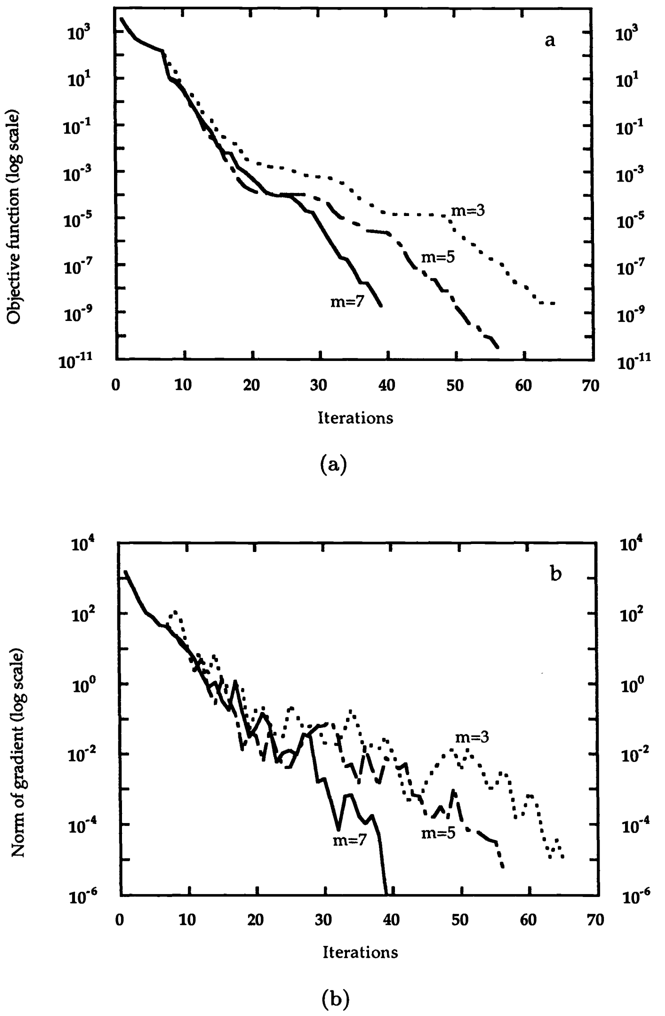
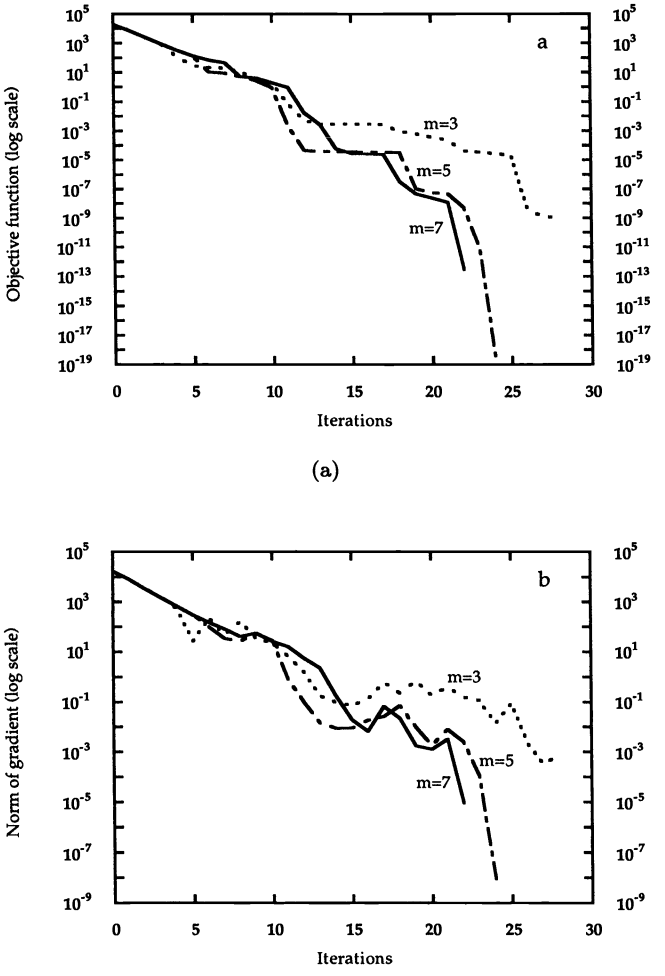


FIG. 3. *Variation of* (a) *the objective function and* (b) *the norm of gradient with the number of iterations using the* L-BFGS *method with m equal* 7 (*solid*), 5 (*dash dot*), *and* 3 (*dotted*) *for the test library problem* 15.

is then markedly reduced (compare Table 2 L-BFGS ($m = 7$) with Table 3). Among those problems, about two-thirds require more function calls, but about one-third require even fewer function calls.

This implementation of L-BFGS is compared with the CONMIN-CG, E04DGF, L-BFGS ($\beta = 0.9$), and BBVSCG codes in Table 4. The "number of wins" describes the number of runs for which a method required fewest function calls and the number of runs for which a method required fewest iterations. Because ties occur, numbers across a row do not add up to the number of different test cases.

We see that L-BFGS ($m = 7$ and $\beta = 0.01$) uses the fewest iterations and that L-BFGS ($m = 7$ and $\beta = 0.9$), CONMIN-CG, and BBVSCG use the fewest function calls. If both the numbers of iterations and function calls are considered, CONMIN-CG seems to be the best.



(a)

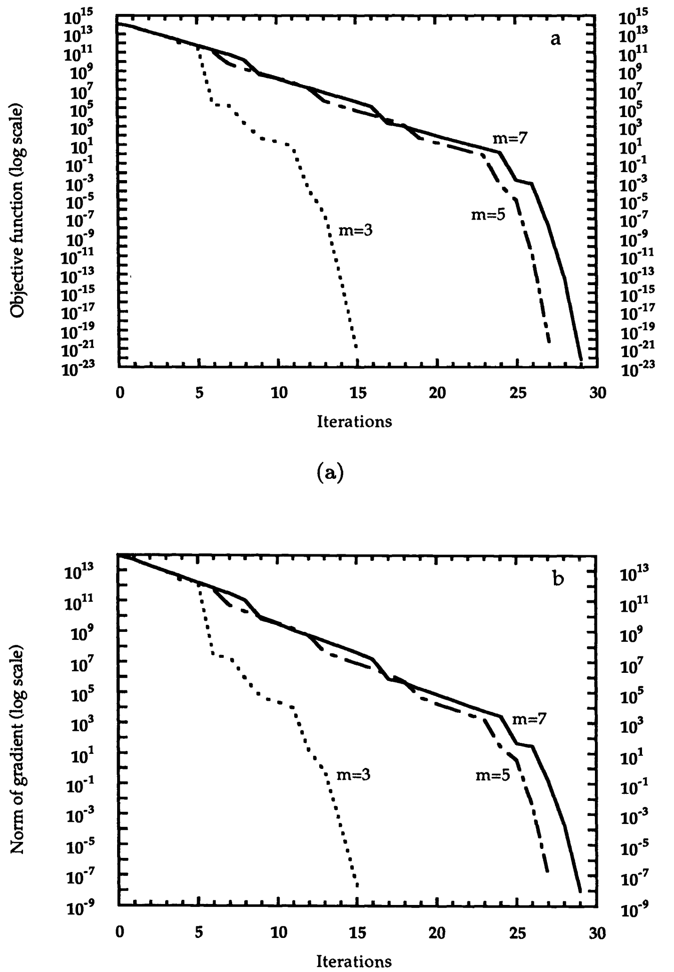
(b)

FIG. 4. *Variation of* (a) *the objective function and* (b) *the norm of gradient with the number of iterations using the* BBVSCG *method with m equal* 7 (*solid*), 5 (*dash dot*), *and* 3 (*dotted*) *for the test library problem* 17.

We also find that L-BFGS still requires the fewest function calls among LMQN methods that use nonstandard starting points (data not shown).

Therefore, from the experiments with the 18 library test problems, L-BFGS with a more accurate line search ($\beta = 0.01$) emerges as the most efficient minimizer for problems for which the function calls are inexpensive and the computational effort of the iteration dominates the cost of evaluating the function and gradient. However, both L-BFGS with inexact line searches and CONMIN are very effective on problems for which the function calls are exceedingly expensive. E04DGF does not perform as well as the other LMQN methods.

5.2. Results for the synthetic cluster function problems. For the first one-cluster hyperellipsoidal problem we tested the sensitivity of all the methods to



(a)

(b)

FIG. 5. *Variation of* (a) *the objective function and* (b) *the norm of gradient with the number of iterations using the* BBVSCG *method with m equal* 7 (*solid*), 5 (*dash dot*), *and* 3 (*dotted*) *for the test library problem* 6 *with dimension* 100.

various degrees of ill conditioning by controlling the value of D_1, the dispersion interval in fractional form. Table 5 presents the results for D_1 taken to be 0.2, 0.8, and 0.99, respectively. The corresponding condition numbers are 2.0, 39.9, and 436.8. The results in Table 5 indicate that L-BFGS performs best when the condition number is small. As the condition number is increased, L-BFGS requires the most iterations and function calls, whereas CONMIN-CG uses the fewest function calls. In CPU time E04DGF is most efficient and CONMIN-BFGS is most expensive (even though the latter requires fewer iterations and function calls than does L-BFGS). The full-memory CONMIN-BFGS code spends about four times as much CPU time as does any other method. This occurs because most iteration time is spent in matrix and vector multiplications.

For the second bicluster problem we control the condition number by changing

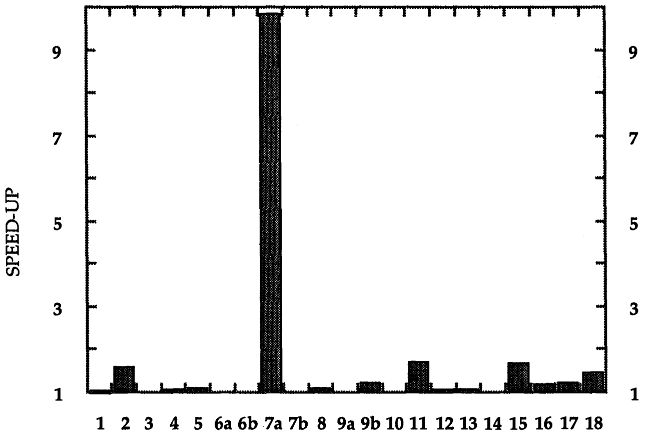


FIG. 6. *Speed-up* NFUN(3)/NFUN(7), *for* L-BFGS *method.*

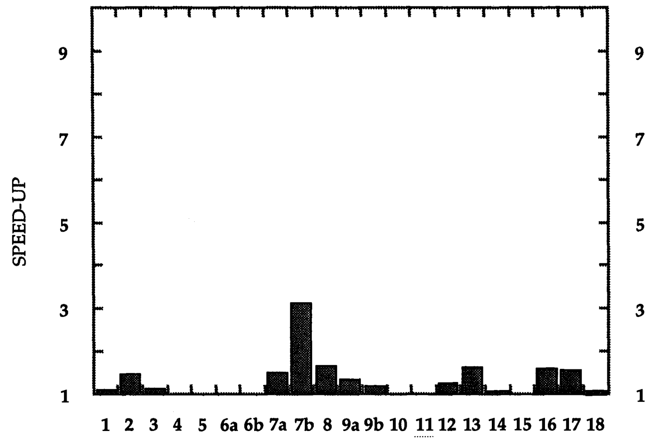


FIG. 7. *Speed-up* NFUN(3)/NFUN(7), *for* BBVSCG *method.*

the position of the center of the second cluster C_2. The performance when the value of the condition number is equal to $8.29, 8.29 \times 10^2$, and 8.29×10^4, respectively, is given in Table 6. We see that when the condition number is equal to 8.29, L-BFGS uses the fewest function calls. However, the differences among the various methods is not significant. When the condition number is increased, L-BFGS again turns out to be the worst. The E04DGF code turns out to be best in all computational respects: number of iterations, number of function calls, and total CPU time. If we use a more accurate line search, L-BFGS is competitive with CONMIN-CG, which is the second best, and is better than BBVSCG.

We also compared the performance of different LMQN methods on a multicluster problem. The same conclusion can be drawn (table omitted): the E04DGF performs best. L-BFGS with a more accurate line search and CONMIN-CG come in second, followed by BBVSCG.

TABLE 3

Eighteen standard library test problems with standard starting points, using the L-BFGS ($m = 7$) *method with more accurate line search.*

P	N	Iter	Nfun	MTM (total CPU time)	FTM (Function calls' CPU time)
1	3	19	55	0.0325	0.0009
2	6	19	57	0.0352	0.0029
3	3	3	9	0.0038	0.0002
4	2	98	355	0.1938	0.0047
5	3	11	42	0.0199	0.0014
	10	4	14	0.0057	0.0002
6	100	7	27	0.0220	0.0005
	12	36	97	0.1130	0.0479
7	30	254	654	1.2429	0.6812
8	100	58	223	0.2491	0.0039
	10	61	226	0.1371	0.0127
9	50	155	444	0.5163	0.1027
10	2	8	30	0.0130	0.0002
11	4	13	35	0.0219	0.0012
12	3	14	47	0.0348	0.0107
13	100	41	107	0.1656	0.0134
14	100	23	77	0.0880	0.0010
15	100	19	56	0.0685	0.0008
16	2	9	28	0.0139	0.0002
17	4	29	83	0.0507	0.0007
18	100	704	1451	9.8829	7.2431

TABLE 4

Number of wins on the whole set of 18 *test problems with the standard starting points, using limited memory* Q-N *methods.*

WINS	CONMN -CG	E04DGF	L-BFGS (β=0.9)			L-BFGS (β=0.01)	BBVSCG		
			m=3	m=5	m=7	m=7	m=3	m=5	m=7
Iter	6	0	1	0	2	13	1	1	1
Nfun	5	0	2	3	5	2	0	0	5

TABLE 5

One cluster problem ($N = 21$; $K = 1$; $N_1 = 21$; $C_1 = 1.0$; $D_1 = 0.2, 0.8,$ *and* 0.99), *the condition numbers are* 2.0, 39.9, *and* 436.8, *respectively.*

	m	Iter			Nfun			MTM (total CPU time)		
D_1		0.2	0.8	0.99	0.2	0.8	0.99	0.2	0.8	0.99
CONMIN-CG		10	21	21	21	43	43	0.0157	0.0342	0.0342
Conmin-BFGS		11	45	56	13	47	58	0.0493	0.2187	0.2679
E04DGF		10	21	32	23	45	67	0.0060	0.0119	0.0179
L-BFGS (β=0.9)	3	10	50	117	12	56	124	0.0129	0.0654	0.1515
	5	10	45	97	12	52	103	0.0146	0.0716	0.1535
	7	10	42	99	12	48	107	0.0157	0.0773	0.1855
BBVSCG	3	10	24	44	19	47	87	0.0152	0.0403	0.0747
	5	10	26	26	17	50	49	0.0165	0.0528	0.0527
	7	10	42	61	15	62	101	0.0170	0.0790	0.1269
L-BFGS' ($\beta=10^{-2}$)	3	10	21	45	23	45	46	0.0168	0.0343	0.0347
	5	10	21	45	23	45	46	0.0185	0.0393	0.0397
	7	10	21	45	23	45	46	0.0196	0.0436	0.0440

TABLE 6

Bi-cluster problem ($N = 21$; $K = 2$; $N_1 = 11$; $N_2 = 10$; $C_1 = 1.0$; $C_2 = 2.0, 20$, *and* 200; $D_1 = 0.2, 0.3$), *the condition numbers are* 8.29, 8.29×10^2, *and* 8.29×10^4, *respectively.*

	m	Iter			Nfun			MTM (total CPU time)		
D_1		2.0	20.0	200.	2.0	20.0	200.	2.0	20.0	200.0
CONMIN-CG		18	23	35	37	47	71	0.0318	0.0412	0.0634
Conmin-BFGS		28	90	141	30	92	143	0.1334	0.4516	0.6958
E04DGF		15	19	22	33	42	47	0.0115	0.0148	0.0167
L-BFGS (β=0.9)	3	24	158	1009	28	170	1083	0.0338	0.2205	1.4129
	5	24	154	895	29	164	951	0.0392	0.2591	1.5187
	7	24	168	579	28	182	610	0.0439	0.3330	1.1523
BBVSCG	3	20	67	168	39	129	289	0.0355	0.1198	0.2695
	5	24	75	167	38	128	256	0.0418	0.1406	0.2895
	7	24	81	169	31	126	232	0.0398	0.1636	0.3232
L-BFGS' (β=10^{-2})	3	18	21	29	39	53	81	0.0335	0.0419	0.0616
	5	18	20	29	39	49	83	0.0376	0.0439	0.0699
	7	18	21	30	39	51	88	0.0410	0.0504	0.0824

TABLE 7

Ocean problem ($N = 7330$), *using limited-memory* Q-N *methods.*

Algorithm	Iter	Nfun	MTM (total CPU time)	FTM (function calls' CPU time)
CONMIN-CG	7	15	15.45	15.42
L-BFGS(β=0.9)	22	25	17.15	16.44
BBVSCG	4	8	8.43	9.23
L-BFGS(β=0.001)	22	25	17.03	16.33

5.3. Results for the oceanographic large-scale minimization problem. Only CONMIN-CG, L-BFGS, and BBVSCG were successful for this large-scale problem. E04DGF failed in its line search. All the methods used the same convergence criterion:

$$\|g_k\| \leq \epsilon, \quad \epsilon = 10^{-8}. \tag{5.1}$$

Numerical experiments indicate that when the number of Q-N updates m is increased from 3 to 7, there is no significant improvement in performance. In Table 7 we present only the results for L-BFGS and BBVSCG when $m = 3$. We see that the function evaluation for this problem is far more expensive than is the iterative procedure. Both L-BFGS and CONMIN-CG require 15 function calls, whereas BBVSCG uses only 8 function calls. Therefore, BBVSCG emerges as the most effective algorithm here. It also uses fewest iterations. No significant improvement was observed for L-BFGS with a more accurate line search.

5.4. Results for the meteorological large-scale minimization problem. Both gradient scaling and nondimensional scaling were applied to the meteorological large-scale minimization problem for all four LMQN methods. CONMIN-CG and BBVSCG failed after the first iteration with either gradient scaling or nondimensional scaling. L-BFGS was successful only with gradient scaling. E04DGF worked only with the nondimensional shallow-water-equations model. It appears that additional scaling is crucial for the success of the LMQN minimization algorithms applied to this real-life, large-scale meteorological problem.

TABLE 8

Meteorological problem with the limited memory quasi-Newton methods.

Control Variables	Algorithm	Iter	Nfun	MTM (total CPUtime)	FTM (function calls' CPU time)
Initial	E04DGF	72	203	36.89	33.56
	L-BFGS	66	89	15.53	14.76
Initial+Boundary	E04DGF	160	481	87.31	79.98
	L-BFGS	179	468	80.70	77.81

TABLE 9

Maximum absolute differences between the retrieval and the unperturbed initial wind and geopotential fields using the limited memory quasi-Newton methods.

Control Variables	Algorithm	$(u^2+v^2)^{1/2}$	ϕ
Initial	E04DGF	0.75E-2	0.12E2
	L-BFGS	0.38E-1	0.90E0
Initial+Boundary	E04DGF	0.10E0	0.64E3
	L-BFGS	0.26E1	0.22E2

Table 8 presents the performance of these two LMQN methods, namely, E04DGF and L-BFGS, when only the initial conditions or the initial-plus-boundary conditions are taken to be the control variables. Because of the different scaling procedures used in the two methods the minimization was stopped when the convergence criterion

$$\|g_k\| \leq 10^{-4} \times \|g_0\| \tag{5.2}$$

was satisfied.

We observe from Table 8 that most of the CPU time is spent on function calls rather than in the minimization iteration. By comparing the number of function calls and CPU time we find that the computational cost of L-BFGS is much lower than that of E04DGF. L-BFGS converged in 66 iterations with 89 function calls. In contrast, E04DGF required 72 iterations and 203 function calls to reach the same convergence criterion. This produces rather large differences in the CPU time spent in minimization. L-BFGS uses less than half of the total CPU time required for E04DGF.

The differences between figures showing the retrieved initial wind and geopotential and Fig. 1 are imperceptible (figures omitted). Table 9 gives the maximum differences between the retrieval and the unperturbed initial conditions from E04DGF and L-BFGS minimization results. An accuracy of at least 10^{-3} is reached for both the wind and geopotential fields by using both the codes of L-BFGS and of E04DGF for the initial control. This clearly shows the capability of the unconstrained LMQN methods to adjust a numerical weather prediction model to a set of observations distributed in both time and space.

When we control both the boundary and initial conditions, we expect to produce a much more difficult problem than when we control only the initial conditions. First, since the dimensionality of the Hessian of the objective function is increased by about one order of magnitude (from 10^3 to 10^4), the condition number of the Hessian will increase as $O(N^{2/d})$ [27], where d is the dimensionality of the space variables and N is

TABLE 10

Initial control problem in meteorology.

algorithm	MXITCG	Iter	Nfun	NCG	MTM	FTM
TN1	3	19	20	50	12.20	11.31
	50	20	26	54	13.79	12.89
TN1 (no prec.)	3	63	64	170	38.82	37.15
	50	39	40	165	32.78	31.53
TN2	3	81	82	242	68.68	67.21
	50	4	5	91	16.41	16.30

the number of components of the vector of control variables. Second, the perturbation of the boundary conditions creates locally an ill-posed problem. This is reflected by an increase of high-frequency noise near the boundary. In turn, the condition number of the Hessian of the objective function increases.

From Tables 8 and 9 we see, indeed, that when we control both the initial and boundary conditions, minimization becomes much more difficult. The computational cost is doubled and the accuracy of the retrieval is decreased by an order of magnitude compared with those of the initial control problem. The largest differences occur near the boundary for both the wind and geopotential fields. However, the differences between the performances of E04DGF and L-BFGS on the initial- and boundary-value problems are small.

6. Results for TN methods. The meteorology problems of §5.4 were tested for TN1 and TN2. In TN methods performance often depends on the specified maximum number of permitted inner iterations per outer iteration (MXITCG). Our experience suggests that different settings for MXITCG have a small impact on the performance of TN1 but a rather large impact on that of TN2 (see Table 10). This results from our current unpreconditioned implementation for TN2 since the inner CG loop requires more iterations to find a search direction.

To clarify this idea and to see what differences in performance between the two TN methods were due to the different truncation criteria, CG versus Lanczos, and to preconditioning, we also performed minimization for TN1 without diagonal preconditioning. The results are presented in Table 10. Similar trends are identified for both TN1 and TN2 in this case: the cost for large MXITCG is much lower than that for small MXITCG. However, TN2 with MXITCG = 50 performs much better than does TN1 with MXITCG = 50 in terms of Newton iterations, CG iterations, function evaluations, and CPU time. This strongly suggests that with a suitable preconditioner for the problem in meteorology, TN2 might perform best.

Numerical results for both initial control and initial and boundary control are summarized in Tables 11 and 12. We see from the tables that time is approximately proportional to the number of inner iterations. Thus the use of preconditioning in TN1 accelerates performance, as expected. Note that without preconditioning TN1 requires more function evaluations than does TN2. Preconditioning is particularly important as the dimension of the minimization problem increases.

Comparison with Table 9 shows that the TN methods are competitive with L-BFGS. TN1 is better than L-BFGS for initial control and much better than L-BFGS for initial and boundary control. TN1 also produces higher accuracy than do the other three methods (see Tables 9 and 12).

TABLE 11

Meteorological problem with the truncated Newton methods.

Control Variables	Algorithm	Iter	Nfun	MTM (total CPU time)	FTM (function calls' CPU time)
Initial	TN1	19	70	12.20	11.31
	TN2	4	96	16.41	16.30
Initial+Boundary	TN1	70	283	49.96	46.22
	TN2	12	520	87.22	86.30

TABLE 12

Maximum absolute differences between the retrieval and the unperturbed initial wind and geopotential fields using the truncated Newton methods.

Control Variables	Algorithm	$(u^2+v^2)^{1/2}$	ϕ
Initial	TN1	0.89E-2	0.54E2
	TN2	0.58E-2	0.41E2
Initial+Boundary	TN1	0.96E1	0.48E3
	TN2	0.14E0	0.90E3

It appears that for this set of test problems TN methods always require far fewer iterations and fewer function calls than do the LMQN methods. The good performance of the TN methods for large-scale minimization of variational data assimilation problems is very encouraging since minimization is the most computationally intensive part of the assimilation procedure and the numerical weather prediction model already taxes the capability of present-day computers. The complexity of these problem stems from the cost of the integration of the model and of the adjoint system required to update the gradient in the minimization procedure.

7. Summary and conclusions. Four recently available LMQN methods and two TN methods were examined for a variety of test and real-life problems. All methods have practical appeal: they are simple to implement, they can be formulated to require only function and gradient (and possibly additional preconditioning) information, and they can be faster than full-memory Q-N methods for large-scale problems. L-BFGS emerged as the most robust code among the LMQN methods tested. It uses the fewest iterations and function calls for most of the 18 standard test library problems, and it can be greatly improved by a simple scaling or by a more accurate line search. All of the LMQN methods (L-BFGS, CONMIN-CG, and BBVSCG) perform better than the full-memory Q-N BFGS method, especially in terms of total CPU time. E04DGF appears to be the least efficient method for the library test problems. However, numerical results obtained for the synthetic cluster function reveal that E04DGF performs quite well on problems whose Hessian matrices have clustered eigenvalues.

Both variable-storage methods (L-BFGS and BBVSCG) were very successful on the large-scale problem from oceanography, and BBVSCG turned out to perform slightly better on this problem than did L-BFGS.

The convergence rate of the variable-storage methods was accelerated when the number m of Q-N updates was increased for medium-sized problems. However, for small- and large-scale problems both methods showed only a slight improvement as the number of Q-N updates m is increased. The reason for this is not yet known, and further research is needed. Implementation of these minimization algorithms on vector and parallel computer architectures is expected to yield a significant reduction in the computational cost of large minimization problems.

Only E04DGF and L-BFGS performed successfully on the large-scale optimal control problems in meteorology, and they were successful only after special scalings were applied. L-BFGS performed better than E04DGF in terms of computational cost.

Although the L-BFGS method may be adequate for most present-day large-scale minimization, TN methods yield the best results for large-scale meteorological problems.

Acknowledgments. The authors thank Maria Hopgood for help in testing the 18 problems. The authors also thank Jorge Nocedal and an anonymous reviewer for their insightful remarks, which contributed to a clear and more concise paper.

REFERENCES

[1] A. G. Buckley, *A combined conjugate-gradient quasi-Newton minimization algorithm*, Math. Programming, 15 (1978), pp. 200–210.

[2] ———, *Remark on algorithm* 630, ACM Trans. Math. Software, 15 (1989), pp. 262–274.

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

[4] ———, *Algorithm* 630-BBVSCG: *A variable storage algorithm for function minimization*, ACM Trans. Math. Software, 11 (1985), pp. 103–119.

[5] W. C. Davidon, *Variable Metric Methods for Minimization*, A.E.C. Research and Development Report ANL-5990, Argonne National Laboratory, Argonne, IL, 1959.

[6] R. S. Dembo and T. Steihaug, *Truncated-Newton algorithms for large-scale unconstrained optimization*, Math. Programming, 26 (1983), pp. 190–212.

[7] J. C. Gilbert and C. Lemaréchal, *Some numerical experiments with variable-storage quasi-Newton algorithms*, Math. Programming, 45 (1989), pp. 407–435.

[8] P. E. Gill and W. Murray, *Conjugate-Gradient Methods for Large-Scale Nonlinear Optimization*, Tech. Report SOL 79-15, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA, 1979.

[9] H. G. Golub and C. van Loan, *Matrix Computation*, 2nd ed., Johns Hopkins Series in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 1989.

[10] A. Grammeltvedt, *A survey of finite-difference schemes for the primitive equations for a barotropic fluid*, Mon. Weather Rev., 97 (1969), pp. 387–404.

[11] D. M. Legler, I. M. Navon, and J. J. O'Brien, *Objective analysis of pseudo-stress over the Indian Ocean using a direct minimization approach*, Mon. Weather Rev., 117 (1989), pp. 709–720.

[12] D. C. Liu and J. Nocedal, *On the limited memory* BFGS *method for large scale optimization*, Math. Programming, 45 (1989), pp. 503–528.

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

[14] NAG, *Fortran Library Reference Manual Mark* 14, No. 3, Numerical Algorithms Group, Downers Grove, IL, 1990.

[15] S. G. Nash, *Newton-type minimization via the Lanczos method*, SIAM J. Numer. Anal., 21 (1984), pp. 770–788.

[16] ———, *Solving nonlinear programming problems using truncated-Newton techniques*, in Numerical Optimization, Proc. SIAM Conference on Numerical Optimization, Society for Industrial and Applied Mathematics, Philadelphia, 1984, pp. 119–136.

[17] ———, *Preconditioning of truncated-Newton methods*, SIAM J. Sci. Statist. Comput, 6 (1985), pp. 599–616.

[18] S. G. Nash and J. Nocedal, *A Numerical Study of the Limited Memory* BFGS *Method and the Truncated-Newton Method for Large Scale Optimization*, Tech. Report NAM 02, Northwestern University, Evanston, IL, 1989.

[19] I. M. NAVON AND D. M. LEGLER, *Conjugate-gradient methods for large-scale minimization in meteorology*, Mon. Weather Rev., 115 (1987), pp. 1479–1502.

[20] I. M. NAVON AND X. ZOU, *Application of the adjoint model in meteorology*, in Automatic Differentiation of Algorithms: Theory, Implementation and Application, A. Griewanek and G. Corliss, eds., Society for Industrial and Applied Mathematics, Philadelphia, 1991, pp. 202–207.

[21] 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; also Tech. Memo ANL-AMD 282, Applied Mathematics Division, Argonne National Laboratory, Argonne, IL, 1976.

[22] ———, *Conjugate gradient methods less dependent on conjugacy*, SIAM Rev., 28 (1986), pp. 501–511.

[23] J. NOCEDAL, *Updating quasi-Newton matrices with limited storage*, Math. Comp., 35 (1980), pp. 773–782.

[24] ———, *Theory of algorithms for unconstrained optimization*, Acta Numerica, 1991, pp. 199–242, p. 340.

[25] A. PERRY, *A Modified Conjugate-Gradient Algorithm*, Discussion Paper 229, Center for Mathematical Studies in Economics and Management Sciences, Northwestern University, Evanston, IL, 1976.

[26] ———, *A Class of Conjugate-Gradient Algorithms with a Two-Step Variable Metric Memory*, Discussion Paper 269, Center for Mathematical Studies in Economics and Management Sciences, Northwestern University, Evanston, IL, 1977.

[27] M. J. D. POWELL, *Restart procedures for the conjugate gradient method*, Math. Programming, 12 (1977), pp. 241–254.

[28] T. SCHLICK, *Modified Cholesky factorizations for sparse preconditioners*, SIAM J. Sci. Comput., 14 (1993), pp. 424–445

[29] T. SCHLICK AND A. FOGELSON, TNPACK—*A truncated Newton minimization package for large-scale problems*, I. *Algorithm and usage*, II. *Implementation examples*, ACM Trans. Math. Software, 18 (1992), pp. 46–111.

[30] T. SCHLICK AND M. L. OVERTON, *A powerful truncated Newton method for potential energy minimization*, J. Comput. Chem., 8 (1987), pp. 1025–1039.

[31] D. F. SHANNO, *On the convergence of a new conjugate gradient algorithm*, SIAM J. Numer. Anal., 15 (1978), pp. 1247–1257.

[32] ———, *Conjugate gradient methods with inexact searches*, Methods Oper. Res., 3 (1978), pp. 244–256.

[33] D. F. SHANNO AND K. H. PHUA, *Remark on algorithm* 500—*a variable method subroutine for unconstrained nonlinear minimization*, ACM Trans. Math. Software, 6 (1980), pp. 618–622.

[34] P. WOLFE, *The secant method for simultaneous nonlinear equations*, Comm. ACM, 2 (1968), pp. 12–13.

SIAM J. OPTIMIZATION
Vol. 3, No. 3, pp. 609–629, August 1993

010

A GLOBALLY CONVERGENT METHOD FOR l_p PROBLEMS*

YUYING LI†

Abstract. The l_p-norm discrete estimation problem $\min_{x\in\Re^n} \|b - A^Tx\|_p^p$ is troublesome when p is close to unity because the objective function approaches a nonsmooth form as p converges to one. This paper presents an efficient algorithm for solving l_p-norm problems for all $1 \leq p < 2$. When $p = 1$ it is essentially the method presented by T. F. Coleman and Y. Li [*Math. Programming*, 56 (1992), pp. 189–222], which is a globally and quadratically convergent algorithm under some nondegeneracy assumptions. The existing iteratively reweighted least-squares (IRLS) method can be obtained from the new approach by updating some dual multipliers in a special fashion. The new method is globally convergent, and it is superlinearly convergent when there is no zero residual at the solution. At each iteration the main computational cost of the new method is the same as that of the IRLS method: solving a reweighted least-squares problem. Numerical experiments indicate that this method is significantly faster than popular iteratively reweighted least-squares methods when p is close or equal to one.

Key words. discrete estimation, data analysis, IRLS method, linear programming, interior-point algorithm, simplex method, Newton method

AMS subject classifications. 65H10, 65K05, 65K10

1. Introduction.

In discrete estimation and data analysis it is often appropriate to solve the following problem:

$$\min_{x\in\Re^n} \|A^Tx - b\|_p^p, \tag{1.1}$$

where $A = [a_1, \ldots, a_m] \in \Re^{n\times m}$, $b \in \Re^m$, and $m > n$. We denote the objective function $\|A^Tx - b\|_p^p = \sum_{i=1}^m |a_i^Tx - b_i|^p$ by $\psi(x)$. In this paper we focus on the case when $1 \leq p < 2$. We assume that A has rank n. When $1 < p < \infty$ this assumption is equivalent to $\psi(x)$ being strictly convex. We also assume that there does not exist any x such that $A^Tx - b = 0$.

The most often used measures for (1.1) are 2-norm, 1-norm, and ∞-norm. The l_∞ solution offers a worst-case guarantee. The 2-norm solution is popular because of its special relationship with the normal distributions. The increasingly important l_1 solution is useful since it is insensitive to a small number of large residuals (*resistant*). Thus one can imagine situations when minimizing the l_p-norm, where $1 < p < 2$, is appropriate [13], [14]. Moreover, the problem is theoretically interesting since it ranges from a piecewise-differentiable minimization problem (equivalent to a constrained minimization) when $p = 1$, through a once-differentiable minimization problem (but not twice differentiable) when $1 < p < 2$, to a twice-differentiable problem when $p = 2$. Clearly, it would be useful, although challenging, to develop a method that works well in all the cases.

The 2-norm problem is easy to solve: it is a simple least-squares problem. The l_1 and l_∞ problems are much more complicated and can be treated as linear programming problems and thus solved by special linear programming methods that usually take advantage of their special structures (e.g., [1], [2]). The objective function $\psi(x)$ is piecewise linear when $p = 1$ or ∞.

* Received by the editors June 20, 1991; accepted for publication (in revised form) June 6, 1992.

† Department of Computer Science, Cornell University, Ithaca, New York 14853. This research was partially supported by the U.S. Army Research Office through Mathematical Sciences Institute, Cornell University, and by the Computational Mathematics Program of the National Science Foundation grant DMS-8706133.

Until recently, the usual methods for l_1 and l_∞ have been *finite* algorithms (e.g., [1], [2]). These methods move along negative projected gradients, and the iterates tend to follow nondifferentiable hyperplanes. In contrast, Coleman and Li have recently developed *iterative* methods for l_1 and l_∞ minimization [4], [5]. These algorithms deal with the nondifferentiable hyperplanes by strategically avoiding landing on them exactly and by being able to cross when necessary. They are computationally efficient. Under a suitable nondegeneracy assumption, both algorithms proved to be globally convergent with a quadratic convergence rate.

For $1 < p < 2$ the most popular method for solving (1.1) is the iteratively reweighted least-squares (IRLS) method (e.g., [10], [13]). This method essentially takes a fixed step size along the Newton direction defined by the optimality condition $\nabla\psi(x) = 0$. It is globally linearly convergent for $1 < p < 2$ (e.g., [13]). This method can also be applied to the case $p = 1$, although no global convergence has been proved to our knowledge. With a suitable line search, the algorithm can be accelerated to be quadratically convergent when there is no zero residual at a solution. However, it is known that the method can be extremely slow, as will be further demonstrated by our numerical examples. Since the second-order derivative of the objective function does not exist when zero residuals occur, this is usually regarded as a main problem with the IRLS approach [3], [8].

The purpose of this paper is to further investigate the performance of the IRLS method and to provide a new method that works well for $1 \leq p < 2$. Our experience indicates (§2) that zero residuals at a solution alone do not, in general, impede the speed of the convergence for the IRLS approach; rather, slow convergence occurs when p is close or equal to unity. This is reasonable because (1.1) is a more complicated problem (linear programming) when $p = 1$, whereas a solution for (1.1) can be obtained by solving one linear system when $p = 2$. Nonetheless, there is an additional reason for the slowness of the IRLS method: the nonlinear equation $\nabla\psi(x) = 0$ that defines the Newton step for the IRLS method does not include the conditions for a solution of (1.1) when $p = 1$. On the basis of this observation, in §3 we develop a new method by considering the system of nonlinear equations that form part of the optimality conditions for (1.1) for all $1 \leq p < 2$. We also present a special line search procedure that exploits the special structure of the objective function and prevents zero residuals at each iteration. The new method performs significantly better than the IRLS methods, and it reduces to the method of Coleman and Li [5] when $p = 1$. We emphasize, however, that the main reason for the improvement is not the prevention of zero residuals in the line search. Rather, the consideration of the appropriate system of nonlinear equations brings about the improvement. In §4 we prove that the new algorithm is globally convergent. Superlinear convergence is achieved when there is no zero residual at the solution for $1 < p < 2$ and when a problem is nondegenerate for $p = 1$. Some numerical experience is reported in §5.

Finally, we introduce some notation. In this paper the superscript directly on a quantity denotes its value at the kth iteration, e.g., x^k. The conventional power operation is denoted by using brackets and superscripts together, e.g., $(x)^k$. We always use r to represent the residual vector $r = A^T x - b$, and σ denotes its sign, i.e., $\sigma = \mathrm{sgn}(r)$. Since we assume that A has full rank, the relation between x and r is a bijection. The objective function is denoted in terms of r by $\phi(r) = \|r\|_p^p, (= \psi(x))$, and the gradient $\nabla\phi(r)$, when it exists, is denoted by $g = p(|r|)^{p-1}\sigma$. In this paper the symbol $\overset{\text{l.s.}}{=}$ means that a linear system is solved in a least-squares sense, e.g., $A^T x \overset{\text{l.s.}}{=} b$

is equivalent to solving

$$\min_{x \in \Re^n} \|A^T x - b\|_2^2.$$

We also adopt a few MATLAB notations [11]. The symbols .* and ./ denote componentwise multiplication and division between vectors. The operator $|\cdot|$ denotes the componentwise absolute values of a number, vector, or matrix. The operator $\max(x, y)$ with two vectors as arguments defines a vector whose components are the maximum of the corresponding argument vectors. The notation $\max(x)$, where x is a vector, denotes the maximum component of x, whereas the operator $\mathrm{diag}(x)$, $x \in \Re^n$, represents the diagonal matrix with the ith diagonal element being x_i. The left arrow $x \leftarrow y$ denotes setting x to y.

2. IRLS methods. It is well known that the l_p-norm is differentiable and strictly convex for $1 < p < \infty$ under the assumption that A has full rank; thus the solution occurs at a point where the gradient $\nabla\psi(x) = Ag$ vanishes. Assume that we are at a point with $r_i \neq 0, 1 \leq i \leq m$. This is equivalent to

$$A(D)^{-2} r = 0, \tag{2.1}$$

where $D = \mathrm{diag}((|r|)^{(2-p)/2})$. The motivation of the IRLS method comes from the fact that (2.1) forms the normal equations for the following weighted least-squares system:

$$(D)^{-1} A^T x \overset{\text{l.s.}}{=} (D)^{-1} b.$$

Thus the IRLS method can simply be described as in Fig. 1.

```
Given a starting point $x^0$
Step 1  Compute $r^k = A^T x^k - b$;
Step 2  Define $D^k = \mathrm{diag}((|r^k|)^{(2-p)/2})$, solve $x^{k+1}$ from

        (2.2)   $(D^k)^{-1} A^T x^{k+1} \overset{\text{l.s.}}{=} (D^k)^{-1} b$;    $k \leftarrow k + 1$;

        Go to Step 1;
```

FIG. 1. IRLS *Algorithm.*

Remark. In the description of the algorithm it is implicitly assumed that at each iteration $r_i^k \neq 0$, $1 \leq i \leq m$. In practice, care must be taken when some $r_i^k = 0$. Let $e^T = [1, \ldots, 1] \in \Re^m$ be the m vector of all ones. Watson [14] suggested using $D^k = \mathrm{diag}(\max\{\delta e, |r^k|\})^{(2-p)/2}$ for some small positive constant δ. In our implementation we use $D^k = \mathrm{diag}(100\epsilon e + (|r^k|)^{(2-p)/2})$, where ϵ is the machine precision.

Let us inspect the step $x^{k+1} - x^k$ taken by IRLS more closely. If it is assumed that $r_i^k \neq 0$ $\forall 1 \leq i \leq m$, a Newton step in x-space for (2.1) can be defined through differentiation as

$$d_N^k = -\frac{1}{p-1}(A(D^k)^{-2} A^T)^{-1} A(D^k)^{-2} r^k. \tag{2.3}$$

It is clear that d_N^k is always a descent direction for the objective function $\psi(x)$. Consider the increment $\Delta x^k = x^{k+1} - x^k$ obtained from IRLS:

$$\begin{aligned}\Delta x^k &= (A(D^k)^{-2}A^T)^{-1}A(D^k)^{-2}b - (A(D^k)^{-2}A^T)^{-1}A(D^k)^{-2}A^Tx^k \\ &= -(A(D^k)^{-2}A^T)^{-1}A(D^k)^{-2}r^k && (p \geq 1) \\ &= (p-1)d_N^k && (p > 1).\end{aligned}$$

Hence Δx^k can be considered as a damped Newton step (e.g., [14]).

In [13] it is proved by assuming $1 < p < 2$ and $r_i^k \neq 0$, $1 \leq i \leq m$, that the limit point of the sequence $\{x^k\}$ generated by the IRLS algorithm is a solution to (1.1) and that the convergence is linear with convergence constant $2 - p$. Wolfe [15] obtained the same local convergence property with a rather involved proof.

When $p = 1$ there is no global convergence result to our knowledge; however, if global convergence is assumed, then the convergence rate will be linear [13]. We claim that a slight modification of the proofs in [5] yields that IRLS, when $p = 1$, is also globally convergent under some nondegeneracy assumptions.

The above IRLS method has a linear convergence rate because of its failure to take a full Newton step. However, taking a full Newton step at each iteration may lead to divergence [10]. Nonetheless, a line search globalization of the Newton method can be made to achieve final quadratic convergence and to maintain global convergence at the same time.

In this paper a line search procedure is used for improving both the IRLS method and the new algorithm. We refer to the modified IRLS algorithm as IRLSL (IRLS with the line search). Note that d_N^k as defined in (2.3) is the solution to the following least-squares problem:

$$(D^k)^{-1}A^Td_x^k \overset{\text{l.s.}}{=} -D^kg^k, \quad \text{where } D^k = (\mathrm{diag}(|r^k|)(\mathrm{diag}(p-1)|g^k|)^{-1})^{1/2}.$$

A model algorithm for IRLSL in terms of r is described in Fig. 2.

```
*Given an initial point* $r^0 = A^Tx^0 - b$ *with* $|r^0| > 0$
  *Step* 1  Compute $g^k = p(|r^k|)^{p-1}\sigma^k$, $D_r^k = \mathrm{diag}(|r^k|)$ and $D^k = (D_r^k\mathrm{diag}((p-1)|g^k|)^{-1})^{1/2}$;
  *Step* 2  Compute the direction $d^k$ by solving
  (2.4)     $\begin{cases} (D^k)^{-1}A^Td_x^k \overset{\text{l.s.}}{=} -D^kg^k \\ d^k = A^Td_x^k; \end{cases}$
  *Step* 3  Perform the line search as described below (see Fig. 3). Update
            $r^{k+1} \leftarrow r^k + \alpha^kd^k, \quad k \leftarrow k+1;$
            Go to Step 1;
```

FIG. 2. IRLSL *Algorithm.*

Remark. We describe IRLSL in this fashion so as to compare it with our new algorithm, which will be presented in §3. Since $r = A^Tx - b$ and A has full rank, x can be recovered from r on termination if needed. Alternatively, one can choose to update x directly at each iteration.

Now we discuss how to determine a suitable step size α^k. Given any descent direction $d^k \in \Re^m$, we determine a suitable step size α^k by attempting to minimize $\phi(r^k + \alpha d^k)$ over $\alpha \geq 0$.

The objective function $\phi(r)$ is continuously differentiable when $p > 1$. Thus, by following [7, Thm. 6.3.2], given $0 < \beta_f < \beta_g < 1$, there exists $0 < \alpha_l^k < \alpha_u^k$ such that, when $\alpha^k \in [\alpha_l^k, \alpha_u^k]$, the following conditions are satisfied at $r^{k+1} = r^k + \alpha^k d^k$:

$$\phi(r^{k+1}) \leq \phi(r^k) + \beta_f \alpha^k \nabla\phi(r^k)^T d^k, \tag{2.5}$$

$$\nabla\phi(r^{k+1})^T d^k \geq \beta_g \nabla\phi(r^k)^T d^k. \tag{2.6}$$

Unfortunately, the objective function $\phi(r)$ is not twice differentiable everywhere. Hence conditions (2.5) and (2.6) do not guarantee convergence to the solution. The difficulty is that condition (2.6) may not guarantee large enough step lengths because of the nonsmoothness of the derivatives.

Since the function $\phi(r^k + \alpha d^k)$ is strictly convex (under our assumption), there can be only one minimum along d^k. However, we do not want to perform an exact line search when $p > 1$ because of concern for efficiency. Instead, we exploit the special structure of the objective function $\phi(r)$ and perform the line search in the following fashion.

Consider the following strictly convex quadratic function $U^k(r)$ around a differentiable point r^k, i.e., $r_i^k \neq 0$, $1 \leq i \leq m$, as defined by Osborne [13, p. 252]:

$$U^k(r) = \frac{1}{2}\sum_{i=1}^{m} \frac{|g_i^k|}{|r_i^k|}(r_i)^2 + \sum_{i=1}^{m}\left((|r_i^k|)^p - \frac{1}{2}\frac{|g_i^k|}{|r_i^k|}(|r_i^k|)^2\right), \tag{2.7}$$

where $g^k = \nabla\phi(r^k)$. This quadratic function has been used [13, p. 252] to prove that $\{r^k\}$ generated by IRLS decreases the objective function $\phi(r)$ monotonically. It has the following properties [13, p. 252]:

$$\phi(r) \leq U^k(r), \qquad \phi(r^k) = U^k(r^k), \tag{2.8}$$

and

$$\nabla U^k(r^k) = \nabla\phi(r^k), \quad \nabla^2 U^k(r) = \mathrm{diag}(p(|r^k|)^{p-2}).$$

Moreover, $\arg\min_x U^k(r) = x^{k+1}$, where x^{k+1} is defined by the IRLS algorithm (cf. (2.2)). In other words, $U^k(r)$ is a special quadratic interpolation of $\phi(r)$.

In this paper we use this quadratic interpolation to facilitate the line search in both IRLSL and in our new method, which will be discussed later. We calculate the minimizer of $U^k(r)$ along any descent direction d^k and use it to approximate the minimizer of $\phi(r)$ along this direction. The minimizer for $U^k(r^k + \alpha d^k)$ equals

$$\breve{\alpha}^k = -\frac{{g^k}^T d^k}{{d^k}^T \mathrm{diag}(p(|r^k|)^{p-2}) d^k}. \tag{2.9}$$

For IRLSL $d^k = A^T d_x^k$, where d_x^k is defined by (2.4). Thus $\breve{\alpha}^k = p - 1$ and $\breve{\alpha}^k$ is the step size that IRLS takes at each iteration when $p > 1$.

The following lemma indicates that for any descent direction d^k, $\breve{\alpha}^k$ always introduces a sufficient decrease in objective function value. This explains why the sequence generated by IRLS converges to a solution when $1 < p < 2$.

LEMMA 2.1. *Assume $1 \leq p \leq 2$. Given any descent direction d^k, the step size $\breve{\alpha}^k$ as defined by* (2.9) *satisfies* (2.5) *with any $0 \leq \beta_f \leq \frac{1}{2}$.*

Proof. Let U^k be as defined by (2.7). Then

$$\begin{aligned}
&\phi(r^k) - \phi(r^k + \breve{\alpha}^k d^k) \\
&\quad \geq U^k(r^k) - U^k(r^k + \breve{\alpha}^k d^k) && \text{(from (2.8))} \\
&\quad = -\breve{\alpha}^k {g^k}^T d^k - \frac{1}{2}(\breve{\alpha}^k)^2 {d^k}^T \mathrm{diag}(p(|r^k|)^{p-2}) d^k && (U^k(r) \text{ is a quadratic}) \\
&\quad = -\breve{\alpha}^k {g^k}^T d^k + \frac{1}{2}\breve{\alpha}^k {g^k}^T d^k && \text{(from (2.9))} \\
&\quad = -\frac{1}{2}\breve{\alpha}^k {g^k}^T d^k \\
&\quad \geq -\beta_f \breve{\alpha}^k {g^k}^T d^k.
\end{aligned}$$

Hence (2.5) is satisfied. □

Quadratic interpolation techniques have been used in line search methods for general nonlinear minimization [9]. However, it is worth emphasizing that for general nonlinear functions the interpolation function is usually a one-dimensional function that is defined along a search direction instead of approximating the objective function in the entire space. For any given problem (1.1), the interpolation function $U^k(r)$ used here guarantees that the step size $\breve{\alpha}^k$ is acceptable for $\phi(r)$ (i.e., sufficient decrease is achieved and the step is not too small), which usually cannot be achieved for general nonlinear functions.

Since $\phi(r)$ becomes increasingly close to being nondifferentiable as p gets close to one, $\breve{\alpha}^k$ may be a bad choice (it converges to zero). When the objective function $\phi(r)$ is not differentiable, i.e., $p = 1$, the exact minimizer $\phi(r^k + \alpha d^k)$ occurs at a nondifferentiable point. Along any direction $d^k \in \Re^m$ the points at which the second-order derivatives fail to exist can easily be calculated. We refer to the step sizes corresponding to such points as breakpoints. The set $\mathcal{J}$ identifies the positive breakpoints:

$$\mathcal{J} = \left\{ \alpha_i^k : \alpha_i^k = -\frac{r_i^k}{d_i^k},\ r_i^k d_i^k < 0 \right\}. \tag{2.10}$$

The basic idea behind our line search procedure is to take larger step sizes when possible. We consider the first positive breakpoint at which d^k becomes an ascent direction, a unit step size, and $\breve{\alpha}^k$ in this order. The first at which the objective function is sufficiently decreased (i.e., (2.5) is satisfied) is accepted.

However, the exact nondifferentiable point needs to be avoided (if a unit step size or $\breve{\alpha}$ is taken, it is unlikely that r^{k+1} is a nondifferentiable point). We achieve this by slightly stepping back from a nondifferentiable point. Assume that $r_i^k + \omega d_i^k = 0$ at the step size $\omega > 0$ under consideration. Let

$$\alpha_\sharp^k \leftarrow \max\{\alpha_i^k : 0 \leq \alpha_i^k < \omega\},$$

and set

$$\alpha^k \leftarrow \alpha_\sharp^k + \tau^k(\omega - \alpha_\sharp^k), \tag{2.11}$$

where $\tau^k \in (0,1)$.

For IRLSL we choose

$$\tau^k = \max\left(\tau, 1 - \frac{\|Ag^k\|_2}{1 + \|Ag^k\|_2}\right) \tag{2.12}$$

and $\tau \in (0,1)$, e.g., $\tau = 0.975$. When $1 < p < 2$, $\|Ag^k\|$ is a measure of optimality. When a solution for $1 < p < 2$ is approached $\|Ag^k\|$ converges to zero and thus τ^k converges to one. Hence if (2.5) is satisfied with the unit step size, the perturbed α^k converges to unity, which is required for fast local convergence. By assuming that (2.5) is satisfied with ω, it is easy to verify that (2.5) is also satisfied with α^k defined by (2.11) since $\phi(r)$ is convex under our assumption. Thus (2.5) is always satisfied for the step size computed.

The line search procedure is summarized in Fig. 3. We point out that when $p = 1$, $g(r^k + \alpha_*^k d^k)$ does not exist and the gradient just past the breakpoint α_*^k is used (for details see [5]). Moreover, if it is assumed that $\beta_f = 0$ and $p = 1$, this line search procedure always locates the exact minimizer and ensures that $r_i^k \neq 0, 1 \leq i \leq m$, at each iteration (the line search procedure always returns at Step 1).

Given $\tau^k, \beta_f \in (0,1)$, d^k, r^k, $\breve{\alpha}^k$, α_i^k *(defined by* (2.10)*), and a large* $\rho_B > 0$ *(e.g.,* 10^6)

Step 1 Let α_*^k be the smallest positive breakpoint in $[\breve{\alpha}^k, \rho_B]$ with $g(r^k + \alpha_*^k d^k)^T d^k \geq 0$. If such a breakpoint α_*^k exists and (2.5) is satisfied with α_*^k, let $\alpha_\sharp^k \leftarrow \max\{\alpha_i^k : 0 \leq \alpha_i^k < \alpha_*^k\}$ and set

$$\alpha^k \leftarrow \alpha_\sharp^k + \tau^k(\alpha_*^k - \alpha_\sharp^k)$$

and return; Otherwise, continue;

Step 2 If (2.5) is not satisfied with $\alpha^k = 1$, continue to the next step. Otherwise, set

$$\alpha^k \leftarrow \begin{cases} 1 & \text{if } \min(|r^k + d^k|) > 0, \\ \alpha_\sharp^k + \tau^k(1 - \alpha_\sharp^k) & \text{otherwise,} \end{cases}$$

where $\alpha_\sharp^k \leftarrow \max\{\alpha_i^k : 0 \leq \alpha_i^k < 1\}$, return;

Step 3 Set

$$\alpha^k \leftarrow \begin{cases} \breve{\alpha}^k & \text{if } \min(|r^k + \breve{\alpha}^k d^k|) > 0, \\ \alpha_\sharp^k + \tau^k(\breve{\alpha}^k - \alpha_\sharp^k) & \text{otherwise,} \end{cases}$$

where $\alpha_\sharp^k \leftarrow \max\{\alpha_i^k : 0 \leq \alpha_i^k < \breve{\alpha}^k\}$, return;

FIG. 3. *Line search procedure.*

In §4 we will prove that the IRLSL algorithm with the above line search procedure is globally convergent. It is quadratically convergent when there is no zero residual at the solution.

Merle and Späth [10] empirically studied the IRLS algorithm and concluded that the IRLS algorithm (without a line search) is satisfactory. We disagree with this claim. To investigate the performance of the algorithms more carefully, we apply both the IRLS and IRLSL algorithms to some randomly generated l_p-norm problems (for details, see §5). The following stopping criterion is used:

$$\textbf{either} \quad \left|\frac{\psi(x^k)-\psi(x^{k+1})}{\psi(x^k)}\right| < \tau_s = \frac{1}{2}\times 10^{-11} \quad \textbf{or} \quad \text{itcount} > 50.$$

Here itcount denotes the number of iterations. For more discussion of the stopping criterion, see §5.

Table 1
Behavior of the two algorithms when p approaches one.

Number of Iterations ($m = 100$, $n = 50$)											
$p =$	1	1.01	1.02	1.03	1.04	1.05	1.06	1.07	1.08	1.09	1.1
IRLSL	50	50	36	50	40	38	28	33	30	29	24
IRLS	50	50	50	50	50	50	50	50	50	50	50

Table 2
Effect of zero residuals.

Number of Iterations ($m = 100$, $n = 50$)					
p (no $r_i^* = 0$)	IRLS	IRLSL	p (five $r_i^* = 0$)	IRLS	IRLSL
1.3	32	14	1.3	31	18
1.4	23	8	1.4	22	12
1.5	17	9	1.5	17	11
1.6	13	7	1.6	13	8
1.7	10	7	1.7	10	7
1.8	8	5	1.8	8	7
1.9	6	5	1.9	6	5

Tables 1 and 2 represent typical performance of IRLS and IRLSL.

First, we observe that IRLSL is more efficient than IRLS. Our computational experience indicates that IRLSL converges faster than does IRLS (e.g., Tables 1 and 2): even when both methods fail to find a solution, IRLSL computes an approximate solution with a lower objective function value. The additional cost per iteration for IRLSL is that of the line search, which is roughly $O(\kappa m)$, where κ is the number of positive breakpoints in $[\breve{\alpha}, \rho_B]$ that have to be inspected in order to find α_*, i.e., an inner product needs to be computed at every such point. In our experiments this number κ is in general much less than n and decreases quickly as p departs from unity. Thus the cost of the line search is of a lower order than that of solving a least-squares problem ($O(mn^2)$). Hence we conclude that IRLSL is more efficient than IRLS, and subsequently we will compare our new method (§3) with IRLSL only.

As indicated in Table 1, both algorithms (with or without a line search) converge increasingly slowly when p approaches unity. It is clear that when $p = 1$ one can always find a solution with n zero residuals (e.g., [1]). Thus when p is close to unity, there usually exist either zero residuals or extremely small residuals. Because the Hessian matrix of the objective function does not exist at points with zero residuals when $1 < p < 2$, it seems to be reasonable to blame the slow convergence on the occurrence of the zero residuals at a solution.

However, we argue that this is not the reason. Our argument is supported by the results in Table 2, which indicate that the presence of zero residuals at solutions does not significantly affect the algorithm when p is further away from unity. When a random l_p-norm problem, $p > 1.5$, is generated, it usually does not have zero residuals at the solution. For comparison, we generate random l_p-norm problems in a special way to guarantee the zero residuals at the solution: we solve an l_p-norm problem first and add more residuals so that they equal zero at the solution. As indicated by Table 1, both algorithms seem to be unaffected by the presence of zero residuals at a solution.

We will further investigate this question in the next section.

3. A new algorithm. The IRLSL method works well when p is sufficiently far from unity (e.g., $p > 1.3$), as indicated by our numerical results. However, when p is close to unity it becomes unsatisfactory. Moreover, our numerical experience indicates that a zero residual does not necessarily impede the speed of convergence. Hence alternative reasons for the slowness of the IRLS methods must be sought.

Recall that the descent directions used by both IRLS and IRLSL are derived from the nonlinear equations $\psi(x) = 0$. This is the optimality condition for (1.1) when $1 < p < 2$ but not when $p = 1$. Hence when $p = 1$ slow progress is made by moving along these descent directions because no attempt is made to satisfy the optimality conditions directly. We believe that this is the cause of the unsatisfactory performance of the IRLS methods when p is close or equal to unity.

Let the rows of the matrix Z form a basis for the null space of A, i.e., $AZ^T = 0$. Recall that $g = p(|r|)^{p-1}\sigma$. We can write (2.1) in the following equivalent form:

$$g - Z^T w = 0. \tag{3.1}$$

The number of equations is m, which is equivalent to the number of variables (x, w) (note that $x \in \Re^n$ and $w \in \Re^{m-n}$).

Let $D_r^k = \text{diag}(|r^k|)$, and denote $\lambda^k = Z^T w^k$. At any point (x^k, w^k) the Newton step for the above equations is defined by

$$\left[p(p-1)\text{diag}((|r^k|)^{p-2})A^T, -Z^T\right] \begin{bmatrix} d_x^k \\ d_w^k \end{bmatrix} = -\left[g^k - \lambda^k\right]. \tag{3.2}$$

Thus the Newton step for the x variables is

$$d_x^k = -\frac{1}{p-1}(A(D_r^k)^{-1}\text{diag}(|g^k|)A^T)^{-1}Ag^k,$$

which is equivalent to the Newton step (2.3) for $\nabla\psi(x) = 0$.

Now we consider the following nonlinear system of equations:

$$D_r(g - Z^T w) = 0. \tag{3.3}$$

When $p = 1$ this is the complementary slackness condition for a solution and λ is often called the vector of *dual multipliers.* In [5] we have used (3.3) to define local Newton steps for l_1-norm problems. When $1 < p < 2$, (3.3) is the optimality condition for (1.1) if D_r is nonsingular.

A solution to (3.1) is always a solution to (3.3). A solution (x, w) to (3.3) is a solution to (3.1) if for any $r_i = 0$, $\lambda_i = 0$. Hence we can compute a solution of (1.1) by satisfying (3.3) and the condition that $\lambda_i = 0$ if $r_i = 0$.

By considering (3.3) instead of (3.1), we capture both the optimality conditions for smooth minimization ($p > 1$) and part of the optimality conditions for nonsmooth minimization ($p = 1$). Given that the objective function $\phi(r)$ becomes nearly nonsmooth when p is close to unity, we argue that it is better to consider (3.3) than to consider (3.1). Since we are concerned with the l_p-norm problem for all $1 \leq p < 2$, taking Newton steps defined by (3.3) is more appropriate than using (3.1). This is the main idea behind our new method. Next we describe our new method in more detail.

Assume for now that the Jacobian of $D_r(g - Z^T w)$ exists at (x^k, w^k) and is nonsingular. Let $D_\lambda^k = \text{diag}(p\sigma^k .* \, g^k - \sigma^k .* \, \lambda^k)$. Then the Newton step for (3.3) is defined by

$$(3.4) \qquad [D_\lambda^k A^T, -D_r^k Z^T] \begin{bmatrix} d_x^k \\ d_w^k \end{bmatrix} = -\left[D_r^k (g^k - \lambda^k)\right].$$

Hence we obtain

$$(3.5) \qquad A(D_r^k)^{-1} D_\lambda^k A^T d_x^k = -Ag^k$$

or, equivalently,

$$(3.6) \qquad d_x^k = -(A(D_r^k)^{-1} D_\lambda^k A^T)^{-1} A g^k.$$

In [5] we have proved that when $p = 1$, $A(D_r)^{-1} D_\lambda A^T$ is positive definite in the neighborhood of the solution, under some nondegeneracy assumptions.

Consider the case in which $1 < p < 2$. If there is no zero residual at the solution, i.e., $|r^*| > 0$, $(D_r^*)^{-1} D_\lambda^*$ is positive definite since $D_\lambda^* = (p-1)\text{diag}(|g^*|)$ and A is assumed to have full rank. Thus $A(D_r^k)^{-1} D_\lambda^k A^T$ is also positive definite when (x^k, w^k) is close to (x^*, w^*). Hence the Newton direction d_x^k becomes a descent direction for $\psi(x)$ in a neighborhood of the solution.

If there exists some $r_i^* = 0$, the Jacobian matrix of (3.3) is singular at the solution when $1 < p < 2$ because $g_i^* = \lambda_i^* = 0$. However, at those points the Jacobian matrix of the original system (2.1) does not exist either. Hence this trouble is not introduced by considering (3.3) instead of (3.1). If there exists a zero residual at a solution x^*, it is difficult to achieve quadratic convergence and we are content with fast linear convergence.

Since $A(D_r^k)^{-1} D_\lambda^k A^T$ may not be positive definite far from a solution, globalization of the Newton step (3.5) is required.

First, we recall the technique used in [5] for $p = 1$. In [5] the Newton method is globalized by defining a diagonal matrix D_θ^k such that $A(D_r^k)^{-1} D_\theta^k A^T$ changes from $A(D_r^k)^{-1} A^T$ to $A(D_r^k)^{-1} D_\lambda^k A^T$ as the solution is approached and by replacing D_λ^k by D_θ^k when a direction is computed by (3.4). Thus the hybrid step can be considered as the solution to the following linear equations:

$$(3.7) \qquad [D_\theta^k A^T, -D_r^k Z^T] \begin{bmatrix} d_x^k \\ d_w^k \end{bmatrix} = -D_r^k (g^k - \lambda^k).$$

Hence

$$(3.8) \qquad d_x^k = -(A(D_r^k)^{-1} D_\theta^k A^T)^{-1} A g^k.$$

If a controlling variable $0 < \theta < 1$ that measures the closeness to the solution is used, the diagonal matrix D_θ is defined in the following way:

$$(3.9) \qquad D_\theta^k = |\theta^k \text{diag}(\sigma^k g^k) + (1 - \theta^k) D_\lambda^k| = \text{diag}(|g^k - (1 - \theta^k)\lambda^k|).$$

Here θ^k measures the satisfaction of the complementary slackness condition and the dual feasibility of an l_1-norm problem

$$\theta^k = \frac{\eta^k}{\gamma + \eta^k}, \tag{3.10}$$
$$\eta^k = \max\left\{\max\left\{\frac{|D_r^k(g^k - \lambda^k)|}{\phi(r^0)}\right\}, \max\{\max\{|\lambda^k| - |g^k|, 0\}\}\right\},$$

and $0 < \gamma < 1$ (in our implementation $\gamma = 0.99$). In other words, η^k is the maximum of the violation of the complementary slackness condition ($D_r(g - \lambda) = 0$) and of dual feasibility ($|\lambda| \le |g|$). Note that $|g| = |p(|r|)^{p-1}| = e$ when $p = 1$. In this case, $\theta = 0$ (or $\eta = 0$) is a necessary and sufficient condition of optimality. (For a more detailed discussion see [5].)

Now we consider the case in which $1 < p < 2$. Since we know that the direction defined by the IRLS methods leads to global convergence, we want to define a diagonal matrix D_θ such that globally the direction defined by replacing D_λ by D_θ is the same direction as that of IRLS and that locally it converges to D_λ. Notice that if we let $D_\theta = |\mathrm{diag}((p-1)g)|$, the direction defined by (3.7) equals the IRLS direction (2.3). Unfortunately, a simple scalar combination $|\theta\,\mathrm{diag}(pg) + (1-\theta)D_\lambda|$ does not lead to the IRLS direction globally because some components of the combination may not approach zero when the corresponding components in $\mathrm{diag}(p|g|)$ converge to zero. We form the diagonal matrix D_θ in a slightly more complicated way: the diagonal is the componentwise convex combination of that of $\mathrm{diag}(pg^k)$ and D_λ^k:

$$\begin{aligned} D_\theta^k &= |\mathrm{diag}(\theta^k)\mathrm{diag}(p\sigma^k g^k) + \mathrm{diag}(e - \theta^k)D_\lambda^k| \\ &= \mathrm{diag}(|pg^k - (e - \theta^k)\ .*\ \lambda^k|). \end{aligned} \tag{3.11}$$

(Recall that the operators .* and ./ denote componentwise multiplications and divisions between vectors.) Here θ^k is a vector

$$\theta^k = (\eta^k e)\ ./\ (\gamma|g^k| + \eta^k e), \tag{3.12}$$

where γ is, again, a constant with $0 < \gamma < 1$ and $e^T = [1, \ldots, 1] \in \Re^m$. The scalar η^k is as defined in (3.10). It is clear that when $p = 1$, (3.11) is the same as definition (3.9), which is used in [5]. Hence when $p = 1$, D_θ^k defined by (3.12) is equivalent to that defined by (3.10). Moreover, x is optimal if and only if there exists $\lambda = Z^T w$ such that $\eta = 0$.

The diagonal matrix D_θ^k has the following properties.

LEMMA 3.1. *Suppose* $0 < \gamma < 1$. *Assume* D_θ^k *is defined by* (3.11). *Then* D_θ^k *satisfies*

$$(p-1)\mathrm{diag}(|g^k|) \le |D_\theta^k| \le (p+1)\mathrm{diag}(|g^k|). \tag{3.13}$$

Proof. By definition (3.11)

$$D_\theta^k = \mathrm{diag}(|pg^k - (e - \theta^k).*\ \lambda^k|).$$

From definition (3.12) of θ

$$\eta^k(e - \theta^k) = \gamma\theta^k.*\ |g^k|.$$

Hence

$$(|\lambda^k| - |g^k|).* (e - \theta^k) \le \gamma\theta^k .* |g^k|.$$

Therefore,

$$\begin{aligned} |\lambda^k| &\le |g^k| + \gamma(\theta^k . * |g^k|)./(e - \theta^k) \\ &\le ((e - \theta^k). * |g^k| + \gamma\theta^k . * |g^k|)./(e - \theta^k) \\ &\le ((e - (1 - \gamma)\theta^k). * g^k)./(e - \theta^k). \end{aligned}$$

Hence

$$(p-1)\mathrm{diag}(|g^k|) \le |D_\theta^k| \le (p+1)\mathrm{diag}(|g^k|). \qquad \Box$$

As will be shown in §4, with a suitable line search this globalization guarantees that when $1 < p < 2$, $\{\lambda_i^k\}$ converges to zero if $\{r_i^k\}$ converges to zero. Hence the corresponding $\{x^k\}$ converges to a solution of (1.1).

We apply the same line search procedure to the new method. However, the definition of τ^k in (2.12) is replaced by

$$\tau^k = \max\left(\tau, 1 - \frac{\eta^k}{\gamma + \eta^k}\right) \tag{3.14}$$

so as to include the measure of the optimality for $p = 1$. Note that when $\{\eta^k\}$ converges to zero, $\{\tau^k\}$ converges to unity. When $p = 1$ the line search procedure for the new method is equivalent to the one used in [5].

For IRLSL $\breve{\alpha}^k$ is a constant $p - 1$. For our new algorithm with d^k defined by (3.8), $\breve{\alpha}^k$ changes at each iteration. However, it is bounded between $p - 1$ and $p + 1$ as indicated by the following lemma.

LEMMA 3.2. *Assume $d = A^T d_x^k$, where d_x^k is defined by* (3.8)*. Then the step size $\breve{\alpha}^k$ as defined by* (2.9) *satisfies*

$$p - 1 \le \breve{\alpha}^k \le p + 1.$$

Proof. By definition (2.9)

$$\begin{aligned} \breve{\alpha}^k &= -\frac{{g^k}^T d^k}{{d^k}^T \mathrm{diag}(p(|r^k|)^{p-2}) d^k} \\ &= \frac{{d^k}^T (D_r^k)^{-1} D_\theta^k d^k}{{d^k}^T \mathrm{diag}(p(|r^k|)^{p-2}) d^k} \qquad \text{(from (3.8))}. \end{aligned}$$

From (3.13)

$$(p-1)\frac{{d^k}^T (D_r^k)^{-1} \mathrm{diag}(|g^k|) d^k}{{d^k}^T \mathrm{diag}(p(|r^k|)^{p-2}) d^k} \le \breve{\alpha}^k \le (p+1)\frac{{d^k}^T (D_r^k)^{-1} \mathrm{diag}(|g^k|) d^k}{{d^k}^T \mathrm{diag}(p(|r^k|)^{p-2}) d^k}.$$

This means

$$(p-1) \le \breve{\alpha}^k \le (p+1). \qquad \Box$$

Computationally, instead of solving an $m \times m$ linear system (3.7) to compute (d_x^k, d_w^k), one may prefer to compute d_x^k by solving an $m \times n$ least-squares problem

$$(D^k)^{-1}A^T d_x^k \overset{\text{l.s.}}{=} -D^k g^k,$$

where $D^k = (D_r^k(D_\theta^k)^{-1})^{1/2}$. Hence

$$\left\{ \begin{array}{l} (D^k)^{-1}A^T d_x^k \overset{\text{l.s.}}{=} -D^k g^k, \\ d^k = A^T d_x^k. \end{array} \right. \tag{3.15}$$

Once $d^k = A^T d_x^k$ is computed, λ can be updated by

$$\lambda^{k+1} \leftarrow (D_r^k)^{-1} D_\theta^k d^k + g^k. \tag{3.16}$$

The new method is referred to as GNCS: a globalized Newton method that uses the complementary slackness conditions for l_p-norm problems. It is summarized in Fig. 4.

```
Given an initial point $r^0 = A^T x^0 - b$ with $|r^0| > 0$ and $\lambda^0$
 Step 1  Compute $\theta^k$ by (3.12) and $g^k = p(|r^k|)^{p-1}\sigma^k$; Let $D_r^k = \mathrm{diag}(|r^k|)$,
         let $D_\theta^k = \mathrm{diag}(|pg^k - (e - \theta^k).*\lambda^k|)$, and define $D^k = (D_r^k(D_\theta^k)^{-1})^{1/2}$;
 Step 2  Compute the direction $d^k$ by
             $\left\{ \begin{array}{l} (D^k)^{-1}A^T d_x^k \overset{\text{l.s.}}{=} -D^k g^k, \\ d^k = A^T d_x^k; \end{array} \right.$
         Update $\lambda^{k+1}$:
             $\lambda^{k+1} \leftarrow (D_r^k)^{-1} D_\theta^k d^k + g^k$;
 Step 3  Compute $\tau^k$ by (3.14); Apply the line search procedure as described
         in Fig. 3; Update:
             $r^{k+1} \leftarrow r^k + \alpha^k d^k, \quad k \leftarrow k + 1$;
         Go to Step 1;
```

FIG. 4. *The* GNCS *Algorithm.*

Remark. It is interesting that we can express the fact that the function is smooth through (3.3): optimality conditions simply require $\lambda^* = g^*$. Thus if we ignore the requirement that $\lambda = Z^T w$, we may set

$$\lambda^{k+1} \leftarrow g^{k+1}. \tag{3.17}$$

If this definition of λ^{k+1} is used, $\eta^{k+1} = 0$ and $\theta^{k+1} = 0$. When $\theta^k = 0$, step (3.15) is equivalent to the Newton step (3.6). Hence GNCS becomes IRLSL if we set $\theta^k = 0$ at each iteration. Indeed, GNCS and IRLSL are computationally very similar. The only difference is that for IRLSL, $D_\theta^k = \mathrm{diag}(|(p-1)g^k|)$ and the multiplier information $\{\lambda^k\}$ is not used in defining descent directions. The multipliers are used in GNCS and can be obtained at almost no cost.

In the next section we prove that GNCS is globally convergent for all $1 < p < 2$. Moreover, when there is no $r_i^* = 0$ at a solution we have $(A(D_r^k)^{-1}D_\theta^k A^T) \rightarrow (A(D_r^k)^{-1}D_\lambda^k A^T)$ fast enough so that superlinear convergence is achieved.

4. Convergence properties. As we have mentioned before, when $p = 1$, the GNCS algorithm is equivalent to the method Coleman and Li proposed and analyzed in [5]. Thus when $p = 1$ GNCS is globally and quadratically convergent under some nondegeneracy assumptions. We need only to consider the convergence of GNCS when $1 < p < 2$. For the rest of this section we assume that $1 < p < 2$.

For IRLS (without the line search) global convergence is established in [13]. However, the convergence of IRLSL (with the line search) still needs to be established. The convergence for GNCS when $1 < p < 2$ does not follow automatically from the convergence theory [7] for general line-search-based algorithms because the objective function is not twice differentiable everywhere. In addition, our line search procedure is not standard.

We first consider global convergence for both IRLSL and GNCS.

Let P^k be the orthogonal projector onto the orthogonal space of ZD^k, i.e.,

$$P^k = I - D^k Z^T (Z(D^k)^2 Z^T)^{-1} Z D^k.$$

Assume D^k equals either $(D_r^k(\text{diag}((p-1)|g^k|)^{-1})^{1/2}$ or $(D_r^k {D_\theta^k}^{-1})^{1/2}$, depending on whether IRLSL or GNCS is being considered. Then

$$\begin{aligned} d^k &= -A^T(A(D^k)^{-2}A^T)^{-1}Ag^k \\ &= -D^k P^k D^k g^k \qquad (4.1)\\ &= -(D^k)^2(g^k - \lambda^{k+1}), \end{aligned}$$

where $\lambda^{k+1} = Z^T w^{k+1}$ and w^{k+1} is the least-squares solution to

$$D^k Z^T w^{k+1} \overset{\text{l.s.}}{=} D^k g^k.$$

First, we prove that $\{d^k\}$ generated by each algorithm converges to zero.

THEOREM 4.1. *Let D^k and d^k be defined by* GNCS (*or by* IRLSL). *Then* $\lim_{k\to\infty} \|P^k D^k g^k\|_2 = 0$ *and* $\lim_{k\to\infty} d^k = 0$.

Proof. It is clear that

$$(4.2) \qquad \phi(r^k) - \phi(r^0) = \sum_{j=0}^{k-1} (\phi(r^{j+1}) - \phi(r^j))$$

$$(4.3) \qquad \le \sum_{j=0}^{k-1} \beta_f \alpha^j {g^j}^T d^j \qquad \text{(from Lemma 2.1)},$$

with $\beta_f > 0$. Since $\{\phi(r^k)\}$ is bounded and $\alpha^k {g^k}^T d^k < 0$ always,

$$\lim_{k\to\infty} \alpha^k {g^k}^T d^k = 0.$$

From the line search procedure we have $\alpha^k \ge \check{\alpha}^k$. Using Lemma 3.2, we have $\alpha^k \ge (p-1)$. Hence

$$\lim_{k\to\infty} {g^k}^T d^k = 0.$$

But ${g^k}^T d^k = -\|P^k D^k g^k\|_2^2$ according to (4.1). This means

$$\lim_{k\to\infty} \|P^k D^k g^k\|_2 = 0.$$

Since $\phi(r^k)$ is bounded below, $\{r^k\}$ is bounded. From Lemma 3.1 there exists $M > 0$ such that

$$|D^k| \leq \frac{1}{p(p-1)}(\text{diag}(|r^k|^{2-p}))^{1/2} \leq M.$$

Using (4.1) again, we obtain

$$\lim_{k\to\infty} d^k = 0. \qquad \square$$

Next we prove that $\{r^k\}$ converges.

THEOREM 4.2. *Let $\{r^k\}$ be obtained by* GNCS (*or by* IRLSL). *Then $\{r^k\}$ converges to r^*.*

Proof. Let $\mathcal{S} = \{\bar{r} \ : \ \bar{r}$ is a limit point of $\{r^k\}\}$. From Lemma (3.2)

$$\check{\alpha}^k \leq p + 1.$$

With our line search, $\alpha^k \leq \max\{\rho_B, 1, \check{\alpha}^k\}$. Hence $\{\alpha^k\}$ is bounded. From Theorem 4.1 we have

$$\lim_{k\to\infty} \alpha^k d^k = 0.$$

Since $\{r^k\}$ is bounded and $\{\alpha^k d^k\}$ converges to zero, $\mathcal{S}$ is closed and connected [12, p. 478].

Since $\{\phi(r^k)\}$ is monotonically decreasing and bounded below and $\phi(r)$ is continuous, there exists an r^* such that

$$\lim_{k\to\infty} \phi(r^k) = \phi(r^*).$$

Hence for any limit point $\bar{r} \in \mathcal{S}$, $\phi(\bar{r}) = \phi(r^*)$. In addition, since $\mathcal{S}$ is closed and connected and $\phi(r)$ is strictly convex, $\mathcal{S}$ can contain only one point. From the boundedness of $\{r^k\}$ and the uniqueness of its limit point, we conclude that $\{r^k\}$ converges to r^*. $\square$

Finally, we prove that by assuming $r^k = A^T x^k - b$, $\{x^k\}$ converges to a solution of (1.1).

THEOREM 4.3. *Let $\{r^k\}$ be obtained by* GNCS (*or by* IRLSL), *and let $r^k = A^T x^k - b^k$. Assume that at the limit point $r^* = A^T x^* - b$, $\{a_i \ : \ b_i - a_i^T x^* = 0\}$ is a linearly independent set. Then $\{\lambda^k\}$ converges to λ^* and $\{x^k\}$ converges to the solution of* (1.1).

Proof. Following Theorem 4.2 there exists r^* such that $\lim_{k\to\infty} r^k = r^*$. Thus there exists x^* with $\lim_{k\to\infty} x^k = x^*$.

Let $Z = [z_1, \ldots, z_m]$, and let $\mathcal{A}_c^* = \{i \mid r_i^* \neq 0\}$. Since $\lim_{k\to\infty} D^k(g^k - Z^T w^{k+1}) = 0$, any limit point $\bar{w}$ of $\{w^{k+1}\}$ satisfies $z_i^T \bar{w} = g_i^*$, $\forall i \in \mathcal{A}_c^*$. By assumption that at the limit point r^*, $\{a_i \ : \ b_i - a_i^T x^* = 0\}$ is linearly independent, $z_i^T w = g_i^*, i \in \mathcal{A}_c^*$ has a unique solution. Hence $\{\lambda^k = Z^T w^k\}$ is bounded and converges to λ^*.

We prove that x^* is a solution by showing that $\lambda_i^* = 0$ if $r_i^* = 0$. Assume otherwise, i.e., that there exists some $\lambda_j^* \neq 0$ with $r_j^* = 0$. Consider the breakpoint α_j^k as defined by (2.10). Then

$$\alpha_j^k = \begin{cases} \dfrac{\sigma_j |p g_j^k - (1-\theta_j^k)\lambda_j^k|}{g_j^k - \lambda_j^{k+1}} & \text{for GNCS,} \\[2ex] \dfrac{r_j^k}{(|r_j^k|)^{2-p}(g_j^k - \lambda_j^{k+1})} & \text{for IRLSL.} \end{cases}$$

It is clear that $\{\alpha_j^k\}$ converges to zero because $\{g_j^k\}$ and $\{1-\theta_j^k\}$ converge to zero. Hence there exists k_1 such that when $k \geq k_1$, $\alpha_j^k < \check{\alpha} = p-1$, all nonzero λ_j^k remain the same sign and $|r_j^k|^{p-1} < |\lambda_j^k|$ for all $r_j^* = 0$ with $\lambda_j^* \neq 0$.

By using Lemma 3.2 (or Lemma 2.1), $\alpha^k > \check{\alpha}^k > \alpha_j^k$ for $k \geq k_1$. If $r_j^{k_1}$ and $\lambda_j^{k_1+1}$ have different signs, at iteration $k = k_1 + 1$, λ_j^{k+2} and r_j^{k+1} will have the same sign because $\alpha^k > \alpha_j^k$. If, for $\hat{k} > k_1$, $r_j^{\hat{k}}$ and $\lambda_j^{\hat{k}+1}$ have the same sign, it will remain so for $k > \hat{k}$ because $g_j^k d_j^k > 0$. But this means $|r_j^k|$ will be increased for $k > k_1$. This contradicts the fact that $r_j^* = 0$. □

Now we discuss the local convergence properties of the two algorithms. If at the solution r^* there is some $r_i^* = 0$, the Hessian matrix of $\psi(x)$ does not exist at a corresponding x^*. Hence, theoretically, we do not expect superlinear convergence for either the IRLSL or the GNCS algorithm.

Assume that $r_i^* \neq 0$ for any $1 \leq i \leq m$. The Hessian matrix of $\psi(x)$ is positive definite at x^*. Following Theorem 4.1, $\{d^k\}$ converges to zero. If our line search procedure is used, every positive breakpoint α_i^k converges to infinity. This means that the unit step size is tested for acceptance for sufficiently large k. Since $\psi(x)$ is twice continuously differentiable near x^*, a unit step size is admissible for a Newton step or a quasi-Newton step (e.g., [7]). In addition, perturbation to the unit step size is not necessary close to the solution because $|r^k + d^k| > 0$ for sufficiently large k. Hence $\alpha^k = 1$ for sufficiently large k. Thus the IRLSL method is locally equivalent to the Newton method for minimizing $\psi(x)$, which is a locally twice continuously differentiable function. Hence, if standard unconstrained minimization convergence analysis is followed (e.g., [7]), the IRLSL method is locally quadratically convergent. Similarly, the GNCS algorithm is locally equivalent to a quasi-Newton method for the minimization of a twice continuously differentiable function $\psi(x)$, with the Hessian matrix replaced by the matrix $A^T(D_r^k)^{-1}D_\theta^k A$. Moreover, we have

$$\begin{aligned}\lim_{k\to\infty} & \frac{\|\nabla\psi(x^k) - \nabla^2\psi(x^k)d_x^k\|}{\|d_x^k\|} \\ &= \lim_{k\to\infty} \frac{\|(A^T {D_r^k}^{-1} D_\theta^k A - A^T {D_r^k}^{-1}\operatorname{diag}(|(p-1)g^k|)A)d_x\|}{\|d_x^k\|} = 0\end{aligned}$$

since $\{D_\theta^k - \operatorname{diag}((p-1)|g^k|)\}$ converges to zero. From [6, Thm. 6.4], $\{x^k\}$ converges superlinearly to x^*.

In summary, we have shown that under the assumptions of Theorem 4.3 a sequence $\{r^k\}$ generated by either IRLSL or GNCS from any starting point $r^0 = A^T x^0 - b$ with $|r^k| > 0$ converges to a solution. If it is assumed that there is no zero residual at the solution, IRLSL is *locally quadratically* convergent, whereas the GNCS method is *locally superlinearly* convergent.

5. Numerical experiments. In this section we compare the computational performance of the IRLSL method with that of the proposed GNCS algorithm. All the experiments are done in MATLAB [11]. The numerical results clearly show the superiority of GNCS over IRLSL (and thus over IRLS as well).

The dominant cost of the computation of the two methods is the same: solving a weighted least-squares problem of the same dimension and structure per iteration. Moreover, the same line search procedure is used.

Now we discuss possible stopping criteria for problem (1.1).

Assume $1 < p < 2$. The optimality condition is simply $Ag^* = 0$ or, equivalently, $\eta^* = 0$ (see (3.12)). We point out, however, that testing $\|Ag\|$ against a tolerance is generally not a good stopping criterion. When p is close to unity, the gradient function $A(D_r)^{p-1}\sigma$ is ill conditioned in the neighborhood of a point where some $r_i = 0$, i.e., a small change of a variable may lead to a large change in the gradient. As an example, let us consider a simple scalar function $\psi(\xi) = |\xi|^{1.001}$. The gradient function is equal to $1.001\xi^{0.001}\,\mathrm{sgn}(\xi)$. Even when $\xi = 2.2204\times 10^{-16}$ (machine precision in MATLAB), the gradient $\nabla\psi(\xi)$ equals 0.9656. Since the gradient should be zero when $\xi = 0$, it is clear that the gradient function is extremely unstable.

In our computation we terminate the calculation when the algorithm has stopped decreasing the objective function. More specifically, we stop the computation when

$$\textbf{either}\ \ \frac{|\phi(r^{k+1}) - \phi(r^k)|}{\phi(r^{k+1})} < \tau_s \ \ \textbf{or}\ \ \eta^k < \tau_s \ \ \textbf{or}\ \ \text{itcount} > 50,$$

where τ_s has been set to $\frac{1}{2}10^{-11}$ and itcount denotes the number of iterations. For the GNCS algorithm, if $p = 1$ or if $1 < p < 2$ but there is no zero residual at the solution, we observe that final superlinear or quadratic convergence is achieved and that the accuracy of the computed solution is about τ_s (since η^k is about τ_s at termination). For the IRLSL method this is true only when $1 < p < 2$ and there is no zero residual at the solution.

For the results reported in this paper the parameters required by the algorithms are set as follows:

$$\tau \leftarrow 0.975, \qquad \beta_f \leftarrow \epsilon, \qquad \gamma \leftarrow 0.99,$$

where ϵ is machine precision.

TABLE 3
Function approximation problems.

$m = 200$, $n = 6$, $f_1(z)$		
p	GNCS	IRLSL
1	11	20
1.001	13	30
1.01	12	25
1.1	11	22
1.2	10	45
1.3	8	26
1.4	9	22
1.5	8	17
1.6	7	17
1.7	6	12
1.8	5	8
1.9	4	5

$m = 200$, $n = 10$, $f_2(z)$		
p	GNCS	IRLSL
1	12	50
1.001	11	50
1.01	15	50
1.1	10	33
1.2	9	23
1.3	7	27
1.4	8	20
1.5	6	15
1.6	6	12
1.7	6	11
1.8	6	7
1.9	4	6

The starting point for both IRLSL and GNCS is computed as the solution to $A^Tx \overset{\text{l.s.}}{=} b$. Our experience indicates that the role of λ^0 is less significant, and we set it in a similar way to that defined in [5]:

$$\lambda^0 = \tau\frac{g^0}{\max(|r^0|)}.$$

Next, we generate some test problems from discrete approximation.

Function approximation problems. Approximate $f(z)$, evaluated at $z = 0, \frac{1}{m}, \ldots, 1$, by a polynomial of degree $n-1$: $\sum_{j=1}^{n} x_j z^{j-1}$ such that the l_p-norm residuals are minimized. The two test functions used are

$$f_1(z) = \sqrt{1+z}, \qquad f_2(z) = e^z + \begin{cases} 5 & \text{if } 0.1 < z < 0.2, \\ 0 & \text{otherwise.} \end{cases}$$

As indicated by Table 3, GNCS is consistently better than IRLSL. The first function $f_1(z)$ is continuous, whereas the second function $f_2(z)$ is not. For $f_1(z)$ and $p = 1.9$, the best l_p-norm residual is $\phi(r^*) = 4.97528518113 \times 10^{-10}$. For the second function $f_2(z)$, if $p = 1.9$, the best l_p-norm residual is $\phi(r^*) = 1.7535105 \times 10^2$.

Random problems. We also generate random test problems by generating random entries for matrix A and right-hand side b by using the random number generator (with normal distribution) in PRO-MATLAB [11].

TABLE 4
$p = 1$.

Number of Steps		$m = 100$
n	GNCS	IRLSL
10	12	50
30	14	50
50	12	50
70	13	50
90	14	50

Number of Steps		$m = 200$
n	GNCS	IRLSL
10	17	50
30	17	50
50	15	50
70	21	50
90	15	50
110	14	50
130	17	50
150	13	50
170	13	50
190	9	50

TABLE 5
$p = 1.001$.

Number of Steps		$m = 100$
n	GNCS	IRLSL
10	11	27
20	14	46
30	20	50
40	16	50
50	16	50
60	17	50
70	14	50
80	11	50
90	13	37

Number of Steps		$m = 200$
n	GNCS	IRLSL
10	15	38
30	18	50
50	15	50
70	17	50
90	21	50
110	15	50
130	17	50
150	14	50
170	18	50
190	13	50

Table 4 exhibits the number of iterations required by GNCS and IRLSL when $p = 1$. The IRLSL method stops after 50 iterations with the objective function having only a few digits of accuracy. The GNCS algorithm is essentially the method presented in [5], and it demonstrates fast convergence.

When p is very close to unity (e.g., see Tables 5 and 6 with $p = 1.001, 1.01$), the number of zero residuals at a solution is usually slightly less than n. The GNCS algorithm exhibits final superlinear convergence behavior because n of the residuals are usually nearly zero at the solution. The GNCS algorithm behaves as though approaching a vertex and thus demonstrates superlinear behavior when approaching the neighborhood of the solution. At termination the objective function values computed by GNCS are always smaller than that of IRLSL. Comparing the IRLSL solutions with the more accurate GNCS solutions, we see that the former typically have about six digits of accuracy. The IRLSL method again shows extremely slow convergence and fails to find a solution after 50 iterations for the majority of problems.

TABLE 6
$p = 1.01$.

Number of Steps	$m = 100$	
n	GNCS	IRLSL
10	12	34
30	12	50
50	13	50
70	13	50
90	16	50

Number of Steps	$m = 200$	
n	GNCS	IRLSL
10	11	33
30	18	50
50	18	48
70	19	41
90	17	50
110	17	50
130	17	50
150	15	47
170	13	50
190	17	50

When p is further away from unity (e.g., Tables 7 and 8 with $p = 1.1, 1.3$), the number of zero residuals at the solution is less. However, many residuals are still relatively small. Hence the GNCS algorithm again approaches the neighborhood of a solution with a few final superlinear steps. Here the IRLSL method finds a solution with the required accuracy, but the number of iterations required by IRLSL is more than twice of that of the GNCS algorithm (see Tables 7 and 8).

When p is significantly larger than unity (e.g., Table 9 with $p = 1.7$), there usually exists no zero residual at the solution. Thus both the GNCS algorithm and the IRLSL method converge quickly to solutions and exhibit fast convergence. For these problems the two methods have roughly the same behavior.

In summary, the GNCS algorithm works very well for all $1 \leq p < 2$. It always performs significantly better than IRLSL when p is close to unity ($p < 1.5$). When p is *very close or equal* to unity, the IRLSL method is extremely inefficient, whereas the GNCS method finds the solutions in about 18 iterations. The latter is slightly better than IRLSL when $p \geq 1.5$ and there exists no zero residual at a solution.

Finally, we point out that the number of iterations required by the GNCS method appears to be relatively insensitive to the problem size.

6. Conclusions. In this paper we have developed a new efficient method that solves the l_p-norm minimization problem with $1 \leq p < 2$. We also have further investigated the performance of the classical IRLS method and have compared it with the new approach. We observed that the slow convergence of the IRLS (or IRLSL) method is not entirely due to the zero residuals at a solution but is also due to the fact that the constrained aspect is not taken care of: the Newton steps for the IRLS methods are based on the optimality conditions for the unconstrained problem ($1 < p < 2$) but not the constrained case ($p = 1$). On the basis of this observation we developed the GNCS method, which uses the Newton directions derived from the optimality conditions for all $1 \leq p < 2$.

TABLE 7
$p = 1.1$.

Number of Steps $m = 100$		
n	GNCS	IRLSL
10	11	19
30	9	24
50	11	24
70	10	25
90	10	28

Number of Steps $m = 200$		
n	GNCS	IRLSL
10	10	15
30	11	26
50	12	28
70	11	26
90	11	29
110	10	34
130	10	27
150	12	27
170	10	37
190	10	32

TABLE 8
$p = 1.3$.

Number of Steps $m = 100$		
n	GNCS	IRLSL
10	7	8
30	8	10
50	8	11
70	9	13
90	8	15

Number of Steps $m = 200$		
n	GNCS	IRLSL
10	8	13
30	9	13
50	8	13
70	8	13
90	8	13
110	9	15
130	9	16
150	9	17
170	9	17
190	9	19

TABLE 9
$p = 1.7$.

Number of Steps $m = 100$		
n	GNCS	IRLSL
10	6	5
30	6	8
50	7	7
70	9	7
90	8	11

Number of Steps $m = 200$		
n	GNCS	IRLSL
10	5	5
30	6	7
50	6	7
70	6	6
90	7	7
110	7	7
130	6	6
150	7	10
170	8	9
190	7	11

The GNCS method is attractive because of its capability to efficiently solve the l_p-norm minimization problem with the entire range $1 \leq p < 2$. When $p = 1$ it is exactly the approach for l_1 presented in [5] and is quadratically convergent under nondegeneracy assumptions. When $p > 1$ the new method is superlinearly convergent when there are no zero residuals at the solution.

The GNCS method is significantly better than the IRLSL algorithm when p is close or equal to unity. The computational cost of each iteration of the two methods is the same: the main cost is solving a weighted least-squares problem of the same size and structure. The difference between the two methods lies only in the definition of the different diagonal scaling matrices that define descent directions: in our new method the multiplier information is incorporated in the diagonal scaling matrix, and this is the key to a significant improvement.

Acknowledgments. The author thanks Thomas Coleman for numerous helpful suggestions. The author is also grateful to Mike Overton and the anonymous referees, whose suggestions improved the presentation of the paper.

REFERENCES

[1] I. Barrodale and F. Roberts, *An improved algorithm for discrete l_1 linear approximation*, SIAM J. Numer. Anal., 10 (1973), pp. 839–848.

[2] R. H. Bartels, A. R. Conn, and J. W. Sinclair, *Minimization techniques for piecewise differentiable functions: The l_1 solution to an overdetermined linear system*, SIAM J. Numer. Anal., 15 (1978), pp. 224–240.

[3] R. H. Byrd, *Algorithms for robust regression*, in Nonlinear Optimization 1981, M. Powell, ed., Academic Press, New York, 1982, pp. 79–89.

[4] T. F. Coleman and Y. Li, *A global and quadratically-convergent method for linear l_∞ problems*, SIAM J. Sci. Statist. Comput., 29 (1992), pp. 1166–1186.

[5] ———, *A globally and quadratically convergent affine scaling method for linear l_1 problems*, Math. Programming, 56 (1992), pp. 189–222.

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

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

[8] J. Fisher, *An algorithm for discrete linear l_p approximation*, Numer. Math., 38 (1981), pp. 129–139.

[9] P. Gill, W. Murray, and M. Wright, *Practical Optimization*, Academic Press, New York, 1981.

[10] G. Merle and H. Späth, *Computational experience with discrete l_p-approximation*, Computing, 12 (1974), pp. 315–321.

[11] C. B. Moler, J. Little, S. Bangert, and S. Kleiman, *ProMatlab User's Guide*, MathWorks, Sherborn, MA, 1987.

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

[13] M. R. Osborne, *Finite Algorithms in Optimization and Data Analysis*, John Wiley, New York, 1985.

[14] G. A. Watson, *On two methods for discrete l_p approximation*, Computing, 18 (1977), pp. 263–266.

[15] J. M. Wolfe, *On the convergence of an algorithm for discrete l_p approximation*, Numer. Math., 32 (1979), pp. 439–459.

SIAM J. OPTIMIZATION
Vol. 3, No. 3, pp. 630–636, August 1993

A COLLINEAR SCALING INTERPRETATION OF KARMARKAR'S LINEAR PROGRAMMING ALGORITHM*

J. C. LAGARIAS†

Abstract. In 1980 W. C. Davidon proposed a class of unconstrained minimization methods, called *collinear scaling algorithms*, that are invariant under projective transformations. In these methods the nonlinear function f to be minimized is approximated near a point x_0 by a suitable conic function $q(x_0+p) = f_0 + \langle g,p\rangle/(1+\langle d,p\rangle) + \frac{1}{2}\langle p,Ap\rangle/(1+\langle d,p\rangle)^2$ and the conic search direction is the global minimizer of $q(x_0+p)$. The full-dimensional version of Karmarkar's 1984 linear programming algorithm is shown to be a collinear scaling method for minimizing Karmarkar's potential function g_K, where the denominator $1+\langle d,p\rangle$ of the conic function is chosen as the normalized linear program objective function and the Taylor series expansions of $g_K(x_0+p)$ and $q(x_0+p)$ agree to second order.

Key words. Karmarkar's algorithm, collinear scaling, conic approximation

AMS subject classifications. 65K05, 90C05, 90C30

1. Introduction.

The interior point linear programming method of Karmarkar [11] can be viewed as a method for unconstrained minimization of a particular nonlinear function, called the Karmarkar potential function. Bayer and Lagarias [2] showed that, after a fixed change of coordinates, the search direction of Karmarkar's algorithm is the Newton direction for this function. Various other relations of Karmarkar's method to nonlinear programming appear in Gill, et al. [8], Bayer and Lagarias [1], Nesterov and Nemirovsky [13], and Powell [14].

Davidon [3] presented a class of *collinear scaling algorithms* for unconstrained minimization that are invariant under projective transformations. Since an intrinsic feature of Karmarkar's algorithm is invariance under projective transformations, it is reasonable to expect that there is some interpretation of Karmarkar's algorithm in Davidon's framework. Here we show that the full-dimensional variant of Karmarkar's algorithm has a simple interpretation as a collinear scaling algorithm, namely, that its search direction is the minimizing direction for a natural conic approximation to the associated potential function.

This collinear scaling interpretation does not explain the nice properties, such as polynomiality, of Karmarkar's algorithm. In addition, it does not prescribe a choice of step size for Karmarkar's algorithm. Rather, it may be viewed in reverse—as supplying, for certain nonlinear functions, a good choice of denominator to use in the conic approximations underlying collinear scaling algorithms. Thus for every quasi-Newton method for minimizing the Karmarkar potential function, a collinear scaling method exists that is likely to be as good or better because a good denominator for the conic approximations is always available for free; see (12) below.

Section 2 describes conic functions and collinear scaling algorithms. Section 3 gives the main result, the interpretation of the Karmarkar search direction as a collinear scaling direction. Section 4 concludes with a brief discussion.

2. Conic functions and collinear scaling algorithms.

The collinear scaling algorithms proposed by Davidon [3] consist of approximating the nonlinear objective function $f(x)$ near a given point x_0 by a conic function that maps $\Re^n$ to $\Re$ and is of

* Received by the editors February 11, 1991; accepted for publication (in revised form) June 24, 1992.

† AT&T Bell Laboratories, Murray Hill, New Jersey 07974.

the form

$$q(x_0 + p) = f_0 + \frac{\langle g, p\rangle}{1 + \langle d, p\rangle} + \frac{1}{2}\frac{\langle p, Ap\rangle}{(1 + \langle d, p\rangle)^2}, \tag{1}$$

where p is interpreted as a perturbation, f_0 is a scalar, g and d are $n \times 1$ column vectors, A is an $n \times n$ matrix, and $\langle y, x\rangle$ denotes the Euclidean inner product of y and x. The function $1 + \langle d, p\rangle$ is called the *denominator* of $q(x_0 + p)$. The conic function (1) is said to be *cupped* if it has a unique minimizer p^* on the half-space $\{d \mid \langle d, p\rangle + 1 > 0\}$ and if all level sets of $q(x_0 + p)$ are convex on this domain. In this case, p^* is the solution of

$$(A + gd^T)p^* = -g. \tag{2}$$

The conic search direction $v_q(x_0)$ at x_0 for a cupped conic function (1) is

$$v_q(x_0) := p^*. \tag{3}$$

Let x_0 be the current approximation to an unconstrained minimizer of f. A *conic algorithm* is one that produces, by various methods, a conic approximation $q(x_0 + p)$ to the function $f(x_0 + p)$ and produces a new iterate $x_1 = x_0 + \lambda_q v_q$ that is defined by taking a suitable step λ_q along the conic search direction defined by (3). Newton's method is a special kind of conic algorithm in which the conic approximation to $f(x_0 + p)$ is taken to have a constant denominator of unity (i.e., $d = 0$) and in which g and A are chosen so that the Taylor series expansions of $q(x_0 + p)$ and $f(x_0 + p)$ about x_0 agree to second order.

Conic approximations to a nonlinear function offer more flexibility in approximation than do quadratic polynomials. The set of conic functions is closed under the subgroup of projective transformations that have the origin as a fixed point and that have a positive denominator at $p = 0$; these transformations are called *collinear scalings* by Davidon. It is possible to create conic algorithms that are formally invariant under such projective transformations. Such algorithms are called *collinear scaling algorithms*. Newton's method is not a collinear scaling algorithm because it does not possess this projective invariance property.

Davidon [3] defined a class of collinear scaling algorithms in which conic models (1) are constructed by using data from the function f and its gradient at several previous iterations. Sorensen [17] described a superlinearly convergent collinear scaling algorithm. These algorithms were designed to have projective invariance analogous to the affine invariance property of quasi-Newton methods; see [5]. Some other conic algorithms are discussed in [4], [9], [15], and [18].

The key choice to make in a conic approximation is the vector d in the denominator, which can be used to match directions of rapid change of f, thereby giving a wider region of accurate approximation than does a quadratic polynomial. Once the denominator is chosen, a natural "infinitesimal" choice of A and g in (1) is to make the Taylor series of $f(x_0 + p)$ and $q(x_0 + p)$ agree through second order. On the other hand, there seems to be no natural infinitesimal criterion for choosing a denominator, i.e., a criterion based solely on the behavior of f in an arbitrarily small neighborhood of x_0. Indeed, the collinear scaling algorithms proposed so far use nonlocal information, e.g., data from previous iterations of the algorithm, to choose a denominator.

3. Karmarkar's algorithm and collinear scaling. We deal next with Karmarkar's algorithm for inequality-form linear programs, as presented in [2] and [6]. This algorithm is equivalent after an affine transformation to Karmarkar's [11] algorithm on standard-form linear programs; see [2].

Thus we are given the inequality-form linear programming problem:

$$(L): \begin{array}{ll} \text{minimize} & c^T x - c_0 \\ \text{subject to} & \langle a_j, x\rangle \geq b_j, \quad 1 \leq j \leq m. \end{array} \tag{4}$$

We assume that the polytope P of feasible points for the constraints (4) of (L) has a nonempty interior and is bounded, that a feasible point $x_0 \in \operatorname{Int}(P)$ is given, and that the objective function has $c \neq 0$ and is normalized to be zero at the optimum, i.e.,

$$c_0 = \min\{\langle c, x\rangle \mid x \in P\}.$$

In particular, x_0 is not optimal; hence $\langle c, x_0\rangle > c_0$. (More generally, the results of this section also apply to linear programs with unbounded feasible regions that are *quasi-bounded* as defined in [2].)

The *Karmarkar potential function* associated with (L) is

$$g_K(x) = m \log(\langle c, x\rangle - c_0) - \sum_{i=1}^{m} \log(\langle a_j, x\rangle - b_j). \tag{5}$$

Finding an optimal solution of (L) is essentially the same as minimizing $g_K(x)$. The function $g_K(x)$ is actually unbounded below on $\operatorname{Int}(P)$, but for any $\epsilon > 0$ there exists a value λ such that any point with $g_K(x) \leq \lambda$ is within an ϵ-neighborhood of some optimal point.

At any point $x_0 \in \operatorname{Int}(P)$ the *Karmarkar direction*, denoted by $v_K(x_0)$, is found as follows. We associate to the constraint set (4) the logarithmic barrier function

$$f_B(x) = -\sum_{j=1}^{m} \log(\langle a_j, x\rangle - b_j). \tag{6}$$

A point x_c is called the *center* of the constraint set (4) if

$$\nabla f_B(x_c) = -\sum_{j=1}^{m} \frac{a_j}{\langle a_j, x_c\rangle - b_j} = 0.$$

The center x_c exists and is unique if P has a nonempty interior and is bounded. This notion of center is due to Sonnevend [16].

The Karmarkar direction $v_K(x_0)$ is obtained by "centering" x_0 by a projective transformation and by pulling back the gradient of the (normalized) objective function in the projectively transformed coordinate system. To compute it we proceed in two steps. First, we translate x_0 to the origin by using the coordinate change $w = x - x_0$; second, we apply a projective transformation

$$\tilde{\Phi}(w) = \frac{Bw}{1 + \langle h, w\rangle}, \tag{7}$$

such that B is invertible, to obtain new coordinates

$$y := \Phi(x) = \frac{B(x - x_0)}{1 + \langle h, x - x_0\rangle}.$$

The projective transformation inverse to (7) is

$$\tilde{\Phi}^{-1}(y) = \frac{B^{-1}y}{1 - \langle h, B^{-1}y\rangle};$$

whence

$$x = \Phi^{-1}(y) := \tilde{\Phi}^{-1}(y) + x_0 = x_0 + \frac{B^{-1}y}{1 - \langle h, B^{-1}y\rangle}. \tag{8}$$

Applying (8) to (L) yields, after clearing of denominators, the transformed linear programming problem

$$\begin{aligned} \Phi(L): \quad & \text{minimize} \quad \langle c^*, y\rangle - c_0^* \\ & \text{subject to} \quad \langle a_j^*, y\rangle \geq b_j^*, \quad 1 \leq j \leq m, \end{aligned} \tag{9}$$

in which

$$\begin{aligned} a_j^* &= B^{-T}(a_j - (\langle a_j, x_0\rangle - b_j)h), & b_j^* &= b_j - \langle a_j, x_0\rangle, \\ c^* &= B^{-T}(c - (\langle c, x_0\rangle - c_0)h), & c_0^* &= c_0 - \langle c, x_0\rangle. \end{aligned}$$

In [12] and [2, Thm. 3.1], it is shown that there exists a projective transformation (7) such that the barrier function $f_B^*(y)$ for the transformed problem $\Phi(L)$ has

$$\nabla f_B^*(0) = 0, \tag{10a}$$

$$\nabla^2 f_B^*(0) = I, \tag{10b}$$

where I is the identity matrix. Condition (10a) says that $0 = \Phi(x_0)$ is the center of $\Phi(L)$. In the y-coordinate system the Karmarkar direction is $-c^*$. In the original coordinate system the Karmarkar direction is its pullback as a tangent vector under Φ^{-1}, which is

$$v_K(x_0) = -(B^TB)^{-1}(c - (\langle c, x_0\rangle - c_0)h). \tag{11}$$

The ray determined by this vector is independent of the choice of Φ used to obtain (10).

Now we turn to the collinear scaling interpretation. Since the Karmarkar potential function is unbounded below as one approaches the hyperplane $\{x \mid \langle c, x\rangle = c_0\}$ on which the optimum of (L) is achieved, a natural choice of conic approximation to $g_K(x)$ is one whose denominator vanishes on this hyperplane. The *Karmarkar conic approximation* $q_K(x_0 + p)$ to $g_K(x)$ at $x_0 \in \operatorname{Int}(P)$ has denominator

$$1 + \langle d, p\rangle := \frac{\langle c, x_0 + p\rangle - c_0}{\langle c, x_0\rangle - c_0}, \tag{12}$$

so that $d = \mu c$, where $\mu = 1/(\langle c, x_0\rangle - c_0)$. The remaining elements of q_K are uniquely determined by requiring that its Taylor series expansion about x_0 agree to second order with that of the Karmarkar potential function $g_K(x_0 + p)$. One has

$$q_K(x_0 + p) = g_K(x_0) + \frac{\langle g_0, p\rangle}{1 + \langle d, p\rangle} + \frac{1}{2}\frac{\langle p, Ap\rangle}{(1 + \langle d, p\rangle)^2}, \tag{13a}$$

where

$$g_0 := \nabla g_K(x_0), \tag{13b}$$

$$A := \nabla^2 g_K(x_0) + g_0 d^T + d g_0^T. \tag{13c}$$

Our main observation is the following result.

THEOREM 3.1. *For the inequality-form linear program* (L), *the Karmarkar conic approximation* $q_K(x_0+p)$ *to the potential function* g_K *at* x_0 *is a cupped conic function with a unique minimizer* p^* *within the open half-space* $\{p \mid \langle c, x_0+p\rangle > c_0\}$. *The conic search direction* $v_q(x_0) = p^*$ *is the same as the full-dimensional Karmarkar algorithm search direction* $v_K(x_0)$.

Proof. This is a computation. One method of proof is to proceed explicitly by calculating Φ directly as in [6]. Once Φ is known, (11) gives

$$B^T B v_K(x_0) = -(c - (\langle c, x_0\rangle - c_0)h).$$

One can then check whether $v_K(x_0)$ satisfies (2) up to multiplication by a scalar, after substituting the values (12) and (13) arising from the Karmarkar conic approximation.

A second method of proof, which we follow here, derives the theorem from the main result of [2]. First, observe that both $v_K(x_0)$ and $v_q(x_0)$ are invariant under all projective transformations *admissible* in the sense of [12], i.e., their denominators are nonzero everywhere on Int(P). This holds for $v_K(x_0)$ by [2, Thm. 2.6]. It holds for $v_q(x_0)$ because a projective transformation Φ maps normalized objective functions to normalized objective functions, maps the potential function of (L) to that of $\Phi(L)$, and preserves power series expansions to any order.

By a translation we may suppose without loss of generality that $x_0 = 0$. Now consider the projective transformation

$$y = \Phi(x) = \frac{x}{1 - \langle c, x\rangle / c_0}.$$

This is an admissible projective transformation, and the transformed polytope $\Phi(P)$ is unbounded. From [2], under this change of variable the Karmarkar direction becomes the Newton direction at $y = 0$ for the logarithmic barrier function

$$f_B^*(y) = -\sum_{j=1}^{m} \log(\langle a_j^*, y_j\rangle - b_j^*) \tag{14}$$

of the transformed constraints (9). Such a logarithmic barrier function is strictly convex, and hence the quadratic approximation

$$f_B^*(0) + \langle \nabla f_B^*(0), y\rangle + \tfrac{1}{2}\langle y, \nabla^2 f_B^*(0) y\rangle \tag{15}$$

to $f_B^*(y)$ has a unique minimizer y^{**}, and y^{**} is the Newton step for $f_B^*(y)$ at 0, which is the transformed Karmarkar direction.

Now we directly calculate that the transformed Karmarkar conic approximation $q_K^*(y)$ is a quadratic polynomial:

$$\begin{aligned} q_K^*(y) &:= q_K(\Phi^{-1}(y)) \\ &= f_0 + \langle g_0, y\rangle + \tfrac{1}{2}\langle y, Ay\rangle, \end{aligned} \tag{16}$$

with g_0 and A given by (13) since $d = 0$ from (12). Theorem 3.1 of [2] asserts that the transformed Karmarkar potential function is just $f_B^*(y)$ in (14) up to an additive constant. Since (16) approximates the transformed potential function $g_K(\Phi^{-1}(y))$ up to second order, it must agree with (15) except for the constant term. Thus y^{**} is a global minimizer for $q_K^*(y)$. Note also that this implies that its inverse image $g_K(x)$ under Φ^{-1} has a global minimizer on the inverse image

$$\Phi^{-1}(\{y \mid \langle c^*, y\rangle > c_0^*\}) \equiv \{p \mid \langle c, x_0 + p\rangle > c_0\}. \tag{17}$$

Hence $q(x_0 + p)$ is a cupped conic function with minimizer p^* in the region (17).

Thus the transformed conic direction matches the transformed Karmarkar direction, and so the directions $v_K(x_0)$ and $v_q(x_0)$ agree. □

4. Discussion. Subsequent developments motivated by Karmarkar's algorithm have focused attention on minimizing various members of the general class of potential functions

$$\sum_{j=1}^{m} d_j \log(\langle a_j, x\rangle - b_j), \tag{18}$$

where the coefficients d_j may be positive or negative and $\{\langle a_j, x\rangle - b_j\}$ are arbitrary linear forms on $\Re^n$. This class of functions includes logarithmic barrier functions, Karmarkar's potential function, the potential functions of Iri and Imai [10] and of Freund [7], the primal–dual potential function of Ye [20], and also some intrinsically nonconvex functions.

When all coefficients $\{d_j\}$ are negative, the function (18) is convex and has an interior minimizer, which can be found by a Newton-type method, as in [13] and [19]. When exactly one d_j is positive, (18) is a Karmarkar-type potential function; a good choice of conic approximation is obtained by choosing the denominator $\langle a_j, x\rangle - b_j$ corresponding to the positive d_j, as in §3. When two or more values of d_j are positive, there is no longer a natural choice of denominator for a conic approximation. One reasonable possibility is to choose as denominator a linear form $\langle a_j, x\rangle - b_j$ from among the indices j for which $d_j > 0$, such that the hyperplane $\{x \mid \langle a_j, x\rangle = b_j\}$ contains a closest point to the current iterate x_0.

Acknowledgments. I am indebted to M. H. Wright and the referees for their careful reading of the paper and for valuable suggestions.

REFERENCES

[1] D. A. Bayer and J. C. Lagarias, *The nonlinear geometry of linear programming*, I *and* II, Trans. Amer. Math. Soc., 314 (1989), pp. 499–526 and 527–581.

[2] ———, *Karmarkar's algorithm and Newton's method*, Math. Programming, 50 (1991), pp. 291–330.

[3] W. C. Davidon, *Conic approximations and collinear scalings for minimizers*, SIAM J. Numer. Anal., 17 (1980), pp. 268–281.

[4] ———, *Conjugate directions for conic functions*, in Nonlinear Optimization 1981, M. J. D. Powell, ed., Academic Press, New York, 1982, pp. 23–28.

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

[6] R. M. Freund, *An analog of Karmarkar's algorithm for inequality constrained linear programs, with a "new" class of projective transformation for centering a polytope*, Oper. Res. Lett., 7 (1988), pp. 9–14.

[7] ———, *Projective transformations for interior-point algorithms and a superlinearly convergent algorithm for the w-center problem*, Math. Programming, to appear.

[8] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright, *On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method*, Math. Programming, 36 (1986), pp. 183–209.

[9] H. Gourgeon and J. Nocedal, *A conic algorithm for minimization*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 253–267.

[10] M. Iri and H. Imai, *A multiplicative barrier function method for linear programming*, Algorithmica, 1 (1986), pp. 455–482.

[11] N. K. Karmarkar, *A new polynomial time algorithm for linear programming*, Combinatorica, 2 (1984), pp. 373–395.

[12] J. C. Lagarias, *The nonlinear geometry of linear programming* III: *Projective Legendre transform coordinates and Hilbert geometry*, Trans. Amer. Math. Soc., 320 (1990), pp. 193–225.

[13] J. E. Nesterov and A. S. Nemirovsky, *Self-Concordant Functions and Polynomial-Time Methods in Convex Programming*, preprint, 1989.

[14] M. J. D. Powell, *Karmarkar's algorithm: A view from nonlinear programming*, IMA Bull., 26 (1990), pp. 165–181.

[15] R. B. Schnabel, *Conic methods for unconstrained optimization and tensor methods for nonlinear equations*, in Mathematical Programming: The State of the Art, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 417–438.

[16] G. Sonnevend, *An "analytic center" for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming*, Lecture Notes in Control and Information Science 84, Springer-Verlag, New York, 1986, pp. 866–876.

[17] D. C. Sorensen, *The q-superlinear convergence of a collinear scaling algorithm for unconstrained optimization*, SIAM J. Numer. Anal., 17 (1980), pp. 84–114.

[18] ———, *Collinear scaling and sequential estimation in sparse optimization*, Math. Programming Study, 18 (1982), pp. 135–159.

[19] P. M. Vaidya, *A locally well-behaved potential function and a simple Newton-type method for finding the center of a polytope*, in Progress in Mathematical Programming: Interior-Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 79–90.

[20] Y. Ye, *An $O(n^3L)$ potential reduction algorithm for linear programming*, in Mathematical Developments Arising From Linear Programming, J. C. Lagarias and M. J. Todd, eds., American Mathematical Society, Providence, RI, 1990, pp. 91–107.

SIAM J. OPTIMIZATION
Vol. 3, No. 3, pp. 637–653, August 1993

012

AUTOMATIC COLUMN SCALING STRATEGIES FOR QUASI-NEWTON METHODS*

MARUCHA LALEE† AND JORGE NOCEDAL‡

Abstract. A class of algorithms is described for unconstrained optimization, based on the BFGS update formula, which includes an automatic column scaling strategy. The new algorithms generalize the method of Powell [*Math. Programming*, 38 (1987), pp. 29–46]. Conditions are given on the scaling strategies that guarantee global and superlinear convergence on convex problems.

Key words. quasi-Newton methods, minimization, nonlinear optimization, scaling

AMS subject classifications. 65, 49

1. Introduction.

Consider the unconstrained optimization problem

$$\min_{x \in \mathbf{R}^n} f(x), \tag{1.1}$$

where f is a nonlinear differentiable function. This problem is often solved by quasi-Newton methods with inexact line searches. At the beginning of each iteration k, a symmetric and positive definite matrix B_k and an estimate of the solution vector x_k are available. The new iterate x_{k+1} is computed by the following two equations,

$$d_k = -B_k^{-1} g_k, \tag{1.2}$$

$$x_{k+1} = x_k + \lambda_k d_k, \qquad k \geq 1, \tag{1.3}$$

where $g_k = g(x_k)$ is the gradient of the objective function at x_k, and λ_k is a steplength parameter. In this paper we assume that λ_k satisfies the Wolfe conditions

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \alpha \lambda_k g_k^T d_k, \tag{1.4}$$

$$g(x_k + \lambda_k d_k)^T d_k \geq \beta g_k^T d_k, \tag{1.5}$$

where $0 < \alpha < \frac{1}{2}$ and $\alpha < \beta < 1$.

Before starting the next iteration of a quasi-Newton method, B_k is updated to B_{k+1} using an updating formula which normally involves B_k, s_k, and y_k, where s_k and y_k are defined as

$$s_k = x_{k+1} - x_k,$$
$$y_k = g_{k+1} - g_k.$$

The particular updating formula studied in this paper is the BFGS formula, which is known to be very effective. It was discovered independently by Broyden [1], Fletcher [6], Goldfarb [7], and Shanno [13], and is given by

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \tag{1.6}$$

* Received by the editors October 7, 1991; accepted for publication (in revised form) June 9, 1992. This work was supported by National Science Foundation grant CCR-9101359 and by Department of Energy grant DE-FG02-87ER25047.

† Department of Industrial Engineering and Management Science, Northwestern University, Evanston, Illinois 60208.

‡ Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, Illinois 60208.

By applying the Sherman–Morrison–Woodbury formula to (1.6), one can express B_{k+1}^{-1} directly in terms of B_k^{-1}, s_k, and y_k resulting in an inverse form of the BFGS update.

Powell [11], [12] observed that the BFGS method can take a large number of iterations to find the minimum of the quadratic function $f(x) = \frac{1}{2}x^T Gx$, where G is a symmetric and positive definite matrix, if the eigenvalues of the initial Hessian approximation B_1 are much larger than those of G. However, if the eigenvalues of B_1 are approximately equal to or less than those of G, then the BFGS method performs very well.

For this reason Powell proposed that some kind of scaling be introduced to automatically improve the magnitude of the matrix B_k with respect to that of G. He works with conjugate direction matrices Z_k, which satisfy $B_k^{-1} = Z_k Z_k^T$. His algorithm is based on the BFGS method, but since the update is modified, it represents a new quasi-Newton method. The following is a brief description of Powell's updating procedure.

Given $B_k^{-1} = Z_k Z_k^T$, one first updates Z_k to $\bar{Z}_k$ so that $\bar{Z}_k \bar{Z}_k^T$ equals the inverse BFGS update of B_k^{-1}. A scaling parameter $\sigma_k \geq 0$ is then computed. Each column of $\bar{Z}_k$ that is smaller than σ_k is scaled up so that its 2-norm is equal to σ_k. Any columns larger than or equal to σ_k remain unchanged. Note that this step is equivalent to postmultiplying $\bar{Z}_k$ by a diagonal matrix D_k whose ith diagonal element equals $\sigma_k / \|\bar{Z}_k e_i\|$ if the corresponding column is to be scaled, or equals 1 otherwise. Z_{k+1} is set to $\bar{Z}_k D_k$ and the next inverse Hessian approximation, B_{k+1}^{-1}, is defined by $Z_{k+1} Z_{k+1}^T$.

Powell showed that if implemented properly, his algorithm possesses quadratic termination. Our numerical tests with this algorithm indicate that in some cases the improvement over the BFGS method is substantial. However, Siegel [15] has given an example that shows that, for a certain choice of σ_k, the algorithm is only linearly convergent when applied to a two-variable quadratic objective function.

This paper, therefore, investigates whether it is possible to select the scaling parameter σ_k so that the superlinear convergence property of the BFGS method is preserved. We choose to do our study on scaling algorithms based on the direct form of the BFGS update, with the intention of later generalizing the results to those based on the inverse form, such as Powell's. The prototype for the class of algorithms that we wish to consider is described in §2. It encompasses many scaling algorithms, based on the direct form, in which scaling down the columns is also allowed. Sections 3 and 4 discuss the global and superlinear convergence properties of these methods. An implementation that is superlinearly convergent for strictly convex problems is presented in §5, and we conclude with final remarks in §6.

Notation. Throughout the paper $\|x\|$ denotes the 2-norm of x, $\|B\|$ the corresponding induced matrix norm of B, and e_i the ith column of the identity matrix.

2. Description of the class of algorithms.

We now describe the prototype for the algorithms with column scaling based on the BFGS method. The description is based on the direct form of the method; however, it can easily be transformed to an analogous prototype based on the inverse form.

ALGORITHM 2.1. Prototype for automatic column scaling BFGS algorithms.

(0) Choose a starting point x_1 and a nonsingular matrix V_1; set $k = 1$.

(1) Terminate if a stopping criterion is satisfied.

(2) Compute

$$d_k = -V_k^{-T} V_k^{-1} g_k,$$
$$x_{k+1} = x_k + \lambda_k d_k,$$

where λ_k is a steplength that satisfies the Wolfe conditions (1.4)–(1.5). (The stepsize $\lambda_k = 1$ is always tried first and is accepted if admissible.) Compute

$$s_k = x_{k+1} - x_k,$$
$$y_k = g_{k+1} - g_k.$$

(3) Update V_k to W_k so that $W_k W_k^T$ is the BFGS update of $V_k V_k^T$.

(4) Compute the scaling parameters $\sigma_k \geq 0$ and $\eta_k > 0$ such that $\sigma_k \leq \eta_k$. Let w_i represent the ith column of W_k. Construct $C_k = \text{diagonal}(c_1, c_2, \ldots, c_n)$, where c_i is given by

$$c_i = \begin{cases} \dfrac{\sigma_k}{\|w_i\|} & \text{if } \|w_i\| < \sigma_k, \\ \dfrac{\eta_k}{\|w_i\|} & \text{if } \|w_i\| > \eta_k, \\ 1 & \text{otherwise.} \end{cases} \tag{2.1}$$

Compute

$$V_{k+1} = W_k C_k.$$

(5) Set $k := k + 1$ and go to step (1).

To elaborate, each iteration of Algorithm 2.1 is of the form (1.2)–(1.3), with the Hessian approximation B_k taken as

$$B_1 = V_1 V_1^T; \quad B_k = V_k V_k^T = W_{k-1} C_{k-1}^2 W_{k-1}^T, \quad k > 1. \tag{2.2}$$

The update is performed directly on V_k so that the resulting matrix W_k is such that $W_k W_k^T$ is the BFGS update of $V_k V_k^T$. Before completing the iteration, the algorithm updates the scaling parameters σ_k and η_k, and scales appropriately any columns of W_k whose 2-norm falls below σ_k or above η_k, as described in step (4). We impose the restriction $\sigma_k \leq \eta_k$ so that the conditions in (2.1) are mutually exclusive.

As mentioned earlier, Algorithm 2.1 is based on the direct form of the BFGS method. However, it is easy to see that this framework can be modified to accommodate an algorithm based on the inverse form, such as Powell's, if we keep $Z_k = V_k^{-1}$ instead of V_k, keep $\bar{Z}_k = W_k^{-1}$ instead of W_k, and replace $\|w_i\|$ by $\|\bar{z}_i\|$, the norm of the ith column of $\bar{Z}_k$. Of course, $\bar{Z}_k$ is such that $\bar{Z}_k \bar{Z}_k^T$ is the *inverse* BFGS update of $Z_k Z_k^T$.

It will be shown that one has considerable freedom in choosing σ_k and η_k at every iteration, while still maintaining global and r-linear convergence for convex problems. However, as we will show in §4, to obtain superlinear convergence, it is necessary that the choice of these values be made carefully.

3. Global and r-linear convergence. In this section, we prove that Algorithm 2.1 with an appropriate choice of the scaling parameters is globally and r-linearly convergent on strictly convex objective functions.

Let $\text{tr}(B)$ and $\det(B)$ be the trace and the determinant of B, respectively. We begin with the following two preliminary technical lemmas.

LEMMA 3.1. *For any $n \times n$ matrices A and C, where C is diagonal,*

$$\text{tr}(ACA^T) = \text{tr}(AA^T) + \text{tr}[(C - I)(A^T A)]. \tag{3.1}$$

Proof. The proof is straightforward by observing that for any matrices A and B, $\operatorname{tr}(AB) = \operatorname{tr}(BA)$. Consequently,

$$\begin{aligned}\operatorname{tr}(ACA^T) &= \operatorname{tr}(CA^TA)\\ &= \operatorname{tr}(AA^T) + \operatorname{tr}(CA^TA) - \operatorname{tr}(A^TA).\end{aligned}$$

Equation (3.1) follows directly from the last equality. □

LEMMA 3.2. *Let $h(u) = \ln u - u$ for $u > 0$. Given positive constants δ_1 and δ_2, there exist constants δ_3 and δ_4 such that*

$$x \in (0, \delta_1] \quad \text{and} \quad y \in (0, x] \Rightarrow h(y) - h(x) \le \delta_3, \tag{3.2}$$

and

$$x \in [\delta_2, \infty) \quad \text{and} \quad y \in [x, \infty) \Rightarrow h(y) - h(x) \le \delta_4. \tag{3.3}$$

Proof. To show (3.2), we first note that $h(u)$ is strictly concave and its maximum occurs at $u = 1$. We consider separately the cases when $x \in (0, \min(\delta_1, 1))$ and when $x \in [\min(\delta_1, 1), \delta_1]$.

If $x \in (0, \min(\delta_1, 1))$, we conclude that for any $y \in (0, x]$,

$$h(y) - h(x) \le 0,$$

since $h(u)$ is strictly increasing for $0 < u \le 1$. On the other hand, if $x \in [\min(\delta_1, 1), \delta_1]$, then for any $y \in (0, x]$, we have

$$h(y) - h(x) \le h(\min(\delta_1, 1)) - h(\delta_1).$$

Thus (3.2) holds in either case with $\delta_3 = h(\min(\delta_1, 1)) - h(\delta_1)$.

A similar line of reasoning shows that (3.3) holds with $\delta_4 = h(\max(\delta_2, 1)) - h(\delta_2)$. □

Let $G(x)$ denote the Hessian matrix of f at x, and let $D(\bar{x}) = \{x \in \mathbf{R}^n : f(x) \le f(\bar{x})\}$ be the level set of f at $\bar{x}$. We now state the assumptions we make on the objective function f and the starting point x_1 in order to prove our convergence results.

ASSUMPTIONS 3.1.

(1) The objective function f is twice continuously differentiable.

(2) The starting point x_1 is such that the level set $D(x_1)$ is convex.

(3) There exist positive constants m and M such that for all $z \in \mathbf{R}^n$ and all $x \in D(x_1)$,

$$m\|z\|^2 \le z^T G(x) z \le M\|z\|^2.$$

These assumptions readily imply that f is strictly convex in $D(x_1)$, and that there is a unique minimizer x_* of f in $D(x_1)$. For any positive definite matrix B, we define the function

$$\psi(B) = \operatorname{tr}(B) - \ln(\det(B)), \tag{3.4}$$

which has been used by Byrd and Nocedal [4] and Griewank [9] in their analyses of quasi-Newton methods. Furthermore, define

$$\cos\theta_k = \frac{s_k^T B_k s_k}{\|s_k\|\,\|B_k s_k\|}, \tag{3.5}$$

so that θ_k is the angle between the search direction d_k and the steepest descent direction $-g_k$, and also define

$$q_k = \frac{s_k^T B_k s_k}{s_k^T s_k}. \tag{3.6}$$

We assume that the scaling parameters σ_k and η_k are bounded: for all k,

$$\sigma_k \le \sigma_{\max}, \qquad \eta_k \ge \eta_{\min}, \tag{3.7}$$

for some positive constants $\sigma_{\max}$ and $\eta_{\min}$. The following lemma provides the foundation for the proof of global and r-linear convergence. It generalizes a similar result given by Byrd and Nocedal [4, Thm. 2.1] for the (unscaled) BFGS method.

LEMMA 3.3. *Let x_1 be a starting point for which f satisfies Assumptions* 3.1, *and let B_1 be a positive definite starting Hessian approximation. Let $\{x_k\}$ be generated by Algorithm* 2.1 *with σ_k and η_k satisfying* (3.7), *then for any $p \in (0,1)$, there exists a constant β_1 such that, for any $k > 1$, the relation*

$$\cos\theta_j \ge \beta_1 \tag{3.8}$$

holds for at least $\lceil pk \rceil$ values of $j \in [1,k]$.

Proof. First we note that the symmetric matrices $B_k = V_k V_k^T = W_{k-1} C_{k-1}^2 W_{k-1}^T$ generated by the algorithm are positive definite, because the W_{k-1} are nonsingular as a consequence of the BFGS update, and the C_{k-1} are nonsingular by construction. Using the definition (3.4) of ψ, (2.2), and Lemma 3.1, we have

$$\begin{aligned}
\psi(B_{k+1}) &= \operatorname{tr}(B_{k+1}) - \ln(\det(B_{k+1})) \\
&= \operatorname{tr}(W_k C_k^2 W_k^T) - \ln(\det(W_k C_k^2 W_k^T)) \\
&= \operatorname{tr}(W_k W_k^T) + \operatorname{tr}((C_k^2 - I) W_k^T W_k) - \ln(\det(W_k W_k^T)) - \ln\det(C_k^2) \\
&= \psi(W_k W_k^T) + \operatorname{tr}((C_k^2 - I) W_k^T W_k) - \ln\det(C_k^2) \\
&= \psi(W_k W_k^T) + \sum_{i=1}^{n} \left[(c_i^2 - 1)\|w_i\|^2 - \ln c_i^2 \right],
\end{aligned}$$

where w_i is the ith column of W_k.

Define the set of indices of the columns of W_k to be scaled up, and the set of indices of the columns to be scaled down as

$$I_k = \{ i \in [1,n] : \|w_i\| < \sigma_k \} \tag{3.9}$$

and

$$J_k = \{ i \in [1,n] : \|w_i\| > \eta_k \}. \tag{3.10}$$

Therefore, by (2.1),

$$\begin{aligned}
\psi(B_{k+1}) &= \psi(W_k W_k^T) + \sum_{i \in I_k} \left[\left(\frac{\sigma_k^2}{\|w_i\|^2} - 1 \right) \|w_i\|^2 - \ln \frac{\sigma_k^2}{\|w_i\|^2} \right] \\
&\quad + \sum_{i \in J_k} \left[\left(\frac{\eta_k^2}{\|w_i\|^2} - 1 \right) \|w_i\|^2 - \ln \frac{\eta_k^2}{\|w_i\|^2} \right] \\
&= \psi(W_k W_k^T) + \sum_{i \in I_k} \left[(\ln \|w_i\|^2 - \|w_i\|^2) - (\ln \sigma_k^2 - \sigma_k^2) \right] \\
&\quad + \sum_{i \in J_k} \left[(\ln \|w_i\|^2 - \|w_i\|^2) - (\ln \eta_k^2 - \eta_k^2) \right].
\end{aligned}$$

We will now invoke Lemma 3.2 with $\delta_1 = \sigma_{\max}$ and $\delta_2 = \eta_{\min}$. Since $\|w_i\| \leq \sigma_k$ for $i \in I_k$, whereas $\|w_i\| \geq \eta_k$ for $i \in J_k$, we can therefore apply (3.2) to each term of the first summation, and (3.3) to each term of the second summation to obtain

$$\psi(B_{k+1}) \leq \psi(W_k W_k^T) + n\delta_3 + n\delta_4 \tag{3.11}$$

for the constants δ_3 and δ_4 given by the lemma.

Step (3) of Algorithm 2.1 indicates that the matrix $W_k W_k^T$ is the BFGS update of B_k. Therefore, as derived in Byrd and Nocedal [4, eq. (2.9)],

(3.12)

$$\psi(W_k W_k^T) = \psi(B_k) + [M_k - \ln m_k - 1] + \left[1 - \frac{q_k}{\cos^2\theta_k} + \ln\frac{q_k}{\cos^2\theta_k}\right] + \ln\cos^2\theta_k,$$

where

$$M_k = \frac{y_k^T y_k}{y_k^T s_k}, \qquad m_k = \frac{y_k^T s_k}{s_k^T s_k}.$$

It has also been shown by Byrd and Nocedal [4] that Assumptions 3.1 imply that $(M_k - \ln m_k - 1)$ is bounded from above by a positive constant, say δ_5. Moreover, the term in the second pair of brackets in (3.12) is nonpositive. We could use this, and the fact that $\psi(B_{k+1}) > 0$ to show that $\cos\theta_k$ cannot converge to 0. Instead, we establish the stronger results of this lemma, which will readily imply r-linear convergence. From (3.11) and (3.12) we have

$$\begin{aligned}\psi(B_{k+1}) &\leq \psi(B_k) + \delta_5 + n(\delta_3 + \delta_4)\\ &\quad + \left(1 - \frac{q_k}{\cos^2\theta_k} + \ln\frac{q_k}{\cos^2\theta_k}\right) + \ln\cos^2\theta_k\\ &\leq \psi(B_k) + \delta_6 - \alpha_k,\end{aligned}$$

where $\delta_6 = \delta_5 + n(\delta_3 + \delta_4)$ and

$$\alpha_k = -\left[\left(1 - \frac{q_k}{\cos^2\theta_k} + \ln\frac{q_k}{\cos^2\theta_k}\right) + \ln\cos^2\theta_k\right] \geq 0. \tag{3.13}$$

Therefore,

$$0 < \psi(B_{k+1}) \leq \psi(B_1) + \delta_6 k - \sum_{j=1}^{k} \alpha_j,$$

and hence

$$\frac{1}{k}\sum_{j=1}^{k} \alpha_j < \psi(B_1) + \delta_6. \tag{3.14}$$

Choose $p \in (0,1)$ and define S_k to be the set consisting of the indices corresponding to the $\lceil pk \rceil$ smallest values of α_j, for $j \leq k$. Let $\bar{\alpha}_k = \max_{j \in S_k}\{\alpha_j\}$, then

$$\frac{1}{k}\sum_{j=1}^{k} \alpha_j \geq \frac{1}{k}\left[\bar{\alpha}_k + \sum_{\substack{j=1\\ j \notin S_k}}^{k} \alpha_j\right] \geq \bar{\alpha}_k(1-p). \tag{3.15}$$

Therefore, combining (3.14) and (3.15), we have that for all $j \in S_k$,

$$\alpha_j \le \bar{\alpha}_k < \frac{1}{(1-p)}(\psi(B_1) + \delta_6) \equiv \beta_0.$$

Since (3.13) implies that $-\ln\cos^2\theta_j \le \alpha_j$, it follows that for all $j \in S_k$,

$$\ln\cos^2\theta_j > -\beta_0$$

or

$$\cos\theta_j > e^{-\beta_0/2} \equiv \beta_1. \qquad \square$$

We are now ready to state the global and r-linear convergence theorem for Algorithm 2.1. It should be noted that the scaling parameters σ_k and η_k are only assumed to be bounded—the former from above and the latter away from zero.

THEOREM 3.4. *Let x_1 be a starting point for which f satisfies Assumptions* 3.1, *and let B_1 be a positive definite starting Hessian approximation. Then Algorithm* 2.1, *with σ_k and η_k satisfying* (3.7), *generates a sequence $\{x_k\}$ that converges to x_*; moreover,*

$$\sum_{k=1}^{\infty} \|x_k - x_*\| < \infty$$

and

$$f_{k+1} - f_* \le r^k(f_1 - f_*)$$

for some constant $r \in [0,1)$.

Proof. The line search conditions (1.4)–(1.5) and the assumptions on f imply that (see, for example, Byrd, Nocedal, and Yuan [3, eq. (2.13)]),

$$f_{k+1} - f_* \le \left[1 - \delta_7 \cos^2\theta_k\right](f_k - f_*), \tag{3.16}$$

where $\delta_7 = \alpha m(1-\beta)/M$. Lemma 3.3 shows that $\cos\theta_j \ge \beta_1$ for at least $\lceil pk \rceil$ values of $j \in [1,k]$. Let j_k be the largest of these $\lceil pk \rceil$ indices. Thus (3.16) implies that

$$\begin{aligned} f_{k+1} - f_* &\le \left[1 - \delta_7\beta_1^2\right](f_{j_k} - f_*) \\ &\le \left[1 - \delta_7\beta_1^2\right]^{pk}(f_1 - f_*) \\ &= r^k(f_1 - f_*), \end{aligned} \tag{3.17}$$

where $r = \left[1 - \delta_7\beta_1^2\right]^p$. The assumptions on f also imply that

$$\tfrac{1}{2}m\|x_k - x_*\|^2 \le f_k - f_*. \tag{3.18}$$

Therefore, combining (3.18) and (3.17), we obtain

$$\begin{aligned} \sum_{k=1}^{\infty} \|x_k - x_*\| &\le \left[\frac{2}{m}\right]^{\frac{1}{2}} \sum_{k=1}^{\infty} (f_k - f_*)^{\frac{1}{2}} \\ &\le \left[\frac{2(f_1 - f_*)}{m}\right]^{\frac{1}{2}} \sum_{k=0}^{\infty} (r^{\frac{1}{2}})^k \\ &< \infty. \qquad \square \end{aligned}$$

4. Superlinear convergence. First, we define the following quantities to be used in this section:

$$\tilde{B}_k = G_*^{-\frac{1}{2}} B_k G_*^{-\frac{1}{2}}, \qquad \tilde{W}_k = G_*^{-\frac{1}{2}} W_k, \tag{4.1}$$

$$\tilde{s}_k = G_*^{\frac{1}{2}} s_k, \qquad \tilde{y}_k = G_*^{-\frac{1}{2}} y_k, \tag{4.2}$$

$$\tilde{M}_k = \frac{\tilde{y}_k^T \tilde{y}_k}{\tilde{y}_k^T \tilde{s}_k}, \qquad \tilde{m}_k = \frac{\tilde{y}_k^T \tilde{s}_k}{\tilde{s}_k^T \tilde{s}_k}, \tag{4.3}$$

$$\tilde{q}_k = \frac{\tilde{s}_k^T \tilde{B}_k \tilde{s}_k}{\tilde{s}_k^T \tilde{s}_k}, \qquad \cos\tilde{\theta}_k = \frac{\tilde{s}_k^T \tilde{B}_k \tilde{s}_k}{\|\tilde{s}_k\|\,\|\tilde{B}_k \tilde{s}_k\|}, \tag{4.4}$$

where G_* is the Hessian of f at the minimizer x_*.

Byrd, Liu, and Nocedal [2, Lemma 3.2] have shown that the limiting behavior of $\tilde{q}_k$ and $\cos\tilde{\theta}_k$ is enough to characterize the asymptotic rate of convergence of a sequence of iterates $\{x_k\}$ generated by a quasi-Newton algorithm. Their result, which can be seen as a restatement of the Dennis and Moré [5] characterization, is reproduced in the following lemma.

LEMMA 4.1. *Suppose that the sequence of iterates* $\{x_k\}$ *is generated by algorithm* (1.2)–(1.3) *using some positive definite sequence* $\{B_k\}$, *and that* $\lambda_k = 1$ *whenever this value satisfies Wolfe conditions* (1.4)–(1.5). *If* $x_k \to x_*$ *then the following two conditions are equivalent:*

(i) *The steplength* $\lambda_k = 1$ *satisfies conditions* (1.4)–(1.5) *for all large* k *and the rate of convergence is superlinear.*

(ii)

$$\lim_{k\to\infty} \cos\tilde{\theta}_k = \lim_{k\to\infty} \tilde{q}_k = 1. \tag{4.5}$$

The next theorem specifies conditions on the scaling parameters σ_k and η_k that allow $\tilde{q}_k$ and $\cos\tilde{\theta}_k$, produced by Algorithm 2.1, to exhibit the desirable limiting behavior of Lemma 4.1. Such conditions involve the following apparently cumbersome quantities:

(4.6)

$$\gamma_k = \sum_{i\in I_k}\left[\left(\ln\|G_*^{-\frac{1}{2}} w_i\|^2 - \|G_*^{-\frac{1}{2}} w_i\|^2\right) - \left(\ln\sigma_k^2 \frac{\|G_*^{-\frac{1}{2}} w_i\|^2}{\|w_i\|^2} - \sigma_k^2 \frac{\|G_*^{-\frac{1}{2}} w_i\|^2}{\|w_i\|^2}\right)\right],$$

and

(4.7)

$$\mu_k = \sum_{i\in J_k}\left[\left(\ln\|G_*^{-\frac{1}{2}} w_i\|^2 - \|G_*^{-\frac{1}{2}} w_i\|^2\right) - \left(\ln\eta_k^2 \frac{\|G_*^{-\frac{1}{2}} w_i\|^2}{\|w_i\|^2} - \eta_k^2 \frac{\|G_*^{-\frac{1}{2}} w_i\|^2}{\|w_i\|^2}\right)\right],$$

and whether or not they sum finitely. Note that γ_k and μ_k need not be positive. Recall that the sets I_k and J_k defined by (3.9) and (3.10) contain the indices of the columns that are scaled up and the indices of the columns that are scaled down at iteration k. We are now ready to state the theorem.

THEOREM 4.1. *Let f, x_1, B_1, σ_k, and η_k satisfy the assumptions in Theorem 3.4. In addition, assume that G is Lipschitz continuous at x_*. Let $\{x_k\} \to x_*$ be generated by Algorithm* 2.1; *then if*

$$\sum_{k=1}^{\infty} \gamma_k < \infty, \tag{4.8}$$

$$\sum_{k=1}^{\infty} \mu_k < \infty, \tag{4.9}$$

the iterates converge superlinearly.

Proof. From the definition (3.4) of ψ and from (2.2), (3.1), and (4.1), we have

$$\begin{aligned}\psi(\tilde{B}_{k+1}) &= \operatorname{tr}(G_*^{-\frac{1}{2}} W_k C_k^2 W_k^T G_*^{-\frac{1}{2}}) - \ln\det(G_*^{-\frac{1}{2}} W_k C_k^2 W_k^T G_*^{-\frac{1}{2}}) \\ &= \operatorname{tr}(\tilde{W}_k C_k^2 \tilde{W}_k^T) - \ln\det(\tilde{W}_k \tilde{W}_k^T) - \ln\det(C_k^2) \\ &= \psi(\tilde{W}_k \tilde{W}_k^T) + \sum_{i=1}^{n} \left[(c_i^2 - 1)\|G_*^{-\frac{1}{2}} w_i\|^2 - \ln c_i^2 \right].\end{aligned}$$

Then, by the definition (2.1) of c_i,

(4.10)
$$\begin{aligned}\psi(\tilde{B}_{k+1}) &= \psi(\tilde{W}_k \tilde{W}_k^T) + \sum_{i\in I_k} \left[\left(\frac{\sigma_k^2}{\|w_i\|^2} - 1 \right) \|G_*^{-\frac{1}{2}} w_i\|^2 - \ln \frac{\sigma_k^2}{\|w_i\|^2} \right] \\ &\quad + \sum_{i\in J_k} \left[\left(\frac{\eta_k^2}{\|w_i\|^2} - 1 \right) \|G_*^{-\frac{1}{2}} w_i\|^2 - \ln \frac{\eta_k^2}{\|w_i\|^2} \right] \\ &= \psi(\tilde{W}_k \tilde{W}_k^T) + \sum_{i\in I_k} \left[\sigma_k^2 \frac{\|G_*^{-\frac{1}{2}} w_i\|^2}{\|w_i\|^2} - \|G_*^{-\frac{1}{2}} w_i\|^2 \right. \\ &\qquad \left. - \ln \sigma_k^2 \frac{\|G_*^{-\frac{1}{2}} w_i\|^2}{\|w_i\|^2} + \ln \|G_*^{-\frac{1}{2}} w_i\|^2 \right] \\ &\quad + \sum_{i\in J_k} \left[\eta_k^2 \frac{\|G_*^{-\frac{1}{2}} w_i\|^2}{\|w_i\|^2} - \|G_*^{-\frac{1}{2}} w_i\|^2 - \ln \eta_k^2 \frac{\|G_*^{-\frac{1}{2}} w_i\|^2}{\|w_i\|^2} + \ln \|G_*^{-\frac{1}{2}} w_i\|^2 \right] \\ &= \psi(\tilde{W}_k \tilde{W}_k^T) + \gamma_k + \mu_k.\end{aligned}$$

Since $W_k W_k^T$ is the matrix obtained by updating B_k using the BFGS formula, which is invariant under the transformation (4.1)–(4.4), we have as in (3.12),

(4.11)
$$\psi(\tilde{W}_k \tilde{W}_k^T) = \psi(\tilde{B}_k) + (\tilde{M}_k - \ln \tilde{m}_k - 1) + \left(1 - \frac{\tilde{q}_k}{\cos^2 \tilde{\theta}_k} + \ln \frac{\tilde{q}_k}{\cos^2 \tilde{\theta}_k} \right) + \ln \cos^2 \tilde{\theta}_k.$$

Therefore, using (4.11) in (4.10), we have

$$\begin{aligned}\psi(\tilde{B}_{k+1}) &= \psi(\tilde{B}_k) + (\tilde{M}_k - \ln \tilde{m}_k - 1) + \left(1 - \frac{\tilde{q}_k}{\cos^2\tilde{\theta}_k} + \ln\frac{\tilde{q}_k}{\cos^2\tilde{\theta}_k}\right) + \ln\cos^2\tilde{\theta}_k \\ &\quad + \gamma_k + \mu_k \\ &= \psi(\tilde{B}_1) + \sum_{j=1}^{k}(\tilde{M}_j - \ln \tilde{m}_j - 1) \\ &\quad + \sum_{j=1}^{k}\left[\left(1 - \frac{\tilde{q}_j}{\cos^2\tilde{\theta}_j} + \ln\frac{\tilde{q}_j}{\cos^2\tilde{\theta}_j}\right) + \ln\cos^2\tilde{\theta}_j\right] + \sum_{j=1}^{k}\gamma_j + \sum_{j=1}^{k}\mu_j. \end{aligned} \tag{4.12}$$

By Theorem 3.4, we know that the iterates converge to x_* r-linearly. Using this and the Lipschitz continuity of G at x_*, it is not difficult to show (see Byrd and Nocedal [4, p. 735]) that

$$\sum_{j=1}^{\infty}(\tilde{M}_j - \ln \tilde{m}_j - 1) < \infty. \tag{4.13}$$

Moreover, the hypothesis of the theorem guarantees that the last two summations in (4.12) are bounded above. Therefore, in order for $\psi(\tilde{B}_{k+1})$ to remain positive as $k \to \infty$, the sum of the nonpositive terms in the square brackets must also be bounded. This can only be true if

$$\lim_{k\to\infty}\left(1 - \frac{\tilde{q}_k}{\cos^2\tilde{\theta}_k} + \ln\frac{\tilde{q}_k}{\cos^2\tilde{\theta}_k}\right) = \lim_{k\to\infty}\ln\cos^2\tilde{\theta}_k = 0,$$

which implies that both $\tilde{q}_k$ and $\cos\tilde{\theta}_k \to 1$. Hence, superlinear convergence follows from Lemma 4.1. □

Next we examine the conditions under which relations (4.8) and (4.9) hold. We give two sets of such conditions in the following two lemmas. However, before doing so, we analyze γ_k and μ_k. Applying the Mean Value Theorem to the function $h(u) = \ln(u) - u$, it follows that

$$(\ln x - x) - (\ln y - y) = \left(\frac{1}{\kappa} - 1\right)(x - y) \tag{4.14}$$

for some scalar κ between x and y. Using this in (4.6), we have that

$$\gamma_k = \sum_{i\in I_k}\left(\frac{1}{\xi_i} - 1\right)\left[\|G_*^{-\frac{1}{2}}w_i\|^2 - \sigma_k^2\frac{\|G_*^{-\frac{1}{2}}w_i\|^2}{\|w_i\|^2}\right] \tag{4.15}$$

for some scalar ξ_i such that

$$\|G_*^{-\frac{1}{2}}w_i\|^2 \leq \xi_i \leq \sigma_k^2\frac{\|G_*^{-\frac{1}{2}}w_i\|^2}{\|w_i\|^2}. \tag{4.16}$$

Similarly, using (4.14) in (4.7), we have that

$$\mu_k = \sum_{i\in J_k}\left(\frac{1}{\zeta_i} - 1\right)\left[\|G_*^{-\frac{1}{2}}w_i\|^2 - \eta_k^2\frac{\|G_*^{-\frac{1}{2}}w_i\|^2}{\|w_i\|^2}\right] \tag{4.17}$$

for some scalar ζ_i such that

$$\eta_k^2 \frac{\|G_*^{-\frac{1}{2}} w_i\|^2}{\|w_i\|^2} \le \zeta_i \le \|G_*^{-\frac{1}{2}} w_i\|^2. \tag{4.18}$$

LEMMA 4.2. *If Algorithm* 2.1 *is applied with* f, x_1, B_1, σ_k, *and* η_k *satisfying the assumptions in Theorem* 3.4, *then*

$$\sigma_k^2 \le \lambda_{\min} \quad \textit{for all large } k \quad \Rightarrow \sum_{k=1}^{\infty} \gamma_k < \infty \tag{4.19}$$

and

$$\eta_k^2 \ge \lambda_{\max} \quad \textit{for all large } k \quad \Rightarrow \sum_{k=1}^{\infty} \mu_k < \infty, \tag{4.20}$$

where $\lambda_{\min}$ *and* $\lambda_{\max}$ *are the smallest and the largest eigenvalues of* G_*.

Proof. The bound on σ_k^2 in (4.19), and the inequality (4.16), imply that the ξ_i in (4.15) are less than or equal to 1 for all large k. Consequently, consideration of the relation (4.15) shows that $\gamma_k \le 0$ for large k, since the term in the parentheses is nonnegative, while the term in the square brackets is nonpositive.

Similarly, the bound on η_k^2 in (4.20), and the inequality (4.18), imply that the ζ_i in (4.17) are greater than or equal to 1, and hence $\mu_k \le 0$ for all large k. Thus (4.8) and (4.9) follow immediately. □

This result is interesting because it relates σ_k and η_k to the curvature of the problem at the solution. In other words, if the smallest and the largest eigenvalues of the Hessian at the solution are known, we can design an algorithm that converges superlinearly for convex problems simply by ensuring that eventually the σ_k drop below $\lambda_{\min}$ and the η_k rise above $\lambda_{\max}$. In practice, however, the eigenvalues are not readily available. Nevertheless, Lemma 4.2 is still of theoretical interest, as we will use it later to show that a practical computation of the scaling parameters gives superlinear convergence.

To proceed with the analysis in the remainder of the paper, we define U_k to be the set of iteration numbers less than or equal to k in which at least one column is scaled up, i.e., $U_k = \{j \le k : I_j \ne \phi\}$. Similarly, we define D_k to be the set of iterations where scaling down occurs. It is clear from these definitions that $\sum_{k=1}^{\infty} \gamma_k = \sum_{k \in U_\infty} \gamma_k$ and $\sum_{k=1}^{\infty} \mu_k = \sum_{k \in D_\infty} \mu_k$.

LEMMA 4.3. *If Algorithm* 2.1 *is applied with* f, x_1, B_1, σ_k, *and* η_k *satisfying the assumptions in Theorem* 3.4, *then*

$$\sum_{k \in U_\infty} (\sigma_k^2 - \min_{i \in I_k}\{\|w_i\|^2\}) < \infty \ \Rightarrow \sum_{k \in U_\infty} \gamma_k < \infty \tag{4.21}$$

and

$$\sum_{k \in D_\infty} (\max_{i \in J_k}\{\|w_i\|^2\} - \eta_k^2) < \infty \ \Rightarrow \sum_{k \in D_\infty} \mu_k < \infty. \tag{4.22}$$

Proof. The expressions (4.15) and (4.16) imply that

$$\gamma_k \le \sum_{i \in I_k} \left(1 - \frac{\|w_i\|^2}{\sigma_k^2 \|G_*^{-\frac{1}{2}} w_i\|^2}\right) \left(\frac{\|G_*^{-\frac{1}{2}} w_i\|^2}{\|w_i\|^2}\right) (\sigma_k^2 - \|w_i\|^2)$$

$$= \sum_{i\in I_k} \left(\frac{\|G_*^{-\frac{1}{2}} w_i\|^2}{\|w_i\|^2} - \frac{1}{\sigma_k^2} \right) (\sigma_k^2 - \|w_i\|^2).$$

Dropping the term $-1/\sigma_k^2$ and using

$$\frac{\|G_*^{-\frac{1}{2}} w_i\|^2}{\|w_i\|^2} \le \|G_*^{-1}\|,$$

we have

$$\begin{aligned} \gamma_k &\le \|G_*^{-1}\| \sum_{i\in I_k} (\sigma_k^2 - \|w_i\|^2) \\ &\le n\|G_*^{-1}\|(\sigma_k^2 - \min_{i\in I_k}\{\|w_i\|^2\}). \end{aligned} \tag{4.23}$$

The relation (4.8) follows directly from summing (4.23) over all $k \in U_\infty$ and applying the bound (4.21).

Similarly, the expressions (4.17) and (4.18) and the assumption $\eta_k \ge \eta_{\min}$ imply that

$$\begin{aligned} \mu_k &\le \frac{1}{\eta_{\min}^2} \sum_{i\in J_k} (\|w_i\|^2 - \eta_k^2) \\ &\le \frac{n}{\eta_{\min}^2} (\max_{i\in J_k}\{\|w_i\|^2\} - \eta_k^2). \end{aligned} \tag{4.24}$$

Summing (4.24) over all $k \in D_\infty$ and applying the bound (4.22) give (4.9). □

5. A superlinearly convergent algorithm.

In this section we describe a specific implementation of Algorithm 2.1, and make use of the theory developed so far to show that it is globally and superlinearly convergent for strictly convex objective functions.

ALGORITHM 5.1. Automatic column scaling BFGS algorithm.

(0) Choose x_1 and a nonsingular and lower Hessenberg matrix V_1; set $k = 1$.

(1) Terminate if a stopping criterion is satisfied.

(2) Find an orthogonal matrix Q_k such that $L_k := V_k Q_k$ is lower triangular. Compute

$$d_k = -L_k^{-T} L_k^{-1} g_k,$$
$$x_{k+1} = x_k + \lambda_k d_k,$$

where λ_k is a steplength that satisfies the Wolfe conditions (1.4)–(1.5). (The stepsize $\lambda_k = 1$ is always tried first and is accepted if admissible.) Compute

$$s_k = x_{k+1} - x_k,$$
$$y_k = g_{k+1} - g_k.$$

(3) Perform the following steps to update L_k to W_k so that $W_k W_k^T$ is the BFGS update of $L_k L_k^T$:

(3.1) Compute $r_k = L_k^T s_k$.

(3.2) Find an orthogonal and lower Hessenberg matrix Ω_k such that $\Omega_k e_1 = r_k/\|r_k\|$.

(3.3) Construct $W_k = [w_1^k, w_2^k, \ldots, w_n^k]$, where w_i^k is given by

$$w_i^k = \begin{cases} y_k/\sqrt{y_k^T s_k}, & i = 1, \\ L_k \Omega_k e_i, & i = 2, 3, \ldots, n. \end{cases} \tag{5.1}$$

(4) Compute the scaling parameters: If $k = 1$,

$$\sigma_1^2 = \eta_1^2 = \frac{y_1^T y_1}{s_1^T y_1},$$

otherwise,

$$\sigma_k^2 = \frac{1}{n}\left[(n- \mid I_{k-1} \mid)\sigma_{k-1}^2 + \sum_{i \in I_{k-1}} \|w_i^{k-1}\|^2\right], \tag{5.2}$$

where

$$I_{k-1} = \{i \in [1, n] : \|w_i^{k-1}\| < \sigma_{k-1}\},$$

and

$$\eta_k^2 = \frac{1}{n}\left[(n- \mid J_{k-1} \mid)\eta_{k-1}^2 + \sum_{i \in J_{k-1}} \|w_i^{k-1}\|^2\right], \tag{5.3}$$

where

$$J_{k-1} = \{i \in [1, n] : \|w_i^{k-1}\| > \eta_{k-1}\}.$$

Construct $C_k = \text{diagonal}(c_1, c_2, \ldots, c_n)$ where c_i is given by

$$c_i = \begin{cases} \dfrac{\sigma_k}{\|w_i^k\|} & \text{if } \|w_i^k\| < \sigma_k, \\ \dfrac{\eta_k}{\|w_i^k\|} & \text{if } \|w_i^k\| > \eta_k, \\ 1 & \text{otherwise.} \end{cases} \tag{5.4}$$

Compute

$$V_{k+1} = W_k C_k.$$

(5) Set $k := k + 1$ and go to step (1).

To elaborate, each iteration k begins with the lower Hessenberg matrix V_k, which defines the Hessian approximation $B_k = V_k V_k^T$. We require that V_k be lower Hessenberg in order that the lower triangular matrix L_k can quickly be obtained by postmultiplying V_k by an orthogonal matrix Q_k. The matrix Q_k is defined implicitly by a sequence of at most n Givens rotations. Note that since $L_k = V_k Q_k$, we also have that $B_k = L_k L_k^T$. This allows us to compute the search direction by two triangular solves. The new iterate x_{k+1} is computed by means of a Wolfe line search.

Next we compute a lower Hessenberg matrix W_k so that $W_k W_k^T$ is the matrix obtained by applying the BFGS update to $L_k L_k^T$. This procedure, which is due to Powell [12], is described in step (3). It is easy to verify that W_k constructed by formula (5.1) has the desired property. Indeed,

$$W_k W_k^T = \sum_{i=1}^{n} w_i^k {w_i^k}^T = \frac{y_k y_k^T}{y_k^T s_k} + \sum_{i=2}^{n} L_k \Omega_k e_i e_i^T \Omega_k^T L_k^T. \tag{5.5}$$

On the other hand, the definitions of Ω_k and r_k (see steps (3.1) and (3.2)) give

$$\begin{aligned} L_k L_k^T &= L_k \Omega_k \Omega_k^T L_k^T \\ &= \sum_{i=1}^{n} L_k \Omega_k e_i e_i^T \Omega_k^T L_k^T \\ &= \frac{L_k L_k^T s_k s_k^T L_k L_k^T}{s_k^T L_k L_k^T s_k} + \sum_{i=2}^{n} L_k \Omega_k e_i e_i^T \Omega_k^T L_k^T. \end{aligned} \tag{5.6}$$

Solving for the summation in (5.6) and substituting the result into (5.5), we see that $W_k W_k^T$ is indeed given by the BFGS formula (1.6). To see that W_k is lower Hessenberg, one need only note the forms of L_k and Ω_k.

The matrix Ω_k need not be formed explicitly. Given L_k and r_k, a downdating procedure similar to the one described by Goldfarb [8] and Powell [12] can be used to obtain W_k without forming Ω_k.

The algorithm then updates the scaling parameters σ_k and η_k by formulas (5.2) and (5.3), and scales appropriately columns of W_k. In words, σ_k^2 is a weighted average of σ_{k-1}^2 and the square of the norms of the columns that were scaled up in iteration $k-1$. Similarly, η_k^2 is a weighted average of η_{k-1}^2 and the square of the norms of the columns that were scaled down in the previous iteration. Note that σ_k is nonincreasing, while η_k is nondecreasing. In particular, if I_{k-1} is empty, that is, if no scaling up occurred in iteration $k-1$, then $\sigma_k = \sigma_{k-1}$. Similarly, $\eta_k = \eta_{k-1}$ if J_{k-1} is empty. We now formally state and prove that Algorithm 5.1 is globally and superlinearly convergent for strictly convex problems, as a corollary to Theorems 3.4 and 4.1.

COROLLARY 5.1. *Let f, x_1, and B_1 satisfy the assumptions in Theorem* 3.4, *and assume that G is Lipschitz continuous at x_*. Then Algorithm* 5.1 *generates a sequence $\{x_k\}$ that converges superlinearly to x_*.*

Proof. Global convergence of the algorithm immediately follows from Theorem 3.4 since $\sigma_k \le \sigma_1$ and $\eta_k \ge \eta_1$, and all other assumptions of the theorem are satisfied.

To establish superlinear convergence, we will show that the inequalities (4.8) and (4.9) hold. We begin by analyzing the sum of $(\sigma_k^2 - \min_{i \in I_k}\{\|w_i^k\|^2\})$ over all iterations in which scaling up occurs. Rearranging the definition (5.2) gives us a new equality,

$$\sum_{i \in I_k} (\sigma_k^2 - \|w_i^k\|) = n(\sigma_k^2 - \sigma_{k+1}^2), \tag{5.7}$$

which is true for any iteration $k \ge 1$. Both sides of the equality (5.7) are 0 when $k \notin U_\infty$. (Recall that U_l contains the indices of the iteration numbers less than or equal to l where scaling up occurs.) Therefore, summing the left-hand side over all iterations k, we get

$$\begin{aligned}\sum_{k=1}^{\infty}\sum_{i\in I_k}(\sigma_k^2 - \|w_i^k\|) &= \sum_{k\in U_\infty}\sum_{i\in I_k}(\sigma_k^2 - \|w_i^k\|) \\ &\geq \sum_{k\in U_\infty}(\sigma_k^2 - \min_{i\in I_k}\{\|w_i^k\|^2\}). \end{aligned} \tag{5.8}$$

Moreover, summing the right-hand side of the equality (5.7) over any number of iterations $l \geq 1$, we get

$$\begin{aligned} n\sum_{k=1}^{l}(\sigma_k^2 - \sigma_{k+1}^2) &= n(\sigma_1^2 - \sigma_{l+1}^2) \\ &\leq n\sigma_1^2. \end{aligned} \tag{5.9}$$

Hence, summing the equality (5.7) over all iterations k, and substituting the relations (5.8) and (5.9) yields

$$\sum_{k\in U_\infty}(\sigma_k^2 - \min_{i\in I_k}\{\|w_i^k\|^2\}) \leq n\sigma_1^2.$$

Thus the inequality (4.8) is implied by invoking the implication (4.21) of Lemma 4.3.

It remains to be shown that the inequality (4.9) holds. We consider separately the case when there exists an iteration $k = \bar{k}$ such that $\eta_k \geq \lambda_{\max}$, the maximum eigenvalue of G_*, and the case when $\eta_k < \lambda_{\max}$ for all k. Suppose first that $\eta_k \geq \lambda_{\max}$ for some iteration $k = \bar{k}$. Since η_k is nondecreasing by the definition (5.3), it follows that $\eta_k \geq \lambda_{\max}$ for all iterations $k \geq \bar{k}$. Thus we can invoke the implication (4.20) of Lemma 4.2 to show that the inequality (4.9) holds in this case. In the other case, we proceed to analyze the sum of $(\max_{i\in J_k}\{\|w_i^k\|^2\} - \eta_k^2)$ over all the iterations in which scaling down occurs. A new equality can again be derived by rearranging the definition (5.3):

$$\sum_{i\in J_k}(\|w_i^k\| - \eta_k^2) = n(\eta_{k+1}^2 - \eta_k^2). \tag{5.10}$$

Recall the definition of D_k, which is the counterpart of U_k. In analogy to the derivation of the relations (5.8) and (5.9), we can derive the relations:

$$\sum_{k=1}^{\infty}\sum_{i\in J_k}(\|w_i^k\| - \eta_k^2) \geq \sum_{k\in D_\infty}(\max_{i\in J_k}\{\|w_i^k\|^2\} - \eta_k^2) \tag{5.11}$$

and

$$n\sum_{k=1}^{l}(\eta_{k+1}^2 - \eta_k^2) < n(\lambda_{\max} - \eta_1^2) \tag{5.12}$$

for any $l \geq 1$. Summing the equality (5.10) over all iterations k, and substituting the relations (5.11) and (5.12), yields

$$\sum_{k\in D_\infty}(\max_{i\in J_k}\{\|w_i^k\|^2\} - \eta_k^2) < n(\lambda_{\max} - \eta_1^2).$$

Thus, by invoking the implication (4.22) of Lemma 4.3, we have that (4.9) holds. Because of this fact and the fact that (4.8) holds as shown earlier, we conclude from Theorem 4.1 that the iterates converge superlinearly. □

6. Final remarks. We have described in this paper the conditions under which an automatic scaling algorithm based on the direct form of the BFGS update can be proven to be globally and superlinearly convergent. It should be noted that rules for scaling that are more general than those in step (4) of Algorithm 2.1 exist. An example is to have different values of σ_k and η_k, say σ_i^k and η_i^k, associated with each column i of W_k. However, such generalization would complicate the notation unnecessarily, and would crowd out the important points of the theory that we wish to convey. Only slight modifications of the proofs are required to accommodate this generalization.

It is also possible to describe an algorithm similar to Algorithm 2.1 but based on the inverse BFGS formula, and to give sufficient conditions for its convergence. Specifically, a theorem similar to Theorem 3.4 can be stated for such an algorithm, but only when η_k is set to ∞, i.e., when scaling down is disallowed. We have not been able to prove convergence for a more general choice of η_k. It remains to be investigated whether this difficulty is inherent to the nature of the algorithm, or is due to a deficiency in our method of proof.

A column scaling algorithm that is not a particular case of Algorithm 2.1 has recently been proposed by Siegel [14]. He also updates a matrix of conjugate directions to define the iteration matrix, but the scaling rules are quite different from the ones considered here. Moreover, when a certain criterion holds, the search direction is determined by dropping a set of columns from the matix of conjugate directions. Siegel shows that his algorithm is superlinearly convergent and gives an example to illustrate its practical behavior.

We believe that the choice of the scaling parameters given in Algorithm 5.1 is adequate asymptotically, but that more aggressive strategies may prove useful away from the solution. Specifically, in our algorithm, σ_k is nonincreasing, and η_k is nondecreasing. It might occasionally be better to increase σ_k and to decrease η_k. For example, a problem may have regions with small curvature far away from the solution, and large curvature near the solution. In this case it may be advantageous for the algorithm to increase σ_k sometimes. A similar argument can be made for a need to decrease η_k. The question of how to implement the best scaling strategy is the subject of our future research. We believe that it would be a mix of an aggressive strategy in the early iterations and a more conservative one towards the end. In any case, the theory developed in this paper will prove to be useful for analyzing the global and asymptotic behavior of any such strategies.

Acknowledgments. We would like to acknowledge several useful conversations with Pei-huang Lu, Michael Powell, and Dirk Siegel. We are very grateful to a referee, whose suggestions resulted in a substantial reduction in one of the proofs.

REFERENCES

[1] C. G. Broyden, *A new double-rank minimization algorithm*, AMS Notices, 16 (1969), p. 670.

[2] R. H. Byrd, D. C. Liu, and J. Nocedal, *On the behavior of Broyden's class of quasi-Newton methods*, SIAM J. Optimization, 2 (1992), pp. 533–557.

[3] R. H. Byrd, J. Nocedal, and Y. Yuan, *Global convergence of a class of quasi-Newton methods on convex problems*, SIAM J. Numer. Anal., 24 (1987), pp. 1171–1190.

[4] R. H. Byrd and J. Nocedal, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization*, SIAM J. Numer. Anal., 26 (1989), pp. 727–739.

[5] J. E. Dennis and J. J. Moré, *A characterization of superlinear convergence and its application to quasi-Newton methods*, Math. Comp., 28 (1974), pp. 549–560.

[6] R. Fletcher, *A new approach to variable metric algorithms*, J. Comput., 13 (1970), pp. 317–322.

[7] D. Goldfarb, *A family of variable metric methods derived by variational means*, Math. Comp., 24 (1970), pp. 23–26.

[8] ———, *Factorized variable metric methods for unconstrained optimization*, Math. Comp., 30 (1976), pp. 796–811.

[9] A. Griewank, *The global convergence of partitioned* BFGS *on problems with convex decompositions and Lipschitzian gradients*, Math. Programming, 50 (1991), pp. 141–175.

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

[11] M. J. D. Powell, *How bad are the* BFGS *and* DFP *methods when the objective function is quadratic?*, Math. Programming, 34 (1986), pp. 34–47.

[12] ———, *Updating conjugate directions by* BFGS *formula*, Math. Programming, 38 (1987), pp. 29–46.

[13] D. F. Shanno, *Conditioning of quasi-Newton methods for function minimization*, Math. Comp., 24 (1970), pp. 647–657.

[14] D. Siegel, *Modifying the* BFGS *update by a new column scaling technique*, Tech. Rep. DAMTP 1991/NA5, Dept. of Applied Mathematics and Theoretical Physics, University of Cambridge, U.K., 1991.

[15] ———, *Modifying the* BFGS *update by column scaling: An example of linear convergence*, Report DAMTP 1991/NA15, Dept. of Applied Mathematics and Theoretical Physics, University of Cambridge, U.K., 1991.

SIAM J. OPTIMIZATION
Vol. 3, No. 3, pp. 654–667, August 1993

MULTI-OBJECTIVE CONTROL-STRUCTURE OPTIMIZATION VIA HOMOTOPY METHODS*

JOANNA RAKOWSKA[†], RAPHAEL T. HAFTKA[‡], AND LAYNE T. WATSON[§]

Abstract. A recently developed active set algorithm for tracing parametrized optima is adapted to multi-objective optimization. The algorithm traces a path of Kuhn–Tucker points using homotopy curve tracking techniques, and is based on identifying and maintaining the set of active constraints. Second order necessary optimality conditions are used to determine nonoptimal stationary points on the path. In the bi-objective optimization case the algorithm is used to trace the curve of efficient solutions (Pareto optima). As an example, the algorithm is applied to the simultaneous minimization of the weight and control force of a ten-bar truss with two collocated sensors and actuators, with some interesting results.

Key words. active set, bi-objective, control-structure optimization, efficient solutions, homotopy, multi-objective optimization, optimal curve tracing, probability-one homotopy

AMS subject classifications. 65F, 65K, 73K, 49B

1. Introduction. In recent years there has been considerable interest in simultaneous control-structure optimization of space structures [5]. Although the problem can be solved by sequential optimization of a structure objective (J_s) and a control system objective (J_c), better designs are obtained when both objectives are optimized simultaneously (e.g., [6]). In the latter approach both objectives are combined into a bi-objective cost function $\mathcal{J} = (J_s, J_c)$. Bi-objective optimization gives the designs (known as *efficient* solutions) where one objective can be improved only at the expense of the other one. Such a formulation of the problem produces a family of design options which can be used in the early stages of the design process to guide the evolution of the design [3].

The optimal solutions to the problem of minimizing the bi-objective cost function $\mathcal{J} = (J_s, J_c)$ can be found by optimizing the convex combination $(1-\alpha)J_s + \alpha J_c$ of J_s and J_c [3]. Homotopy curve tracking methods can be used to generate the curve of solutions for $\alpha \in [0,1]$ whenever the curve is smooth (e.g., [9], [13]). However, the curve of optimum solutions is not necessarily smooth at points corresponding to changes in the set of active constraints. Therefore it is necessary to locate such points and restart the tracing algorithm with a new set of active constraints.

* Received by the editors December 20, 1990; accepted for publication (in revised form) March 30, 1992.

[†] Department of Mathematics, Virginia Polytechnic Institute & State University, Blacksburg, Virginia 24061-0123. The work of this author was supported in part by National Aeronautics and Space Administration grant NAG-1-224.

[‡] Department of Aerospace and Ocean Engineering, Virginia Polytechnic Institute & State University, Blacksburg, Virginia 24061-0203. The work of this author was supported in part by National Aeronautics and Space Administration grant NAG-1-224.

[§] Departments of Computer Science and Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061-0106. The work of this author was supported in part by Department of Energy grant DE-FG05-88ER25068, National Aeronautics and Space Administration grant NAG-1-1079, and Air Force Office of Scientific Research grant 89–0497.

There have been recent attempts to construct algorithms for tracing a path of optimal solutions. Rao and Papalambros [12] use simple continuation to find the family of parametrized optima for large changes in a parameter. Lundberg and Poore [7] use a sophisticated predictor-corrector homotopy curve tracking algorithm to investigate the dependence of the solution on a parameter and to locate bifurcations and points of extreme solution sensitivity. The objective of the present paper is to describe the application of a recently developed homotopy algorithm [10] to tracing optima of bi-objective optimization problems.

Section 2 develops the control-structure optimization problem, used as a representative application of the algorithm. Section 3 briefly recounts some homotopy theory, although the probability-one aspect of globally convergent homotopy methods is not used in any essential way here. The heart of the active set homotopy algorithm proposed here, detecting and correctly managing changes in the active set of constraints, is described in detail in §4. Section 5 presents numerical results for a ten-bar truss, which illustrates several subtle and complicated phenomena associated with bi-objective optimization.

2. Control-structure optimization.

The problem of simultaneous structure-control optimization is formulated as the minimization of the structural weight W and maximum control force $F_{\max}$ subject to constraints on the damping ratios ξ_i of the first n_m vibration modes of the structure.

The equations of motion of the structure controlled by n_c collocated sensors and actuators are written as

$$M\ddot{u} + D_0\dot{u} + Ku = F,$$

where M, D_0 and K are the mass, structural damping and stiffness matrices, respectively, u is the displacement vector, F is the applied control force vector, and a dot denotes differentiation with respect to time. A simple direct-rate feedback control law [8] is used for the actuator force vector F given as

$$F = -D_c\dot{u},$$

where D_c is the control matrix which has nonzero rows and columns at positions corresponding to components of $\dot{u}$ measured by the sensors. Assuming that there is no structural damping ($D_0 = 0$), the structure is described by the system

$$M\ddot{u} + D_c\dot{u} + Ku = 0$$

with the general solution $u = u_0e^{\mu t}$. The stability of the system is controlled by the real parts of the eigenvalues μ_i. The stability margins are characterized by the damping ratios ξ_i defined as

$$\xi_i = \frac{-\sigma_i}{\sqrt{\sigma_i^2 + \omega_i^2}},$$

where σ_i and ω_i are the real and imaginary parts of μ_i.

We assume that the matrix D_c is positive semidefinite so that the closed loop system has at least the same stability as the open loop system. Following [8] the goal is to have a control system which minimizes the maximum control forces for a given velocity bound $\|\dot{u}\|_\infty \le U$. The maximum control force applied by the actuators is

$$F_{\max} = \max \frac{\|F\|_\infty}{\|\dot{u}\|_\infty} = \|D_c\|_\infty = \max_i \sum_j |d_{ij}|,$$

where the d_{ij} are the elements of the control matrix D_c.

The problem of simultaneous control-structure optimization is the bi-objective optimization problem

$$\begin{aligned} &\text{minimize } (W(a),\, F_{\max}(a, D_c)) \\ &\text{such that } \sum_j |d_{ij}| \leqq F_{\max}, \\ &\xi_i(a, D_c) \geqq \xi_{0i} \quad \text{for} \quad i = 1, \dots, n_m, \\ &D_c \geqq 0, \quad (D_c \text{ positive semidefinite}), \\ &a_i \geqq a_{0i} \quad \text{for} \quad i = 1, \dots, n_s, \end{aligned} \tag{1}$$

where a is a vector of structural dimensions and $W(a)$ is the structure's weight. The curve of all efficient solutions (designs for which neither $W(a)$ nor $F_{\max}$ can be simultaneously improved) can be obtained by minimizing the combination $(1-\alpha)W + \alpha F_{\max}$ of the two objective functions for all values of α between 0 and 1. To simplify the later algorithmic discussion of the constraints, the problem can be rewritten as

$$\text{minimize } c(x, \alpha) = (1-\alpha)\, W + \alpha F_{\max} \tag{2}$$

$$\text{subject to } G_i(x) = x_{0i} - x_i \leqq 0, \qquad i = 1, \dots, n_1, \tag{3}$$

$$G_{j+n_1}(x) \leqq 0, \qquad j = 1, \dots, n_2, \tag{4}$$

where x is the n_1-vector of design variables including a structural size vector a, the nonzero elements of the matrix D_c, and $F_{\max}$. The design variables are subject to the minimum value constraints $x_i \geqq x_{0i}$; the constraints (4) correspond to the other constraints in the problem (1); and α is the parameter assuming all values between 0 and 1. The Lagrangian function and Kuhn–Tucker conditions for this problem are:

$$L(x, \alpha, \lambda) = c(x, \alpha) + \sum_{i=1}^{n_1} \lambda_i (x_{0i} - x_i) + \sum_{j=n_1+1}^{n_1+n_2} \lambda_j G_j(x), \tag{5}$$

$$\frac{\partial c}{\partial x_i} + \sum_{j=n_1+1}^{n_1+n_2} \lambda_j \frac{\partial G_j}{\partial x_i} - \lambda_i = 0, \qquad i = 1, \dots, n_1, \tag{6}$$

$$G_j \lambda_j = 0, \qquad j = 1, \dots, n_1 + n_2, \tag{7}$$

$$\lambda_j \geqq 0, \qquad j = 1, \dots, n_1 + n_2, \tag{8}$$

$$G_j \leqq 0, \qquad j = 1, \dots, n_1 + n_2. \tag{9}$$

Equations (6)–(7) form a system of nonlinear equations to be solved for the design variables x_i and the Lagrange multipliers λ_j associated with active constraints of the form (4) and with the bounds for design variables (3). The solution (x, α, λ) of these equations, in the generic case, follows a path (not necessarily monotone in α) that consists of several smooth segments, each segment characterized by a different set of active constraints.

3. Homotopy curve tracking. The system of nonlinear equations (6)–(7) is solved by a homotopy curve tracking method. By the Implicit Function Theorem, if $F : E^{N+1} \to E^N$ is C^1, the system of equations

$$F(x, \alpha) = 0 \tag{10}$$

has some solution (x_0, α_0), and the Jacobian matrix $DF(x_0, \alpha_0)$ of the function F at (x_0, α_0) has full rank, then there is some neighbourhood U of (x_0, α_0) such that there is a unique curve of zeros of $F(x, \alpha)$ in U passing through (x_0, α_0). Assuming that 0 is a regular value of F, this full rank of the Jacobian matrix implies that the zero set of (10) contains a smooth curve Γ in $(N+1)$-dimensional (x, α) space, emanating from (x_0, α_0); Γ has no bifurcations and is disjoint from any other zeros of (10). The curve Γ can be parametrized by arc length s:

$$x = x(s), \qquad \alpha = \alpha(s). \tag{11}$$

Taking the derivative of (10) with respect to arc length, the nonlinear system of equations is transformed into the ordinary differential equations

$$\left[F_x(x(s), \alpha(s)), \quad F_\alpha(x(s), \alpha(s)) \right] \begin{pmatrix} \dfrac{dx}{ds} \\ \dfrac{d\alpha}{ds} \end{pmatrix} = 0, \tag{12}$$

and

$$\left\| \begin{pmatrix} \dfrac{dx}{ds} \\ \dfrac{d\alpha}{ds} \end{pmatrix} \right\|_2 = 1, \tag{13}$$

where F_x and F_α denote the partial derivatives of F with respect to x and α, respectively. With the initial conditions at $s = 0$,

$$x(0) = x_0, \qquad \alpha(0) = \alpha_0, \tag{14}$$

(12)–(14) can be treated as an initial value problem. Its trajectory is the path Γ of optimal solutions $Z(s) = (x(s), \alpha(s))$.

A probability-one homotopy approach would construct a homotopy map $\rho_b(\sigma, x; \alpha)$, where $\sigma \in [0, 1)$ and b is a random parameter vector, such that tracking a zero curve of ρ_b would lead to a solution of (10) for fixed α. It would *not* be necessary to assume that 0 is a regular value of either F or ρ_b—the supporting theory [15], [16] says that 0 is a regular value of ρ_b for almost all b, but F must be C^2. Algorithms based on such homotopy maps ρ_b are powerful and robust, but provide solutions only for fixed α, and cannot easily track the entire zero set of (10) (which is the goal here). Thus, strictly speaking, the algorithm used here is not a modern (probability-one) homotopy method but a variant of *arc length continuation*, on which there is a huge literature. See the references in [1], [7], or [14]–[17].

A software package HOMPACK [15], [17], which implements several homotopy curve tracking algorithms, is used to track the zero curve Γ. The HOMPACK algorithms take steps along the zero curve using prediction and correction to find the next point. Just to give the flavor of such algorithms, one of the algorithms implemented in HOMPACK, called the "normal flow" algorithm, is sketched here. In the prediction phase a Hermite cubic $p(s)$ is constructed which interpolates the zero curve Γ at two known points, $Z(s_1)$ and $Z(s_2)$. The predicted next point is

$$Z^{(0)} = p(s_2 + h), \tag{15}$$

where $p(s)$ is the Hermite cubic, and h is an estimate of the optimal step (in arc length) to take along Γ.

The corrector iteration is

$$Z^{(k+1)} = Z^{(k)} - \left[DF(Z^{(k)})\right]^+ F(Z^{(k)}), \qquad k = 0, 1, \ldots$$

where $\left[DF(Z^{(k)})\right]^+$ is the Moore–Penrose pseudoinverse of the $N \times (N+1)$ Jacobian matrix DF. In practice this pseudoinverse is not calculated explicitly; see [15] for the details of the Hermite cubic interpolant construction and the corrector iteration.

The optimal step size h is chosen to prevent the corrector iteration from being too costly. HOMPACK lets the user specify nondefault values used in determining the step size, for example, the maximum and minimum allowed step size. Lundberg and Poore [7] have probably the best algorithm to date for determining h. The parameter α in equations (12)–(14) is a dependent variable, which distinguishes modern homotopy methods from standard continuation, imbedding, or incremental methods. The modern homotopy approach is also different from initial value or differentiation methods, since the controlling variable is arc length s, rather than α.

4. Solution along a segment and transition to the next segment. Since the active constraints in a segment are fixed, they can be treated as equality constraints. Furthermore, along each segment some design variables are fixed at their lower bound. The vector of these inactive (passive) variables is denoted x_p and need not be considered as design variables for that segment. The vector of active design variables x_i ($i \in \mathcal{I}_a$) is denoted as x_a. Along each segment the Kuhn–Tucker conditions are solved for the active design variables x_i ($i \in \mathcal{I}_a$) and for the Lagrange multipliers λ_g associated with the active constraints of the form (4) (λ_j, $j \in \mathcal{I}_g$). For each segment there are two types of equations:

$$V1: G_j(x) = 0, \qquad j \in \mathcal{I}_g, \tag{16}$$

$$V2: \frac{\partial c}{\partial x_i} + \sum_{j \in \mathcal{I}_g} \lambda_j \frac{\partial G_j}{\partial x_i} = 0, \qquad i \in \mathcal{I}_a. \tag{17}$$

The active design variables and the Lagrange multipliers associated with active constraints (4) are the variables in these equations. The homotopy algorithm needs the Jacobian matrix of these functions with respect to α, x_a, and λ_g.

As suggested by the discussion in §3, it is explicitly assumed here that 0 is a regular value of the system defined by (16) and (17), i.e., the Jacobian matrix has full rank along a segment. Let $y = (\alpha, x_a, \lambda_g)$. At the start of a segment the set of active design variables and active constraints for this segment has to be found, so that the vector y is defined. A set of equations is then generated, with the type of each variable determining the form of the equation appended to the system of equations. For a Lagrange multiplier associated with an active constraint of the form (4), the equation has the form (16), and for an active design variable, the equation has the form (17). The system of equations for the segment is solved using the previously described homotopy curve tracking technique. Next the Lagrange multipliers for inactive design variables are calculated according to (6). In these equations the Lagrange multipliers associated with active constraints of the form (4) have been computed by the homotopy method, and the Lagrange multipliers associated with inactive constraints (4) are known to be zero. At each point of a segment the Lagrange multipliers associated with the lower bound of the inactive design variables or the active constraints of the form (4) in the segment should be nonnegative, the value of each G_j, $j = n_1, \ldots, n_1+n_2$ should be less than or equal to zero, and all design variables should be larger than or equal to their lower bound. If any of the above conditions is not satisfied the segment is terminated and a new one is started. The transition point to a new segment is called here a *switching point*. Depending on the type of termination, the switching point is the point (which is calculated using a guarded secant method, since the curve tracker will have overshot) where

1) one of the positive Lagrange multipliers becomes equal to zero, or
2) a previously negative G_j of the form (4) becomes equal to zero, or
3) an active design variable $x_i (i \in \mathcal{I}_a)$ becomes inactive (equal to x_{0i}).

At the beginning of each segment the system of linear equations (6) is solved for $\lambda_1, \ldots, \lambda_m$, $m = n_1 + n_2$, to check which design variables and constraints are active and to find the initial values of the Lagrange multipliers for the new segment. First the Lagrange multipliers for inactive constraints are set to zero so that Lagrange multipliers only for potentially active constraints (those equal to zero) are considered.

Since some of the constraints (4) may be inactive (their values at the switching point are less than zero), or the derivatives of the constraints (4) with respect to the design variables can assume values for which some columns or rows in the coefficient matrix of the system (6) are linearly dependent, the rank of this matrix can be less than n_2. The rank of the coefficient matrix for the system (6) determines the number of the constraints (4) that are assumed to be active in the next segment.

The QR factorization with column pivoting (or the singular value decomposition) is used to find the rank r of the coefficient matrix. (Needing to numerically calculate the rank is a fundamental weakness, closely related to the need to get the active set right in any active set algorithm.) Next the system (6) is solved for all subsets of r columns that are linearly independent assuming that the Lagrange multipliers for the constraints (4) corresponding to the remaining columns are zero. To get the solution for each subset at least r design variables are assumed to be active (the corresponding Lagrange multipliers are set to zero). For each subset of r columns (corresponding to r constraints) all combinations of r out of n_1 design variables are assumed to be active. The system is solved in turn for each combination to find all sets of active design variables and active constraints (4) such that the Lagrange multipliers are nonnegative.

Sometimes there are several solutions satisfying the condition that all the Lagrange multipliers be nonnegative. Then for each solution the signs of the derivatives of the design variables with respect to the arc length s are calculated. A set of active constraints (4) and active design variables is accepted when the values of these signs indicate that no active constraint will be immediately violated for increasing values of s.

To calculate the values of the derivatives of the design variables with respect to α, the Kuhn–Tucker conditions (6)–(7) are differentiated with respect to α. This gives:

$$(A+Z)\frac{\partial x_a}{\partial \alpha} + N\frac{\partial \lambda_g}{\partial \alpha} + \frac{\partial(\nabla c)}{\partial \alpha} + \left(\frac{\partial N}{\partial \alpha}\right)\lambda_g = 0, \tag{18}$$

$$N^T\frac{\partial x_a}{\partial \alpha} + \frac{\partial G_g}{\partial \alpha} = 0, \tag{19}$$

where x_a is a vector of design variables, λ_g is a vector of the Lagrange multipliers for active G_j, G_g is a vector of active constraints G_j, $j \in \mathcal{I}_g$, N has components

$$n_{ij} = \frac{\partial G_j}{\partial x_i} \qquad (j \in \mathcal{I}_g, \quad i \in \mathcal{I}_a),$$

A is the Hessian of the objective function c,

$$a_{ij} = \frac{\partial^2 c}{\partial x_i \partial x_j},$$

and Z is a matrix with elements

$$z_{il} = \sum_{j\in\mathcal{I}_g} \frac{\partial^2 G_j}{\partial x_i \partial x_l}\lambda_j.$$

After equations (18) and (19) are solved, derivatives of each G_j corresponding to an active constraint (4) with respect to α are calculated according to

$$\frac{\partial G_j}{\partial \alpha} = \sum_{i\in\mathcal{I}_a} \frac{\partial G_j}{\partial x_i}\frac{\partial x_i}{\partial \alpha}, \qquad j \in \mathcal{I}_g. \tag{20}$$

For each candidate solution satisfying the Kuhn–Tucker conditions, the signs of the derivatives with respect to arc length s are then calculated by multiplication by $\operatorname{sgn}(d\alpha/ds)$ (determined by the direction in which a segment is to be tracked). The signs of the derivatives with respect to arc length s are calculated for design variables, Lagrange multipliers and G_j's corresponding to active constraints. A solution is accepted if the derivatives with respect to s of active design variables that are at their lower bound are nonnegative, the derivatives with respect to s of zero Lagrange multipliers that correspond to active constraints (4) are nonnegative and the derivatives of G_j's that are equal to zero are nonpositive.

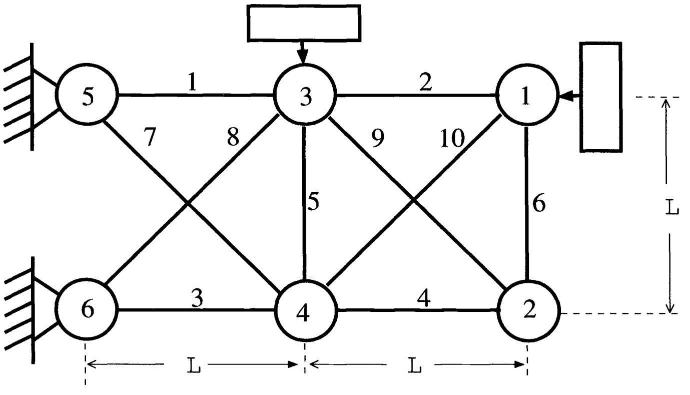


FIG. 1. *Ten-bar truss with actuators.*

The path of optimal points can be discontinuous [10], [11]. It is possible that beyond some value of α there are no neighbouring optima. At this point α is fixed and the problem must be solved by a standard optimization algorithm to find a new optimum. Tracking a path of optimal solutions can then be resumed at this new point. It is also possible that beyond a certain value of α no optimum exists, for example, if the problem becomes unbounded. Furthermore, singular points such as bifurcation and fold points may occur [7]. Singular points correspond to a rank deficiency of the Jacobian matrix of the functions given in (16) and (17), which has explicitly been assumed not to occur. A more detailed description of this segment switching algorithm is given in Rakowska et al. [10].

Second order optimality conditions [4] are checked to verify that the stationary points found by solving the Kuhn–Tucker conditions are indeed minima. Second order necessary conditions are

$$(21) \qquad d^t\left[\nabla^2_{x_a}L\right]d \geqq 0 \quad \text{for every } d \text{ such that } (\nabla G_j)^t d = 0 \quad \forall j \in I_g,$$

where

$$\left[\nabla^2_{x_a}L\right]_{lm} = \frac{\partial^2 L}{\partial x_l \partial x_m}, \qquad l,m \in \mathcal{I}_a.$$

Recall that N is a matrix whose columns are the gradients of active constraints G_j $(j \in \mathcal{I}_g)$. Then a QR factorization of N,

$$N = QR = [\underbrace{Q_1}_{|\mathcal{I}_g|} : \underbrace{\;Q_2\;}_{|\mathcal{I}_a|-|\mathcal{I}_g|}]R,$$

gives a basis (columns of Q_2) for ker N^t = (im $N)^\perp$, i.e., a basis for all vectors $d \perp \nabla G_j \; \forall j \in \mathcal{I}_g$. Therefore the second order necessary condition (21) is equivalent to $Q_2^t\left[\nabla^2_{x_a}L\right]Q_2$ being positive semidefinite. When the second order necessary conditions are not satisfied it may still be useful to follow the path of stationary points until the solutions again become optimal. An alternative way of dealing with nonoptimality along Γ is to find a point on another path in the zero set using a standard optimization algorithm.

5. Ten-bar truss example. Numerical results are presented here for the ten-bar truss structure shown in Fig. 1. Numbers in circles indicate joints and plain numbers label truss elements. The truss is controlled by two pairs of direct-rate feedback collocated sensors and actuators shown by boxes in the figure. The sensors measure velocities, and the actuators apply forces at the positions and directions indicated in Fig. 1. The positions of the actuators have been obtained by an optimization that determined the most effective locations for controlling the first four modes. The sensor and actuator pairs are associated with the first (horizontal velocity at joint 1) and sixth (vertical velocity at joint 3) components of the velocity vector $\dot{u}$. The weight of the truss (excluding constant masses of 10 kg at the nodes) is given by $\sum_{i=1}^{10} \rho a_i l_i$, where a_i and l_i are the cross-sectional area and length, respectively, of the ith truss member and ρ is the weight density. The first four modes are required to have at least three percent damping ($\xi_{0i} = 0.03$), $L = 354\,\text{in}$, and the minimum area gage for all truss members is $a_{0i} = 0.1085\,\text{in}^2$. The optimization problem (2)–(4) then becomes

$$
\begin{aligned}
\text{minimize } c(a,\alpha) &= (1-\alpha)k\sum_{i=1}^{10} \rho a_i l_i + \alpha F_{\max},\\
\text{subject to } G_i &= a_{0i} - a_i \leqq 0, \qquad i = 1,\dots,10,\\
G_{11} &= -d_{11} \leqq 0,\\
G_{12} &= -d_{66} \leqq 0,\\
G_{13} &= -F_{\max} \leqq 0,\\
G_{14} &= |d_{11}| + |d_{16}| - F_{\max} \leqq 0,\\
G_{15} &= |d_{16}| + |d_{66}| - F_{\max} \leqq 0,\\
G_{j+15} &= 0.03 - \xi_j(a, d_{11}, d_{16}, d_{66}) \leqq 0, \qquad j = 1,\dots,4,\\
G_{20} &= d_{16}^2 - d_{11}d_{66} \leqq 0,
\end{aligned}
$$

where a is a vector of truss element cross-sectional areas, l is a truss element length vector, d_{11}, d_{16}, d_{66} are the nonzero entries of the control matrix D_c, $F_{\max}$ is the control force applied by actuators, and k is a scaling constant taken here to be 0.0261. The design variables in this formulation include a, d_{11}, d_{16}, d_{66} and $F_{\max}$. Since $F_{\max}$ is not a smooth function of the other design variables, adding it as a design variable removes discontinuities in the derivative of the objective function. Furthermore, the absolute value function $|d_{ij}|$ is not differentiable at zero and so is replaced by a quartic polynomial (matching the slope of $|d_{ij}|$ at $\pm d_t$) near zero:

$$
|d_{ij}| \approx \frac{d_t}{2}\left[3\left(\frac{d_{ij}}{d_t}\right)^2 - \left(\frac{d_{ij}}{d_t}\right)^4\right] \qquad \text{for } |d_{ij}| \leqq d_t,
$$

where d_t is taken to be 5% of a typical value for d_{ij}.

The switching points on the path of stationary points are shown in Table 1. For $\alpha = 0$ the weight is the only objective, hence the cost function is minimized when all the areas are at minimum gage. The values for d_{11}, d_{16}, d_{66} and $F_{\max}$ were obtained by minimizing the control objective with a standard sequential quadratic

TABLE 1
Path of solutions for ten-bar truss example.

Segment	α	$F_{\max}$	W	c	Event
0.	0.00000	3.02251	48.46283	1.45844	$F_{\max}$, d_{11}, d_{16}, d_{66} and G_{15}, G_{16}, G_{20} are active
1.	0.10921	3.02251	48.46283	1.45844	a_1 becomes active
2.	0.16123	2.74944	50.15051	1.54109	Constraint on ξ_2 becomes active
3.	0.28693	2.74943	50.15056	1.72217	a_7 becomes active
4.	0.31255	2.65683	51.66604	1.75732	Constraint on ξ_1 becomes active
5.	0.83345	2.65683	51.66609	2.43892	a_4 becomes active
6.	0.86770	2.65520	52.02666	2.48356	a_6 becomes active
7.	0.73754	2.60414	58.87609	2.32371	a_7 becomes inactive
8.	0.87005	2.59906	59.62525	2.46354	Constraint on ξ_2 becomes inactive
9.	0.93036	2.54966	76.44878	2.51105	a_5 becomes active
10.	0.94390	2.53224	86.29556	2.51653	a_3 becomes active
11.	0.94940	2.52316	92.48853	2.51763	a_1 becomes inactive
12.	1.00183	2.51446	105.45971	2.51403	α becomes greater than 1

programming algorithm (VMCON [2]). The same solution holds for small values of α. For $\alpha \geqq 0.1092$ the derivative of the objective function with respect to a_1 becomes negative and therefore the objective function can be reduced by using a_1 as an active design variable. The homotopy method is used to follow the path of stationary points starting with this value of α.

The path shown in Table 1 consists of 12 segments, with the first column in the table giving α at the beginning of the segment. The last column in the table describes the event that signaled the switching point at the beginning of the segment. Segments are terminated when a design variable or a constraint becomes active, or when an active design variable becomes inactive. Plots of the objective function and its two components W and $F_{\max}$ are given in Figs. 2, 3, and 4, respectively.

Plots of the weight and the maximum control force indicate that the best designs can be obtained for values of α near 0.8. For these values of α the maximum control force $F_{\max}$ is reduced by 83% of its maximum decrease (corresponding to α changing from 0 to 1), whereas the weight is increased only by 20% of its maximum change.

Along Segments 2 and 4 the design variables stay essentially at the same value, whereas the Lagrange multipliers for active constraints change considerably. At the end of Segment 5 no new segment for increasing α can be found. However it is possible to continue the path by decreasing α to obtain Segment 6. The second order necessary conditions are not satisfied along this segment, so points of Segment 6 are only stationary points for the problem. The path of optimal solutions is resumed in Segment 7. The plot of the objective function in Segments 5, 6, and 7 is magnified in Fig. 5. At points of discontinuity of the path of optimal solutions a standard

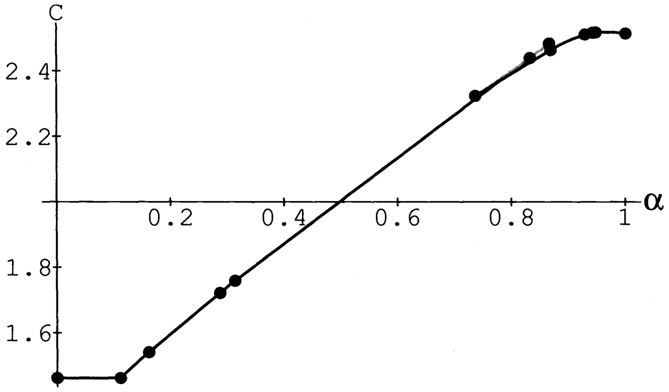


FIG. 2. *Objective function c along Segments* 0–11 (*gray line denotes nonoptimal stationary points, black line denotes optimal points*).

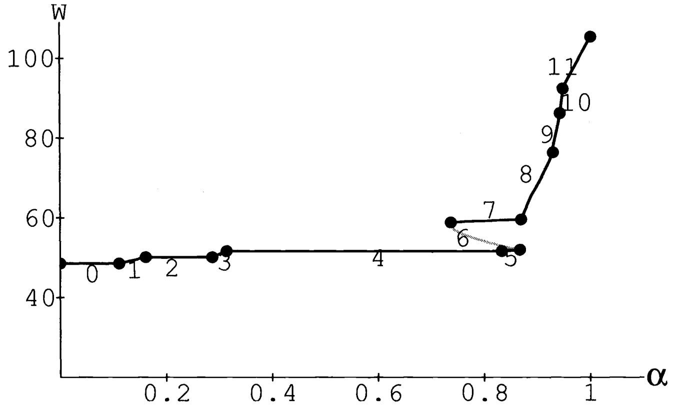


FIG. 3. *Weight W* (*pounds*) *along Segments* 0–11 (*gray line denotes nonoptimal stationary points, black line denotes optimal points*).

optimization program (e.g., VMCON) can be used to find a point where the solutions again become optimal. It can be also worthwhile to follow the path of nonoptimal stationary points until a new optimal point is encountered, if the nonoptimal segment is short or if it is difficult to find a point on another optimal branch using standard optimization. In this work the path of stationary points was followed even if they did not satisfy the necessary optimality conditions.

At the beginning of Segment 8 the path of the stationary points can again be tracked only by decreasing the parameter α along a nonoptimal segment. After α decreases from 0.8700583 to 0.8700568 the path of stationary points turns smoothly and continues for increasing values of α, becoming optimal again. The two components of the objective function, the structural weight W and the control force $F_{\max}$, at the beginning of Segment 8 are shown in Figs. 6 and 7, respectively. The scale in Figs. 6

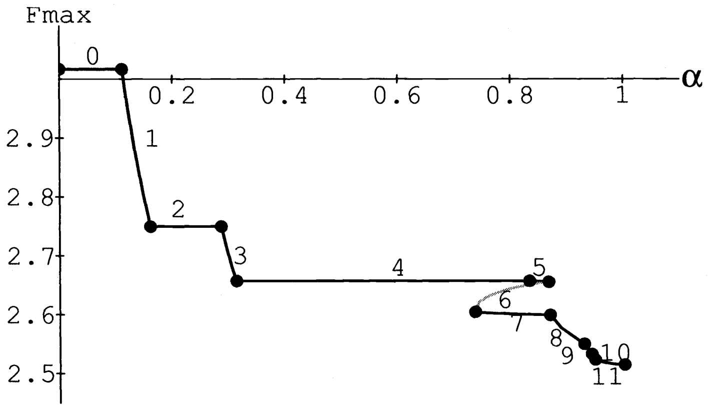


FIG. 4. *$F_{\max}$ (pounds) along Segments 0–11 (gray line denotes stationary nonoptimal points, black line denotes optimal points).*

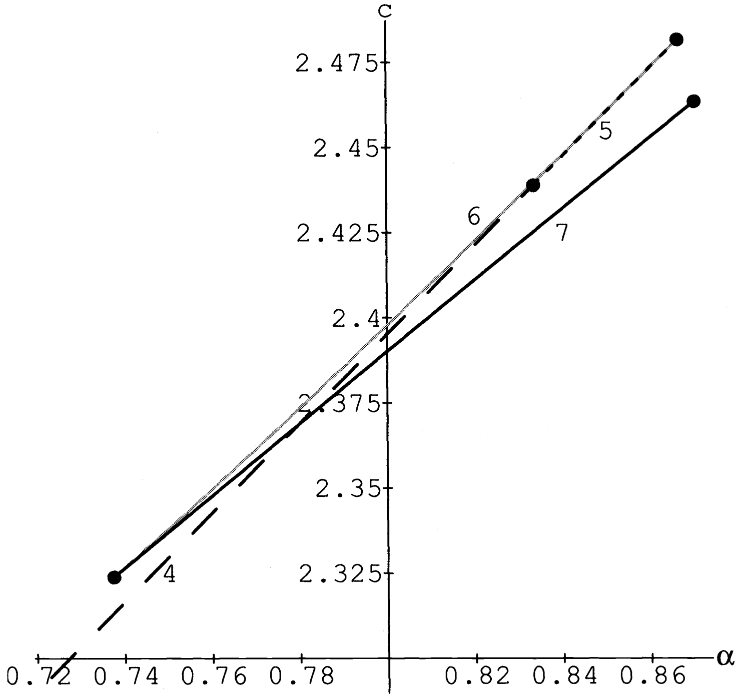


FIG. 5. *Objective function c along Segments 4–7; black lines (4: dashed, 5: dotted, 7: solid) denote optimal solutions, gray line (6) denotes nonoptimal stationary points.*

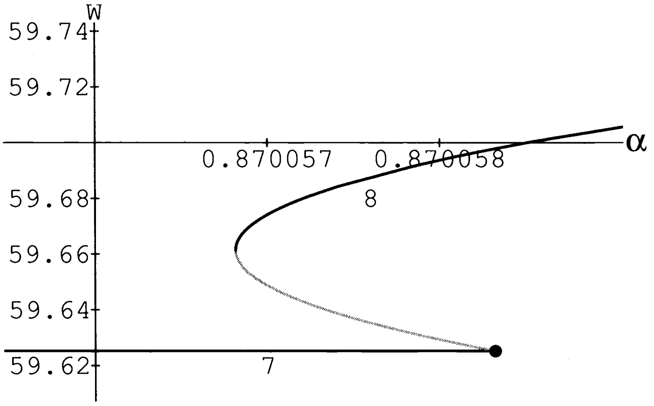


FIG. 6. *Weight W at the beginning of Segment* 8 (*black line denotes optimal solutions, gray line denotes stationary nonoptimal points*).

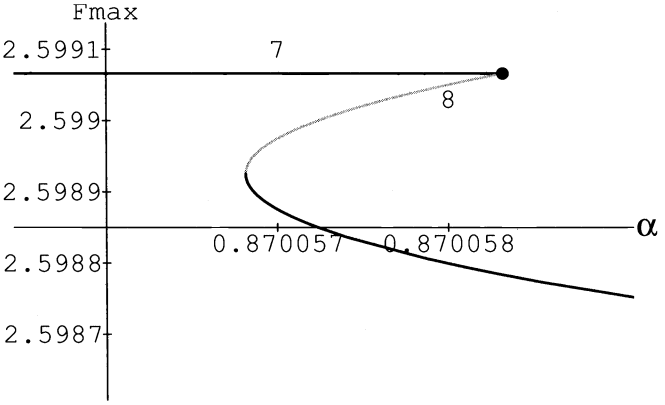


FIG. 7. *$F_{\max}$ at the beginning of Segment* 8 (*black line denotes optimal solutions, gray line denotes stationary nonoptimal points*).

and 7 indicates that the solution undergoes extreme changes in that region with the logarithmic derivative of the weight with respect to α (percent change in W divided by percent change in α) being of the order of 300. This requires tracing the curve with high accuracy.

A similar behavior of the objective function is observed at the beginning of Segment 9. The path of stationary points exists only for decreasing values of α. The path turns smoothly after α decreases by about 0.00013 and continues for increasing values of α. Points corresponding to decreasing values of α are again nonoptimal points satisfying the first order necessary conditions.

6. Concluding remarks. An active set algorithm for tracing parametrized optima was shown to be effective in tracing the efficient curve in bi-objective optimization. Interesting results were obtained for the combined control-structure optimization of a ten-bar truss. In particular it was found that the efficient curve is discontinuous and has both low and extremely high variations. Furthermore, for this example, nonoptimal segments of the curve of stationary solutions bridged the discontinuities of the efficient curve and thus served as an easy way to continue the tracing process at such discontinuities.

REFERENCES

[1] E. L. ALLGOWER AND K. GEORG, *Introduction to Numerical Continuation Methods*, Springer-Verlag, Berlin, Heidelberg, New York, 1990.

[2] R. L. CRANE, K. E. HILLSTROM, AND M. MINKOFF, *Solution of the general nonlinear programming problem with subroutine VMCON*, Argonne National Lab. Tech. Report ANL-80-64, Argonne, IL, 1980.

[3] H. ESCHENANER, J. KOSKI, A. OSYCZKA, *Multicriteria Design Optimization*, Springer-Verlag, New York, 1990.

[4] R. T. HAFTKA, Z. GÜRDAL, AND M. P. KAMAT, *Elements of Structural Optimization*, 2nd ed., Kluwer, Dordrecht, the Netherlands, 1990.

[5] R. T. HAFTKA, *Integrated structures-controls optimization of space structures*, Proc. AIAA Dynamics Specialist Conference, Long Beach, CA, April 5-6, 1990, pp. 1–9.

[6] K. LIM AND J. JUNKINS, *Robust optimization of structural and controller parameters*, J. Guidance, 12 (1989), pp. 89–96.

[7] B. N. LUNDBERG AND A. B. POORE, *Bifurcations and sensitivity in parametric programming*, Proc. Third Air Force/NASA Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, San Francisco, CA, September 24–26, 1990, pp. 50–55.

[8] Z. N. MARTINOVIC, G. SCHAMEL, R. T. HAFTKA, AND W. L. HALLAUER, *An analytical and experimental investigation of output feedback vs. linear quadratic regulator*, J. Guidance, 13 (1990), pp. 160–167.

[9] M. MILMAN, R. E. SCHEID, M. SALAMA, AND R. BRUNO, *Methods for combined control-structure optimization*, Proc. Conference on Dynamics and Control of Large Structures, Blacksburg, VA, May 8–10, 1989, pp. 191–206.

[10] J. RAKOWSKA, R. T. HAFTKA, AND L. T. WATSON, *An active set algorithm for tracing parametrized optima*, Structural Optimization, 3 (1991), pp. 29–44.

[11] J. R. J. RAO AND P. Y. PAPALAMBROS, *Extremal behavior of one parameter families of optimal design models*, Proc. ASME Design Automation Conference, Montreal, Quebec, Canada, Sept. 17–20, 1989, pp. 91–100.

[12] ———, *A nonlinear programming continuation strategy for one parameter design optimization problems*, Proc. ASME Design Automation Conference, Montreal, Quebec, Canada, Sept. 17–20, 1989, pp. 77–89.

[13] M. SALAMA AND J. GARBA, *Simultaneous optimization of controlled structures*, Comput. Mech., 3 (1988), pp. 275–282.

[14] H. WACKER, *Continuation Methods*, Academic Press, New York, 1978.

[15] L. T. WATSON, *Numerical linear algebra aspects of globally convergent homotopy methods*, Tech. Report TR-85-14, Dept. of Computer Sci., VPI&SU, Blacksburg, VA, 1985; SIAM Rev., 28 (1986), pp. 529–545.

[16] ———, *A globally convergent algorithm for computing fixed points of C^2 maps*, Appl. Math. Comput., 5 (1979), pp. 297–311.

[17] L. T. WATSON, S. C. BILLUPS, AND A. P. MORGAN, *HOMPACK: A suite of codes for globally convergent homotopy algorithms*, Tech. Report 85-34, Dept. of Industrial and Operations Engrg., Univ. of Michigan, Ann Arbor, 1985; ACM Trans. Math. Software, 13 (1987), pp. 281–310.

SIAM J. OPTIMIZATION
Vol. 3, No. 4, pp. 669–687, November 1993

001

CONVEX FUNCTIONS WITH UNBOUNDED LEVEL SETS AND APPLICATIONS TO DUALITY THEORY*

A. AUSLENDER[†], R. COMINETTI[‡], AND J.-P. CROUZEIX[†]

Abstract. A class of convex functions with unbounded level sets but good behavior at infinity [*Analyse non-linéaire*, Gauthier-Villars, Paris, 1989, pp. 101–122] is investigated. Characterizations and properties are given. The results are then applied to studying sequential approximation schemes for optimization problems and to duality theory, when the involved functions have unbounded level sets. In particular, the convergence properties of stationary sequences for the dual of a convex program are studied, and methods for associating with it a primal sequence converging to a solution of the primal problem are demonstrated.

Key words. convex optimization, duality, inf-compactness, good asymptotic behavior, relaxed constraint qualification, algorithms

AMS subject classifications. 90C25, 90C31, 90C34, 49J52, 49M45, 49N15

1. Introduction.

Let us consider an abstract optimization problem

$$\min_{x\in X} f(x),$$

where X denotes a finite-dimensional euclidean space and f is a closed proper convex function on X.

Many algorithms for solving such a problem will only generate a *stationary* sequence x_n, that is, a sequence satisfying

$$d(0, \partial f(x_n)) \to 0,$$

where as usual $d(y, S)$ denotes the distance from the point $y \in X$ to the set $S \subset X$. The natural question which arises is whether such a sequence will also be minimizing or not, and the answer is generally no, as shown in [2]. In fact, in that paper, Auslender and Crouzeix addressed precisely the problem of characterizing the class of functions for which all stationary sequences are minimizing: the so-called *asymptotically well behaved convex functions.*

In the present paper we identify and investigate a subclass of the asymptotically well behaved convex functions which appears to be of great relevance in theory and applications, namely, the class $\mathcal{R}$ of closed convex functions for which 0 belongs to the relative interior of the domain of its Fenchel conjugate.

As we show in the next section, for such functions we have more than a merely good asymptotical behavior: the set of minima is nonempty and stationary sequences are not only minimizing, but they converge towards this set.

A first application showing the relevance of the class $\mathcal{R}$ in applications is also presented in §2, and concerns a general convergence theorem for monotone approximation schemes for optimization problems. We obtain as a particular case a significant result of [4] in semi-infinite linear programming.

The second application, developed in full detail in §3 and particularized to special classes of mathematical programs in §4, is related to duality theory. Duality in convex

* Received by the editors April 29, 1991; accepted for publication (in revised form) July 12, 1992.

[†] Department of Applied Mathematics, Université Blaise Pascal, 63177 Aubière Cédex, France.

[‡] Universidad de Chile, Casilla 170/3 Correo 3, Santiago, Chile. This work was partially supported by FONDECYT 92–0947.

programming is a very powerful technique, both theoretically and computationally. Typically, to solve a *primal* problem

$$(P) \quad \alpha = \inf_{x \in X} f(x),$$

one considers a suitable perturbation function $\varphi : X \times U \to \overline{\mathbb{R}}$ such that $\varphi(\cdot, 0) = f$, with which one associates the *dual* problem

$$(D) \quad \beta = \inf_{u^* \in U^*} h^*(u^*),$$

where h^* is the Fenchel conjugate of the marginal value function $h(u) = \inf_{x \in X} \varphi(x, u)$. The hope is that (D) will be easier to solve than (P) and once an optimal solution for the dual is found, an optimal solution for (P) can be recovered from it.

We notice that the assumption $0 \in \mathrm{ri}[\mathrm{dom}(h)]$, which in duality theorems ensures the equality $\alpha = -\beta$ and the existence of dual optimal solutions, amounts to $h^* \in \mathcal{R}$. This observation allows us to complement the classical duality results by asserting that the functional to be minimized in the dual problem is asymptotically well behaved and, moreover, every dual stationary sequence approaches the dual optimal set which is a compact set up to an orthogonal subspace. We show in Theorem 3.1 that these nice properties are also shared by the "perturbed" dual problems

$$k(x^*) = \inf_{u^* \in U^*} \varphi^*(x^*, u^*),$$

while in Theorem 3.2 we investigate the stability of the optimal solution set of these problems.

These results represent significant extensions of the ones presented in [1] where it is shown that in three different practical problems, namely those considered by Tseng and Bertsekas [7], Censor and Lent [3], and Han and Lou [5], a decomposition algorithm is used for the minimization of certain dual functionals which fail to be inf-compact but are asymptotically well behaved.

Another point observed in [1] is that to each stationary sequence u_n^* for the dual functional, one could associate a primal sequence x_n converging to the solution of the primal problem. We shall extend these results to the general framework of duality theory in convex programming, as an application of Theorem 3.2.

In §4 we sketch the application and meaning of the previous results in some particular duality schemes that often arise in practice: vertical perturbations, Fenchel duality, and linearly constrained decomposable problems. In particular we show how the cases of Tseng and Bertsekas, Censor and Lent, and Han and Lou are recovered.

We complement all the previous results in §5 with a brief discussion of some algorithmical issues.

Also, a general discussion of the class of asymptotically well behaved convex functions is included in the final section, where we improve the results already presented in [2], avoiding some technical hypotheses considered in the cited paper, and also giving simplified proofs of some statements.

In the sequel we shall be working in a *finite*-dimensional setting so that here and afterwards X and U will represent arbitrary finite-dimensional euclidean spaces. We shall assume a certain familiarity with convex analysis, for which we shall basically follow Rockafellar [6]. In particular we shall denote $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ the usual inner product and norm in X and U, and $B(x, r)$ the ball centered at x with radius r.

2. An important subclass of asymptotically well behaved functions. Given a convex function $f \in \Gamma(X)$, we say that the sequence $x_k \in X$ is *stationary* for f if

$$d(0, \partial f(x_k)) \to 0,$$

that is, if we can find $x_k^* \in \partial f(x_k)$ such that $x_k^* \to 0$. Then we recall [2] that $f \in \Gamma(X)$ is said to be *asymptotically well behaved* if every stationary sequence $x_k \in X$ is minimizing, that is,

$$\lim_{k \to \infty} f(x_k) = \inf_{x \in X} f(x).$$

This class of functions will be denoted by $\mathcal{F}$.

In this section we shall present an important subclass of asymptotically well behaved functions which enjoy the additional properties:

(a) its set of minima is nonempty,

(b) every stationary sequence converges towards this set.

This subclass, admitting a very simple characterization, appears in fact very frequently in convex analysis and particularly in duality theory. Let us begin with some preliminary results.

2.1. Preliminaries. Let a (not necessarily closed) convex function $h : U \to \overline{\mathbb{R}}$ be given and denote by E the affine hull of its domain. We shall assume throughout that $0 \in \operatorname{dom}(h)$ so that E is in fact a vector subspace of U. Following [4] we associate with h the function h_E given by

$$h_E(u) = h(\Pi_E u),$$

where Π_E denotes the orthogonal projector from U onto E.

LEMMA 2.1. *With the previous notation we have,*

(a) $\operatorname{dom}(h_E) = \operatorname{dom}(h) + E^\perp$ *and* $\operatorname{int}[\operatorname{dom}(h_E)] = \operatorname{ri}[\operatorname{dom}(h)] + E^\perp$.

(b) $h^*(u^*) = h_E^*(\Pi_E u^*)$.

(c) $\partial h_E(u) = \partial h(\Pi_E u) \cap E$.

(d) $\partial h(u) = \begin{cases} \partial h_E(u) + E^\perp & \textit{if } u \in E \\ \phi & \textit{otherwise.} \end{cases}$

Proof. Properties (a), (c), and (d) were proved in [4] assuming that h is closed, but the proof remains valid without this hypothesis. An alternative proof may be based on property (b), which we show next by direct calculation:

$$\begin{aligned} h^*(u^*) &= \sup_{u \in E} \langle u^*, u\rangle - h(u) \\ &= \sup_{u \in U} \langle u^*, \Pi_E u\rangle - h(\Pi_E u) \\ &= \sup_{u \in U} \langle \Pi_E u^*, u\rangle - h_E(u) = h_E^*(\Pi_E u^*). \qquad \square \end{aligned}$$

The advantage of working with the function h_E instead of h is that the former is continuous on the interior of its domain, which is now *nonempty.* We obtain for instance that when h is finite at least at one point of ri[dom (h)], then it never takes the value $-\infty$, so that h_E is proper and continuous on the interior of its domain. Also, the multifunction $u \to \partial h_E(u)$ is nonempty compact valued and upper semicontinuous on this set. As a consequence, we get the following result.

PROPOSITION 2.2. *Suppose* $0 \in \operatorname{ri}[\operatorname{dom}(h)]$. *Then, for every sequence* $u_n \to 0$ *and* $u_n^* \in \partial h(u_n)$ *we have*

$$d(u_n^*, \partial h(0)) \to 0 \quad \textit{and} \quad h^*(u_n^*) \to -h(0) = \inf h^*(u^*).$$

Proof. The proof being obvious when $h(0) = -\infty$, we just consider the case when $h(0)$ is finite. Since $\partial h(u_n)$ is not empty we must have $u_n \in E$, and from Lemma 2.1 we get $\Pi_E u_n^* \in \partial h_E(u_n)$. The upper semicontinuity of $\partial h_E(\cdot)$ at $0 \in \operatorname{int}[\operatorname{dom}(h_E)]$ yields at once

$$d(u_n^*, \partial h(0)) = d(\Pi_E u_n^*, \partial h_E(0)) \to 0$$

as well as the boundedness of the sequence $\Pi_E u_n^*$. Therefore, since $u_n \in E$ we have that $h^*(u_n^*) + h(u_n) = \langle \Pi_E u_n^*, u_n \rangle$ tends to zero, and since h is continuous relative to $\operatorname{ri}[\operatorname{dom}(h)]$ and in particular at 0, we conclude $h^*(u_n^*) \to -h(0)$ as claimed. □

2.2. The subclass $\mathcal{R}$. The results in the previous subsection tell us that if a function h satisfies $0 \in \operatorname{ri}[\operatorname{dom}(h)]$ (or equivalently $0 \in \operatorname{ri}[\operatorname{dom}(h^{**})]$), then its Fenchel conjugate h^* is asymptotically well behaved, but something else as well: if u_n^* is a stationary sequence for h^* then it is not only minimizing ($h^*(u_n^*) \to \inf h^*(u^*)$) but it approaches the solution set Argmin $h^*(u^*) = \partial h(0)$ which is a nonempty compact set up to an orthogonal subspace.

These observations lead us to consider the class $\mathcal{R}$ of functions $f \in \Gamma(X)$ such that

$$(1) \qquad 0 \in \operatorname{ri}[\operatorname{dom}(f^*)],$$

which is a subclass of the asymptotically well behaved convex functions defined on X. In view of [6, Cor. 13.3.4 b], we may also characterize this class by the equivalent property

$$(2) \qquad f_\infty(v) > 0 \quad \text{for all } v \in L_f^\perp, v \neq 0,$$

where f_∞ denotes the recession function of f (cf. [6, p. 70]), and L_f the constancy space of f, $L_f = \{v \in X : f_\infty(v) = f_\infty(-v) = 0\}$. Let us recall here the characteristic property of L_f (cf. [6, Thm. 8.8]), namely,

$$f(x+v) = f(x) \quad \forall\, v \in L_f, \quad \forall\, x \in \operatorname{dom}(f).$$

Let us also point out that the functions f satisfying (1) can also be characterized (see Lemma 2.1(b)) as images of inf-compact functions under linear transformations, namely, $f(x) = \bar{f}(\Pi_E x)$ where $E = \operatorname{aff}(\operatorname{dom} f^*)$ and $\bar{f} = (f^* \circ \Pi_E)^*$, which is indeed inf-compact.

As we shall see in the sequel, both characterizations (1) and (2) turn out to be useful depending on the particular features of the problem at hand.

The previous results and discussion show that the stationary sequences of functions in $\mathcal{R}$ enjoy very interesting features. However, many algorithms do not construct stationary sequences but *approximate* stationary sequences. The following theorem summarizes the above discussion and can also be used to handle these approximate stationary sequences. We recall that the ε-subgradient of a convex function f is defined as

$$\partial_\varepsilon f(x) = \{x^* \in X^* : f(y) \geq f(x) + \langle x^*, y - x\rangle - \varepsilon \ \ \forall\, y \in X\},$$

and that x is an ε-minimum of f if and only if $0 \in \partial_\varepsilon f(x)$.

THEOREM 2.3. *Let $f \in \mathcal{R}$. Then the problem*

$$(P) \quad \alpha = \inf_{x \in X} f(x)$$

has a nonempty optimal solution set of the form $S = K + E^\perp$ where $E = \text{aff}[\text{dom}(f^*)]$ and K is a compact subset of E.

Moreover, if x_n is a sequence such that $d(0, \partial_{\varepsilon_n} f(x_n)) \to 0$ where $\varepsilon_n \to 0$ (possibly $\varepsilon_n = 0$), then

(a) *$f(x_n) \to \alpha$,*

(b) *$d(x_n, S) \to 0$,*

(c) *the sequence $\Pi_E x_n$ is bounded and all its cluster points belong to $S \cap E = K$.*

Proof. From the discussion that motivated the introduction of $\mathcal{R}$, the result holds when $\varepsilon_n = 0$ (property (c) follows from (b) given that $d(x_n, S) = d(\Pi_E x_n, K)$).

Let us then consider the general case and select $x_n^* \in \partial_{\varepsilon_n} f(x_n)$ with $x_n^* \to 0$. Using the Bronsted–Rockafellar theorem we may find y_n such that $\|y_n - x_n\| \le \sqrt{\varepsilon_n}$ and $y_n^* \in \partial f(y_n)$ with $\|y_n^* - x_n^*\| \le \sqrt{\varepsilon_n}$. It follows that $y_n^* \to 0$ so, applying this result (in the case $\varepsilon_n = 0$) to the sequence y_n, we get $d(y_n, S) \to 0$ and $f(y_n) \to \alpha$. From this it is clear that (b) must hold, and since

$$\alpha \le f(x_n) \le f(y_n) - \langle x_n^*, y_n - x_n \rangle + \varepsilon_n,$$

assertion (a) follows immediately. □

To conclude this subsection, let us point out that often the functions f that are to be manipulated appear by applying different operations to other simpler functions. One of the most interesting operations is the infimal convolution. We provide next a criteria for verifying if a function expressed as the infimal convolution of two convex functions belongs to $\mathcal{R}$ or not.

PROPOSITION 2.4. *Let $f, g \in \Gamma(X)$ with $g \in \mathcal{R}$, and denote $h = f \nabla g$ their infimal convolution. Then*

(a) *if $f \in \mathcal{R}$ then $h \in \mathcal{R}$;*

(b) *conversely, if g is co-finite and $h \in \mathcal{R}$ then $f \in \mathcal{R}$;*

(c) *in both cases $L_h = L_f + L_g$.*

Proof. From the equality $h^* = f^* + g^*$ we get $\text{dom}(h^*) = \text{dom}(f^*) \cap \text{dom}(g^*)$, and therefore using [6, Thm. 6.5] we deduce both in (a) and (b) that

$$\text{ri}[\text{dom}(h^*)] = \text{ri}[\text{dom}(f^*)] \cap \text{ri}[\text{dom}(g^*)],$$

so the first two claims are obvious (recall that g is co-finite if and only if $\text{dom}(g^*) = X^*$). Property (c) follows directly from the previous formula for $\text{dom}(h^*)$ and the characterization $L_h = \text{dom}(h^*)^\perp$, as well as the corresponding characterizations for L_f and L_g. □

2.3. Applications. Examples of functions belonging to the class $\mathcal{R}$ appear naturally in the setting of duality theory. Since this application is so important we shall develop it in full detail in the next two sections. Let us turn instead to another setting where the functions in $\mathcal{R}$ appear to be of importance. Namely, in various optimization techniques as penalty methods or finite-dimensional approximation to semi-infinite optimization problems, an original problem of the form

$$\min_{x \in X} f(x)$$

is replaced by an infinite sequence of simpler subproblems

$$\min_{x\in X} f_k(x).$$

It is often the case that the sequence f_k monotonically increases towards f. We shall investigate the relationship between the original and the approximate problems, under the assumption $f \in \mathcal{R}$. To this end it is useful to have the following result (which can also be obtained trivially using epiconvergence theory).

LEMMA 2.5. *Let $f_k, f \in \Gamma_0(X)$ with f_k a nondecreasing sequence converging pointwise to f. Then, if $x_k \to x$ we have*

$$f(x) \leq \liminf_{k\to\infty} f_k(x_k).$$

Proof. For each neighborhood V of x we have $\inf_{y\in V} f_k(y) \leq f_k(x_k)$ for all k large enough, and since the left-hand side is nondecreasing with k we may pass to the limit in order to obtain

$$\sup_k \inf_{y\in V} f_k(y) \leq r = \liminf_{k\to\infty} f_k(x_k).$$

This inequality holds for all neighborhoods V of x so that

$$\sup_k \sup_V \inf_{y\in V} f_k(y) \leq r$$

and the lower semicontinuity of f_k gives $f(x) = \sup_k f_k(x) \leq r$. □

THEOREM 2.6. *Let $f_k \in \Gamma_0(X)$ be a nondecreasing sequence and $f = \sup_k f_k$.*

(1) *The sequence f_k^* is nonincreasing and $f^* = \mathrm{cl}(\inf_k f_k^*)$. Moreover, the linear manifolds E_k =aff[dom(f_k^*)] coincide, for large k, with E =aff[dom(f^*)].*

(2) *If $f \in \mathcal{R}$ then $f_k \in \mathcal{R}$ for large k. Moreover, if x_k is an ε_k-minimizer of f_k with $\varepsilon_k \to 0$ then we have (with S the minimizing set of f) that*

(a) *$f_k(x_k) \to m = \min_{x\in X} f(x)$,*

(b) *$d(x_k, S) \to 0$,*

(c) *the sequence $\Pi_E x_k$ is bounded and all its cluster points belong to $S \cap E$.*

Proof. (1) Since Fenchel conjugacy reverses the inequalities ($g \leq h \Rightarrow g^* \geq h^*$) it follows at once that f_k^* is nonincreasing. Now, by direct computation

$$(\inf_k f_k^*)^* = \sup_k f_k^{**} = \sup_k f_k = f,$$

so that taking Fenchel conjugate we deduce $f^* = \mathrm{cl}(\inf_k f_k^*)$.

From this characterization and noting that a convex function and its closure have the same affine hull of their corresponding domains (cf. [6, Cor. 7.4.1]) we deduce

$$E = \mathrm{aff}[\mathrm{dom}(\inf_k f_k^*)] = \mathrm{aff}\left[\bigcup_k \mathrm{dom}(f_k^*)\right] = \bigcup_k \mathrm{aff}[\mathrm{dom}(f_k^*)] = \bigcup_k E_k.$$

Since f_k^* is nonincreasing, the sets E_k form a nondecreasing sequence of linear manifolds contained in E and whose union gives all of E. The conclusion follows easily: all the E_k must coincide with E for k large enough.

(2) Since a convex function and its closure have the same relative interior of their domains (cf. [6, Cor. 7.4.1]) we deduce from (1),

$$\mathrm{ri}[\mathrm{dom}(f^*)] = \mathrm{ri}\left[\bigcup_k \mathrm{dom}(f_k^*)\right].$$

Thus, $f \in \mathcal{R}$ implies $0 \in \mathrm{ri}[\bigcup_k \mathrm{dom}(f_k^*)]$, which can also be written as

$$E = \mathbb{R}_+ \left[\bigcup_k \mathrm{dom}(f_k^*)\right] = \bigcup_k \mathbb{R}_+ \mathrm{dom}(f_k^*).$$

Again the linear manifold E has been expressed as a monotone union of convex cones $\mathbb{R}_+\mathrm{dom}(f_k^*)$, so that for k large enough we must have the equality $\mathbb{R}_+\mathrm{dom}(f_k^*) = E = E_k$, which amounts precisely to $0 \in \mathrm{ri}[\mathrm{dom}(f_k^*)]$, that is, $f_k \in \mathcal{R}$.

We arrive at the most interesting part of the theorem. Let us take a sequence x_k of ε_k-minimizers of f_k. Thus we have $0 \in \partial_{\varepsilon_k} f_k(x_k)$, and then

$$(3) \qquad 0 \leq f_k(x_k) + f_k^*(0) \leq \varepsilon_k.$$

Since $0 \in \mathrm{ri}[\mathrm{dom}(f^*)]$ we deduce from [6, Thm. 7.4] that $f_k^*(0) \to f^*(0)$, which combined with the previous inequality yields $f_k(x_k) \to -f^*(0) = m$, proving (a).

To show the boundedness of $\Pi_E x_k$ we choose $r > 0$ so that the set $B = \{y \in E : \|y\| \leq r\}$ is contained in $\mathrm{ri}[\mathrm{dom}(f^*)]$, and we use [6, Thm. 10.8] to assert that f_k^* converges uniformly to f^* on B. Now, since $E^\perp = E_k^\perp$ is the constancy space of f_k for large k, we get $f_k(\Pi_E x_k) = f_k(x_k)$ and using (3) we deduce $\Pi_E x_k \in \partial_{\varepsilon_k} f_k^*(0)$, so that we may write

$$f_k^*(0) + \langle \Pi_E x_k, y\rangle \leq f_k^*(y) + \varepsilon_k,$$

which used with $y = y_k = r\Pi_E x_k / \|\Pi_E x_k\|$ gives us

$$\|\Pi_E x_k\| \leq \frac{1}{r}[f_k^*(y_k) + \varepsilon_k - f_k^*(0)] \leq \frac{1}{r}[\sup_{y\in B} f_k^*(y) + \varepsilon_k - f_k^*(0)].$$

The uniform convergence of f_k^* towards f^* on B, and the continuity of the latter relative to $\mathrm{ri}[\mathrm{dom}(f^*)]$ allows us to conclude that the right-hand side above stays bounded.

Now, take any cluster point x_E of $\Pi_E x_k$ and assume with no loss of generality that in fact it converges to it. Since $f_k(\Pi_E x_k) = f_k(x_k)$ we obtain from (3)

$$f_k(\Pi_E x_k) \leq \varepsilon_k - f_k^*(0),$$

so letting $k \to \infty$ and using the previous lemma we conclude that

$$f(x_E) \leq \liminf_{k\to\infty} f_k(\Pi_E x_k) \leq \lim_{k\to\infty} \varepsilon_k - f_k^*(0) = -f^*(0) = m$$

and x_E is a minimum of f. This proves (c).

Since (b) is a simple consequence of (c) and the structure of $S = \partial f_E(0) + E^\perp$, the theorem has been proved. □

In §5 we shall briefly discuss how the previous result applies to penalty methods. As another application let us consider as in [4] the linear semi-infinite program

$$(P) \quad \alpha = \min\{\langle c, x\rangle : \langle a_t, x\rangle \leq b_t, t \in T\},$$

where c, a_t, x belong to $\mathbb{R}^n$, $b_t \in \mathbb{R}$, and T represents an arbitrary index set.

If we denote by M the homogeneous moment cone generated by the a_t's, that is,

$$M = \operatorname{cone}\{a_t : t \in T\},$$

then the main results in [4] are obtained under the assumption

$$(H) \quad -c \in \operatorname{ri}(M).$$

As a matter of fact this condition ensures also that the function to be minimized in (P) belongs to the class $\mathcal{R}$. More precisely, we have the following.

PROPOSITION 2.7. *Let $C_t = \{x : \langle a_t, x\rangle \le b_t\}$, $C = \bigcap_{t\in T} C_t$, and $f(x) = \langle c, x\rangle + \mathcal{X}_C(x)$. Then $f \in \mathcal{R}$ if and only if condition (H) is satisfied.*

Proof. By general calculus rules of asymptotic functions and cones we have $f_\infty(d) = \langle c, d\rangle + \mathcal{X}_{C_\infty}(d)$. Since $C_\infty = \bigcap_{t\in T}(C_t)_\infty$ and $(C_t)_\infty = \{d : \langle a_t, d\rangle \le 0\}$ it follows that

$$L_f = \{d : \langle c, d\rangle = 0; \langle a_t, d\rangle = 0 \ \forall\ t \in T\}.$$

From this equality it follows that $L_f^\perp$ is the linear space E generated by $\{c\} \cup \{a_t : t \in T\}$, so that the condition $f \in \mathcal{R}$ expressed in its form (2) turns out to be in this case

$$\langle c, d\rangle > 0 \quad \text{for all } d \in C_\infty \cap E, \quad d \neq 0,$$

which is in turn, by [4, Lemma 2.1], equivalent to hypothesis (H). □

Now, it has also been shown (see [4, §4] and references therein) that under the assumption (H) problem (P) is *discretizable* in the sense that there exists a sequence of *finite* subproblems

$$(P_k) \quad \alpha_k = \min\{\langle c, x\rangle : \langle a_t, x\rangle \le b_t, t \in T_k\}$$

with T_k an increasing sequence of finite subsets of T such that $\cup T_k = T$ and $\alpha_k \to \alpha$. If we denote by C_k the corresponding feasible set of problem (P_k), that is,

$$C_k = \{x : \langle a_t, x\rangle \le b_t, t \in T_k\},$$

then the functions $f_k(x) = \langle c, x\rangle + \mathcal{X}_{C_k}(x)$ form a nondecreasing sequence converging towards $f(x) = \langle c, x\rangle + \mathcal{X}_C(x)$ and the previous theorem applies: for k large enough, problem (P_k) has solutions and if x_k solves problem (P_k), then

(a) $\langle c, x_k\rangle \to \alpha$,

(b) $d(x_k, S) \to 0$ where S is the solution set of (P), and

(c) the sequence $\Pi_E x_k$ is bounded and all its cluster points are solutions for (P).

These claims amount essentially to [4, Thm. 4.1].

3. Application to duality theory. Let us see how the previous results apply in the context of convex duality. Let us then fix a closed proper convex function $\varphi : X \times U \to \overline{\mathbb{R}}$ and consider the marginal value function

$$h(u) = \inf_{x\in X} \varphi(x, u)$$

corresponding to the perturbation φ of the *primal* problem

$$(P) \quad \alpha := \inf_{x\in X} \varphi(x, 0).$$

It is well known that the conjugate of h is given by

$$h^*(u^*) = \varphi^*(0, u^*)$$

so that the *dual* problem

$$(D) \quad \beta := \inf_{u^* \in U^*} \varphi^*(0, u^*)$$

is closely linked with (P) as $\alpha = h(0)$ and $\beta = -h^{**}(0)$. In fact, $\alpha \geq -\beta$, and when $\alpha < +\infty$ the equality holds if and only if h is lower semicontinuous at 0. In this case the optimal solution set of (D) is given by U^* when $\beta = +\infty$ or $\partial h(0)$ when $\beta < +\infty$.

The standard duality result asserts that $\alpha = -\beta$ and (D) has optimal solutions whenever $0 \in \mathrm{ri}[\mathrm{dom}(h)]$. In this sense, Theorem 2.3 complements this result in the following way: the function $\varphi^*(0, \cdot)$ to be minimized in the dual problem has good behavior at infinity, every stationary sequence for the dual is a minimizing sequence which approaches the optimal solution set of (D), and this solution set is either the whole space U^* when $\beta = +\infty$ (which corresponds to an unfeasible dual), or a compact set up to an orthogonal vector subspace otherwise.

Furthermore, all these properties hold not only for $\varphi^*(0, \cdot)$ but for $\varphi^*(x^*, \cdot)$, that is, for all the *perturbed* dual problems

$$k(x^*) = \inf_{u^* \in U^*} \varphi^*(x^*, u^*).$$

To see this it suffices to apply the same reasoning not to h but to

$$h^{x^*}(u) = \inf_{x \in X} \{\varphi(x, u) - \langle x^*, x \rangle\}.$$

Indeed, the conjugate function of h^{x^*} is precisely $\varphi^*(x^*, \cdot)$. On the other hand, since $\mathrm{dom}(h^{x^*}) = \mathrm{dom}(h)$, one has $0 \in \mathrm{ri}[\mathrm{dom}(h^{x^*})]$ as soon as $0 \in \mathrm{ri}[\mathrm{dom}(h)]$ and this allows us to apply the previous results to h^{x^*}. Moreover, it also follows that the space $E = \mathrm{aff}[\mathrm{dom}(h^{x^*})]$ *does not* depend on x^* and that $E^\perp$ is the constancy space of all the functions $\varphi^*(x^*, \cdot)$, that is,

$$\varphi^*(x^*, u^* + v) = \varphi^*(x^*, u^*) \quad \forall\, v \in E^\perp. \tag{4}$$

We summarize this discussion in the following theorem.

THEOREM 3.1. *With the previous notation and assuming* $0 \in \mathrm{ri}[\mathrm{dom}(h)]$, *for each* $x^* \in X^*$ *it holds that*

$$-h^{x^*}(0) = k(x^*) = \min_{u^* \in U^*} \varphi^*(x^*, u^*),$$

the minimum being attained, with (nonvoid) optimal solution set given by

$$S(x^*) = \begin{cases} U^* & \text{if } k(x^*) = +\infty, \\ \partial h^{x^*}(0) = \partial h_E^{x^*}(0) + E^\perp & \text{otherwise.} \end{cases} \tag{5}$$

Moreover, every stationary sequence u_n^* *for* $\varphi^*(x^*, \cdot)$ *is minimizing and converges to* $S(x^*)$, *that is,*

$$\begin{aligned} &\text{(i)} \quad \varphi^*(x^*, u_n^*) \to k(x^*), \\ &\text{(ii)} \quad d(u_n^*, S(x^*)) \to 0. \end{aligned}$$

The same holds true merely if $d(0, \partial_{\varepsilon_n}\varphi(x^*,\cdot)(u_n^*)) \to 0$ *where* $\varepsilon_n \to 0$. □

This theorem is concerned with the perturbed dual problems, but $x^* \in X^*$ is considered fixed. Now, for an algorithmic approach of the dual problem (D), we must also study the upper-semicontinuity of the optimal solution set $S(x^*)$ at $x^* = 0$. This type of continuity is hopeless in general as the optimal solution sets may be unbounded. Nevertheless, the projection onto E of this set-valued map has the desired continuity relative to ri[dom(k)]. In the sequel we shall denote $F = \text{aff}[\text{dom}(k)]$.

In sequential terms, this upper-semicontinuity corresponds to the following situation: we take any sequence $x_n^* \to 0, x_n^* \in \text{dom}(k) \subset F$, and solve the sequence of problems

$$(D_n) \qquad \min_{u^* \in U^*} \varphi^*(x_n^*, u^*).$$

Then, we look for conditions ensuring that any solution $u_n^* \in S(x_n^*)$ will satisfy

(a) $\varphi^*(x_n^*, u_n^*) \to k(0) = \beta$,
(6) (b) $d(u_n^*, S(0)) \to 0$,
(c) the sequence $\Pi_E u_n^*$ is bounded and all its limit points belong to $S(0) \cap E$.

In particular, when $S(0) \cap E$ is reduced to a singleton we will have convergence of the whole sequence $\Pi_E u_n^*$ towards this particular solution. The following result gives such conditions, and moreover it can handle approximate minimization of the problems (D_n), which may be of algorithmic relevance.

THEOREM 3.2. *Suppose* $0 \in \text{ri}[\text{dom}(h)]$ *and* $\beta < \infty$. *If* $x_n^* \to 0, x_n^* \in F$ *with* $k(x_n^*) \to k(0)$ (*in particular, if* $0 \in \text{ri}[\text{dom}(k)]$), *and* u_n^* *is an* ε_n*-minimum for* $\varphi^*(x_n^*, \cdot)$ *where* $\varepsilon_n \to 0$; *then properties* (6) (a), (b), *and* (c) *hold.*

Proof. Since $k(x_n^*) \le \varphi^*(x_n^*, u_n^*) \le k(x_n^*) + \varepsilon_n$ and $k(x_n^*) \to k(0) = \beta$, assertion (6)(a) is immediate. In order to prove (6)(b) and (c) we observe that $d(u_n^*, S(0)) = d(\Pi_E u_n^*, S(0))$. Suppose $\Pi_E u_n^*$ is bounded. Since from (4) we have $\varphi^*(x_n^*, \Pi_E u_n^*) = \varphi^*(x_n^*, u_n^*) \to \beta$, the lower semicontinuity of φ^* implies that each cluster point v^* of $\Pi_E u_n^*$ satisfies $\varphi^*(0, v^*) \le \beta$, that is, $v^* \in S(0) \cap E$, and therefore $d(\Pi_E u_n^*, S(0)) \to 0$, which proves (6)(b) and (c).

Now, to prove the boundedness of $\Pi_E u_n^*$, we suppose the contrary. Passing to a subsequence we may assume that $\|\Pi_E u_n^*\| \to \infty$ and $\Pi_E u_n^* / \|\Pi_E u_n^*\| \to v^*$ for some nonzero $v^* \in E$. Again, $\varphi^*(x_n^*, \Pi_E u_n^*) = \varphi^*(x_n^*, u_n^*)$ converges to β so it is bounded above, say by M, and then

$$((0, v^*), 0) = \lim_{n \to \infty} \frac{((x_n^*, \Pi_E u_n^*), M)}{\|\Pi_E u_n^*\|} \in \text{epi } (\varphi^*)_\infty,$$

which means $(\varphi^*)_\infty(0, v^*) \le 0$. But since $h_\infty^*(v^*) = (\varphi^*)_\infty(0, v^*)$ this is in contradiction to the assumption $0 \in \text{ri}[\text{dom}(h)] = \text{ri}[\text{dom}(h^{**})]$ ([6, Cor. 13.3.4(b)]). □

There are obvious dual versions of the previous theorems, since the roles of h, φ and k, φ^* are completely symmetric. In the following discussion it will be useful to have them stated explicitly, for which we introduce the functions

$$k^u(x^*) = \inf_{u^* \in U^*} \varphi^*(x^*, u^*) - \langle u^*, u \rangle,$$

and we denote by k_F^u its extension $k^u \circ \Pi_F$.

THEOREM 3.3. *With the previous notation and assuming* $0 \in \text{ri}[\text{dom}(k)]$, *for each* $u \in U$ *it holds that*

$$-k^u(0) = h(u) = \min_{x \in X} \varphi(x, u),$$

the minimum being attained, with (nonvoid) optimal solution set given by

$$M(u) = \begin{cases} X & \textit{if } h(u) = +\infty, \\ \partial k^u(0) = \partial k^u_F(0) + F^\perp & \textit{otherwise.} \end{cases}$$

Moreover, every stationary sequence x_n for $\varphi(\cdot, u)$ satisfies

$$\begin{aligned} &\text{(i)} \qquad \varphi(x_n, u) \to h(u), \\ &\text{(ii)} \qquad d(x_n, M(u)) \to 0. \end{aligned}$$

The same is true merely if $d(0, \partial_{\varepsilon_n}\varphi(\cdot, u)(x_n)) \to 0$ with $\varepsilon_n \to 0$. In particular, $k^* \in \mathcal{R}$.

Suppose in addition that $0 \in \operatorname{ri}[\operatorname{dom}(h)]$. Then we have $g = h^* \in \mathcal{R}$, the optimal solution set of the dual $S(0)$ is given by (5), and for every sequence $u_n \to 0, u_n \in \operatorname{dom}(h) \subset E$ (in particular, if u_n^* is approximately stationary for the dual problem we can take $u_n \in \partial_{\varepsilon_n} g(u_n^*)$ with $u_n \to 0$) and each ε_n-minimum x_n of $\varphi(\cdot, u_n)$, where $\varepsilon_n \to 0$, we have

$$\begin{aligned} &\text{(a)} \qquad \varphi(x_n, u_n) \to \alpha, \\ &\text{(b)} \qquad d(x_n, M(0)) \to 0. \qquad \square \end{aligned}$$

In order to use properties (*a*) and (*b*) above for devising algorithms, we must be able to construct the sequence x_n, which solves the perturbed primal problems. This may be a difficult task, but the following observation may sometimes help.

Remark. Suppose that, by using a suitable algorithm on the dual problem (D), we get a stationary sequence u_n^* and a subgradient $u_n \in \partial h^*(u_n^*)$ with $u_n \to 0$ (recall the dual consists in minimizing $h^* = \varphi^*(0, \cdot)$). Then, finding a solution $x_n \in M(u_n)$ for the perturbed primal problem is equivalent to solving

$$(x_n, u_n) \in \partial\varphi^*(0, u_n^*),$$

which may be simpler, as when φ^* happens to be differentiable (examples where this holds will be given in the next section).

This remark, together with the previous theorem, gives an answer to the question raised in the introduction, namely, how to associate with each stationary sequence u_n^* for the dual, a primal minimizing sequence x_n that converges to the optimal solution set of (P).

4. Special classes of perturbations. For making the results of the previous section readily applicable, we shall discuss in this section the meaning of the hypothesis $0 \in \operatorname{ri}[\operatorname{dom}(h)]$ and $0 \in \operatorname{ri}[\operatorname{dom}(k)]$ for some natural perturbation schemes which appear when formulating convex mathematical programs. We shall also point out where the previous formulas and results simplify for these particular structures. We shall only sketch the proofs in this section since they are based on fairly standard arguments in convex analysis.

4.1. Vertical perturbations. Let us consider a constrained convex *primal* program of the type

$$(V) \quad \alpha = \inf\{f(x) : Ax = a, g_i(x) \le 0 \text{ for } i = 1, \ldots, p\},$$

where the functions f and g_i are closed proper convex functions defined on $\mathbb{R}^n$, A is an $m \times n$ matrix, and $a \in \mathbb{R}^m$. We shall denote $H := \mathrm{dom}(f) \cap \bigcap_{i=1}^p \mathrm{dom}(g_i)$ and we shall assume that this set is nonempty. We also denote $G(x) = (g_1(x), \ldots, g_p(x))$.

The *vertical perturbation* function associated with this problem is given by

$$\varphi(x,(v,w)) = \begin{cases} f(x) & \text{if } Ax + v = a, \quad G(x) + w \leq 0, \\ +\infty & \text{otherwise} \end{cases}$$

defined for $(v, w) \in U = \mathbb{R}^m \times \mathbb{R}^p$, which is a closed proper convex function under the previous hypothesis. The meaning of conditions $0 \in \mathrm{ri}[\mathrm{dom}(h)]$ and $0 \in \mathrm{ri}[\mathrm{dom}(k)]$ is made clear by the following proposition.

PROPOSITION 4.1. *For problem* (V) *we have*

(a) $0 \in \mathrm{ri}[\mathrm{dom}(h)] \Leftrightarrow (P_V)$ *there exists* $\bar{x} \in \mathrm{ri}\, H$ *such that* $A\bar{x} = a, G(\bar{x}) < 0$.

(b) $0 \in \mathrm{ri}[\mathrm{dom}(k)] \Leftrightarrow (D_V)$ $f_\infty(v) > 0$ *for all* $v \in L^\perp \setminus \{0\}$ *such that* $Av = 0, (g_i)_\infty(v) \leq 0$. *Here* $L = \{v : f_\infty(v) = f_\infty(-v) = 0, Av = 0, (g_i)_\infty(v) = (g_i)_\infty(-v) = 0, i = 1 \ldots p\}$.

Proof. (a) Defining $C = \{(x, w) : x \in H, G(x) + w \leq 0\}$ and $L(x, w) = (a - Ax, w)$ we have $\mathrm{dom}(h) = L(C)$ so that $\mathrm{ri}[\mathrm{dom}(h)] = L(\mathrm{ri}\, C)$. But $\mathrm{ri}\, C = \{(x, w) : x \in \mathrm{ri}\, H, G(x) + w < 0\}$, from which (a) follows.

(b) The primal functional is $F(x) = f(x) + \chi_B(x)$ where $B = \{x : Ax = a, g_i(x) \leq 0, i = 1 \ldots p\}$ is the primal feasible set. Then (b) follows by observing that $F_\infty(v) = f_\infty(v) + \chi_{B_\infty}(v)$ and $B_\infty = \{v : Av = 0, (g_i)_\infty(v) \leq 0, i = 1 \ldots p\}$. □

When the inequality constraints are linear, that is, $G(x) = Bx - b$ with B a $p \times n$ matrix and $b \in \mathbb{R}^p$, part (b) may be improved by showing the equivalence between $0 \in \mathrm{ri}[\mathrm{dom}(k)]$ and

$$(D'_V) \quad \exists\, v^* \in \mathbb{R}^m, w^* \in \mathbb{R}^p \quad \text{such that } w^* > 0, \quad A^t v^* + B^t w^* \in \mathrm{ri}[\mathrm{dom}(f^*)].$$

This condition is certainly satisfied when f is co-finite [6, p. 116] (since in such a case $\mathrm{dom}(f^*) = \mathbb{R}^n$), and more generally if $0 \in \mathrm{int}(\mathrm{dom}(f^*))$.

Concerning the remark made after Theorem 3.3 on the computation of a solution $x \in M(u)$ of the perturbed primal problem when we have at our disposal a u^* such that $u \in \partial h^*(u^*)$, we mention the following proposition, still in the case of linear inequalities.

PROPOSITION 4.2. *Assume* (P_V) *and* (D'_V) *are satisfied. If* $(v, w) \in \partial h^*(v^*, w^*)$ *then* $x \in M(v, w)$ *if and only if*

$$Ax + v = a, Bx + w \leq b \quad \textit{and} \quad x \in \partial f^*(-A^t v^* - B^t w^*).$$

Proof. From the remark following Theorem 3.3, one gets that $x \in M(v, w)$ if and only if

(i) $Ax + v = a, Bx + w \leq b$,

(ii) $f(x) + f^*(-A^t v^* - B^t w^*) = \langle v^*, v - a\rangle + \langle w^*, w - b\rangle$.

Using (i), Fenchel's inequality, and the fact that $w^* \geq 0$, we may rewrite (ii) as

(ii)′ $f(x) + f^*(-A^t v^* - B^t w^*) = \langle v^*, -Ax\rangle + \langle w^*, -Bx\rangle$, which is simply $x \in \partial f^*(-A^t v^* - B^t w^*)$. □

When f is essentially strictly convex we know [6, §26] that f^* is differentiable at every point where ∂f^* is nonempty. We obtain the following as a corollary.

COROLLARY 4.3. *Assume* (P_V) *and* (D'_V). *If* f *is essentially strictly convex and* $(v, w) \in \partial h^*(v^*, w^*)$, *then*

$$M(v, w) = \{\nabla f^*(-A^t v^* - B^t w^*)\}.$$

Moreover, when f is essentially smooth the solution set of the dual problem $S(0)$ is such that $S(0) \cap E$ is reduced to a singleton.

Proof. By Theorem 3.3 we know that $M(v, w)$ is nonempty, so the previous proposition forces $\partial f^*(-A^t v^* - B^t w^*)$ to be nonempty also and the characterization follows. Now, if f is essentially smooth then its conjugate f^*, and therefore h^*, are essentially strictly convex on E and the second assertion follows as well. □

As an application of this result and the remark following Theorem 3.3, suppose f is strictly convex and $(P_V), (D'_V)$ are satisfied. Then, associated with each stationary sequence (v_n^*, w_n^*) for the dual, we have the sequence

$$x_n = \nabla f^*(-A^t v_n^* - B^t w_n^*),$$

which is well defined and converges to the unique optimal solution of (V). Also $f(x_n) \to \alpha$, which is seen by choosing $(v_n, w_n) \in \partial h^*(v_n^*, w_n^*)$ tending to zero so that according to Theorem 3.3 we have $f(x_n) = \varphi(x_n, (v_n, w_n)) \to \alpha$. Moreover, since the dual functional belongs to $\mathcal{R}$ we also have that $d((v_n^*, w_n^*), S(0)) \to 0$.

An important special case of (V) concerns linearly constrained decomposable problems of the form

$$(L) \quad \alpha = \inf\left\{\sum_{i=1}^{k} f_i(x_i) : A_1 x_1 + \cdots + A_k x_k = b\right\},$$

where $f_i : \mathbb{R}^{n_i} \to \overline{\mathbb{R}}$ are closed proper convex functions, A_i are $m \times n_i$ matrices and $a \in \mathbb{R}^m$. For this problem we have the following corollary.

COROLLARY 4.4. *For problem (L) we have*

(a) $0 \in \text{ri}[\text{dom}(h)] \Leftrightarrow (P_L)$ *there exists $\bar{x}_i \in \text{ri}[\text{dom}(f_i)]$ such that $A_1\bar{x}_1 + \cdots + A_k\bar{x}_k = a$.*

(b) $0 \in \text{ri}[\text{dom}(k)] \Leftrightarrow (D_L)$ *there exists $\bar{u}^* \in \mathbb{R}^m$ with $-A_i^t u^* \in \text{ri}[\text{dom}(f_i^*)]$, for all $i = 1, \ldots, k$.*

(c) *Assume (P_L) and (D_L). If $u \in \partial h^*(u^*)$ then $(x_1, \ldots, x_k) \in M(u)$ if and only if*

$$\sum_{i=1}^{k} A_i x_i + u = a \quad \textit{and} \quad x_i \in \partial f_i^*(-A_i^t u^*) \quad \textit{for } i = 1, \ldots, k.$$

If moreover all the f_i's are essentially strictly convex then

$$M(u) = \{(\nabla f_1^*(-A_1^t u^*), \ldots, \nabla f_k^*(-A_k^t u^*))\}.$$

Also, when the f_i's are essentially smooth the solution set of the dual problem $S(0)$ is such that $S(0) \cap E$ is reduced to a singleton.

Proof. (P_L) corresponds obviously to (P_V) of Proposition 4.1. Furthermore, since $f^*(x^*) = \sum_{i=1}^k f_i^*(x_i^*)$, the condition $A^t v^* \in \text{ri}(\text{dom } f^*)$ is equivalent to $A_i^t v^* \in \text{ri}(\text{dom} f_i^*)$ and (D_L) is equivalent to (D'_V) without inequality constraints. The rest of the proof is an immediate consequence of Proposition 4.2 and Corollary 4.3. □

An interesting feature of the dual problem in this case is that, when the f_i's are essentially strictly convex and co-finite, this dual is an unconstrained differentiable program for which a variety of algorithms can be applied to generate a stationary sequence.

4.2. Fenchel duality. This perturbation scheme concerns a *primal* problem of the type

$$(F) \quad \alpha = \inf_{x \in X} f(x) + g(Ax)$$

with f and g closed proper convex functions defined on $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, and A is an $m \times n$ matrix. The perturbation function is given by

$$\varphi(x, u) = f(x) + g(Ax + u),$$

whose conjugate is

$$\varphi^*(x^*, u^*) = f^*(-A^t u^* + x^*) + g^*(u^*),$$

and we may quote the following well-known result [6, p. 330].

PROPOSITION 4.5. *For problem* (F) *we have*

(a) $0 \in \mathrm{ri}[\mathrm{dom}(h)] \Leftrightarrow (P_F)$ *there exists* $x \in \mathrm{ri}[\mathrm{dom}(f)]$ *such that* $Ax \in \mathrm{ri}[\mathrm{dom}(g)]$.

(b) $0 \in \mathrm{ri}[\mathrm{dom}(k)] \Leftrightarrow (D_F)$ *there exists* $u^* \in \mathrm{ri}[\mathrm{dom}(g^*)]$ *such that* $-A^t u^* \in \mathrm{ri}[\mathrm{dom}(f^*)]$. □

Let us simply mention that (P_F) obviously holds when g is everywhere finite, and similarly, (D_F) is true when f is co-finite. Concerning the analogs of Proposition 4.2 and its corollary, we obtain the following proposition.

PROPOSITION 4.6. *Assume* (P_F) *and* (D_F). *If* $u \in \partial h^*(u^*)$ *then*

$$x \in M(u) \Leftrightarrow x \in \partial f^*(-A^t u^*) \quad \text{and} \quad Ax + u \in \partial g^*(u^*).$$

Moreover, if f *is essentially strictly convex, then* $M(u) = \{\nabla f^*(-A^t u^*)\}$. □

Under assumptions (P_F) and (D_F) and when f is essentially strictly convex, from each stationary sequence u_n^* for the dual we get the sequence

$$x_n = \nabla f^*(-A^t u_n^*),$$

which will converge to the optimal solution of the primal problem. Moreover, if $u_n \in \partial h^*(u_n^*)$ converges to zero then $f(x_n) + g(Ax_n + u_n) \to \alpha$, and the distance from u_n^* to the optimal set of the dual problem $S(0)$ tends to zero.

To conclude this section, let us show how the results in [7], [3], and [5] can be obtained from the ones we have presented.

EXAMPLE 1. In [7], Tseng and Bertsekas are concerned with network flow problems of the type

$$\inf \left\{ \sum_{j=1}^{n} f_j(x_j) : Ex = 0 \right\},$$

where E is an $m \times n$ network incidence matrix, and under the assumptions

(a1) $\mathrm{Ker}(E) \cap \prod_{j=1}^{n} \mathrm{ri}[\mathrm{dom}(f_j)] \neq \phi$,

(b1) each $f_j : \mathbb{R} \to \overline{\mathbb{R}}$ is closed, proper, co-finite, and essentially strictly convex.

This problem falls into the class of linearly constrained decomposable problems and, as discussed in [7], the dual turns out to be an unconstrained differentiable program. Trivially (a1) and (b1) give conditions $0 \in \mathrm{ri}[\mathrm{dom}(h)]$ and $0 \in \mathrm{ri}[\mathrm{dom}(k)]$, respectively, as seen from Corollary 4.4.

EXAMPLE 2. In [3], Censor and Lent are concerned with the minimization of "$\log x$" entropy under linear constraints, a problem which arises in image restoration as an alternative to the classical "$x \log x$" entropy. Their problem is

$$\inf \left\{ -\sum_{i=1}^{n} \log x_i : Ax = b, x > 0 \right\},$$

where A is an $m \times n$ matrix and $b \in \mathbb{R}^m$. Their assumptions are

(a2) there exists $x \in \mathbb{R}^n_{++}$ such that $Ax = b$,

(b2) $\operatorname{Ker}(A) \cap \mathbb{R}^n_+ = \{0\}$.

This problem also falls into the class of linearly constrained decomposable problems with $f_i(x) = -\log(x)$ if $x > 0$ and $f_i(x) = +\infty$ otherwise. Conditions (a2) and (b2) give $0 \in \operatorname{ri}[\operatorname{dom}(h)]$ and $0 \in \operatorname{ri}[\operatorname{dom}(k)]$, respectively, as seen from Corollary 4.4 and Gordan's transposition theorem. Moreover, in this case the functions f_i are essentially smooth, so the solution set $S(0)$ of the dual is such that $S(0) \cap E$ is a singleton.

EXAMPLE 3. In [5], Han and Lou consider an abstract problem of the type

$$\inf \{q(x) : x \in C_1 \cap \cdots \cap C_m\},$$

where the C_i are closed convex subsets of $\mathbb{R}^n$ and q is a *finite* convex function on $\mathbb{R}^n$. They assume

(a3) $\bigcap_{i=1}^m \operatorname{ri}(C_i) \neq \phi$,

(b3) q is strongly convex,

(c3) q is differentiable everywhere.

Among the various ways to dualize, we have chosen the Fenchel scheme with $f(x) = q(x)$, $g(y_1, \ldots, y_m) = \sum_1^m \chi_{C_i}(y_i)$ and $Ax = (x, \ldots, x) \in (\mathbb{R}^n)^m$.

Again, conditions (a3) and (b3) give $0 \in \operatorname{ri}[\operatorname{dom}(h)]$ and $0 \in \operatorname{ri}[\operatorname{dom}(k)]$, respectively, as seen from Proposition 4.5.

5. Algorithmic remarks.

5.1. If we consider the classical mathematical programming problem (V) of §4.1, and if we apply to it an exterior penalty method, then Theorem 2.6 gives us new convergence results. To be more precise, let us consider the classical quadratic penalty function

$$f_n(x) = f(x) + k_n \left[\|Ax - b\|^2 + \sum_{i=1}^{p} (g_i^+(x))^2 \right], \qquad k_n \to \infty.$$

Then, under hypothesis (D_V) (see Proposition 4.1) we have from Theorem 2.6,

(1) for n large enough the minimum of f_n is attained at some point x_n (which is not evident a priori);

(2) the sequence x_n approaches the optimal solution set of (V).

This result is known when f is inf-compact but not under the weaker assumption (D_V). Let us also mention here that penalty methods are receiving a renewed attention since it has been shown how to overcome the difficulties raised by the increasing ill-conditioning associated with the divergence of the penalty parameter.

5.2. In the examples from [7], [3], and [5] presented at the end of the previous section, the numerical method proposed by the authors in all three cases is the Gauss–Seidel method applied to the dual problem, with a choice of stepsize given by exact minimization in [3] and [5], and a specific choice in [7]. These methods are highly decomposable in the sense that both the primal and the dual are decomposable.

In [3] the only convergence results presented concern the primal sequences, and nothing is said about the dual. Now, since the dual sequence is stationary for the dual and the dual functional belongs to $\mathcal{R}$, Theorem 3.2 shows that this dual stationary sequence approaches the dual solution set. Moreover, as mentioned in Example 2, the set $S(0) \cap E$ is reduced to a singleton so that the projection of the dual stationary sequence onto E *converges* towards this dual optimal solution, while Theorem 3.3 gives the convergence of the primal sequence.

In [5] no convergence result is given. Now, as remarked in [1, Thm. 4.3] the dual sequence generated by Han and Lou's algorithm is stationary for the dual, and therefore, the results in §3 allow us to conclude that the associated primal sequence converges towards a primal solution, and that the dual sequence tends towards the dual optimal set.

In [7] it is proved that the dual sequence generated by their algorithm is minimizing and that the associated primal sequence converges towards a solution of the primal problem. We also obtain this result from §3, but we can add that the dual sequence approaches the dual solution set. Moreover, under the additional assumption that the functions f_j's are essentially smooth we can associate with this dual sequence its projection onto E, which will converge to the singleton $S(0) \cap E$. Let us point out nevertheless that, by a suitable modification of their original algorithm which takes into account the specific structure of the problem, Tseng and Bertsekas have recently proved convergence of the dual sequence without this extra assumption on the f_j's.

5.3. Finally, let us point out that in [1] a variant of the Gauss–Seidel method has been proposed, which is shown to converge for a subclass of $\mathcal{F}$ containing the cases in [7], [3], and [5]. It is shown that this algorithm generates a stationary sequence, so that when applied to the dual problems in [7], [3], and [5] it gives a dual sequence that approaches the dual solution set and a primal sequence converging towards a primal optimal solution. Moreover, in [3] and [5] we can associate with the dual sequence another one which converges towards a dual optimal solution. More generally, every method generating a stationary sequence will enjoy the same properties.

6. More on asymptotically well behaved convex functions.

We shall now improve some theorems in [2] concerning the characterization of the class $\mathcal{F}$ of asymptotically well behaved convex functions, that is, those functions $f \in \Gamma(X)$ satisfying

$$d(0, \partial f(x_k)) \to 0 \Longrightarrow \lim_{k\to\infty} f(x_k) = m := \inf_{x\in X} f(x).$$

To this end we shall consider the following quantities, which are defined for each $\lambda > m$:

$$r(\lambda) = \inf_{f(x)=\lambda} \inf_{x^*\in\partial f(x)} \|x^*\|,$$
$$k(\lambda) = \inf_{f(x)=\lambda} \inf_{x^*\in\partial f(x)} f'(x; x^*/\|x^*\|),$$
$$l(\lambda) = \inf_{f(x)>\lambda} \frac{f(x)-\lambda}{d(x, S_\lambda(f))},$$

where $f'(x;\cdot)$ denotes the directional derivative of f and the level set $S_\lambda(f)$ is defined as usual by $\{x \in X : f(x) \le \lambda\}$. It will also be useful to state and prove the following lemma.

LEMMA 6.1. *Let $\lambda > m$ and $x \notin S_\lambda(f)$ such that $f(x) < +\infty$. If we denote by y the projection of x onto $S_\lambda(f)$ then we have $f(y) = \lambda$ and for some $\alpha > 0$,*

$$\alpha(x-y) \in \partial f(y).$$

Proof. The projection y is the unique solution of the minimization problem

$$\min_{f(z)\le\lambda} \frac{1}{2}\|z-x\|^2,$$

so that since Slater's condition is satisfied ($\lambda > m$), the optimality condition gives the existence of a multiplier $\mu \geq 0$ such that

$$0 \in (y-x) + \mu\partial f(y).$$

Since y belongs to $S_\lambda(f)$ while x does not, we conclude $\mu > 0$ and we may just take $\alpha = 1/\mu$.

We must show also that $f(y) = \lambda$. From feasibility we have $f(y) \leq \lambda$. On the other hand, for every $t \in]0,1[$ the point $x + t(y-x)$ does not belong to $S_\lambda(f)$ and therefore

$$\lambda < f(x+t(y-x)) \leq (1-t)f(x) + tf(y),$$

which, after letting $t \to 1$, gives us $f(y) \geq \lambda$. □

With this lemma we may now present some relations between the quantities $r(\lambda)$, $k(\lambda)$, and $l(\lambda)$ introduced above.

PROPOSITION 6.2. *For each $\lambda > m$ we have $l(\lambda) = k(\lambda)$.*

Proof. We must prove the two inequalities $l \geq k$ and $l \leq k$. The first amounts to saying that for every x such that $f(x) > \lambda$ we have

$$k(\lambda) \leq \frac{f(x)-\lambda}{d(x, S_\lambda(f))},$$

which follows from the previous lemma. In fact, it suffices to consider the case $f(x) < +\infty$ and then, taking the projection y of x onto the set $S_\lambda(f)$, we have for some $\alpha > 0$,

$$k(\lambda) \leq f'(y;\alpha(x-y)/\|\alpha(x-y)\|) = \frac{f'(y;x-y)}{\|x-y\|} \leq \frac{f(x)-f(y)}{d(x,S_\lambda(f))},$$

from which the result follows since $f(y) = \lambda$.

For the converse inequality we must show that given x with $f(x) = \lambda$ and given $x^* \in \partial f(x)$ we have

$$l(\lambda) \leq f'(x; x^*/\|x^*\|).$$

In fact, for every $z \in S_\lambda(f)$ we have

$$0 \geq f(z) - f(x) \geq \langle x^*, z-x\rangle,$$

so that x^* is on the normal cone to $S_\lambda(f)$ at x and then, for $t > 0$ we have $d(x + tx^*, S_\lambda(f)) = t\|x^*\| > 0$, the last inequality since x is not a minimum ($f(x) = \lambda > m$). We deduce

$$\frac{f(x+tx^*)-f(x)}{t\|x^*\|} = \frac{f(x+tx^*)-\lambda}{d(x+tx^*, S_\lambda(f))} \geq l(\lambda),$$

so that letting $t \downarrow 0$ we get the desired conclusion. □

PROPOSITION 6.3. *If $\lambda' > \lambda > m$, then $k(\lambda') \geq r(\lambda') \geq k(\lambda) \geq r(\lambda)$.*

Proof. Since for $x^* \in \partial f(x)$ one has $\langle x^*, d\rangle \leq f'(x; d)$ for all $d \in X$, it follows easily that $k \geq r$. Thus, it suffices to show $r(\lambda') \geq k(\lambda)$. Let us take x' with $f(x') = \lambda'$ and let x be its projection onto $S_\lambda(f)$. Then we may use the lemma and write for some $\alpha > 0$

$$k(\lambda) \leq f'(x; \alpha(x'-x)/\|\alpha(x'-x)\|) = \frac{f'(x; x'-x)}{\|x'-x\|} \leq \frac{f(x')-f(x)}{\|x'-x\|}.$$

Hence, for every $x^* \in \partial f(x')$ we get

$$k(\lambda) \leq \frac{\langle x^*, x'-x\rangle}{\|x'-x\|} \leq \|x^*\|,$$

and the desired inequality follows. □

COROLLARY 6.4. *For $\lambda > m$ we have the following alternative characterizations*

$$r(\lambda) = \inf_{f(x)\geq\lambda} \inf_{x^*\in\partial f(x)} \|x^*\|,$$
$$k(\lambda) = \inf_{f(x)\geq\lambda} \inf_{x^*\in\partial f(x)} f'(x; x^*/\|x^*\|).$$

Proof. This is a consequence of the monotonicity of r and k. □

We may now give the announced characterizations of the asymptotically well behaved convex functions.

THEOREM 6.5. *The following statements are equivalent:*

(1) *$f \in \mathcal{F}$.*

(2) *All stationary sequences x_k with $f(x_k)$ bounded satisfy $f(x_k) \to \inf f(x)$.*

(3) *$r(\lambda) > 0$ for all $\lambda > m$.*

(4) *$k(\lambda) > 0$ for all $\lambda > m$.*

(5) *$l(\lambda) > 0$ for all $\lambda > m$.*

Proof. The implication (1) ⇒ (2) as well as the equivalence between (3), (4), and (5) are obvious from the definition of $\mathcal{F}$ and the previous results, respectively.

To prove (2) ⇒ (3) we observe that otherwise there exists $\lambda > m$ with $r(\lambda)$=0 so we can find sequences x_k and $x_k^* \in \partial f(x_k)$ with $x_k^* \to 0$ and $f(x_k) = \lambda > m$, contradicting (2).

The implication (3) ⇒ (1) follows similarly. If (1) did not hold we could find a stationary sequence that is not minimizing. Extracting a subsequence we could find $\lambda > m$ and sequences x_k and $x_k^* \in \partial f(x_k)$ such that $f(x_k) \geq \lambda$ and $x_k^* \to 0$. The alternative characterization of r in the previous corollary yields $r(\lambda) = 0$, contradicting (3). □

Remark. The results presented above were shown in [2] under the supplementary assumption

$$S_\lambda(f) \subset \mathrm{ri}[\mathrm{dom}(f)] \quad \forall\, \lambda > m.$$

Also, the proof of the monotonicity of r and k has been considerably simplified. Furthermore, the proofs presented above have the additional advantage of passing over, with minor modifications, to the reflexive Banach space setting.

REFERENCES

[1] A. AUSLENDER, *Asymptotic properties of the Fenchel's dual functional and their applications to decomposition problems*, J. Optim. Theory Appl., 73 (1992), pp. 427–450.

[2] A. AUSLENDER AND J.-P. CROUZEIX, *Well behaved asymptotical convex functions*, Analyse non-linéaire, Gauthier-Villars, Paris, 1989, pp. 101–122.

[3] Y. CENSOR AND A. LENT, *Optimization of* "log(x)" *entropy over linear equality constraints*, SIAM J. Control Optim., 25 (1987), pp. 921–933.

[4] M. GOBERNA AND A. LOPEZ, *Optimal value function in semi-infinite programming*, J. Optim. Theory Appl., 59 (1988), pp. 261–278.

[5] S. P. HAN AND G. LOU, *A parallel algorithm for a class of convex programs*, SIAM J. Control Optim., 26 (1988), pp. 345–355.

[6] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1968.

[7] P. TSENG AND D. BERTSEKAS, *Relaxation methods for problems with strictly convex costs and linear constraints*, Math. Programming, 38 (1987), pp. 303–321.

SIAM J. OPTIMIZATION
Vol. 3, No. 4, pp. 688–695, November 1993

002

REDUCING MATCHING TO POLYNOMIAL SIZE LINEAR PROGRAMMING*

FRANCISCO BARAHONA†

Abstract. The question of whether the maximum weight matching problem can be reduced to a linear program of polynomial size is studied. A partial answer to it is given; i.e., it is shown that the Chinese postman problem (and optimum matching) reduces to a sequence of $O(m^2 \log n)$ minimum mean cycle problems. It is shown that this last problem can be formulated as a linear program of polynomial size. This gives a polynomial algorithm for matching based on any polynomial method for linear programming. A combinatorial algorithm for finding minimum mean cycles in undirected graphs is also given.

Key words. matching, polynomial size linear programming

AMS subject classifications. 05C70, 05C85, 90C27

1. Introduction.

The convex hull of the incidence vectors of matchings in a graph has been characterized by Edmonds [8], with a system that contains exponentially many inequalities. Subsequently, polyhedra related to several other combinatorial problems have been characterized. In all these cases the linear systems also involve exponentially many inequalities.

An important question in the theory of integer programming is whether these problems can be formulated as "small" linear programs; i.e., linear programs with a polynomial number of variables and a polynomial number of inequalities. Such a formulation is called *compact.* A compact system for optimum arborescences has been presented in Wong [26] and in Maculan [19]. Ball, Liu, and Pulleyblank [1] gave a compact system for two terminal Steiner trees. In Barahona and Mahjoub [5], [6] we presented compact systems for the following problems in series-parallel graphs: stable sets, acyclic induced subgraphs, and bipartite induced subgraphs. In [4] we gave compact systems for the max cut problem in graphs with no K_5 minor and optimum perfect matching in planar graphs. If we have a compact system for a problem, we can solve it in polynomial time by means of any polynomial algorithm for linear programming.

Given a graph $G = (V, E)$, we denote by n the number of nodes and by m the number of edges. An outstanding open question is whether the optimum matching problem in general graphs can be formulated as a linear program whose size is bounded by a polynomial in n. Yannakakis [27] proved that it is not possible by means of a *symmetric* system. In this paper we show that the Chinese postman problem (and optimum matching) reduces to a sequence of $O(m^2 \log n)$ minimum mean cycle problems. We show that this latter problem can be formulated as a linear program of polynomial size. This gives a polynomial algorithm for matching, whose only nontrivial operation is solving a linear program of polynomial size. We also give a combinatorial algorithm for finding minimum mean cycles in undirected graphs.

2. Chinese postman and the minimum mean cycle problem.

Edmonds and Johnson [9] gave the first polynomial algorithm for the Chinese postman problem.

*Received by the editors May 25, 1990; accepted for publication (in revised form) June 17, 1992.

†IBM T. J. Watson Research Center, Yorktown Heights, New York 10598.

Given a graph $G = (V, E)$, $T \subseteq V$, with $|T|$ even, and a set of integer weights $w(e) \geq 0$, for $e \in E$ the problem can be formulated as

$$\begin{aligned} &\text{minimize } \sum w(e)x(e) \\ &\text{subject to} \\ &\sum_{e \in \delta(v)} x(e) \equiv \begin{cases} 1 \,(\text{mod } 2), & \text{if } v \in T, \\ 0 \,(\text{mod } 2), & \text{if } v \notin T, \end{cases} \\ &x(e) \in \{0, 1\}, \quad \text{for } e \in E. \end{aligned} \tag{2.1}$$

We use $\delta(S)$ to denote the set of edges with exactly one endnode in S, for $S \subseteq V$.

They proved that this problem is equivalent to the linear program

$$\begin{aligned} &\text{minimize } \sum w(e)x(e) \\ &\text{subject to} \\ &\sum_{e \in \delta(S)} x(e) \geq 1, \quad \text{for every set } S \subseteq V, \text{ with } |S \cap T| \text{ odd}, \\ &x \geq 0. \end{aligned} \tag{2.2}$$

Their proof is based on a combinatorial algorithm whose complexity is $O(n^3)$ for complete graphs, and $O(nm \log n)$ for sparse graphs.

For planar graphs this algorithm can be combined with the separator theorem of Lipton and Tarjan [18] to solve the problem in $O(n^{3/2} \log n)$ time; see [3].

After Khachiyan [16] proved that linear programming is polynomial via the ellipsoid method, Padberg and Rao [23] gave a combinatorial algorithm to solve the so-called *separation* problem:

Given a vector $\bar{x}$, prove that it satisfies the constraints (2.2) *or find a violated inequality.*

The algorithm of Padberg and Rao, combined with the ellipsoid method, also gives a polynomial algorithm for solving (2.2); cf. Grötschel, Lovász, and Schrijver [13], Karp and Papadimitriou [15], and Padberg and Rao [23]. This result is highly dependent on the ellipsoid method; i.e., replacing the ellipsoid algorithm by any other polynomially bounded algorithm for linear programming or the simplex method would not necessarily lead to a polynomial number of iterations.

In what follows we shall prove that the Chinese postman problem reduces to a polynomially bounded sequence of minimum mean cycle problems.

If $\bar{x}$ is a 0-1 vector that satisfies the equations (2.1), the set $F = \{e : \bar{x}(e) = 1\}$ is called a T-join. If F and F' are T-joins then their symmetric difference $F \triangle F'$ is a set of edge-disjoint cycles. If C is a cycle then $F \triangle C$ is also a T-join.

Given a T-join $\bar{F}$, let us define w' by

$$w'(e) = \begin{cases} -w(e), & \text{if } e \in \bar{F}, \\ w(e), & \text{if } e \notin \bar{F}. \end{cases} \tag{2.3}$$

We use $w(S)$ to denote $\sum\{w(e) : e \in S\}$.

If C is a cycle with $w'(C) < 0$, we have

$$w(\bar{F} \triangle C) = w(\bar{F}) + w'(C) < w(\bar{F}).$$

As Mei-Ko [20] suggested, a negative cycle with respect to the weights w' leads to a better T-join. Finding a most negative cycle is an NP-hard problem. Instead, we propose looking for a cycle of minimum mean weight; i.e., a cycle $\widetilde{C}$ such that

$$\frac{w'(\widetilde{C})}{|\widetilde{C}|} \leq \frac{w'(C)}{|C|} \quad \text{for every cycle } C.$$

The following "negative cycle" algorithm is very similar to the algorithm of Goldberg and Tarjan [11] for minimum cost network flows.

```
Step 0. Choose any $T$-join $\bar{F}$.
Step 1. Find a minimum mean cycle $\widetilde{C}$.
Step 2. If $w'(\widetilde{C}) \geq 0$ stop.
        If $w'(\widetilde{C}) < 0$, set $\bar{F} \leftarrow \bar{F} \triangle \widetilde{C}$ and go to Step 1.
```

The remainder of this section is devoted to proving that the number of iterations is polynomially bounded.

Let $\bar{F}$ be a T-join and $\widehat{F}$ be an optimum T-join. We have that

$$w(\widehat{F}) = w(\bar{F}) + w'(C_1) + \cdots + w'(C_k),$$

where $C_1, \ldots, C_k$ is the set of cycles that forms $\bar{F} \triangle \widehat{F}$, and $w'(C_i) < 0$, for $1 \leq i \leq k$.

Let $\widetilde{C}$ be a minimum mean cycle. Since

$$\frac{w'(\widetilde{C})}{|\widetilde{C}|} \leq \frac{w'(C_i)}{|C_i|} \quad \text{for } 1 \leq i \leq k,$$

we have

$$\frac{w'(\widetilde{C})}{|\widetilde{C}|} \leq \frac{w'(C_1) + \cdots + w'(C_k)}{|C_1| + \cdots + |C_k|};$$

therefore,

$$|w'(\widetilde{C})| \geq |w(\widehat{F}) - w(\bar{F})|/m.$$

Letting

$$F' = \bar{F} \triangle \widetilde{C}, \tag{2.4}$$

we have

$$|w(\widehat{F}) - w(F')| \leq (1 - 1/m)|w(\widehat{F}) - w(\bar{F})|. \tag{2.5}$$

We could not see how to tighten inequality (2.5), even by replacing $\widetilde{C}$ by a most negative cycle in (2.4).

Since the weights are integer, the number of iterations is bounded by a number k such that $(1 - 1/m)^k |w(\widehat{F}) - w(\bar{F})| < 1$. Since $1/e > (1 - 1/m)^m$, the number k is $O(m \log \omega)$, where ω is a bound for the value of the objective function. In what follows we shall prove that the term $\log \omega$ can be replaced by a polynomial in n. A similar argument appears in Orlin [22].

LEMMA 2.6. *There is a weight function $\widetilde{w}$, with integer coefficients, such that the algorithm produces the same sequence of intermediate solutions as with w, and $|\widetilde{w}(e)| \leq (mn)^m$ for all e.*

Proof. The weights w' and the weights $\widetilde{w}'$ are defined relative to a current T-join by formula (2.3). In order for the algorithm to produce the same intermediate solutions, the new weights should satisfy the following inequalities:

$$\begin{array}{llll} \text{if} & w'(C) \geq 0 & \text{then} & \widetilde{w}'(C) \geq 0, \\ \text{if} & w'(C) \leq -1 & \text{then} & \widetilde{w}'(C) \leq -1, \text{ for every cycle } C, \\ \text{if} & \dfrac{w'(C)}{|C|} \leq \dfrac{w'(C')}{|C'|} & \text{then} & \dfrac{\widetilde{w}'(C)}{|C|} \leq \dfrac{\widetilde{w}'(C')}{|C'|}, \end{array}$$

for every pair of cycles C and C'.

We require these inequalities for every T-join.

If we consider the weights $\widetilde{w}$ as variables, then this is a system of linear inequalities whose coefficients are bounded by n. This polyhedron is nonempty because the original weights satisfy these inequalities. Thus there is a rational solution $(p_1/q, \ldots, p_m/q)$, with $|p_i|$ integer and bounded by $m!n^m$, for all i. □

Let us remark that the weights $\widetilde{w}$ need not be computed. We can state the following.

THEOREM 2.7. *The number of iterations of this algorithm is bounded by* $O(m^2 \log n)$.

3. The minimum mean cycle problem as a compact linear program. This section is devoted to showing that the minimum mean cycle problem and the problem of finding a negative cycle in an undirected graph can be formulated as a linear program of polynomial size. This is based on the following result of Seymour [25].

THEOREM 3.1. *The cone generated by the incidence vectors of the cycles of a graph is defined by the system*

$$\begin{array}{l} x(e) - x(C\backslash e) \leq 0, \text{ for each cut } C, \text{ for every edge } e \in C, \\ x \geq 0. \end{array}$$

Consider the linear program

$$\begin{array}{ll} & \text{minimize } \sum w(e)x(e) \\ & \text{subject to} \\ (3.2) & x(e) - x(C\backslash e) \leq 0 \quad \text{for each cut } C, \text{ for every edge } e \in C, \\ & \sum x(e) = 1, \\ & x \geq 0. \end{array}$$

An optimal basic solution of (3.2) gives a cycle of minimum mean weight. It also gives a negative cycle, if there is any.

We use x^C to denote the incidence vector of a cycle C. Finding a characterization of the convex hull of incidence vectors of simple cycles seems to be a difficult problem. The system (3.2) defines the convex hull of all incidence vectors of simple cycles divided by their cardinality; i.e.,

$$\text{Conv}\left\{\frac{x^C}{|C|} : \text{ for every simple cycle } C\right\}.$$

Now we have to give a compact formulation of (3.2). Consider the edge $\bar{e} = \{u, v\}$, and the system of inequalities

$$\begin{array}{ll} (3.3) & x(\bar{e}) - x(C\backslash\bar{e}) \leq 0 \quad \text{for each cut } C \text{ such that } \bar{e} \in C, \\ & x \geq 0. \end{array}$$

It follows from the max flow min cut theorem of Ford and Fulkerson [10] that x satisfies (3.3) if and only if there is a vector y such that (x, y) satisfies

$$\sum_j y_{ij} - \sum_j y_{ji} = \begin{cases} 0, & \text{if } \ i \neq u, i \neq v, \\ x(\bar{e}), & \text{if } \ i = u, \\ -x(\bar{e}), & \text{if } \ i = v, \end{cases}$$
$$0 \leq y_{ij} \leq x(e), \quad \text{if } e = \{i, j\}, \quad e \neq \bar{e}.$$

Therefore, problem (3.2) is equivalent to

$$\begin{aligned} & \text{minimize } \sum w(e)x(e) \\ & \text{subject to} \\ & \sum x(e) = 1, \\ & \sum_j y^{\bar{e}}_{ij} - \sum_j y^{\bar{e}}_{ji} = \begin{cases} 0, & \text{if } \ i \neq u, i \neq v, \\ x(\bar{e}), & \text{if } \ i = u, \\ -x(\bar{e}), & \text{if } \ i = v, \end{cases} \\ & \text{for all } i \in V, \\ & 0 \leq y^{\bar{e}}_{ij} \leq x(e), \quad \text{if } e = \{i, j\}, \quad e \neq \bar{e}, \\ & \text{for every edge } \bar{e} \in E. \end{aligned} \tag{3.4}$$

This is a linear program with $O(m^2)$ variables, $O(nm)$ equations, and $O(m^2)$ inequalities.

It is clear that from the point of view of worst case analysis, Edmonds's algorithm is better than the algorithm of §2. The practical efficiency of our algorithm depends on a good heuristic to find an initial T-join, and on how fast one can solve problem (3.2) or (3.4). For instance, for solving network flow problems there is no need to write down the flow conservation equations; they can be treated implicitly. This suggests that when solving problem (3.4), one should treat those constraints implicitly. The details of such an implementation are beyond the scope of this paper. We should mention that Grötschel and Holland [12] have implemented a cutting plane algorithm that is not even polynomial but competes well with combinatorial methods.

4. A combinatorial algorithm for minimum mean cycles.

In §2 we showed that the Chinese postman problem (and optimum matching) reduces to a sequence of minimum mean cycle problems. In this section the reverse direction is shown; i.e., that the minimum mean cycle problem reduces to a sequence of Chinese postman problems. We should point out that Megiddo [21] gave a general procedure for ratio problems that would yield an $O(n^6)$ algorithm in our case.

Given a set of weights $w(e)$ (unrestricted in sign), for $e \in E$, we are looking for a cycle $\widetilde{C}$ such that

$$\frac{w(\widetilde{C})}{|\widetilde{C}|} \leq \frac{w(C)}{|C|},$$

for every cycle C.

There is a well-known method to solve ratio problems; see [21]. Given a cycle $\widetilde{C}$, one should find a cycle $\bar{C}$ that minimizes $w(C) - \lambda|C|$, where $\lambda = w(\widetilde{C})/|\widetilde{C}|$. If the minimum is negative then $\bar{C}$ is better than $\widetilde{C}$. There are two difficulties: first, finding that minimum is an NP-hard problem; second, we want to find a polynomial bound for the number of iterations. We propose the following algorithm.

Step 0. Choose any cycle $\widetilde{C}$.

Step 1. Define $\bar{w}(e) = w(e) - \lambda$, for $e \in E$, where $\lambda = w(\widetilde{C})/|\widetilde{C}|$.

Step 2. Solve

$$\begin{aligned} &\text{minimize } \sum \bar{w}(e)x(e) \\ &\text{subject to} \\ (4.1) \qquad &\sum_{e\in\delta(v)} x(e) \equiv 0 \,(\text{Mod } 2), \quad \text{for } v \in V, \\ &x(e) \in \{0,1\}, \quad \text{for } e \in E. \end{aligned}$$

Step 3. If the value of the minimum is zero, stop.
Otherwise, let $\hat{x}$ be an optimal solution of (4.1). Set $\widetilde{C} = \{e : \hat{x}(e) = 1\}$ and go to Step 1.

Let $\widetilde{C}$ be the set given by this algorithm. This consists of a union of edge-disjoint cycles. Any of those cycles is an optimal solution.

Two lemmas are needed to prove that this is a polynomial algorithm.

LEMMA 4.2. *Problem* (4.1) *reduces to a Chinese postman problem.*

Proof. Define $E_1 = \{e : \bar{w}(e) < 0\}$, $E_2 = \{e : \bar{w}(e) \geq 0\}$, and

$$d(e) = \begin{cases} -\bar{w}(e), & \text{if } e \in E_1, \\ w(e), & \text{if } e \in E_2, \end{cases} \qquad x'(e) = \begin{cases} x(e), & \text{if } e \in E_1, \\ 1 - x(e), & \text{if } e \in E_2. \end{cases}$$

Notice that $d \geq 0$. Problem (4.1) is equivalent to

$$\begin{aligned} &\text{minimize } dx' \\ &\text{subject to} \\ (4.3) \qquad &\sum_{e\in\delta(v)} x'(e) \equiv \begin{cases} 1 \,(\text{mod } 2), & \text{if } |\delta(v) \cap E_2| \text{ is odd,} \\ 0 \,(\text{mod } 2), & \text{if } |\delta(v) \cap E_2| \text{ is even,} \end{cases} \\ &\text{for } v \in V, \\ &x'(e) \in \{0,1\}, \quad \text{for } e \in E. \quad \square \end{aligned}$$

The following result, which is an adaptation of a lemma of Cunningham [7], gives a bound for the number of iterations.

LEMMA 4.4. *Let $\widetilde{C}$ be the set obtained in Step* 3 *for some value of λ, and $\widetilde{\lambda} = w(\widetilde{C})/|\widetilde{C}|$. If C' is the set obtained in the next iteration and $w(C') - \widetilde{\lambda}|C'| < 0$, then $|C'| < |\widetilde{C}|$.*

Proof.

$$\begin{aligned} 0 &> w(C') - \widetilde{\lambda}|C'| \\ &= w(C') - \lambda|C'| + \lambda|C'| - \widetilde{\lambda}|C'| \\ &\geq w(\widetilde{C}) - \lambda|\widetilde{C}| + \lambda|C'| - \widetilde{\lambda}|C'| \\ &= (|\widetilde{C}| - |C'|)(\widetilde{\lambda} - \lambda). \end{aligned}$$

Thus $|\widetilde{C}| > |C'|$. □

Now we can state the following.

THEOREM 4.5. *The problem of finding a minimum mean cycle reduces to a sequence of at most m Chinese postman problems.*

Thus the complexity of this procedure is $O(n^5)$ for complete graphs and $O(nm^2 \log n)$ in the general case.

We conclude this section with some simple observations. Assume now that the edge weights are nonnegative. Consider the following problems.

P1. Find a cycle of minimum weight.

P2. Find a cycle of minimum mean weight.

P3. Find a cycle of maximum weight.

P4. Find a cycle of maximum mean weight.

The directed versions of P1 and P2 can be solved with shortest path algorithms; see Lawler [17]. Problem P1 also reduces to m shortest path problems. However, P2 reduces to $O(m)$ Chinese postman problems.

It is well known that P3 is an NP-hard problem; however, P4 reduces to $O(m)$ Chinese postman problems.

5. Minimum mean cuts.

Now consider the problem of finding a cut of minimum mean weight. For planar graphs we can use planar duality to reduce the problem to the minimum mean cycle problem. For graphs with no K_5 minor we propose the following algorithm.

```
Step 0. Choose any cut $\widetilde{C}$.
Step 1. Define $\bar{w}(e) = w(e) - \lambda$, for $e \in E$, where $\lambda = w(\widetilde{C})/|\widetilde{C}|$.
Step 2. Find a cut of minimum weight with respect to $\bar{w}$.
Step 3. If the value of the cut is zero, stop.
        Otherwise, let $\widetilde{C}$ be the cut just obtained, go to Step 1.
```

The algorithm given in [2] can be used in Step 2. In this case the problem also reduces to a sequence of Chinese postman problems. Lemma 4.4 also applies to this case.

The minimum mean cut problem for general graphs is NP-hard, even if the weights are restricted to be nonnegative.

6. Final remarks.

We have seen that matching and minimum mean cycles are close relatives. A polynomial algorithm to solve one gives a simple algorithm for the other. It is surprising that a compact formulation for matching is not known and that the minimum mean cycle problem can be written as a polynomial size linear program.

A simple polynomial algorithm for matching has been given in §2. Its main operation is solving a polynomial size linear program. This can be done with any polynomial algorithm for linear programming, for instance, Karmarkar's method. The geometric interpretation is as follows. At any iteration we have an extreme point of the Chinese postman polyhedron and we optimize over the cone associated with this extreme point.

Acknowledgments. I am grateful to the referee for his suggestions on the presentation of this paper.

REFERENCES

[1] M. Ball, W. G. Liu, and W. R. Pulleyblank, *Two terminal Steiner tree polyhedra*, Report 87466-OR, Institut für Operations Research, Univ. Bonn, Germany, 1987.

[2] F. Barahona, *The max cut problem on graphs not contractible to K_5*, Oper. Res. Lett., 2 (1983), pp. 107–111.

[3] ———, *Planar multicommodity flows, max cut and the Chinese Postman Problem*, in Polyhedral Combinatorics, W. Cook and P. Seymour, eds., DIMACS, 1 (1990), pp. 189–202.

[4] F. Barahona, *On cuts and matchings in planar graphs*, Math. Programming, to appear.

[5] F. Barahona and A. R. Mahjoub, *Compositions of graphs and polyhedra* I: *Balanced and acyclic induced subgraphs*, SIAM J. Discrete Math., 7 (1994), to appear.

[6] ———, *Compositions of graphs and polyhedra* II: *Stable sets*, SIAM J. Discrete Math., 7 (1994), to appear.

[7] W. H. Cunningham, *Optimal attack and reinforcement of a network*, J. Assoc. Comput. Mach., 32 (1985), pp. 549–561.

[8] J. Edmonds, *Maximum matching and a polyhedron with* $(0,1)$*-vertices*, J. Res. Nat. Bur. Standards, 69 (1965) B, pp. 125–130.

[9] J. Edmonds and E. L. Johnson, *Matching, Euler Tours and the Chinese Postman*, Math. Programming, 5 (1973), pp. 88–124.

[10] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.

[11] A. V. Goldberg and R. E. Tarjan, *Finding minimum-cost circulations by canceling negative cycles*, J. Assoc. Comput. Mach., 36 (1989), pp. 873–886.

[12] M. Grötschel and O. Holland, *Solving matching problems with linear programming*, Math. Programming, 33 (1985), pp. 243–259.

[13] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–191.

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

[15] R. M. Karp and C. H. Papadimitriou, *On linear characterizations of combinatorial optimization problems*, SIAM J. Comput., 11 (1982), pp. 620–632.

[16] L. Khachiyan, *A polynomial algorithm in linear programming*, Soviet Math. Dokl., 20 (1979), pp. 191–194.

[17] E. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

[18] R. J. Lipton and R. E. Tarjan, *A separator theorem for planar graphs*, SIAM J. Appl. Math., 36 (1979), pp. 177–189.

[19] N. Maculan, *A new linear programming formulation for the shortest s-directed spanning tree problem*, Tech. Rep. ES 54-85, Systems Engineering and Computer Science, COPPE, Federal University of Rio de Janeiro, Brazil, 1985.

[20] K. Mei-Ko, *Graphic programming using odd or even points*, Chinese Math., 1 (1962), pp. 273–277.

[21] N. Megiddo, *Combinatorial Optimization with rational objective functions*, Math. Oper. Res., 4 (1979), pp. 414–424.

[22] J. B. Orlin, *On the simplex algorithm for networks and generalized networks*, Math. Programming Stud., 24 (1985), pp. 166–178.

[23] M. Padberg and M. R. Rao, *The Russian method for linear inequalities* III: *Bounded integer programming*, Report 81-39, GBA, New York University, 1981.

[24] ———, *Odd minimum cut-sets and b-matchings*, Math. Oper. Res., 7 (1982), pp. 67–80.

[25] P. D. Seymour, *Sums of circuits*, in Graph Theory and Related Topics, J. A. Bondy and U. S. R. Murty, eds., Academic Press, New York, 1979, pp. 341–355.

[26] R. T. Wong, *A dual ascent approach to Steiner tree problems in graphs*, Math. Programming, 28 (1984), pp. 271–287.

[27] M. Yannakakis, *Expressing combinatorial optimization problems by linear programs*, in Proc. 29th IEEE Symp. on Foundations of Computer Science, 1988, pp. 223–228.

SIAM J. OPTIMIZATION
Vol. 3, No. 4, pp. 696–725, November 1993

HIGHER-ORDER PREDICTOR-CORRECTOR INTERIOR POINT METHODS WITH APPLICATION TO QUADRATIC OBJECTIVES*

TAMRA J. CARPENTER[†], IRVIN J. LUSTIG[†], JOHN M. MULVEY[†], AND DAVID F. SHANNO[‡]

Abstract. In this paper, the authors explore the full utility of Mehrotra's predictor-corrector method in the context of linear and convex quadratic programs. They describe a procedure for doing multiple corrections at each iteration and implement it within the framework of OB1. Computational results are provided for the multiple correcting procedure using several strategies for determining the number of corrections in a given iteration. The results indicate that iteration counts can be significantly reduced by allowing higher-order corrections but at the the cost of extra work per iteration. The procedure is shown to be a level-m composite Newton interior point method, where m is the number of corrections performed in an iteration.

Key words. interior point methods, linear programming, quadratic programming, higher-order methods, predictor-corrector method, composite Newton method

AMS subject classifications. 90C05, 90C20

1. Introduction.

Mizuno, Todd, and Ye [9] introduce the term "predictor-corrector" into the lexicon of interior point methods to describe a particular algorithm which alternately takes primal-dual affine steps and centered steps. Their predictor step is the (uncentered) primal-dual affine step (studied by Monteiro, Adler, and Resende [11]), which is corrected by taking a (centering) step toward the central path. The algorithm, therefore, takes two steps within each interior point iteration. Each step requires factoring a matrix to obtain the step direction.

Motivated by predictor-corrector methods used in the differential equations literature, Mehrotra describes another predictor-corrector method in [7], which he derives using a second-order Taylor series approximation of the primal-dual trajectory in [8]. While Mehrotra's method also entails solving for two directions—the predictor and the corrector—in each iteration, it obtains both directions using a single factorization. Since both the predictor and the corrector are based on the same factorization, there is little additional work required to compute the corrector. Indeed, Mehrotra's combined strategy for computing the predictor-corrector direction, centering parameter, and steplength at each iteration performs remarkably well on a subset of the NETLIB (Gay [2]) problems.

Lustig, Marsten, and Shanno [5] extend Mehrotra's presentation to include bounds and also to implement the predictor-corrector method within the framework of the primal-dual interior point solver OB1 (Lustig, Marsten, and Shanno [4]). Their results on the full NETLIB test set demonstrate consistent reduction in both iterations and solution time. Tapia, Zhang, Saltzman, and Weiser [13] show that the predictor-

* Received by the editors December 6, 1990; accepted for publication (in revised form) July 10, 1992. The United States Government is authorized to reproduce and distribute reprints for governmental purposes not withstanding any copyright notations thereon.

[†] Program in Statistics and Operations Research, Department of Civil Engineering and Operations Research, Princeton University, Princeton, New Jersey 08544. The research of the third author was supported by National Science Foundation grant CCR-9102660 and the Mathematical and Analytics Computation Center of the IBM Corporation.

[‡] Rutgers Center for Operations Research, Rutgers University, New Brunswick, New Jersey 08903, and Visiting Fellow, Department of Civil Engineering and Operations Research, Princeton University. The research of this author was sponsored by the Air Force Office of Scientific Research, Air Force Systems Command grant AFOSR-87-0215.

corrector interior point method is equivalent to a level-1 composite Newton method and prove that it has a local convergence rate that is quadratic, under the standard assumptions. They note that the level-1 composite Newton method, without the interior point requirement, is cubically convergent under standard assumptions; however, the interior point aspect of the predictor-corrector method precludes a proof of cubic convergence. They demonstrate that the cubic convergence rate is preserved if the interior point requirement is abandoned locally to allow a steplength of one to be taken near the solution.

In this paper, we explore the full utility of Mehrotra's predictor-corrector method in the context of both linear and convex quadratic programs. First, we describe a procedure for doing multiple corrections at each iteration that is based on the Lustig, Marsten, and Shanno [5] implementation of the predictor-corrector method for linear programs, and we show that this procedure is a level-m composite Newton interior point method as described in Tapia et al. [13]. The procedure is tested using several strategies for dynamically determining m, the number of corrections in a given iteration. The results indicate that iteration counts may be reduced by allowing higher-order correcting but with more work per iteration.

The second part extends the predictor-corrector method to convex quadratic programs. The quadratic predictor-corrector procedure is implemented within the framework of the quadratic extension (OBN) of OB1 described in Carpenter, Lustig, Mulvey, and Shanno [1]. The predictor-corrector implementation (one corrector step) is compared with higher-order variants and the basic primal-dual method.

Section 2 describes Mehrotra's predictor-corrector procedure and discusses the extension of the Lustig, Marsten, and Shanno [5] implementation required to perform multiple corrections at each iteration. The multiple predictor-corrector procedure is then shown to be a level-m composite Newton method. Section 3 develops criteria for determining when to stop correcting. Computational results using several strategies for correcting appear in §4. The extension to convex quadratic programs is presented in §5 with computational results reported in §6. The last section contains conclusions and a discussion of future research.

2. Multiple predictor-corrector as a composite Newton method. We consider the following linear programming problem in standard form:

$$\begin{aligned} \text{minimize} \quad & c^T x, \\ \text{subject to} \quad & Ax = b, \\ & x + s = u, \\ & x, s \geq 0. \end{aligned} \tag{1}$$

Some or all of the upper bounds u may be infinite, and slack variables s are added to transform upper bound inequalities to equalities. We assume that $A \in \Re^{m \times n}$, $b \in \Re^m$, and c, u, x, $s \in \Re^n$. The standard logarithmic barrier interior point method eliminates the remaining inequalities by incorporating them into a logarithmic barrier term appended to the objective to obtain the following transformed problem:

$$\begin{aligned} \text{minimize} \quad & c^T x - \mu \sum_{j=1}^{n} \ln x_j - \mu \sum_{j=1}^{n} \ln s_j, \\ \text{subject to} \quad & Ax = b, \\ & x + s = u. \end{aligned} \tag{2}$$

The first-order conditions for (1) are

$$F(x,s,y,z,w) = \begin{pmatrix} Ax-b \\ x+s-u \\ A^Ty+z-w-c \\ XZe \\ SWe \end{pmatrix} = 0 \quad \text{and} \quad x,s,z,w \geq 0, \tag{3}$$

where X, Z, S, and W are diagonal matrices with the elements x_j, z_j, s_j, and w_j, respectively, and y, w, and z are dual variables. Similarly, the first-order conditions for (2) are

$$\begin{pmatrix} Ax-b \\ x+s-u \\ A^Ty+z-w-c \\ XZe-\mu e \\ SWe-\mu e \end{pmatrix} = 0. \tag{4}$$

The search direction of the standard primal-dual interior point algorithm as described in [4] has two components: the "affine" direction and the "centering" direction. If we let $\Delta v = (\Delta x, \Delta s, \Delta y, \Delta z, \Delta w)$ and $v = (x,s,y,z,w)$, applying Newton's method to (3) yields the following system of equations (5), which is solved for the affine direction Δv^0:

$$F'(v)(\Delta v^0) = \begin{pmatrix} A\Delta x^0 \\ \Delta x^0 + \Delta s^0 \\ A^T\Delta y^0 + \Delta z^0 - \Delta w^0 \\ Z\Delta x^0 + X\Delta z^0 \\ W\Delta s^0 + S\Delta w^0 \end{pmatrix} = \begin{pmatrix} b-Ax \\ u-x-s \\ c-A^Ty-z+w \\ -XZe \\ -SWe \end{pmatrix} = -F(v). \tag{5}$$

The centering direction Δv_μ is the solution to

$$\begin{pmatrix} A\Delta x_\mu \\ \Delta x_\mu + \Delta s_\mu \\ A^T\Delta y_\mu + \Delta z_\mu - \Delta w_\mu \\ Z\Delta x_\mu + X\Delta z_\mu \\ W\Delta s_\mu + S\Delta w_\mu \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \mu e \\ \mu e \end{pmatrix}. \tag{6}$$

The primal-dual search direction is then $\Delta v = \Delta v^0 + \Delta v_\mu$. Alternatively, the direction Δv is obtained by applying Newton's method directly to (4). Thus, the standard primal-dual logarithmic barrier method applies Newton's method directly to (4), while the affine variant of the primal-dual interior point method applies Newton's method to (3) to obtain only Δv^0. The steps of the primal-dual logarithmic barrier method include a centering component incorporated through μ, whereas the steps of the affine variant do not. In either case, the step direction in the resulting interior point method is obtained by applying Newton's method to a system of nonlinear equations—either (3) or (4).

In general, Newton's method is an iterative procedure that finds zeros of a nonlinear function $f(x)$. At each iteration it

$$\begin{aligned} &\text{solves } f'(x^k)\Delta x = -f(x^k) \quad \text{for } \Delta x \\ &\text{and sets } x^{k+1} = x^k + \alpha \Delta x. \end{aligned} \tag{7}$$

The inclusion of $0 < \alpha < 1$ makes this a *damped* Newton method. Since it is often difficult to compute the requisite derivative in (7), it may be advantageous to use the same derivative evaluation in several solves. This approach is beneficial if it reduces the overall number of derivative evaluations without performing an unreasonable number of extra solves. This is the idea behind the composite Newton method. At each iteration the damped level-m composite Newton method

$$\begin{aligned}
&\text{solves } f'(x^k)\Delta x^0 = -f(x^k) \quad \text{for } \Delta x^0\\
&\text{then solves } f'(x^k)\Delta x^i = -f\left(x^k + \sum_{j=0}^{i-1} \Delta x^j\right) \quad \text{for } \Delta x^i, \quad i = 1, \dots, m\\
&\text{and sets } x^{k+1} = x^k + \alpha \sum_{j=0}^{m} \Delta x^j.
\end{aligned}$$

In this case, the derivative is employed $m + 1$ times to iteratively obtain the direction before a step is taken. Thus, the composite Newton method performs more solves within each iteration with the intent of performing fewer iterations and therefore fewer derivative evaluations overall.

The composite Newton interior point method is presented in [13]. The statement that they provide allows for μ to be respecified in each inner iteration, but we state a slightly less general method which fixes μ outside of the inner loop and consider the more general statement later.

```
ALGORITHM CNM (composite Newton interior point method).
Given $v^k = (x^k, s^k, y^k, z^k, w^k)$ with $x^k, s^k, z^k, w^k > 0$.
  Step 1: Solve (5) for the affine direction $\Delta v^0$. Let $\Delta \hat{v}^0 = \Delta v^0$.
  Step 2: Compute $\mu(v^k, \Delta \hat{v}^0)$.
  Step 3:
          For $i = 1, \dots, m_k$ do
              Solve $F'(v^k)\Delta \hat{v}^i = -F(v^k + \sum_{j=0}^{i-1} \Delta \hat{v}^j) + \mu \hat{e}$ for $\Delta \hat{v}^i$,
              where $\hat{e}$ is a vector with 1 in the last $2n$ components and 0
              otherwise.
          end do
          Define $\Delta \hat{v} = \sum_{j=0}^{m_k} \Delta \hat{v}^j$.
  Step 4: Perform ratio test to determine primal and dual steplengths $\alpha_p$ and $\alpha_d$.
  Step 5: Move to the new point $v^{k+1}$ defined by
```

$$\begin{aligned}
x^{k+1} &= x^k + \alpha_p \Delta \hat{x},\\
s^{k+1} &= s^k + \alpha_p \Delta \hat{s},\\
y^{k+1} &= y^k + \alpha_d \Delta \hat{y},\\
z^{k+1} &= z^k + \alpha_d \Delta \hat{z},\\
w^{k+1} &= w^k + \alpha_d \Delta \hat{w}.
\end{aligned}$$

Tapia et al. [13] proved that Mehrotra's predictor-corrector method is a damped level-1 composite Newton interior point method. The predictor-corrector method computes its search direction in two stages. First, it computes the affine direction and uses it as a predictor. The predictor direction is used in two ways: (1) to set the barrier parameter μ; and (2) to correct the centered direction that would be obtained by applying Newton's method to (4).

First, Mehrotra solves (5) for the predictor direction Δv^0, then he computes the barrier parameter μ as a function of both the current point v and Δv^0, and finally, he uses Δv^0 to correct the centered direction that would be obtained by applying Newton's method to (4). Mehrotra suggests computing a combined centering/correction direction Δv_c as the solution to the system

$$\begin{aligned} A\Delta x_c &= 0, \\ \Delta x_c + \Delta s_c &= 0, \\ A^T\Delta y_c + \Delta z_c - \Delta w_c &= 0, \\ X\Delta z_c + Z\Delta x_c &= \mu e - \Delta X^0 \Delta Z^0 e, \\ S\Delta w_c + W\Delta s_c &= \mu e - \Delta S^0 \Delta W^0 e, \end{aligned} \tag{8}$$

where ΔX^0, ΔZ^0, ΔS^0, and ΔW^0 are diagonal matrices having elements Δx^0, Δz^0, Δs^0, and Δw^0, respectively. The full predictor-corrector direction is then

$$\begin{aligned} \Delta y &= \Delta y^0 + \Delta y_c, \\ \Delta x &= \Delta x^0 + \Delta x_c, \\ \Delta z &= \Delta z^0 + \Delta z_c, \\ \Delta w &= \Delta w^0 + \Delta w_c, \\ \Delta s &= \Delta s^0 + \Delta s_c. \end{aligned} \tag{9}$$

Alternatively, Lustig et al. [5] compute the *full* direction Δv directly by solving

$$\begin{aligned} A\Delta x &= b - Ax, \\ \Delta x + \Delta s &= u - x - s, \\ A^T\Delta y + \Delta z - \Delta w &= c - A^T y - z + w, \\ X\Delta z + Z\Delta x &= \mu e - XZe - \Delta X^0 \Delta Z^0 e, \\ S\Delta w + W\Delta s &= \mu e - SWe - \Delta S^0 \Delta W^0 e, \end{aligned} \tag{10}$$

for Δx, Δy, Δz, Δw, and Δs. System (10) includes the correction terms $\Delta X^0 \Delta Z^0 e$ and $\Delta S^0 \Delta W^0 e$ in the right-hand side of the Newton system for (2).

The systems we solve to obtain the predictor, the corrector, or the full predictor-corrector direction each involve the same matrix. That is, each of these directions is obtained based on the evaluation of the same derivative, just as successive directions in the composite Newton method involve the same derivative. The idea of the predictor-corrector procedure is to reduce the work required in the primal-dual interior point procedure by reusing the factorization required to solve the Newton system (5).

This method performs only one correction in obtaining the direction, but can easily be generalized to perform several corrections in a multiple predictor-corrector procedure which directly extends (10). Instead of solving this system once at each step of the primal-dual interior point method, it can be solved repetitively with each direction corrected based on the previous direction. The number of corrections m_k in an iteration k is dynamically chosen. The resulting procedure is outlined below in Algorithm MPC, which is invoked at each iteration.

```
ALGORITHM MPC (multiple predictor-corrector).
Given $v^k = (x^k, s^k, y^k, z^k, w^k)$ with $x^k, s^k, z^k, w^k > 0$.
    Step 1: Solve (5) for the affine direction $\Delta v^0$.
    Step 2: Compute $\mu(v^k, \Delta v^0)$.
```

Step 3:

```
    For $i = 1, \ldots, m_k$ do
        Solve the following system for $\Delta v^i$:
```

$$
(11) \qquad \begin{pmatrix} A\Delta x^i \\ \Delta x^i + \Delta s^i \\ A^T \Delta y^i + \Delta z^i - \Delta w^i \\ Z\Delta x^i + X\Delta z^i \\ W\Delta s^i + S\Delta w^i \end{pmatrix} = \begin{pmatrix} b - Ax \\ u - x - s \\ c - A^T y - z + w \\ \mu e - XZe - \Delta X^{i-1}\Delta Z^{i-1} e \\ \mu e - SWe - \Delta S^{i-1}\Delta W^{i-1} e \end{pmatrix}.
$$

```
    end do
    Define $\Delta v = \Delta v^{m_k}$.
```

Step 4: Perform ratio test to determine primal and dual steplengths α_p and α_d.

Step 5: Move to the new point v^{k+1} defined by

$$
(12) \qquad \begin{aligned} x^{k+1} &= x^k + \alpha_p \Delta x, \\ s^{k+1} &= s^k + \alpha_p \Delta s, \\ y^{k+1} &= y^k + \alpha_d \Delta y, \\ z^{k+1} &= z^k + \alpha_d \Delta z, \\ w^{k+1} &= w^k + \alpha_d \Delta w. \end{aligned}
$$

We typically solve the system in Step 3 by reducing it to a positive definite system by expressing all variables in terms of Δy. The matrix in the system for Δy is then AD^2A^T, where $D = (X^{-1}Z + S^{-1}W)^{-1}$. Solving (12), therefore, requires a factorization of the matrix AD^2A^T and a *backsolve* with the factorization $LL^T = AD^2A^T$ yields Δv^i. Hence, the predictor-corrector procedure reduces the work of the primal-dual interior point method by reusing the factorization of AD^2A^T.

The full statement of MPC requires the definition of μ, α_p, α_d, m_k, and an initial solution, which we defer until the next section. The remainder of this section is devoted to proving our main result, which is stated in Theorem 2.1.

THEOREM 2.1. *The multiple predictor-corrector algorithm* (MPC) *is equivalent to the composite Newton method* (CNM).

We say that two algorithms are equivalent if they yield the same sequence of iterates when started from the same initial point. Therefore, since the statements of MPC and CNM are identical with the exception of Step 3, the proof of the theorem requires only that the direction Δv obtained by MPC is the same as $\Delta \hat{v}$ from CNM. Tapia et al. [13] have shown that Theorem 2.1 is true when $m_k = 1$ in every iteration k; this is a special case of the more general statement in Theorem 2.1. Before proceeding with the proof of the theorem we recall that

$$
(13) \qquad F'(v)(\Delta v) = \begin{pmatrix} A\Delta x \\ \Delta x + \Delta s \\ A^T \Delta y + \Delta z - \Delta w \\ Z\Delta x + X\Delta z \\ W\Delta s + S\Delta w \end{pmatrix} \quad \text{and} \quad -F(v) = \begin{pmatrix} b - Ax \\ u - x - s \\ c - A^T y - z + w \\ -XZe \\ -SWe \end{pmatrix},
$$

and we present the following lemmas.

LEMMA 2.2. *For $i \geq 1$,*

$$\begin{pmatrix} b - A(x^k + \sum_{j=0}^{i} \Delta\hat{x}^j) \\ u - (x^k + \sum_{j=0}^{i} \Delta\hat{x}^j) - (s^k + \sum_{j=0}^{i} \Delta\hat{s}^j) \\ c - A^T(y^k + \sum_{j=0}^{i} \Delta\hat{y}^j) - (z^k + \sum_{j=0}^{i} \Delta\hat{z}^j) + (w^k + \sum_{j=0}^{i} \Delta\hat{w}^j) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Proof. By Step 3 of CNM and the definitions of $F(v)$ and $F'(v)$ in (13), we have that

$$\begin{pmatrix} A\Delta\hat{x}^i \\ \Delta\hat{x}^i + \Delta\hat{s}^i \\ A^T\Delta\hat{y}^i + \Delta\hat{z}^i - \Delta\hat{w}^i \end{pmatrix}$$
$$= \begin{pmatrix} b - A(x^k + \sum_{j=0}^{i-1} \Delta\hat{x}^j) \\ u - (x^k + \sum_{j=0}^{i-1} \Delta\hat{x}^j) - (s^k + \sum_{j=0}^{i-1} \Delta\hat{s}^j) \\ c - A^T(y^k + \sum_{j=0}^{i-1} \Delta\hat{y}^j) - (z^k + \sum_{j=0}^{i-1} \Delta\hat{z}^j) + (w^k + \sum_{j=0}^{i-1} \Delta\hat{w}^j) \end{pmatrix},$$

which immediately implies the result. □

LEMMA 2.3. *For $i \geq 1$,*

$$\begin{pmatrix} (X^k + \sum_{j=0}^{i} \Delta\hat{X}^j)(Z^k + \sum_{j=0}^{i} \Delta\hat{Z}^j)e \\ (S^k + \sum_{j=0}^{i} \Delta\hat{S}^j)(W^k + \sum_{j=0}^{i} \Delta\hat{W}^j)e \end{pmatrix}$$
$$= \begin{pmatrix} \mu e + (\sum_{j=0}^{i} \Delta\hat{X}^j)(\sum_{j=0}^{i} \Delta\hat{Z}^j)e - (\sum_{j=0}^{i-1} \Delta\hat{X}^j)(\sum_{j=0}^{i-1} \Delta\hat{Z}^j)e \\ \mu e + (\sum_{j=0}^{i} \Delta\hat{S}^j)(\sum_{j=0}^{i} \Delta\hat{W}^j)e - (\sum_{j=0}^{i-1} \Delta\hat{S}^j)(\sum_{j=0}^{i-1} \Delta\hat{W}^j)e \end{pmatrix}.$$

Proof. Multiplying term by term, we have that

$$\left(X^k + \sum_{j=0}^{i} \Delta\hat{X}^j\right)\left(Z^k + \sum_{j=0}^{i} \Delta\hat{Z}^j\right)e \tag{14}$$

$$= X^k Z^k e + \sum_{j=0}^{i} (X^k \Delta\hat{Z}^j + Z^k \Delta\hat{X}^j)e + \left(\sum_{j=0}^{i} \Delta\hat{X}^j\right)\left(\sum_{j=0}^{i} \Delta\hat{Z}^j\right)e. \tag{15}$$

Step 3 of CNM and (13) provide that

$$X^k \Delta\hat{Z}^i + Z^k \Delta\hat{X}^i = \mu e - \left(X^k + \sum_{j=0}^{i-1} \Delta\hat{X}^j\right)\left(Z^k + \sum_{j=0}^{i-1} \Delta\hat{Z}^j\right)e, \tag{16}$$

which implies

$$X^k Z^k e + \sum_{j=0}^{i} (X^k \Delta\hat{Z}^j + Z^k \Delta\hat{X}^j)e + \left(\sum_{j=0}^{i-1} \Delta\hat{X}^j\right)\left(\sum_{j=0}^{i-1} \Delta\hat{Z}^j\right)e = \mu e. \tag{17}$$

We can add and subtract $(\sum_{j=0}^{i-1} \Delta\hat{X}^j)(\sum_{j=0}^{i-1} \Delta\hat{Z}^j)e$ in the right-hand side of equation (15) to get

$$\begin{aligned}\left(X^k+\sum_{j=0}^{i}\Delta\hat{X}^j\right)\left(Z^k+\sum_{j=0}^{i}\Delta\hat{Z}^j\right)e &= X^kZ^ke+\sum_{j=0}^{i}(X^k\Delta\hat{Z}^j+Z^k\Delta\hat{X}^j)e\\ &+\left(\sum_{j=0}^{i-1}\Delta\hat{X}^j\right)\left(\sum_{j=0}^{i-1}\Delta\hat{Z}^j\right)e+\left(\sum_{j=0}^{i}\Delta\hat{X}^j\right)\left(\sum_{j=0}^{i}\Delta\hat{Z}^j\right)e\\ &-\left(\sum_{j=0}^{i-1}\Delta\hat{X}^j\right)\left(\sum_{j=0}^{i-1}\Delta\hat{Z}^j\right)e.\end{aligned}\tag{18}$$

Using (17) to substitute μe for the first three terms yields the result

$$\begin{aligned}&\left(X^k+\sum_{j=0}^{i}\Delta\hat{X}^j\right)\left(Z^k+\sum_{j=0}^{i}\Delta\hat{Z}^j\right)e\\ &\quad=\mu e+\left(\sum_{j=0}^{i}\Delta\hat{X}^j\right)\left(\sum_{j=0}^{i}\Delta\hat{Z}^j\right)e-\left(\sum_{j=0}^{i-1}\Delta\hat{X}^j\right)\left(\sum_{j=0}^{i-1}\Delta\hat{Z}^j\right)e.\end{aligned}$$

Applying the analogous approach to the S and W equations proves the lemma. □

Combining the results of Lemmas 2.2 and 2.3 we have that

$$\begin{aligned}&F\left(v^k+\sum_{j=0}^{i}\Delta\hat{v}^j\right)\\ &=\begin{pmatrix} b-A(x^k+\sum_{j=0}^{i}\Delta\hat{x}^j)\\ u-(x^k+\sum_{j=0}^{i}\Delta\hat{x}^j)-(s^k+\sum_{j=0}^{i}\Delta\hat{s}^j)\\ c-A^T(y^k+\sum_{j=0}^{i}\Delta\hat{y}^j)-(z^k+\sum_{j=0}^{i}\Delta\hat{z}^j)+(w^k+\sum_{j=0}^{i}\Delta\hat{w}^j)\\ (X^k+\sum_{j=0}^{i}\Delta\hat{X}^j)(Z^k+\sum_{j=0}^{i}\Delta\hat{Z}^j)e\\ (S^k+\sum_{j=0}^{i}\Delta\hat{S}^j)(W^k+\sum_{j=0}^{i}\Delta\hat{W}^j)e\end{pmatrix}\\ &=\begin{pmatrix}0\\0\\0\\ \mu e+(\sum_{j=0}^{i}\Delta\hat{X}^j)(\sum_{j=0}^{i}\Delta\hat{Z}^j)e-(\sum_{j=0}^{i-1}\Delta\hat{X}^j)(\sum_{j=0}^{i-1}\Delta\hat{Z}^j)e\\ \mu e+(\sum_{j=0}^{i}\Delta\hat{S}^j)(\sum_{j=0}^{i}\Delta\hat{W}^j)e-(\sum_{j=0}^{i-1}\Delta\hat{S}^j)(\sum_{j=0}^{i-1}\Delta\hat{W}^j)e\end{pmatrix}.\end{aligned}\tag{19}$$

Proving Theorem 2.1 requires that we show that the direction $\Delta\hat{v}=\sum_{i=0}^{m_k}\Delta\hat{v}^i$ is the same as the direction Δv^{m_k} obtained by MPC. To do this we note that $\Delta\hat{v}$ solves the system

$$\begin{pmatrix}A\Delta\hat{x}\\ \Delta\hat{x}+\Delta\hat{s}\\ A^T\Delta\hat{y}+\Delta\hat{z}-\Delta\hat{w}\\ Z\Delta\hat{x}+X\Delta\hat{z}\\ W\Delta\hat{s}+S\Delta\hat{w}\end{pmatrix}=\begin{pmatrix}b-Ax\\ u-x-s\\ c-A^Ty-z+w\\ \mu e-XZe-(\sum_{i=0}^{m_k-1}\Delta\hat{X}^i)(\sum_{i=0}^{m_k-1}\Delta\hat{Z}^i)e\\ \mu e-SWe-(\sum_{i=0}^{m_k-1}\Delta\hat{S}^i)(\sum_{i=0}^{m_k-1}\Delta\hat{W}^i)e\end{pmatrix}.\tag{20}$$

Thus, Δv^{m_k} and $\Delta\hat{v}$ solve precisely the same system when $(\sum_{i=0}^{m_k-1}\Delta\hat{V}^i)=\Delta V^{m_k-1}$.

LEMMA 2.4. $(\sum_{i=0}^{p}\Delta\hat{v}^i) = \Delta v^p$ *for all* $p \geq 0$.

Proof (by induction). $(p = 0)$ $\Delta v^0 = \Delta\hat{v}^0$ by definition in CNM. Both Δv^0 and $\Delta\hat{v}^0$ are the affine direction.

$(p = m)$ Assume that the lemma is true for $1 \leq p \leq m-1$. We must demonstrate that it is true for $p = m$. By adding up the systems defining each $\Delta\hat{v}^i$ we have that $\sum_{i=0}^{m}\Delta\hat{v}^i$ solves the system

(21)

$$\begin{pmatrix} A\sum_{i=0}^{m}\Delta\hat{x}^i \\ \sum_{i=0}^{m}\Delta\hat{x}^i + \sum_{i=0}^{m}\Delta\hat{s}^i \\ A^T(\sum_{i=0}^{m}\Delta\hat{y}^i) + \sum_{i=0}^{m}\Delta\hat{z}^i - \sum_{i=0}^{m}\Delta\hat{w}^i \\ Z(\sum_{i=0}^{m}\Delta\hat{x}^i) + X(\sum_{i=0}^{m}\Delta\hat{z}^i) \\ W(\sum_{i=0}^{m}\Delta\hat{s}^i) + S(\sum_{i=0}^{m}\Delta\hat{w}^i) \end{pmatrix}$$

$$= -F(v) + \sum_{i=0}^{m-1}\left[\mu\hat{e} - F\left(v + \sum_{j=0}^{i}\Delta\hat{v}^j\right)\right].$$

Applying (19) to replace $F(v + \sum_{j=0}^{i}\Delta\hat{v}^j)$ in (21) yields

(22)

$$-F(v) + \sum_{i=0}^{m-1}\left[\mu\hat{e} - F\left(v + \sum_{j=0}^{i}\Delta\hat{v}^j\right)\right] =$$

$$\begin{pmatrix} b - Ax \\ u - x - s \\ c - A^Ty - z + w \\ \mu e - XZe - \Delta\hat{X}^0\Delta\hat{Z}^0 + \{\sum_{i=1}^{m-1}(\mu e - [\mu e + (\sum_{j=0}^{i}\Delta\hat{X}^j)(\sum_{j=0}^{i}\Delta\hat{Z}^j)e \\ \qquad -(\sum_{j=0}^{i-1}\Delta\hat{X}^j)(\sum_{j=0}^{i-1}\Delta\hat{Z}^j)e])\} \\ \mu e - SWe - \Delta\hat{S}^0\Delta\hat{W}^0 + \{\sum_{i=1}^{m-1}(\mu e - [\mu e + (\sum_{j=0}^{i}\Delta\hat{S}^j)(\sum_{j=0}^{i}\Delta\hat{W}^j)e \\ \qquad -(\sum_{j=0}^{i-1}\Delta\hat{S}^j)(\sum_{j=0}^{i-1}\Delta\hat{W}^j)e])\} \end{pmatrix}.$$

Within the braces, the μ terms cancel as do alternating terms in the summation over i. This implies

$$-F(v) + \sum_{i=0}^{m-1}\left[\mu\hat{e} - F\left(v + \sum_{j=0}^{i}\Delta\hat{v}^j\right)\right]$$

$$= \begin{pmatrix} b - Ax \\ u - x - s \\ c - A^Ty - z + w \\ \mu e - XZe - (\sum_{i=0}^{m-1}\Delta\hat{X}^i)(\sum_{i=0}^{m-1}\Delta\hat{Z}^i)e \\ \mu e - SWe - (\sum_{i=0}^{m-1}\Delta\hat{S}^i)(\sum_{i=0}^{m-1}\Delta\hat{W}^i)e \end{pmatrix}$$

$$= \begin{pmatrix} b - Ax \\ u - x - s \\ c - A^Ty - z + w \\ \mu e - XZe - \Delta X^{m-1}\Delta Z^{m-1}e \\ \mu e - SWe - \Delta S^{m-1}\Delta W^{m-1}e \end{pmatrix} \quad \text{(by the induction hypothesis).}$$

Thus, $\sum_{i=0}^{m}\Delta\hat{v}^i$ and Δv^m are the solution to the same system. □

Proof of Theorem 2.1. The proof follows directly from Lemma 2.4 applied when $p = m_k$. □

3. How much correcting? We now define the parameters used in MPC to provide a complete statement of the algorithm. The starting solution and the parameters μ, α_p, and α_d are computed as described in [5]. We briefly review these definitions and then discuss setting m_k in the remainder of this section.

The primal and dual steplengths α_p and α_d are chosen to insure the nonnegativity of the variables x, s, z, and w. Given v and a direction Δv, the ratio functions $r_p(v, \Delta v)$ and $r_d(v, \Delta v)$ are defined as

$$\begin{aligned} r_p(v,\Delta v) &= \min\left\{\min_j\left\{\frac{x_j}{-\Delta x_j}, \Delta x_j < 0\right\}, \min_j\left\{\frac{s_j}{-\Delta s_j}, \Delta s_j < 0\right\}\right\}, \\ r_d(v,\Delta v) &= \min\left\{\min_j\left\{\frac{z_j}{-\Delta z_j}, \Delta z_j < 0\right\}, \min_j\left\{\frac{w_j}{-\Delta w_j}, \Delta w_j < 0\right\}\right\}. \end{aligned} \tag{23}$$

$r_p(v, \Delta v)$ and $r_d(v, \Delta v)$ are the maximum steps that can be taken before encountering a boundary in the primal and dual, respectively, whereas α_p and α_d are the steps that are actually taken in MPC. They are defined as

$$\alpha_p = 0.99995\, r_p(v, \Delta v), \qquad \alpha_d = 0.99995\, r_d(v, \Delta v). \tag{24}$$

The barrier parameter μ is computed as a function of the complementarity that would result if a step were taken in the affine direction. The allowable primal and dual steplengths in the affine direction are

$$\alpha_p^0 = 0.99995\, r_p(v^k, \Delta v^0), \qquad \alpha_d^0 = 0.99995\, r_d(v^k, \Delta v^0), \tag{25}$$

and the resulting complementarity would be

$$g^0 = (x + \alpha_p^0 \Delta x^0)^T (z + \alpha_d^0 \Delta z^0) + (s + \alpha_p^0 \Delta s^0)^T (w + \alpha_d^0 \Delta w^0). \tag{26}$$

Then we choose

$$\mu = \left(\frac{g^0}{x^T z + s^T w}\right)^2 \left(\frac{g^0}{n}\right) \quad \text{when } x^T z + s^T w \geq 1, \tag{27}$$

or

$$\mu = \frac{(x^T z + s^T w)}{\phi(n)} \quad \text{when } x^T z + s^T w < 1, \tag{28}$$

where

$$\phi(n) = \begin{cases} n^2 & \text{if } n \leq 5000, \\ n^{\frac{3}{2}} & \text{if } n > 5000, \end{cases} \tag{29}$$

as defined in [5].

The initial starting point is prescribed according to Mehrotra [7] using the procedure implemented by Lustig et al. [5].

Now, we consider the definition of m_k and begin by noting that when $m_k = 1$ for all k, MPC is the predictor-corrector procedure implemented in [5]. Tapia et al. [13] suggested allowing the number of corrections to vary at each iteration of the predictor-corrector interior point method. Our computational results demonstrate that it is not only desirable but imperative to the success of a higher-order strategy. An example in which we run MPC on the problem AGG with $m_k = 2$ in every iteration highlights this point; the results are summarized in Table 1. Let d_p, d_d, and d_u be the primal, dual, and upper bound infeasibilities at the current point defined as

$$\begin{aligned} d_p &= b - Ax, \\ d_d &= c - A^T y - z + w, \quad \text{and} \\ d_u &= u - x - s. \end{aligned} \tag{30}$$

The total infeasibility reported in Table 1 is the sum of the absolute infeasibilities which is $\|d_p\|_1 + \|d_u\|_1 + \|d_u\|_1$ while the total complementarity is $x^T z + s^T w$. At almost every iteration, the second correction degrades performance of the algorithm. Complementarity and infeasibility often increase with the second correction, and the steplengths are reduced to the point where the algorithm stalls by iteration 19.

In this section, we examine the critical issue of how much correcting *is* advantageous. We first consider the effect of correcting from a *feasible* point and then examine the general case of correcting from an *infeasible* point. Based on this analysis, we develop a heuristic strategy for determining when to stop correcting, which defines the value m_k.

3.1. The feasible case. Given that the current estimate to the optimal point is primal and dual feasible, the directions Δv computed in MPC are always *feasible directions*; therefore, they satisfy

$$\begin{aligned} A\Delta x &= 0, \\ \Delta x + \Delta s &= 0, \\ A^T \Delta y + \Delta z - \Delta w &= 0. \end{aligned} \tag{31}$$

LEMMA 3.1. *For any* $(\Delta x, \Delta s, \Delta y, \Delta z, \Delta w)$ *that satisfy* (31),

$$\Delta x^T \Delta z + \Delta s^T \Delta w = 0.$$

Proof. Using (31) to substitute $\Delta z = \Delta w - A^T \Delta y$ and $\Delta x = -\Delta s$, we have

$$\begin{aligned} \Delta x^T \Delta z + \Delta s^T \Delta w &= \Delta x^T (\Delta w - A^T \Delta y) + \Delta s^T \Delta w \\ &= -\Delta s^T \Delta w - (A\Delta x)^T \Delta y + \Delta s^T \Delta w \\ &= 0. \qquad \square \end{aligned}$$

In addition to feasibility, the only requirement for optimality is that the duality gap is zero. When feasible, the complementarity $x^T z + s^T w$ equals the duality gap $c^T x - b^T y - u^T w$. The sole motivation behind performing corrections from a feasible point is reducing complementarity. We show that correcting to achieve either steeper decrease in complementarity or a longer steplength is advantageous.

Kojima, Mizuno, and Yoshise [3] showed that with a common steplength α and without upper bounds, the complementarity decreases linearly at each iteration of the primal-dual interior point method

$$(x + \alpha \Delta x)^T (z + \alpha \Delta z) = x^T z - \alpha (x^T z - n\mu).$$

A similar result is also true for the more general case of upper bounded variables and separately chosen primal and dual steplengths. The key observation is that

TABLE 1
Iteration statistics for problem AGG.

		Total	Total	Predicted Steplength	
		Comple.	Infeasibility	Primal	Dual
Iteration 1	Correction: 1	.650 D+09	.428 D+08	.850 D+00	.954 D+00
	2	.638 D+09	.411 D+08	.855 D+00	.951 D+00
Iteration 2	Correction: 1	.447 D+09	.275 D+08	.436 D+00	.677 D+00
	2	.639 D+09	.402 D+08	.515 D-03	.178 D-02
Iteration 3	Correction: 1	.370 D+09	.227 D+08	.437 D+00	.677 D+00
	2	.639 D+09	.402 D+08	.257 D-07	.954 D-07
Iteration 4	Correction: 1	.370 D+09	.227 D+08	.437 D+00	.677 D+00
	2	.639 D+09	.402 D+08	.129 D-11	.477 D-11
Iteration 5	Correction: 1	.382 D+09	.225 D+08	.442 D+00	.664 D+00
	2	.644 D+09	.402 D+08	.702 D-16	.444 D+00
Iteration 6	Correction: 1	.212 D+09	.101 D+08	.752 D+00	.753 D+00
	2	.353 D+09	.120 D+08	.704 D+00	.159 D-02
Iteration 7	Correction: 1	.152 D+09	.600 D+07	.497 D+00	.716 D+00
	2	.237 D+09	.119 D+08	.238 D-02	.657 D+00
Iteration 8	Correction: 1	.747 D+08	.404 D+07	.662 D+00	.853 D+00
	2	.137 D+09	.332 D+07	.723 D+00	.498 D-02
Iteration 9	Correction: 1	.559 D+08	.240 D+07	.271 D+00	.827 D+00
	2	.898 D+08	.328 D+07	.295 D-02	.578 D+00
Iteration 10	Correction: 1	.734 D+08	.212 D+07	.355 D+00	.153 D+00
	2	.895 D+08	.279 D+07	.150 D+00	.185 D+00
Iteration 11	Correction: 1	.217 D+08	.871 D+06	.691 D+00	.948 D+00
	2	.274 D+08	.493 D+06	.619 D+00	.827 D+00
Iteration 12	Correction: 1	.199 D+08	.305 D+06	.372 D+00	.285 D+00
	2	.121 D+08	.377 D+06	.223 D+00	.357 D+00
Iteration 13	Correction: 1	.141 D+08	.200 D+06	.470 D+00	.426 D+00
	2	.213 D+08	.376 D+06	.552 D-04	.109 D-03
Iteration 14	Correction: 1	.141 D+08	.200 D+06	.470 D+00	.425 D+00
	2	.213 D+08	.376 D+06	.277 D-08	.543 D-08
Iteration 15	Correction: 1	.141 D+08	.200 D+06	.470 D+00	.425 D+00
	2	.213 D+08	.376 D+06	.139 D-12	.271 D-12
Iteration 16	Correction: 1	.141 D+08	.200 D+06	.470 D+00	.425 D+00
	2	.213 D+08	.376 D+06	.570 D-17	.231 D-01
Iteration 17	Correction: 1	.213 D+08	.376 D+06	.121 D-03	.302 D-04
	2	.213 D+08	.376 D+06	.366 D-34	.262 D-18
Iteration 18	Correction: 1	.212 D+08	.376 D+06	.604 D-08	.151 D-08
	2	.211 D+08	.376 D+06	.559 D-51	.104 D-25

$x^T\Delta z + z^T\Delta x = n\mu - x^Tz$ and $s^T\Delta w + w^T\Delta s = n\mu - s^Tw$. We first consider the upper bounded case and then demonstrate the effect of separately chosen steplengths.

PROPOSITION 3.2. *The complementarity $x^Tz + s^Tw$ decreases linearly at each step of the primal-dual interior point method when a common step α is used in the primal and the dual.*

Proof.

$$\begin{aligned}
&(x+\alpha\Delta x)^T(z+\alpha\Delta z) + (s+\alpha\Delta s)^T(w+\alpha\Delta w)\\
&= x^Tz + s^Tw + \alpha(x^T\Delta z + s^T\Delta w + \Delta x^Tz + \Delta s^Tw)\\
&\quad +\alpha^2(\Delta x^T\Delta z + \Delta s^T\Delta w)\\
&= x^Tz + s^Tw - \alpha(x^Tz + s^Tw - 2n\mu) + \alpha^2(\Delta x^T\Delta z + \Delta s^T\Delta w).
\end{aligned}$$

Using (31) to substitute for Δz and applying Lemma 3.1 we have

$$\begin{aligned}&(x+\alpha\Delta x)^T(z+\alpha\Delta z)+(s+\alpha\Delta s)^T(w+\alpha\Delta w)\\&=x^Tz+s^Tw-\alpha(x^Tz+s^Tw-2n\mu).\qquad\square\end{aligned}$$

When separate primal and dual steplengths are allowed, two cases result. The complementarity is as follows.

If $(\alpha_p \le \alpha_d)$ we have

$$\begin{aligned}&(x+\alpha_p\Delta x)^T(z+\alpha_d\Delta z)+(s+\alpha_p\Delta s)^T(w+\alpha_d\Delta w)\\&\quad=x^Tz+s^Tw+\alpha_d(x^T\Delta z+s^T\Delta w)+\alpha_p(\Delta x^Tz+\Delta s^Tw)\\&\qquad+\alpha_d\alpha_p(\Delta x^T\Delta z+\Delta s^T\Delta w)\\&\quad=x^Tz+s^Tw-\alpha_p(x^Tz+s^Tw-2n\mu)+(\alpha_d-\alpha_p)(x^T\Delta z+s^T\Delta w)\\&\qquad+\alpha_p\alpha_d(\Delta x^T\Delta z+\Delta s^T\Delta w).\end{aligned}$$

Invoking Lemma 3.1 to eliminate the last term and observing that

$$\begin{aligned}x^T\Delta z+s^T\Delta w&=x^T(\Delta w-A^T\Delta y)+(u-x)^T\Delta w\\&=-b^T\Delta y+u^T\Delta w\end{aligned}$$

as a result of (31), we have

$$\begin{aligned}&(x+\alpha_p\Delta x)^T(z+\alpha_d\Delta z)+(s+\alpha_p\Delta s)^T(w+\alpha_d\Delta w)\\&\quad=x^Tz+s^Tw-\alpha_p(x^Tz+s^Tw-2n\mu)-(\alpha_d-\alpha_p)(b^T\Delta y-u^T\Delta w).\end{aligned}$$

Similarly, when $(\alpha_d < \alpha_p)$,

$$\begin{aligned}&(x+\alpha_p\Delta x)^T(z+\alpha_d\Delta z)+(s+\alpha_p\Delta s)^T(w+\alpha_d\Delta w)\\&\quad=x^Tz+s^Tw-\alpha_d(x^Tz+s^Tw-2n\mu)+(\alpha_p-\alpha_d)c^T\Delta x,\end{aligned}$$

because $\Delta x^Tz+\Delta s^Tw=c^T\Delta x$.

When $\alpha_d=\alpha_p$ the final term disappears in both cases, and the new complementarity is simply as stated in Proposition 3.2. If $\alpha_d\neq\alpha_p$, the added term is either the directional derivative with respect to the primal linear objective (1) or the negative directional derivative for its dual. When the primal objective is decreasing and the dual objective is increasing, this added term further reduces the complementarity. In practice, these terms usually, but do not necessarily, further decrease complementarity.

The effect of the predictor-corrector procedure on the complementarity can be analyzed in a similar manner.

PROPOSITION 3.3. *The complementarity x^Tz+s^Tw decreases linearly at each step of the MPC algorithm when the same step α is taken in the primal and the dual.*

Proof.

$$\begin{aligned}&(x+\alpha\Delta x^i)^T(z+\alpha\Delta z^i)+(s+\alpha\Delta s^i)^T(w+\alpha\Delta w^i)\\&=x^Tz+s^Tw+\alpha(x^T\Delta z^i+s^T\Delta w^i+z^T\Delta x^i+w^T\Delta s^i)+\alpha^2(\Delta x^{i^T}\Delta z^i+\Delta s^{i^T}\Delta w^i)\\&=x^Tz+s^Tw+\alpha(x^T\Delta z^i+s^T\Delta w^i+z^T\Delta x^i+w^T\Delta s^i)\ \text{(by Lemma 3.1)}\\&=x^Tz+s^Tw-\alpha[x^Tz+s^Tw+(\Delta x^{i-1^T}\Delta z^{i-1}+\Delta s^{i-1^T}\Delta w^{i-1})-2n\mu]\ \text{(by (11))}\\&=x^Tz+s^Tw-\alpha(x^Tz+s^Tw-2n\mu)\ \text{(by Lemma 3.1).}\qquad\square\end{aligned}$$

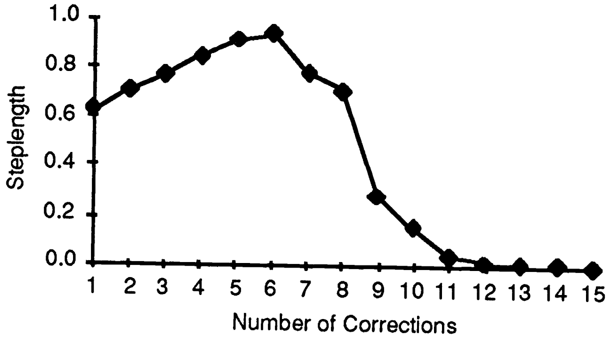


FIG. 1. *Allowable steplength at each correction.*

Again, allowing separately chosen primal and dual steplengths introduces a directional derivative term. When $(\alpha_p \le \alpha_d)$,

$$\begin{aligned}&(x+\alpha_p\Delta x^i)^T(z+\alpha_d\Delta z^i)+(s+\alpha_p\Delta s^i)^T(w+\alpha_d\Delta w^i)\\&=x^Tz+s^Tw-\alpha_p(x^Tz+s^Tw-2n\mu)-(\alpha_d-\alpha_p)(b^T\Delta y^i-u^T\Delta w^i),\end{aligned}$$

and when $(\alpha_d < \alpha_p)$,

$$\begin{aligned}&(x+\alpha_p\Delta x^i)^T(z+\alpha_d\Delta z^i)+(s+\alpha_p\Delta s^i)^T(w+\alpha_d\Delta w^i)\\&=x^Tz+s^Tw-\alpha_d(x^Tz+s^Tw-2n\mu)+(\alpha_p-\alpha_d)c^T\Delta x^i.\end{aligned}$$

Predicting and correcting can affect complementarity only through the steplengths, which are a function of the direction, and through the directional derivative term. (The directional derivative term should be small as the optimum is approached and when α_d and α_p are close.) Within MPC we test to insure that the same step is taken in the primal and dual spaces when taking separate steps does not further reduce complementarity. When the same primal and dual steplengths are chosen, predicting and correcting affects only the steplength. Multiple correcting from a feasible point with a common primal and dual steplength $\alpha = \min(\alpha_p, \alpha_d)$ is performed on the problem AFIRO with the progressive effect of predicting and correcting on the steplength presented in Fig. 1. It illustrates that predicting and correcting, even from a feasible point, will not monotonically increase steplength and thereby decrease complementarity. Thus, performing a fixed number of corrections at each iteration may not be productive.

At each *feasible* iteration of MPC, we choose the number of corrections dynamically based on the complementarity that would result if a step were taken. Let g^i be the complementarity that would result from taking a step in the direction obtained after i corrections. We consider performing an $(i+1)$st correction only if $g^i < g^{i-1}$ and i is less than some maximum number of corrections. If it is true that $g^i \ge g^{i-1}$, we stop correcting and use the direction Δv^{i-1}. That is, another correction is considered only if the previous one decreased the complementarity. Thus, a direction Δv^m requires $m+2$ backsolves when correcting terminates based on complementarity and $m+1$ backsolves when the maximum number of corrections allowed is m. Both the desire to avoid extra backsolves and the convergence results of [13] suggest that the maximum number of corrections should be small in practice.

3.2. The infeasible case. In the feasible case, complementarity provides a definitive measure of the value of a correction. When correcting from an infeasible point, however, this is no longer the case. Based on the definitions given in (30), note that

$$\begin{aligned} &A\Delta x^i = d_p, \\ &A^T\Delta y^i + \Delta z^i - \Delta w^i = d_d \quad \text{and} \quad \Delta x^i + \Delta s^i = d_u \end{aligned} \tag{32}$$

for any value of i. Hence, if $\alpha_p = 1$, then the new point will be primal feasible and if $\alpha_d = 1$, the new point will be dual feasible.

Determining m_k—the number of corrections to perform at iteration k—must integrate reducing complementarity with reducing infeasibility. We first consider the complementarity g^0 resulting from a step α in the affine direction

$$\begin{aligned} g^0 &= (x + \alpha\Delta x^0)^T(z + \alpha\Delta z^0) + (s + \alpha\Delta s^0)^T(w + \alpha\Delta w^0) \\ &= x^Tz + s^Tw - \alpha(x^Tz + s^Tw) + \alpha^2(\Delta x^{0^T}\Delta z^0 + \Delta s^{0^T}\Delta w^0) \\ &= x^Tz + s^Tw - \alpha(x^Tz + s^Tw) + \alpha^2(d_d^T\Delta x^0 - d_p^T\Delta y^0 + d_u^T\Delta w^0). \end{aligned}$$

Note that now the infeasibilities also affect the complementarity that would result from a step. This, in turn, affects the choice of μ which depends on g^0.

When Δv_c is the correction to the affine direction, we have $\Delta v = \Delta v^0 + \Delta v_c$. The complementarity that would result from a step α in the direction Δv is

$$\begin{aligned} &(x + \alpha(\Delta x^0 + \Delta x_c))^T(z + \alpha(\Delta z^0 + \Delta z_c)) \\ &\qquad + (s + \alpha(\Delta s^0 + \Delta s_c))^T(w + \alpha(\Delta w^0 + \Delta w_c)) \\ &\quad = x^Tz + s^Tw \\ &\qquad + \alpha[x^T(\Delta z^0 + \Delta z_c) + z^T(\Delta x^0 + \Delta x_c) + s^T(\Delta w^0 + \Delta w_c) \\ &\qquad\quad + w^T(\Delta s^0 + \Delta s_c)] \\ &\qquad + \alpha^2[(\Delta x^0 + \Delta x_c)^T(\Delta z^0 + \Delta z_c) + (\Delta s^0 + \Delta s_c)^T(\Delta w^0 + \Delta w_c)] \\ &\quad = x^Tz + s^Tw - \alpha(x^Tz + s^Tw - 2n\mu) - \alpha(d_d^T\Delta x^0 - d_p^T\Delta y^0 + d_u^T\Delta w^0) \\ &\qquad + \alpha^2[d_d^T(\Delta x^0 + \Delta x_c) - d_p^T(\Delta y^0 + \Delta y_c) + d_u^T(\Delta w^0 + \Delta w_c)]. \end{aligned} \tag{33}$$

While steplengths near 1.0 are *always* beneficial for reducing infeasibility and for reducing complementarity from a *feasible* point, this is not necessarily true when reducing complementarity from an *infeasible* point. Infeasibility influences complementarity explicitly through the additional terms in (33) and implicitly through μ. The combined effect may be to *increase* complementarity. Since we ultimately want to reduce both complementarity and infeasibility, we look at the combined infeasibility and complementarity for determining m_k. We define

$$G^i = g^i + (1 - \alpha_d)\|d_d\|_1 + (1 - \alpha_p)(\|d_p\|_1 + \|d_u\|_1)$$

and attempt an $(i + 1)$st correction only if i is less than the allowable maximum and $G^i < G^{i-1}$. Note that G^i measures the norm of $F(v)$.

Table 2 provides an example from the problem AGG where estimated complementarity increases from that of the previous iteration. To compensate, the allowable steplength gets smaller with each correction until virtually no step can be taken, and complementarity does not increase. In fact, when infeasibility is small relative to complementarity, using the combined measure G does not prevent this behavior. Thus,

TABLE 2
Correcting after an increase in complementarity.

	Total Comple.	Total Infeasibility	Steplengths	
			primal	dual
previous iteration	.2748 D+08	.1939 D+07		
1 - correction	.2928 D+08	.1849 D+07	.467 D-01	.215 D+00
2 - correction	.2756 D+08	.1939 D+07	.120 D-06	.549 D-03
3 - correction	.2749 D+08	.1939 D+07	.658 D-10	.346 D-07
4 - correction	.2755 D+08	.1939 D+07	.234 D-16	.123 D-12

within MPC we allow no further correcting if the steplengths at iteration i are both less than $\min(\alpha_p, \alpha_d)$ for the previous iteration.

It is interesting to note the effect of the initial point $(x^0, s^0, y^0, z^0, w^0)$ on *all* iterations of MPC. Let $d_p^k = b - Ax^k$, $d_d^k = c - A^T y^k - z^k + w^k$, and $d_u^k = u - x^k - s^k$. The relationship between iterations of these vectors is exhibited by the following proposition.

PROPOSITION 3.4. *On iteration k of* MPC, $d_p^k = \gamma_p^k d_p^0$, $d_u^k = \gamma_p^k d_u^0$, *and* $d_d^k = \gamma_d^k d_d^0$ *for some values of* γ_p^k *and* γ_d^k *satisfying* $0 \le \gamma_p^k \le 1$ *and* $0 \le \gamma_d^k \le 1$.

Proof. For any value of $k \ge 0$, let α_p^k and α_d^k be the primal and dual stepsizes, respectively, on iteration k. From (12) and (32), it follows that

$$\begin{aligned} d_p^{k+1} &= b - Ax^{k+1} = b - A(x^k + \alpha_p^k \Delta x^{m_k}) \\ &= (1 - \alpha_p^k) d_p^k, \\ d_u^{k+1} &= u - (x^{k+1} + s^{k+1}) = u - ((x^k + \alpha_p^k \Delta x^{m_k}) + (s^k + \alpha_p^k \Delta s^{m_k})) \\ &= (1 - \alpha_p^k) d_u^k, \end{aligned}$$

and

$$\begin{aligned} d_d^{k+1} &= c - A^T y^{k+1} - z^{k+1} + w^{k+1} \\ &= c - A^T(y^k + \alpha_d^k \Delta y^{m_k}) - (z^k y^k + \alpha_d^k \Delta z^{m_k}) + (w^k + \alpha_d^k \Delta w^{m_k}) \\ &= (1 - \alpha_d^k) d_d^k. \end{aligned}$$

By setting $\gamma_p^{k+1} = (1 - \alpha_p^k)\gamma_p^k$ and $\gamma_d^{k+1} = (1 - \alpha_d^k)\gamma_d^k$ for $k \ge 0$, with $\gamma_p^0 = 1 - \alpha_p^0$ and $\gamma_d^0 = 1 - \alpha_d^0$, the result clearly follows by induction on k. □

This proposition indicates that the performance of either the primal-dual algorithm or the predictor-corrector algorithm is heavily dependent on the choice of the starting point $(x^0, s^0, y^0, z^0, w^0)$ since the vectors d_p^0, d_d^0, and d_u^0 are affecting the calculations in any iteration where the current iterate is primal and/or dual infeasible.

4. Computational experiments with linear programs. Any savings derived from a multiple predictor-corrector procedure must arise from reducing the number of interior point method iterations. In this section, we examine the potential of several correcting strategies to reduce iterations. This is the first step in evaluating the viability of higher-order methods.

The multiple predictor-corrector procedure described in §§2 and 3 has been implemented within OB1 [5]. The maximum number of corrections is a prespecified parameter, while the *actual* number at any iteration is determined by the method described in §3. Numerical experiments were conducted on a representative subset of the NETLIB problems chosen by eliminating problems with dense columns and selecting every other problem from the resulting alphabetic list. The problems PILOT4, PILOTWE, and GREENBEA were omitted after testing because numerical difficulties prevented solution under several strategies.

Computational tests were performed on a Silicon Graphics 4D/70 workstation running SGI Unix V3.2 with FORTRAN code, compiled with the MIPS `f77` compiler, using the default optimization and `-Olimit` 1000. Testing was conducted with all default OB1 options except the steplength reduction factor (DARE) was set to the predictor-corrector suggested value of 0.99995.

Table 3 provides iteration counts for several correction strategies. The following are brief descriptions of the strategies tested.

- 1-correction. This is the *base case* in which exactly one correction is performed in each iteration as presented in [5].
- 3-correction. At least one and a maximum of three corrections are performed at each iteration.
- 10-correction. At least one and a maximum of ten corrections are performed at each iteration.
- 99-correction. At least one and a maximum of 99 corrections are performed at each iteration.
- feasible-99. Exactly one correction is performed at each infeasible iteration. Once feasible, at least one and at most 99 corrections are performed.
- dynamic mu-99. At least one and a maximum of 99 corrections are performed at each iteration with μ dynamically set after each correction.
- 0.5 heuristic. At least one and at most three corrections are performed at each iteration. An additional correction is tested only if $\alpha_p, \alpha_d \geq 0.5$.

The first four strategies examine the effect of changing the maximum allowable number of corrections. We examine limits of 3, 10, and an essentially "unbounded" case of 99. While the iterations do not decrease monotonically as the number of corrections increase, higher corrections do tend to yield lower iteration counts. There are, however, instances such as AGG where the 10 and 99 strategies perform more iterations. Although we consider 99 to be an unreasonably high bound on the number of corrections, there are several instances where 99 corrections are performed. A striking example is BEACONFD in which 99 corrections are performed in each of the five iterations.

Overall, the 99-correction strategy increases iterations on 2 problems, leaves 6 unchanged, and decreases iterations relative to the 1-correction strategy in the remaining 31 problems. The 10-correction strategy increases iterations in 4 cases, leaves 5 unchanged, and decreases the remaining 30. The 3-correction maximum increases iterations in 5 problems, causes no change in 8, and decreases 26. The percent change in iterations for the 99- and 3-correction strategies relative to the 1-correction procedure are displayed graphically in Fig. 2.

When corrections are performed only after feasibility is attained, there are no cases in which the number of iterations increases, 27 in which it remains the same, and 12 in which it decreases. With this strategy, exactly one correction is performed in each iteration until the relative primal and dual infeasibilities are less than 10^{-6}. In several problems, this requirement is satisfied just before optimality is attained, leaving few iterations in which to attempt higher-order correcting. Consequently, many of the iteration counts stay the same. For this strategy there are no increases in iterations. This apparently occurs as a result of two contributing factors. First, there is no longer a tradeoff between reducing infeasibility and complementarity, so the complementarity provides a definitive measure of the benefit from predicting and

TABLE 3
Comparing number of iterations for various strategies.

Problem	1-correction	3-correction	10-correction	99-correction	.5 heuristic	dyn mu -99	feasible - 99
25fv47	25	25	25	25	25	25	25
adlittle	12	10	9	9	10	11	12
agg	24	24	27	27	28	24	23
agg3	17	14	13	12	14	16	15
beaconfd	10	8	7	5	8	6	10
bnl1	27	24	24	24	24	25	27
boeing1	24	22	24	24	22	25	24
bore3d	18	18	18	18	17	17	18
capri	18	17	17	17	16	17	18
czprob	35	34	31	30	34	29	35
degen2	14	13	11	13	13	13*	14
e226	22	21	21	21	25	19	22
fffff800	28	29	28	28	28	29	28
fit1d	18	15	14	15	15	14	18
ganges	16	13	15	12	13	13	16
grow15	16	14	12	12	14	15	15
grow7	14	14	13	12	14	14	14
kb2	15	19	14	14	19	14	15
nesm	30	31	31	31	29	28	30
pilotnov	20	25	19	19	25	19	20
sc105	10	10	8	7	10	8	10
sc50a	10	9	8	6	9	7	10
scagr25	16	14	12	13	14	13	15
scfxm1	17	14	14	14	15	15	16
scfxm3	20	17	17	17	17	18	20
scrs8	27	24	22	22	25	25	26
scsd6	12	10	10	10	10	10	10
sctap1	15	14	13	12	15	14	15
sctap3	17	16	16	16	16	16	13
share1b	20	20	19	19	21	24	20
shell	21	18	18	18	19	21	20
ship04s	15	13	10	10	13	13	14
ship08l	16	13	15	14	13	16	16
ship12s	18	15	14	13	14	15	17
stair	16	14	18	13	14	14	14
standmps	24	24	24	24	24	24	24
stocfor2	22	22	22	22	22	20	22
vtpbase	13	11	11	11	11	12	13
woodw	20	22	22	14	24	N/A	20
	732	690	666	643	699	658 **	714

* Only seven digits of accuracy achieved; ** Based on incomplete data as noted.

correcting. Second, feasibility may occur close to optimality, so we are more likely to observe the local convergence properties of the composite Newton method.

An appealing variant of Mehrotra's dynamic strategy for choosing μ is to reset μ after each correction. That is, compute μ within the predictor-corrector loop by replacing Steps 2 and 3 of MPC with the following:

```
For $i = 1, \ldots, m_k$ do
    Compute $\mu^i(v^k, \Delta v^{i-1})$.
    Solve the following system for $\Delta v^i$
```

$$
\begin{pmatrix} A\Delta x^i \\ \Delta x^i + \Delta s^i \\ A^T\Delta y^i + \Delta z^i - \Delta w^i \\ Z\Delta x^i + X\Delta z^i \\ W\Delta s^i + S\Delta w^i \end{pmatrix} = \begin{pmatrix} b - Ax \\ u - x - s \\ c - A^T y - z + w \\ \mu^i e - XZe - \Delta X^{i-1}\Delta Z^{i-1}e \\ \mu^i e - SWe - \Delta S^{i-1}\Delta W^{i-1}e \end{pmatrix}. \tag{34}
$$

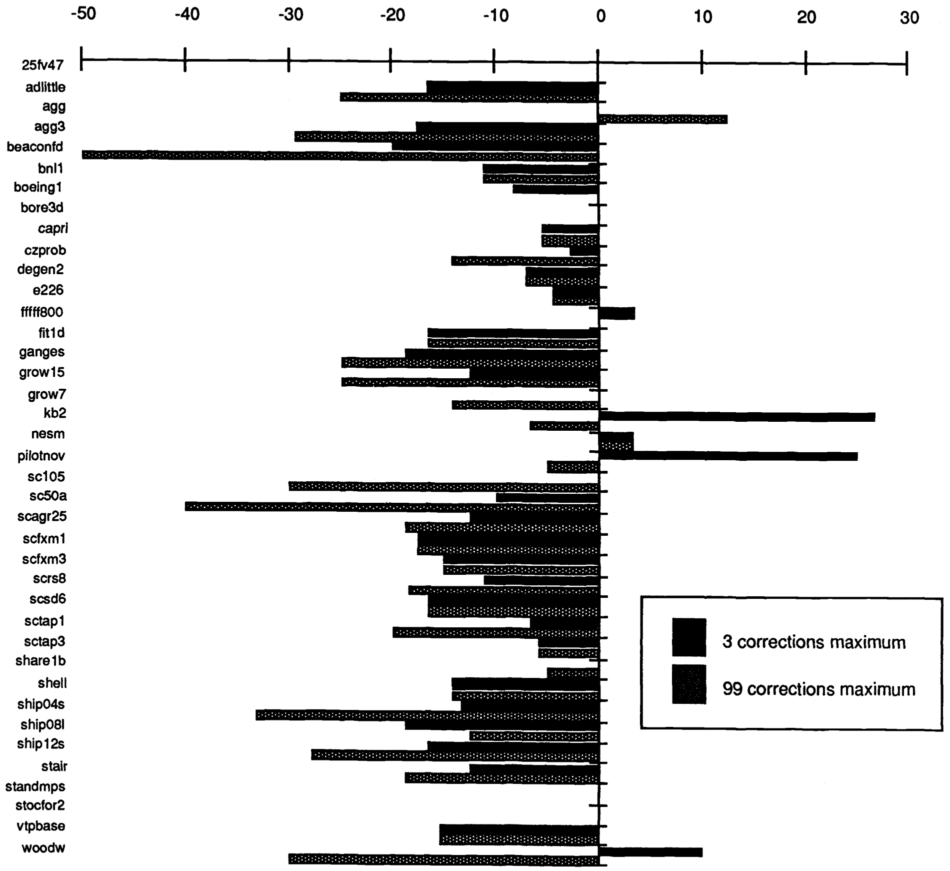


FIG. 2. *Percent change in iterations relative to one correction strategy.*

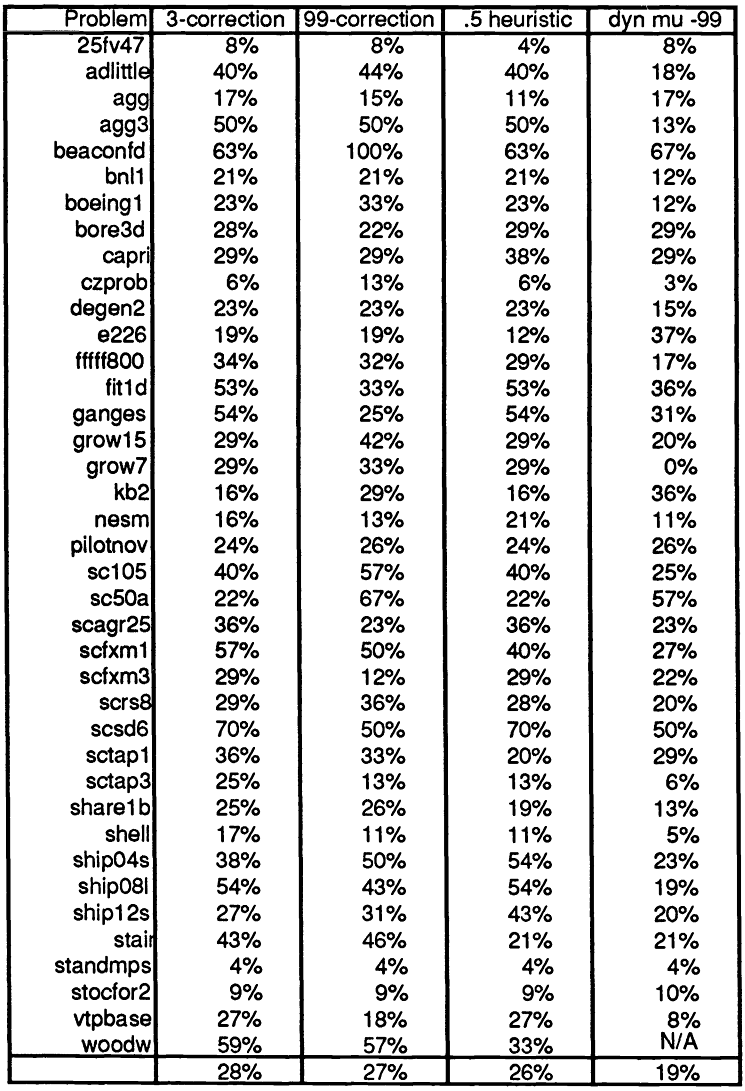
```
end
Define $\Delta v = \Delta v^{m_k}$.
```

If complementarity is decreasing with correcting, the value of μ used to compute the subsequent corrected direction will be smaller. The iterations that result from using the dynamic strategy increase in 3 problems, remain the same in 6, and decrease in 29. WOODW could not be solved with this strategy. The results of Table 3 indicate that this method for choosing μ is generally inferior to the method described earlier. It is for this reason that we employ the "fixed μ" strategy in our basic algorithm.

The composite Newton interior point method can also be adapted to reset μ in the inner iterations. The corresponding substitution in CNM yields the composite Newton method precisely as stated in [13]. In this case we replace Steps 2 and 3 of CNM with the following.

```
For $i = 1, \ldots, m_k$ do
    Compute $\mu^i(v^k, \Delta v^{i-1})$.
    Solve $F'(v^k)\Delta\hat{v}^i = -F(v^k + \sum_{j=0}^{i-1} \Delta\hat{v}^j) + \mu^i\hat{e}$ for $\Delta\hat{v}^i$
end
```

TABLE 4
Percent of iterations in which multiple corrections are performed.



Problem	3-correction	99-correction	.5 heuristic	dyn mu -99
25fv47	8%	8%	4%	8%
adlittle	40%	44%	40%	18%
agg	17%	15%	11%	17%
agg3	50%	50%	50%	13%
beaconfd	63%	100%	63%	67%
bnl1	21%	21%	21%	12%
boeing1	23%	33%	23%	12%
bore3d	28%	22%	29%	29%
capri	29%	29%	38%	29%
czprob	6%	13%	6%	3%
degen2	23%	23%	23%	15%
e226	19%	19%	12%	37%
fffff800	34%	32%	29%	17%
fit1d	53%	33%	53%	36%
ganges	54%	25%	54%	31%
grow15	29%	42%	29%	20%
grow7	29%	33%	29%	0%
kb2	16%	29%	16%	36%
nesm	16%	13%	21%	11%
pilotnov	24%	26%	24%	26%
sc105	40%	57%	40%	25%
sc50a	22%	67%	22%	57%
scagr25	36%	23%	36%	23%
scfxm1	57%	50%	40%	27%
scfxm3	29%	12%	29%	22%
scrs8	29%	36%	28%	20%
scsd6	70%	50%	70%	50%
sctap1	36%	33%	20%	29%
sctap3	25%	13%	13%	6%
share1b	25%	26%	19%	13%
shell	17%	11%	11%	5%
ship04s	38%	50%	54%	23%
ship08l	54%	43%	54%	19%
ship12s	27%	31%	43%	20%
stair	43%	46%	21%	21%
standmps	4%	4%	4%	4%
stocfor2	9%	9%	9%	10%
vtpbase	27%	18%	27%	8%
woodw	59%	57%	33%	N/A
	28%	27%	26%	19%

Even with the additional generality in these statements of MPC and CNM, a result analagous to that of Theorem 1 guarantees that the more general MPC algorithm is still a composite Newton method as stated above.

Reducing the number of iterations reduces the number of matrix factorizations, but MPC does this at the cost of performing more backsolves per iteration. A practical implementation must limit the number of backsolves. One way to do this is to keep the maximum possible corrections small. Thus, it is impractical to allow 99—or even 10—corrections in a realistic implementation. Another approach is to try to anticipate when an additional correction will be advantageous. Table 4 illustrates that multiple corrections are performed in a relatively small percent of the total iterations, so an

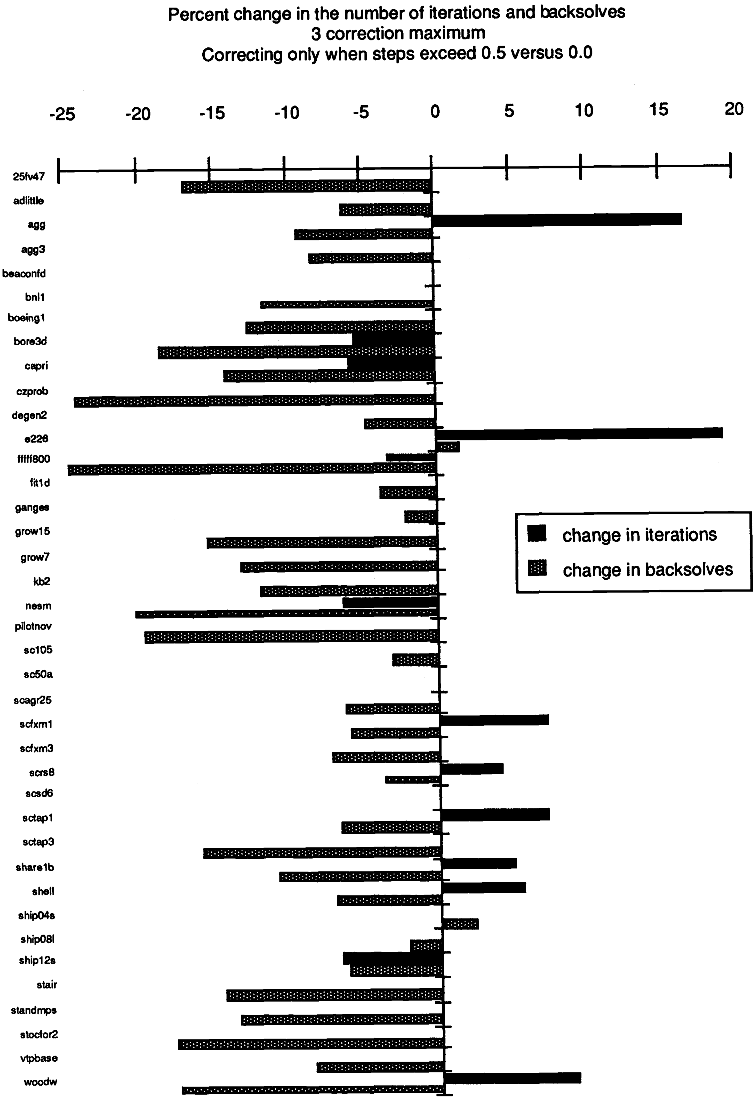


FIG. 3. *Comparison of* 0.5 *heuristic with* 3*-correction strategy.*

extra backsolve per iteration could often be saved if we could determine when to attempt an extra correction. Our final correction strategy uses a heuristic rule for determining when to attempt another correction (up to the maximum). This method is based on observing the behavior of the previous correction strategies. Each of the former strategies *always* attempts another correction if the maximum has not been reached. We observed that this correction often yields an improvement (relative to the measures described in §3) only when the allowable steplengths in the previous direction were relatively long. In the 0.5 *heuristic*, primal and dual steplengths must be greater than 0.5 to consider another correction. That is, we attempt correction $i+1$ only if $\alpha_p, \alpha_d \geq 0.5$ in the direction Δv^i.

Figure 3 displays a graphic comparison of the number of iterations and the number of backsolves for this approach with a maximum of three corrections per iteration versus the 3-*correction* strategy described previously. Relative to the 3-*correction* method, 8 problems increase iterations, 5 decrease, and 26 remain the same. The advantage of this strategy, however, is seen in the number of backsolves also presented in Table 5. In 2 cases the number of backsolves increases with the increased iterations, in 3 it stays the same, but it decreases—sometimes dramatically—in each of the remaining 34 test problems. With respect to the 1-*correction* method, the number of backsolves always increases, while the number of iterations increases for 6 cases, remains the same for 7, and decreases in 26.

The tradeoff between reducing iterations and increasing the number of backsolves is the critical issue that will determine the viability of higher-order predictor-corrector methods. In this section, we have presented results demonstrating that iterations can often be reduced by judiciously performing higher-order corrections. Unfortunately, this is always at the cost of performing extra backsolves, as seen in Table 5. Thus, higher-order methods may be promising at least in cases where performing large dense factorizations is expensive relative to the cost of performing backsolves. Examining this tradeoff in efficient implementation remains to be addressed in future research. In view of Proposition 3.4, the starting point may affect the performance of *any* of the MPC procedures. Further analysis of these effects remains for future research.

5. Predicting and correcting with quadratic objectives.

As suggested in Mehrotra [8], the predictor-corrector method is easily extended to include convex quadratic objectives. In this section, we derive the extension of Mehrotra's predictor-corrector procedure to quadratic objectives and then present a multiple predictor-corrector variant.

The quadratic programming problem in standard form is

$$\begin{aligned} &\min\ c^T x + \tfrac{1}{2}\, x^T Q x \\ &\text{subject to } Ax = b, \\ &\qquad x + s = u, \\ &\qquad x,\ s \geq 0, \end{aligned} \tag{35}$$

where $A \in \Re^{m\times n}$, $b \in \Re^m$, $c \in \Re^n$, $x \in \Re^n$, $u \in \Re^n$, $s \in \Re^n$, and $Q \in \Re^{n\times n}$ are positive semidefinite. The related barrier transformed problem is

$$\begin{aligned} &\min\ c^T x + \frac{1}{2}\, x^T Q x - \mu \sum_{j=1}^{n} \ln x_j - \mu \sum_{j=1}^{n} \ln s_j \\ &\text{subject to } Ax = b, \\ &\qquad x + s = u. \end{aligned} \tag{36}$$

The first-order conditions for (36) are now

$$\begin{pmatrix} Ax - b \\ x + s - u \\ A^T y + z - w - Qx - c \\ XZe \\ SWe \end{pmatrix} = 0, \quad \text{and} \quad x, s, z, w \geq 0, \tag{37}$$

TABLE 5
Number of backsolves.

Problem	3-correction	.5 heuristic	1-correction
25fv47	77	64	50
adlittle	32	30	24
agg	75	68	48
agg3	48	44	34
beaconfd	29	29	20
bnl1	77	68	54
boeing1	71	62	48
bore3d	59	48	36
capri	56	48	36
czprob	103	78	70
degen2	41	39	28
e226	67	68	44
fffff800	97	73	56
fit1d	52	50	36
ganges	46	45	32
grow15	45	38	32
grow7	45	39	28
kb2	58	51	30
nesm	98	78	60
pilotnov	81	65	40
sc105	32	31	20
sc50a	25	25	20
scagr25	47	44	32
scfxm1	50	47	34
scfxm3	55	51	40
scrs8	79	76	54
scsd6	36	36	24
sctap1	45	42	30
sctap3	50	42	34
share1b	64	57	40
shell	57	53	42
ship04s	43	44	30
ship08l	46	45	32
ship12s	48	45	36
stair	48	41	32
standmps	73	63	48
stocfor2	67	55	44
vtpbase	35	32	26
woodw	79	65	40
	2236	1979	1464

and similarly the first-order conditions for the barrier transformed problem (37) are

$$\begin{pmatrix} Ax - b \\ x + s - u \\ A^T y + z - w - Qx - c \\ XZe - \mu e \\ SWe - \mu e \end{pmatrix} = 0. \tag{38}$$

Whereas the primal-dual method described in Monteiro and Adler [10] and Carpenter et al. [1] applies Newton's method directly to (38), the predictor-corrector procedure first obtains the *predictor* or *affine* direction by applying Newton's method to (37).

This entails solving the system

$$(39)\qquad \begin{pmatrix} A\Delta x^0 \\ \Delta x^0 + \Delta s^0 \\ A^T\Delta y^0 + \Delta z^0 - \Delta w^0 - Q\Delta x^0 \\ Z\Delta x^0 + X\Delta z^0 \\ W\Delta s^0 + S\Delta w^0 \end{pmatrix} = \begin{pmatrix} b - Ax \\ u - x - s \\ c - A^Ty - z + w + Qx \\ -XZe \\ -SWe \end{pmatrix}$$

for the direction Δv^0. Having obtained the predictor direction, we can again obtain the centered correcting direction by solving the system

$$(40)\qquad \begin{aligned} A\Delta x_c &= 0, \\ \Delta x_c + \Delta s_c &= 0, \\ A^T\Delta y_c + \Delta z_c - \Delta w_c - Q\Delta x_c &= 0, \\ X\Delta z_c + Z\Delta x_c &= \mu e - \Delta X^0\Delta Z^0 e, \\ S\Delta w_c + W\Delta s_c &= \mu e - \Delta S^0\Delta W^0 e \end{aligned}$$

for Δx_c, Δs_c, Δy_c, and Δz_c, Δw_c. Again, the *full* predictor-corrector direction is then $\Delta v = \Delta v^0 + \Delta v_c$. The barrier parameter μ is computed precisely as described in §3 using equations (26)–(28). Since the complementarity is defined by $x^Tz + s^Tw$ for both linear and quadratic programs, the extension of the predictor-corrector method to quadratic objectives is straightforward.

Given that the current estimate to the optimal point is primal and dual feasible, the directions Δv^0 and Δv_c are *feasible directions*; a feasible direction Δv satisfies

$$(41)\qquad \begin{aligned} A\Delta x &= 0, \\ \Delta x + \Delta s &= 0, \\ A^T\Delta y + \Delta z - \Delta w - Q\Delta x &= 0. \end{aligned}$$

LEMMA 5.1. *For any* $(\Delta x, \Delta s, \Delta y, \Delta z, \Delta w)$ *that satisfy* (42), $\Delta x^T\Delta z + \Delta s^T\Delta w = \Delta x^TQ\Delta x$.

Proof. Using (42) to substitute $\Delta z = Q\Delta x + \Delta w - A^T\Delta y$ and $\Delta x = -\Delta s$, we have

$$\begin{aligned} \Delta x^T\Delta z + \Delta s^T\Delta w &= \Delta x^T(Q\Delta x + \Delta w - A^T\Delta y) + \Delta s^T\Delta w \\ &= \Delta x^TQ\Delta x - \Delta s^T\Delta w - (A\Delta x)^T\Delta y + \Delta s^T\Delta w \\ &= \Delta x^TQ\Delta x. \qquad \square \end{aligned}$$

In the linear case, the complementarity reduction from a feasible point (assuming the same steplengths are taken in the primal and dual) for the primal-dual and predictor-corrector methods differ only as a result of the allowable steplength for the same μ. In the quadratic case, the predictor-corrector procedure offers an extra advantage. In the primal-dual method the complementarity is changed as follows:

$$\begin{aligned} &(x + \alpha\Delta x)^T(z + \alpha\Delta z) + (s + \alpha\Delta s)^T(w + \alpha\Delta w) \\ &\quad = x^Tz + s^Tw + \alpha(x^T\Delta z + s^T\Delta w + \Delta x^Tz + \Delta s^Tw) + \alpha^2(\Delta x^T\Delta z + \Delta s^T\Delta w) \\ &\quad = x^Tz + s^Tw - \alpha(x^Tz + s^Tw - 2n\mu) + \alpha^2(\Delta x^T\Delta z + \Delta s^T\Delta w) \\ &\quad = x^Tz + s^Tw - \alpha(x^Tz + s^Tw - 2n\mu) + \alpha^2(\Delta x^TQ\Delta x) \quad \text{(by Lemma 5.1).} \end{aligned}$$

By convexity we have that the second-order term is nonnegative. In the predictor-corrector method the change in complementarity is the following:

$$\begin{aligned}
&(x+\alpha(\Delta x^0+\Delta x_c))^T(z+\alpha(\Delta z^0+\Delta z_c))\\
&\qquad +(s+\alpha(\Delta s^0+\Delta s_c))^T(w+\alpha(\Delta w^0+\Delta w_c))\\
&= x^Tz+s^Tw\\
&\qquad +\alpha(x^T\Delta z^0+z^T\Delta x^0+x^T\Delta z_c+z^T\Delta x_c+s^T\Delta w^0+w^T\Delta s^0+s^T\Delta w_c+w^T\Delta s_c)\\
&\qquad +\alpha^2((\Delta x^0+\Delta x_c)^T(\Delta z^0+\Delta z_c)+(\Delta s^0+\Delta s_c)^T(\Delta w^0+\Delta w_c))\\
&= x^Tz+s^Tw-\alpha(x^Tz+s^Tw)-\alpha(\Delta x^{0^T}\Delta z^0+\Delta s^{0^T}\Delta w^0-2n\mu)\\
&\qquad +\alpha^2((\Delta x^0+\Delta x_c)^T(\Delta z^0+\Delta z_c)+(\Delta s^0+\Delta s_c)^T(\Delta w^0+\Delta w_c))\\
&= x^Tz+s^Tw-\alpha(x^Tz+s^Tw+\Delta x^{0^T}Q\Delta x^0-2n\mu)\\
&\qquad +\alpha^2((\Delta x^0+\Delta x_c)^T(\Delta z^0+\Delta z_c)+(\Delta s^0+\Delta s_c)^T(\Delta w^0+\Delta w_c)),
\end{aligned}$$

by Lemma 5.1. Since $(\Delta v^0+\Delta v_c)$ is itself a feasible direction, we invoke Lemma 5.1 to obtain

$$\begin{aligned}
&(x+\alpha(\Delta x^0+\Delta x_c))^T(z+\alpha(\Delta z^0+\Delta z_c))\\
&\quad +(s+\alpha(\Delta s^0+\Delta s_c))^T(w+\alpha(\Delta w^0+\Delta w_c))\\
&\quad = x^Tz+s^Tw-\alpha(x^Tz+s^Tw+\Delta x^{0^T}Q\Delta x^0-2n\mu)\\
&\qquad +\alpha^2(\Delta x^0+\Delta x_c)^TQ(\Delta x^0+\Delta x_c).
\end{aligned}$$

Although there is still a second-order term increasing the complementarity, there is now a new first-order term reducing the complementarity. Thus, at least for the case of small enough steplength, the predictor-corrector method offers better complementarity reduction than the primal-dual method.

As in the linear case, instead of solving (41) for the combined centering and correction direction and adding this to the affine direction to obtain the full direction, we can solve directly for the full direction by solving

$$(42)\qquad \begin{pmatrix} A\Delta x\\ \Delta x+\Delta s\\ A^T\Delta y+\Delta z-\Delta w-Q\Delta x\\ X\Delta z+Z\Delta x\\ S\Delta w+W\Delta s\end{pmatrix}=\begin{pmatrix} b-Ax\\ u-x-s\\ c+Qx-A^Ty-z+w\\ \mu e-XZe-\Delta X^0\Delta Z^0e\\ \mu e-SWe-\Delta S^0\Delta W^0e\end{pmatrix}.$$

The system (42) can be solved repetitively for updated correction terms yielding the multiple predictor-corrector procedure for quadratic objectives.

ALGORITHM QMPC (quadratic multiple predictor-corrector).
Given $v^k=(x^k,s^k,y^k,z^k,w^k)$ with $x^k,s^k,z^k,w^k>0$.

```
    Step 1: Solve (5) for the affine direction $\Delta v^0$.
    Step 2: Compute $\mu(v^k,\Delta v^0)$ using (26)–(28).
    Step 3:
        For $i=1,\dots,m_k$ do
            Solve the following system for $\Delta v^i$
```

$$(43)\qquad \begin{pmatrix} A\Delta x^i\\ \Delta x^i+\Delta s^i\\ A^T\Delta y^i+\Delta z^i-\Delta w^i-Q\Delta x^i\\ Z\Delta x^i+X\Delta z^i\\ W\Delta s^i+S\Delta w^i\end{pmatrix}=\begin{pmatrix} b-Ax\\ u-x-s\\ c+Qx-A^Ty-z+w\\ \mu e-XZe-\Delta X^{i-1}\Delta Z^{i-1}e\\ \mu e-SWe-\Delta S^{i-1}\Delta W^{i-1}e\end{pmatrix}.$$

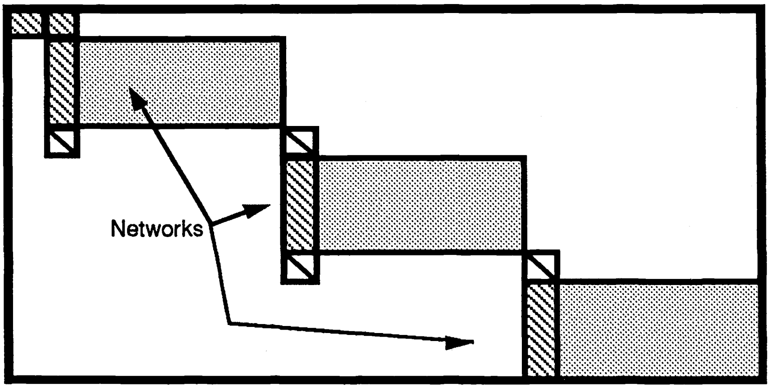


FIG. 4. *Staircase representation for stochastic network models.*

```
    end do
    Define $\Delta v = \Delta v^{m_k}$.
Step 4: Perform ratio test to determine allowable steplengths $\alpha_p$ and $\alpha_d$.
    Set $\alpha = \min(\alpha_p, \alpha_d)$.
Step 5: Move to the new point $v^{k+1}$ defined by
```

$$\begin{aligned} x^{k+1} &= x^k + \alpha\Delta x, \\ s^{k+1} &= s^k + \alpha\Delta s, \\ y^{k+1} &= y^k + \alpha\Delta y, \\ z^{k+1} &= z^k + \alpha\Delta z, \\ w^{k+1} &= w^k + \alpha\Delta w. \end{aligned} \tag{44}$$

This procedure is a simple extension of MPC for nonzero Q with the main difference being in the choice of α. Since x now appears in the dual constraints, the same step is taken in the primal and dual to avoid increasing dual infeasibility.

As in the linear case, the quadratic multiple predictor-corrector procedure is a level-m composite Newton interior point method. Since complementarity is given by $x^Tz + s^Tw$ in both the linear and quadratic cases, the proof of this is precisely as presented in §2.

6. Computational results for quadratic objectives.

In this section, we briefly present computational results demonstrating the potential of the predictor-corrector method for savings in both time and iterations. While the derivations of the previous section are valid for both separable and nonseparable quadratic problems, we solve only separable problems. To date, there is no efficient primal-dual interior point code available for solving nonseparable problems.

Our test problems are derived from financial stochastic network models developed by Mulvey and Vladimirou [12]. These models possess a staircase structure with network blocks along the diagonal. They have only a few variables in the objective—all of which appear nonlinearly. Generally, the objective is a nonlinear utility function, but for the purpose of experimentation, we have replaced the true nonlinear objective with an artificial quadratic objective. Any variable x_j that appears nonlinearly in the true objective is included here with $c_j = -1$ and $q_{jj} = 1$. The basic structure of

TABLE 6
Test problem description.

Name	# Scenarios	# Nonlinears	# Rows	# Columns
DETER0	18	18	2178	5723
DETER4	70	70	4270	10168
DETER8	36	36	4356	11430
DETER6	40	40	4840	12698
DETER5	48	48	5808	15234
DETER1	52	52	6292	16502
DETER2	80	80	7280	18498
DETER7	60	60	7260	19038
DETER3	72	72	8712	22842

TABLE 7
Total time in seconds.

	primal-dual	1 - correction
DETER0	133.29	120.59
DETER4	186.36	186.68
DETER8	282.94	280.71
DETER6	347.06	326.09
DETER5	465.89	496.27
DETER1	495.38	428.35
DETER2	518.45	509.12
DETER7	577.52	521.93
DETER3	757.61	698.00
	3764.50	3567.74

the models is depicted in Fig. 4 with a full description given by Lustig, Mulvey, and Carpenter [6]. Problem sizes are provided in Table 6.

The computing environment is as described in §4. We have implemented the predictor-corrector procedures within the framework of OBN—the separable nonlinear extension of OB1 described in [1]. The testing was performed under default OB1 settings except that the optimality tolerance was reduced to 10^{-7}.

Solution times with both primal-dual and predictor-corrector procedures are presented in Table 7. Table 8 provides iteration counts for the primal-dual method and several strategies for predicting and correcting. As in the linear case, the predictor-corrector method with one correction at each iteration offers consistent savings in both iterations and time. Iterations are reduced for all problems, while total time is reduced for seven of the nine problems. Because our test problems have extremely sparse constraint matrices which yield sparse Cholesky factorizations, these timings may understate the benefit of Mehrotra's procedure. For these problems, the matrix factorization is relatively inexpensive. We expect that when dense Cholesky factorizations are present the savings in computation time will be more dramatic. A graphic comparison of the predictor-corrector method relative to the primal-dual procedure is provided in Fig. 5.

The potential for reduction in iterations by predicting and correcting is highlighted in Fig. 6, which displays the percent change in iterations for higher-order strategies relative to one correction. Both the 3- and 99-correction strategies significantly reduce the number of iterations on all problems. Once again, the quadratic case

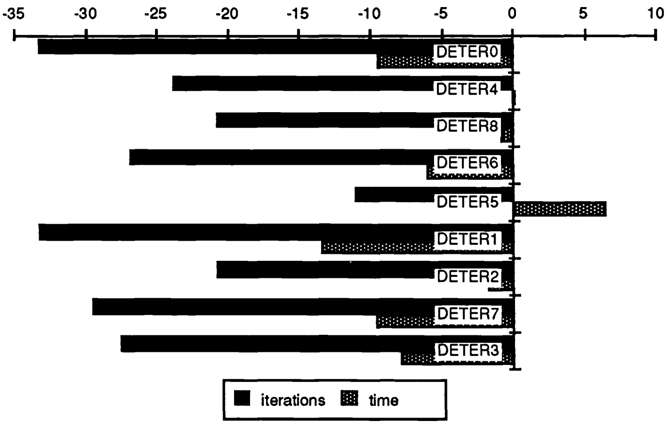


FIG. 5. *Comparison of predictor-corrector to primal-dual.*

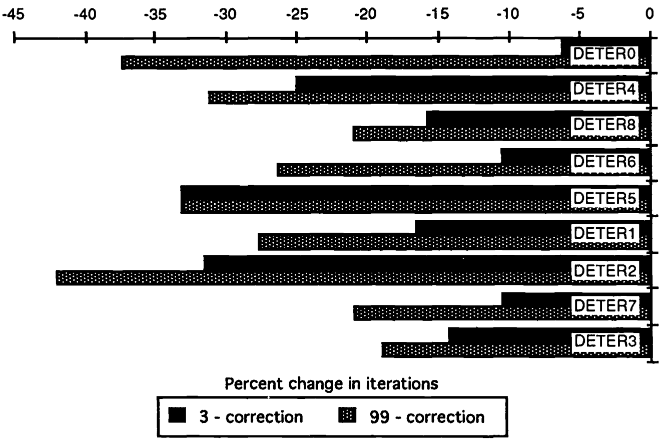


FIG. 6. *Comparison of multiple correction strategies to one correction.*

TABLE 8
Iterations to solution.

		predictor-corrector		
	primal-dual	1 - correction	3 - correction	99 - correction
DETER0	24	16	15	10
DETER4	21	16	12	11
DETER8	24	19	16	15
DETER6	26	19	17	14
DETER5	27	24	16	16
DETER1	27	18	15	13
DETER2	24	19	13	11
DETER7	27	19	17	15
DETER3	29	21	18	17
	229	171	139	122

demonstrates that there is clearly a potential for reducing iterations via higher-order correction strategies, but the computational efficacy remains to be demonstrated.

7. Conclusions and directions for further research. The computational results of §§4 and 6 demonstrate the potential for higher-order predictor-corrector methods to reduce iterations. What remains to be shown, however, is whether or not reducing the number of iterations can offset the extra work per iteration in an efficient implementation. We expect that in all but very large dense systems, correcting once at every iteration will be the preferred strategy.

There are several avenues for improving the multiple predictor-corrector procedure. The most notable is to determine a priori whether or not another correction will be beneficial. This will avoid two extra backsolves and will eliminate the need to store previous correction directions. Our .5 *heuristic* was a simple means to address this issue. With it we were able to reduce the number of backsolves but we still needed data structures to store previous correction directions.

It may also be advantageous to consider zero corrections as an option. Particularly when correcting from an infeasible point, there may be times when taking a primal-dual step is preferable to taking a correcting step. To do this with Mehrotra's method for choosing μ, however, requires an extra backsolve per iteration. That is, an iteration of MPC would consist of solving for the affine direction, solving for the primal-dual direction, and then comparing the first corrected direction with the primal-dual before proceeding with higher-order corrections. One alternative to the extra backsolve per iteration is to compute a centering direction and then use it for perhaps several iterations.

While we have demonstrated the potential of higher-order predictor-corrector methods for reducing iterations, its viability as a generally applicable procedure remains to be shown. A better understanding of when correcting is advantageous and the interplay of correction and infeasibility is required. In addition, the starting point may have impact in both of these issues.

REFERENCES

[1] T. J. Carpenter, I. J. Lustig, J. M. Mulvey, and D. F. Shanno, *A primal-dual interior point method for convex separable nonlinear programs*, Tech. Rep. SOR-90-2, Dept. of Civil Engineering and Operations Research, Princeton Univ., Princeton, NJ, May 1990; ORSA J. Comput., to appear.

[2] D. M. Gay, *Electronic mail distribution of linear programming test problems*, Mathematical Programming Society COAL Newsletter, 1988.

[3] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior point algorithm for linear programming*, in Progress in Mathematical Programming: Interior Point and Related Methods, N. Megiddo, ed., Springer-Verlag, New York, 1989, pp. 29–47.

[4] I. J. Lustig, R. E. Marsten, and D. F. Shanno, *Computational experience with a primal–dual interior point method for linear programming*, Linear Algebra Appl., 152 (1991), pp. 191–222.

[5] ———, *On implementing Mehrotra's predictor–corrector interior-point method for linear programming*, SIAM J. Optim., 2 (1992), pp. 435–449.

[6] I. J. Lustig, J. M. Mulvey, and T. J. Carpenter, *The formulation of stochastic programs for interior point methods*, Oper. Res., 39 (1991), pp. 757–770.

[7] S. Mehrotra, *On finding a vertex solution using interior point methods*, Linear Algebra Appl., 152 (1991), pp. 233–253.

[8] ———, *On the implementation of a primal-dual interior point method*, SIAM J. Optim., 2 (1992), pp. 575–601.

[9] S. Mizuno, M. J. Todd, and Y. Ye, *Anticipated behavior of path-following algorithms for linear programming*, Tech. Rep. 878, School of Operations Research and Industrial Engineering, Cornell Univ., Ithaca, NY, 1989.

[10] R. D. C. Monteiro and I. Adler, *Interior path following primal-dual algorithms: Part* II: *Convex quadratic programming*, Math. Programming, 44 (1989), pp. 43–66.

[11] R. D. C. Monteiro, I. Adler, and M. G. C. Resende, *A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension*, Math. Oper. Res., 15 (1990), pp. 191–214.

[12] J. M. Mulvey and H. Vladimirou, *Stochastic network optimization models for investment planning*, in Annals of Operations Research: Network Optimization and Applications, Vol. 20, B. Shetty, ed., J. C. Baltzer AG, Switzerland, 1989, pp. 187–217.

[13] R. A. Tapia, Y. Zhang, M. Saltzman, and A. Weiser, *The predictor-corrector interior-point method as a composite Newton method*, Tech. Rep. TR 90–17, Dept. of Mathematical Sciences, Rice Univ., Houston, TX, 1990.

A NOTE ON K-BEST SOLUTIONS TO THE CHINESE POSTMAN PROBLEM*

YASUFUMI SARUWATARI[†] AND TOMOMI MATSUI[‡]

Abstract. The K-*best problems* on combinatorial optimization problems, in which K-best solutions are considered instead of an optimal solution under the same conditions, have been widely studied. In this paper, the K-best problem on the famous Chinese postman problem is considered and an algorithm that finds K-best solutions is developed. The time complexity of the algorithm is $O(S(n,m) + K(n + m + \log K + nT(n + m, m)))$ where $S(s,t)$ denotes the time complexity of an algorithm for ordinary Chinese postman problems and $T(s,t)$ denotes the time complexity of a post-optimal algorithm for non-bipartite matching problems defined on a graph with s vertices and t edges.

Key words. combinatorial optimization, Chinese postman problem, K-best problem, T-join problem, matching theory, graph theory

AMS subject classifications. 05C38, 05C45

1. Introduction.

The Chinese postman problem was first proposed by Mei-ko Kwan in [8]. The problem is interpreted as follows (see [10]). The postman delivers mail along a set of streets and he must traverse each street at least once, in either direction. He starts at the post office and must return to this starting point. The Chinese postman problem finds a tour which enables the postman to walk the shortest possible distance.

It is well known that the above problem is reformulated as follows. Let G be a graph whose edges correspond to the streets in the city. A (nonnegative) length (we call a weight) of the street is associated with each edge. If an Eulerian graph arises from G by parallelizing some edges, then an Eulerian cycle of this graph yields a postman's tour of the original. Thus, the problem finds an Eulerian graph of minimum total weight which is obtained by replacing some edges in the graph G by a set of parallel ones.

Now we give a formal description of the problem. Let $G = (V, E)$ be an undirected connected graph without loops or parallel edges. Denote by $\boldsymbol{w} \in \mathcal{Q}_+^E$ a nonnegative weight function, where $\mathcal{Q}_+$ is the set of nonnegative rational numbers. Then the Chinese postman problem is formulated as:

$$\begin{array}{lll} \text{minimize} & \boldsymbol{wx} = \sum_{e\in E} \boldsymbol{w}(e)\boldsymbol{x}(e), & \\ \text{subject to} & \boldsymbol{x}(e) \geq 1, & \forall e \in E, \\ & \sum_{e\in\delta(v)} \boldsymbol{x}(e) \text{ is even}, & \forall v \in V, \\ & \boldsymbol{x} \in \mathcal{Z}_+^E, & \end{array}$$

* Received by the editors March 17, 1992; accepted for publication (in revised form) July 30, 1992.

[†] Department of Management Science, Science University of Tokyo, 1-3 Kagurazaka, Shinjuku-ku, Tokyo 162, Japan. Present address, Department of Social Sciences, National Defense Academy, Hashirimizu, Yokosuka, Kanagawa 239, Japan (saru@ms.kagu.sut.ac.jp or saruwata@gssm.otsuka.tsukuba.ac.jp).

[‡] Department of Industrial Administration, Science University of Tokyo, 2641 Yamazaki, Noda-shi, Chiba 278, Japan. Present address, Department of Mathematical Engineering and Information Physics, University of Tokyo, Bunkyo-ku, Tokyo 113, Japan (tomomi@misojiro.t.u-tokyo.ac.jp).

where $\mathcal{Z}_+$ denotes the set of nonnegative integer numbers and $\delta(v)$ denotes the set of edges incident with the vertex v. The variable $\boldsymbol{x}(e)$ denotes the number of times the edge e is traversed in the postman's tour. The Chinese postman problem is a well-solved problem [8] and Edmonds actually presented a polynomial time algorithm by transforming the problem into a non-bipartite matching problem [5], [6].

Here we consider the K-best Chinese postman problem, which finds K distinct feasible solutions $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^K$ such that $\boldsymbol{w}\boldsymbol{x}^1 \leq \boldsymbol{w}\boldsymbol{x}^2 \leq \cdots \leq \boldsymbol{w}\boldsymbol{x}^K \leq \boldsymbol{w}\boldsymbol{x}'$ for any feasible solution $\boldsymbol{x}' \neq \boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^K$. In this paper, the solutions $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^K$ are called *K-best solutions.* The K-best problem was first introduced by Murty, and he developed an algorithm for finding K-best solutions of the assignment problem [12]. In 1972, Lawler generalized Murty's algorithm for finding K-best solutions of general 0-1 integer problems [9]. However, it is hard to extend Lawler's algorithm for the Chinese postman problem, since the variables are not 0-1 valued in this problem.

In §2, we introduce some properties of a solution for the 2-best Chinese postman problem instead of considering K-best solutions directly. In §3, we develop a polynomial time algorithm for the 2-best Chinese postman problem. In §4, we construct an algorithm for K-best Chinese postman problems as an extension, which finds K-best solutions by solving 2-best Chinese postman problems iteratively.

2. Properties of a solution of the 2-best Chinese postman problem.

In this section, we first show a property of K-best solutions of the Chinese postman problem. The property induces an alternative formulation of the Chinese postman problem which is applicable for solving the K-best Chinese postman problem.

Since the weight of each edge is nonnegative, there exists an optimal postman's tour for the Chinese postman problem such that each edge is traversed at most twice. The following lemma is an extension of this property.

LEMMA 2.1. *The Chinese postman problem has K distinct feasible solutions $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^K$ which satisfy the following two conditions:*

(1) $\boldsymbol{w}\boldsymbol{x}^1 \leq \boldsymbol{w}\boldsymbol{x}^2 \leq \cdots \leq \boldsymbol{w}\boldsymbol{x}^K \leq \boldsymbol{w}\boldsymbol{x}'$ for any feasible solution $\boldsymbol{x}' \neq \boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^K$,

(2) $k = 1, 2, \ldots, K$, $\boldsymbol{x}^k(e) \leq 2k$ for all $e \in E$.

Proof. Let $(\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^{k-1})$ be a sequence of $k-1$ feasible solutions satisfying the conditions (1) and (2). Since the edge weights $\boldsymbol{w}(e)$ are nonnegative, there exists a feasible solution $\boldsymbol{x}^k$ such that the sequence $(\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^k)$ satisfies the condition (1). Now consider the case that the sequence $(\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^k)$ violates the condition (2). Let $e' \in E$ be an edge with $\boldsymbol{x}^k(e') > 2k$. Since $\boldsymbol{x}^k$ is feasible to the Chinese postman problem, it is clear that the solution:

$$\boldsymbol{x}'(e) = \begin{cases} \boldsymbol{x}^k(e) & \text{if } e \neq e', \\ \boldsymbol{x}^k(e) - 2 & \text{if } e = e', \end{cases}$$

is also feasible and $\boldsymbol{x}'(e') \geq 2k-1 > 2(k-1)$. Thus the solution $\boldsymbol{x}'$ is distinct from the solutions $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^{k-1}$. From the assumption that $\boldsymbol{w}$ is nonnegative, the solutions $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^{k-1}, \boldsymbol{x}'$ are k-best solutions. By applying this procedure iteratively, we can construct a sequence of K feasible solutions satisfying the conditions (1) and (2). □

The above lemma shows that when we solve K-best Chinese postman problems, it is sufficient to consider the set of finite number of feasible solutions satisfying $\boldsymbol{x}(e) \leq 2K$ for all $e \in E$.

In the rest of this paper, we consider the following problem, rather than the problem described in the previous section, for simplicity of the notation. We call the following problem **CPP** in this paper:

$$\begin{array}{lll}(\mathbf{CPP}): & \text{minimize} & \boldsymbol{w}\boldsymbol{x} = \sum_{e\in E} \boldsymbol{w}(e)\boldsymbol{x}(e),\\ & \text{subject to} & \boldsymbol{b} \geq \boldsymbol{x} \geq \boldsymbol{a},\\ & & \sum_{e\in\delta(v)} \boldsymbol{x}(e) \text{ is even} \quad \forall v \in V_1,\\ & & \sum_{e\in\delta(v)} \boldsymbol{x}(e) \text{ is odd} \quad \forall v \in V \setminus V_1,\\ & & \boldsymbol{x} \in \mathcal{Z}_+^E,\end{array}$$

where $\boldsymbol{w} \in \mathcal{Q}_+^E$ is a nonnegative weight function, $\boldsymbol{a}$ and $\boldsymbol{b}$ are nonnegative integer vectors in $\mathcal{Z}_+^E$, and V_1 denotes a subset of vertices. Clearly, an optimal solution of the ordinary Chinese postman problem is obtained by setting $\boldsymbol{a} = (1, 1, \ldots, 1)^T$, $\boldsymbol{b} = (2, 2, \ldots, 2)^T$, and $V_1 = V$. When we need K-best solutions of the ordinary Chinese postman problem, it is sufficient to replace $\boldsymbol{b}$ by $(2K, 2K, \ldots, 2K)^T$. It is easily seen that in the case $\boldsymbol{a} \leq \boldsymbol{b} \leq \boldsymbol{a} + \mathbf{1}$, this problem becomes the T-join problem [6], [11].

As in the ordinary Chinese postman problem, an optimal solution of **CPP** has the following property.

CLAIM 2.2. *If* **CPP** *has a feasible solution, then it has an optimal solution* $\boldsymbol{x}^*$ *such that, for all edges* $e \in E$, $\boldsymbol{x}^*(e)$ *is either* $\boldsymbol{a}(e)$ *or* $\boldsymbol{a}(e) + 1$.

Proof. From the assumption that $\boldsymbol{w}$ is nonnegative, the proof is clear. □

When an integer vector $\boldsymbol{x} \in \mathcal{Z}_+^E$ is feasible to **CPP** and it satisfies the conditions that $\boldsymbol{x}(e)$ is either $\boldsymbol{a}(e)$ or $\boldsymbol{a}(e)+1$ for each $e \in E$, we say $\boldsymbol{x}$ is a *matching-type solution of* **CPP**. Edmonds and Johnson [6] developed a polynomial time algorithm for the T-join problem and an optimal solution obtained by the algorithm is a matching-type solution.

Given an optimal **CPP** solution $\boldsymbol{x}^*$, a feasible solution $\boldsymbol{x}^{\text{2nd}}$ of **CPP** is called a *second-best solution* of **CPP** with respect to $\boldsymbol{x}^*$ if $\boldsymbol{w}\boldsymbol{x}^* \leq \boldsymbol{w}\boldsymbol{x}^{\text{2nd}} \leq \boldsymbol{w}\boldsymbol{x}'$ holds for all feasible solutions $\boldsymbol{x}' \neq \boldsymbol{x}^*$. In the rest of this section, we show some properties of a second-best solution of **CPP**.

For a given edge $e \in E$, let $\boldsymbol{u}^e$ be the 0-1-valued vector indexed by E such that

$$\boldsymbol{u}^e(e') = \begin{cases} 0 & \text{if } e' \neq e,\\ 1 & \text{if } e' = e.\end{cases}$$

We now have the following two lemmas.

LEMMA 2.3. *Let* $\boldsymbol{x}^*$ *be a matching-type optimal solution of* **CPP**. *If a second-best solution* $\boldsymbol{x}^{\text{2nd}}$ *of* **CPP** *with respect to* $\boldsymbol{x}^*$ *exists and satisfies* $\boldsymbol{x}^{\text{2nd}}(e) \geq \boldsymbol{x}^*(e) + 2$ *for some edge* $e \in E$, *then* $\boldsymbol{x}' = \boldsymbol{x}^* + 2\boldsymbol{u}^e$ *is also a second-best solution of* **CPP** (*with respect to* $\boldsymbol{x}^*$).

Proof. Since $\boldsymbol{b}(e') \geq \max\{\boldsymbol{x}^{\text{2nd}}(e'), \boldsymbol{x}^*(e')\} \geq \boldsymbol{x}'(e') \geq \boldsymbol{x}^*(e') \geq \boldsymbol{a}(e')$ for every $e' \in E$, $\boldsymbol{x}'$ is a feasible solution of **CPP**.

Since $\boldsymbol{w}\boldsymbol{x}' \geq \boldsymbol{w}\boldsymbol{x}^{\text{2nd}}$ is obvious, it is sufficient to show the reverse inequality. By the definition of $\boldsymbol{x}^{\text{2nd}}$, it is clear that $\boldsymbol{x}^{\text{2nd}} - 2\boldsymbol{u}^e$ is feasible to **CPP**. Then, it follows that $\boldsymbol{w}\boldsymbol{x}^{\text{2nd}} = \boldsymbol{w}(\boldsymbol{x}^{\text{2nd}} - 2\boldsymbol{u}^e) + 2\boldsymbol{w}\boldsymbol{u}^e \geq \boldsymbol{w}\boldsymbol{x}^* + 2\boldsymbol{w}\boldsymbol{u}^e = \boldsymbol{w}(\boldsymbol{x}^* + 2\boldsymbol{u}^e) = \boldsymbol{w}\boldsymbol{x}'$. □

LEMMA 2.4. *Let* $\boldsymbol{x}^*$ *be a matching-type optimal solution of* **CPP**. *Assume that there is a second-best solution* $\boldsymbol{x}^{\text{2nd}}$ *with respect to* $\boldsymbol{x}^*$ *satisfying, for any* $e \in E$, $\boldsymbol{x}^{\text{2nd}}(e) < \boldsymbol{x}^*(e)+2$. *Then there also exists a second-best solution* $\boldsymbol{x}'$ *that is of matching type.*

Proof. If $\boldsymbol{x}^{\text{2nd}}$ is of matching type, there is nothing to prove. Suppose that there exists an edge e with $\boldsymbol{b}(e) \geq \boldsymbol{x}^{\text{2nd}}(e) > \boldsymbol{a}(e) + 1$. Since $\boldsymbol{x}^{\text{2nd}}(e) < \boldsymbol{x}^*(e) + 2$ and $\boldsymbol{x}^*(e) \leq \boldsymbol{a}(e) + 1$, $\boldsymbol{x}^{\text{2nd}}(e) \leq \boldsymbol{a}(e) + 2$ holds; thus it implies $\boldsymbol{x}^{\text{2nd}}(e) = \boldsymbol{a}(e) + 2$.

Here we consider the case $\boldsymbol{x}^*(e) = \boldsymbol{a}(e)$. Then we have $\boldsymbol{x}^{\text{2nd}}(e) = \boldsymbol{a}(e) + 2 = \boldsymbol{x}^*(e) + 2$, which contradicts $\boldsymbol{x}^{\text{2nd}}(e) < \boldsymbol{x}^*(e) + 2$.

Now we just have the case $\boldsymbol{x}^*(e) = \boldsymbol{a}(e) + 1$. Let $\boldsymbol{x}' = \boldsymbol{x}^{\text{2nd}} - 2\boldsymbol{u}^e$. With respect to $\boldsymbol{x}'$, the parity of the degree of each vertex is the same as $\boldsymbol{x}^{\text{2nd}}$ and $\boldsymbol{b} \geq \boldsymbol{x}' \geq \boldsymbol{a}$ holds by $\boldsymbol{x}'(e) = \boldsymbol{a}(e)$. It implies that $\boldsymbol{x}'$ is feasible to **CPP**. Clearly, $\boldsymbol{x}^* \neq \boldsymbol{x}'$ since $\boldsymbol{x}^*(e) = \boldsymbol{a}(e) + 1 \neq \boldsymbol{a}(e) = \boldsymbol{x}'(e)$. By the definitions of $\boldsymbol{x}'$ and $\boldsymbol{x}^{\text{2nd}}$, $\boldsymbol{w}\boldsymbol{x}' = \boldsymbol{w}\boldsymbol{x}^{\text{2nd}}$. In this way, we can decrease the number of edges satisfying $\boldsymbol{x}^{\text{2nd}}(e) > \boldsymbol{a}(e) + 1$ from $\boldsymbol{x}^{\text{2nd}}$ and in the sequel, a matching-type second-best solution is obtained. □

Summarizing the lemmas above, we have the following theorem, which shows the existence of a second-best solution possessing the properties in Lemma 2.3 or 2.4.

THEOREM 2.5. *Assume that there exist an optimal and a second-best solution of* **CPP**. *Let $\boldsymbol{x}^*$ be a matching-type optimal solution of* **CPP**. *Then, there always exists a second-best solution $\boldsymbol{x}^{\text{2nd}}$ with respect to $\boldsymbol{x}^*$ such that either $\boldsymbol{x}^{\text{2nd}}$ is of matching type or $\boldsymbol{x}^{\text{2nd}} = \boldsymbol{x}^* + 2\boldsymbol{u}^e$ for an edge $e \in E$.*

Given a matching-type optimal solution $\boldsymbol{x}^*$ of **CPP**, we call a second-best solution $\boldsymbol{x}^{\text{2nd}} = \boldsymbol{x}^* + 2\boldsymbol{u}^e$ for an edge $e \in E$ a *non-matching-type second-best solution* (with respect to $\boldsymbol{x}^*$).

Now we show a simple property of a matching-type second best-solution of **CPP**. It plays an important role in our algorithm when a matching-type second-best solution of **CPP** exists. Let $\boldsymbol{x}^*$ be a matching-type optimal solution of **CPP** and $\boldsymbol{x}^{\text{2nd}}$ a matching-type second-best solution with respect to $\boldsymbol{x}^*$. Then it is clear that $-1 \leq \boldsymbol{x}^{\text{2nd}}(e) - \boldsymbol{x}^*(e) \leq 1$ for any $e \in E$. Denote by $G(\boldsymbol{x}^*, \boldsymbol{x}^{\text{2nd}})$ a graph induced by the edge subset $\{e \in E \mid \boldsymbol{x}^{\text{2nd}}(e) - \boldsymbol{x}^*(e) \neq 0\}$. For each vertex $v \in V$,

$$\left| \sum_{e \in \delta(v)} (\boldsymbol{x}^{\text{2nd}}(e) - \boldsymbol{x}^*(e)) \right|$$

is even, and it implies that

$$\sum_{e \in \delta(v)} |\boldsymbol{x}^{\text{2nd}}(e) - \boldsymbol{x}^*(e)|$$

is also even.

Therefore, the graph $G(\boldsymbol{x}^*, \boldsymbol{x}^{\text{2nd}})$ satisfies the requirement that the number of edges incident with each vertex be even, i.e., that it be Eulerian.

Now we have the following lemma.

LEMMA 2.6. *Let $\boldsymbol{x}^*$ be a matching-type optimal solution of* **CPP**. *Assume that there exists a matching-type second-best solution with respect to $\boldsymbol{x}^*$. Then we can construct a matching-type second-best solution $\boldsymbol{x}'$ such that the graph $G(\boldsymbol{x}^*, \boldsymbol{x}')$ consists of a single elementary cycle.*

Proof. From the assumption, there exists a matching-type second-best solution $\boldsymbol{x}^{\text{2nd}}$ with respect to $\boldsymbol{x}^*$. The case that $G(\boldsymbol{x}^*, \boldsymbol{x}^{\text{2nd}})$ consists of exactly one elementary cycle is trivial. If not, the graph $G(\boldsymbol{x}^*, \boldsymbol{x}^{\text{2nd}})$ contains at least two elementary cycles since $G(\boldsymbol{x}^*, \boldsymbol{x}^{\text{2nd}})$ is Eulerian. Let $C \subseteq E$ be one of such cycles. Let

$$\boldsymbol{d}(e) = \begin{cases} 1 & \text{if } e \in C \text{ and } \boldsymbol{x}^*(e) = \boldsymbol{a}(e), \\ -1 & \text{if } e \in C \text{ and } \boldsymbol{x}^*(e) = \boldsymbol{a}(e) + 1, \\ 0 & \text{if } e \notin C, \end{cases}$$

and $\boldsymbol{x}' = \boldsymbol{x}^* + \boldsymbol{d}$. From the definitions of $G(\boldsymbol{x}^*, \boldsymbol{x}^{2\text{nd}})$ and $\boldsymbol{d}$, $\boldsymbol{x}'$ satisfies $\boldsymbol{a}(e) \leq \min\{\boldsymbol{x}^*(e), \boldsymbol{x}^{2\text{nd}}(e)\} \leq \boldsymbol{x}'(e) \leq \max\{\boldsymbol{x}^*(e), \boldsymbol{x}^{2\text{nd}}(e)\} \leq \boldsymbol{b}(e)$ for any $e \in E$. Then, it is clear that $\boldsymbol{x}'$ is also feasible to **CPP**. Obviously $\boldsymbol{wd} \geq 0$ since if $\boldsymbol{wd} < 0$ then $\boldsymbol{wx}' = \boldsymbol{wx}^* + \boldsymbol{wd} < \boldsymbol{wx}^*$, which leads to a contradiction. If $\boldsymbol{wd} = 0$, then $\boldsymbol{wx}' = \boldsymbol{wx}^*$ and we can choose $\boldsymbol{x}'$ as a second-best solution. By the definition of $\boldsymbol{x}'$, it is clear that $G(\boldsymbol{x}^*, \boldsymbol{x}')$ consists of one elementary cycle. Now consider the case $\boldsymbol{wd} > 0$. Let $\boldsymbol{x}'' = \boldsymbol{x}^{2\text{nd}} - \boldsymbol{d}$. Then $\boldsymbol{x}''$ is feasible to **CPP** and $\boldsymbol{wx}'' = \boldsymbol{wx}^{2\text{nd}} - \boldsymbol{wd} < \boldsymbol{wx}^{2\text{nd}}$. Since $\boldsymbol{x}'' \neq \boldsymbol{x}^*$, this is a contradiction. □

The above lemma leads to an algorithm for finding a matching-type second-best solution, if it exists.

3. An algorithm for finding a second-best solution of CPP. In this section, we describe an algorithm for finding a second-best solution of **CPP**.

Given a matching-type optimal solution $\boldsymbol{x}^*$, we define a weight function $\tilde{\boldsymbol{w}}$ on E as:

$$\tilde{\boldsymbol{w}}(e) = \begin{cases} \boldsymbol{w}(e) & \text{if } \boldsymbol{x}^*(e) = \boldsymbol{a}(e), \\ -\boldsymbol{w}(e) & \text{if } \boldsymbol{x}^*(e) = \boldsymbol{a}(e) + 1. \end{cases}$$

Denote by C any elementary cycle in G. For a matching-type solution $\boldsymbol{x}$, let $\boldsymbol{x} \triangle C$ be the integer vector in $\mathcal{Z}_+^E$ such that:

$$\boldsymbol{x} \triangle C\ (e) = \begin{cases} \boldsymbol{x}(e) + 1 & \text{if } e \in C \quad \text{and} \quad \boldsymbol{x}(e) = \boldsymbol{a}(e), \\ \boldsymbol{x}(e) - 1 & \text{if } e \in C \quad \text{and} \quad \boldsymbol{x}(e) = \boldsymbol{a}(e) + 1, \\ \boldsymbol{x}(e) & \text{if } e \notin C. \end{cases}$$

It is clear that $\boldsymbol{x} \triangle C$ is also a matching type. If a matching-type second-best solution exists, then according to Lemma 2.6, we can construct one by finding an elementary cycle $C \subseteq E' = \{e \in E \mid \boldsymbol{a}(e) + 1 \leq \boldsymbol{b}(e)\}$ that minimizes $\sum_{e \in C} \tilde{\boldsymbol{w}}(e)$. Let $\mathcal{C}$ be the set of elementary cycles in the graph $G' = (V, E')$. Now we denote by $P(G, \boldsymbol{x}^*)$ the problem

$$\begin{array}{lll} P(G, \boldsymbol{x}^*): & \text{minimize} & \displaystyle\sum_{e \in C} \tilde{\boldsymbol{w}}(e), \\ & \text{subject to} & C \in \mathcal{C}. \end{array}$$

An elementary cycle C^* obtained by solving $P(G, \boldsymbol{x}^*)$ is called a *minimum elementary cycle*; then $\boldsymbol{x}^* \triangle C^*$ is a matching-type second-best solution of **CPP**. Here, from Theorem 2.5, we can develop the following algorithm that finds a second-best solution with respect to a matching-type optimal solution $\boldsymbol{x}^*$.

ALGORITHM 2-BEST.

Inputs: Graph $G = (V, E)$, weight function $\boldsymbol{w}$, lower bound $\boldsymbol{a}$, upper bound $\boldsymbol{b}$, vertex subset V_1, and a matching-type optimal **CPP** solution $\boldsymbol{x}^*$.

Output: A second-best solution $\boldsymbol{x}^{2\text{nd}}$, if it exists; and else say "none exist".

Step 1. Define a weight function $\tilde{\boldsymbol{w}}$ on E as:

$$\tilde{\boldsymbol{w}}(e) = \begin{cases} \boldsymbol{w}(e) & \text{if } \boldsymbol{x}^*(e) = \boldsymbol{a}(e), \\ -\boldsymbol{w}(e) & \text{if } \boldsymbol{x}^*(e) = \boldsymbol{a}(e) + 1. \end{cases}$$

Step 2.1 Solve $P(G, \boldsymbol{x}^*)$ and obtain a minimum elementary cycle $C^* \subseteq E'$.

If a minimum elementary cycle exists, set $W_{C^*} = \sum_{e \in C^*} \tilde{\boldsymbol{w}}(e)$; else, set $W_{C^*} = \infty$.

Step 2.2 Find an edge $e^* \in E'' = \{e' \in E \mid \boldsymbol{x}^*(e') + 2 \leq \boldsymbol{b}(e')\}$ such that $\boldsymbol{w}(e^*) = \min_{e' \in E''} \boldsymbol{w}(e')$.
If $E'' \neq \emptyset$, set $W_{e^*} = 2\boldsymbol{w}(e^*)$; else, set $W_{e^*} = \infty$.

Step 3. In the case that $W_{C^*} = W_{e^*} = \infty$, then say "none exist" and stop.
If $W_{e^*} \leq W_{C^*}$, then set $\boldsymbol{x}^{\text{2nd}} = \boldsymbol{x}^* + 2\boldsymbol{u}^{e^*}$; otherwise, set $\boldsymbol{x}^{\text{2nd}} = \boldsymbol{x}^* \triangle C^*$.
Output $\boldsymbol{x}^{\text{2nd}}$ and stop.

To show the computational effort of the above algorithm, let $n = |V|$ and $m = |E|$. In Step 2, the problem $P(G, \boldsymbol{x}^*)$ can be solved in polynomial time since the problem finding a minimum elementary cycle on a graph without negative cycle is reduced to the minimum-cost perfect matching problem (for details, see Lawler [10, §6.2]). In the above algorithm, we already have a minimum-cost perfect matching. Hence, it is sufficient to apply a post-optimal algorithm for non-bipartite matching problems [1], [3], [4] in Step 2. The computational effort required in other steps is less than $O(n+m)$. The overall complexity of the above algorithm is $O(m + n + nT(n + m, m))$, where $T(s, t)$ denotes the time complexity of a post-optimal algorithm for non-bipartite matching problems on a graph with s vertices and t edges.

4. An extension of the algorithm to K-best CPP. In this section, we develop an algorithm that finds K-best solutions of **CPP**. Our algorithm is based upon the *binary partitioning method,* which is used in [2] and [7] for solving some K-best problems. More precisely, we partition all the feasible solutions of the given **CPP** into two subsets iteratively. Such a partition is realized by constructing two **CPP**s.

For the convenience, the problem **CPP** with graph G, weight function $\boldsymbol{w}$, lower and upper bound $\boldsymbol{a}$, $\boldsymbol{b}$, and vertex subset V_1 is denoted by $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}, \boldsymbol{b}, V_1)$. We assume that an optimal solution $\boldsymbol{x}^*$ and a second-best solution $\boldsymbol{x}^{\text{2nd}}$ of $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}, \boldsymbol{b}, V_1)$ are obtained.

Since $\boldsymbol{x}^* \neq \boldsymbol{x}^{\text{2nd}}$, there exists an edge $e \in E$ such that $\boldsymbol{x}^*(e) \neq \boldsymbol{x}^{\text{2nd}}(e)$. With respect to the edge e, we define the two integer vectors $\boldsymbol{a}'$, $\boldsymbol{b}'$ indexed by E as:

$$\boldsymbol{a}'(e') = \begin{cases} \boldsymbol{a}(e'), & \text{if } e' \neq e, \\ \min\{\boldsymbol{x}^*(e), \boldsymbol{x}^{\text{2nd}}(e)\} + 1, & \text{if } e' = e, \end{cases}$$

$$\boldsymbol{b}'(e') = \begin{cases} \boldsymbol{b}(e'), & \text{if } e' \neq e, \\ \min\{\boldsymbol{x}^*(e), \boldsymbol{x}^{\text{2nd}}(e)\}, & \text{if } e' = e. \end{cases}$$

In our algorithm, two problems $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}, \boldsymbol{b}', V_1)$ and $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}', \boldsymbol{b}, V_1)$ are constructed and maintained. Then it is clear that each feasible solution of the original problem $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}, \boldsymbol{b}, V_1)$ is feasible to exactly one of two problems $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}, \boldsymbol{b}', V_1)$ and $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}', \boldsymbol{b}, V_1)$. In addition, when the solution $\boldsymbol{x}^*$ is feasible to one of these two problems, then $\boldsymbol{x}^{\text{2nd}}$ is feasible to another one. Here we denote these two problems by $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}_1, \boldsymbol{b}_1, V_1)$ and $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}_2, \boldsymbol{b}_2, V_1)$, and we may assume that $\boldsymbol{x}^*$ is feasible to $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}_1, \boldsymbol{b}_1, V_1)$ and $\boldsymbol{x}^{\text{2nd}}$ is feasible to $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}_2, \boldsymbol{b}_2, V_1)$ without loss of generality. From the definition of these two problems, it is obvious that $\boldsymbol{x}^*$ is a matching-type optimal solution of $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}_1, \boldsymbol{b}_1, V_1)$ and $\boldsymbol{x}^{\text{2nd}}$ is a matching-type optimal solution of $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}_2, \boldsymbol{b}_2, V_1)$. Thus, the conditions of Theorem 2.5 are maintained and Algorithm 2-Best finds second-best solutions of these two problems, respectively.

By using the *best first search rule,* the following algorithm is applied.

```
ALGORITHM $K$-BEST.
Inputs: Graph $G = (V, E)$, weight function $\boldsymbol{w}$, lower bound $\boldsymbol{a}$, upper bound $\boldsymbol{b}$,
    vertex subset $V_1$, and positive integer $K$.
Outputs: Sequence of $K$ distinct solutions $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^K$ feasible to
    $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}, \boldsymbol{b}, V_1)$ such that $\boldsymbol{wx}^1 \leq \boldsymbol{wx}^2 \leq \cdots \leq \boldsymbol{wx}^K \leq \boldsymbol{wx}'$ for any feasible solution $\boldsymbol{x}' \neq \boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^K$, if it exists; else say "none exist".
Step 0. Solve $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}, \boldsymbol{b}, V_1)$ and find a matching-type best solution $\boldsymbol{x}^*$.
    Find a second-best solution $\boldsymbol{x}^{\text{2nd}}$ of $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}, \boldsymbol{b}, V_1)$ with respect to $\boldsymbol{x}^*$.
    Set $\mathcal{P} = \{(\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}, \boldsymbol{b}, V_1), \boldsymbol{x}^*, \boldsymbol{x}^{\text{2nd}})\}$.
    Output $\boldsymbol{x}^*$ as $\boldsymbol{x}^1$ (a best solution).
    Set $k = 2$.
Step 1. If $k > K$, then stop. Else if $\mathcal{P} = \emptyset$, then say "none exist" and stop.
Step 2. Let $(\mathbf{CPP}(G, \boldsymbol{w}, \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{b}}, V_1), \boldsymbol{x}^*, \boldsymbol{x}^{\text{2nd}})$ be an element of $\mathcal{P}$ such that

        $$\boldsymbol{wx}^{\text{2nd}} = \min\{\boldsymbol{wx}'' \mid (\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}', \boldsymbol{b}', V_1), \boldsymbol{x}', \boldsymbol{x}'') \in \mathcal{P}\}.$$

    Output $\boldsymbol{x}^{\text{2nd}}$ as $\boldsymbol{x}^k$ (a $k$th-best solution).
    Delete $(\mathbf{CPP}(G, \boldsymbol{w}, \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{b}}, V_1), \boldsymbol{x}^*, \boldsymbol{x}^{\text{2nd}})$ from $\mathcal{P}$.
Step 3. Construct two problems

        $$\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}_1, \boldsymbol{b}_1, V_1) \quad \text{and} \quad \mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}_2, \boldsymbol{b}_2, V_1).$$

Step 4.1. Find a second-best solution $\boldsymbol{x}'$ of $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}_1, \boldsymbol{b}_1, V_1)$.
    If no second-best solution exists, then go to Step 4.2.
    Else, add $(\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}_1, \boldsymbol{b}_1, V_1), \boldsymbol{x}^*, \boldsymbol{x}')$ to $\mathcal{P}$.
Step 4.2. Find a second-best solution $\boldsymbol{x}''$ of $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}_2, \boldsymbol{b}_2, V_1)$.
    If no second-best solution exists, then go to Step 5.
    Else add $(\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}_2, \boldsymbol{b}_2, V_1), \boldsymbol{x}^{\text{2nd}}, \boldsymbol{x}'')$ to $\mathcal{P}$.
Step 5. Set $k = k + 1$, and go to Step 1.
```

Now we discuss the memory requirement and the time complexity of the above algorithm.

In each iteration, we delete one **CPP** from the set of problems $\mathcal{P}$ and add at most two **CPP**s to $\mathcal{P}$; i.e., the number of problems in the set $\mathcal{P}$ increases by at most 1. Hence, the memory requirement of the algorithm is less than $O(K(n+m))$.

By applying Edmonds's technique in [5], the ordinary Chinese postman problem is reduced to a non-bipartite matching problem and we can obtain a matching-type optimal solution of $\mathbf{CPP}(G, \boldsymbol{w}, \boldsymbol{a}, \boldsymbol{b}, V_1)$ in polynomial time [1], [3], [4]. Here we denote the computational efforts required to obtain a matching-type optimal solution in Step 1 by $S(n, m)$. In §3, we described an $O(m + n + nT(n + m, m))$ algorithm for solving a 2-best **CPP**, where $T(s, t)$ denotes the time complexity of a post-optimal algorithm for non-bipartite matching problems defined on a graph with s vertices and t edges [1], [3], [4]. Since the number of problems in the set $\mathcal{P}$ is bounded by $O(K)$, we can find a triplet $(\mathbf{CPP}(G, \boldsymbol{w}, \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{b}}, V_1),\ \boldsymbol{x}^*, \boldsymbol{x}^{\text{2nd}})$ and delete it from $\mathcal{P}$ in Step 2 in $O(n + m + \log K)$ time. Two triplets are added in Steps 4.1 and 4.2 with $O(n + m + \log K)$ computational efforts, by using a comfortable data structure. The above algorithm outputs one solution and solves two 2-best **CPP**s in each iteration. Thus overall time complexity is $O(S(n, m) + K(n + m + \log K + nT(n + m, m)))$.

5. Conclusion. In this paper, we treat the 2-best Chinese postman problem that finds a second-best solution of the problem. We also consider the K-best solutions of the problem as an extension of the 2-best problem. We developed an algorithm to solve the problem.

REFERENCES

[1] M. Ball and U. Derigs, *An analysis of alternative strategies for implementing matching algorithms*, Networks, 13 (1983), pp. 517–549.

[2] C. R. Chegireddy and H. W. Hamacher, *Algorithms for finding K-best perfect matchings*, Discrete Appl. Math., 18 (1987), pp. 155–165.

[3] W. Cunningham and A. Marsh, *A primal algorithm for optimum matching*, Math. Programming Stud., 8 (1983), pp. 517–549.

[4] U. Derigs, *A shortest augmenting path method for solving minimal perfect matching problems*, Networks, 11 (1981), pp. 379–390.

[5] J. Edmonds, *Path, trees, and flowers*, Canad. J. Math., 17 (1965), pp. 449–467.

[6] J. Edmonds and E. L. Johnson, *Matching, Euler tour and the chinese postman*, Math. Programming, 5 (1973), pp. 88–124.

[7] H. Hamacher and M. Queyranne, *K-best solutions to combinatorial optimization problems*, Ann. Oper. Res., 4 (1985/6), pp. 123–143; Res. Rep. No. 83-5, Industrial and Systems Engineering Dept., Univ. of Florida, Gainesville, IL, 1981.

[8] M. ko Kwan, *Graphic programming using odd or even points*, Chinese J. Math., 1 (1962), pp. 237–277.

[9] E. L. Lawler, *A procedure for computing the k-th best solutions to discrete optimization problems and its application to the shortest path problem*, Management Sci., 18 (1972), pp. 401–405.

[10] ———, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York, 1976.

[11] L. Lovász and M. D. Plummer, *Matching Theory*, North-Holland, 1986.

[12] K. G. Murty, *An algorithm for ranking all the assignments in order of increasing cost*, Oper. Res., 16 (1968), pp. 682–687.

SIAM J. OPTIMIZATION
Vol. 3, No. 4, pp. 734–750, November 1993

GENERATING FENCHEL CUTTING PLANES FOR KNAPSACK POLYHEDRA*

E. A. BOYD[†]

Abstract. The author recently proposed a class of cutting planes for integer programs called Fenchel cuts which distinguish themselves from more conventional cuts in that they are generated by directly seeking to solve the separation problem rather than by using explicit knowledge of the polyhedral structure of the integer program. An algorithm for generating Fenchel cuts is presented and described in detail for the separation problem associated with knapsack polyhedra. Computational results are presented for a collection of real-world integer programs to demonstrate the effectiveness of the cutting planes.

Key words. cutting planes, integer programming, knapsack polyhedron

AMS subject classification. 52B12

1. Introduction.

In a 1983 paper [6], Crowder, Johnson, and Padberg demonstrated the effectiveness of using strong cutting planes to solve integer programs. They were able to solve all but one of a collection of integer programs to optimality in under 15 minutes even though "most [of the problems] were originally considered not amenable to exact solution in economically feasible computation times" ([6, p. 828]). The largest, most difficult problem required only slightly less than an hour to solve. Their work sparked renewed interest in the possibility of solving large integer programs with no special structure to optimality.

While the work by Crowder, Johnson, and Padberg represents an important success story, successful research on cutting plane methods extends far beyond this single work. Phenomenal computational results have also been achieved on a number of important specialized problem classes using cutting planes. Early work by Grötschel [11], Padberg and Rinaldi [20], and others led to the solution of much larger traveling salesman problems than had ever been solved previously and ongoing work on this problem continues to yield substantial performance improvements. Cutting plane methods have even begun to find their way into publicly available general purpose algorithms such as IBM's OSL and Georgia Tech's MINTO [22], [23].

The most fundamental problem arising in the use of cutting planes to solve integer programs is the *separation problem*—the problem of finding an inequality that is valid for the polyhedron defined by the convex hull of all feasible integer points but that is violated by the optimal solution to the linear programming relaxation of the problem. Commonly, separation algorithms are devised for known classes of valid inequalities with good theoretical characteristics, usually facet classes. In practice, good separation algorithms are far more scarce than known classes of cutting planes and in general they appear to be more elusive.

We recently proposed a class of cutting planes called Fenchel cuts which differ from more conventional cutting planes in that they focus directly on the separation problem without reference to an underlying class of cutting planes. Fenchel cuts are generated by maximizing a piecewise linear concave function $v(\lambda)$, with cutting planes corresponding to values of λ for which $v(\lambda) > 0$. It can be shown that Fenchel cuts

* Received by the editors January 3, 1991; accepted for publication (in revised form) July 9, 1992. This work was sponsored in part by National Science Foundation grant ECS-8809053.

[†] Department of Industrial Engineering, Texas A & M University, College Station, Texas 77843-3131.

are the deepest cuts that can be generated for a problem in a well-defined sense and that if the maximum value of $v(\lambda)$ is nonpositive then no cutting plane exists. Fenchel cuts and their relation to another class of cutting planes associated with Lagrangian relaxation are described in [3].

In this paper we present an algorithm for generating Fenchel cutting planes for knapsack polyhedra. The algorithm can be applied to general integer programs by using cutting planes generated from the knapsack polyhedra associated with each individual constraint of an integer program. The computational value of an exact separation algorithm is then demonstrated by *provably* optimizing a linear function over the intersection of the knapsack polyhedra defined by the constraints of the integer programs used by Crowder, Johnson, and Padberg in [6], a result that has not been achieved prior to this paper.

2. Fenchel cuts. Theoretical aspects of Fenchel cuts are developed in [3] and [5]. In this section, we outline sufficient theory for the developments presented in this paper.

Consider the following problem.

$$\text{(P)}\qquad \begin{array}{ll} \max & cx \\ \text{s.t.} & Qx = q, \\ & Rx \le r, \\ & Ax = a, \\ & Bx \le b, \\ & x \text{ integer.} \end{array}$$

Let $F = \{x : Ax = a, Bx \le b, x \text{ integer}\}$ and let $\mathcal{P}_F$ denote the convex hull of F. We assume for simplicity of exposition that $\mathcal{P}_F$ is bounded. Further, let $\hat{x}$ be feasible for the constraints of (P) with the exception of the integrality restriction. Conceptually, $\hat{x}$ can be thought of as a point generated by solving the linear programming relaxation of (P). The cut generation procedure to be described seeks an inequality that is not satisfied by $\hat{x}$ but that contains $\mathcal{P}_F$ and therefore the feasible region of (P).

As but one example, F might be defined by upper and lower bound constraints on the variables together with a single constraint taken from the original problem (P). This collection of relaxations F, one defined by each row of (P), is exactly the collection of relaxations used by Crowder, Johnson, and Padberg in [6].

Let the rows of D span the nullspace of A. We define $f(\lambda)$ and $v(\lambda)$ as follows.

$$f(\lambda) = \max\{\lambda Dx : x \in \mathcal{P}_F\},$$
$$v(\lambda) = \lambda D\hat{x} - f(\lambda).$$

The following is proved in [3].

THEOREM 2.1. *There exists a value λ for which $v(\lambda) > 0$ if and only if there exists a hyperplane $\lambda Dx \le f(\lambda)$ separating $\hat{x}$ from $\mathcal{P}_F$.*

The practical implication of Theorem 2.1 is that the question of whether or not there exists a cutting plane separating $\hat{x}$ from $\mathcal{P}_F$ can be answered by investigating whether or not the function $v(\lambda)$ achieves a positive value. For any fixed value of λ the inequality

$$\lambda Dx \le f(\lambda)$$

is valid for $\mathcal{P}_F$, and this inequality separates $\mathcal{P}_F$ from $\hat{x}$ if and only if $v(\lambda) > 0$. Due to connections with Fenchel duality such cuts were deemed Fenchel cuts.

The following theorem, also proved in [3], makes note of important theoretical properties that simplify finding values of λ for which $v(\lambda) > 0$.

THEOREM 2.2. *The function $v(\lambda)$ is piecewise linear and concave. Specifically, $v(\lambda)$ can be expressed as*

$$v(\lambda) = \min\{\lambda D\hat{x} - \lambda D x^i : x^i \in E(\mathcal{P}_F)\}$$

where $E(\mathcal{P}_F)$ is the set of extreme points of $\mathcal{P}_F$.

From the definition of $v(\lambda)$ we have the following observation.

OBSERVATION 1. *For any scalar $\omega > 0$, $v(\omega\lambda) = \omega v(\lambda)$.*

The immediate implication of this observation is that if $v(\lambda)$ achieves a positive value it achieves a positive value on any full-dimensional set containing the origin in its strict interior. In fact, it is not difficult to verify the following observation.

OBSERVATION 2. *$v(\lambda)/\|\lambda D\|$ is the distance from $\hat{x}$ to the plane $\lambda D x = f(\lambda)$ when $\lambda D\hat{x} \le f(\lambda)$ separates $\hat{x}$ and $\mathcal{P}_F$, and the negative of this distance when it does not.*

Thus, solving the maximization problem

$$\begin{array}{ll} \max & v(\lambda) \\ \text{s.t.} & \lambda \in \Lambda = \{\lambda : \ \|\lambda D\| \le 1\} \end{array}$$

generates the deepest cut separating a point $\hat{x}$ from $\mathcal{P}_F$. In practice, it is easier to attempt to maximize $v(\lambda)$ on a linearly defined domain $\Lambda = \{\lambda : \ P\lambda \le p\}$. Further, through the appropriate choice of domain it is possible to affect the polyhedral characteristics of the generated cut. By Theorem 2.2, it follows that the problem of maximizing $v(\lambda)$ on a linearly defined domain can be solved by introducing the variable z and solving the following linear program.

$$\begin{array}{lll} & \max & z \\ (\text{G}) & \text{s.t.} & z \le \lambda D\hat{x} - \lambda D x^i, \qquad x^i \in E(\mathcal{P}_F), \\ & & \lambda \in \Lambda = \{\lambda : P\lambda \le p\}. \end{array}$$

Given an optimal solution $[\bar{\lambda}, \bar{z}]$ to (G), $\bar{\lambda}$ maximizes $v(\lambda)$ on the domain $\Lambda = \{\lambda : P\lambda \le p\}$ and has value $v(\bar{\lambda}) = \bar{z}$. This particular formulation for the problem of maximizing $v(\lambda)$ will be studied in the following sections.

In summary, Fenchel cuts are generated by seeking to maximize the function $v(\lambda)$ on any domain that is full dimensional and contains the origin in its strict interior. Any value of λ with $v(\lambda) > 0$ corresponds to a cutting plane, and if the maximum value of $v(\lambda)$ is zero then this represents a proof that there exists no cutting plane separating $\hat{x}$ from $\mathcal{P}_F$.

The way in which Fenchel cuts are generated is fundamentally different from the way in which most cutting planes are generated, and as such they provide unique theoretical and computational possibilities. Most cutting planes are derived from classes of theoretically studied facets for given problems and the effectiveness of these cutting planes is governed by the existence of good separation algorithms for generating violated cuts. In contrast, separation is the *essence* of Fenchel cuts.

3. Fenchel cuts for knapsack polyhedra. While Fenchel cuts are broadly applicable, if they are to prove effective in practice it must be possible to quickly maximize or nearly maximize $v(\lambda)$ since cutting plane routines are normally called many times in the course of algorithms for solving integer programs. Thus, it becomes necessary to discuss the application of Fenchel cuts to a specific class of polyhedra. In this paper, we focus on the problem of generating Fenchel cuts for knapsack polyhedra, formally defined as

$$\mathcal{P}_F^i = \text{conv}\{x : a^i x \leq b_i, \;\; 0 \leq x \leq 1, \;\; x \text{ integer}\},$$

where a^i and x are n-vectors. For simplicity of exposition we assume that $a^i > 0$ since this is easily achieved by complementing variables. In the following two sections we describe how to efficiently generate Fenchel cuts for knapsack polyhedra.

3.1. Domain restrictions. One very important way in which maximizing $v(\lambda)$ can be accelerated is through a more thorough analysis of the domain on which maximization actually occurs. As noted in Observation 2, maximizing $v(\lambda)$ on the domain $\|\lambda D\| \leq 1$ has the attractive theoretical property of generating a hyperplane for $\mathcal{P}_F^i$ that is as deep as possible in a well-defined sense. Further, if the rows of D are chosen so that they are orthonormal the domain $\|\lambda D\| \leq 1$ reduces to the simple domain $\|\lambda\| \leq 1$. Nonetheless, the nonlinearity is unattractive for obvious reasons, especially when Observation 1 presents the opportunity of choosing a linearly defined domain. From the standpoint of implementation, two natural alternative choices for the domain Λ are the L^1 unit sphere and the L^∞ unit sphere, and a strong theoretical case for these domains is made in [5].

The key to accelerating cut generation, however, comes from domain restrictions which can substantially reduce the dimension of the space in which $v(\lambda)$ must be maximized. The following theorem demonstrates how the domain can be restricted for the specific polyhedron $\mathcal{P}_F^i$ while still guaranteeing the generation of a cutting plane if one exists.

THEOREM 3.1. *Let $v(\lambda)$ be defined by $\mathcal{P}_F^i$, assume $\hat{x} \geq 0$, and let $D = I$. Then the following are true.*

1. *There exists a $\lambda \geq 0$ which maximizes $v(\lambda)$ on the domain $\|\lambda\| \leq 1$.*
2. *If $\hat{x}_j = 0$ then there exists a λ with $\lambda_j = 0$ which maximizes $v(\lambda)$ on the domain $\|\lambda\| \leq 1$.*
3. *If $\hat{x}_i = \hat{x}_j = 1$ and there exists a value of λ for which $v(\lambda) > 0$ then there exists a value of λ with $\lambda_i = \lambda_j$ for which $v(\lambda) > 0$ on the domain $\|\lambda\| \leq 1$.*

Proof. (1) Since all of the coefficients in the knapsack constraint defining the polyhedron $\mathcal{P}_F^i$ are positive by assumption, if $y \in \mathcal{P}_F^i$ then for any $x \geq 0$, $x \leq y$, it follows that $x \in \mathcal{P}_F^i$. Suppose $\overline{\lambda}$ maximizes $v(\lambda)$ on the domain $\|\lambda\| \leq 1$ but that $\overline{\lambda}_j < 0$ for some index j. Let $\overline{x} \in \mathcal{P}_F^i$ be a maximizing value of x associated with $\overline{\lambda}$; that is, $f(\overline{\lambda}) = \overline{\lambda}\overline{x}$. If $\overline{x}_j > 0$ then $\overline{x}'_j \in \mathcal{P}_F^i$ obtained from $\overline{x}$ by setting $\overline{x}_j = 0$ has $\overline{\lambda}\overline{x}' > \overline{\lambda}\overline{x}$; that is, $\overline{x}$ cannot be a maximizing value of x associated with $\overline{\lambda}$. Thus, assume $\overline{x}_j = 0$ and consider $\overline{\lambda}'$ formed from $\overline{\lambda}$ by setting $\overline{\lambda}_j = 0$. Clearly, $\overline{x}$ must be a maximizing value of x associated with $\overline{\lambda}'$ as well as with $\overline{\lambda}$. If not, then there exists an $\tilde{x} \in \mathcal{P}_F^i$ such that $\overline{\lambda}'\tilde{x} > \overline{\lambda}'\overline{x}$, and letting $\tilde{x}' \in \mathcal{P}_F^i$ be $\tilde{x}$ with $\tilde{x}_j = 0$ it follows that $\overline{\lambda}\tilde{x}' = \overline{\lambda}'\tilde{x} > \overline{\lambda}'\overline{x} = \overline{\lambda}\overline{x}$; that is, $\overline{x}$ is not a maximizing value of x for $\overline{\lambda}$. Thus, $v(\overline{\lambda}) = \overline{\lambda}\hat{x} - \overline{\lambda}\overline{x} \leq \overline{\lambda}'\hat{x} - \overline{\lambda}'\overline{x} = v(\overline{\lambda}')$. Using this technique to eliminate any $\lambda_j < 0$ it follows that there exists a value of $\lambda \geq 0$ which maximizes $v(\lambda)$ on the domain $\|\lambda\| \leq 1$.

(2) Let $\overline{\lambda} \geq 0$ maximize $v(\lambda)$ on the domain $\|\lambda\| \leq 1$, let $\overline{x} \in \mathcal{P}_F^i$ be a maximizing value of x associated with $\overline{\lambda}$, and assume $\hat{x}_j = 0$ but $\overline{\lambda}_j > 0$. Let $\overline{\lambda}'$ be $\overline{\lambda}$ with $\overline{\lambda}'_j = 0$. Since $\overline{\lambda}x \leq \overline{\lambda}\overline{x}$ is valid for $\mathcal{P}_F^i$ and since $\overline{\lambda} \geq 0$ it follows that $\overline{\lambda}'x \leq \overline{\lambda}\overline{x}$ is valid for all $x \in \mathcal{P}_F^i$, although it may not be a face of $\mathcal{P}_F^i$. Letting $\overline{x}' \in \mathcal{P}_F^i$ be a maximizing value of

x for $\overline{\lambda}'$ it follows from the validity of $\overline{\lambda}'x \leq \overline{\lambda}\overline{x}$ that $\overline{\lambda}'\overline{x}' \leq \overline{\lambda}\overline{x}$. Clearly, since $\overline{\lambda}\hat{x} = \overline{\lambda}'\hat{x}$ under the assumption that $\hat{x}_j = 0$, it follows that $v(\overline{\lambda}) = \overline{\lambda}\hat{x} - \overline{\lambda}\overline{x} \leq \overline{\lambda}'\hat{x} - \overline{\lambda}'\overline{x}' = v(\overline{\lambda}')$. Using this technique to eliminate any $\overline{\lambda}_j > 0$ when $\hat{x}_j = 0$ it follows that if $\hat{x}_j = 0$ then there exists a λ with $\lambda_j = 0$ which maximizes $v(\lambda)$ on the domain $\|\lambda\| \leq 1$.

(3) Let $\overline{\lambda} \geq 0$ maximize $v(\lambda)$ on the domain $\|\lambda\| \leq 1$, assume that $v(\overline{\lambda}) > 0$, and let $\overline{x} \in \mathcal{P}_F^i$ be a maximizing value of x associated with $\overline{\lambda}$. Further, suppose $\hat{x}_i = \hat{x}_j = 1$ but $\overline{\lambda}_i \neq \overline{\lambda}_j$, and assume without loss of generality that $\overline{\lambda}_i > \overline{\lambda}_j$. Let $\overline{\lambda}'$ be $\overline{\lambda}$ with $\overline{\lambda}_j' = \overline{\lambda}_i$ and let $\overline{x}' \in \mathcal{P}_F^i$ be a maximizing value of x associated with $\overline{\lambda}'$. Clearly, $\overline{\lambda}'\hat{x} = \overline{\lambda}\hat{x} + (\overline{\lambda}_j' - \overline{\lambda}_j)\hat{x}_j = \overline{\lambda}\hat{x} + \overline{\lambda}_j' - \overline{\lambda}_j$ since $\hat{x}_j = 1$. Similarly, $\overline{\lambda}'\overline{x}' = \overline{\lambda}\overline{x}' + (\overline{\lambda}_j' - \overline{\lambda}_j)\overline{x}_j' \leq \overline{\lambda}\overline{x}' + \overline{\lambda}_j' - \overline{\lambda}_j \leq \overline{\lambda}\overline{x} + \overline{\lambda}_j' - \overline{\lambda}_j$ where the first inequality follows from the fact that $\overline{x}_j' \leq 1$ and the second inequality follows from the definition of $\overline{x}$. Thus, $v(\overline{\lambda}') = \overline{\lambda}'\hat{x} - \overline{\lambda}'\overline{x}' \geq (\overline{\lambda}\hat{x} + \overline{\lambda}_j' - \overline{\lambda}_j) - (\overline{\lambda}\overline{x} + \overline{\lambda}_j' - \overline{\lambda}_j) = \overline{\lambda}\hat{x} - \overline{\lambda}\overline{x} = v(\overline{\lambda}) > 0$.

While $v(\overline{\lambda}') > 0$, it is not necessarily true that $\|\overline{\lambda}'\| \leq 1$. However, multiplying $\overline{\lambda}'$ by $1/\|\overline{\lambda}'\|$ it follows from Observation 1 that $v(\overline{\lambda}'/\|\overline{\lambda}'\|) > 0$ while $\|(\overline{\lambda}'/\|\overline{\lambda}'\|)\| = 1$. □

In point of fact, Theorem 3.1 is true when $\|\lambda\| \leq 1$ is replaced by the interior of the unit sphere of an arbitrary norm, but we do not dwell upon this point. Instead, we note that an immediate corollary of this theorem is that for *any* full-dimensional set Λ containing the origin in its strict interior a slightly weaker version of the above theorem remains true; namely, if there exists a value of λ for which $v(\lambda) > 0$ then it is always possible to find a value of λ for which $v(\lambda) > 0$ under the domain restrictions of Theorem 3.1, although not necessarily a value that maximizes $v(\lambda)$ on Λ. The proof follows by simply taking a value of λ satisfying the domain restrictions of Theorem 3.1 and scaling it so that it is contained in Λ. By Observation 1, it follows that if $v(\lambda) > 0$ then the scaled value of λ also has a positive value. We state this corollary in an equivalent but computationally more suggestive form.

COROLLARY 3.2. *Let $v(\lambda)$ be defined by $\mathcal{P}_F^i$, assume $\hat{x} \geq 0$, and let $D = I$. Further, let Λ be any full-dimensional set containing the origin in its strict interior, let S_0 be the set of indices for which $\hat{x}_i = 0$, and let S_1 be the set of indices for which $\hat{x}_i = 1$ with $k \in S_1$ some fixed index. If there exists a value of λ for which $v(\lambda) > 0$ then there exists a value of $\lambda \in \overline{\Lambda}$ for which $v(\lambda) > 0$, where*

$$\begin{aligned} \overline{\Lambda} = \{\lambda : \quad & \lambda \in \Lambda \\ & \lambda \geq 0 \\ & \lambda_i = 0 \qquad i \in S_0 \\ & \lambda_i = \lambda_k \quad\ i \in S_1\}. \end{aligned}$$

The ability to restrict some values of λ has an extremely important impact on finding values of λ for which $v(\lambda) > 0$, as will be seen in the section on computational results. In practice, when $\hat{x}$ is a subvector of a solution to a larger linear program, many of the $\hat{x}_i$ are either 0 or 1. By Corollary 3.2 the search space can have its dimension reduced by one for each variable $\hat{x}_i = 0$ and by one for each variable $\hat{x}_i = 1$ other than $i = k$ and still be *guaranteed* of finding a value of λ for which $v(\lambda) > 0$ if one exists. This reduction of dimension translates into reduced time to maximize $v(\lambda)$.

The restriction $\lambda \geq 0$ has the side effect of greatly simplifying how the L^1 unit sphere can be represented. Together with the constraints $\lambda \geq 0$ the L^1 unit sphere can simply be represented by the constraint $\sum_{i=1}^{n} \lambda_i \leq 1$ since all remaining constraints become redundant. As a computational convenience, we consider Λ defined as

$$\Lambda = \left\{ \lambda : \sum_{i=1}^{n} \lambda_i \leq \beta, \ \ \lambda \leq 1 \right\},$$

where β is any constant satisfying $0 < \beta \leq n$. In the presence of the constraints $\lambda \geq 0$, if $\beta = 1$ the domain Λ corresponds to the L^1 unit sphere intersected with the nonnegative orthant, while if $\beta = n$ the domain Λ corresponds to the L^∞ unit sphere intersected with the nonnegative orthant. We use this set Λ together with the additional domain restrictions outlined in Theorem 3.1 in the computational results presented later in this paper.

3.2. Maximizing $v(\lambda)$**.** Before embarking upon a discussion of an algorithm for maximizing $v(\lambda)$ for knapsack polyhedra $\mathcal{P}_F^i$, it is useful to address the problem of maximizing $v(\lambda)$ for a general polyhedron $\mathcal{P}_F$. The piecewise linear concavity of $v(\lambda)$, together with the ability to choose a convex domain, provide necessary theoretical properties for maximizing $v(\lambda)$. In addition, as with the maximization of Lagrangian dual functions, it is possible to prove that a subgradient of $v(\lambda)$ is generated whenever a value of $v(\lambda)$ is calculated. It is therefore possible to seek to maximize $v(\lambda)$ using subgradient techniques, generalized programming, or other well-established nondifferentiable optimization techniques. In practice, however, subgradient techniques and generalized programming very commonly demonstrate extremely poor convergence. Often this convergence is sufficiently bad to render these methods effectively useless. (The ineffectiveness of these methods is discussed in §4 and serves as a primary motivation for the work described in this paper.)

The practical motivation for the use of subgradient techniques and generalized programming is that they require only an oracle that for any value of λ returns $v(\lambda)$ and an associated extreme point x^i of $\mathcal{P}_F$ defining $v(\lambda)$. An algorithm for optimizing a linear function on $\mathcal{P}_F$ serves this purpose. However, when it is possible to *parametrically* optimize a linear function on $\mathcal{P}_F$, this stronger oracle makes it possible to develop a more efficient algorithm for maximizing $v(\lambda)$ than algorithms based on subgradient techniques or generalized programming.

Formally, we address the problem of maximizing $v(\lambda)$ by developing an algorithm for solving the problem (G) (introduced in §2) which makes use of the following oracle. For purposes of exposition we refer to the constraints $P\lambda \leq p$ as the Λ constraints and all the remaining constraints as the X constraints.

Problem FPARAM. Let $[\overline{\lambda}, \overline{z}]$ be a feasible vector for (G), let S_X be a subset of the X constraints satisfied at equality by $[\overline{\lambda}, \overline{z}]$, and let S_Λ be a subset of the Λ constraints satisfied at equality by $[\overline{\lambda}, \overline{z}]$. Given a direction $[d, 1]$ such that $[\overline{\lambda}, \overline{z}] + \theta[d, 1]$ satisfies all of the constraints S_X and S_Λ for all $\theta \geq 0$, find the largest value $\hat{\theta}$ of θ such that $[\overline{\lambda}, \overline{z}] + \hat{\theta}[d, 1]$ does not violate any of the constraints in (G), and a constraint that is violated for $\theta > \hat{\theta}$.

An algorithm for maximizing $v(\lambda)$ that makes use of an oracle for solving Problem FPARAM is presented in Fig. 1. This algorithm is fundamentally an active set method, and although it is not immediately apparent, it is identical to the primal simplex algorithm if $[\overline{\lambda}, \overline{z}]$ is initially an extreme point of the polyhedron defined by (G) and if the choice of the ascent direction in step 2 is limited to edges of this polyhedron. A detailed description of this algorithm using the primal simplex interpretation can be used to formally establish that the procedure maximizes $v(\lambda)$ after a finite number of iterations. As described, the algorithm is somewhat more flexible than the

1. *Initialize.* Choose a vector $[\overline{\lambda}, \overline{z}]$ feasible for (G), a subset S_X of the X constraints satisfied at equality by $[\overline{\lambda}, \overline{z}]$, and a subset S_Λ of the Λ constraints satisfied at equality by $[\overline{\lambda}, \overline{z}]$.
2. Choose an ascent direction $[d, 1]$ such that $[\overline{\lambda}, \overline{z}] + \theta[d, 1]$ satisfies all of the constraints S_X and S_Λ for all $\theta \geq 0$. If no such direction exists, stop; $[\overline{\lambda}, \overline{z}]$ is optimal for (G).
3. Solve Problem FPARAM for $\hat{\theta}$ and let $[\overline{\lambda}, \overline{z}] = [\overline{\lambda}, \overline{z}] + \hat{\theta}[d, 1]$. Include the constraint that defined $\hat{\theta}$ in the appropriate set S_X or S_Λ and remove from S_X and S_Λ those constraints that are not satisfied at equality by the new $[\overline{\lambda}, \overline{z}]$. Go to step 2.

FIG. 1. *Algorithm to maximize $v(\lambda)$.*

primal simplex algorithm and instead of dwelling upon this interpretation we choose simply to make some instructive observations about the algorithm.

The termination criterion given in step 2 follows from the fact that the candidate solution $[\overline{\lambda}, \overline{z}]$ is feasible for (G) and satisfies optimality conditions for a relaxation of (G). The direction $[d, 1]$ chosen in step 2 is an ascent direction for the relaxation of (G) defined by the constraints S_X and S_Λ, but may not be a true ascent direction for (G). Specifically, there may be X or Λ constraints that are candidates to be in the sets S_X and S_Λ that are not in these sets, so that any positive step length moves outside the feasible region of (G). The algorithm for solving Problem FPARAM in step 3 answers the question of how long a step may be taken in the direction $[d, 1]$ without violating any of the constraints in (G). As just noted, this step length may be 0. The algorithm presented in Fig. 1 can thus be seen to be an ascent algorithm which makes use of an oracle for generating step length.

Clearly, a good algorithm for solving Problem FPARAM will not generally exist for an arbitrary subproblem polyhedron $\mathcal{P}_F$. However, as a general observation, the ability to optimize a linear function on $\mathcal{P}_F$ is indicative of a subproblem structure that will allow the Problem FPARAM to be solved as well. This is the case for the knapsack polyhedron, and the remainder of this section examines an instance of the algorithm presented in Fig. 1 as applied to this polyhedron.

Using the domain Λ described in the previous section together with the constraints $\lambda \geq 0$, the problem (G) becomes

$$\text{(G)} \qquad \begin{array}{lll} \max & z & \\ \text{s.t.} & z - \lambda(D\hat{x} - Dx^i) \leq 0, & x^i \in E(\mathcal{P}_F), \\ & \sum_{j=1}^n \lambda_j \leq \beta, & \\ & 0 \leq \lambda \leq 1, & \end{array}$$

where β is any constant satisfying $0 < \beta \leq n$. In actuality, when solving (G) we include the constraints

$$\begin{array}{lll} \lambda_j = 0, & j \in S_0 = \{j : \hat{x}_j = 0\}, & \\ \lambda_j = \lambda_k, & j \in S_1 = \{j : \hat{x}_j = 1\}, & k \in S_1, \end{array}$$

since this was the entire point of the previous section. However, we ignore these constraints for the purposes of describing the algorithm for solving (G) since they complicate the description but add nothing conceptual; the algorithm is simply performed as stated in the subspace defined by these additional equalities.

It will prove useful in the following discussion to refer to the constraint $\sum_{j=1}^n \lambda_j \leq \beta$ as the β constraint, keeping in mind that this constraint and the variable bounds jointly comprise what are referred to as the Λ constraints. We describe the algorithm for solving (G) by describing each of the three steps outlined in Fig. 1.

Implementation of step 1. The initial value of $\overline{\lambda}$ is chosen so that it satisfies the β constraint and the bounds on λ. The value $\overline{z}$ is initialized to $v(\overline{\lambda})$, and S_X is initialized so that it contains some X constraint defining $v(\overline{\lambda})$. Throughout the algorithm, the ascent direction is always chosen so that the condition $\overline{z} = v(\overline{\lambda})$ is maintained. S_Λ is initialized to the empty set.

Implementation of step 2. Conceptually, step 2 can be performed by solving the following linear program.

$$\text{(H)} \qquad \begin{array}{ll} \max & z \\ \text{s.t.} & S_X \text{ and } S_\Lambda. \end{array}$$

Since $[\overline{\lambda}, \overline{z}]$ satisfies all of the S_X and S_Λ constraints at equality by the operation of the algorithm to maximize $v(\lambda)$, by Farkas's lemma either $[\overline{\lambda}, \overline{z}]$ is optimal for (H) (and therefore (G)) or there exists a direction $[d, 1]$ such that $[\overline{\lambda}, \overline{z}] + \theta[d, 1]$ satisfies all of the constraints S_X and S_Λ for all $\theta \geq 0$. In practice, the need to invoke a linear program solver to solve (H) each time step 2 is performed is alleviated by maintaining sets S_X and S_Λ such that together they contain a collection of at most $n{+}1$ constraints with linearly independent normals. Recall that $\lambda \in \mathbb{R}^n$ so that (G) is a problem in $\mathbb{R}^{n+1}$.

The importance of this condition is that it provides an easy way to determine if a direction of ascent relative to the constraints in S_X and S_Λ exists at $[\overline{\lambda}, \overline{z}]$, and it is an easy condition to maintain throughout the algorithm. Conceptually, if there are less than $n + 1$ constraints, then if an ascent direction exists one can be found in the nullspace of the matrix comprised of the gradients of the S_X and S_Λ constraints. In this case, the constraint returned from a solution of Problem FPARAM in step 3 is simply included in the appropriate set S_X or S_Λ. Linear independence of the new sets S_X and S_Λ is guaranteed by the choice of ascent direction from the nullspace of the initial constraints and the definition of Problem FPARAM. If there are $n + 1$ constraints then if an ascent direction exists, one of the directions defined by the nullspace of some subset of n constraints is an ascent direction. In this case, the constraint returned from a solution of Problem FPARAM in step 3 is again included in the appropriate set S_X or S_Λ, but in addition the constraint that is not included in defining the direction of ascent is removed from these sets. Again, linear independence of the new sets S_X and S_Λ is guaranteed by the choice of ascent direction and the definition of Problem FPARAM.

In summary, given the stated properties on S_X and S_Λ, step 2 of the algorithm can be completed quite efficiently. In effect, one pivot is required to prove that $[\overline{\lambda}, \overline{z}]$ is optimal or to find a direction of ascent relative to the constraints in S_X and S_Λ. Of course, while this approach overcomes a potentially long sequence of pivots each time step 2 is performed, if the point $[\overline{\lambda}, \overline{z}]$ is highly degenerate in (G) then a potentially long sequence of steps 2 and 3 combined could occur in which a value of $\theta = 0$ is returned from step 3. In fact, without taking appropriate measures step 2 could encounter the same sets S_X and S_Λ encountered previously. This possibility is discussed further in §4.

Implementation of step 3. Before discussing the algorithm for solving Problem FPARAM it is useful to consider the simpler problem of maximizing a linear function on $\mathcal{P}_F^i$. It is well known that optimizing a linear function on $\mathcal{P}_F^i$ is an NP-complete problem. It is also well known that this problem can be solved using dynamic programming. In the formulation of the dynamic program with λ defining the linear function to be maximized, the recursive relation is given by

```
if $g(j-1,k) > g(j-1,k-a_j) + \lambda_j$ then
    $g(j,k) = g(j-1,k)$
    $\mathrm{pr}(j,k) = (j-1,k)$
else
    $g(j,k) = g(j-1,k-a_j) + \lambda_j$
    $\mathrm{pr}(j,k) = (j-1,k-a_j)$
```

where $g(j,k)$ conceptually represents the optimal solution to the problem

$$\begin{array}{ll} \max & \sum_{t=1}^{j} \lambda_t x_t \\ \text{s.t.} & \sum_{t=1}^{j} a_t x_t \leq k, \\ & 0 \leq x_t \leq 1, \\ & x_t \text{ integer,} \end{array}$$

and the predecessor array $\mathrm{pr}(j,k)$ implicitly defines the x_j values by

$$\begin{array}{ll} x_j = 0 & \text{if } \mathrm{pr}(j,k) = (j-1,k), \\ x_j = 1 & \text{if } \mathrm{pr}(j,k) = (j-1,k-a_j). \end{array}$$

The value $g(n,b_i)$ is thus the optimal solution value associated with maximizing λx on $\mathcal{P}_F^i$.

In general, each stage j of the dynamic program can have as many as b_i states k and thus this dynamic programming formulation is only pseudopolynomial in the size of the problem it seeks to solve. However, using appropriate data structures it is not necessary to consider such a large number of states, and in practice the dynamic programming recursion can be solved very quickly.

The recursive relation presented in Fig. 2 is a modification of the basic dynamic programming recursion which presents a tie-breaking rule in the presence of a parameterizing vector d. An additional array $h(j,k)$ is used which is conceptually defined so that $h(j,k) = \sum_{t=1}^{j} d_t \overline{x}_t$ where the vector $\overline{x} = [\overline{x}_1, \ldots, \overline{x}_j]$ is the optimal solution defined by $\mathrm{pr}(j,k)$. The relevance of this array is that $g(j,k) = g(j,k) + \theta h(j,k)$ over the range of θ for which $\lambda + \theta d$ has the same optimal predecessor array $\mathrm{pr}(j,k)$ as when $\theta = 0$. The array $h(j,k)$ is actually used to calculate this range on θ, as the proof of the following theorem demonstrates.

THEOREM 3.3. *If the algorithm presented in Fig.* 2 *is used to maximize* λx *on* $\mathcal{P}_F^i$ *given a parameterizing vector d, then* $\theta_{\max} > 0$ *and for* $0 \leq \theta \leq \theta_{\max}$ *the optimal predecessor array* $\mathrm{pr}(j,k)$ *remains optimal when* $\lambda + \theta d$ *is maximized on* $\mathcal{P}_F^i$.

Proof. Clearly, the recursive relation of Fig. 2 yields an optimal predecessor array since if $g(j-1,k) \neq g(j-1,k-a_j) + \lambda_j$ the arrays $g(j,k)$ and $\mathrm{pr}(j,k)$ are chosen in exactly the same way as in the unmodified recursive relation, and if $g(j-1,k) = g(j-1,k-a_j) + \lambda_j$ then $\mathrm{pr}(j,k)$ can be chosen arbitrarily while yielding an optimal predecessor array.

The optimal predecessor array $\mathrm{pr}(j,k)$ remains optimal as long as for each (j,k) with $\mathrm{pr}(j,k) = (j-1,k)$ it is true that

$$g(j-1,k) + \theta h(j-1,k) \geq g(j-1,k-a_j) + \lambda_j + \theta[h(j-1,k-a_j) + d_j],$$

and for each (j,k) with $\mathrm{pr}(j,k) = (j-1,k-a_j)$ it is true that

$$g(j-1,k) + \theta h(j-1,k) \leq g(j-1,k-a_j) + \lambda_j + \theta[h(j-1,k-a_j) + d_j].$$

The value

$$\overline{\theta} = [g(j-1,k) - g(j-1,k-a_j) - \lambda_j]/[h(j-1,k-a_j) + d_j - h(j-1,k)]$$

```
Given:
    —a knapsack polyhedron $\mathcal{P}_F^i$
    —a linear function $\lambda x$ to maximize on $\mathcal{P}_F^i$
    —a direction $d$ for parametrically altering $\lambda$

Initialize:
    $g(0,k) = 0$
    $h(0,k) = 0$
    $pr(0,k) = 0$
    $\theta_{\max} = \infty$

Recursive Relation:
    if $g(j-1,k) > g(j-1,k-a_j) + \lambda_j$ then
        $g(j,k) = g(j-1,k)$
        $h(j,k) = h(j-1,k)$
        $pr(j,k) = (j-1,k)$
        $\theta = [g(j-1,k) - g(j-1,k-a_j) - \lambda_j]/[h(j-1,k-a_j) + d_j - h(j-1,k)]$
        if $\theta > 0$ then $\theta_{\max} = \min\{\theta_{\max}, \theta\}$
    else if $g(j-1,k) < g(j-1,k-a_j) + \lambda_j$ then
        $g(j,k) = g(j-1,k-a_j) + \lambda_j$
        $h(j,k) = h(j-1,k-a_j) + d_j$
        $pr(j,k) = (j-1,k-a_j)$
        $\theta = [g(j-1,k) - g(j-1,k-a_j) - \lambda_j]/[h(j-1,k-a_j) + d_j - h(j-1,k)]$
        if $\theta > 0$ then $\theta_{\max} = \min\{\theta_{\max}, \theta\}$
    else if $h(j-1,k) > h(j-1,k-a_j) + d_j$ then
        $g(j,k) = g(j-1,k)$
        $h(j,k) = h(j-1,k)$
        $pr(j,k) = (j-1,k)$
    else
        $g(j,k) = g(j-1,k-a_j) + \lambda_j$
        $h(j,k) = h(j-1,k-a_j) + d_j$
        $pr(j,k) = (j-1,k-a_j)$
```

FIG. 2. *Recursive relation for optimizing* $(\lambda + \theta d)x$ *on* $\mathcal{P}_F^i$.

satisfies these expressions at equality, and it follows that in either case the largest that θ can be made without violating the appropriate inequality is $\bar{\theta}$ if $\bar{\theta} > 0$ and ∞ if $\bar{\theta} < 0$. When $g(j-1,k) \neq g(j-1,k-a_j) + \lambda_j$ it is not possible for $\bar{\theta}$ to be 0 so the maximum allowable increase in θ must be strictly positive in each case. When $g(j-1,k) = g(j-1,k-a_j) + \lambda_j$ the expressions reduce to

$$\theta h(j-1,k) \geq \theta[h(j-1,k-a_j) + d_j]$$

and

$$\theta h(j-1,k) \leq \theta[h(j-1,k-a_j) + d_j].$$

Alternatively stated, when $g(j-1,k) = g(j-1,k-a_j) + \lambda_j$ then the optimal predecessor array $\text{pr}(j,k)$ remains optimal for any $\theta \geq 0$ if and only if for each (j,k) with $\text{pr}(j,k) = (j-1,k)$ it is true that

$$h(j-1,k) \geq h(j-1,k-a_j) + d_j,$$

and for each (j,k) with $\text{pr}(j,k) = (j-1,k-a_j)$ it is true that

$$h(j-1,k) \leq h(j-1,k-a_j) + d_j.$$

It can be seen that when $g(j-1,k) = g(j-1,k-a_j) + \lambda_j$ the modified recursive relation of Fig. 2 chooses the predecessor of (j,k) to satisfy these last conditions and that $\theta_{\max} > 0$ represents the maximum allowable increase in θ defined by pairs (j,k) for which $g(j-1,k) \neq g(j-1,k-a_j)+\lambda_j$. The desired result follows. □

The importance of Theorem 3.3 is that it can be used to show how the recursive relation presented in Fig. 2 solves Problem FPARAM.

THEOREM 3.4. *Problem FPARAM associated with $\mathcal{P}_F^i$ can be solved using at most a finite number of evaluations of the recursive relation presented in Fig.* 2.

Proof. Consider the X constraint $z \leq \lambda(\hat{x}-\overline{x})$ where $\overline{x}$ maximizes $\overline{\lambda}x$ on $\mathcal{P}_F^i$ and is determined by applying the recursive relation presented in Fig. 2. Clearly, this constraint is a candidate to be in the set S_X associated with the point $[\overline{\lambda},\overline{z}]$ since $\overline{z}=v(\overline{\lambda})=\overline{\lambda}(\hat{x}-\overline{x})$ is maintained throughout the course of the algorithm. It can be determined if $[\overline{\lambda},\overline{z}]+\theta_{\max}[d,1]$ satisfies this constraint by determining if $\overline{z}+\theta_{\max} \leq (\overline{\lambda}+\theta_{\max}d)(\hat{x}-\overline{x}) = v(\overline{\lambda}+\theta_{\max}d)$. If not, that is, if $\overline{z}+\theta_{\max} > v(\overline{\lambda}+\theta_{\max}d)$, then since $[\overline{\lambda},\overline{z}]+\theta_{\max}[d,1]$ satisfies all of the S_X constraints for all $\theta \geq 0$ by assumption it follows that the X constraint associated with $\overline{x}$ is not in S_X and therefore the value $\theta=0$, together with this X constraint, represent a solution to Problem FPARAM.

Thus, suppose $\overline{z}+\theta_{\max} \leq v(\overline{\lambda}+\theta_{\max}d)$ so that $[\overline{\lambda},\overline{z}]+\theta[d,1]$ satisfies all of the X constraints for $0 \leq \theta \leq \theta_{\max}$. Let $\theta_{\max}^{\Lambda}$ be the largest value of $\theta \geq 0$ such that $\overline{\lambda}+\theta d$ satisfies all of the Λ constraints. All of the constraints in S_Λ must be satisfied for any $\theta \geq 0$ by assumption and thus $\theta_{\max}^{\Lambda}$ can be determined by finding when $\lambda+\theta d$ first violates each of the remaining constraints and taking the minimum of these values. If $\theta_{\max}^{\Lambda} \leq \theta_{\max}$ then $\theta=\theta_{\max}^{\Lambda}$, together with the Λ constraint that defined $\theta_{\max}^{\Lambda}$, represent a solution to Problem FPARAM.

Thus, suppose that $\theta_{\max}^{\Lambda} > \theta_{\max}$ and let $\overline{\lambda}' = \overline{\lambda}+\theta_{\max}d$. By Theorem 3.3 there exists a $\theta'_{\max} > 0$ such that the predecessor array $\operatorname{pr}(j,k)$ remains optimal when $\overline{\lambda}'+\theta d$ is maximized on $\mathcal{P}_F^i$ and so the process just described can be repeated using this new value of λ. Finite termination of the procedure is ensured by recognizing that $\theta_{\max}$ is chosen so that applying the recursive relation at $\overline{\lambda}'$ corresponds to a nondegenerate parametric simplex pivot. □

4. Computational results. The algorithm described in the previous section was tested by applying it to the collection of 0/1 integer programs studied by Crowder, Johnson, and Padberg in [6]. A summary of these problems is shown in Table 1. Problems from this test set were chosen for a number of reasons. First, they represent a collection of real-world problems and thus exhibit characteristics that are not generally exhibited by randomly generated problems. Second, the problems have been solved to optimality and thus it is possible to measure how much of the gap has been closed between the optimal value of the original integer program and its linear programming relaxation. Finally, the problems are rapidly becoming a standard test set of integer programs. Crowder, Johnson, and Padberg argued that under the assumption that an integer program was sparse the polyhedron $\mathcal{P}$ defined by the convex hull of feasible integer points often would be reasonably well approximated by $\bigcap_{i=1}^{m}\mathcal{P}_F^i$. This claim was strongly supported by their computational results.

The algorithm in which the cut generation algorithm of the previous section was embedded proceeded as follows. The linear programming relaxation of a given integer program was first solved to obtain an optimal solution $\hat{x}$. A pass was then made through the constraints, during which a Fenchel cut was sought for each of the subproblem polyhedra $\mathcal{P}_F^i$. Any Fenchel cuts that were found were then appended to the original problem formulation and the process was repeated. On subsequent passes not every polyhedron $\mathcal{P}_F^i$ was examined for a Fenchel cut since some of these polyhedra were clearly not defining active constraints near the optimal solution. However, in

TABLE 1
Summary of problems.

Name	Variables	Constraints	Nonzeros	v_{LP}	v_{IP}
P0033	33	15	98	2520.6	3089.0
P0040	40	23	110	61796.5	62027.0
P0201	201	133	1923	6875.0	7615.0
P0282	282	241	1966	176867.5	258411.0
P0291	291	252	2031	1705.1	5223.8
P0548	548	176	1711	315.3	8691.0
P2756	2756	755	8937	2688.7	3124.0

TABLE 2
Cut summary using generalized programming.

Name	$\Delta Gap^{1.0}$	$\Delta Gap^{2.0}$	$\Delta Gap^{3.0}$	ΔGap^{T}	v_{LP}^{T}	T†	Cuts
P0033	—	—	—	87.42%	3017.50	.14	53
P0040	—	—	—	100.00%	62027.00	.01	4
P0201	27.03%	33.78%	—	33.78%	7125.00	2.88	30
P0282	89.36%	94.39%	96.14%	98.59%	257261.97	20.70	466
P0291	74.83%	76.91%	95.24%	99.43%	5203.87	16.70	142
P0548	51.27%	75.00%	82.81%	84.34%	7379.28	4.16	320
P2756	2.47%	3.00%	3.00%	86.16%	3063.75	22.05	509

†minutes on an IBM RISC Station 550

every problem except problem P2756 the algorithm did not terminate until a pass had been made in which it was demonstrated that no cutting plane existed for any polyhedron $\mathcal{P}_F^i$. Thus, in these instances the algorithm provided a *proof* that the most recently determined optimal solution $\hat{x}$ for the linear programming relaxation of the problem was optimal over $\bigcap_{i=1}^m \mathcal{P}_F^i$. Initially it was not clear that this would be achievable within reasonable computation times, since the polyhedra $\mathcal{P}_F^i$ are associated with NP-complete problems. The algorithm used by Crowder, Johnson, and Padberg for generating cuts, even if solved exactly rather than heuristically as is done in [6], does not provide a proof of optimality over $\bigcap_{i=1}^m \mathcal{P}_F^i$.

Computational results for the algorithm described above are shown in Tables 2, 3, 4, and 5. Each table corresponds to a different way of solving (G) associated with cutting plane generation. The basic domain for which all of the results are generated is

$$\{\lambda : \begin{array}{l} \lambda \in \Lambda \\ \lambda \geq 0 \\ \lambda_i = 0 \quad i \in S_0\}, \end{array}$$

where $S_0 = \{i : \hat{x}_i = 0\}$ and

$$\Lambda = \left\{ \lambda : \sum_{i=1}^{n} \lambda_i \leq \beta, \ 0 \leq \lambda \leq 1 \right\},$$

with β equal to 0.5 or $n + 1$. The constraints $\lambda_i = 0$, $i \in S_0$ were included from the outset since without this restriction the generalized programming algorithm was so slow that for practical purposes it was incapable of solving (G). The results in Table 2 correspond to solving (G) using generalized programming on the basic domain, while in Table 3 (G) is solved using the ascent algorithm described in §3.2. The results in Table 4 correspond to using the ascent algorithm of §3.2 together with the additional domain restriction that $\lambda_i = \lambda_j$ whenever $\hat{x}_i = \hat{x}_j = 1$, as described in §3.1. Finally,

TABLE 3
Cut summary using ascent algorithm.

Name	$\Delta Gap^{1.0}$	$\Delta Gap^{2.0}$	$\Delta Gap^{3.0}$	ΔGap^{T}	v_{LP}^{T}	T†	Cuts
P0033	—	—	—	87.42%	3017.50	.04	65
P0040	—	—	—	100.00%	62027.00	.01	4
P0201	—	—	—	33.78%	7125.00	.08	30
P0282	96.62%	97.14%	97.52%	98.59%	257261.97	9.53	524
P0291	97.37%	98.87%	99.15%	99.43%	5203.87	5.63	149
P0548	82.09%	—	—	84.34%	7379.28	1.58	377
P2756	2.47%	3.00%	3.00%	86.16%	3063.75	16.78	483

†minutes on an IBM RISC Station 550

TABLE 4
Cut summary using ascent algorithm with domain restrictions.

Name	$\Delta Gap^{1.0}$	$\Delta Gap^{2.0}$	$\Delta Gap^{3.0}$	ΔGap^{T}	v_{LP}^{T}	T†	Cuts
P0033	—	—	—	87.42%	3017.50	.04	62
P0040	—	—	—	100.00%	62027.00	.01	4
P0201	—	—	—	33.78%	7125.00	.16	54
P0282	97.06%	97.86%	98.32%	98.59%	257261.97	4.61	688
P0291	99.27%	—	—	99.43%	5203.87	1.53	269
P0548	—	—	—	84.34%	7379.28	.51	547
P2756	—	—	—	86.16%	3063.75	.74	621

†minutes on an IBM RISC Station 550

TABLE 5
Cut summary using ascent algorithm, domain restrictions, and variable fixing.

Name	$\Delta Gap^{1.0}$	$\Delta Gap^{2.0}$	$\Delta Gap^{3.0}$	ΔGap^{T}	v_{LP}^{T}	T†	Cuts
P0033	—	—	—	87.42%	3017.50	.02	54
P0040	—	—	—	100.00%	62027.00	.01	2
P0201	—	—	—	33.78%	7125.00	.31	50
P0282	—	—	—	98.59%	257261.97	.31	371
P0291	—	—	—	99.44%	5204.17	.13	130
P0548	—	—	—	84.34%	7379.28	.55	430
P2756	—	—	—	86.16%	3063.75	.53	432

†minutes on an IBM RISC Station 550

in Table 5 results are given for the same algorithm and domain as in Table 4, but in addition some very basic variable fixing techniques based on reduced cost information are also used. While we do not discuss these techniques here, this last table is included to demonstrate the best times achieved by the author while provably optimizing over $\bigcap_{i=1}^{m} \mathcal{P}_F^i$.

In each of these tables, the column labeled Cuts is the total number of cuts appended to the problem. The columns labeled $\Delta Gap^{1.0}$, $\Delta Gap^{2.0}$, and $\Delta Gap^{3.0}$ represent the percentage by which the gap between v_{LP} and v_{IP} was reduced in 1, 2, and 3 minutes, respectively, with ΔGap^{T} representing the percentage by which this gap was closed in T minutes, where T is given in the table. In addition, the column v_{LP}^{T} gives the value of the linear programming relaxation after T minutes. For all of the problems except P2756 the values v_{LP}^{T} represent the provably best gap reduction that can be achieved using only cutting planes associated with the individual knapsack constraints. This result was not achieved for problem P2756 since two of the constraints were large enough that $v(\lambda)$ could not be maximized in reasonable computation times with the existing ascent algorithm. All computational tests were performed on an IBM RISC Station 550.

As can be seen, the use of the ascent algorithm and the additional domain restrictions introduced in Table 4 both had a significant impact on the running time of the algorithm. As would be expected, the number of cutting planes found in Tables 2 and 3 are roughly the same, but there is an increase in the number of cutting planes in Table 4. In spite of this, the running times reported in Table 4 are faster due to the smaller dimension of the problems being solved to generate these cutting planes. The reason for the increased number of cuts is that the domain restriction introduced for the results in Table 4 nullifies any guarantee that a maximizing value of $v(\lambda)$ will be found on the basic domain and thus that a "deepest" cut will be generated; the guarantee is only that a positive value of $v(\lambda)$ will be found on the restricted domain if a positive value of $v(\lambda)$ exists on the basic domain. While domain restrictions are seemingly innocuous on the surface, if used without care they can lead to very poor results. In early computational experiments, all λ_i with $\hat{x}_i = 1$ were set equal to a fixed constant value, and it can be shown that in doing so a positive value of $v(\lambda)$ is guaranteed to be found if one exists. However, fixing the value of the λ_i in this way leads to a very undesirable scaling of the coefficients in the associated cutting plane, and computational experiments demonstrated that such cutting planes were very weak. The results in Table 4, however, show that the domain restriction $\lambda_i = \lambda_j$ when $\hat{x}_i = \hat{x}_j$ can be quite effective. These results illustrate the fundamental importance of domain issues, and many theoretical aspects of the choice of domain are discussed in detail in [5].

One issue that proved to be a significant obstacle to the speed of the ascent algorithm was degeneracy. Degeneracy expresses itself as a value of 0 returned by the algorithm for solving Problem FPARAM. Common wisdom is that degeneracy is best dealt with simply by ignoring it. However, for the functions $v(\lambda)$ associated with the $\mathcal{P}_F^i$ of the test problems this proved to be a very bad idea in some instances. Long sequences of degenerate pivots were sometimes observed that severely impeded the progress of the algorithm.

The perturbation method and Bland's anticycling rule were both considered as potential ways of overcoming the problem of degeneracy. Beyond the practical inefficiency of these two approaches they both require that the active constraints at the degenerate vertex be known explicitly and this is not the case for (G). While it is possible to use these techniques while simultaneously refining the known collection of active constraints at the vertex, this requires continually adding constraints to the set S_X returned from the function for solving Problem FPARAM. This works against the effort to maintain a small set S_X and the correspondingly small amount of work required to complete step 2 of the algorithm.

To overcome the problem of degeneracy it was decided to choose the steepest edge in step 2 of the algorithm rather than an arbitrary ascent direction; specifically, the direction of ascent $[d, 1]$ was chosen as the edge that maximized $1/\|[d, 1]\|$. Recent evidence supporting the practical efficiency of steepest edge pivoting rules can be found in [2] and [8], and recent theoretical results on the use of steepest edge pivoting rules to resolve degeneracy can be found in [4]. The computational expense associated with this approach, of course, is that the edge of ascent associated with each variable of positive reduced cost must be calculated, and finding each edge entails solving a square linear system in $|S_X|$ variables ($|S_X| + 1$ if the β constraint is included in (H)). If only reduced cost information is used to choose the entering variable and the associated edge of ascent, then only one linear system must be solved beyond the system solved to calculate the dual variables. In practice, while steepest edge was much better at resolving degeneracy it was also so computationally expensive that it was only used after a sufficiently long sequence of degenerate pivots was encountered, that is, after a sufficiently long sequence of 0's was returned by the algorithm for solving Problem FPARAM. This strategy of using a degeneracy mechanism only when needed proved to be the best strategy by far.

The generalized programming algorithm used to generate the results presented in Table 2 proceeded as follows. The problem (G) was approximated using all of the Λ constraints and some subset of the X constraints; initially, one arbitrarily chosen X constraint was used. The resultant linear programming approximation to (G) was then solved to obtain a vector $[\overline{\lambda}, \overline{z}]$. The linear function $\overline{\lambda}x$ was then maximized on $\mathcal{P}_F^i$ to determine $v(\overline{\lambda})$. If $v(\overline{\lambda}) = \overline{z}$ it follows that $[\overline{\lambda}, \overline{z}]$ is feasible and optimal for a relaxation of (G) and must therefore be optimal for (G). In this case the algorithm terminates with the Fenchel cut $\overline{\lambda}x \leq f(\overline{\lambda})$. If $v(\overline{\lambda}) < \overline{z}$ the X constraint associated with the $x^i \in E(\mathcal{P}_F^i)$ satisfying $\overline{\lambda}x^i = v(\overline{\lambda})$ is violated by $[\overline{\lambda}, \overline{z}]$. In this case, the new X constraint associated with x^i is included in the approximation of (G) and a new optimal solution to this approximation is sought.

A number of modifications were made to the basic generalized programming algorithm in an effort to improve its performance. A simpler version of the dynamic programming algorithm for maximizing $v(\lambda)$ was required for the generalized programming algorithm than for the ascent algorithm since generalized programming requires only $v(\lambda)$ and no parametric information. Constraints were selectively dropped from the approximation to (G) when they had not been active at the optimal solution for a fixed number of iterations. This kept the size of the approximation to (G) relatively small without affecting the number of iterations required to maximize (G) and as a consequence reduced the time spent in the linear programming routines used to solve the approximation to (G). Finally, a version of the CPLEX callable library that makes use of the dual simplex algorithm was used to solve the sequence of linear programs encountered as successive constraints were appended to the generalized program.

With the modifications just described the generalized programming algorithm was extremely efficient. Significant time was spent developing this algorithm before we felt that the potential of the generalized programming approach had been reached. In fact, it was the limitations of this approach that led us to develop the ascent algorithm described in this paper. To fully appreciate the improvement of the ascent algorithm over the generalized programming algorithm it is necessary to recognize that the generalized programming algorithm represented an extremely efficient implementation using a state-of-the-art linear programming code. The difference in running times between the two algorithms thus represents a fundmental difference in the algorithms themselves and not their implementations. In fact, there remain areas of potential improvement for the ascent algorithm that have not been investigated.

Initially a major effort was made to develop an ascent algorithm based on subgradients, but subgradient techniques were abandoned when they proved hopelessly inadequate. Even when a reasonable sequence of iterates λ^i could be generated—and this in itself was not always easy to accomplish—the empirical rate of convergence was generally so poor that the algorithm was useless for all practical purposes. Subgradient algorithms are so intuitive, simple to code, and generally well regarded in the literature that it took us a long time to realize how inadequate they can be. However, it was a full appreciation of the information that subgradient techniques disregard—namely, an intelligent choice of ascent direction—that led to the algorithm presented in this paper.

5. Conclusions. An algorithm for efficiently generating Fenchel cuts for knapsack polyhedra has been presented. Two main aspects of this algorithm were highlighted that were profoundly important for its efficient implementation. First, the algorithm was based on an intelligent ascent procedure rather than subgradients or generalized programming. Second, domain restrictions were outlined showing that the dual maximization problem necessary to generate Fenchel cuts could be performed in a space of reduced dimension. An implementation of this algorithm was then applied to generate cutting planes for a collection of integer programs and to provably optimize over the intersection of the knapsack polyhedra defined by the constraints of these problems.

The ascent algorithm presented in this paper achieved good performance by taking advantage of the underlying combinatorial structure of the subproblem defining the dual maximization problem, rather than disregarding this fundamental information, as alternative techniques do. It is more than conceivable that for some subproblems there exist efficient combinatorial methods for choosing the ascent direction in step 2 of the ascent algorithm, or even efficient combinatorial methods for solving the dual maximization problem itself. The types of combinatorial problems that arise in this context deserve further attention.

While this paper has focused on the specific problem of generating Fenchel cuts for knapsack polyhedra, it is important to recognize that most of the ideas are directly applicable in other contexts. The ascent algorithm can be applied whenever an algorithm for solving Problem FPARAM exists, and domain restrictions similar to those discussed in this paper should be derivable for many classes of problems. Further, the ascent algorithm can be used to solve dual problems that arise in applications of Lagrangian relaxation whenever an algorithm for solving Problem FPARAM exists. Finally, while the implementation is more complex, the basic ideas presented in this paper are directly applicable to the separation problem for mixed-integer knapsack polyhedra. This is an important direction for further research.

6. Acknowledgments. The author would like to acknowledge the Computational Optimization Center in the Department of Industrial and Systems Engineering at Georgia Tech for making computational resources available for the results presented in this paper.

REFERENCES

[1] D. Bertsekas, *Nonlinear programming and discrete-time optimal control*, Tech. Rep. LIDS-R-919, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1979.

[2] R. E. Bixby, Unpublished computational results, Dept. of Mathematical Sciences, Rice Univ., Houston, TX, 1991.

[3] E. A. Boyd, *Fenchel cutting planes for integer programs*, Oper. Res., to appear.

[4] ———, *Resolving degeneracy in combinatorial linear programs: Steepest edge, steepest ascent, and parametric ascent*, Tech. Report TR91-21, Dept. of Mathematical Sciences, Rice Univ., Houston, TX, 1991.

[5] ———, *On the convergence of Fenchel cutting planes in mixed-integer programming*, SIAM J. Optimization, to appear.

[6] H. Crowder, E. L. Johnson, and M. W. Padberg, *Solving large-scale zero-one linear programming problems*, Oper. Res., 31 (1983), pp. 803–834.

[7] M. L. Fisher, *The lagrangian relaxation method for solving integer programming problems*, Management Sci., 27 (1981), pp. 1–18.

[8] J. Forrest and D. Goldfarb, *Steepest edge simplex algorithms for linear programming*, IBM Res. Rep., T. J. Watson Research Center, Yorktown Heights, NY, 1991.

[9] A. M. Geoffrion, *Lagrangian relaxation and its uses in integer programming*, Math. Programming Stud., 2 (1974), pp. 82–114.

[10] R. E. Gomory, *Outline of an algorithm for integer solutions to linear programs*, Bull. AMS, 64 (1958), pp. 275–278.

[11] M. Grötschel, *On the symmetric travelling salesman problem: Solution of a* 120*-city problem* Math. Programming Stud., 12 (1980), pp. 61–77.

[12] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, New York, 1988.

[13] M. Guignard and G. Plateau, *Lagrangean decomposition applied to* 0/1 *bi knapsack problems*, Presentation at the Spring 1989 ORSA/TIMS conference, Vancouver, British Columbia, Canada.

[14] M. Held and R. M. Karp, *The traveling salesman problem and minimal spanning trees*, Oper. Res., 18 (1970), pp. 1138–1162.

[15] ———, *The traveling salesman problem and minimal spanning trees: Part* II, Math. Programming, 1 (1971), pp. 6–25.

[16] K. O. Jornsten and M. Nasberg, *A new lagrangian relaxation approach to the generalized assignment problem*, European J. Oper. Res., 27 (1986), pp. 313–323.

[17] C. Lemaréchal, *Nondifferentiable optimization*, in Optimization, G. L. Nemhauser et. al., Handbooks in Operations Research and Management Science, 1, North Holland, New York, 1989, pp. 529–572.

[18] O. L. Mangasarian, *Linear and nonlinear separation of patterns by linear programming*, Oper. Res., 13 (1965), pp. 444–452.

[19] G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization*, Wiley, New York, 1988.

[20] M. Padberg and G. Rinaldi, *Optimization of a* 532*-city traveling salesman problem by branch and cut*, Oper. Res. Lett., 6 (1987), pp. 1–7.

[21] C. Ribeiro and M. Minoux, *Solving hard constrained shortest path problems by lagrangian relaxation and branch and bound algorithms*, in Proceedings of the X Symposium on Operations Research, M. Beckmann et. al., eds., Methods of Operations Research, 53 (1985), pp. 303–316.

[22] M. W. P. Savelsbergh, G. C. Sigismondi, and G. L. Nemhauser, *Functional description of MINTO: A Mixed INTeger Optimizer*, Tech. Rep. COC 91-03, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1991.

[23] M. W. P. Savelsbergh, G. C. Sigismondi, and G. L. Nemhauser, *MINTO: A Mixed INTeger Optimizer*, Tech. Rep. COC 91-04, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1991.

[24] J. F. Shapiro, *Generalized lagrange multipliers in integer programming*, Oper. Res., 19 (1971), pp. 68–76.

[25] ———, *Mathematical programming: Structures and algorithms*, Wiley, New York, 1979.

[26] M. Trick, *Networks with additional structured constraints*, Ph.D. thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1987.

[27] P. Wolfe, *Finding the nearest point in a polytope*, Math. Programming, 11 (1976), pp. 128–149.

SIAM J. OPTIMIZATION
Vol. 3, No. 4, pp. 751–783, November 1993

006

PRIMAL-DUAL PROJECTED GRADIENT ALGORITHMS FOR EXTENDED LINEAR-QUADRATIC PROGRAMMING*

CIYOU ZHU† AND R. T. ROCKAFELLAR‡

Abstract. Many large-scale problems in dynamic and stochastic optimization can be modeled with extended linear-quadratic programming, which admits penalty terms and treats them through duality. In general, the objective functions in such problems are only piecewise smooth and must be minimized or maximized relative to polyhedral sets of high dimensionality. This paper proposes a new class of numerical methods for "fully quadratic" problems within this framework, which exhibit second-order nonsmoothness. These methods, combining the idea of finite-envelope representation with that of modified gradient projection, work with local structure in the primal and dual problems simultaneously, feeding information back and forth to trigger advantageous restarts.

Versions resembling steepest descent methods and conjugate gradient methods are presented. When a positive threshold of ε-optimality is specified, both methods converge in a finite number of iterations. With threshold 0, it is shown under mild assumptions that the steepest descent version converges linearly, while the conjugate gradient version still has a finite termination property. The algorithms are designed to exploit features of primal and dual decomposability of the Lagrangian, which are typically available in a large-scale setting, and they are open to considerable parallelization.

Key words. extended linear-quadratic programming, large-scale numerical optimization, finite-envelope representation, gradient projection, primal-dual methods, steepest descent methods, conjugate gradient methods

AMS subject classifications. 65K05, 65K10, 90C20

1. Introduction.

A number of recent papers have described "extended linear-quadratic programming" as a modeling scheme that is much more flexible for problems of optimization than conventional quadratic programming and that seems especially suited to large-scale applications, in particular because of the way penalty terms can be incorporated. Rockafellar and Wets [1], [2] first used the concept in two-stage stochastic programming, where the primal dimension is low but the dual dimension is high. It was developed further in its own right in Rockafellar [3], [4], and carried in the latter paper into the context of continuous-time optimal control. Discrete-time problems of optimal control, both deterministic and stochastic (i.e., multistage stochastic programming) were analyzed as extended linear-quadratic programming problems in Rockafellar and Wets [5] and were shown to have a remarkable property of Lagrangian decomposability in the primal and dual arguments, both of which can be high-dimensional. These models raise new computational challenges and possibilities.

A foundation for numerical schemes in large-scale extended linear-quadratic programming has been laid by Rockafellar in [6] and elaborated upon for problems in multistage format in [7]. The emphasis in [6] is on basic *finite-envelope methods*, which use duality in generating envelope approximations to the primal and dual objective functions through a finite sequence of separate minimizations or maximizations of the

*Received by the editors January 11, 1991; accepted for publication (in revised form) July 25, 1992. This work was supported in part by Air Force Office of Scientific Research grants 87-0281 and 89-0081 and National Science Foundation grants DMS-8701768 and DMS-8819586.

†Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, Maryland 21218.

‡Department of Applied Mathematics, FS-20, University of Washington, Seattle, Washington 98195.

Lagrangian. These methods generalize the one originally proposed in [1] for two-stage stochastic programming and implemented by King [8] and Wagner [9]. They center on the "fully quadratic" case, where strong convexity is present in both the primal and dual objectives, relying on exterior schemes, such as the proximal point algorithm, to create such strong convexity iteratively when it might otherwise be lacking.

Here we propose new algorithms which for fully quadratic problems combine the idea of finite-envelope representation with that of nonlinear gradient projection. In these methods the envelope approximations are utilized in a sort of steepest descent or conjugate gradient format in the primal and dual problems simultaneously. A type of feedback is introduced between primal and dual that takes advantage of information jointly uncovered in computations, which in practice greatly speeds convergence. Both algorithms fit into a fundamental scheme for which global convergence is established. Under a weak geometric assumption akin to strict complementary slackness at optimality, the steepest descent version is shown to converge at a linear rate, while the conjugate gradient version has a finite termination property.

Both versions differ significantly from their traditional namesakes not only through the incorporation of a primal-dual scheme of gradient projection, but also in handling objective functions that generally could involve a complicated polyhedral "cell" structure not conducive to explicit description by linear equations and inequalities. They treat the underlying constraints without resorting to an active set strategy, which would not be suitable for problems having high dimensionality in both primal and dual.

An important feature is that the computations are not carried out in terms of a large, sparse matrix, such as might in principle serve in part to specify the two problems, but through subroutines for separate minimization and maximization of the Lagrangian in its primal and dual arguments. This framework appears much better adapted to the special structure available in dynamic and stochastic applications, and it supports extensive parallelization. To make this point clearer, and to introduce facts and notation that will later be needed, we discuss briefly the nature of extended linear-quadratic programming and the way it differs from ordinary quadratic programming.

From the Lagrangian point of view, extended linear-quadratic programming is directed toward finding a saddle point $(\bar{u}, \bar{v})$ of a function

$$L(u,v) = p{\cdot}u + \tfrac{1}{2}u{\cdot}Pu + q{\cdot}v - \tfrac{1}{2}v{\cdot}Qv - v{\cdot}Ru \quad \text{over} \quad U \times V, \tag{1.1}$$

where U and V are nonempty polyhedral (convex) sets in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, and the matrices $P \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{m\times m}$ are symmetric and positive semidefinite. (One has $p \in \mathbb{R}^n$, $q \in \mathbb{R}^m$, and $R \in \mathbb{R}^{m\times n}$.) Associated with L, U, and V are the primal and dual problems

$$(\mathcal{P}) \qquad \text{minimize } f(u) \text{ over all } u \in U, \text{ where } f(u) := \sup_{v\in V} L(u,v),$$

$$(\mathcal{Q}) \qquad \text{maximize } g(v) \text{ over all } v \in V, \text{ where } g(v) := \inf_{u\in U} L(u,v).$$

We speak of the *fully quadratic* case of $(\mathcal{P})$ and $(\mathcal{Q})$ when both of the matrices P and Q are actually positive definite.

Standard quadratic programming would correspond to $Q = 0$ and $V = \mathbb{R}^{m_1}_{+} \times \mathbb{R}^{m_2}$. Then f would consist of a quadratic function plus the indicator of a system of m_1 linear inequality constraints and m_2 linear equations, the indicator being the function which assigns an infinite penalty whenever these constraints are violated.

Other choices of Q and V yield *finite* penalty expressions of various kinds. This is explained in [4, §§2 and 3] with many examples. For sound modeling in large-scale applications with dynamics and stochastics such as in [1], [2], and [5], it appears wise to use finite rather than infinite penalties whenever constraints are "soft." Extended linear-quadratic programming makes this option conveniently available. To the extent that constraints in the primal problem are "hard," they can be handled either by placing them in the definition of the polyhedron U or through an augmented Lagrangian technique which corresponds to an exterior scheme of iterations of the proximal point algorithm, as already mentioned.

THEOREM 1.1 [4] (properties of the objective functions). *The objective functions f in $(\mathcal{P})$ and g in $(\mathcal{Q})$ are piecewise linear-quadratic: in each case the space can be partitioned in principle into a finite collection of polyhedral cells, relative to which the function has a linear or quadratic formula. Moreover, f is convex while g is concave. In the fully quadratic case of $(\mathcal{P})$ and $(\mathcal{Q})$, f is strongly convex and g is strongly concave, both functions having continuous first derivatives.*

THEOREM 1.2 [4], [1] (duality and optimality). (a) *If either of the optimal values* $\inf(\mathcal{P})$ *or* $\sup(\mathcal{Q})$ *is finite, then both are finite and equal, in which event optimal solutions $\bar{u}$ and $\bar{v}$ exist for the two problems. In the fully quadratic case in particular, the optimal values* $\inf(\mathcal{P})$ *and* $\sup(\mathcal{Q})$ *are finite and equal; then, moreover, the optimal solutions $\bar{u}$ and $\bar{v}$ are unique.*

(b) *A pair $(\bar{u}, \bar{v})$ is a saddle point of $L(u, v)$ over $U \times V$ if and only if $\bar{u}$ solves $(\mathcal{P})$ and $\bar{v}$ solves $(\mathcal{Q})$, or equivalently, $f(\bar{u}) = g(\bar{v})$.*

Current numerical methods in standard quadratic programming, and the somewhat more general area of linear complementarity problems [10], where $U = \mathbb{R}^n_+$, $V = \mathbb{R}^m_+$, and Q is not necessarily the zero matrix, are surveyed by Lin and Pang [11]. Other efforts in recent times have been made by Ye and Tse [12], Monteiro and Adler [13], and Goldfarb and Liu [14].

None of these approaches is consonant with the large-scale applications that attract our interest, because the structure in such applications is not well served by the wholesale reformulations that would be required when penalty expressions are involved. Although any problem of extended linear-quadratic programming can in principle be recast as a standard problem in quadratic programming, as established in [1, Thm. 1], there is a substantial price to be paid in dimensionality and loss of symmetry, as well as in potential ill-conditioning. If the original problem had n primal and m dual variables, and the expression of U and V involved m' and n' constraints beyond nonnegativity of variables, then the reformulated problem in standard format would generally have $n + n' + m$ primal and $m + m'$ dual variables, and its full constraint system would tend to degeneracy (see [1, Proof of Thm. 1]). The dual problem would be quite different in its theoretical properties from the primal problem, so that computational ideas developed for the one could not be applied to the other.

Any problem of extended linear-quadratic programming can alternatively be posed in terms of solving a certain linear variational inequality (generalized equation) as explained in [6, Thm. 2.3], and from that one could pass to a linear complementarity model. Symmetry and the meaningful representation of dynamic and stochastic structure could be maintained to a larger extent in this manner. But linear complementarity algorithms tend to be less robust than methods utilizing objective function values, and an increase in dimensionality would still be required in handling constraints, even if these are simply upper and lower bounds on the variables. Furthermore, such algorithms typically have to be carried to completion. They do not generate sequences of primal-feasible and dual-feasible solutions along with estimates of how far these are from being optimal, which is highly desirable when problem size borders on the difficult.

While much could be said about the special problem structure in dynamic and stochastic applications [5], [7], it can be summarized for present purposes in the assertion that such problems, when formulated with care, satisfy the *double decomposability assumption* [6]. This means that for any fixed $u \in U$ it is relatively easy to maximize $L(u, v)$ over $v \in V$, and likewise, for any fixed $v \in V$ it is relatively easy to minimize $L(u, v)$ over $u \in U$, usually because of separability when either of the Lagrangian arguments is considered by itself. These subproblems of maximization and minimization calculate not only the objective values $f(u)$ and $g(v)$ but also, in the fully quadratic case where L is strongly convex-concave, the uniquely determined vectors

$$F(u) = \operatorname*{argmax}_{v \in V} L(u, v) \quad \text{and} \quad G(v) = \operatorname*{argmin}_{u \in U} L(u, v). \tag{1.2}$$

The issue is how to make use of such information in the design of numerical methods. Some proposals have already been made in Rockafellar [6]. Other ideas, which involve splitting algorithms, have been explored by Tseng [15], [16]. Here we aim at adapting classical descent algorithms with help from convex analysis [17].

In this paper we make the blanket assumption of double decomposability, taking it as license also for exact *line searchability* [6]: the supposition that it is possible to minimize $f(u)$ over any line segment joining two points in U, and likewise, to maximize $g(v)$ over any line segment joining two points in V. We focus on the fully quadratic case, even though standard quadratic programming is thereby excluded and a direct comparison with other computational approaches, apart from the finite-envelope methods in [6], becomes difficult. Our attention to that case is justified by its own potential in mathematical modeling (cf. [2] and [4]) and because strong convexity-concavity of the Lagrangian can be created, if need be, through some outer implementation of the proximal point algorithm [18], [19], as carried out in [1] and [8]. The questions concerning such an outer algorithm are best handled elsewhere, since they have a different character and relate to a host of primal-dual procedures in extended linear-quadratic programming besides the ones developed here; cf. [1], [2], and [6]. In particular, such questions are taken up in Zhu [20].

The supposition that line searches can be carried out exactly is an expedient to allow us to concentrate on more important matters for now. It is also in keeping with the exploration of finite termination properties of the kind usually associated with conjugate gradient-like algorithms, which is part of our agenda. One may observe also that because of the piecewise linear-quadratic nature of the objective functions in Theorem 1.1, line searches in our context are of a special kind where "exactness" is not far-fetched.

A common sort of problem structure which fits with double decomposability is the *box-diagonal case*, where P and Q are diagonal matrices,

$$P = \operatorname{diag}[\alpha_1, \ldots, \alpha_n] \quad \text{and} \quad Q = \operatorname{diag}[\beta_1, \ldots, \beta_m], \tag{1.3}$$

the entries α_j and β_i being positive (for fully quadratic problems), while U and V are boxes representing upper and lower bounds (not necessarily finite) on the components of $u = (u_1, \ldots, u_n)$ and $v = (v_1, \ldots, v_m)$:

$$U = [u_1^-, u_1^+] \times \cdots \times [u_n^-, u_n^+] \quad \text{and} \quad V = [v_1^-, v_1^+] \times \cdots \times [v_m^-, v_m^+]. \tag{1.4}$$

In this case, we have for each $u \in U$ that the problem of maximizing $L(u, v)$ over $v \in V$ to obtain $f(u)$ and $F(u)$ decomposes into separate *one-dimensional* subproblems in the individual coordinates: for $i = 1, \ldots, m$,

$$\text{(1.5)} \qquad \text{maximize } \left[q_i - \sum_{j=1}^{n} r_{ij} u_j\right] \cdot v_i - \frac{1}{2}\beta_i v_i^2 \text{ subject to } v_i^- \leq v_i \leq v_i^+.$$

Likewise, the problem of minimizing $L(u, v)$ over $u \in U$ for given $v \in V$, so as to calculate $g(v)$ and $G(v)$, reduces to the separate problems

$$\text{(1.6)} \qquad \text{minimize } \left[p_j - \sum_{i=1}^{m} v_i r_{ij}\right] \cdot u_i + \frac{1}{2}\alpha_j u_j^2 \text{ subject to } u_j^- \leq u_j \leq u_j^+.$$

Clearly, there exist very simple *closed-form* solutions to these one-dimensional subproblems. No actual minimization or maximization routine needs to be invoked. Often there are also ways of obtaining the answers without explicitly introducing the r_{ij}'s.

In notation, we shall refer consistently to

$$\text{(1.7)} \qquad \begin{aligned} \bar{u} &= \text{the unique optimal solution to } (\mathcal{P}), \\ \bar{v} &= \text{the unique optimal solution to } (\mathcal{Q}), \end{aligned}$$

these properties meaning by Theorem 1.2 that

$$\text{(1.8)} \qquad (\bar{u}, \bar{v}) = \text{the unique saddle point of } L \text{ on } U \times V,$$

or equivalently in terms of the mappings F and G that

$$\text{(1.9)} \qquad \bar{v} = F(\bar{u}) \quad \text{and} \quad \bar{u} = G(\bar{v}).$$

Furthermore, we shall write

$$\text{(1.10)} \qquad \begin{aligned} \|u\|_P &= [u \cdot Pu]^{\frac{1}{2}} \quad \text{and} \quad \|v\|_Q = [v \cdot Qv]^{\frac{1}{2}}, \\ \langle w, u \rangle_P &= w \cdot Pu \quad \text{and} \quad \langle z, v \rangle_Q = z \cdot Qv \end{aligned}$$

for the norms and inner products corresponding to the positive definite matrices P and Q. It is these norms and inner products, rather than the canonical ones, that intrinsically underlie the analysis of our problems, and it is good to bear this in mind. Just as the function f, if it is C^2 around a point u, can be expanded as

$$f(u') = f(u) + \langle \nabla f(u), u' - u \rangle + \tfrac{1}{2}\langle u' - u, \nabla^2 f(u)(u' - u) \rangle + o(\|u' - u\|^2),$$

it can also be expanded as

$$f(u') = f(u) + \langle \nabla_P f(u), u' - u \rangle_P + \tfrac{1}{2}\langle u' - u, \nabla_P^2 f(u)(u' - u) \rangle_P + o(\|u' - u\|_P^2)$$

for a certain vector $\nabla_P f(u)$ and a certain matrix $\nabla_P^2 f(u)$; similarly for g in terms of $\nabla_Q g(v)$ and $\nabla_Q^2 g(v)$. Clearly,

$$\text{(1.11)} \qquad \begin{aligned} \nabla_P f(u) &= P^{-1} \nabla f(u), \qquad \nabla_P^2 f(u) = P^{-1} \nabla^2 f(u), \\ \nabla_Q g(v) &= Q^{-1} \nabla g(v), \qquad \nabla_Q^2 g(v) = Q^{-1} \nabla^2 g(v). \end{aligned}$$

In appealing to this symbolism we shall be better able to bring out the basic structure and convergence properties of the proposed algorithms.

We now cite from [6] several fundamental properties on which the algorithmic developments in this paper will depend.

PROPOSITION 1.3 [6, p. 459] (optimality estimates). *Suppose $\hat{u}$ and $\hat{v}$ are elements of U and V satisfying $f(\hat{u})-g(\hat{v}) \leq \varepsilon$, where $\varepsilon \geq 0$. Then $\hat{u}$ and $\hat{v}$ are ε-optimal in the sense that $|f(\hat{u}) - f(\bar{u})| \leq \varepsilon$ and $|g(\hat{v}) - g(\bar{v})| \leq \varepsilon$. Moreover, $\|\hat{u} - \bar{u}\|_P \leq \sqrt{2\varepsilon}$ and $\|\hat{v} - \bar{v}\|_Q \leq \sqrt{2\varepsilon}$.*

PROPOSITION 1.4 [6, pp. 438, 469] (regularity properties). *The functions f and g are continuously differentiable everywhere, and the mappings F and G are Lipschitz continuous:*

$$\begin{aligned} \nabla f(u) &= \nabla_u L(u, F(u)) = p + Pu - R^T F(u), \\ \nabla g(v) &= \nabla_v L(G(v), v) = q - Qv - RG(v), \end{aligned} \tag{1.12}$$

where in terms of the constant

$$\gamma(P, Q, R) := \|Q^{-\frac{1}{2}} R P^{-\frac{1}{2}}\|, \tag{1.13}$$

one has

$$\begin{aligned} &\|F(u') - F(u)\|_Q \leq \gamma(P, Q, R)\|u' - u\|_P \text{ for all } u \text{ and } u', \\ &\|G(v') - G(v)\|_P \leq \gamma(P, Q, R)\|v' - v\|_Q \text{ for all } v \text{ and } v'. \end{aligned} \tag{1.14}$$

The finite-envelope idea enters through repeated application of the mappings F and G. The rationale is discussed at length in [6], but the main facts needed here are in the next two propositions.

PROPOSITION 1.5 [6, p. 460] (envelope properties). *For arbitrary $u_0 \in U$ and $v_0 \in V$, let $v_1 = F(u_0)$ and $u_1 = G(v_0)$, followed by $v_2 = F(u_1)$ and $u_2 = G(v_1)$. Then in the primal problem,*

$$\begin{aligned} &f(u) \geq L(u, v_1) \text{ for all } u, \text{ with } L(u_0, v_1) = f(u_0) \text{ and } \nabla_u L(u_0, v_1) = \nabla f(u_0), \\ &f(u) \geq L(u, v_2) \text{ for all } u, \text{ with } L(u_1, v_2) = f(u_1) \text{ and } \nabla_u L(u_1, v_2) = \nabla f(u_1), \end{aligned} \tag{1.15}$$

while in the dual problem

$$\begin{aligned} &g(v) \leq L(u_1, v) \text{ for all } v, \text{ with } L(u_1, v_0) = g(v_0) \text{ and } \nabla_v L(u_1, v_0) = \nabla g(v_0), \\ &g(v) \leq L(u_2, v) \text{ for all } v, \text{ with } L(u_2, v_1) = g(v_1) \text{ and } \nabla_v L(u_2, v_1) = \nabla g(v_1). \end{aligned} \tag{1.16}$$

PROPOSITION 1.6 [6, p. 470] (modified gradient projection). *For arbitrary $u_0 \in U$ and $v_0 \in V$, let $v_1 = F(u_0)$ and $u_1 = G(v_0)$, followed by $v_2 = F(u_1)$ and $u_2 = G(v_1)$. Then*

$$\begin{aligned} L(u, v_1) &= f(u_0) + \nabla f(u_0)\cdot(u - u_0) + \tfrac{1}{2}(u - u_0)\cdot P(u - u_0) \\ &= f(u_0) + \langle \nabla_P f(u_0), u - u_0 \rangle_P + \tfrac{1}{2}\|u - u_0\|_P^2, \\ &= \tfrac{1}{2}\left\|(u - u_0) + \nabla_P f(u_0)\right\|_P^2 + \text{ const.}, \end{aligned} \tag{1.17}$$

$$\begin{aligned} L(u_1, v) &= g(v_0) + \nabla g(v_0)\cdot(v - v_0) - \tfrac{1}{2}(v - v_0)\cdot Q(v - v_0) \\ &= g(v_0) + \langle \nabla_Q g(v_0), v - v_0 \rangle_Q - \tfrac{1}{2}\|v - v_0\|_Q^2, \\ &= -\tfrac{1}{2}\left\|(v - v_0) - \nabla_Q g(v_0)\right\|_Q^2 + \text{ const.}, \end{aligned} \tag{1.18}$$

so from the definition of u_2 and v_2 one has that

$$(1.19)\qquad \begin{aligned} u_2 - u_0 &= P\text{-}projection\ of\ -\nabla_P f(u_0)\ on\ U - u_0,\\ v_2 - v_0 &= Q\text{-}projection\ of\ \nabla_Q g(v_0)\ on\ V - v_0. \end{aligned}$$

Proof. The first equation in (1.17) expands $L(\cdot, v_1)$ at u_0 in accordance with (1.15), and the rest of (1.17) re-expresses this via (1.10) and (1.11). Since $u_2 := \operatorname{argmin}_{u \in U} L(u, v_1)$, u_2 is thus the $\|\cdot\|_P$-nearest point of U to $u_0 - \nabla_P f(u_0)$, so $u_2 - u_0$ is the $\|\cdot\|_P$-projection of $-\nabla_P f(u_0)$ on $U - u_0$. The assertions in the v-argument are verified similarly. □

The formulas in (1.19) give the precise form of (nonlinear) *gradient projection* that is available through our assumed ability to calculate $F(u)$ and $G(v)$ whenever we please. It is this form, therefore, that we shall incorporate in our algorithms. The reader should note this carefully, or a crucial feature of our approach, in its applicability to large-scale problems, will be missed. Although the gradients of f and g exist and are expressed by the formulas in Proposition 1.4, we do not have to calculate them through these formulas, much less apply a subroutine for gradient projection. In particular, *it is not necessary to generate or store the potentially huge or dense matrix R.* To execute our algorithms, one only needs to be able to generate the points u_1, u_2, v_1, and v_2 from a given pair u_0 and v_0. As explained, this can be done through subroutines which minimize or maximize the Lagrangian individually in the primal or dual argument; cf. (1.2). For multistage, possibly stochastic, optimization problems expressed in the format of [1], [2], and [6], such subroutines can easily be written in terms of the underlying data structure (without ever introducing R!).

In obtaining our results about local rates of convergence, a mild condition on the optimal solutions $\bar{u}$ and $\bar{v}$ will eventually be required. To formulate it, we introduce the sets

$$(1.20)\quad U_0 := \operatorname*{argmin}_{u \in U} \nabla_u L(\bar{u}, \bar{v})\cdot u = \operatorname*{argmin}_{u \in U} \nabla f(\bar{u})\cdot u = \operatorname*{argmin}_{u \in U} \langle \nabla_P f(\bar{u}), u \rangle_P,$$

$$(1.21)\quad V_0 := \operatorname*{argmax}_{v \in V} \nabla_v L(\bar{u}, \bar{v})\cdot v = \operatorname*{argmax}_{v \in V} \nabla g(\bar{v})\cdot v = \operatorname*{argmax}_{v \in V} \langle \nabla_Q g(\bar{v}), v \rangle_Q,$$

which are called the *critical faces* of U and V in $(\mathcal{P})$ and $(\mathcal{Q})$ [6]. They are closed faces of the polyhedral sets U and V, and they contain the optimal solutions $\bar{u}$ and $\bar{v}$, respectively, by virtue of the elementary conditions for the minimum of a smooth convex function (or the maximum of a smooth concave function).

DEFINITION 1.7 (critical face condition). The *critical face* condition will be said to be satisfied at the optimal solutions $\bar{u}$ and $\bar{v}$ if $\bar{u} \in \operatorname{ri} U_0$ and $\bar{v} \in \operatorname{ri} V_0$ (where "ri" denotes relative interior in the sense of convex analysis).

We do not add this condition as a standing assumption, but it will be invoked several times in connection with the following property of the envelope mappings F and G, which is implicit in [6, Thm. 5.4] in its proof, but is stated here explicitly.

PROPOSITION 1.8 (envelope behavior near the critical faces). *There exist neighborhoods of $\bar{u}$ and $\bar{v}$ with the property that if the points $u_0 \in U$ and $v_0 \in V$ belong to these neighborhoods, then the points*

$$v_1 = F(u_0), \qquad u_1 = G(v_0), \qquad v_2 = F(u_1), \qquad u_2 = G(v_1)$$

will be such that u_1 and u_2 belong to the primal critical face U_0, while v_1 and v_2 belong to the dual critical face V_0. Under the critical face condition, the neighborhoods can be chosen so that u_1 and u_2 actually belong to $\operatorname{ri} U_0$, *while v_1 and v_2 belong to* $\operatorname{ri} V_0$.

Proof. We adapt the argument given for [6, Thm. 5.4]. From (1.9) and the continuity of F and G in Proposition 1.4, we know that by making u_0 and v_0 close to $\bar{u}$ and $\bar{v}$ we will make u_1 and u_2 close to $\bar{u}$ and v_1 and v_2 close to $\bar{v}$. For each vector $w \in \mathbb{R}^n$, let $M(w)$ be the closed face of the polyhedron U on which the function $u \mapsto w{\cdot}u$ achieves its minimum. This could be empty for some choices of w, but in the case of $\bar{w} = \nabla_u L(\bar{u}, \bar{v})$ it is U_0, which contains $\bar{u}$. The graph of M as a set-valued mapping is closed (as can be verified directly or through the observation that M is the subdifferential of the support function of U; cf. [17, §§13, 23]), and M has only finitely many values (since U has only finitely many faces). It follows that $M(w) \subset M(\bar{w}) = U_0$ when w is in some neighborhood of $\bar{w}$. We can apply this in particular to $w = \nabla_u L(u_1, v_0)$, noting that this vector will be close to $\bar{w}$ when u_0 and v_0 are sufficiently close to $\bar{u}$ and $\bar{v}$. The point u_1 minimizes $L(u, v_0)$ over $u \in U$ and therefore has the property that $\nabla_u L(u_1, v_0){\cdot}(u - u_1) \leq 0$ for all $u \in U$, which means $u_1 \in M(w)$. Therefore, $u_1 \in U_0$ when u_0 and v_0 are sufficiently close to $\bar{u}$ and $\bar{v}$.

Parallel reasoning demonstrates that $v_1 \in V_0$ under such circumstances. If the critical face condition holds, then as u_1 and v_1 approach $\bar{u}$ and $\bar{v}$ they must actually enter the relative interiors $\operatorname{ri} U_0$ and $\operatorname{ri} V_0$. The same argument can be applied now to reach these conclusions for u_2 and v_2. □

2. Formulation of the algorithms.

The new methods for the fully quadratic case of problems $(\mathcal{P})$ and $(\mathcal{Q})$ will be formulated as conceptual algorithms involving line search. The convergence analysis will be undertaken in §§3, 4, and 5, and the numerical test results will be given in §6.

In what follows, we use $[w_1, w_2]$ to denote the line segment between two points w_1 and w_2, and we use ν as the running index for iterations.

The main characteristic of the new methods is the coupling of line search procedures in the primal and dual problems with interactive restarts. To assist the reader in understanding this, we first formulate the method analogous to steepest descent, where there are fewer parameters and the algorithmic logic is simpler.

ALGORITHM 1 (Primal-Dual Steepest Descent Algorithm (PDSD)). Construct primal and dual sequences $\{u_0^\nu\} \subset U$ and $\{v_0^\nu\} \subset V$ as follows.

Step 0 (initialization). Choose a real value for the parameter $\varepsilon \geq 0$ (optimality threshold). Set $\nu := 0$ (iteration counter). Specify starting points $\hat{u}_0^0 \in U$ and $\hat{v}_0^0 \in V$ for the sequences $\{\hat{u}_0^\nu\} \subset U$ and $\{\hat{v}_0^\nu\} \subset V$ that will be generated along with $\{u_0^\nu\}$ and $\{v_0^\nu\}$.

Step 1 (evaluation). Calculate

$$\begin{cases} f(\hat{u}_0^\nu),\ g(\hat{v}_0^\nu), & \text{obtaining as by-products } \hat{v}_1^\nu = F(\hat{u}_0^\nu),\ \hat{u}_1^\nu = G(\hat{v}_0^\nu), \\ g(\hat{v}_1^\nu),\ f(\hat{u}_1^\nu), & \text{obtaining as by-products } \hat{u}_2^\nu = G(\hat{v}_1^\nu),\ \hat{v}_2^\nu = F(\hat{u}_1^\nu). \end{cases}$$

Step 2 (interactive restarts). Take

$$\begin{cases} u_0^\nu := \hat{u}_0^\nu, v_1^\nu := \hat{v}_1^\nu, u_2^\nu := \hat{u}_2^\nu & \text{if } f(\hat{u}_0^\nu) \leq f(\hat{u}_1^\nu), \\ u_0^\nu := \hat{u}_1^\nu, v_1^\nu := \hat{v}_2^\nu, u_2^\nu := G(v_1^\nu) & \text{otherwise (this is an interactive primal restart),} \end{cases}$$

$$\begin{cases} v_0^\nu := \hat{v}_0^\nu, u_1^\nu := \hat{u}_1^\nu, v_2^\nu := \hat{v}_2^\nu & \text{if } g(\hat{v}_0^\nu) \geq g(\hat{v}_1^\nu), \\ v_0^\nu := \hat{v}_1^\nu, u_1^\nu := \hat{u}_2^\nu, v_2^\nu := F(u_1^\nu) & \text{otherwise (this is an interactive dual restart).} \end{cases}$$

(In an interactive primal restart, the calculation of $G(v_1^\nu)$ yields the new $g(v_1^\nu)$. Likewise, in an interactive dual restart, the calculation of $F(u_1^\nu)$ yields the new $f(u_1^\nu)$.)

Step 3 (optimality test). Let

$$\hat{u} := \begin{cases} u_0^\nu & \text{if } f(u_0^\nu) \le f(u_1^\nu), \\ u_1^\nu & \text{if } f(u_0^\nu) > f(u_1^\nu), \end{cases} \quad \text{and} \quad \hat{v} := \begin{cases} v_0^\nu & \text{if } g(v_0^\nu) \ge g(v_1^\nu), \\ v_1^\nu & \text{if } g(v_0^\nu) < g(v_1^\nu). \end{cases}$$

If $f(\hat{u}) - g(\hat{v}) \le \varepsilon$, terminate with $\hat{u}$ and $\hat{v}$ being ε-optimal solutions to $(\mathcal{P})$ and $(\mathcal{Q})$.

Step 4 (line segment search). Search for

$$\hat{u}_0^{\nu+1} := \operatorname*{argmin}_{u \in [u_0^\nu, u_2^\nu]} f(u) \quad \text{and} \quad \hat{v}_0^{\nu+1} := \operatorname*{argmax}_{v \in [v_0^\nu, v_2^\nu]} g(v).$$

Return then to Step 1 with the counter ν increased by 1.

Basically, the idea in this method is that if the point $\hat{u}_1^\nu$ calculated as a by-product of finding the projected gradient (1.19) in the dual problem gives a better value to the objective in the primal problem than does the current primal point $\hat{u}_0^\nu$, we take it instead as the current primal point (and accordingly recalculate the projected gradient in the primal problem). Likewise, if the point $\hat{v}_1^\nu$ calculated as a by-product of finding the projected gradient (1.19) in the primal problem happens to give a better value to the objective in the primal problem than the current dual point $\hat{v}_0^\nu$, we take it instead as the current dual point (and accordingly recalculate the projected gradient in the dual problem). Here it may be recalled that $\hat{u}_1^\nu$ minimizes over U the convex quadratic function $L(\cdot, \hat{v}_0^\nu)$, which is a lower approximant to the objective function f in $(\mathcal{P})$ that would have the same minimum value as f over U if $\hat{v}_0^\nu$ were dual optimal. By the same token, $\hat{v}_1^\nu$ maximizes over V the concave quadratic function $L(\hat{u}_0^\nu, \cdot)$, which is an upper approximant to the objective function g in $(\mathcal{Q})$ that would have the same maximum value as g over V if $\hat{u}_0^\nu$ were primal optimal.

Once the issue of triggering a primal or dual interactive restart (or both) settles down in a given iteration, we perform line searches in the directions indicated by the projected gradients in the two problems. If U were the whole space $\mathbb{R}^n$, the primal search direction would be the true direction of steepest descent for f (relative to the geometry induced by the Euclidean norm $\|\cdot\|_P$ on $\mathbb{R}^n$). Similarly, if V were the whole space $\mathbb{R}^m$, the dual search direction would be the true direction of steepest ascent for g (relative to the geometry of the Euclidean norm $\|\cdot\|_Q$ on $\mathbb{R}^m$). However, even in this unconstrained case there would be a difference in the way the searches are carried out, in comparison with classical steepest descent, because instead of looking along an entire half-line we only optimize along a line segment whose length is that of the gradient, i.e., we restrict the step size to be at most 1. (Also, we call for an "exact" optimum because the objective is piecewise strictly quadratic with only finitely many pieces. Clearly, this requirement could be loosened, but the issue is minor and we do not wish to be distracted by it here.)

The restriction to a line segment instead of a half-line is motivated in part by the fact that the line segment is known to lie entirely in the feasible set. A search over a half-line would have to cope with detecting the feasibility boundary in the search parameter, which could be a disadvantage in a high-dimensional setting, although this topic could be explored further. Heuristic motivation for the restriction comes also from evidence of second-order effects induced by the primal-dual feedback, as discussed below. It turns out that under mild assumptions the optimal step sizes along a half-line would eventually be no greater than 1 anyway.

The interactive restarts may seem like a merely opportunistic feature of Algorithm 1, but they have a marked effect, as the numerical tests in §6 will reveal. When interactive restarts are blocked, so that the algorithm reverts to two independent procedures in the primal and dual settings (through a sort of computational "lobotomy"), the performance is slowed down to what one might expect from a steepest-descent-like algorithm. On the other hand, when the interactions are permitted the performance in practice is quite comparable to that of more complicated procedures which attempt to exploit second-order properties. The feedback between primal and dual appears able to supply some such information to the calculations.

In order to develop a broader range of interactive-restart methods, analogous not only to steepest descent but to conjugate gradients, we next formulate as Algorithm 0 a bare-bones procedure which will serve in establishing convergence properties of such methods, including Algorithm 1. The chief complication in Algorithm 0 beyond what has already been seen in Algorithm 1 comes through the introduction of *cycles* for primal and dual restarts. With respect to these cycles an additional threshold parameter is introduced as a technical safeguard against interactive restarts being triggered too freely, without assurance of adequate progress.

ALGORITHM 0 (General Primal-Dual Projected Gradient Algorithm (PDPG)). Construct primal and dual sequences $\{u_0^\nu\} \subset U$ and $\{v_0^\nu\} \subset V$ as follows.

Step 0 (initialization). Choose an integer value for the parameter $k > 0$ (cycle size) and real values for the parameters $\varepsilon \geq 0$ (optimality threshold) and $\delta > 0$ (progress threshold). Set $\nu := 0$ (iteration counter), $k_p := 0$ (primal restart counter), and $k_d := 0$ (dual restart counter). Specify starting points $\hat{u}_0^0 \in U$ and $\hat{v}_0^0 \in V$ for the sequences $\{\hat{u}_0^\nu\} \subset U$ and $\{\hat{v}_0^\nu\} \subset V$ that will be generated along with $\{u_0^\nu\}$ and $\{v_0^\nu\}$.

Step 1 (evaluation). Calculate

$$\begin{cases} f(\hat{u}_0^\nu),\ g(\hat{v}_0^\nu), & \text{obtaining as by-products } \hat{v}_1^\nu = F(\hat{u}_0^\nu),\ \hat{u}_1^\nu = G(\hat{v}_0^\nu), \\ g(\hat{v}_1^\nu),\ f(\hat{u}_1^\nu), & \text{obtaining as by-products } \hat{u}_2^\nu = G(\hat{v}_1^\nu),\ \hat{v}_2^\nu = F(\hat{u}_1^\nu). \end{cases}$$

Step 2 (interactive restarts). Take

$$\begin{cases} u_0^\nu := \hat{u}_0^\nu, v_1^\nu := \hat{v}_1^\nu, u_2^\nu := \hat{u}_2^\nu & \text{if } f(\hat{u}_0^\nu) \leq f(\hat{u}_1^\nu), \text{ or } f(\hat{u}_0^\nu) < f(\hat{u}_1^\nu) + \delta \text{ and } k_p < k, \\ u_0^\nu := \hat{u}_1^\nu, v_1^\nu := \hat{v}_2^\nu, u_2^\nu := G(v_1^\nu) & \text{otherwise (this is an interactive primal restart),} \end{cases}$$

$$\begin{cases} v_0^\nu := \hat{v}_0^\nu, u_1^\nu := \hat{u}_1^\nu, v_2^\nu := \hat{v}_2^\nu & \text{if } g(\hat{v}_0^\nu) \geq g(\hat{v}_1^\nu), \text{ or } g(\hat{v}_0^\nu) > g(\hat{v}_1^\nu) - \delta \text{ and } k_d < k, \\ v_0^\nu := \hat{v}_1^\nu, u_1^\nu := \hat{u}_2^\nu, v_2^\nu := F(u_1^\nu) & \text{otherwise (this is an interactive dual restart).} \end{cases}$$

(In an interactive primal restart the calculation of $G(v_1^\nu)$ yields the new $g(v_1^\nu)$. Likewise, in an interactive dual restart the calculation of $F(u_1^\nu)$ yields the new $f(u_1^\nu)$.) Set

$$\begin{cases} k_p := 0 & \text{if an interactive primal restart occurred in this step,} \\ k_d := 0 & \text{if an interactive dual restart occurred in this step.} \end{cases}$$

Step 3 (optimality test). Let

$$\hat{u} := \begin{cases} u_0^\nu & \text{if } f(u_0^\nu) \leq f(u_1^\nu), \\ u_1^\nu & \text{if } f(u_0^\nu) > f(u_1^\nu), \end{cases} \quad \text{and} \quad \hat{v} := \begin{cases} v_0^\nu & \text{if } g(v_0^\nu) \geq g(v_1^\nu), \\ v_1^\nu & \text{if } g(v_0^\nu) < g(v_1^\nu). \end{cases}$$

If $f(\hat{u}) - g(\hat{v}) \leq \varepsilon$, terminate with $\hat{u}$ and $\hat{v}$ being ε-optimal solutions to $(\mathcal{P})$ and $(\mathcal{Q})$.

Step 4 (search endpoint generation). Take

$$\begin{cases} u_e^\nu := u_2^\nu & \text{if } k_p \equiv 0 (\operatorname{mod} k), \\ u_e^\nu \in U \text{ according to an auxiliary rule} & \text{otherwise,} \end{cases}$$

$$\begin{cases} v_e^\nu := v_2^\nu & \text{if } k_d \equiv 0 (\operatorname{mod} k), \\ v_e^\nu \in V \text{ according an auxiliary rule} & \text{otherwise.} \end{cases}$$

Step 5 (line segment search). Search for

$$\hat{u}_0^{\nu+1} := \operatorname*{argmin}_{u\in[u_0^\nu,u_e^\nu]} f(u) \quad \text{and} \quad \hat{v}_0^{\nu+1} := \operatorname*{argmax}_{v\in[v_0^\nu,v_e^\nu]} g(v).$$

Return then to Step 1 with the counters ν, k_p, and k_d increased by 1.

By specifying the auxiliary rules in Step 4 for generating the search interval endpoints u_e^ν and v_e^ν in iterations where k_p or k_d is not a multiple of k, we obtain particular *realizations* of Algorithm 0. An attractive case in which these rules correspond to a "conjugate gradient" approach with cycle size k will be developed presently as Algorithm 2. Before proceeding, however, we want to emphasize for theoretical purposes that Algorithm 1 is itself a particular realization of Algorithm 0.

PROPOSITION 2.1. *Algorithm* 0 *reduces to Algorithm* 1 *when the cycle size is* $k = 1$ (*except for a slight difference in iteration* $\nu = 0$).

Proof. In returning from Step 4 of Algorithm 0 to Step 1, the counters k_p and k_d are always at least 1. It follows that if $k = 1$ the condition in Step 2 with progress threshold δ will never come into play after such a return. Thus, the only possible effect of this threshold will be in iteration $\nu = 0$, where a restart will be avoided unless it improves the objective by at least δ. In Step 4, k_p and k_d will always be multiples of k, so we will always have $u_e^\nu = u_2^\nu$ and $v_e^\nu = v_2^\nu$. Thus the counters k_p and k_d become redundant and the auxiliary rules moot. □

In Algorithm 0 in general, k_p counts iterations in the primal problem from the start or the most recent interactive primal restart. An iteration that begins with k_p being a positive multiple of k is said to be one in which an *ordinary primal restart* takes place (whether or not an interactive primal restart also takes place), because it marks the completion of a cycle of k iterations not cut short by an interactive primal restart. Every iteration involving an ordinary or interactive primal restart ends by searching the line segment $[u_0^\nu, u_2^\nu]$, where $u_2^\nu - u_0^\nu$ is the negative of the current projected gradient of f in (1.19). The dual situation is parallel in terms of the counter k_d and the notion of an *ordinary dual restart*.

The role of the parameter $\delta > 0$ is to control the extent to which the algorithm forgoes interactive restarts and insists on waiting for ordinary restarts. Interactive restarts are always accepted if they improve the corresponding objective value by the amount δ or more, but there can only be finitely many iterations with this size of improvement, due to the finiteness of the joint optimal value in $(\mathcal{P})$ and $(\mathcal{Q})$ (Theorem 1.1). When such improvement is no longer possible, interactive restarts are blocked in the primal until an ordinary restart has again intervened, unless one is already occurring in the same iteration; the same holds in the dual. This feature ensures that full cycles of k iterations will continue to be performed in the primal and dual as long as the algorithm keeps running, which is important in establishing certain properties of convergence.

Recall that the point u_2^ν minimizes over U the lower envelope function $L(u, v_1)$ as a representation of $f(u)$ at u_0^ν (Proposition 1.5), which has $\nabla_u L(u_0^\nu, v_1^\nu) = \nabla f(u_0^\nu)$. Even apart from the projected gradient interpretation, therefore, there is motivation in searching the line segment $[u_0^\nu, u_2^\nu]$ in order to reduce the objective value $f(u)$ in primal. The same motivation exists for searching $[v_0^\nu, v_2^\nu]$ in the dual.

As a matter of fact, we shall prove in Proposition 5.1 that on exiting from Step 5 (line segment search) of Algorithm 0, the point $\hat{u}_1^{\nu+1} = G(\hat{v}_0^{\nu+1})$ will be the minimum point relative to U for the envelope function

$$f^\nu(u) := \max_{v\in[v_0^\nu,v_2^\nu]} L(u,v) \le \max_{v\in V} L(u,v) = f(u).$$

When the algorithm reaches Step 2 in the iteration, it will compare the point $\hat{u}_0^{\nu+1}$ resulting from the just-completed line search in the primal with the point $\hat{u}_1^{\nu+1}$ resulting from minimizing the lower envelope function $f^\nu(u)$, and it will take the "better" of the two as the next primal iterate. In the dual procedure there are corresponding comparisons between $\hat{v}_0^{\nu+1}$ and $\hat{v}_1^{\nu+1}$.

We focus now on a specialization of Algorithm 0 in which, in contrast to Algorithm 1, the cycle provisions are crucial and the auxiliary rules nontrivial. The rules emulate those of the classical conjugate gradient method (Hestenes–Stiefel).

ALGORITHM 2 (Primal-Dual Conjugate Gradient Method (PDCG)). In the implementation of Algorithm 0, choose a cycle size $k > 1$ and use the following auxiliary rules to get the search intervals in Step 4. Unless $k_p \equiv 0 (\bmod k)$, set

$$(2.1)\ w_p^\nu := \nabla_P f(u_0^\nu) - \nabla_P f(u_0^{\nu-1}),$$

$$(2.2)\ \beta_p^\nu := \begin{cases} \max\{0, \langle w_p^\nu,\, u_0^\nu - u_2^\nu\rangle_P\}/\langle w_p^\nu,\, u_e^{\nu-1} - u_0^\nu\rangle_P & \text{if } \langle w_p^\nu,\, u_e^{\nu-1} - u_0^\nu\rangle_P > 0, \\ 0 & \text{otherwise,} \end{cases}$$

$$(2.3)\ u_{cg}^\nu := (u_2^\nu + \beta_p^\nu u_e^{\nu-1})/(1+\beta_p^\nu),$$

$$(2.4)\ [u_0^\nu, u_e^\nu] := \begin{cases} [u_0^\nu, u_{cg}^\nu] & \text{if } \|u_{cg}^\nu - u_0^\nu\|_P \ge 1, \\ L_p^\nu \cap U & \text{otherwise,} \end{cases}$$

where $L_p^\nu = \{u \in \mathbb{R}^n \mid u = u_0^\nu + \lambda(u_{cg}^\nu - u_0^\nu),\ 0 \le \lambda \le \|u_{cg}^\nu - u_0^\nu\|_P^{-1}\}$. Similarly, unless $k_d \equiv 0 (\bmod k)$, set

$$(2.5)\ w_d^\nu := -\nabla_Q g(v_0^\nu) + \nabla_Q g(v_0^{\nu-1}),$$

$$(2.6)\ \beta_d^\nu := \begin{cases} \max\{0, \langle w_d^\nu,\, v_0^\nu - v_2^\nu\rangle_Q\}/\langle w_d^\nu,\, v_e^{\nu-1} - v_0^\nu\rangle_Q & \text{if } \langle w_d^\nu,\, v_e^{\nu-1} - v_0^\nu\rangle_Q > 0, \\ 0 & \text{otherwise,} \end{cases}$$

$$(2.7)\ v_{cg}^\nu := (v_2^\nu + \beta_d^\nu v_e^{\nu-1})/(1+\beta_d^\nu),$$

$$(2.8)\ [v_0^\nu, v_e^\nu] := \begin{cases} [v_0^\nu, v_{cg}^\nu] & \text{if } \|v_{cg}^\nu - v_0^\nu\|_Q \ge 1, \\ L_d^\nu \cap V & \text{otherwise,} \end{cases}$$

where $L_d^\nu = \{v \in \mathbb{R}^m \mid v = v_0^\nu + \lambda(v_{cg}^\nu - v_0^\nu),\ 0 \le \lambda \le \|v_{cg}^\nu - v_0^\nu\|_Q^{-1}\}$.

Note that because the auxiliary rules are never invoked in iteration $\nu = 0$ (where $k_p = 0$ and $k_d = 0$), the points indexed with $\nu - 1$ in the statement of Algorithm 2 are all well defined. Another thing to observe is the fact that in (2.2) and (2.6) we actually have

$$(2.9)\qquad \langle w_p^\nu,\, u_e^{\nu-1} - u_0^\nu\rangle_P \ge 0 \quad \text{and} \quad \langle w_d^\nu,\, v_e^{\nu-1} - v_0^\nu\rangle_Q \ge 0.$$

These inequalities follow from (2.1) and (2.5) and the monotonicity of gradient mappings of convex functions. In Proposition 4.4 we shall prove that under the critical face condition the inequalities in (2.9) hold *strictly* in a vicinity of the optimal solution if the critical faces are reached by the corresponding iterates.

On the other hand, it is apparent from (2.3) and (2.7) that

$$(2.10)\quad u_{cg}^\nu - u_0^\nu = \frac{u_2^\nu - u_0^\nu + \beta_p^\nu(u_e^{\nu-1} - u_0^\nu)}{(1+\beta_p^\nu)} \quad \text{and} \quad v_{cg}^\nu - v_0^\nu = \frac{v_2^\nu - v_0^\nu + \beta_d^\nu(u_e^{\nu-1} - v_0^\nu)}{(1+\beta_d^\nu)}.$$

Hence, the search direction vector in the primal is, in fact, a convex combination of the P-projection of $-\nabla_P f(u_0)$ and the search direction vector in the previous primal iteration. Similarly, the search direction vector in the dual is a convex combination of the Q-projection of $\nabla_Q g(v_0)$ and the search direction vector in the previous dual iteration.

We shall prove in Theorem 4.5 that under the critical face condition, the primal iterations in Algorithm 2 reduce in a vicinity of the optimal solution to $(\mathcal{P})$ to the execution of the Hestenes–Stiefel conjugate gradient method if the critical face U_0 is eventually reached by the primal iterates, and similarly for the dual iterations. From this we will obtain a termination property for Algorithm 2, which will be invoked by an interactive restart of the algorithm.

Algorithm 2 departs a bit from the philosophy of Algorithm 1 in utilizing unprojected gradients in (2.1) and (2.5) instead of just projected gradients. These unprojected gradients are available through (1.11) and (1.12) (also (1.15) or (1.16)), and for multistage optimization problems in the pattern laid out in [7] they can still be calculated without having to invoke the gigantic R matrix. An earlier version of Algorithm 2 that we worked with did use the projected gradients exclusively, and it performed similarly, but there were technical difficulties in establishing a finite termination property. Future research may shed more light on this issue. The same can be said of another small departure in Algorithm 2 from the philosophy one might hope to maintain in a "conjugate gradient" method: the introduction on occasion of step sizes possibly greater than 1 relative to $[u_0^\nu, u_{cg}^\nu]$ or $[v_0^\nu, v_{cg}^\nu]$ (although not, of course, relative to the designated intervals $[u_0^\nu, u_e^\nu]$ or $[v_0^\nu, v_e^\nu]$) through the second alternatives in (2.4) or (2.8).

3. Global convergence and local quadratic structure. This section establishes some basic convergence properties of Algorithms 0, 1, and 2. It also reveals the special quadratic structure in $(\mathcal{P})$ and $(\mathcal{Q})$ around the optimal solutions $\bar{u}$ and $\bar{v}$ in the case where the critical face condition is satisfied, which will be utilized in further convergence analysis in §5.

PROPOSITION 3.1 (feasible descent and ascent). (a) *In Algorithm* 0 (*hence also in Algorithms* 1 *and* 2) *the vector* $u_2^\nu - u_0^\nu$ *gives a feasible descent direction for the primal objective function* f *at* u_0^ν (*unless* $u_2^\nu - u_0^\nu = 0$, *in which case* $u_0^\nu = \bar{u}$). *Similarly, the vector* $v_2^\nu - v_0^\nu$ *gives a feasible ascent direction for the dual objective function* g *at* v_0^ν (*unless* $v_2^\nu - v_0^\nu = 0$, *in which case* $v_0^\nu = \bar{v}$).

(b) *In Algorithm* 2, *the vector* $u_{cg}^\nu - u_0^\nu$ *gives a feasible descent direction for the primal objective* f *at* u_0^ν *unless* $u_0^\nu = \bar{u}$. *Similarly, the vector* $v_{cg}^\nu - v_0^\nu$ *gives a feasible ascent direction for the dual objective* g *at* v_0^ν *unless* $v_0^\nu = \bar{v}$. *Thus, Algorithm* 2 *is well defined in the sense that, regardless of the type of iteration, as long as it does not terminate in optimality, the vector* $u_e^\nu - u_0^\nu$ *gives a feasible descent direction at* u_0^ν *in the primal while the vector* $v_e^\nu - v_0^\nu$ *gives a feasible ascent direction at* v_0^ν *in the dual.*

Proof. (a) We know that u_2^ν minimizes $L(u, v_1^\nu)$ over $u \in U$, where $L(u, v_1^\nu)$ is given by formula (1.17). We obtain from this formula that unless $u_2^\nu = u_0^\nu$, implying u_0^ν is optimal for the primal, we must have $\nabla f(u_0^\nu)\cdot(u_2^\nu - u_0^\nu) < 0$. Descent in this direction is feasible because the line segment $[u_0^\nu, u_2^\nu]$ is included in U by convexity. The proof of the dual part is parallel.

(b) The argument is by induction. From the optimality test in Step 3 we see that the algorithm will terminate at $(\bar{u}, \bar{v})$ if either $u_0^\nu = \bar{u}$ in the primal or $v_0^\nu = \bar{v}$ in the dual. (For instance, if $u_0^\nu = \bar{u}$, then $v_1^\nu = \bar{v}$, so that $f(\hat{u}) - g(\hat{v}) = 0$.) Suppose neither u_0^ν nor v_0^ν is optimal. Proposition 3.1(a) covers our claims for the initial iteration of each primal or dual cycle. Suppose that the claims are true for iteration $l - 1$ of a primal cycle, $0 < l < k$, which corresponds to iteration $\nu - 1$ of the algorithm as a whole. We have $(u_2^\nu - u_0^\nu)\cdot\nabla f(u_0^\nu) < 0$ by part (a) and $(u_e^{\nu-1} - u_0^\nu)\cdot\nabla f(u_0^\nu) \leq 0$ through the line search. (Note that we get this inequality instead of an equation because the search is over a segment rather than a half-line; the minimizing point could be at the end of the segment.) Hence

$$(u_{cg}^\nu - u_0^\nu)\cdot\nabla f(u_0^\nu) = \frac{(u_2^\nu - u_0^\nu)\cdot\nabla f(u_0^\nu) + \beta_p^\nu (u_e^{\nu-1} - u_0^\nu)\cdot\nabla f(u_0^\nu)}{1+\beta_p^\nu} < 0.$$

The vector $u_{cg}^\nu - u_0^\nu \neq 0$, therefore, gives a descent direction, so the segment L_p^ν in (2.4) is nontrivial. From (2.3), we see further that u_{cg}^ν is a convex combination of two feasible points $u_2^\nu \in U$ and $u_e^{\nu-1} \in U$. Hence the point u_{cg}^ν is feasible, i.e., $u_{cg}^\nu \in U$, and the direction of $u_{cg}^\nu - u_0^\nu$ is a feasible direction in the primal at u_0^ν. The vector $u_e^\nu - u_0^\nu$, therefore, gives a feasible descent direction for f at u_0^ν, since it results from a scaling of the vector $u_{cg}^\nu - u_0^\nu$. Iteration l of the primal cycle thus again satisfies the claim. The case of dual cycles is handled similarly. □

THEOREM 3.2 (global convergence). *In Algorithm* 0 (*hence also in Algorithms* 1 *and* 2) *with optimality threshold* $\varepsilon > 0$, *termination must come with* ε*-optimal solutions* $\hat{u}$ *and* $\hat{v}$ *in just a finite number of iterations. With* $\varepsilon = 0$, *unless the procedure happens to terminate with the exact optimal solutions* $\bar{u}$ *and* $\bar{v}$ *in a finite number of iterations, the sequences generated will be such that* $u_0^\nu \to \bar{u}$ *and* $v_0^\nu \to \bar{v}$ *as* $\nu \to \infty$. *Furthermore, then* $u_1^\nu \to \bar{u}$ *and* $u_2^\nu \to \bar{u}$, *as well as* $v_1^\nu \to \bar{v}$ *and* $v_2^\nu \to \bar{v}$.

Proof. Consider first the case where $\varepsilon = 0$. From Proposition 1.4, the point $u_2 = G(F(u_0))$ depends continuously on u_0. Denote by $\mathcal{D}$ the continuous mapping $u_0 \mapsto (u_0, u_2 - u_0)$ from U to $U \times \mathbb{R}^n$. Let $\mathcal{M} : U \times \mathbb{R}^n \to U$ be the line search mapping defined by

$$\mathcal{M}(u_0, d) = \operatorname*{argmin}_{u\in[u_0,u_0+d]} f(u).$$

The mapping $\mathcal{M}$ is closed at the point (u_0, d) with $d \neq 0$; cf. [21, Thm. 8.3.1]. Now by Proposition 3.1(a), $u_2 - u_0 \neq 0$ for $u_0 \neq \bar{u}$. Hence the composite mapping $\mathcal{M}\circ\mathcal{D}$ is closed on $U \setminus \{\bar{u}\}$; cf. [21, Thm. 7.3.2]. Define

$$\mathcal{A} = \mathcal{B}\circ\mathcal{M}\circ\mathcal{D},$$

where $\mathcal{B} : U \rightrightarrows U$ is the point-to-set mapping $\mathcal{B}(u) = \{u' \in U \mid f(u') \leq f(u)\}$. Note that the sequence $\{f(u_0^\nu)\}$ is nonincreasing. Now let $\mathcal{K}_p$ be the sequence that consists of the indices of those iterations in which a line search on $[u_0^\nu, u_2^\nu]$ is performed for the primal objective function. Then $\mathcal{K}_p$ is an infinite subsequence of $\{\nu\}$ unless the procedure happens to terminate with the exact optimal solutions $\bar{u}$ and $\bar{v}$ in a finite number of iterations. Let ν'' and ν' be two consecutive elements in $\mathcal{K}_p$ with $\nu'' > \nu'$. Then we can write

$$u_0^{\nu''} \in \mathcal{A}(u_0^{\nu'}).$$

By Proposition 3.1, moreover, the vector $u_2 - u_0$ is a descent direction for the primal objective $f(u)$ at u_0 unless u_0 is already optimal. Since we are in the fully quadratic case, the set $\{u \in \mathbb{R}^n \mid f(u) \leq f(u_0^0)\}$ is compact, and the optimal solution $\bar{u}$ for problem $(\mathcal{P})$ is unique. It follows then that $u_0^\nu \to \bar{u}$ as $\nu \to \infty$, $\nu \in \mathcal{K}_p$; cf. [21, Thm. 7.3.4]. Therefore, $f(u_0^\nu) \to f(\bar{u})$ as $\nu \to \infty$, which in turn implies $u_0^\nu \to \bar{u}$ as $\nu \to \infty$ since f is strongly convex (Theorem 1.1).

For analogous reasons, $v_0^\nu \to \bar{v}$. Then, since $u_1^\nu = G(v_0^\nu)$ and $\bar{u} = G(\bar{v})$ with the mapping G continuous (Proposition 1.4), we have $u_1^\nu \to \bar{u}$. Likewise, $v_1^\nu \to \bar{v}$. The argument can be applied then again: we have $u_2^\nu = G(v_1^\nu)$, so $u_2^\nu \to \bar{u}$, and, in parallel fashion, $v_2^\nu \to \bar{v}$.

In particular, we have $f(u_0^\nu) - g(v_0^\nu) \to f(\bar{u}) - g(\bar{v}) = 0$ because f and g are continuous (Theorems 1.1 and 1.2(a)). In the case where $\varepsilon > 0$, this guarantees termination in finitely many iterations. □

COROLLARY 3.3 (points in the critical faces). *The sequences generated by Algorithm* 0 *have the property that eventually u_1^ν and u_2^ν belong to the primal critical face U_0, while v_1^ν and v_2^ν belong to the dual critical face V_0.*

Proof. This follows via Proposition 1.8. □

COROLLARY 3.4 (a special case of finite termination). *If $\varepsilon = 0$ and either of the critical faces U_0 or V_0 consists of just a single point, Algorithm* 0 (*and therefore also Algorithms* 1 *and* 2) *will terminate at the optimal solution pair* $(\bar{u}, \bar{v})$ *after a finite number of iterations.*

Proof. When U_0 consists of the single point $\bar{u}$, we have by Corollary 3.3 that $u_2^\nu = \bar{u}$ for all sufficiently large ν. Once this is the situation, the line search in the first iteration of the next primal cycle will yield $\bar{u}$. On returning to Step 1 for the succeeding iteration, $\bar{v}$ will be generated as $F(\bar{u})$, and termination must then come in Step 3. The situation is analogous when V_0 consists of just $\bar{v}$. □

A companion result to Corollary 3.3 is the following.

PROPOSITION 3.5 (convergence onto critical faces). *Let $\{u_0^\nu\}$ and $\{v_0^\nu\}$ be sequences generated by Algorithms* 1 *or* 2. *Then for the primal critical face U_0, we have either $u_0^\nu \in U_0$ for all sufficiently large ν or $u_0^\nu \notin U_0$ for all sufficiently large ν. Similarly, for the dual critical face V_0 we have either $v_0^\nu \in V_0$ for all sufficiently large ν or $v_0^\nu \notin V_0$ for all sufficiently large ν.*

Proof. We prove the primal part. The proof of the dual part is similar. Observe that $\hat{v}_0^\nu \to \bar{v}$ as $v_0^\nu \to \bar{v}$ in the algorithm. Hence by Proposition 1.8, we have $\hat{u}_1^\nu \in U_0$ as well as $u_2^\nu \in U_0$ for sufficiently large ν. Then in Algorithm 1 we have

$$u_0^\nu \in U_0 \Rightarrow [u_0^\nu, u_2^\nu] \subset U_0 \Rightarrow \hat{u}_0^{\nu+1} \in U_0 \Rightarrow u_0^{\nu+1} \in U_0$$

since $u_0^{\nu+1}$ is defined either as $\hat{u}_0^{\nu+1}$ or as $\hat{u}_1^{\nu+1}$. From this it is apparent that our assertion is valid in the case of sequences generated by Algorithm 1.

For Algorithm 2, we claim that for sufficiently large ν we have $u_e^\nu \in U_0$ when $u_0^\nu \in U_0$. For if $u_e^\nu = u_2^\nu$, we certainly have $u_e^\nu = u_2^\nu \in U_0$. If $u_e^\nu \neq u_2^\nu$, then u_{cg}^ν is a convex combination of $u_e^{\nu-1}$ and $u_2^\nu \in U_0$, and there is no interactive restart in iteration ν, i.e., $\hat{u}_0^\nu = u_0^\nu \in U_0$. Now $\hat{u}_0^\nu \neq u_0^{\nu-1}$ by Proposition 3.1(b). Hence we have either $\hat{u}_0^\nu = u_e^{\nu-1}$, which implies $u_e^{\nu-1} \in U_0$, or $\hat{u}_0^\nu \in \mathrm{ri}[u_0^{\nu-1}, u_e^{\nu-1}]$, which also implies $u_e^{\nu-1} \in U_0$ since U_0 is a face of U. Then $u_{cg}^\nu \in U_0$, and by the definition of u_e^ν in the algorithm we have $u_e^\nu \in U_0$. Therefore,

$$u_0^\nu \in U_0 \quad \Rightarrow \quad [u_0^\nu, u_e^\nu] \subset U_0 \quad \Rightarrow \quad \hat{u}_0^{\nu+1} \in U_0 \quad \Rightarrow \quad u_0^{\nu+1} \in U_0$$

for sufficiently large ν. Thus, our assertion is valid also in the case of sequences generated by Algorithm 2. □

Remark. With the aid of the concept of an *ultimate quadratic region* introduced in Definition 3.7 below, it will be seen that when the critical face condition is satisfied, the assertion of the proposition can be written as follows: after the sequences $\{u_0^\nu\}$ and $\{v_0^\nu\}$ have entered an ultimate quadratic region, once $u_0^{\nu'} \in U_0$ for some ν', then $u_0^\nu \in U_0$ for all $\nu \geq \nu'$; and similarly once $v_0^{\nu''} \in V_0$ for some ν'', then $v_0^\nu \in V_0$ for all $\nu \geq \nu''$.

For Algorithm 2, broader results on finite termination than the one in Corollary 3.4 will be obtained when the critical face condition is satisfied through reduction to a simpler quadratic structure which is identified as governing in a neighborhood of the solution. This local structure will also be the basis for developing convergence rates for Algorithms 1 and 2 in cases without finite termination. In developing it in the next theorem, we recall the notion of the affine hull $\operatorname{aff} C$ of a convex set C: this is the smallest affine set that includes C, or equivalently, the intersection of all the hyperplanes that include C [17].

THEOREM 3.6 (quadratic structure near optimality). *Suppose the critical face condition is satisfied. Then f is quadratic in some neighborhood of $\bar{u}$, while g is quadratic in some neighborhood of $\bar{v}$. Furthermore, for points $u_0 \in U$ and $v_0 \in V$ sufficiently close to $\bar{u}$ and $\bar{v}$, the P-projection of $-\nabla_P f(u_0)$ on $U - u_0$ is the same as that on $\operatorname{aff} U_0 - u_0$, while the Q-projection of $\nabla_Q g(v_0)$ on $V - v_0$ is the same as that on $\operatorname{aff} V_0 - v_0$.*

Proof. Since by Proposition 1.8 the point $v_1 = F(u_0)$ lies in the critical face V_0 when u_0 is sufficiently close to $\bar{u}$, we have

$$\max_{v \in V}\{v\cdot(q - Ru) - \tfrac{1}{2}v\cdot Qv\} = \max_{v \in V_0}\{v\cdot(q - Ru) - \tfrac{1}{2}v\cdot Qv\}. \tag{3.1}$$

The mapping F is continuous (Proposition 1.4) and $\bar{v} \in \operatorname{ri} V_0$ by assumption, so we have $v_1 \in \operatorname{ri} V_0$ when u_0 is sufficiently close to $\bar{u}$. Then (3.1) can further be written instead as

$$\max_{v \in V}\{v\cdot(q - Ru) - \tfrac{1}{2}v\cdot Qv\} = \max_{v \in \operatorname{aff} V_0}\{v\cdot(q - Ru) - \tfrac{1}{2}v\cdot Qv\}.$$

Locally, therefore,

$$f(u) = p\cdot u + \tfrac{1}{2}u\cdot Pu + \max_{v \in \operatorname{aff} V_0}\{v\cdot(q - Ru) - \tfrac{1}{2}v\cdot Qv\}. \tag{3.2}$$

Similarly, for v in some neighborhood of $\bar{v}$ we have

$$g(v) = q\cdot v - \tfrac{1}{2}v\cdot Qv + \min_{u \in \operatorname{aff} U_0}\{u\cdot(p - R^T v) + \tfrac{1}{2}u\cdot Pu\}. \tag{3.3}$$

The set $\operatorname{aff} V_0$, because it is affine and contains $\bar{v}$, has the form $\bar{v} + S$ for a certain subspace S of $\mathbb{R}^m$, which in turn can be written as the set of all vectors of the form $v' = Dw$ for a certain $m \times d$ matrix D of rank d (the dimension of S). In substituting $v = \bar{v} + Dw$ in (3.2) and taking the maximum instead over all $w \in \mathbb{R}^d$, we see through elementary calculus and linear algebra that the maximum value is a quadratic function of u. This establishes that $f(u)$ is quadratic in u on a neighborhood of $\bar{u}$. The same argument can be pursued in (3.3) to verify that $g(v)$ is quadratic around $\bar{v}$.

Next we consider the projected gradients. According to Proposition 1.6, the P-projection of $-\nabla_P f(u_0)$ on $U - u_0$ is the vector $u_2 - u_0$, where $u_2 = G\big(F(u_0)\big)$. When u_0 is close enough to $\bar{u}$ in U, u_2 belongs by Proposition 1.8 to $\operatorname{ri} U_0$, which is the interior of U_0 relative to $\operatorname{aff} U_0$. Thus, for u_0 in some neighborhood of $\bar{u}$ in U_0, the P-projection of $-\nabla_P f(u_0)$ on $U - u_0$ belongs to the relatively open convex subset $\operatorname{ri} U_0 - u_0$ of $U - u_0$ and must be the same as the projection on this subset or on $U_0 - u_0$ itself. When the nearest point of a convex set C belongs to $\operatorname{ri} C$, it is the same as the nearest point of $\operatorname{aff} C$. The P-projection of $-\nabla_P f(u_0)$ on $U - u_0$ is therefore the same as the P-projection of $-\nabla_P f(u_0)$ on $\operatorname{aff} U_0 - u_0$. The Q-projection of $\nabla_Q g(v_0)$ on $V - v_0$ is analyzed in parallel fashion. □

Theorem 3.6, together with Proposition 1.8, makes it possible for us to concentrate our analysis of the terminal behavior of our algorithms, in the case of optimality threshold $\varepsilon = 0$, on regions around $(\bar{u}, \bar{v})$ of the following special kind.

Definition 3.7 (ultimate quadratic regions). By an *ultimate quadratic region* for problems $(\mathcal{P})$ and $(\mathcal{Q})$ when the critical face condition is satisfied, we shall mean an open convex neighborhood $U^* \times V^*$ of $(\bar{u}, \bar{v})$ with the properties that

(a) $U^* \cap U_0 = U^* \cap \operatorname{ri} U_0$ and $V^* \cap V_0 = V^* \cap \operatorname{ri} V_0$;

(b) f is quadratic on U^* and g is quadratic on V^*;

(c) for all $u_0 \in U^* \cap U$ the P-projection of $-\nabla_P f(u_0)$ on $U - u_0$ is that on $(\operatorname{aff} U_0) - u_0$, while for all $v_0 \in V^* \cap V$ the Q-projection of $\nabla_Q g(v_0)$ on $V - v_0$ is that on $(\operatorname{aff} V_0) - v_0$;

(d) for all $u_0 \in U^* \cap U$ and $v_0 \in V^* \cap V$ the points $u_1 = G(v_0)$, $v_1 = F(u_0)$, $u_2 = G(v_1)$, and $v_2 = F(u_1)$ are such that u_1 and u_2 belong to $\operatorname{ri} U_0$, while v_1 and v_2 belong to $\operatorname{ri} V_0$.

Here we recognize that the affine sets $\operatorname{aff} U_0$ and $\operatorname{aff} V_0$ are translates of certain subspaces, which in fact are the sets $(\operatorname{aff} U_0) - \bar{u}$ and $(\operatorname{aff} V_0) - \bar{v}$. The projections in (c) of this definition can also be described in terms of these subspaces. Let

$$\begin{aligned} &S_p = P\text{-projection mapping onto the subspace}(\operatorname{aff} U_0) - \bar{u}, \\ (3.4) \qquad &S_d = Q\text{-projection mapping onto the subspace}(\operatorname{aff} V_0) - \bar{v}, \\ &S_p^\perp = I - S_p, \qquad S_d^\perp = I - S_d. \end{aligned}$$

The mapping $S_p^\perp$ projects onto the subspace of $\mathbb{R}^n$ that is orthogonally complementary to $(\operatorname{aff} U_0) - \bar{u}$ with respect to the P-inner product in (1.10), while the mapping $S_d^\perp$ projects onto the subspace of $\mathbb{R}^m$ that is orthogonally complementary to $(\operatorname{aff} V_0) - \bar{v}$ with respect to the Q-inner product. All these projections are linear transformations, of course.

Proposition 3.8 (projection decomposition). *For (u_0, v_0) in an ultimate quadratic region $U^* \times V^*$, one has for $u_2 := G(F(u_0))$ and $v_2 := F(G(v_0))$ that*

$$\begin{aligned} u_2 - u_0 &= S_p\bigl(-\nabla_P f(u_0)\bigr) - S_p^\perp(u_0 - \bar{u}) = -S_p\bigl(\nabla_P^2 f(\bar{u})(u_0 - \bar{u})\bigr) - S_p^\perp(u_0 - \bar{u}), \\ v_2 - v_0 &= S_d\bigl(\nabla_Q g(v_0)\bigr) - S_d^\perp(v_0 - \bar{v}) = S_d\bigl(\nabla_Q^2 g(\bar{v})(v_0 - \bar{v})\bigr) - S_d^\perp(v_0 - \bar{v}). \end{aligned}$$

Proof. The P-projection of $-\nabla_P f(u_0)$ on $(\operatorname{aff} U_0) - u_0$ can be realized by taking the P-projection of $-\nabla_P f(u_0) + (u_0 - \bar{u})$ on the set $(\operatorname{aff} U_0) - u_0 + (u_0 - \bar{u})$ and then subtracting $(u_0 - \bar{u})$. Therefore, in a region with property (c) of Definition 3.7 we have by (1.17) in Proposition 1.6 that

$$u_2 - u_0 = S_p\bigl(-\nabla_P f(u_0) + (u_0 - \bar{u})\bigr) - (u_0 - \bar{u}) = S_p\bigl(-\nabla_P f(u_0)\bigr) - (I - S_p)(u_0 - \bar{u}),$$

which is the first equality asserted. The second equality comes from having $\nabla_P f(u_0) = \nabla_P f(\bar{u}) + \nabla_P^2 f(\bar{u})(u_0 - \bar{u})$ (since f is quadratic in the region in question), and $S_p\bigl(\nabla_P f(\bar{u})\bigr) = 0$ by the optimality of $\bar{u}$. The proof of the dual equalities is along the same lines. □

4. Rate of convergence. In taking advantage of the existence of an ultimate quadratic region, we shall utilize in our technical arguments a change of variables that will make a number of basic properties clearer. This change of variables amounts to the introduction of orthonormal coordinate systems relative to the inner products naturally associated with our problems, namely, $\langle\cdot,\cdot\rangle_P$ on $\mathbb{R}^n$ and $\langle\cdot,\cdot\rangle_Q$ on $\mathbb{R}^m$, as given in (1.10). The coordinate systems are introduced in such a way that the subspaces $(\operatorname{aff} U_0) - \bar{u}$ and $(\operatorname{aff} V_0) - \bar{v}$ for the projections in (3.4) and Proposition 3.8 take a very simple form.

Let W be an $n \times n$ orthogonal matrix and Z an $m \times m$ orthogonal matrix. Our shift will be from u and v to $\tilde{u} = WP^{1/2}u$ and $\tilde{v} = ZQ^{1/2}v$. In these variables and with

$$\tilde{U} = WP^{\frac{1}{2}}U, \qquad \tilde{V} = ZQ^{\frac{1}{2}}V,$$

our primal and dual problems take the form

$$(\tilde{\mathcal{P}}) \qquad \text{minimize } \tilde{f}(\tilde{u}) \text{ over all } \tilde{u} \in \tilde{U},$$

$$(\tilde{\mathcal{Q}}) \qquad \text{maximize } \tilde{g}(\tilde{v}) \text{ over all } \tilde{v} \in \tilde{V},$$

where we have

$$\tilde{f}(\tilde{u}) = \sup_{\tilde{v}\in\tilde{V}} \tilde{L}(\tilde{u},\tilde{v}) \quad \text{and} \quad \tilde{g}(\tilde{v}) = \inf_{\tilde{u}\in\tilde{U}} \tilde{L}(\tilde{u},\tilde{v}), \tag{4.1}$$

$$\tilde{F}(\tilde{u}) = \operatorname*{argmax}_{\tilde{v}\in\tilde{V}} \tilde{L}(\tilde{u},\tilde{v}) \quad \text{and} \quad \tilde{G}(\tilde{v}) = \operatorname*{argmin}_{\tilde{u}\in\tilde{U}} \tilde{L}(\tilde{u},\tilde{v}), \tag{4.2}$$

in the notation that

$$\tilde{L}(\tilde{u},\tilde{v}) = \tilde{p}\cdot\tilde{u} + \tfrac{1}{2}\|\tilde{u}\|^2 + \tilde{q}\cdot\tilde{v} - \tfrac{1}{2}\|\tilde{v}\|^2 - \tilde{v}\cdot\tilde{R}\tilde{u} \quad \text{on} \quad \tilde{U}\times\tilde{V}, \tag{4.3}$$

$$\tilde{p} = WP^{-\frac{1}{2}}p, \qquad \tilde{q} = ZQ^{-\frac{1}{2}}q, \qquad \tilde{R} = ZQ^{-\frac{1}{2}}RP^{-\frac{1}{2}}W^T. \tag{4.4}$$

The optimal solutions $\bar{u}$ and $\bar{u}$ to $(\mathcal{P})$ and $(\mathcal{Q})$ translate into optimal solutions $\bar{\tilde{u}}$ and $\bar{\tilde{v}}$ to $(\tilde{\mathcal{P}})$ and $(\tilde{\mathcal{Q}})$, namely,

$$\bar{\tilde{u}} = WP^{\frac{1}{2}}\bar{u} \quad \text{and} \quad \bar{\tilde{v}} = ZQ^{\frac{1}{2}}\bar{v}. \tag{4.5}$$

Let d_1 be the dimension of the subspace $(\operatorname{aff} U_0) - \bar{u}$ and d_2 the dimension of the subspace $(\operatorname{aff} V_0) - \bar{v}$. We choose W such that, in the new coordinates corresponding to the components of $\tilde{u}$, the set $WP^{1/2}(\operatorname{aff} U_0 - \bar{u}) = \operatorname{aff} \tilde{U}_0 - \bar{\tilde{u}}$ is the subspace spanned by the first d_1 columns of I_n. Likewise, we choose Z such that in the $\tilde{v}$ coordinates the set $ZQ^{1/2}(\operatorname{aff} V_0 - \bar{v}) = \operatorname{aff} \tilde{V}_0 - \bar{\tilde{v}}$ is the subspace spanned by the first d_2 columns of I_m. We partition the vectors $\tilde{u} \in \mathbb{R}^n$ and $\tilde{v} \in \mathbb{R}^m$ into

$$\tilde{u} = \begin{pmatrix} \tilde{u}_f \\ \tilde{u}_r \end{pmatrix} \quad \text{and} \quad \tilde{v} = \begin{pmatrix} \tilde{v}_f \\ \tilde{v}_r \end{pmatrix}, \tag{4.6}$$

where $\tilde{u}_f$ consists of the first d_1 components of $\tilde{u}$ and $\tilde{v}_f$ consists of the first d_2 components of $\tilde{v}$. (Here u_f is the "free" part of $\tilde{u}$, relative to $(\operatorname{aff} U_0) - \bar{u}$ being the subspace that indicates the remaining degrees of freedom in the tail of our convergence analysis when the critical face condition is satisfied, whereas u_r is the "restricted" part of $\tilde{u}$.) The projection mappings S_p, $S_p^{\perp}$, S_d, and $S_d^{\perp}$ reduce in this way to the simple projections

$$\tag{4.7} \begin{aligned} &\tilde{S}_p : \begin{pmatrix} \tilde{u}_f \\ \tilde{u}_r \end{pmatrix} \mapsto \begin{pmatrix} \tilde{u}_f \\ 0 \end{pmatrix}, \qquad \tilde{S}_p^{\perp} : \begin{pmatrix} \tilde{u}_f \\ \tilde{u}_r \end{pmatrix} \mapsto \begin{pmatrix} 0 \\ \tilde{u}_r \end{pmatrix}, \\ &\tilde{S}_d : \begin{pmatrix} \tilde{v}_f \\ \tilde{v}_r \end{pmatrix} \mapsto \begin{pmatrix} \tilde{v}_f \\ 0 \end{pmatrix}, \qquad \tilde{S}_d^{\perp} : \begin{pmatrix} \tilde{v}_f \\ \tilde{v}_r \end{pmatrix} \mapsto \begin{pmatrix} 0 \\ \tilde{v}_r \end{pmatrix}. \end{aligned}$$

We partition the columns of the matrix $\tilde{R}$ in accordance with $\tilde{u}$ and the rows in accordance with $\tilde{v}$. Thus,

$$\tag{4.8} \tilde{R} = \begin{pmatrix} \tilde{R}_{ff} & \tilde{R}_{fr} \\ \tilde{R}_{rf} & \tilde{R}_{rr} \end{pmatrix}.$$

In this notation the primal objective function in the transformed problem $(\tilde{\mathcal{P}})$ takes, in an ultimate quadratic region, the simple form

$$\tag{4.9} \begin{aligned} &\tilde{f}(\tilde{u}) = \tfrac{1}{2}(\tilde{u} - \tilde{u}^*)\cdot A(\tilde{u} - \tilde{u}^*) + \text{const.} \quad \text{for some } \tilde{u}^*, \text{ where} \\ &A := I + \left(\tilde{R}_{ff}\ \tilde{R}_{fr}\right)^T \left(\tilde{R}_{ff}\ \tilde{R}_{fr}\right), \end{aligned}$$

while in the dual problem one similarly has

$$\tag{4.10} \begin{aligned} &\tilde{g}(\tilde{v}) = -\tfrac{1}{2}(\tilde{v} - v^*)\cdot B(\tilde{v} - \tilde{v}^*) + \text{const.} \quad \text{for some } \tilde{v}^*, \text{ where} \\ &B := I + \begin{pmatrix} \tilde{R}_{ff} \\ \tilde{R}_{rf} \end{pmatrix} \begin{pmatrix} \tilde{R}_{ff} \\ \tilde{R}_{rf} \end{pmatrix}^T. \end{aligned}$$

In fact, in the notation (4.5) and with $\tilde{U}_0$ and $\tilde{V}_0$ denoting the critical faces $WP^{1/2}U_0$ and $ZQ^{1/2}V_0$ in the transformed problems, one has the expansions

$$\tag{4.11} \tilde{f}(\tilde{u}) = \tilde{f}(\bar{\tilde{u}}) + \tfrac{1}{2}(\tilde{u}_f - \bar{\tilde{u}}_f)\cdot\left(I + \tilde{R}_{ff}^T \tilde{R}_{ff}\right)(\tilde{u}_f - \bar{\tilde{u}}_f) \text{ for } \tilde{u} \in \operatorname{aff} \tilde{U}_0,$$

$$\tag{4.12} \tilde{g}(\tilde{v}) = \tilde{g}(\bar{\tilde{v}}) - \tfrac{1}{2}(\tilde{v}_f - \bar{\tilde{v}}_f)\cdot\left(I + \tilde{R}_{ff}\tilde{R}_{ff}^T\right)(v_f - \bar{\tilde{v}}_f) \text{ for } \tilde{v} \in \operatorname{aff} \tilde{V}_0.$$

It will be helpful to write the Hessian matrices A and B in (4.9) and (4.10) as

$$\tag{4.13} A = \begin{bmatrix} A_{ff} & A_{fr} \\ A_{rf} & A_{rr} \end{bmatrix} = \begin{bmatrix} I + \tilde{R}_{ff}^T \tilde{R}_{ff} & \tilde{R}_{ff}^T \tilde{R}_{fr}, \\ \tilde{R}_{fr}^T \tilde{R}_{ff} & I + \tilde{R}_{fr}^T \tilde{R}_{fr} \end{bmatrix},$$

$$\tag{4.14} B = \begin{bmatrix} B_{ff} & B_{fr} \\ B_{rf} & B_{rr} \end{bmatrix} = \begin{bmatrix} I + \tilde{R}_{ff}\tilde{R}_{ff}^T & \tilde{R}_{ff}\tilde{R}_{rf}^T \\ \tilde{R}_{rf}\tilde{R}_{ff}^T & I + \tilde{R}_{rf}\tilde{R}_{rf}^T \end{bmatrix}.$$

A crucial property of our change of variables $\tilde{u} = WP^{1/2}u$ and $\tilde{v} = ZQ^{1/2}v$ is that

$$\|\tilde{u}\| = \|u\|_P \quad \text{and} \quad \|\tilde{v}\| = \|v\|_Q,$$

and accordingly,

$$\begin{aligned} \|\nabla \tilde{f}(\tilde{u})\| = \|\nabla_P f(u)\|_P \quad &\text{and} \quad \|\nabla \tilde{g}(\tilde{v})\| = \|\nabla_Q g(v)\|_Q, \\ \|\nabla^2 \tilde{f}(\tilde{u})\| = \|\nabla_P^2 f(u)\|_P \quad &\text{and} \quad \|\nabla^2 \tilde{g}(\tilde{v})\| = \|\nabla_Q^2 g(v)\|_Q. \end{aligned}$$

The following result is a strengthening of Proposition 3.1 in the sense that it gives a quantitative estimate for the relationship between $\|u_0 - u_2\|_P$ and $\|u_0 - \bar{u}\|_P$ in the primal, and between $\|v_0 - v_2\|_Q$ and $\|v_0 - \bar{v}\|_Q$ in the dual.

PROPOSITION 4.1 (norm estimates). *Suppose the critical face condition is satisfied. Then for u_0 and v_0 in an ultimate quadratic region for problems $(\mathcal{P})$ and $(\mathcal{Q})$, and with $u_2 := G(F(u_0))$ and $v_2 := F(G(v_0))$, one has*

$$(4.15)\quad (5+4\|\nabla_P^2 f(\bar{u})\|_P^2)^{-\frac{1}{2}}\|u_0-\bar{u}\|_P \le \|u_0-u_2\|_P \le (1+\|\nabla_P^2 f(\bar{u})\|_P^2)^{\frac{1}{2}}\|u_0-\bar{u}\|_P,$$

$$(4.16)\quad (5+4\|\nabla_Q^2 g(\bar{v})\|_Q^2)^{-\frac{1}{2}}\|v_0-\bar{v}\|_Q \le \|v_0-v_2\|_Q \le (1+\|\nabla_Q^2 g(\bar{v})\|_Q^2)^{\frac{1}{2}}\|v_0-\bar{v}\|_Q.$$

Proof. In the transformed coordinates the first equation in Proposition 3.8 gives us $\tilde{u}_2 - \tilde{u}_0 = -\tilde{S}_p\big(\nabla^2\tilde{f}(\bar{\tilde{u}})(\tilde{u}_0 - \bar{\tilde{u}})\big) - \tilde{S}_p^{\perp}(\tilde{u}_0 - \bar{\tilde{u}})$. In the notation (4.13) for $\nabla^2 f(\bar{u})$ this gives

$$\begin{aligned}\|\tilde{u}_0 - \tilde{u}_2\|^2 &= \|\tilde{S}_p\big(A(\tilde{u}_0 - \bar{\tilde{u}})\big)\|^2 + \|\tilde{S}_p^{\perp}(\tilde{u}_0 - \bar{\tilde{u}})\|^2 \\ &\le \|A(\tilde{u}_0 - \bar{\tilde{u}})\|^2 + \|\tilde{u}_0 - \bar{\tilde{u}}\|^2 \le (\|A\|^2 + 1)\,\|\tilde{u}_0 - \bar{\tilde{u}}\|^2.\end{aligned}$$

This gives the right half of (4.15). To get the left half, decompose $\tilde{u}_0 - \bar{\tilde{u}}$ into $\mu_1\xi + \mu_2\eta$, where ξ is a unit vector in the null space of $(A_{ff}\ A_{fr})$ while η is a unit vector in the orthogonal complement of that null space, and the direction of η is so chosen that $\mu_2 > 0$. Partition ξ and η as well:

$$\xi = \begin{pmatrix}\xi_f\\ \xi_r\end{pmatrix}, \qquad \eta = \begin{pmatrix}\eta_f\\ \eta_r\end{pmatrix}.$$

It follows from $(A_{ff}\ A_{fr})\xi = A_{ff}\xi_f + A_{fr}\xi_r = 0$ that $\xi_f = -A_{ff}^{-1}A_{fr}\xi_r$ and

$$\|\xi_f\|^2 \le \|A_{ff}^{-1}\|^2\|A_{fr}\|^2\|\xi_r\|^2 \le \|A_{fr}\|^2\|\xi_r\|^2,$$

because the smallest eigenvalue of A_{ff} is no less than 1. Therefore,

$$\|\xi\|^2 = \|\xi_f\|^2 + \|\xi_r\|^2 \le (1+\|A_{fr}\|^2)\|\xi_r\|^2 \quad\Rightarrow\quad \|\xi_r\|^2 \ge \frac{1}{1+\|A_{fr}\|^2} \ge \frac{1}{1+\|A\|^2}.$$

Denote $\|\tilde{u}_0 - \bar{\tilde{u}}\|$ by κ. We get

$$\|\mu_1\xi_r + \mu_2\eta_r\| \ge \mu_1\|\xi_r\| - \mu_2\|\eta_r\| \ge (\kappa^2 - (\mu_2)^2)^{1/2}(1+\|A\|^2)^{-1/2} - \mu_2.$$

Recalling that all the eigenvalues of A_{ff} are no less than 1, we obtain

$$\begin{aligned}\|\tilde{u}_0 - \tilde{u}_2\|^2 &= \|(A_{ff}\ A_{fr})\mu_2\eta\|^2 + \|\mu_1\xi_r + \mu_2\eta_r\|^2 \\ &\ge \mu_2^2 + \big(\max\{0, (\kappa^2 - \mu_2^2)^{1/2}(1+\|A\|^2)^{-1/2} - \mu_2\}\big)^2.\end{aligned}$$

But the term $(\kappa^2 - \mu_2^2)^{1/2}(1+\|A\|^2)^{-1/2} - \mu_2$ decreases monotonically as μ_2 increases from 0. This term equals $\bar{\mu}_2 := (5+4\|A\|^2)^{-1/2}\kappa$ when $\mu_2 = \bar{\mu}_2$. Therefore, $\|\tilde{u}_0 - \tilde{u}_2\|^2 \ge (\bar{\mu}_2)^2$, from which the left half of (4.15) follows. The proof of (4.16) is similar. □

THEOREM 4.2 (rate of convergence of PDSD). *Consider Algorithm* 1 *in the case of threshold* $\varepsilon = 0$, *and suppose the critical face condition is satisfied. In terms of* $\gamma := \gamma(P, Q, R) := \|Q^{-1/2}RP^{-1/2}\|$, *let*

$$c_1 := 1 - \frac{1}{(1+\gamma^2)\left[2 + 5(1+\gamma^2) + 4(1+\gamma^2)^3\right]} < 1, \tag{4.17}$$

$$c_2 := \left(1 - \frac{1}{1+\gamma^2/2}\right)^2 < 1. \tag{4.18}$$

Unless the algorithm actually terminates in a finite number of iterations with $(\hat{u}, \hat{v}) = (\bar{u}, \bar{v})$, *the sequences* $\{f(u_0^\nu)\}$ *and* $\{g(v_0^\nu)\}$ *generated by it converge linearly to the common optimal value* $f(\bar{u}) = g(\bar{v})$ *in the sense that*

$$\limsup_{\nu\to\infty} \frac{f(u_0^{\nu+1}) - f(\bar{u})}{f(u_0^\nu) - f(\bar{u})} \le c_1 \quad \textit{and} \quad \limsup_{\nu\to\infty} \frac{g(v_0^{\nu+1}) - g(\bar{v})}{g(v_0^\nu) - g(\bar{v})} \le c_1. \tag{4.19}$$

Moreover, let $\bar{\nu}$ *be an iteration number such that for* $\nu \ge \bar{\nu}$ *all the points* u_0^ν, u_2^ν *and* v_0^ν, v_2^ν *are in an ultimate quadratic region in Definition* 3.7. *Then once* $u_0^{\nu'} \in U_0$ *for some* $\nu' \ge \bar{\nu}$ (*as is sure to happen in an interactive primal restart at that stage*) *one has*

$$\frac{f(u_0^{\nu+1}) - f(\bar{u})}{f(u_0^\nu) - f(\bar{u})} \le c_2 \quad \forall \nu \ge \nu', \tag{4.20}$$

and similarly, once $v_0^{\nu''} \in V_0$ *for some* $\nu'' \ge \bar{\nu}$ (*as is sure to happen in an interactive dual restart at that stage*) *one has*

$$\frac{g(v_0^{\nu+1}) - g(\bar{v})}{g(v_0^\nu) - g(\bar{v})} \le c_2 \quad \forall \nu \ge \nu''. \tag{4.21}$$

Proof. Under the assumption that the algorithm does not terminate after a finite number of iterations at $(\bar{u}, \bar{v})$, neither u_0^ν nor v_0^ν is optimal, as we have shown in the proof of Proposition 3.1(b).

Again we work in the transformed coordinates. Consider $\nu \ge \bar{\nu}$, i.e., the sequences $\{\tilde{u}_0^\nu\}$, $\{\tilde{u}_2^\nu\}$ and $\{\tilde{v}_0^\nu\}$, $\{\tilde{v}_2^\nu\}$ have entered the ultimate quadratic region. With respect to the direction vector $d^\nu := \tilde{u}_2^\nu - \tilde{u}_0^\nu$, the optimal step length $\bar{\lambda}^\nu$ for $\tilde{u} = \tilde{u}_0^\nu + \lambda d^\nu$ to minimize the quadratic form (4.9) over all $\lambda \in [0, \infty)$ can be written as

$$\begin{aligned} \bar{\lambda}^\nu &= \frac{-d^\nu \cdot A(\tilde{u}_0^\nu - \tilde{u}^*)}{d^\nu \cdot A d^\nu} \\ &= \frac{\left[\tilde{S}_p A(\tilde{u}_0^\nu - u^*) + \tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}})\right] \cdot A(\tilde{u}_0^\nu - \tilde{u}^*)}{\left[\tilde{S}_p A(\tilde{u}_0^\nu - \tilde{u}^*) + \tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}})\right] \cdot A\left[\tilde{S}_p A(\tilde{u}_0^\nu - \tilde{u}^*) + \tilde{S}_p^\perp(\tilde{u}_0^\nu - \bar{\tilde{u}})\right]}. \end{aligned} \tag{4.22}$$

In the following, we first show that $\bar{\lambda}^\nu \le 1$. Then the search on $[\tilde{u}_0^\nu, \tilde{u}_2^\nu]$ in Step 5 of the algorithm is equivalent to a search on the corresponding half-line (or is "perfect," for short), and there exist easy ways to estimate progress in the line search step. By Proposition 3.5 (cf. also the remark afterward), we have the following.

Case 1. There exists some $\nu' \ge \bar{\nu}$ such that $\tilde{u}_0^\nu \in \tilde{U}_0$ for all $\nu \ge \nu'$.

Case 2. $\tilde{u}_0^\nu \notin \tilde{U}_0$ for all $\nu \ge \bar{\nu}$.

In Case 1 the equation $\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}}) = 0$ holds for all $\nu \geq \nu'$. Then it follows from (4.22) that

$$\bar{\lambda}^\nu = \frac{\tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big)\cdot A(\tilde{u}_0^\nu - \tilde{u}^*)}{\tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big)\cdot A\tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big)} = \frac{\tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big)\cdot \tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big)}{\tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big)\cdot A\tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big)} \leq 1,$$

because all the eigenvalues of A are at least 1. Now Step 5 of the algorithm must coincide with the steepest descent method for $\tilde{f}$ on aff $\tilde{U}_0$ with "perfect" line search, since $[\tilde{u}_0^\nu, \tilde{u}_2^\nu]$ is in an ultimate quadratic region of the problem. Note that all the eigenvalues of the Hessian matrix A_{ff} are in the interval $[1, 1 + \|\tilde{R}_{ff}\|^2]$, where $\|\tilde{R}_{ff}\|^2 \leq \|\tilde{R}\|^2 = \gamma^2$. Hence by using the expression of $\tilde{f}$ in (4.11), we have [22]

$$\text{(4.23)} \qquad \frac{\tilde{f}(\hat{\tilde{u}}_0^{\nu+1}) - \tilde{f}(\bar{\tilde{u}})}{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\bar{\tilde{u}})} \leq \left(\frac{\|\tilde{R}_{ff}\|^2}{\|\tilde{R}_{ff}\|^2 + 2}\right)^2 \leq \left(1 - \frac{1}{1 + \frac{1}{2}\|\tilde{R}\|^2}\right)^2,$$

which yields (4.20) since $\tilde{f}(\tilde{u}_0^{\nu+1}) \leq f(\hat{\tilde{u}}_0^{\nu+1})$ in the algorithm.

In Case 2 we have $\bar{\lambda}^\nu < 1$ for all $\nu \geq \bar{\nu}$, since otherwise $\tilde{u}_2^\nu$ would be taken as the next point $\tilde{u}_0^{\nu+1}$ and the iteration would be on the critical face $\tilde{U}_0$ thereafter. Hence the line search restricted to $[\tilde{u}_0^\nu, \tilde{u}_2^\nu]$ is again "perfect." On exiting from the line search in Step 5, we have

$$\begin{aligned}\frac{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\hat{\tilde{u}}_0^{\nu+1})}{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\bar{\tilde{u}})} &= \frac{(\bar{\lambda}^\nu)^2 d^\nu\cdot Ad^\nu}{2\big[\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\bar{\tilde{u}})\big]} \\ &= \frac{\big[A(\tilde{u}_0^\nu - \tilde{u}^*)\cdot\tilde{S}_p\big(A(\tilde{u}_0^\nu - \tilde{u}^*)\big) + A(\tilde{u}_0^\nu - \tilde{u}^*)\cdot\tilde{S}_p^{\perp}(u_0^\nu - \bar{u})\big]^2}{(d^\nu\cdot Ad^\nu)\big[(\tilde{u}_0^\nu - \tilde{u}^*)\cdot A(\tilde{u}_0^\nu - \tilde{u}^*) - (\bar{\tilde{u}} - \tilde{u}^*)\cdot A(\bar{\tilde{u}} - \tilde{u}^*)\big]} \\ &= \frac{\big[d^\nu\cdot d^\nu - \big(\tilde{u}_0^\nu - \bar{\tilde{u}} - A(\tilde{u}_0^\nu - \tilde{u}^*)\big)\cdot\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})\big]^2}{(d^\nu\cdot Ad^\nu)\big[(\tilde{u}_0^\nu - \bar{\tilde{u}})\cdot A(\tilde{u}_0^\nu - \bar{\tilde{u}}) + 2(\tilde{u}_0^\nu - \bar{\tilde{u}})\cdot A(\bar{\tilde{u}} - \tilde{u}^*)\big]}.\end{aligned}$$

Defining $b(\tilde{u}) := \tilde{u} - \bar{\tilde{u}} - A(\tilde{u} - \tilde{u}^*)$ and observing $\tilde{f}(\tilde{u}_0^{\nu+1}) \leq \tilde{f}(\hat{\tilde{u}}_0^{\nu+1})$, we obtain from the equation $\tilde{S}_p\big(A(\bar{\tilde{u}} - \tilde{u}^*)\big) = 0$ (which is based on the optimality of $\hat{\tilde{u}}$) that

$$\begin{aligned}\frac{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\tilde{u}_0^{\nu+1})}{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\bar{\tilde{u}})} &\geq \frac{\big[d^\nu\cdot d^\nu + b(u_0^\nu)\cdot\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})\big]^2}{(d^\nu\cdot Ad^\nu)\big[(\tilde{u}_0^\nu - \bar{\tilde{u}})\cdot A(\tilde{u}_0^\nu - \bar{\tilde{u}}) - 2b(\bar{\tilde{u}})\cdot\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})\big]} \\ &\geq \frac{(d^\nu\cdot d^\nu)^2 + \big[b(\tilde{u}_0^\nu)\cdot\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})\big]^2}{(d^\nu\cdot Ad^\nu)\big[(\tilde{u}_0^\nu - \bar{\tilde{u}})\cdot A(\tilde{u}_0^\nu - \bar{\tilde{u}}) - 2b(\bar{\tilde{u}})\cdot\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})\big]}.\end{aligned}$$

By Theorem 3.2 the algorithm converges, hence for arbitrarily chosen $\tilde{\varepsilon} > 0$, we have $\|b(\tilde{u}_0^\nu) - b(\bar{\tilde{u}})\| \leq \tilde{\varepsilon}$ for sufficiently large ν. Then

$$\begin{aligned}|b(\tilde{u}_0^\nu)\cdot\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})| &= |b(\bar{\tilde{u}})\cdot\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}}) + \big(b(\tilde{u}_0^\nu) - b(\bar{\tilde{u}})\big)\cdot\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})| \\ &\geq |b(\bar{\tilde{u}})\cdot\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})| - |\big(b(\tilde{u}_0^\nu) - b(\bar{\tilde{u}})\big)\cdot\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})| \\ &\geq |b(\bar{\tilde{u}})\cdot\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})| - \tilde{\varepsilon}\|\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})\|,\end{aligned}$$

but $|b(\bar{\tilde{u}})\cdot\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})| = \|\nabla\tilde{f}(\bar{\tilde{u}})\|\cdot\|\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})\|$. Therefore,

$$\text{(4.24)} \qquad \frac{|b(\tilde{u}_0^\nu)\cdot\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})|}{|b(\bar{\tilde{u}})\cdot\tilde{S}_p^{\perp}(\tilde{u}_0^\nu - \bar{\tilde{u}})|} \geq 1 - \frac{\tilde{\varepsilon}}{\|\nabla\tilde{f}(\bar{\tilde{u}})\|},$$

where $\nabla\tilde{f}(\tilde{\bar{u}}) \neq 0$, for otherwise $\tilde{U}_0 = \tilde{U}$, and then $\tilde{u}_0^\nu \in \tilde{U}_0$ in contradiction to our assumption in Case 2.

Now, if $d^\nu \cdot d^\nu \geq -b(\tilde{u}_0^\nu)\cdot\tilde{S}_p^\perp(\tilde{u}_0^\nu - \tilde{\bar{u}})$, we obtain from (4.15) that

$$\begin{aligned}\frac{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\tilde{u}_0^{\nu+1})}{\tilde{f}(\tilde{u}_0^\nu) - f(\tilde{\bar{u}})} &\geq \frac{(d^\nu \cdot d^\nu)^2}{(d^\nu \cdot A d^\nu)\left[(\tilde{u}_0^\nu - \tilde{\bar{u}})\cdot A(\tilde{u}_0^\nu - \tilde{\bar{u}}) + 2 d^\nu \cdot d^\nu\right]} \\ &\geq \frac{1}{\|A\|\left[2 + \|A\|(5 + 4\|A\|^2)\right]}.\end{aligned}$$

Otherwise $d^\nu \cdot d^\nu < -b(\tilde{u}_0^\nu)\cdot\tilde{S}_p^\perp(\tilde{u}_0^\nu - \tilde{\bar{u}})$, and then

$$\begin{aligned}&\frac{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\tilde{u}_0^{\nu+1})}{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\tilde{\bar{u}})} \\ &\quad\geq \frac{\left[b(\tilde{u}_0^\nu)\cdot\tilde{S}_p^\perp(\tilde{u}_0^\nu - \tilde{\bar{u}})\right]^2}{\|A\|\left[b(\tilde{\bar{u}})\cdot\tilde{S}_p^\perp(\tilde{u}_0^\nu - \tilde{\bar{u}})\right]\left[\|A\|(5 + 4\|A\|^2) b(\tilde{\bar{u}})\cdot\tilde{S}_p^\perp(\tilde{u}_0^\nu - \tilde{\bar{u}}) + 2 b(\tilde{\bar{u}})\cdot\tilde{S}_p^\perp(\tilde{u}_0^\nu - \tilde{\bar{u}})\right]} \\ &\quad= \frac{1}{\|A\|\left(2 + \|A\|(5 + 4\|A\|^2)\right)}\left(1 - \frac{\tilde{\varepsilon}}{\|\nabla\tilde{f}(\tilde{\bar{u}})\|}\right)\end{aligned}$$

by (4.24). 'Thus, we have

$$\liminf_{\nu\to\infty} \frac{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\tilde{u}_0^{\nu+1})}{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\tilde{\bar{u}})} \geq \frac{1}{\|A\|\left(2 + \|A\|(5 + 4\|A\|^2)\right)},$$

which can be written as

$$\limsup_{\nu\to\infty} \frac{\tilde{f}(\tilde{u}_0^{\nu+1}) - \tilde{f}(\tilde{\bar{u}})}{\tilde{f}(\tilde{u}_0^\nu) - \tilde{f}(\tilde{\bar{u}})} \leq 1 - \frac{1}{\|A\|\left(2 + \|A\|(5 + 4\|A\|^2)\right)}. \tag{4.25}$$

Noting that $\|A\| = 1 + \|(\tilde{R}_{ff}\ \tilde{R}_{fr})\|^2 \leq 1 + \|\tilde{R}\|^2 = 1 + \gamma^2$, we get the first inequality in (4.19), which is also true for Case 1 in view of (4.20) since $c_2 < c_1$. The dual part has a parallel argument. □

Observe that the rates in (4.20) and (4.21) are much better than the ones in (4.19). The former will be effective if any interactive restarts occur for $\nu \geq \bar{\nu}$, as indicated in the theorem. This partially explains the effects of interactive restarts on the algorithm as observed in our numerical tests.

The role of the constant $\gamma = \gamma(P, Q, R)$ in the convergence rate in Theorem 4.2 has been borne out in our numerical tests, although because of the interactive restarts the method appears to work much better than one might expect from "steepest descent." We have definitely observed in small-scale problems where some idea of the size of γ is available that the convergence is faster with low γ than with high γ.

Although Theorem 4.2 centers on the specialization of Algorithm 0 to Algorithm 1, the argument has content also for Algorithm 2. Recall from the discussion after the statement of Algorithm 0 in §2 that in every k iterations of Algorithm 0 (when implemented with cycle size $k > 1$) there is at least one primal line search on $[u_0^\nu, u_2^\nu]$ and at least one dual line search on $[v_0^\nu, v_2^\nu]$. This gives us the following result about Algorithm 2, which will be complemented by a finite termination result in Theorem 4.5.

COROLLARY 4.3 (rate of convergence of PDCG). *Suppose the critical face condition is satisfied. Then Algorithm* 2 *with* $\varepsilon = 0$ *converges at least k-step linearly in the sense that*

$$\limsup_{\nu\to\infty} \frac{f(u_0^{\nu+k}) - f(\bar{u})}{f(u_0^{\nu}) - f(\bar{u})} \le c_1 \text{ and } \limsup_{\nu\to\infty} \frac{g(v_0^{\nu+k}) - g(\bar{v})}{g(v_0^{\nu}) - g(\bar{v})} \le c_1, \tag{4.26}$$

where c_1 *is the value defined in* (4.17), *unless the algorithm terminates after a finite number of iterations with* $(\hat{u}, \hat{v}) = (\bar{u}, \bar{v})$.

To derive a special finite termination property of Algorithm 2, we need the following.

PROPOSITION 4.4 (inequalities in PDCG). *Suppose the critical face condition is satisfied. Let* $\hat{\nu}$ *be an iteration number such that for* $\nu \ge \hat{\nu}$, *all the points* u_0^ν, u_2^ν *and* v_0^ν, v_2^ν *are in an ultimate quadratic region* $U^* \times V^*$ *in Definition* 3.7, *where* U^* *is contained in the* $\|\cdot\|_P$*-ball around* $\bar{u}$ *of radius* $\frac{1}{2}$, *and likewise* V^* *is contained in the* $\|\cdot\|_Q$*-ball around* $\bar{v}$ *of radius* $\frac{1}{2}$. *If* $u_0^{\nu'} \in U_0$ *for some* $\nu' \ge \hat{\nu}$, *then in Algorithm* 2 *one has*

$$\langle w_p^\nu, u_e^{\nu-1} - u_0^\nu \rangle_P > 0 \tag{4.27}$$

whenever (2.1)–(2.4) *are used to generate* u_e^ν *for* $\nu > \nu'$, *and similarly if* $v_0^{\nu''} \in V_0$ *for some* $\nu'' \ge \hat{\nu}$, *then in Algorithm* 2 *one has*

$$\langle w_d^\nu, v_e^{\nu-1} - v_0^\nu \rangle_Q > 0 \tag{4.28}$$

whenever (2.5)–(2.8) *are used to generate* v_e^ν *for* $\nu > \nu''$.

Proof. It suffices once more to give the argument in the context of the transformed variables. Observe that the gradient mapping $\nabla\tilde{f}$ is strongly monotone, and that $\tilde{w}_p^\nu = \nabla\tilde{f}(\tilde{u}_0^\nu) - \nabla\tilde{f}(\tilde{u}_0^{\nu-1})$ with $\tilde{u}_0^\nu \in [\tilde{u}_0^{\nu-1}, \tilde{u}_e^{\nu-1}]$ when (2.1)–(2.4) are used to generate u_e^ν in the primal. Hence the primal part of the assertion is true if $\tilde{u}_0^\nu \ne \tilde{u}_e^{\nu-1}$ for $\nu > \nu'$. According to Proposition 3.5 (cf. also the remark after it), one has $\tilde{u}_0^\nu \in \tilde{U}_0$ for all $\nu \ge \nu'$. We partition all vectors in conformity with the scheme in (4.6). Then $\tilde{u}_{0,r}^\nu = \bar{\tilde{u}}_r$ and $\tilde{u}_{2,r}^\nu = \bar{\tilde{u}}_r$.

If the $(\nu-1)$th iteration with $\nu > \nu'$ is the first iteration of a primal cycle, then the line search is performed on $[\tilde{u}_0^{\nu-1}, \tilde{u}_2^{\nu-1}]$. For the direction vector $d^{\nu-1} := \tilde{u}_2^{\nu-1} - \tilde{u}_0^{\nu-1}$, the optimal step length $\bar{\lambda}^\nu$ for $\tilde{u} = \tilde{u}_0^\nu + \lambda d^\nu$ to minimize the quadratic form (4.9) over all $\lambda \in [0, \infty)$ can be derived from the expression in (4.11) as

$$\bar{\lambda}^{\nu-1} = \frac{-d^{\nu-1}\cdot\nabla\tilde{f}(\tilde{u}_0^{\nu-1})}{d^{\nu-1}\cdot A d^{\nu-1}} = \frac{d_f^{\nu-1}\cdot d_f^{\nu-1}}{d_f^{\nu-1}\cdot A_{ff} d_f^{\nu-1}},$$

where the first equation in Proposition 3.8 has been used with $\nabla\tilde{f}(\tilde{u}_0^{\nu-1})$, and A_{ff} is the Hessian component in (4.13). Note that none of the eigenvalues of A_{ff} is less than 1. Hence $\bar{\lambda}^{\nu-1} \le 1$, and the equality holds only if $d_f^{\nu-1}$ is an eigenvector corresponding to 1 as an eigenvalue of A_{ff}, i.e., $A_{ff} d_f^{\nu-1} = d_f^{\nu-1}$. But it follows from (4.11) and the first equation in Proposition 3.8 that we also have $A_{ff}(\bar{\tilde{u}}_f - \tilde{u}_{0,f}^{\nu-1}) = d_f^{\nu-1}$; therefore, $\bar{\lambda}^{\nu-1} = 1$ implies $\tilde{u}_{2,f}^{\nu-1} = \bar{\tilde{u}}_f$ and $\tilde{u}_2^{\nu-1} = \bar{\tilde{u}}$. And then $\tilde{u}_0^\nu = \bar{\tilde{u}}$, i.e., the iteration terminates at the primal optimal solution.

If the $(\nu-1)$th iteration with $\nu > \nu'$ is not the first iteration of a primal cycle, then formulas (2.1)–(2.4) are used to define $\tilde{u}_e^{\nu-1}$. In the proof of Proposition 3.5, we have actually shown that $\tilde{u}_e^{\nu-1} \in \tilde{U}_0$ for all $\nu > \nu'$. Hence $[\tilde{u}_0^{\nu-1}, \tilde{u}_e^{\nu-1}] \subset \tilde{U}_0$ for all $\nu > \nu'$. Then it follows from (2.4) that $\|\tilde{u}_e^{\nu-1} - \tilde{u}_0^{\nu-1}\| \geq 1$ unless $\tilde{u}_e^{\nu-1}$ is on the relative boundary of $\tilde{U}_0$. In either case we have $\tilde{u}_0^{\nu} \neq \tilde{u}_e^{\nu-1}$ again, since $\tilde{u}_0^{\nu-1} \in \tilde{U}^*$ for $\nu > \nu'$ and $\tilde{U}^*$ is contained in the $\|\cdot\|_P$-ball around $\bar{u}$ of radius $\frac{1}{2}$. The dual claims can be verified similarly. □

THEOREM 4.5 (a finite termination property of PDCG). *Assume that the critical face condition is satisfied. Suppose that the cycle size k chosen in Algorithm 2 is such that $k > \bar{k}$, where $\bar{k}$ denotes the rank of the linear transformation $u \mapsto S_d\big(RS_p(u)\big)$. (It suffices in this to have $k > \min\{m, n\}$.) Let $\hat{\nu}$ be an iteration number as defined in Proposition 4.4 and satisfying the conditions there. If $u_0^{\nu'} \in U_0$ for some $\nu' \geq \hat{\nu}$ (as is sure to happen in an interactive primal restart at that stage), then the algorithm will terminate in the next full primal cycle, if not earlier. Similarly, if $v_0^{\nu''} \in V_0$ for some $\nu'' \geq \hat{\nu}$ (as is sure to happen in an interactive dual restart at that stage), then the algorithm will terminate in the next full dual cycle, if not earlier.*

Proof. We concentrate on the primal part; the proof of the dual part is parallel. In the transformed variables, where we place the argument once more, $\bar{k}$ is the rank of the submatrix $\tilde{R}_{ff}$ of $\tilde{R}$ in (4.8). Note that for $\nu \geq \hat{\nu}$ the process is in a quadratic region of the problem as specified in Proposition 4.4. In the proof of Proposition 4.4, we have shown that for all $\nu \geq \nu'$, $[\tilde{u}_0^{\nu}, \tilde{u}_e^{\nu}] \subset \tilde{U}_0$, and that the line searches on $[\tilde{u}_0^{\nu}, \tilde{u}_e^{\nu}]$ are "perfect" in the sense that, on exiting Step 5 of iteration ν, $\tilde{u}_0^{\nu+1}$ minimizes $\tilde{f}$ on the half-line from $\tilde{u}_0^{\nu}$ in the direction of $\tilde{u}_e^{\nu} - \tilde{u}_0^{\nu}$. Observe there is no interactive primal restart in the first $k-1$ iterations of a full primal cycle, i.e., $\hat{\tilde{u}}_0^{\nu} = \tilde{u}_0^{\nu}$ for these iterations. We claim now that the search direction vectors $\tilde{u}_e^{\nu} - \tilde{u}_0^{\nu}$ and $\tilde{v}_e^{\nu} - \tilde{v}_0^{\nu}$ are the same as the ones that would be generated by a conjugate gradient algorithm on $\tilde{f}$ relative to $\operatorname{aff} \tilde{U}_0$. The finite termination property will be a consequence of observing that the Hessians of $\tilde{f}$ in an quadratic region of the problem (cf. (4.11)) have at most $\bar{k} + 1$ different eigenvalues.

The proof of the claim will go by induction. We know from Proposition 3.8 that the claim is true for the first iteration of the full primal cycle in question. Suppose it is true for the $(\nu-1)$th iteration generating $\tilde{u}_0^{\nu}$ in that cycle, but $\tilde{u}_0^{\nu} \neq \bar{\tilde{u}}$. Then by (4.27) in Proposition 4.4, the first alternative of (2.2) will be used to generate β_p^{ν}. Hence it follows from (2.1)–(2.3) and Proposition 3.8 that

(4.29)

$$(\tilde{u}_{cg}^{\nu} - \tilde{u}_0^{\nu})_f = \frac{(\tilde{u}_2^{\nu} - \tilde{u}_0^{\nu})_f + \beta_p^{\nu}(\tilde{u}_e^{\nu-1} - \tilde{u}_0^{\nu})_f}{1 + \beta_p^{\nu}} = \frac{-\nabla \tilde{f}(\tilde{u}_0^{\nu})_f + \beta_p^{\nu}(\tilde{u}_e^{\nu-1} - \tilde{u}_0^{\nu})_f}{1 + \beta_p^{\nu}},$$

$$\beta_p^{\nu}(\tilde{u}_e^{\nu-1} - \tilde{u}_0^{\nu})_f = \frac{\max\{0,\ (\nabla \tilde{f}(u_0^{\nu})_f - \nabla \tilde{f}(\tilde{u}_0^{\nu-1})_f) \cdot \nabla \tilde{f}(\tilde{u}_0^{\nu})_f\}}{(\nabla \tilde{f}(\tilde{u}_0^{\nu})_f - \nabla \tilde{f}(\tilde{u}_0^{\nu-1})_f) \cdot (\tilde{u}_e^{\nu-1} - \tilde{u}_0^{\nu})_f} (\tilde{u}_e^{\nu-1} - \tilde{u}_0^{\nu})_f,$$

where all the points $\tilde{u}_e^{\nu-1}$, $\tilde{u}_0^{\nu}$, and $\tilde{u}_2^{\nu}$ are on the critical face $\tilde{U}_0$. By the induction hypothesis, the directions of line search are the same as the ones generated by the conjugate gradient algorithm in all the previous iterations of the cycle. Hence

$$\nabla \tilde{f}(\tilde{u}_0^{\nu})_f \cdot \nabla \tilde{f}(\tilde{u}_0^{\nu-1})_f = 0,$$

which implies $(\nabla \tilde{f}(\tilde{u}_0^{\nu})_f - \nabla \tilde{f}(\tilde{u}_0^{\nu-1})_f) \cdot \nabla \tilde{f}(\tilde{u}_0^{\nu})_f \geq 0$; therefore, by noting that $\tilde{u}_e^{\nu-1} - \tilde{u}_0^{\nu}$ is a positive multiple of $\tilde{u}_0^{\nu} - \tilde{u}_0^{\nu-1}$, we obtain

$$\beta_p^{\nu}(\tilde{u}_e^{\nu-1} - \tilde{u}_0^{\nu})_f = \frac{(\nabla \tilde{f}(\tilde{u}_0^{\nu})_f - \nabla \tilde{f}(\tilde{u}_0^{\nu-1})_f) \cdot \nabla \tilde{f}(\tilde{u}_0^{\nu})_f}{(\nabla \tilde{f}(\tilde{u}_0^{\nu})_f - \nabla \tilde{f}(\tilde{u}_0^{\nu-1})_f) \cdot (\tilde{u}_0^{\nu-1} - \tilde{u}_0^{\nu})_f} (\tilde{u}_0^{\nu-1} - \tilde{u}_0^{\nu})_f. \tag{4.30}$$

Comparing (4.29) and (4.30) with the conjugate gradient formulas of Hestenes and Stiefel [22], we see that the vector $\tilde{u}^{\nu}_{cg} - \tilde{u}^{\nu}_{0}$ is equivalent to the search direction vector in a standard conjugate gradient algorithm for $\tilde{f}$ relative to the free variables, i.e., over $\operatorname{aff} \tilde{U}_0$. □

Observe that the rank of linear transformation in Theorem 4.5 is bounded above by the ranks of the projection mappings S_p and S_d, which are $\dim U_0$ and $\dim V_0$. Hence

$$\bar{k} \leq \min\{\dim U_0, \dim V_0\}.$$

Therefore, even in the case that the original problems $(\mathcal{P})$ and $(\mathcal{Q})$ are of high dimension, the optimal solution can still be reached in a relatively short cycle after entering an ultimate quadratic region for the problem if merely one of the critical faces U_0 and V_0 happens to be of low dimension, provided that at least one of the critical faces is eventually reached by the corresponding iterates. This condition will certainly be satisfied if any interactive restarts occur for $\nu \geq \hat{\nu}$, since all points $\hat{u}^{\nu}_1$ and $\hat{v}^{\nu}_1$ will be on the critical faces U_0 and V_0 by Proposition 1.8, and once u^{ν}_0 or v^{ν}_0 are on the critical faces, they will stay there (Proposition 3.5).

There are ways to force this condition to be satisfied, such as to insert at the beginning of each primal cycle a line search in the direction of the projection of $-\nabla f(u^{\nu}_0)$ on the tangent cone to U at u^{ν}_0, and similarly in the dual. (See Burke and Moré [23].) But even without such remedies, we often find in our test problems that the critical faces are identified in the tail of iteration, and that restarts do occur in most cases, after which the iteration terminates at the optimal solution in a few steps.

5. Envelope properties. To finish off, we establish two results on the finite-envelope property of the points u^{ν}_1 and v^{ν}_1 in our algorithms.

PROPOSITION 5.1 (general saddle point property of iterates). *On exiting from Step* 5 *of Algorithm* 0 *with* $\hat{u}^{\nu+1}_0$ *and* $\hat{v}^{\nu+1}_0$, *the elements* $\hat{u}^{\nu+1}_1 \in G(\hat{v}^{\nu+1}_0)$ *and* $\hat{v}^{\nu+1}_1 \in F(\hat{u}^{\nu+1}_0)$ *that will be calculated on return to Step* 1 *will be such that the pair* $(\hat{u}^{\nu+1}_0, \hat{v}^{\nu+1}_1)$ *is the unique saddle point of* $L(u,v)$ *on* $[u^{\nu}_0, u^{\nu}_2] \times V$, *while the pair* $(\hat{u}^{\nu+1}_1, \hat{v}^{\nu+1}_0)$ *is the unique saddle point of* $L(u,v)$ *on* $U \times [v^{\nu}_0, v^{\nu}_2]$. *In particular,* $\hat{u}^{\nu+1}_1$ *will be the unique minimizing point relative to* U *for the envelope function*

$$f^{\nu}(u) := \max_{v \in [v^{\nu}_0, v^{\nu}_2]} L(u,v) \leq \max_{v\in V} L(u,v) = f(u),$$

whereas $\hat{v}^{\nu+1}_1$ *will be the unique maximizing point relative to* V *for the envelope function*

$$g^{\nu}(u) := \min_{u \in [u^{\nu}_0, u^{\nu}_2]} L(u,v) \geq \min_{u\in U} L(u,v) = g(u).$$

Proof. Recall that because we are in the fully quadratic case, $L(u,v)$ and $f(u)$ are strictly convex in u, while $L(u,v)$ and $g(v)$ are strictly concave in v. In particular, $\hat{u}^{\nu+1}_0$ must be the unique solution to the problem in Step 5 of minimizing $f(u)$ over $u \in [u^{\nu}_0, u^{\nu}_2]$. This is the primal problem of extended linear-quadratic programming that corresponds to L on $[u^{\nu}_0, u^{\nu}_2] \times V$ instead of $U \times V$. Applying Theorem 1.1 to it instead of to the original problem we deduce the existence of a vector v' such that $(\hat{u}^{\nu+1}_0, v')$ is a saddle point of L relative to $[u^{\nu}_0, u^{\nu}_2] \times V$. Then v' is the unique point maximizing $L(\hat{u}^{\nu+1}_0, v)$ with respect to $v \in V$ (by the strict concavity of $L(u,v)$ in v). Thus, v' is the unique element of $F(\hat{u}^{\nu+1}_0)$, so $v' = \hat{v}^{\nu+1}_1$. It follows from Theorem 1.1 again that $(\hat{u}^{\nu+1}_0, \hat{v}^{\nu+1}_1)$ is the unique saddle point of $L(u,v)$ on $[u^{\nu}_0, u^{\nu}_2] \times V$, and $\hat{v}^{\nu+1}_1$ is the unique solution to the corresponding dual problem, which by definition consists of maximizing the function g^{ν} over V.

The rest of the assertions are true by a parallel argument in which Theorem 1.1 is applied to the primal and dual problems that correspond to L on $U \times [v_0^\nu, v_2^\nu]$. □

PROPOSITION 5.2 (ultimate saddle point property of iterates). *Suppose the critical face condition is satisfied. Let $\hat\nu$ be an iteration number as specified in Proposition 4.4 and satisfying the conditions there. If $\nu = r \geq \hat\nu$ is the first iteration of some primal cycle with $v_0^r \in U_0$, then for all $\nu \geq r$ in that cycle, on exiting from Step* 5 *of Algorithm* 2 *(as implementing Algorithm* 0*) with $\hat u_0^{\nu+1}$ the element $\hat v_1^{\nu+1} \in F(\hat u_0^{\nu+1})$ that will be calculated on return to Step* 1 *will be such that $(\hat u_0^{\nu+1}, \hat v_1^{\nu+1})$ is the unique saddle point of $L(u,v)$ on $U^\nu \times V$, where*

$$(5.1) \qquad U^\nu := \operatorname{aff}\left\{[u_0^r, u_e^r] \times \cdots \times [u_0^\nu, u_e^\nu]\right\} \cap U_0$$

and $\dim(\operatorname{aff}\left\{[u_0^r, u_e^r] \times \cdots \times [u_0^\nu, u_e^\nu]\right\}) = \nu - r + 1$. *In particular, $\hat v_1^{\nu+1}$ will be the unique maximizing point relative to V for the envelope function*

$$g^\nu(v) := \min_{u \in U^\nu} L(u,v) \geq \min_{u \in U} L(u,v) = g(u),$$

and one will have $g^{\nu+1} \leq g^\nu$ in that primal cycle. Moreover, for $\nu = r + d_1 - 1$ with $d_1 := \dim U_0$, it will be true that $g^\nu = g$ in an ultimate quadratic region for the problem, and also that $\hat v_1^{\nu+1} = \bar v$, as long as the algorithm does not terminate earlier.

Similarly, if $\nu = s \geq \hat\nu$ is the first iteration of some dual cycle with $v_0^s \in V_0$, then for all $\nu \geq s$ in that cycle, on exiting from Step 5 *of Algorithm* 2 *with $\hat v_0^{\nu+1}$ the element $\hat u_1^{\nu+1} \in G(\hat v_0^{\nu+1})$ that will be calculated on return to Step* 1 *will be such that $(\hat u_1^{\nu+1}, \hat v_0^{\nu+1})$ is the unique saddle point of $L(u,v)$ on $U \times V^\nu$, where*

$$(5.2) \qquad V^\nu := \operatorname{aff}\left\{[v_0^s, v_e^s] \times \cdots \times [v_0^\nu, v_e^\nu]\right\} \cap V_0,$$

with $\dim(\operatorname{aff}\left\{[v_0^s, v_e^s] \times \cdots \times [v_0^\nu, v_e^\nu]\right\}) = \nu - s + 1$. *In particular, $\hat u_1^{\nu+1}$ will be the unique minimizing point relative to U for the envelope function*

$$f^\nu(u) := \max_{v \in V^\nu} L(u,v) \leq \max_{v \in V} L(u,v) = f(u),$$

and one will have $f^{\nu+1} \geq f^\nu$ in that dual cycle. Moreover, for $\nu = s + d_2 - 1$ with $d_2 := \dim V_0$, it will be true that $f^\nu = f$ in an ultimate quadratic region for the problem, and also that $\hat u_1^{\nu+1} = \bar u$, as long as the algorithm does not terminate earlier.

Proof. We concentrate on the primal part; the proof of the dual part is parallel. The argument is similar to the one given for Proposition 5.1, but with the segment $[u_0^\nu, u_2^\nu]$ replaced by U^ν. Recall from the proof of Theorem 4.5 that for $\nu \geq r$, the primal procedure is equivalent to the conjugate gradient algorithm on the restriction of f to the affine hull $\operatorname{aff} U_0$ of the critical face U_0. Therefore, the vectors $u_e^r - u_0^r, \cdots, u_e^\nu - u_0^\nu$ are linearly independent, and $\hat u_0^{\nu+1}$ minimizes $f(u)$ over $u \in U^\nu$. The inequality $g^{\nu+1} \leq g^\nu$ follows from the inclusion $U^{\nu+1} \supset U^\nu$. When $\nu = r + d_1 - 1$ we have $\dim\left([u_0^r, u_e^r] \times \cdots \times [u_0^\nu, u_e^\nu]\right) = d_1$, and then $U^\nu = U_0$. From the fact that (3.3) holds in $\tilde V^*$ (cf. the derivation of this relation in the proof of Theorem 3.6) we get $g^\nu = g$ in an ultimate quadratic region. □

This result tells us that on entering an ultimate quadratic region, the primal iterations in Algorithm 2 produce an improving envelope for the dual objective function which approaches that function, whereas the dual iterations produce an improving envelope for the primal objectives which approaches that function. To some extent this explains the phenomenon we have observed in our numerical experiments that restarts often incur fast termination, or at least bring significant progress in the next few iterations.

6. Numerical tests. Numerical tests of Algorithm 1, the Primal-Dual Steepest Descent Algorithm (PDSD), and Algorithm 2, the Primal-Dual Conjugate Gradient Algorithm (PDCG), have been conducted on a DECstation 3100 with double precision on some medium-to large-sized problems. For comparisons we have used the Basic Finite-Envelope Method (BFEM) of [6] and the Stanford LSSOL code of Gill et al. [24] for quadratic programming. To enhance the performance of LSSOL in this situation, we tailored its Cholesky factorization subroutine to take advantage of the special structure of the P and Q matrices in our examples.

Comparisons with LSSOL are based on the fact that any extended-linear-quadratic programming problem can be converted into a standard quadratic programming problem by introducing auxiliary variables and additional constraints [1, Thm. 1]. It must be kept in mind, however, that such a transformation not only increases the dimension substantially but disrupts much of the large-scale structure that might be put to use. A fundamental difficulty with any comparisons with available QP methods, therefore, is that such methods are not really designed to handle the kinds of problems we wish to tackle, which stem from [1], [2], and [6]. They typically require setting up and working with the huge R matrix, and trying to exploit any sparsity patterns that might be present in it, whereas we never need this matrix but work with decomposition in the calculation of the F and G, as explained in §1, after Proposition 1.6.

The integer recorded as the "size" of each problem is the number of primal variables and also the number of dual variables. (The two would not have to be the same.) Thus, size $= 100$ means that problem $(\mathcal{P})$ is an extended linear-quadratic programming problem on $\mathbb{R}^{100}$ for which the dual $(\mathcal{D})$ is likewise such a problem on $\mathbb{R}^{100}$, while the associated Lagrangian saddle point problem concerns a quadratic convex-concave function on a product of polyhedral sets in $\mathbb{R}^{100} \times \mathbb{R}^{100}$. In order to solve such a problem using LSSOL, it must be reformulated as a primal problem in 400 variables with 100 general equality constraints and 200 lower bounds on the auxiliary variables, in addition to having the original polyhedral constraints on the 100 primal variables.

In all the tests of PDCG and PDSD we have taken $\delta = 10^{-2}$ as the progress threshold and $\varepsilon = 10^{-8}$ as the optimality threshold. For PDCG we have taken $k = 5$ as the cycle size parameter (whereas PDSD always has $k = 1$ by definition). We have run BFEM with "mode=1," which means that in each iteration a quadratic saddle point subproblem is solved over a product of two triangles. For the sake of expediency in solving this small subproblem we have set it up as a standard QP problem in the manner of [1, Thm. 1] and have applied LSSOL. No doubt the CPU time could be improved by using a customized procedure within BFEM instead of this heavy-handed approach.

The generation of test problems of large size raises serious questions about the representative nature of such problems. It does not make sense to think of a large problem simply in terms of a large matrix, the elements of which are all random. Rather, a certain amount of structure must be respected. As an attempt to address this issue, we have taken all our problems to have the (deterministic) dynamical structure described in Rockafellar and Wets [5]. Only the parameters natural to this structure have been randomized. The dynamical structure enables us to use special routines in calculating $f(u)$ and $F(u)$, and on the other hand $g(v)$ and $G(v)$ [7]. For this purpose, and in implementing BFEM, we rely on code written by Wright [25] at the University of Washington.

The problems have been obtained as discretized versions of certain continuous-time problems of extended linear-quadratic optimal control of the kind developed in Rockafellar [4]. The first digit of the problem number corresponds to different continuous-time problems and the second digit corresponds to different discretization levels, i.e., the number of subintervals into which the fixed time interval has been partitioned, which determines the size of the discretized problem. Hence, e.g., the problems 0.1, 1.1, ..., 9.1 are the discretization of 10 different continuous-time problems with the same discretization level (*a transverse family of test problems*), and the problems 1.0, 1.1, ..., 1.7 are the discretization of one continuous-time problem with 8 different discretization levels (*a vertical family of test problems*). Only the data values in the continuous-time model have been generated randomly, and in each vertical family these are the same for all the problems. By increasing the number of subintervals, one can get larger and larger problems which remain stable with respect to the numerical scaling.

TABLE 1
Test results of problems 0.1–9.1.

		CPU time (sec.)				Iterations			
Prb.	Size	PDCG	PDSD	BFEM	LSSOL	PDCG	PDSD	BFEM	LSSOL
0.1	100	4.6	4.8	6.6	283.1	23	34	31	500
1.1	100	5.0	5.8	7.5	295.0	28	50	37	497
2.1	100	5.0	4.0	8.1	299.7	28	24	41	495
3.1	100	3.0	2.6	3.4	339.8	5	5	8	562
4.1	100	3.8	3.5	3.8	353.2	13	17	11	619
5.1	100	3.2	2.7	3.5	314.5	8	6	9	544
6.1	100	3.5	3.0	3.8	339.2	11	11	11	552
7.1	100	3.6	3.7	4.3	256.0	13	20	14	445
8.1	100	4.5	5.2	*17.5	290.6	22	42	**	481
9.1	100	3.5	3.3	4.0	347.2	12	15	12	591

TABLE 2
Test results of problems 0.2–9.2.

		CPU time (sec.)				Iterations			
Prb.	Size	PDCG	PDSD	BFEM	LSSOL	PDCG	PDSD	BFEM	LSSOL
0.2	340	9.2	8.9	15.3		24	28	31	
1.2	340	12.5	14.4	19.3		35	50	39	
2.2	340	10.1	11.9	20.5		25	38	42	
3.2	340	5.2	4.3	6.8		9	8	11	
4.2	340	7.8	6.6	8.7		18	17	15	
5.2	340	6.5	5.5	8.0		14	12	12	
6.2	340	5.7	5.1	7.3		11	11	12	
7.2	340	5.4	5.9	7.7		10	15	13	
8.2	340	9.8	11.2	20.3		25	38	42	
9.2	340	6.0	6.4	9.5		12	17	17	

TABLE 3
Test results of problems 0.4–9.4.

		CPU time (sec.)				Iterations			
Prb.	Size	PDCG	PDSD	BFEM	LSSOL	PDCG	PDSD	BFEM	LSSOL
0.4	5140	122.6	138.6	270.8		23	32	38	
1.4	5140	177.9	230.9	315.7		32	52	44	
2.4	5140	218.6	191.7	399.3		40	44	56	
3.4	5140	46.7	45.0	110.1		8	9	16	
4.4	5140	111.8	94.8	126.7		20	20	18	
5.4	5140	71.4	64.5	133.2		12	13	19	
6.4	5140	80.5	78.2	141.2		14	16	20	
7.4	5140	54.9	85.1	104.7		10	19	15	
8.4	5140	161.1	235.9	362.9		29	55	50	
9.4	5140	76.5	77.8	115.5		14	17	16	

The test results in Tables 1, 2, and 3 concern transverse families of size 100, 340, and 5140, respectively. The problems in the first family are small enough for the LSSOL approach to be viable as a comparison. But for the second and third families, our DECstation 3100 falls short of having enough memory for the LSSOL approach. Here we see that PDCG and PDSD are in the leading positions with BFEM not very far behind in terms of CPU times.

The notation ** for the iterations of BFEM on problem 8.1 signifies that the method failed to terminate with optimality in 100 iterations. The corresponding figure for CPU time is preceded by * since it only indicates how long the first 100 iterations took. (The same conventions are adopted in all other tables.)

TABLE 4
Test results of discretized problems 0.0–0.7.

		CPU time (sec.)/Iterations				
Prb.	Size	PDCG	PDSD	BFEM	LSSOL	Value
0.0	40	2.9/11	3.0/15	3.3/13	35.3/327	23.8626
0.1	100	4.3/23	4.8/34	6.6/31	244.4/500	15.7824
0.2	340	9.0/24	9.1/28	15.2/31		15.2107
0.3	1300	27.1/22	32.1/32	58.7/34		15.2145
0.4	5140	122.5/23	137.2/32	269.2/38		15.2179
0.5	20500	568.6/27	593.7/32	1396.2/46		15.2188
0.6	81940	2873.8/27	2722.6/32	*10637.6/**		15.2190
0.7	100020	4209.3/28	3976.5/32	7086.3/38		15.2191

TABLE 5
Test results of discretized problems 1.0–1.7.

		CPU time (sec.)/Iterations				
Prb.	Size	PDCG	PDSD	BFEM	LSSOL	Value
1.0	40	2.9/15	3.0/21	3.9/22	40.9/360	242.05983
1.1	100	4.9/28	5.9/50	7.5/37	294.8/497	249.07378
1.2	340	12.4/35	14.4/50	19.1/39		249.77975
1.3	1300	45.3/37	52.2/52	76.0/44		249.79866
1.4	5140	178.4/32	230.7/52	317.9/44		249.79956
1.5	20500	812.4/36	1007.5/52	1421.5/45		249.79972
1.6	81940	4015.8/36	4699.9/52	6119.9/45		249.79976
1.7	100020	5749.6/36	6538.5/52	8264.0/44		249.79976

TABLE 6
Test results of discretized problems 2.0–2.7.

		CPU time (sec.)/Iterations				
Prb.	Size	PDCG	PDSD	BFEM	LSSOL	Value
2.0	40	3.6/28	4.4/63	*9.7/**	44.7/446	-261.5042
2.1	100	4.7/28	4.1/24	8.2/41	254.8/495	-362.2297
2.2	340	9.5/25	11.1/38	20.0/42		-369.7334
2.3	1300	41.4/33	39.8/39	97.6/56		-369.8073
2.4	5140	220.3/40	191.9/44	402.9/56		-369.8046
2.5	20500	936.9/43	769.4/40	1945.4/63		-369.8036
2.6	81940	4370.3/40	3396.5/39	8893.3/71		-369.8034
2.7	100020	6247.2/40	5123.7/40	10032.1/53		-369.8033

The test results in Tables 4, 5, and 6 refer to vertical families based on the first three continuous-time problems. They cover sizes that are generally too large for the LSSOL approach to be workable. The aim in this case is to examine the effects of increasing size in a context where these effects can be isolated from other aspects of the testing.

In these results the stability of the scaling is reflected by the way the optimal value settles down and converges. Note the fact that, although the CPU time goes up as the problem size becomes larger, the number of iterations remains almost constant once the approximation is close. This suggests that the methods are able to identify the general location of the primal and dual optimal solutions fairly quickly, and that they accomplish this in a manner that is relatively insensitive to the number of variables and constraints. Quite the opposite behavior would be expected, of course, from an active-set QP method. The increase in CPU time seems mainly due to the increase in overhead in setting up the line searches as well as in the evaluations of $f(u)$, $F(u)$, $g(v)$, and $G(v)$ when the dimension is high.

Tables 7, 8, and 9 test the importance of the interactive restarts in PDCG and PDSD. The problems in this case are the same as in Tables 4, 5, and 6 correspondingly. For each problem, the methods were applied in the proposed form, allowing interactive restarts (the *yes* case), but then also in the modified form in which all such restarts are suppressed (the *no* case). The difference that this makes is evident. Interactive restarts have a big effect on the performance, and in the case of PDSD even dictate whether the method is successful or not, in the sense of terminating within 100 iterations. The tables also furnish information on the number of interactive restarts that occurred. For instance, for problem 0.5 under the interactive version of the PDSD method one reads that termination came in 32 iterations, and that in the course of these there were 7 interactive primal restarts and 6 interactive dual restarts. The noninteractive version took 89 iterations.

Another fact to be observed in these large problems is that the simplicity of PDSD sometimes overtakes the carefully designed properties of PDCG in CPU time. An interpretation is that when the dimension is very high, but PDCG is not yet near to the solutions and is just using cycle size $k = 5$ anyway, the conjugate gradient-like features do not always provide a gain that offsets the extra overhead. While the number of iterations in PDCG remains less, the time it takes, in comparison to PDSD, can be more. Perhaps the greatest advantage of these methods comes, therefore, from the information feedback involved in the interactive restarts, rather than from the attention paid to the choice of the descent (or ascent) direction.

TABLE 7

Test results of restart role in problems 0.0–0.7.

	CPU time (sec.)				Iterations			
	PDCG		PDSD		PDCG		PDSD	
Prb.	Yes	No	Yes	No	Yes	No	Yes	No
0.0	2.9	3.0	3.0	3.4	11(3/3)	16	15(3/8)	38
0.1	4.3	4.4	4.8	8.7	23(4/6)	25	34(4/9)	94
0.2	9.0	9.8	9.1	*24.8	24(3/7)	28	28(4/8)	**
0.3	27.1	35.6	32.1	*92.6	22(3/6)	31	32(7/6)	**
0.4	122.5	137.1	137.2	*416.0	23(4/6)	27	32(7/6)	**
0.5	568.6	561.6	593.7	1671.3	27(4/7)	27	32(7/6)	89
0.6	2873.8	2949.1	2722.6	*8326.5	27(4/7)	27	32(7/6)	**
0.7	4209.3	4012.9	3976.5	*12423.5	28(4/7)	27	32(7/6)	**

TABLE 8

Test results of restart role in problems 1.0–1.7.

	CPU time (sec.)				Iterations			
	PDCG		PDSD		PDCG		PDSD	
Prb.	Yes	No	Yes	No	Yes	No	Yes	No
1.0	2.9	3.3	3.0	4.3	15(3/4)	25	21(5/6)	64
1.1	4.9	5.7	5.9	*9.5	28(3/7)	38	50(5/7)	**
1.2	12.4	13.2	14.4	*26.2	35(2/7)	39	50(8/6)	**
1.3	45.3	54.1	52.2	*97.3	37(2/8)	47	52(4/8)	**
1.4	178.4	233.7	230.7	*444.9	32(3/6)	45	52(4/6)	**
1.5	812.4	1011.5	1007.5	*1978.8	36(2/7)	48	52(4/6)	**
1.6	4015.8	5043.6	4699.9	*8795.4	36(2/7)	48	52(4/6)	**
1.7	5749.6	7325.6	6538.5	*12726.2	36(2/7)	48	52(4/6)	**

TABLE 9

Test results of restart role in problems 2.0–2.7.

	CPU time (sec.)				Iterations			
	PDCG		PDSD		PDCG		PDSD	
Prb.	Yes	No	Yes	No	Yes	No	Yes	No
2.0	3.6	4.4	4.4	*5.4	28(5/6)	52	63(7/8)	**
2.1	4.7	6.6	4.1	7.0	28(7/4)	54	24(8/5)	68
2.2	9.5	16.1	11.1	*24.9	25(7/4)	51	38(7/3)	**
2.3	41.4	59.6	39.8	*92.9	33(9/5)	52	39(10/5)	**
2.4	220.3	295.2	191.9	*423.2	40(9/5)	58	44(9/5)	**
2.5	936.9	1362.4	769.4	*1899.4	43(9/6)	65	40(11/5)	**
2.6	4370.3	6385.8	3396.5	*8497.5	40(9/5)	61	39(11/5)	**
2.7	6247.2	9387.8	5123.7	*12359.7	40(9/5)	61	40(11/5)	**

REFERENCES

[1] R. T. ROCKAFELLAR AND R. J.-B. WETS, *A Lagrangian finite generation technique for solving linear-quadratic problems in stochastic programming*, Math. Programming Stud., 28 (1986), pp. 63–93.

[2] ———, *Linear-quadratic problems with stochastic penalties: The finite generation algorithm*, in Numerical Techniques for Stochastic Optimization Problems, Y. Ermoliev and R. J.-B. Wets, eds., Lecture Notes in Control and Information Sciences 81, Springer-Verlag, Berlin, 1987, pp. 545–560.

[3] R. T. ROCKAFELLAR, *A generalized approach to linear-quadratic programming*, in Proc. Internat. Conf. on Numerical Optimization and Appl., Xi'an, China, 1986, pp. 58–66.

[4] ———, *Linear-quadratic programming and optimal control*, SIAM J. Control Optim., 25 (1987), pp. 781–814.

[5] R. T. ROCKAFELLAR AND R. J.-B.WETS, *Generalized linear-quadratic problems of deterministic and stochastic optimal control in discrete time*, SIAM J. Control Optim. 28 (1990), pp. 810–822.

[6] R. T. ROCKAFELLAR, *Computational schemes for solving large-scale problems in extended linear-quadratic programming*, Math. Programming, 48 (1990), pp. 447–474.

[7] ———, *Large-scale extended linear-quadratic programming and multistage optimization*, in Proc. Fifth Mexico-U.S. Workshop on Numerical Analysis, S. Gomez, J.-P. Hennart, R. Tapia, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990.

[8] A. KING, *An implementation of the Lagrangian finite generation method*, in Numerical Techniques for Stochastic Programming Problems, Y. Ermoliev and R. J.-B. Wets, eds., Springer-Verlag, Berlin, 1988.

[9] J. M. WAGNER, *Stochastic Programming with Recourse Applied to Groundwater Quality Management*, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA.

[10] J.-S. PANG, *Methods for quadratic programming: A survey*, Comput. Chem. Engrg., 7 (1983), pp. 583–594.

[11] 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.

[12] Y. YE AND E. TSE, *An extension of Karmarkar's projective algorithm for convex quadratic programming*, Math. Programming, 44 (1989), pp. 157–179.

[13] R. D. C. MONTEIRO AND I. ADLER, *Interior path following primal-dual algorithms. Part* II: *Convex quadratic programming*, Math. Programming, 44 (1989), pp. 43–66.

[14] D. GOLDFARB AND S. LIU, *An $O(n^3L)$ primal interior point algorithm for convex quadratic programming*, preprint.

[15] P. TSENG, *Applications of a splitting algorithm to decomposition in convex programming and variational inequalities*, SIAM J. Control Optim., 29 (1991), pp. 119–138.

[16] ———, *Further applications of a splitting algorithm to decomposition in variational inequalities and convex programming*, Math. Programming Stud., 48 (1990), pp. 249–263.

[17] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton Univ. Press, Princeton, NJ, 1970.

[18] ———, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898.

[19] ———, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.

[20] C. ZHU, *Modified proximal point algorithm for extended linear-quadratic programming*, Comput. Optim. Appl., to appear.

[21] M. S. BAZARAA AND C. M. SHETTY, *Nonlinear Programming: Theory and Algorithms*, Wiley, New York, 1979.

[22] M. AVRIEL, *Nonlinear Programming: Analysis and Methods*, Prentice Hall, Englewood Cliffs, NJ, 1976

[23] J. V. BURKE AND J. J. MORÉ, *On the identification of active constraints*, SIAM J. Numer. Anal., 25 (1988), pp. 1197–1211.

[24] P. E. GILL, S. J. HAMMARLING, W. MURRAY, M. A. SAUNDERS, AND M. H. WRIGHT, *User's guide for LSSOL (Version* 1.0): *A FORTRAN package for constrained linear least-squares and convex quadratic programming*, Tech. Report SOL 86–1, Dept. of Operations Research, Stanford Univ., Stanford, CA, 1986.

[25] S. E. WRIGHT (with introduction by R. T. ROCKAFELLAR), *DYNFGM: Dynamic Finite Generation Method*, Report, Dept. of Mathematics, Univ. of Washington, Seattle, WA, 1989.

SIAM J. OPTIMIZATION
Vol. 3, No. 4, pp. 784–799, November 1993

007

THE D_2^*-TRIANGULATION FOR CONTINUOUS DEFORMATION ALGORITHMS TO COMPUTE SOLUTIONS OF NONLINEAR EQUATIONS*

CHUANGYIN DANG†

Abstract. A new triangulation of continuous refinement of grid size of $(0,1] \times R^n$ for use in a continuous deformation algorithm to compute solutions of nonlinear equations is proposed. It is called the D_2^*-triangulation. One can choose any positive even integer as a factor of refinement of grid size of this triangulation. The author proves that the D_2^*-triangulation is superior to the K_2^*-triangulation and J_2^*-triangulation in the number of simplices. Numerical tests show that the continuous deformation algorithm based on the D_2^*-triangulation is indeed much more efficient.

Key words. continuous deformation algorithms, triangulations, measures of efficiency of triangulations, numerical solutions of nonlinear equations

AMS subject classification. 90C30

1. Introduction.

Simplicial methods, also known as fixed point methods, were originated by Scarf in his seminal paper [20] to compute fixed points of a continuous mapping from the unit simplex to itself. Simplicial methods have been developing for over twenty years. As a tool for solving highly nonlinear problems, which are derived from decision making, economic modelling, and engineering, simplicial methods are very powerful.

The so-called continuous deformation algorithm is one of the most successful simplicial methods. It was initiated by Eaves in [9] to compute fixed points on the unit simplex and generalized to R^n by Eaves and Saigal in [11] to find solutions of nonlinear equations. This method is also called the simplicial homotopy algorithm.

The principles of the continuous deformation algorithm are as follows. Let $f : R^n \rightarrow R^n$ be a nonlinear mapping, $f = (f_1, f_2, \ldots, f_n)^\top$. We want to compute a zero point of f. Let $g : R^n \rightarrow R^n$ be an affinely linear mapping with a unique zero point x^0, i.e., $g(x) = A(x - x^0)$, where A is an $n \times n$ nonsingular matrix. Then the homotopy function h is given by

$$h(t,x) = \begin{cases} 2(1-t)f(x) + (2t-1)g(x), & \frac{1}{2} \le t \le 1, \\ f(x), & 0 \le t < \frac{1}{2} \end{cases}$$

for $(t,x) \in [0,1] \times R^n$. The underlying space $(0,1] \times R^n$ is subdivided into simplices by a triangulation, denoted by T, with continuous refinement of grid size. The piecewise linear approximation H of h with respect to T is given by, for $(t,x) = \sum_{i=-1}^{n} \lambda_i y^i \in \sigma$, a simplex in T, with $\lambda_i \ge 0$, for $i = -1, 0, \ldots, n$, and $\sum_{i=-1}^{0} \lambda_i = 1$,

$$H(t,x) = \sum_{i=-1}^{n} \lambda_i h(y^i),$$

where y^i is a vertex of σ for $i = -1, 0, \ldots, n$. Then there exist some piecewise linear paths defined by the set of zero points of H. In particular, one of the paths starts at

* Received by the editors October 16, 1991; accepted for publication (in revised form) August 24, 1992.

† Department of Mathematics, University of California, Davis, California 95616-8633. Present address, Department of Engineering Science, University of Aukland, Private Bag 92019, Auckland, New Zealand.

x^0 and either goes to infinity or converges to a zero point of f. One can trace this path with the standard lexicographical pivoting rule.

It was recognized very early (see [17]) that efficiency of a simplicial algorithm depends on the underlying triangulation. In order to improve efficiency of continuous deformation algorithms, a number of triangulations with continuous refinement of grid size have been proposed. These include the K_3-triangulation and J_3-triangulation of Todd [22]; the D_3-triangulation and D_2-triangulation of Dang [4], [5]; the arbitrary grid size refinement triangulation of van der Laan and Talman [15] and Shamir [21]; the K_2-triangulation, J_2-triangulation, K_2^*-triangulation, and J_2^*-triangulation of Kojima and Yamamoto [14]; the triangulation of Broadie and Eaves [2]; and the triangulation of Doup and Talman [8]. All these triangulations were derived from the well-known K_1-triangulation or J_1-triangulation, except the D_3-triangulation and D_2-triangulation, which were obtained from the D_1-triangulation. The latter triangulation of R^n was proposed in [3] and is superior to the K_1-triangulation and J_1-triangulation according to all the measures of efficiency.

The development of triangulations of arbitrary continuous refinement of grid size is stimulated by the implementation of an acceleration technique originated by Saigal in [18]. Theoretical results and numerical tests have proved that the D_3-triangulation is superior to the K_3-triangulation and J_3-triangulation and that the D_2-triangulation is superior to the K_2-triangulation and J_2-triangulation. As mentioned by Kojima and Yamamoto in [14], the K_3-triangulation is a special case of the K_2^*-triangulation with all the factors of refinement equal to two, and the J_3-triangulation is a special case of the J_2^*-triangulation with all the factors of refinement equal to two. Numerical tests have shown [4] that the continuous deformation algorithm based on the D_3-triangulation is very efficient. However, all of its factors of refinement are also equal to two. Motivated by the results in [14] and using the D_1-triangulation, we construct a new triangulation of continuous refinement of grid size of $(0,1] \times R^n$ for use in a continuous deformation algorithm. It is called the D_2^*-triangulation. One can choose any positive even integer as a factor of refinement of this triangulation. This feature is the same as that of the K_2^*-triangulation or J_2^*-triangulation. Similarly to the K_3-triangulation and J_3-triangulation, the D_3-triangulation now becomes a special case of the D_2^*-triangulation with all the factors of refinement equal to two.

For comparison with the D_2^*-triangulation, we also present the K_2^*-triangulation and J_2^*-triangulation, which were given by Kojima and Yamamoto in [14] without an algebraic definition. We prove that the D_2^*-triangulation is superior to the K_2^*-triangulation and J_2^*-triangulation in the number of simplices. Since it is rather complicated to calculate the surface density of these triangulations, we will not reproduce it here. We refer the reader instead to [3] and [12].

Numerical tests show that the continuous deformation algorithm based on the D_2^*-triangulation is indeed more efficient. We remark that the structure of the D_2^*-triangulation is quite different from that of the D_2-triangulation. Numerical tests show that the D_2^*-triangulation is generally faster than the D_2-triangulation. However, the number of simplices of the D_2-triangulation is less than that of the D_2^*-triangulation when their mesh sizes are equal and one can choose any positive number as a factor of refinement of grid size of the D_2-triangulation. Note that there exists a number of other interesting triangulations of R^n; see [19], [16], [24], and [13]. However, it is not known how these triangulations of R^n can be used to obtain triangulations of $(0,1] \times R^n$ with continuous refinement of grid size.

In §2, an algebraic definition of the D_2^*-triangulation is presented. In §3, we prove that the definition given in §2 yields a triangulation. The pivot rules of the D_2^*-triangulation for how to generate one of its adjacent simplices from a simplex when moving along the path are described in §4. Comparison with some other triangulations is presented in §5.

2. Algebraic definition of the D_2^*-triangulation. Let N_0 denote the index set $\{0, 1, \ldots, n\}$ and let u^i be the ith unit vector in R^{n+1} for $i = 0, 1, \ldots, n$. Let α_0 be a positive number and $\beta_i \in \{1/j \mid j = 1, 2, \ldots\}$ for $i = 0, 1, \ldots$. We choose α_{j+1} such that $\alpha_{j+1} = \alpha_j \beta_j / 2$ for $j = 0, 1, \ldots$. Set $\beta_{-1} = 1$.

Let $\pi = (\pi(0), \pi(1), \ldots, \pi(n))$ be a permutation of the elements of N_0. Let q denote the integer with $\pi(q) = 0$. We take a vector $y \in (0, 1] \times R^n$ such that, for an integer $k \geq 0$, $y_0 = 2^{-(k+1)}$, $y_{\pi(i)}/2\alpha_{k+1}$ is an integer for $i = 0, \ldots, q-1$, and $y_{\pi(i)}/\alpha_k$ is odd for $i = q+1, \ldots, n$. We define

$$w_{\pi(i)} = \begin{cases} \lfloor y_{\pi(i)}/\alpha_k \rfloor + 1 & \text{if } \lfloor y_{\pi(i)}/\alpha_k \rfloor \text{ is odd,} \\ \lfloor y_{\pi(i)}/\alpha_k \rfloor & \text{otherwise,} \end{cases}$$

for $i = 0, 1, \ldots, q-1$.

DEFINITION 2.1. Let y and π be as above. Then vectors y^{-1}, y^0, $\ldots$, y^n are given as follows.

$$y^{-1} = y,$$

$$y^i = y^{i-1} + 2\alpha_{k+1} u^{\pi(i)}, \qquad i = 0, 1, \ldots, q-1,$$

$$y^q = \alpha_k \textstyle\sum_{j=0}^{q-1} w_{\pi(j)} u^{\pi(j)} + \sum_{j=q+1}^{n} (y_{\pi(j)} - \alpha_k) u^{\pi(j)} + 2y_0 u^0,$$

$$y^i = y^{i-1} + 2\alpha_k u^{\pi(i)}, \qquad i = q+1, \ldots, n.$$

Let y^{-1}, y^0, $\ldots$, y^n be obtained in the above manner. Then it can be seen that they are affinely independent. Thus their convex hull is a simplex. Let us denote this simplex by $K_2^*(y, \pi)$. Let K_2^* denote the collection of all such simplices $K_2^*(y, \pi)$. It will be shown in the next section that K_2^* is a triangulation of $(0, 1] \times R^n$ such that one can choose any positive even integer as its factor of refinement of grid size and when all of its factors of refinement of grid size are equal to two, it becomes the same as the K_3-triangulation. We call it the K_2^*-triangulation.

Let $\pi = (\pi(0), \pi(1), \ldots, \pi(n))$ be a permutation of the elements of N_0. Let q denote the integer with $\pi(q) = 0$. We take a vector $y \in (0, 1] \times R^n$ such that, for an integer $k \geq 0$, $y_0 = 2^{-(k+1)}$, $y_{\pi(i)}/2\alpha_{k+1}$ is even for $i = 0, \ldots, q-1$ if $1/\beta_k$ is even, $y_{\pi(i)}/2\alpha_{k+1}$ is odd for $i = 0, \ldots, q-1$ if $1/\beta_k$ is odd, and $y_{\pi(i)}/\alpha_k$ is odd for $i = q+1, \ldots, n$. If $1/\beta_{k-1}$ is odd, let us define

$$t_{\pi(i)} = \begin{cases} -1 & \text{if } y_{\pi(i)}/\alpha_k = 1(\text{mod} 4), \\ 1 & \text{if } y_{\pi(i)}/\alpha_k = 3(\text{mod} 4), \end{cases}$$

for $i = q+1, \ldots, n$, and if $1/\beta_{k-1}$ is even, let us define

$$t_{\pi(i)} = \begin{cases} 1 & \text{if } y_{\pi(i)}/\alpha_k = 1(\text{mod} 4), \\ -1 & \text{if } y_{\pi(i)}/\alpha_k = 3(\text{mod} 4), \end{cases}$$

for $i = q+1, \ldots, n$. We take a sign vector $s = (s_1, s_2, \ldots, s_n)^\top$ such that $s_i \in \{-1, +1\}$ for $i = 1, 2, \ldots, n$ and $s_{\pi(i)} = t_{\pi(i)}$ for $i = q+1, \ldots, n$. We define

$$w_{\pi(i)} = \begin{cases} \lfloor y_{\pi(i)}/\alpha_k \rfloor + 1 & \text{if } \lfloor y_{\pi(i)}/\alpha_k \rfloor \text{ is odd and either } y_{\pi(i)}/\alpha_k \neq \lfloor y_{\pi(i)}/\alpha_k \rfloor \\ & \quad \text{or both } \lfloor y_{\pi(i)}/\alpha_k \rfloor = y_{\pi(i)}/\alpha_k \text{ and } s_{\pi(i)} = 1, \\ \lfloor y_{\pi(i)}/\alpha_k \rfloor & \text{if } \lfloor y_{\pi(i)}/\alpha_k \rfloor \text{ is even,} \\ \lfloor y_{\pi(i)}/\alpha_k \rfloor - 1 & \text{otherwise,} \end{cases}$$

for $i = 0, 1, \ldots, q-1$.

DEFINITION 2.2. Let y, π, and s be as above. Then vectors y^{-1}, y^0, $\ldots$, y^n are given as follows.

$$\begin{aligned} y^{-1} &= y, \\ y^i &= y^{i-1} + 2\alpha_{k+1} s_{\pi(i)} u^{\pi(i)}, \qquad i = 0, 1, \ldots, q-1, \\ y^q &= \alpha_k \textstyle\sum_{j=0}^{q-1} w_{\pi(j)} u^{\pi(j)} + \sum_{j=q+1}^{n} (y_{\pi(j)} - \alpha_k s_{\pi(j)}) u^{\pi(j)} + 2y_0 u^0, \\ y^i &= y^{i-1} + 2\alpha_k s_{\pi(i)} u^{\pi(i)}, \qquad i = q+1, \ldots, n. \end{aligned}$$

Let y^{-1}, y^0, $\ldots$, y^n be obtained in the above manner. Then it can be seen that they are affinely independent. Thus their convex hull is a simplex. Let us denote this simplex by $J_2^*(y, \pi, s)$. Let J_2^* denote the set of all such simplices $J_2^*(y, \pi, s)$. It will be shown in the next section that J_2^* is a triangulation of $(0, 1] \times R^n$ such that one can choose any positive even integer as its factor of refinement of grid size and when all of its factors of refinement of grid size are equal to two, it becomes the same as the J_3-triangulation. We call it the J_2^*-triangulation.

Let $\pi = (\pi(0), \pi(1), \ldots, \pi(n))$ be a permutation of the elements of N_0. Let q denote the integer with $\pi(q) = 0$. We take a vector $y \in (0, 1] \times R^n$ such that, for an integer $k \geq 0$, $y_0 = 2^{-(k+1)}$, $y_{\pi(i)}/2\alpha_{k+1}$ is even for $i = 0, \ldots, q-1$ if $1/\beta_k$ is even, $y_{\pi(i)}/2\alpha_{k+1}$ is odd for $i = 0, \ldots, q-1$ if $1/\beta_k$ is odd, and $y_{\pi(i)}/\alpha_k$ is odd for $i = q+1, \ldots, n$. If $1/\beta_{k-1}$ is odd, let us define

$$t_{\pi(i)} = \begin{cases} -1 & \text{if } y_{\pi(i)}/\alpha_k = 1(\text{mod}4), \\ 1 & \text{if } y_{\pi(i)}/\alpha_k = 3(\text{mod}4), \end{cases}$$

for $i = q+1, \ldots, n$, and if $1/\beta_{k-1}$ is even, let us define

$$t_{\pi(i)} = \begin{cases} 1 & \text{if } y_{\pi(i)}/\alpha_k = 1(\text{mod}4), \\ -1 & \text{if } y_{\pi(i)}/\alpha_k = 3(\text{mod}4), \end{cases}$$

for $i = q+1, \ldots, n$. We take a sign vector $s = (s_1, s_2, \ldots, s_n)^\top$ such that $s_i \in \{-1, +1\}$

for $i = 1, 2, \dots, n$ and $s_{\pi(i)} = t_{\pi(i)}$ for $i = q+1, \dots, n$. We define

$$w_{\pi(i)} = \begin{cases} \lfloor y_{\pi(i)}/\alpha_k \rfloor + 1 & \text{if } \lfloor y_{\pi(i)}/\alpha_k \rfloor \text{ is odd and either } y_{\pi(i)}/\alpha_k \neq \lfloor y_{\pi(i)}/\alpha_k \rfloor \\ & \quad \text{or both } \lfloor y_{\pi(i)}/\alpha_k \rfloor = y_{\pi(i)}/\alpha_k \text{ and } s_{\pi(i)} = 1, \\ \lfloor y_{\pi(i)}/\alpha_k \rfloor & \text{if } \lfloor y_{\pi(i)}/\alpha_k \rfloor \text{ is even}, \\ \lfloor y_{\pi(i)}/\alpha_k \rfloor - 1 & \text{otherwise}, \end{cases}$$

for $i = 0, 1, \dots, q-1$. Set

$$I = \begin{cases} \{\pi(i) \mid w_{\pi(i)}/2 \text{ is even and } 0 \le i \le q-1\} & \text{if } 1/\beta_{k-1} \text{ is odd}, \\ \{\pi(i) \mid w_{\pi(i)}/2 \text{ is odd and } 0 \le i \le q-1\} & \text{if } 1/\beta_{k-1} \text{ is even}. \end{cases}$$

Let h denote the number of elements in I. We take two integers p_1 and p_2 such that $-1 \le p_1 \le q-2$ if $q \ge 1$; $p_1 = -1$ if $q = 0$; when $h = 0$, $0 \le p_2 \le n-q-1$ if $q < n$, and $p_2 = 0$ if $q = n$; when $h > 0$, $p_2 = n - q$.

DEFINITION 2.3. Let y, π, s, p_1, and p_2 be as above. Then vectors y^{-1}, y^0, $\dots$, y^n are given as follows. When $p_1 = -1$,

$$y^{-1} = y,$$

$$y^i = y + 2\alpha_{k+1} s_{\pi(i)} u^{\pi(i)}, \qquad i = 0, 1, \dots, q-1,$$

and when $p_1 \ge 0$,

$$y^{-1} = y + 2\alpha_{k+1} \textstyle\sum_{j=0}^{q-1} s_{\pi(j)} u^{\pi(j)},$$

$$y^i = y^{i-1} - 2\alpha_{k+1} s_{\pi(i)} u^{\pi(i)}, \qquad i = 0, 1, \dots, p_1 - 1,$$

$$y^i = y + 2\alpha_{k+1} s_{\pi(i)} u^{\pi(i)}, \qquad i = p_1, \dots, q-1.$$

When $h > 0$,

$$y^q = \alpha_k \textstyle\sum_{j=0}^{q-1} w_{\pi(j)} u^{\pi(j)} + \sum_{j=q+1}^{n} (y_{\pi(j)} + \alpha_k s_{\pi(j)}) u^{\pi(j)} + 2y_0 u^0,$$

$$y^i = y^{i-1} - 2\alpha_k s_{\pi(i)} u^{\pi(i)}, \qquad i = q+1, \dots, n,$$

and when $h = 0$, if $p_2 = 0$, then

$$y^q = \alpha_k \textstyle\sum_{j=0}^{q-1} w_{\pi(j)} u^{\pi(j)} + \sum_{j=q+1}^{n} (y_{\pi(j)} - \alpha_k s_{\pi(j)}) u^{\pi(j)} + 2y_0 u^0,$$

$$y^i = y^q + 2\alpha_k s_{\pi(i)} u^{\pi(i)}, \qquad i = q+1, \dots, n,$$

and if $p_2 \ge 1$, then

$$y^q = \alpha_k \textstyle\sum_{j=0}^{q-1} w_{\pi(j)} u^{\pi(j)} + \sum_{j=q+1}^{n} (y_{\pi(j)} + \alpha_k s_{\pi(j)}) u^{\pi(j)} + 2y_0 u^0,$$

$$y^i = y^{i-1} - 2\alpha_k s_{\pi(i)} u^{\pi(i)}, \qquad i = q+1, \dots, q+p_2-1,$$

$$y^i = y^* + 2\alpha_k s_{\pi(i)} u^{\pi(i)}, \qquad i = q+p_2, \dots, n,$$

where

$$y^* = \alpha_k \sum_{j=0}^{q-1} w_{\pi(j)} u^{\pi(j)} + \sum_{j=q+1}^{n} (y_{\pi(j)} - \alpha_k s_{\pi(j)}) u^{\pi(j)} + 2y_0 u^0.$$

Let y^{-1}, y^0, ..., y^n be obtained in the above manner. Then it can be seen that they are affinely independent. Thus their convex hull is a simplex. Let us denote this simplex by $D_2^*(y, \pi, s, p_1, p_2)$. Let D_2^* denote the set of all such simplices $D_2^*(y, \pi, s, p_1, p_2)$. It will be shown in the next section that D_2^* is a triangulation of $(0,1] \times R^n$ such that one can choose any positive even integer as its factor of refinement of grid size and when all of its factors of refinement of grid size are equal to two, it becomes the same as the D_3-triangulation. We call it the D_2^*-triangulation.

3. Construction of the D_2^*-triangulation. As follows, we prove that the sets K_2^*, J_2^*, and D_2^* defined in the second section yield a triangulation of $(0,1] \times R^n$, respectively. Let N denote the index set $\{1, 2, \dots, n\}$ and let Q denote the set

$$\{w \mid \text{all components of } w \text{ are integers}\}.$$

We take an arbitrary element $w \in Q$. We define

$$I_o(w) = \{i \in N \mid w_i \text{ is odd}\} \quad \text{and} \quad I_e(w) = \{j \in N \mid w_j \text{ is even}\}.$$

Furthermore, let $A(w)$ denote the set

$$\{x \in R^n \mid w_i - 1 \le x_i \le w_i + 1 \text{ for } i \in I_o(w),\, x_i = w_i \text{ for } i \in I_e(w)\}$$

and let $B(w)$ denote the set

$$\{x \in R^n \mid x_i = w_i \text{ for } i \in I_o(w),\, w_i - 1 \le x_i \le w_i + 1 \text{ for } i \in I_e(w)\}.$$

Let k be a nonnegative integer. Let $D^k(w)$ denote the convex hull of the set

$$(\{2^{-k}\} \times A(w)) \cup (\{2^{-(k+1)}\} \times B(w)).$$

The following lemmas can be found in [4] and [14].

LEMMA 3.1.

$$D^k(w) = \left\{ d \in [2^{-(k+1)}, 2^{-k}] \times R^n \;\middle|\; \begin{array}{l} \mid d_i - w_i \mid \le 2^{k+1} d_0 - 1 \; \textit{for } i \in I_o(w) \\ \\ \mid d_i - w_i \mid \le 2 - 2^{k+1} d_0 \; \textit{for } i \in I_e(w) \end{array} \right\}.$$

LEMMA 3.2. $\cup_{w \in Q} D^k(w) = [2^{-(k+1)}, 2^{-k}] \times R^n$.

LEMMA 3.3. *For w^1, $w^2 \in Q$, $D^k(w^1) \cap D^k(w^2)$ is either empty or a common face of both $D^k(w^1)$ and $D^k(w^2)$, and when $D^k(w^1) \cap D^k(w^2)$ is not empty, it is equal to the convex hull of the set*

$$(\{2^{-k}\} \times (A(w^1) \cap A(w^2))) \cup (\{2^{-(k+1)}\} \times (B(w^1) \cap B(w^2))).$$

For convenience of the following discussion, we give the definitions of the D_1-triangulation, K_1-triangulation, and J_1-triangulation. For more details, see [3] and [22]. Let e^i be the ith unit vector of R^n for $i = 1, 2, \dots, n$.

Let D denote either the set

$$\{x \in R^n \mid \text{all components of } x \text{ are odd}\}$$

or the set

$$\{x \in R^n \mid \text{all components of } x \text{ are even}\}.$$

Let $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$ be a permutation of the elements of N. We take a vector y from the set D and a sign vector $s = (s_1, s_2, \ldots, s_n)^\top$ such that $s_i \in \{-1, 1\}$ for $i = 1, 2, \ldots, n$. Let p be an integer with $0 \leq p \leq n - 1$.

DEFINITION 3.1. Let y, π, s, and p be as above. Then vectors y^0, y^1, $\ldots$, y^n are given as follows. If $p = 0$, then $y^0 = y$ and $y^j = y + s_{\pi(j)} e^{\pi(j)}$, $j = 1, 2, \ldots, n$. If $p \geq 1$, then

$$\begin{aligned} y^0 &= y + s, \\ y^j &= y^{j-1} - s_{\pi(j)} e^{\pi(j)}, \qquad j = 1, 2, \ldots, p-1, \\ y^j &= y + s_{\pi(j)} e^{\pi(j)}, \qquad j = p, p+1, \ldots, n. \end{aligned}$$

Let D_1 denote the collection of all simplices $D_1(y, \pi, s, p)$ that are the convex hull of y^0, y^1, $\ldots$, y^n, as obtained from the above definition. Then D_1 is a triangulation of R^n, called the D_1-triangulation.

Let K denote the set

$$\{x \in R^n \mid \text{all components of } x \text{ are integers}\}.$$

Let $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$ be a permutation of the elements of N. We take a vector y from the set K.

DEFINITION 3.2. Let y and π be as above. Then vectors y^0, y^1, $\ldots$, y^n are given as follows

$$y^0 = y \quad \text{and} \quad y^j = y^{j-1} + e^{\pi(j)}, \quad j = 1, 2, \ldots, n.$$

Let K_1 denote the collection of all simplices $K_1(y, \pi)$ that are the convex hull of y^0, y^1, $\ldots$, y^n, as obtained from the above definition. Then K_1 is a triangulation of R^n, called the K_1-triangulation.

Let J denote either the set

$$\{x \in R^n \mid \text{all components of } x \text{ are odd}\}$$

or the set

$$\{x \in R^n \mid \text{all components of } x \text{ are even}\}.$$

Let $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$ be a permutation of the elements of N. We take a vector y from the set J and a sign vector $s = (s_1, s_2, \ldots, s_n)^\top$ such that $s_i \in \{-1, 1\}$ for $i = 1, 2, \ldots, n$.

DEFINITION 3.3. Let y, π, and s be as above. Then vectors y^0, y^1, $\ldots$, y^n are given as follows.

$$y^0 = y \quad \text{and} \quad y^j = y^{j-1} + s_{\pi(j)} e^{\pi(j)}, \quad j = 1, 2, \ldots, n.$$

Let J_1 denote the collection of all simplices $J_1(y, \pi, s)$ that are the convex hull of y^0, y^1, $\ldots$, y^n, as obtained from the above definition. Then J_1 is a triangulation of R^n, called the J_1-triangulation.

Let G be one of these triangulations of R^n. Let $\bar{G}$ denote the set of faces of simplices in G. Then let α_0 be a positive number and $\beta_i \in \{1/j \mid j = 1, 2, \ldots\}$ for $i = 0, 1, \ldots$. We choose α_{j+1} such that $\alpha_{j+1} = \alpha_j \beta_j / 2$ for $j = 0, 1, \ldots$. Set $\beta_{-1} = 1$. Note that α_i and β_i are the same as before in the second section.

Let $2\alpha_k \bar{G} \mid \alpha_k A(w)$ be the set given by

$$\{\sigma \subseteq \alpha_k A(w) \mid \sigma \in 2\alpha_k \bar{G} \text{ and } \dim(\sigma) = \dim(A(w))\}$$

and let $2\alpha_{k+1}\bar{G} \mid \alpha_k B(w)$ be the set given by

$$\{\sigma \subseteq \alpha_k B(w) \mid \sigma \in 2\alpha_{k+1} \bar{G} \text{ and } \dim(\sigma) = \dim(B(w))\}.$$

For the D_1-triangulation, the K_1-triangulation, and the J_1-triangulation, it can be seen that $2\alpha_k \bar{G} \mid \alpha_k A(w)$ is a triangulation of $\alpha_k A(w)$, and $2\alpha_{k+1}\bar{G} \mid \alpha_k B(w)$ is a triangulation of $\alpha_k B(w)$.

Let a denote the number of elements in the set $I_o(w)$ and let b denote the number of elements in the set $I_e(w)$. Let $\sigma_A \in 2\alpha_k \bar{G} \mid \alpha_k A(w)$ be equal to the convex hull of y_A^0, y_A^1, $\ldots$, y_A^a and let $\sigma_B \in 2\alpha_{k+1}\bar{G} \mid \alpha_k B(w)$ be equal to the convex hull of y_B^0, y_B^1, $\ldots$, y_B^b. Furthermore, let σ denote the convex hull of the set $(\{2^{-k}\} \times \sigma_A) \cup (\{2^{-(k+1)}\} \times \sigma_B)$. It can be easily proved that σ is a simplex in $[2^{-(k+1)}, 2^{-k}] \times R^n$ and σ is equal to the convex hull of $(2^{-k}, y_A^0)^\top$, $(2^{-k}, y_A^1)^\top$, $\ldots$, $(2^{-k}, y_A^a)^\top$, $(2^{-(k+1)}, y_B^0)^\top$, $(2^{-(k+1)}, y_B^1)^\top$, $\ldots$, $(2^{-(k+1)}, y_B^b)^\top$.

Let $T(k, k+1)$ denote the collection of all such simplices σ. Then, following the conclusions mentioned above, we have that, for σ^1 and σ^2 in $T(k, k+1)$, the intersection $\sigma^1 \cap \sigma^2$ is either empty or a common face of both σ^1 and σ^2, and that the union of all $\sigma \in T(k, k+1)$ is equal to $[2^{-(k+1)}, 2^{-k}] \times R^n$. Hence, $T(k, k+1)$ is a triangulation of $[2^{-(k+1)}, 2^{-k}] \times R^n$.

THEOREM 3.4. *The union of* $T(k, k+1)$ *over all nonnegative integers* k *is a triangulation of* $(0, 1] \times R^n$.

Proof. From the choice of α_j and β_j for $j = 0, 1, \ldots$, the theorem follows immediately. □

We call the triangulation obtained in the above manner the G_2^*-triangulation. In this way, we obtain the K_2^*-triangulation, the J_2^*-triangulation, and the D_2^*-triangulation, as described in §2. Considering consistency, one can easily prove these results.

4. Pivot rules of the D_2^*-triangulation. As described in the first section, when a piecewise linear path of zero points is traced, the problem one faces is how to generate one of its adjacent simplices from a simplex when moving along the path. As follows, the pivot rules of the K_2^*-triangulation, J_2^*-triangulation, and D_2^*-triangulation are described. The continuous deformation algorithm based on one of these triangulations can be implemented according to these pivot rules. In the following pivot rules, $y_0 = 2^{-(k+1)}$, $y = (y_1, \ldots, y_n)^\top$, $\bar{y}_0 = 2^{-(\bar{k}+1)}$, $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_n)^\top$, and $u = (1, 1, \ldots, 1)^\top$.

Let a simplex of the K_2^*-triangulation, $\sigma = K_2^*(y, \pi)$, be given with vertices y^{-1}, y^0, $\ldots$, y^n. We want to obtain the simplex of the K_2^*-triangulation, $\bar{\sigma} = K_2^*(\bar{y}, \bar{\pi})$, such that all vertices of σ are also vertices of $\bar{\sigma}$ except the vertex y^i. As follows, we show how $\bar{y}$ and $\bar{\pi}$ depend on y, π, and i.

$\underline{i=-1}$: In case $q=0$, $\bar{y}=y-\alpha_k u$, $\bar{\pi}=(\pi(1),\dots,\pi(n),\pi(0))$, $\bar{q}=n$, and $\bar{k}=k-1$. In case $q\geq 1$, if $y^0_{\pi(0)}=\alpha_k(w_{\pi(0)}+1)$, then $\bar{y}=y-(y_{\pi(0)}-\alpha_k(w_{\pi(0)}+1))u^{\pi(0)}$, $\bar{\pi}=(\pi(1),\dots,\pi(n),\pi(0))$, $\bar{q}=q-1$, and $\bar{k}=k$; if $y^0_{\pi(0)}\neq\alpha_k(w_{\pi(0)}+1)$, then $\bar{y}=y+2\alpha_{k+1}u^{\pi(0)}$, $\bar{\pi}=(\pi(1),\dots,\pi(q-1),\pi(0),\pi(q),\dots,\pi(n))$, $\bar{q}=q$, and $\bar{k}=k$.

$\underline{0\leq i<q-1}$: $\bar{y}=y$, $\bar{\pi}=(\pi(0),\dots,\pi(i+1),\pi(i),\dots,\pi(n))$, $\bar{q}=q$, and $\bar{k}=k$.

$\underline{0\leq i=q-1}$: If $y_{\pi(q-1)}=\alpha_k(w_{\pi(q-1)}-1)$, then $\bar{y}=y$, $\bar{\pi}=(\pi(0),\dots,\pi(q),\pi(q-1),\dots,\pi(n))$, $\bar{q}=q-1$, and $\bar{k}=k$. If $y_{\pi(q-1)}\neq\alpha_k(w_{\pi(q-1)}-1)$, then $\bar{y}=y-2\alpha_{k+1}u^{\pi(q-1)}$, $\bar{\pi}=(\pi(q-1),\pi(0),\dots,\pi(q-2),\pi(q),\dots,\pi(n))$, $\bar{q}=q$, and $\bar{k}=k$.

$\underline{q=i<n}$: $\bar{y}=y$, $\bar{\pi}=(\pi(0),\dots,\pi(q+1),\pi(q),\dots,\pi(n))$, $\bar{q}=q+1$, and $\bar{k}=k$.

$\underline{q<i<n}$: $\bar{y}=y$, $\bar{\pi}=(\pi(0),\dots,\pi(i+1),\pi(i),\dots,\pi(n))$, $\bar{q}=q$, and $\bar{k}=k$.

$\underline{i=n}$: In case $q<n$, $\bar{y}=y-2\alpha_{k+1}u^{\pi(n)}$, $\bar{\pi}=(\pi(n),\pi(0),\dots,\pi(n-1))$, $\bar{q}=q+1$, and $\bar{k}=k$. In case $q=n$, $\bar{y}=y+\alpha_{k+1}u$, $\bar{\pi}=(\pi(n),\pi(0),\dots,\pi(n-1))$, $\bar{q}=0$, and $\bar{k}=k+1$.

Next, let a simplex of the J_2^*-triangulation, $\sigma=J_2^*(y,\pi,s)$, be given with vertices y^{-1}, y^0, $\dots$, y^n. We want to obtain the simplex of the J_2^*-triangulation, $\bar{\sigma}=J_2^*(\bar{y},\bar{\pi},\bar{s})$, such that all vertices of σ are also vertices of $\bar{\sigma}$ except the vertex y^i. As follows, we show how $\bar{y}$, $\bar{\pi}$, and $\bar{s}$ depend on y, π, s, and i.

$\underline{i=-1}$: In case $q=0$, $\bar{y}=y-\alpha_k s$, $\bar{s}=s$, $\bar{\pi}=(\pi(1),\pi(2),\dots,\pi(n),\pi(0))$, $\bar{q}=n$, and $\bar{k}=k-1$. In case $q>0$, $\bar{y}=y+4\alpha_{k+1}s_{\pi(0)}u^{\pi(0)}$, $\bar{s}=s-2s_{\pi(0)}u^{\pi(0)}$, $\bar{\pi}=\pi$, $\bar{q}=q$, and $\bar{k}=k$.

$\underline{0\leq i<q-1}$: $\bar{y}=y$, $\bar{s}=s$, $\bar{\pi}=(\pi(0),\dots,\pi(i+1),\pi(i),\dots,\pi(n))$, $\bar{q}=q$, and $\bar{k}=k$.

$\underline{0\leq i=q-1}$: In case $y_{\pi(q-1)}=\alpha_k(w_{\pi(q-1)}-s_{\pi(q-1)})$, if $s_{\pi(q-1)}=t_{\pi(q-1)}$, then $\bar{y}=y$, $\bar{s}=s$, $\bar{\pi}=(\pi(0),\dots,\pi(q),\pi(q-1),\dots,\pi(n))$, $\bar{q}=q-1$, and $\bar{k}=k$; if $s_{\pi(q-1)}\neq t_{\pi(q-1)}$, then $\bar{y}=y$, $\bar{s}=s-2s_{\pi(q-1)}u^{\pi(q-1)}$, $\bar{\pi}=(\pi(0),\dots,\pi(q-2),\pi(q),\dots,\pi(n),\pi(q-1))$, $\bar{q}=q-1$, and $\bar{k}=k$. In case $y_{\pi(q-1)}\neq\alpha_k(w_{\pi(q-1)}-s_{\pi(q-1)})$, $\bar{y}=y$, $\bar{s}=s-2s_{\pi(q-1)}u^{\pi(q-1)}$, $\bar{\pi}=\pi$, $\bar{q}=q$, and $\bar{k}=k$.

$\underline{q=i<n}$: $\bar{y}=y$, $\bar{s}=s$, $\bar{\pi}=(\pi(0),\dots,\pi(q+1),\pi(q),\dots,\pi(n))$, $\bar{q}=q+1$, and $\bar{k}=k$.

$\underline{q<i<n}$: $\bar{y}=y$, $\bar{s}=s$, $\bar{\pi}=(\pi(0),\dots,\pi(i+1),\pi(i),\dots,\pi(n))$, $\bar{q}=q$, and $\bar{k}=k$.

$\underline{i=n}$: In case $q<n$, $\bar{y}=y$, $\bar{s}=s-2s_{\pi(n)}u^{\pi(n)}$, $\bar{\pi}=(\pi(0),\dots,\pi(q-1),\pi(n),\pi(q),\dots,\pi(n-1))$, $\bar{q}=q+1$, and $\bar{k}=k$. In case $q=n$, $\bar{y}=y+\alpha_{k+1}s$, $\bar{s}=s$, $\bar{\pi}=(\pi(n),\pi(0),\dots,\pi(n-1))$, $q=0$, and $\bar{k}=k+1$.

Finally, let a simplex of the D_2^*-triangulation, $\sigma=D_2^*(y,\pi,s,p_1,p_2)$, be given with vertices y^{-1}, y^0, $\dots$, y^n. We want to obtain the simplex of the D_2^*-triangulation, $\bar{\sigma}=D_2^*(\bar{y},\bar{\pi},\bar{s},\bar{p}_1,\bar{p}_2)$, such that all vertices of σ are also vertices of $\bar{\sigma}$ except the vertex y^i. As follows, we show how $\bar{y}$, $\bar{\pi}$, $\bar{s}$, $\bar{p}_1$, and $\bar{p}_2$ depend on y, π, s, p_1, p_2, and i.

$\underline{i=-1}$: In case $q=0$, $\bar{y}=y-\alpha_k s$, $\bar{s}=s$, $\bar{\pi}=(\pi(1),\dots,\pi(n),\pi(0))$, $\bar{p}_1=p_2-1$, $\bar{p}_2=0$, $\bar{q}=n$, and $\bar{k}=k-1$. In case $q=1$, $\bar{y}=y+4\alpha_{k+1}s_{\pi(0)}u^{\pi(0)}$, $\bar{s}=s-2s_{\pi(0)}u^{\pi(0)}$, $\bar{\pi}=\pi$, $\bar{p}_1=p_1$, $\bar{p}_2=p_2$, $\bar{q}=q$, and $\bar{k}=k$. In case $q>1$, when $p_1=-1$, $\bar{y}=y$, $\bar{s}=s$, $\bar{\pi}=\pi$, $\bar{p}_1=p_1+1$, $\bar{p}_2=p_2$, $\bar{q}=q$, and $\bar{k}=k$; when $p_1=0$, $\bar{y}=y$, $\bar{s}=s$, $\bar{\pi}=\pi$, $\bar{p}_1=p_1-1$, $\bar{p}_2=p_2$,

$\bar{q} = q$, and $\bar{k} = k$; when $p_1 \geq 1$ and $y_{\pi(0)} = \alpha_k(w_{\pi(0)} - s_{\pi(0)})$, if $h = 0$ and $p_2 = 0$, then $\bar{y} = y$, $\bar{s} = s - 2s_{\pi(0)}u^{\pi(0)}$, $\bar{\pi} = (\pi(1), \dots, \pi(n), \pi(0))$, $\bar{p}_1 = p_1 - 1$, $\bar{p}_2 = p_2$, $\bar{q} = q - 1$, and $\bar{k} = k$, if $h = 0$ and $p_2 \geq 1$, then $\bar{y} = y$, $\bar{s} = s - 2s_{\pi(0)}u^{\pi(0)}$, $\bar{\pi} = (\pi(1), \dots, \pi(q), \pi(0), \pi(q+1), \dots, \pi(n))$, $\bar{p}_1 = p_1 - 1$, $\bar{p}_2 = p_2 + 1$, $\bar{q} = q - 1$, and $\bar{k} = k$, if $s_{\pi(0)} = t_{\pi(0)}$ and $h = 1$, then $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(1), \dots, \pi(n), \pi(0))$, $\bar{p}_1 = p_1 - 1$, $\bar{p}_2 = p_2$, $\bar{q} = q - 1$, and $\bar{k} = k$, if $s_{\pi(0)} = t_{\pi(0)}$ and $h > 1$, then $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(1), \dots, \pi(n), \pi(0))$, $\bar{p}_1 = p_1 - 1$, $\bar{p}_2 = p_2 + 1$, $\bar{q} = q - 1$, and $\bar{k} = k$, and if $s_{\pi(0)} \neq t_{\pi(0)}$ and $h > 0$, then $\bar{y} = y$, $\bar{s} = s - 2s_{\pi(0)}u^{\pi(0)}$, $\bar{\pi} = (\pi(1), \dots, \pi(q), \pi(0), \pi(q+1), \dots, \pi(n))$, $\bar{p}_1 = p_1 - 1$, $\bar{p}_2 = p_2 + 1$, $\bar{q} = q - 1$, and $\bar{k} = k$; when $p_1 \geq 1$ and $y_{\pi(0)} \neq \alpha_k(w_{\pi(0)} - s_{\pi(0)})$, $\bar{y} = y$, $\bar{s} = s - 2s_{\pi(0)}u^{\pi(0)}$, $\bar{\pi} = \pi$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2$, $\bar{q} = q$, and $\bar{k} = k$.

$\underline{0 \leq i < q}$: In case $p_1 = -1$, when $y_{\pi(i)} = \alpha_k(w_{\pi(i)} - s_{\pi(i)})$, if $h = 0$ and $p_2 = 0$, then $\bar{y} = y$, $\bar{s} = s - 2s_{\pi(i)}u^{\pi(i)}$, $\bar{\pi} = (\pi(0), \dots, \pi(i-1), \pi(i+1), \dots, \pi(n), \pi(i))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2$, $\bar{q} = q - 1$, and $\bar{k} = k$, if $h = 0$ and $p_2 \geq 1$, then $\bar{y} = y$, $\bar{s} = s - 2s_{\pi(i)}u^{\pi(i)}$, $\bar{\pi} = (\pi(0), \dots, \pi(i-1), \pi(i+1), \dots, \pi(q), \pi(i), \pi(q+1), \dots, \pi(n))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2 + 1$, $\bar{q} = q - 1$, and $\bar{k} = k$, if $s_{\pi(i)} = t_{\pi(i)}$ and $h = 1$, then $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(0), \dots, \pi(i-1), \pi(i+1), \dots, \pi(n), \pi(i))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2$, $\bar{q} = q - 1$, and $\bar{k} = k$, if $s_{\pi(i)} = t_{\pi(i)}$ and $h > 1$, then $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(0), \dots, \pi(i-1), \pi(i+1), \dots, \pi(n), \pi(i))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2 + 1$, $\bar{q} = q - 1$, and $\bar{k} = k$, and if $s_{\pi(i)} \neq t_{\pi(i)}$ and $h > 0$, then $\bar{y} = y$, $\bar{s} = s - 2s_{\pi(i)}u^{\pi(i)}$, $\bar{\pi} = (\pi(0), \dots, \pi(i-1), \pi(i+1), \dots, \pi(q), \pi(i), \pi(q+1), \dots, \pi(n))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2 + 1$, $\bar{q} = q - 1$, and $\bar{k} = k$; when $y_{\pi(i)} \neq \alpha_k(w_{\pi(i)} - s_{\pi(i)})$, $\bar{y} = y$, $\bar{s} = s - 2s_{\pi(i)}u^{\pi(i)}$, $\bar{\pi} = \pi$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2$, $\bar{q} = q$, and $\bar{k} = k$. In case $i < p_1 - 1$, $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(0), \dots, \pi(i+1), \pi(i), \dots, \pi(n))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2$, $\bar{q} = q$, and $\bar{k} = k$. In case $i = p_1 - 1$, $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = \pi$, $\bar{p}_1 = p_1 - 1$, $\bar{p}_2 = p_2$, $\bar{q} = q$, and $\bar{k} = k$. In case $i \geq p_1$ and $0 \leq p_1 < q - 2$, $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(0), \dots, \pi(p_1 - 1), \pi(i), \pi(p_1), \dots, \pi(i-1), \pi(i+1), \dots, \pi(n))$, $\bar{p}_1 = p_1 + 1$, $\bar{p}_2 = p_2$, $\bar{q} = q$, and $\bar{k} = k$. In case $i \geq q - 2$ and $0 \leq p_1 = q - 2$, $\bar{y} = y + 4\alpha_{k+1}s_{\pi(i^*)}u^{\pi(i^*)}$, $\bar{s} = s - 2s_{\pi(i^*)}u^{\pi(i^*)}$, $\bar{\pi} = \pi$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2$, $\bar{q} = q$, and $\bar{k} = k$, where

$$i^* = \begin{cases} q-1 & \text{if } i = q-2, \\ q-2 & \text{if } i = q-1. \end{cases}$$

$\underline{i = q}$: In case $h = 0$, when $p_2 = 0$, if $q < n - 1$, then $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = \pi$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2 + 1$, $\bar{q} = q$, and $\bar{k} = k$, if $q = n - 1$ and $p_1 = -1$, then $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(0), \dots, \pi(q+1), \pi(q), \dots, \pi(n))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2$, $\bar{q} = q + 1$, and $\bar{k} = k$, if $q = n - 1$ and $p_1 \geq 0$, then $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(q+1), \pi(0), \dots, \pi(q), \pi(q+2), \dots, \pi(n))$, $\bar{p}_1 = p_1 + 1$, $\bar{p}_2 = p_2$, $\bar{q} = q + 1$, and $\bar{k} = k$; when $p_2 = 1$, $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = \pi$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2 - 1$, $\bar{q} = q$, and $\bar{k} = k$; when $p_2 \geq 2$, if $p_1 = -1$, then $\bar{y} = y$, $\bar{s} = s - 2s_{\pi(q+1)}u^{\pi(q+1)}$, $\bar{\pi} = (\pi(0), \dots, \pi(q+1), \pi(q), \dots, \pi(n))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2 - 1$, $\bar{q} = q + 1$, and $\bar{k} = k$, if $p_1 \geq 0$, then $\bar{y} = y$, $\bar{s} = s - 2s_{\pi(q+1)}u^{\pi(q+1)}$, $\bar{\pi} = (\pi(q+1), \pi(0), \dots, \pi(q), \pi(q+2), \dots, \pi(n))$, $\bar{p}_1 = p_1 + 1$, $\bar{p}_2 = p_2 - 1$, $\bar{q} = q + 1$, and $\bar{k} = k$. In case $h > 0$, when $q < n$, if $p_1 = -1$, then $\bar{y} = y$, $\bar{s} = s - 2s_{\pi(q+1)}u^{\pi(q+1)}$, $\bar{\pi} = (\pi(0), \dots, \pi(q+1), \pi(q), \dots, \pi(n))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2 - 1$, $\bar{q} = q + 1$, and $\bar{k} = k$, and if $p_1 \geq 0$, then $\bar{y} = y$, $\bar{s} = s - 2s_{\pi(q+1)}u^{\pi(q+1)}$, $\bar{\pi} = (\pi(q+1), \pi(0), \dots, \pi(q), \pi(q+2), \dots, \pi(n))$,

$\bar{p}_1 = p_1 + 1$, $\bar{p}_2 = p_2 - 1$, $\bar{q} = q + 1$, and $\bar{k} = k$. In case $q = n$, $\bar{y} = y + \alpha_{k+1}s$, $\bar{s} = s$, $\bar{\pi} = (\pi(n), \pi(0), \ldots, \pi(n-1))$, $\bar{p}_1 = -1$, $\bar{p}_2 = p_1 + 1$, $\bar{q} = 0$ and $\bar{k} = k + 1$.

$\underline{q < i \leq n}$: In case $h = 0$, when $p_2 = 0$, if $p_1 = -1$, then $\bar{y} = y$, $\bar{s} = s - 2s_{\pi(i)}u^{\pi(i)}$, $\bar{\pi} = (\pi(0), \ldots, \pi(q-1), \pi(i), \pi(q), \ldots, \pi(i-1), \pi(i+1), \ldots, \pi(n))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2$, $\bar{q} = q + 1$, and $\bar{k} = k$, and if $p_1 \geq 0$, then $\bar{y} = y$, $\bar{s} = s - 2s_{\pi(i)}u^{\pi(i)}$, $\bar{\pi} = (\pi(i), \pi(0), \ldots, \pi(i-1), \pi(i+1), \ldots, \pi(n))$, $\bar{p}_1 = p_1 + 1$, $\bar{p}_2 = p_2$, $\bar{q} = q + 1$, and $\bar{k} = k$; when $i < q + p_2 - 1$, $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(0), \ldots, \pi(i+1), \pi(i), \ldots, \pi(n))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2$, $\bar{q} = q$, and $\bar{k} = k$; when $i = q + p_2 - 1$, $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = \pi$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2 - 1$, $\bar{q} = q$, and $\bar{k} = k$; when $i \geq q + p_2$ and $1 \leq p_2 < n - q - 1$, $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(0), \ldots, \pi(q + p_2 - 1), \pi(i), \pi(q + p_2), \ldots, \pi(i-1), \pi(i+1), \ldots, \pi(n))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2 + 1$, $\bar{q} = q$, and $\bar{k} = k$; when $i \geq n - 1$ and $1 \leq p_2 = n - q - 1$, if $p_1 = -1$, then $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(0), \ldots, \pi(q-1), \pi(i^{**}), \pi(q), \ldots, \pi(i^{**} - 1), \pi(i^{**} + 1), \ldots, \pi(n))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2$, $\bar{q} = q + 1$, and $\bar{k} = k$, and if $p_1 \geq 0$, then $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(i^{**}), \pi(0), \ldots, \pi(i^{**} - 1), \pi(i^{**} + 1), \ldots, \pi(n))$, $\bar{p}_1 = p_1 + 1$, $\bar{p}_2 = p_2$, $\bar{q} = q + 1$, and $\bar{k} = k$, where

$$i^{**} = \begin{cases} n & \text{if } i = n - 1, \\ n - 1 & \text{if } i = n. \end{cases}$$

In case $h > 0$, when $i < n$, $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(0), \ldots, \pi(i+1), \pi(i), \ldots, \pi(n))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2$, $\bar{q} = q$, and $\bar{k} = k$; when $i = n$, if $p_1 = -1$, then $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(0), \ldots, \pi(q-1), \pi(n), \pi(q), \ldots, \pi(n-1))$, $\bar{p}_1 = p_1$, $\bar{p}_2 = p_2 - 1$, $\bar{q} = q + 1$, and $\bar{k} = k$, and if $p_1 \geq 0$, then $\bar{y} = y$, $\bar{s} = s$, $\bar{\pi} = (\pi(n), \pi(0), \ldots, \pi(n-1))$, $\bar{p}_1 = p_1 + 1$, $\bar{p}_2 = p_2 - 1$, $\bar{q} = q + 1$, and $\bar{k} = k$.

5. Comparison of triangulations.

Since it is very complicated to calculate the surface density of the K_2^*-triangulation, the J_2^*-triangulation, and the D_2^*-triangulation, we only compare the number of simplices of these triangulations. For details about the surface density, we refer to [3] and [12]. Let H^n denote the unit cube $\{x \in R^n \mid 0 \leq x_i \leq 1 \text{ for } i = 1, 2, \ldots, n\}$. We set $\alpha = 1/\beta_k$.

THEOREM 5.1. *The number of simplices of the K_2^*-triangulation or J_2^*-triangulation in the set $[2^{-(k+1)}, 2^{-k}] \times 2\alpha_k H^n$ is equal to $p_n(\alpha)$ given by*

$$p_n(\alpha) = ((2\alpha)^{n+1} - 1)n!/(2\alpha - 1).$$

The number of simplices of the D_2^*-triangulation in the same set is equal to $q_n(\alpha)$ given by

$$q_n(\alpha) = \sum_{m=0}^{n} ((2^m - 1)C_n^m \alpha^m d_m (n-m)! + C_n^m \alpha^m d_m d_{n-m}),$$

where

$$d_j = j + j(j-1) + \cdots + j(j-1)\cdots 4 \cdot 3 + 2$$

for $j \geq 2$, $d_0 = d_1 = 1$, and $C_n^m = n!/m!(n-m)!$.

Proof. Let $\bar{Q}$ denote the set $\{w \in R^n \mid w_i \in \{0, 1, 2\} \text{ for } i = 1, 2, \ldots, n\}$. We take an arbitrary vector $w \in \bar{Q}$. Let $\bar{A}(w)$ denote the set

$$\{x \in R^n \mid w_i - 1 \leq x_i \leq w_i + 1 \text{ for } i \in I_o(w),\ x_i = w_i \text{ for } i \in I_e(w)\}$$

and let $\bar{B}(w)$ denote the set

$$\left\{ x \in R^n \;\middle|\; \begin{array}{l} x_i = w_i \text{ for } i \in I_o(w), \\ w_i \le x_i \le w_i + 1 \text{ for } i \in I_e(w) \text{ and } w_i = 0, \\ w_i - 1 \le x_i \le w_i \text{ for } i \in I_e(w) \text{ and } w_i = 2 \end{array} \right\}.$$

Furthermore, let $\alpha_k\bar{D}(w)$ denote the convex hull of the set

$$(\{2^{-k}\} \times \alpha_k\bar{A}(w)) \cup (\{2^{-(k+1)}\} \times \alpha_k\bar{B}(w)).$$

Then it can be seen that

$$[2^{-(k+1)}, 2^{-k}] \times 2\alpha_k H^n = \cup_{w \in \bar{Q}} \alpha_k \bar{D}(w).$$

Let m denote the number of elements in $I_e(w)$. Then there are $2^m C_n^m$ elements in $\bar{Q}$ such that m components of each of them are even. Thus the numbers of simplices of the K_2^*-triangulation or J_2^*-triangulation in the set

$$\cup_{w \in \bar{Q},\ |I_e(w)|=m} \alpha_k \bar{D}(w)$$

is equal to

$$2^m \alpha^m C_n^m (n-m)!m!(= (2\alpha)^m n!).$$

The number of simplices of the D_2^*-triangulation in the same set is equal to

$$(2^m - 1)C_n^m \alpha^m d_m (n-m)! + C_n^m \alpha^m d_m d_{n-m}.$$

Since

$$\cup_{m=0}^n (\cup_{w \in \bar{Q},\ |I_e(w)|=m} \alpha_k \bar{D}(w)) = [2^{-(k+1)}, 2^{-k}] \times 2\alpha_k H^n,$$

the theorem follows immediately. □

THEOREM 5.2. *When $n \ge 3$, $q_n(\alpha) < p_n(\alpha)$. As n approaches infinity, $q_n(\alpha)/p_n(\alpha)$ converges to $e - 2$.*

Proof. The conclusion is obvious, so the proof is omitted. □

From Theorem 5.2, we have that the number of simplices of the D_2^*-triangulation is the smallest of these three triangulations.

As follows, a few numerical tests are given to show that the continuous deformation algorithm based on the D_2^*-triangulation is indeed much more efficient. Let us denote the continuous deformation algorithms based on the K_2^*-triangulation, J_2^*-triangulation, and D_2^*-triangulation by CDAK$_2^*$, CDAJ$_2^*$, and CDAD$_2^*$, respectively. We have made computer codes of these algorithms in PASCAL. As we noted when discussing the principles of the continuous deformation algorithm in §1, letting A be the identity matrix, we have run these computer codes on a few tests for finding a zero point with several different initial points x^0. Let NFE denote the number of function evaluations. The algorithm terminates when the accuracy for $\max_{1 \le i \le n} |f_i(x^*)|$ of less than 10^{-5} has been reached. In Tables 1–9, if the accuracy has not been satisfied when the number of function evaluations is equal to 50,000, a symbol * is marked.

TABLE 1

n	NFE(CDAK$_2^*$)	NFE(CDAJ$_2^*$)	NFE(CDAD$_2^*$)
5	397	314	195
6	862	396	612
7	2086	1418	878
8	5099	2675	1847
9	9686	3991	2996
10	43,719	4476	3001

Problem A. The function $f : R^n \longrightarrow R^n$ is given by

$$f_i(x) = x_i - \cos\left(i\sum_{j=1}^{n} x_j\right), \qquad i = 1, 2, \ldots, n.$$

When $x_i^0 = 10$ for $i = 1, 2, \ldots, n$, $\alpha_0 = 5$, and $\beta_j = 1$ for $j = 0, 1, \ldots$, the numerical results are given in Table 1.

When $x_i^0 = 10$ for $i = 1, 2, \ldots, n$, $\alpha_0 = 5$, and $\beta_{2j} = 1$ and $\beta_{2j+1} = 0.5$ for $j = 0, 1, \ldots$, the numerical results are given in Table 2.

TABLE 2

n	NFE(CDAK$_2^*$)	NFE(CDAJ$_2^*$)	NFE(CDAD$_2^*$)
5	384	351	216
6	762	475	802
7	2070	2218	755
8	3063	2907	1516
9	15,045	9520	8008
10	24,404	8696	7447

When $x_i^0 = i$ for $i = 1, 2, \ldots, n$, $\alpha_0 = 5$, and $\beta_j = 1$ for $j = 0, 1, \ldots$, the numerical results are given in Table 3.

TABLE 3

n	NFE(CDAK$_2^*$)	NFE(CDAJ$_2^*$)	NFE(CDAD$_2^*$)
5	356	248	209
6	789	292	307
7	1908	585	653
8	4478	896	665
9	8616	2496	1929
10	24,774	3206	3315
11	*	4198	4513
12	*	10,289	7245

When $x_i^0 = i$ for $i = 1, 2, \ldots, n$ and $\alpha_0 = 5$, and $\beta_{2j} = 1$, and $\beta_{2j+1} = 0.5$ for $j = 0, 1, \ldots$, the numerical results are given in Table 4.

Problem B. The function $f : R^n \longrightarrow R^n$ is given by

$$f_i(x) = x_i - e^{\cos(i\sum_{j=1}^{n} x_j)}, \qquad i = 1, 2, \ldots, n.$$

When $x_i^0 = 10$ for $i = 1, 2, \ldots, n$ and $\alpha_0 = 5$, and $\beta_j = 1$ for $j = 0, 1, \ldots$, the numerical results are given in Table 5.

TABLE 4

n	NFE(CDAK$_2^*$)	NFE(CDAJ$_2^*$)	NFE(CDAD$_2^*$)
5	508	223	179
6	821	363	321
7	1423	736	518
8	4260	1006	865
9	9665	1771	2335
10	31,156	6226	4875

TABLE 5

n	NFE(CDAK$_2^*$)	NFE(CDAJ$_2^*$)	NFE(CDAD$_2^*$)
5	463	403	228
6	1483	707	450
7	7230	1978	1180
8	10,529	3123	2293
9	*	8938	4163
10	*	11,943	3777

When $x_i^0 = 10$ for $i = 1, 2, \ldots, n$, $\alpha_0 = 5$, and $\beta_{2j} = 1$; and $\beta_{2j+1} = 0.5$ for $j = 0, 1, \ldots$, the numerical results are given in Table 6.

TABLE 6

n	NFE(CDAK$_2^*$)	NFE(CDAJ$_2^*$)	NFE(CDAD$_2^*$)
5	454	430	234
6	1004	672	336
7	4506	2448	1761
8	15,397	3320	3532
9	*	12,588	13,962
10	*	6853	14,460
11	*	*	40,286

When $x_i^0 = i$ for $i = 1, 2, \ldots, n$ and $\alpha_0 = 5$, and $\beta_j = 1$ for $j = 0, 1, \ldots$, the numerical results are given in Table 7.

When $x_i^0 = i$ for $i = 1, 2, \ldots, n$, $\alpha_0 = 5$ and $\beta_{2j} = 1$; and $\beta_{2j+1} = 0.5$ for $j = 0, 1, \ldots$, the numerical results are given in Table 8.

Problem C. The function $f : R^n \to R^n$ is given by

$$f_i(x) = x_i - \left(i + \sum_{j=1}^{n} x_j^3 \right) / n^2, \qquad i = 1, 2, \ldots, n.$$

When $x_i^0 = 0$ for $i = 1, 2, \ldots, n$ and $\alpha_0 = 0.5$, and $\beta_j = 1$ for $j = 0, 1, \ldots$, the numerical results are given in Table 9.

From these numerical examples, it seems clear that the continuous deformation algorithm based on the D_2^*-triangulation is indeed more efficient.

Acknowledgments. The author would like to thank Dolf Talman for his remarks on an earlier version of this paper and two anonymous referees for their constructive suggestions. The first version of this paper was completed when the author

TABLE 7

n	NFE(CDAK$_2^*$)	NFE(CDAJ$_2^*$)	NFE(CDAD$_2^*$)
5	384	169	201
6	1615	823	646
7	3850	1415	1000
8	7256	1609	933
9	*	1422	8071
10	*	8462	5942

TABLE 8

n	NFE(CDAK$_2^*$)	NFE(CDAJ$_2^*$)	NFE(CDAD$_2^*$)
5	494	167	217
6	1218	673	491
7	7176	2780	1106
8	10,825	3207	2116
9	*	2444	3775
10	*	6799	5553

TABLE 9

n	NFE(CDAK$_2^*$)	NFE(CDAJ$_2^*$)	NFE(CDAD$_2^*$)
10	115	100	78
20	274	274	125
30	559	559	185
40	944	944	245
50	1429	1429	305
60	2014	2014	365
70	2699	2699	425
80	3484	3484	485
90	4369	4369	545
100	5354	5354	605

was visiting the Center for Economic Research of Tilburg University in the Netherlands.

REFERENCES

[1] E. L. ALLGOWER AND K. GEORG, *Simplicial and continuation methods for approximating fixed points and solutions to systems of equations*, SIAM Review, 22 (1980), pp. 28–85.

[2] M. N. BROADIE AND B. C. EAVES, *A variable rate refining triangulation*, Math. Programming, 38 (1987), pp. 161–202.

[3] C. DANG, *The D_1-triangulation of R^n for simplicial algorithms for computing solutions of nonlinear equations*, Math. Oper. Res., 16 (1991), pp. 148–161.

[4] ———, *D_3-triangualtion for simplicial deformation algorithms for computing solutions of nonlinear equations*, Discussion paper 8949, Center for Economic Research, Tilburg Univ., the Netherlands, 1989; J. Optim. Theory Appl., to appear.

[5] ———, *The D_2-triangulation for simplicial homotopy algorithms for computing solutions of nonlinear equations*, Discussion paper 9024, Center for Economic Research, Tilburg Univ., the Netherlands, 1990.

[6] ———, *The D_1-Triangulation in Simplicial Algorithms*, Ph.D. thesis, Center for Economic Research, Tilburg Univ., the Netherlands, 1991.

[7] T. M. DOUP, *Simplicial Algorithms on the Simplotope*, Lecture Notes on Economics and Math-

ematical Systems, Springer-Verlag, Berlin, 1988.

[8] T. M. Doup and A. J. J. Talman, *A continuous deformation algorithm on the product space of unit simplices*, Math. Oper. Res., 12 (1987), pp. 485–521.

[9] B. C. Eaves, *Homotopies for computation of fixed points*, Math. Programming, 3 (1972), pp. 1–22.

[10] ———, *A Course in Triangulations for Solving Equations with Deformations*, Lecture Notes on Economics and Mathematical Systems, Springer-Verlag, Berlin, 1984.

[11] B. C. Eaves and R. Saigal, *Homotopies for the computation of fixed points on unbounded regions*, Math. Programming, 3 (1972), pp. 225–237.

[12] B. C. Eaves and J. A. Yorke, *Equivalence of surface density and average directional density*, Math. Oper. Res., 9 (1984), pp. 363–375.

[13] M. Haiman, *A simple and relatively efficient triangulation of the n-cube*, manuscript, Dept. of Mathematics, Massachusetts Inst. of Technology, Cambridge, MA, 1989.

[14] M. Kojima and Y. Yamamoto, *Variable dimension algorithms: Basic theory, interpretation, and extensions of some existing methods*, Math. Programming, 24 (1982), pp. 177–215.

[15] G. van der Laan and A. J. J. Talman, *A new subdivision for computing fixed points with a homotopy algorithm*, Math. Programming, 19 (1980), pp. 78–91.

[16] C. Lee, *Triangulating the d-cube*, in Discrete Geometry and Convexity, J. E. Goodman, E. Lutwak, J. Malkevitch, and R. Pollack, eds., New York Academy of Sciences, New York, 1985, pp. 205–211.

[17] R. Saigal, *Investigations into the efficiency of fixed point algorithms*, in Fixed Points: Algorithms and Applications, S. Karamardian, ed., Academic Press, New York, pp. 203–223.

[18] ———, *On the convergence rate of algorithms for solving equations that are based on methods of complementary pivoting*, Math. Oper. Res., 1 (1976), pp. 359–380.

[19] J. F. Sallee, *Middle cut triangulations of the n-cube*, SIAM J. Algebraic Discrete Meth., 5 (1984), pp. 407–418.

[20] H. Scarf, *The approximation of fixed points of a continuous mapping*, SIAM J. Appl. Math., 15 (1967), pp. 1328–1343.

[21] S. Shamir, *Two triangulations for homotopy fixed point algorithms with an arbitrary refinement factor*, in Analysis and Computation of Fixed Points, S. M. Robinson, ed., Academic Press, New York, 1980, pp. 25–56.

[22] M. J. Todd, *The Computation of Fixed Points and Applications*, Lecture Notes on Economics and Mathematical Systems, Springer-Verlag, Berlin, 1976.

[23] ———, *On triangulations for computing fixed points*, Math. Programming, 10 (1976), pp. 322–346.

[24] M. J. Todd and L. Tuncel, *A new triangulation for simplicial algorithms*, Tech. Rep. 946, School of Operations Research and Industrial Engineering, Cornell Univ., New York, 1990.

SIAM J. OPTIMIZATION
Vol. 3, No. 4, pp. 800–825, November 1993

α-LOWER SUBDIFFERENTIABLE FUNCTIONS*

J. E. MARTÍNEZ-LEGAZ† AND S. ROMANO-RODRÍGUEZ‡

Abstract. In this paper the authors introduce the notion of α-lower subdifferentiability, with $\alpha \in (0,1]$, for extended real-valued functions defined on a locally convex real topological vector space X. This is a generalization of the concept of lower subdifferentiability due to Plastria, which corresponds to the case $\alpha = 1$. When X is a normed space, the class of α-lower subdifferentiable functions appears to be closely related to that of α-Hölder quasi-convex functions. Two applications to quasi-convex optimization are given: a duality theorem, based on conjugation with respect to h_α, and Kuhn–Tucker-type optimality conditions in terms of α-lower subdifferentials.

Key words. generalized subdifferentiability, quasi convexity, generalized conjugation theory, dual problem, lower semicontinuous quasi-convex hull

AMS subject classification. 26B25

1. Introduction.

For algorithmic purposes, Plastria [11] introduced the notions of lower subdifferentiable (l.s.d.) functions and boundedly lower subdifferentiable (b.l.s.d.) functions by relaxing the subdifferentiability concept of convex functions. He proved that a l.s.d. function defined on a closed convex set is quasi-convex and lower semicontinuous (l.s.c.) and that a function defined on the whole space is b.l.s.d. if and only if it is Lipschitzian and quasi-convex.

We will introduce the concept of α-lower subdifferentiable (α-l.s.d.) functions, for $\alpha \in (0,1]$, in such a way that Plastria's notion corresponds to the case $\alpha = 1$. We will see that α-Hölder functions play, with respect to this notion, an analogous role as Lipschitzian functions with respect to lower subdifferentiability.

We will study the notion of α-lower subdifferentiability from the viewpoint of the generalized conjugation theory of Moreau [9]. We will also see that the concept of α-lower subdifferentiability can be obtained as a particular case of the generalized subdifferentiability of Balder [1]. For the case $\alpha = 1$, this study was made by Martínez-Legaz [7], [8] and Penot and Volle [10].

We shall use the following notation. X will be a locally convex real topological vector space, $X \neq \{0\}$, with topological dual X^*. In §§3 and 5, we shall impose X to be a normed space. By $\overline{\mathbb{R}}$ we shall denote the extended real line $[-\infty, +\infty]$; $\mathbb{R}^+$ will be the set of nonnegative real numbers. The level sets (strict level sets, respectively) of $f : K \subset X \longrightarrow \overline{\mathbb{R}}$ are

$$S_\lambda(f) = \{x \in K \mid f(x) \leq \lambda\}$$

and

$$\dot{S}_\lambda(f) = \{x \in K \mid f(x) < \lambda\},$$

where $\lambda \in \mathbb{R}$. A function f is quasi-convex when its level sets (or, equivalently, its strict level sets) are convex. One says that f is quasi-affine if $K = X$ and both

* Received by the editors January 6, 1992; accepted for publication (in revised form) August 24, 1992.

† Departament Economica i Història Econòmica, Universitat Autònoma de Barcelona and Departament Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via 585, 08071 Barcelona, Spain. This author was supported by Dirección General de Investigación Científica y Técnica (DGICYT) project PB92-0615.

‡ Departament Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via 585, 08071 Barcelona, Spain.

f and $-f$ are quasi-convex. Given a function f, we will denote its epigraph, lower semicontinuous hull, quasi-convex hull and lower semicontinuous quasi-convex hull by epi$f$, $\bar{f}$, f_q, and $f_{\bar{q}}$, respectively. The convex hull and the closed convex hull of $K \subset X$ will be denoted by co$K$ and $\overline{\text{co}}K$, respectively; we will use the symbol cone K to represent the cone generated by K : cone$K = \{\lambda x | \lambda > 0, x \in K\}$.

We will consider the extension of the potential function of exponent $\alpha \in (0,1)$ to the real line that results in defining $x^\alpha = -\infty$ if $x < 0$. It is easy to verify that this is the unique extension that preserves concavity.

We recall the generalized conjugation theory of Moreau [9] and the generalized subdifferential of Balder [1].

Let C and D be two arbitrary sets and let $c : C \times D \longrightarrow \mathbb{R}$ be a function that we will call a *coupling function.*

Given $f : C \longrightarrow \overline{\mathbb{R}}$ we define its c-conjugate, $f^c : D \longrightarrow \overline{\mathbb{R}}$, by

$$f^c(y) = \sup_{x \in C}\{c(x,y) - f(x)\}.$$

In the same way, given $g : D \longrightarrow \overline{\mathbb{R}}$, its c-conjugate, $g^c : C \longrightarrow \overline{\mathbb{R}}$, is defined by means of

$$g^c(x) = \sup_{y \in D}\{c(x,y) - g(y)\}.$$

An elementary function on C with respect to c is a function of the form $c(\cdot, b) + \beta$ where $b \in D$ and $\beta \in \overline{\mathbb{R}}$. The elementary functions on D are defined analogously.

The set of functions from C into $\overline{\mathbb{R}}$ (from D into $\overline{\mathbb{R}}$, respectively) that are suprema of elementary functions is denoted by $\Gamma(C,D)$ ($\Gamma(D,C)$, respectively). From this definition, one deduces that $f^c \in \Gamma(D,C)$ for every function $f : C \longrightarrow \overline{\mathbb{R}}$ and, analogously, $g^c \in \Gamma(C,D)$ for every function $g : D \longrightarrow \overline{\mathbb{R}}$.

The Γ-regularized of $f : C \longrightarrow \overline{\mathbb{R}}$ is the supremum of elementary functions which are minorants of f. The Γ-regularized of f coincides with f^{cc} (see [9, p. 123]). Therefore,

$$f = f^{cc} \Longleftrightarrow f \in \Gamma(C,D).$$

For $g : D \longrightarrow \overline{\mathbb{R}}$ analogous properties hold.

Following the definition of Balder (see [1, p. 332]), we say that a function $f : C \longrightarrow \overline{\mathbb{R}}$ is *c-subdifferentiable at* $x_0 \in C$ if $f(x_0) \in \mathbb{R}$ and there exists $y_0 \in D$ such that

$$f(x) - f(x_0) \geq c(x, y_0) - c(x_0, y_0) \quad \text{for any } x \in C.$$

An element y_0 satisfying this property is called a *c-subgradient of f at* x_0. The set of all c-subgradients of f at x_0 is called the *c-subdifferential of f at* x_0 and is denoted by $\partial_c f(x_0)$.

The following properties hold (see Balder [1, p. 332]):

$$y_0 \in \partial_c f(x_0) \Longleftrightarrow f(x_0) + f^c(y_0) = c(x_0, y_0) \quad \text{for } x_0 \in C, y_0 \in D, \tag{1.1}$$

$$\partial_c f(x_0) \neq \emptyset \Longrightarrow f^{cc}(x_0) = f(x_0) \quad \text{for } x_0 \in C, \tag{1.2}$$

$$f^{cc}(x_0) = f(x_0) \Longrightarrow \partial_c f^{cc}(x_0) = \partial_c f(x_0) \quad \text{for } x_0 \in C. \tag{1.3}$$

In §§2 and 3, we give the definition and properties of α-l.s.d. functions and of α-b.l.s.d. functions. In §4, we provide a conjugation theory for α-l.s.d. functions and we see that, for a certain coupling function, the notions of generalized subdifferentiability and α-lower subdifferentiability coincide. By restricting this coupling function to an appropriate set, we obtain in §5 a conjugation theory suitable for α-b.l.s.d. functions defined on normed spaces; we shall also prove that they are just the α-Hölder quasi-convex functions. An application of the above theory to quasi-convex optimization is given in §6.

2. Definition and properties of α-lower subdifferentiable functions.

DEFINITION 2.1. Let $\alpha \in (0,1]$, $K \subset X$ be a convex set and $f : K \longrightarrow \overline{\mathbb{R}}$. We say that f is *α-l.s.d. at* $x_0 \in K$ if $f(x_0) \in \mathbb{R}$ and there exists $\omega \in X^*$ such that $f(x) \geq f(x_0) - (\omega(x_0 - x))^\alpha$ for any $x \in K$ with $f(x) < f(x_0)$.

The continuous linear function ω is said to be an *α-lower subgradient of f at* x_0. The set of α-lower subgradients of f at x_0 is called the *α-lower subdifferential of f at* x_0 and is denoted by $\partial_\alpha^- f(x_0)$. We say that f is *α-l.s.d.* if it has an α-lower subgradient at every point of K.

Notice that, according to the way we have extended the potential function, if $\omega \in \partial_\alpha^- f(x_0)$ and $x \in K$ is such that $f(x) < f(x_0)$, then $\omega(x_0 - x) > 0$.

By extending $f : K \longrightarrow \overline{\mathbb{R}}$ by means of $\hat{f}$ defined by $\hat{f}(x) = f(x)$ if $x \in K$ and $\hat{f}(x) = +\infty$ if $x \notin K$, we can consider each function as defined in the whole space; then $\partial_\alpha^- \hat{f}(x_0) = \partial_\alpha^- f(x_0)$ if $x_0 \in K$ and $\partial_\alpha^- \hat{f}(x_0) = \emptyset$ if $x_0 \notin K$.

PROPOSITION 2.1. *Let $f : X \longrightarrow \mathbb{R}$. If f is α-l.s.d., then f is quasi-convex and l.s.c.*

Proof. Let $\lambda \in \mathbb{R}$ and $x_0 \in X$ be such that $x_0 \notin S_\lambda(f)$. Take $\omega \in \partial_\alpha^- f(x_0)$. Define

$$H_{x_0} = \{x \in X \mid \omega(x_0 - x) \geq (f(x_0) - \lambda)^{1/\alpha}\}.$$

Let $x \in S_\lambda(f)$. Since $\lambda \geq f(x) \geq f(x_0) - (\omega(x_0 - x))^\alpha$, we can write

$$(f(x_0) - \lambda)^{1/\alpha} \leq \omega(x_0 - x).$$

We deduce that $S_\lambda(f) \subset H_{x_0}$. Therefore, as $x_0 \notin H_{x_0}$, $S_\lambda(f)$ is an intersection of closed halfspaces, whence it is convex and closed. □

There exist quasi-convex continuous functions that are not α-l.s.d. For example, take the function $f : \mathbb{R} \longrightarrow \mathbb{R}$ given by $f(x) = x^3$. Let any $\alpha \in (0,1]$. Then f is not α-l.s.d. at any $x \in \mathbb{R}$. Actually, $f(x) = x$ is not α-l.s.d. at any $x \in \mathbb{R}$ for $\alpha \in (0,1)$ either (but it is l.s.d. at all points).

In general, α-lower subdifferentials of f at x_0 are not monotonically depending on α. To see this, take the function $f : \mathbb{R} \longrightarrow \mathbb{R}$ defined by

$$f(x) = \min\{-(-x)^\alpha, 0\}.$$

It is easy to prove that $1 \in \partial_\alpha^- f(0)$ and that, for any $\beta \in (0,1]$ such that $\beta \neq \alpha$, we have $\partial_\beta^- f(0) = \emptyset$. Therefore, $\partial_\alpha^- f(0) \not\subset \partial_\beta^- f(0)$ if $\beta \neq \alpha$.

The following proposition states some properties of α-lower subdifferentials.

PROPOSITION 2.2. *Let $f : X \longrightarrow \overline{\mathbb{R}}$ and $x_0 \in X$. Then*

(1) *$\partial_\alpha^- f(x_0)$ is an ω^*-closed convex set.*

(2) *If $\omega \in \partial_\alpha^- f(x_0)$ and $a \geq 1$, then $a\omega \in \partial_\alpha^- f(x_0)$.*

(3) *$0 \in \partial_\alpha^- f(x_0) \Longleftrightarrow x_0$ is a global minimum of $f \Longleftrightarrow \partial_\alpha^- f(x_0) = X^*$.*

Proof. (1) We can write

$$\partial_\alpha^- f(x_0) = \{\omega \mid \omega(x_0 - x) \geq (f(x_0) - f(x))^{1/\alpha} \text{ for all } x \in X \text{ with } f(x) < f(x_0)\},$$

which shows that this set is ω^*-closed and convex.

(2) Let $\omega \in \partial_\alpha^- f(x_0)$, $a \geq 1$, and $x \in X$ be such that $f(x) < f(x_0)$. We have

$$f(x_0) > f(x) \geq f(x_0) - (\omega(x_0 - x))^\alpha;$$

hence $\omega(x_0 - x) > 0$ and therefore

$$f(x_0) - (a\omega(x_0 - x))^\alpha \leq f(x_0) - (\omega(x_0 - x))^\alpha \leq f(x).$$

The proof of (3) follows immediately from the definition of $\partial_\alpha^- f(x_0)$. □

The following result is easy to prove.

LEMMA 2.3. *Let $g : \mathbb{R} \longrightarrow \overline{\mathbb{R}}$. If g is nondecreasing, then, for any $x_0 \in \mathbb{R}$ that is not a global minimum of g, we have $\partial_\alpha^- g(x_0) \subset \mathbb{R}^+ \setminus \{0\}$. If g is α-l.s.d., the converse also holds.*

PROPOSITION 2.4. *Let $f : X \longrightarrow \mathbb{R}$, $g : \mathbb{R} \longrightarrow \overline{\mathbb{R}}$ be a nondecreasing function on $f(X)$ and $\alpha, \beta \in (0, 1]$. Then*

$$\partial_{\alpha\beta}^-(g \circ f)(x_0) \supset (\partial_\beta^- g(f(x_0)))^{1/\alpha} \partial_\alpha^- f(x_0)$$

for any $x_0 \in X$ such that $f(x_0)$ is not a global minimum of g.

Proof. Let $\lambda^* \in \partial_\beta^- g(f(x_0))$, $\omega \in \partial_\alpha^- f(x_0)$, and $x \in K$ be such that $(g \circ f)(x) < (g \circ f)(x_0)$. Then, we have

$$g(f(x)) \geq g(f(x_0)) - [\lambda^*(f(x_0) - f(x))]^\beta,$$

where $\lambda^* > 0$ by the preceding lemma. On the other hand, since g is nondecreasing, we deduce $f(x) < f(x_0)$ and then

$$f(x) \geq f(x_0) - (\omega(x_0 - x))^\alpha.$$

Hence,

$$\begin{aligned}(g \circ f)(x) &\geq (g \circ f)(x_0) - [\lambda^*(f(x_0) - f(x))]^\beta \\ &\geq (g \circ f)(x_0) - [(\lambda^*)^{1/\alpha}\omega(x_0 - x)]^{\alpha\beta}.\end{aligned}$$

Therefore, $(\lambda^*)^{1/\alpha}\omega \in \partial_{\alpha\beta}^-(g \circ f)(x_0)$, which proves the statement. □

As a consequence of the preceding theorem, we have that if one of the functions in the statement is l.s.d. and the other one is α-l.s.d., the composite function is also α-l.s.d.

The composition of an α-l.s.d. function with a continuous linear one is also α-l.s.d., as we can see in the following result.

PROPOSITION 2.5. *Let $f : X \longrightarrow \overline{\mathbb{R}}$, Y be a locally convex space and let $T : Y \longrightarrow X$ be a continuous linear operator. Then, for every $y_0 \in Y$, we have*

$$\partial_\alpha^-(f \circ T)(y_0) \supset T^*(\partial_\alpha^- f(T(y_0))),$$

where T^* denotes the dual operator of T (in particular, if f is α-l.s.d. at $T(y_0)$, so is $f \circ T$ at y_0). If T is a topological isomorphism, equality holds.

Proof. Let $\omega \in \partial_\alpha^- f(T(y_0))$ and let $y \in Y$ be such that $(f \circ T)(y) < (f \circ T)(y_0)$. By the definitions of $\partial_\alpha^- f(T(y_0))$ and T^*, we have

$$\begin{aligned}(f \circ T)(y) = f(T(y)) &\geq f(T(y_0)) - [\omega(T(y_0) - T(y))]^\alpha \\ &= (f \circ T)(y_0) - [T^*(\omega)(y_0 - y)]^\alpha,\end{aligned}$$

which proves that $T^*(\omega) \in \partial_\alpha^-(f \circ T)(y_0)$. When T is a topological isomorphism, we get

$$\partial_\alpha^-((f \circ T) \circ T^{-1})(T(y_0)) \supset (T^*)^{-1}(\partial_\alpha^-(f \circ T)(y_0)),$$

which proves the equality. □

The α-lower subdifferential is related to two notions of generalized subdifferential which play some role in quasi-convex analysis: the quasi subdifferential of Greenberg and Pierskalla [5, p. 441] and the tangential of Crouzeix [2, p. 42], defined for $f : X \longrightarrow \overline{\mathbb{R}}$ and $x_0 \in X$ by

$$\partial^* f(x_0) = \{\omega \in X^* \mid \omega(x) \geq \omega(x_0) \Rightarrow f(x) \geq f(x_0)\}$$

and

$$Tf(x_0) = \left\{\omega \in X^* \mid \sup_{x \in X}\{\omega(x - x_0) \mid f(x) \leq \lambda\} < 0 \ \ \forall \lambda < f(x_0)\right\},$$

respectively. Crouzeix [2, Prop. 12, p. 42] proved that

$$Tf(x_0) \subset \partial^* f(x_0).$$

The following inclusion involving the α-lower subdifferential is also satisfied.

PROPOSITION 2.6. *Let $f : X \longrightarrow \overline{\mathbb{R}}$ and $x_0 \in X$ be such that $f(x_0) \in \mathbb{R}$. Then*

$$\partial_\alpha^- f(x_0) \subset Tf(x_0).$$

Proof. Let $\omega \in \partial_\alpha^- f(x_0), \lambda < f(x_0)$, and $x \in X$ be such that $f(x) \leq \lambda$. We have $f(x) < f(x_0)$ and, thus,

$$f(x) \geq f(x_0) - (\omega(x_0 - x))^\alpha.$$

We deduce

$$(f(x_0) - \lambda)^{1/\alpha} \leq (f(x_0) - f(x))^{1/\alpha} \leq \omega(x_0 - x).$$

Thus,

$$\sup_{x \in X}\{\omega(x - x_0) | f(x) \leq \lambda\} \leq -(f(x_0) - \lambda)^{1/\alpha} < 0,$$

which concludes the proof. □

In general, equality in the preceding proposition is not satisfied; unlike $\partial_\alpha^- f(x_0)$, $Tf(x_0)$ is necessarily a cone. In fact, we will prove (see Proposition 5.12 below) that under suitable conditions $Tf(x_0)$ coincides with the cone generated by $\partial_\alpha^- f(x_0)$.

If $f : X \longrightarrow \overline{\mathbb{R}}$ is quasi-convex and Gâteaux-differentiable at x_0 and the Gâteaux-differential $\nabla f(x_0)$ does not vanish, then, since $\partial^* f(x_0) \subset \{k \nabla f(x_0) \mid k > 0\}$ (see [2, Prop. 20, p. 53]), evidently

$$\partial_\alpha^- f(x_0) \subset \{k \nabla f(x_0) \mid k > 0\}.$$

When $\alpha = 1$, the second member in this inclusion can be replaced by $\{k\nabla f(x_0) | k \geq 1\}$ (see [7, Cor. 4.16, p. 217]). However, for $\alpha < 1$, there is no $k_0 > 0$ such that

$$\partial_\alpha^- f(x_0) \subset \{k\nabla f(x_0) \,|\, k \geq k_0\}$$

for any quasi-convex function f and any x_0 at which f is Gâteaux-differentiable with nonzero Gâteaux-differential. Indeed, take $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = \max\{|x|^\alpha, 1\}$ and let $x_0 > 1$. It is easy to prove that f is quasi-convex and differentiable at x_0, with $f'(x_0) = \alpha x_0^{\alpha-1}$. On the other hand, we have

$$\partial_\alpha^- f(x_0) = \left[\frac{(x_0^\alpha - 1)^{1/\alpha}}{x_0 - 1}, +\infty\right).$$

The nonexistence of $k_0 > 0$ with the required property can be deduced from the fact that

$$\lim_{x_0 \to 1^+} \frac{(x_0^\alpha - 1)^{1/\alpha}}{(x_0 - 1)f'(x_0)} = \lim_{x_0 \to 1^+} \frac{(x_0^\alpha - 1)^{1/\alpha}}{(x_0 - 1)\alpha x_0^{\alpha-1}} = 0.$$

3. Definition and properties of α-boundedly lower subdifferentiable functions. In this section, X will be a normed space, whose norm will be denoted by $\| \cdot \|$. In X^* we will consider, as usual, the norm $\| \cdot \|^*$ induced by $\| \cdot \|$, that is, $\| \omega \|^* = \sup\{\omega(x)| \, \| x \| \leq 1\}$ $(\omega \in X^*)$. More generally, if $(Y, \| \cdot \|)$ is another normed space with dual $(Y^*, \| \cdot \|^*)$ and $T : Y \longrightarrow X$ is a continuous linear operator, we will write $\| T \| = \sup\{\| T(y) \| \,|\, \| y \| \leq 1\}$. The following formula of Ascoli for the distance from a point to a closed hyperplane [13, Lem. 1.2, p. 24] will be used several times:

$$\inf\{\| x - y \| \,|\, \omega(y) = \omega(x_0)\} = \frac{|\omega(x - x_0)|}{\| \omega \|^*}. \tag{3.1}$$

Let $\alpha \in (0, 1]$. One says that $f : K \subset X \longrightarrow \mathbb{R}$ is *α-Hölder with constant N* if

$$f(x_1) - f(x_2) \leq N \| x_1 - x_2 \|^\alpha \quad \text{for any } x_1, x_2 \in K.$$

DEFINITION 3.1. Let $f : K \subset X \longrightarrow \overline{\mathbb{R}}$. We say that f is *α-b.l.s.d.* if there is $N > 0$ such that for any $x \in K$ there exists $\omega \in \partial_\alpha^- f(x)$ with $\| \omega \|^* \leq N$.

Constant N is called an *α-b.l.s.d. bound of f.*

α-Hölder functions are related to α-b.l.s.d. functions, as we can see in the following results.

THEOREM 3.1. *Every function $f : K \subset X \longrightarrow \mathbb{R}$ α-b.l.s.d. with α-b.l.s.d. bound N is α-Hölder with constant N^α.*

Proof. Let x, x_0 be such that $f(x) < f(x_0)$, and take $\omega \in \partial_\alpha^- f(x_0)$ with $\| \omega \|^* \leq N$; then

$$0 < f(x_0) - f(x) \leq (\omega(x_0 - x))^\alpha \leq (\| \omega \|^*)^\alpha \| x_0 - x \|^\alpha \leq N^\alpha \| x_0 - x \|^\alpha,$$

which proves the statement. □

THEOREM 3.2. *Let $f : X \longrightarrow \mathbb{R}$ be quasi-convex and α-Hölder with constant N. Then f is α-b.l.s.d. with α-b.l.s.d. bound $N^{1/\alpha}$.*

Proof. Let $x_0 \in X$. Since $x_0 \notin \dot{S}_{f(x_0)}(f)$ and this set is nonempty, open, and convex, by the Hahn–Banach theorem [4, p. 5] there exists $\omega \in X^*$ such that $\| \omega \|^* =$

1 and $\omega(x - x_0) < 0$ for every $x \in \dot{S}_{f(x_0)}(f)$. Let $x \in \dot{S}_{f(x_0)}(f)$. For any $\epsilon > 0$, there exists $x_\epsilon \in X$ such that

$$\| x - x_\epsilon \| \leq \inf\{\| x - y \| \mid \omega(y) = \omega(x_0)\} + \epsilon \quad \text{and} \quad \omega(x_\epsilon) = \omega(x_0).$$

Using (3.1), we can write the preceding inequality as $\| x - x_\epsilon \| \leq \omega(x_0 - x) + \epsilon$. Since $\omega(x_\epsilon - x_0) = 0$, we have $x_\epsilon \notin \dot{S}_{f(x_0)}(f)$ and, therefore,

$$\begin{aligned} 0 > f(x) - f(x_0) &\geq f(x) - f(x_\epsilon) \geq -N\| x - x_\epsilon \|^\alpha \\ &\geq -N(\omega(x_0 - x) + \epsilon)^\alpha = -[N^{1/\alpha}(\omega(x_0 - x) + \epsilon)]^\alpha. \end{aligned}$$

Since this is true for any $\epsilon > 0$, we obtain

$$f(x) - f(x_0) \geq -[N^{1/\alpha}(\omega(x_0 - x))]^\alpha,$$

which proves that $N^{1/\alpha}\omega \in \partial_\alpha^- f(x_0)$. As $\| N^{1/\alpha}\omega \|^* = N^{1/\alpha}$, we have completed the proof. □

COROLLARY 3.3. *Let $f : X \longrightarrow \mathbb{R}$. Then f is α-b.l.s.d. with α-b.l.s.d. bound N if and only if it is quasi-convex and α-Hölder with constant N^α.*

As a direct consequence of Proposition 2.4, we have the following.

PROPOSITION 3.4. *Let $f : K \subset X \longrightarrow \mathbb{R}$, $g : \mathbb{R} \longrightarrow \overline{\mathbb{R}}$ be a nondecreasing function on $f(K)$ and $\alpha, \beta \in (0, 1]$. If f and g are α-b.l.s.d. and β-b.l.s.d., respectively, then $g \circ f$ is $\alpha\beta$-b.l.s.d.*

The composition of an α-b.l.s.d. function with a continuous linear operator is also α-b.l.s.d., as the following result says.

PROPOSITION 3.5. *Let $f : X \longrightarrow \overline{\mathbb{R}}$, Y be a normed space and $T : Y \longrightarrow X$ be linear and continuous. If f is α-b.l.s.d. with α-b.l.s.d. bound N, then $f \circ T$ is α-b.l.s.d. with α-b.l.s.d. bound $N\| T \|$.*

Proof. By the demonstration of Proposition 2.5, we know that, for all $y_0 \in Y$ and $\omega \in \partial_\alpha^- T(y_0)$, one has $T^*(\omega) \in \partial_\alpha^-(f \circ T)(y_0)$. On the other hand, if $\| \omega \|^* \leq N$, then

$$\| T^*(\omega) \|^* = \| \omega \circ T \|^* \leq \| \omega \|^* \cdot \| T \| \leq N\| T \|. \qquad \square$$

4. Conjugation theory for α-lower subdifferentiable functions. We will apply the generalized conjugation theory of Moreau [9], described in §1, to the case when $C = X, D = X^* \times \mathbb{R}$, and $h_\alpha : C \times D \longrightarrow \mathbb{R}$, with $\alpha \in (0, 1]$, is the coupling function defined by

$$h_\alpha(x, (\omega, k)) = \min\{-(k - \omega(x))^\alpha, 0\} + k.$$

The h_α-conjugates of $f : X \longrightarrow \overline{\mathbb{R}}$ and $g : X^* \times \mathbb{R} \longrightarrow \overline{\mathbb{R}}$ are $f^{h_\alpha} : X^* \times \mathbb{R} \longrightarrow \overline{\mathbb{R}}$ and $g^{h_\alpha} : X \longrightarrow \overline{\mathbb{R}}$, respectively, defined by

$$f^{h_\alpha}(\omega, k) = \sup_{x \in X}\{\min\{-(k - \omega(x))^\alpha, 0\} + k - f(x)\}$$

and

$$g^{h_\alpha}(x) = \sup_{\omega \in X^*, k \in \mathbb{R}}\{\min\{-(k - \omega(x))^\alpha, 0\} + k - g(\omega, k)\}.$$

The elementary functions on X with respect to h_α are of the form

$$\min\{-(k - \omega)^\alpha, 0\} + \mu \quad \text{with } \omega \in X^*, \quad k \in \mathbb{R} \quad \text{and} \quad \mu \in \overline{\mathbb{R}}.$$

We will denote the set of their suprema as $\Delta_\alpha(X)$. Since the elementary functions are quasi-affine and continuous, the functions belonging to $\Delta_\alpha(X)$ are quasi-convex and l.s.c.

In some proofs, we will use the following elementary result.

LEMMA 4.1. *Let $\alpha \in (0,1]$ and $a, b \in \mathbb{R}$ be such that $a, b \geq 0$. Then $(a+b)^\alpha \leq a^\alpha + b^\alpha$.*

A first characterization of the functions belonging to $\Delta_\alpha(X)$ appears in the following lemma.

LEMMA 4.2. *Let $f : X \longrightarrow \overline{\mathbb{R}}$. Then the following results are equivalent:*

(1) *$f \in \Delta_\alpha(X)$;*

(2) *for every $(x_0, \lambda) \in (X \times \mathbb{R}) \setminus \operatorname{epi} f$ there exists $\omega \in X^*$ such that $\lambda - (\omega(x_0 - x))^\alpha \leq f(x)$ for all $x \in \dot{S}_\lambda(f)$.*

Proof. Let us see first that (1) $\Longrightarrow$ (2). Let $(x_0, \lambda) \in (X \times \mathbb{R}) \setminus \operatorname{epi} f$. Then $\lambda < f(x_0)$ and, by definition of $\Delta_\alpha(X)$, there exists an elementary minorant of f

$$\min\{-(k-\omega)^\alpha, 0\} + \mu$$

such that

$$\min\{-(k-\omega(x_0))^\alpha, 0\} + \mu \geq \lambda. \tag{4.1}$$

Let $x \in \dot{S}_\lambda(f)$; hence, we have

$$\min\{-(k-\omega(x))^\alpha, 0\} + \mu \leq f(x) < \lambda \leq \min\{-(k-\omega(x_0))^\alpha, 0\} + \mu \leq \mu$$

and, therefore,

$$k > \omega(x).$$

We will distinguish two cases: $k \leq \omega(x_0)$ and $k > \omega(x_0)$.

If $k \leq \omega(x_0)$, by (4.1) we have

$$\lambda - (k-\omega(x))^\alpha \leq -(k-\omega(x))^\alpha + \mu \leq f(x),$$

and, since $\omega(x) < k \leq \omega(x_0)$, we can write

$$f(x) \geq \lambda - (k-\omega(x))^\alpha \geq \lambda - (\omega(x_0 - x))^\alpha.$$

Suppose now that $k > \omega(x_0)$. From (4.1) we deduce

$$-(k-\omega(x_0))^\alpha + \mu \geq \lambda; \tag{4.2}$$

hence, we have

$$\begin{aligned} -(k-\omega(x))^\alpha + \mu = \min\{-(k-\omega(x))^\alpha, 0\} + \mu \leq f(x) \\ < \lambda \leq -(k-\omega(x_0))^\alpha + \mu \end{aligned}$$

and, therefore, $0 < \omega(x_0 - x)$. Now applying Lemma 4.1, for $a = k - \omega(x_0)$ and $b = \omega(x_0 - x)$, and using (4.2), we deduce

$$\begin{aligned} \lambda - (\omega(x_0 - x))^\alpha &\leq \lambda - (k-\omega(x))^\alpha + (k-\omega(x_0))^\alpha \\ &\leq -(k-\omega(x))^\alpha + \mu \\ &\leq f(x), \end{aligned}$$

which we wanted to see.

Let us now prove the implication (2) $\Longrightarrow$ (1). Given $x_0 \in X$ and $\lambda < f(x_0)$, that is, such that $(x_0, \lambda) \in (X \times \mathbb{R}) \backslash \operatorname{epi} f$, there exists $\omega \in X^*$ with $\lambda - (\omega(x_0 - x))^\alpha \leq f(x)$ for any $x \in \dot{S}_\lambda(f)$. Choose $k = \omega(x_0)$.

If $x \in \dot{S}_\lambda(f)$ we can write

$$\begin{aligned} \min\{-(k - \omega(x))^\alpha, 0\} + \lambda &\leq -(k - \omega(x))^\alpha + \lambda \\ &= -(\omega(x_0 - x))^\alpha + \lambda \leq f(x). \end{aligned}$$

If $x \in X \setminus \dot{S}_\lambda(f)$ we have

$$f(x) \geq \lambda \geq \min\{-(k - \omega(x))^\alpha, 0\} + \lambda.$$

We have seen that

$$\min\{-(k - \omega)^\alpha, 0\} + \lambda$$

is a minorant of f; moreover, it takes the value λ at x_0. As this happens for any $x_0 \in X$ and $\lambda < f(x_0)$, it follows that $f \in \Delta_\alpha(X)$. □

The following theorem characterizes those functions that are suprema of elementary functions. As mentioned at the beginning of this section, we recall that such functions are quasi-convex and l.s.c.

THEOREM 4.3. *Let $f : X \longrightarrow \overline{\mathbb{R}}$ be quasi-convex and l.s.c. Then the following statements are equivalent:*

(1) $f \in \Delta_\alpha(X)$;

(2) *for every $\lambda < \sup_{x \in X} f(x)$, there exist $\omega_\lambda \in X^*$ and $k_\lambda \in \mathbb{R}$ such that*

$$-(k_\lambda - \omega_\lambda)^\alpha + \lambda \quad \textit{minorizes } f \textit{ on } S_\lambda(f);$$

(3) *for every $\lambda < \sup_{x \in X} f(x)$, there exist $\omega_\lambda \in X^*$ and $k_\lambda \in \mathbb{R}$ such that*

$$-(k_\lambda - \omega_\lambda)^\alpha + \lambda \quad \textit{minorizes } f \textit{ on } \dot{S}_\lambda(f);$$

(4) *for every $\lambda < \sup_{x \in X} f(x)$, there exist $\omega_\lambda \in X^*$ and $k_\lambda \in \mathbb{R}$ such that*

$$\min\{-(k_\lambda - \omega_\lambda)^\alpha, 0\} + \lambda \quad \textit{minorizes } f;$$

(5) $\sup_{x \in X} f(x) = \sup_{x \in X} f^{h_\alpha h_\alpha}(x)$.

Proof. (1) $\Longrightarrow$ (5). This implication is obvious, since (1) is equivalent to the equality $f = f^{h_\alpha h_\alpha}$.

(5) $\Longrightarrow$ (4). Let $\lambda < \sup_{x \in X} f(x) = \sup_{x \in X} f^{h_\alpha h_\alpha}(x)$. There exists, therefore, $x_0 \in X$ such that $\lambda < f^{h_\alpha h_\alpha}(x_0)$. Since $f^{h_\alpha h_\alpha}$ is a supremum of elementary minorants of f, there exist $\omega_\lambda \in X^*$, $k_\lambda \in \mathbb{R}$, and $\mu_\lambda \in \overline{\mathbb{R}}$ satisfying

$$\min\{-(k_\lambda - \omega_\lambda)^\alpha, 0\} + \mu_\lambda \leq f$$

and

$$\min\{-(k_\lambda - \omega_\lambda(x_0))^\alpha, 0\} + \mu_\lambda > \lambda.$$

From this inequality we deduce $\mu_\lambda > \lambda$ and hence

$$\min\{-(k_\lambda - \omega_\lambda)^\alpha, 0\} + \lambda \leq f.$$

(4) $\Longleftrightarrow$ (3). This equivalence is evident, because a function g minorizes f on $\dot{S}_\lambda(f)$ if and only if $\min\{g, \lambda\} \leq f$.

(3) $\Longleftrightarrow$ (2). This equivalence is obvious.

(3) $\Longrightarrow$ (1). We will apply Lemma 4.2.

Let $(x_0, \lambda) \in (X \times \mathbb{R}) \setminus \operatorname{epi} f$; thus $x_0 \notin S_\lambda(f)$. Since f is quasi-convex and l.s.c., $S_\lambda(f)$ is a closed convex set. Therefore, since $x_0 \notin S_\lambda(f)$, by the Hahn–Banach theorem, there are $\psi \in X^*$ and $t \in \mathbb{R}$ such that

$$\psi(x) \leq t < \psi(x_0) \quad \text{for every } x \in S_\lambda(f).$$

On the other hand, by (3), there exist $\omega_\lambda \in X^*$ and $k_\lambda \in \mathbb{R}$ such that

$$-(k_\lambda - \omega_\lambda)^\alpha + \lambda \text{ minorizes } f \text{ on } \dot{S}_\lambda(f).$$

Define $\omega = a\psi + \omega_\lambda$ with

$$a \geq \max\left\{\frac{k_\lambda - \omega_\lambda(x_0)}{\psi(x_0) - t}, 0\right\}.$$

Let $x \in \dot{S}_\lambda(f)$. One can write

$$\begin{aligned}
\omega(x_0 - x) &= a\psi(x_0 - x) + \omega_\lambda(x_0 - x) \\
&= a\psi(x_0) - a\psi(x) - (k_\lambda - \omega_\lambda(x_0)) - \omega_\lambda(x) + k_\lambda \\
&\geq a\psi(x_0) - a\psi(x) - a(\psi(x_0) - t) - \omega_\lambda(x) + k_\lambda \\
&= a(t - \psi(x)) - \omega_\lambda(x) + k_\lambda \\
&\geq k_\lambda - \omega_\lambda(x).
\end{aligned}$$

Therefore,

$$\lambda - (\omega(x_0 - x))^\alpha \leq \lambda - (k_\lambda - \omega_\lambda(x))^\alpha \leq f(x)$$

for any $x \in \dot{S}_\lambda(f)$. Using Lemma 4.2, we conclude that $f \in \Delta_\alpha(X)$. □

The hypotheses on f have been used only to prove implication (3) $\Longrightarrow$ (1). If we substitute statement (4) by

(4′) for any $\lambda < \sup_{x \in X} f(x)$, there exist $\omega_\lambda \in X^* \setminus \{0\}$ and $k_\lambda \in \mathbb{R}$ such that

$$\min\{-(k_\lambda - \omega_\lambda)^\alpha, 0\} + \lambda \text{ minorizes } f,$$

then one can prove (4′) $\Longrightarrow$ (5) directly.

Let $\lambda < \sup_{x \in X} f(x)$. Since $f^{h_\alpha h_\alpha}$ is the supremum of elementary minorants of f, one has

$$\min\{-(k_\lambda - \omega_\lambda)^\alpha, 0\} + \lambda \leq f^{h_\alpha h_\alpha}.$$

Let $y \in X$ be a point that satisfies $\omega_\lambda(y) = k_\lambda$; then, taking the supremum, we get

$$\begin{aligned}
\sup_{x \in X} f^{h_\alpha h_\alpha}(x) &\geq \sup_{x \in X}\{\min\{-(k_\lambda - \omega_\lambda(x))^\alpha, 0\} + \lambda\} \\
&\geq \min\{-(k_\lambda - \omega_\lambda(y))^\alpha, 0\} + \lambda = \lambda.
\end{aligned}$$

Letting $\lambda \longrightarrow \sup_{x \in X} f(x)$, we obtain

$$\sup_{x \in X} f^{h_\alpha h_\alpha}(x) \geq \sup_{x \in X} f(x).$$

Since the other inequality holds for any function f, we have demonstrated (5).

It is evident that (4′) is equivalent to (3′) and (2′), which consist in (3) and (2), respectively, adding the condition $\omega_\lambda \neq 0$.

It is easy to see that any constant function is a supremum of elementary nonconstant functions. From this, by an obvious modification of the above proof of the implication (5) $\Longrightarrow$ (4), we deduce that (5) $\Longrightarrow$ (4′) which demonstrates that (5) is equivalent to (4′), (3′), and (2′). We say that a function which satisfies these conditions *has property* (H_α).

We define the "h_α-level of f" as

$$\lambda_{\alpha,f} = \sup\{\lambda \mid \exists \omega \in X^* \setminus \{0\}, k \in \mathbb{R} \text{ such that } \\ -(k-\omega(x))^\alpha + \lambda \leq f(x) \text{ for any } x \in \dot{S}_\lambda(f)\}.$$

The h_α-level is related to the second conjugate as follows.

LEMMA 4.4. *Let* $f : X \longrightarrow \overline{\mathbb{R}}$. *Then*

$$\lambda_{\alpha,f} = \sup_{x\in X} f^{h_\alpha h_\alpha}(x).$$

Proof. Let $\lambda < \lambda_{\alpha,f}$. There exist, therefore, $\omega_\lambda \in X^* \setminus \{0\}$ and $k_\lambda \in \mathbb{R}$ such that $-(k_\lambda - \omega_\lambda(x))^\alpha + \lambda \leq f(x)$ for any $x \in \dot{S}_\lambda(f)$. Hence $\min\{-(k_\lambda - \omega_\lambda)^\alpha, 0\} + \lambda \leq f$ and thus $\min\{-(k_\lambda - \omega_\lambda)^\alpha, 0\} + \lambda \leq f^{h_\alpha h_\alpha}$. We have

$$\begin{aligned}\sup_{x\in X} f^{h_\alpha h_\alpha}(x) &\geq \sup_{x\in X}\{\min\{-(k_\lambda - \omega_\lambda(x))^\alpha, 0\} + \lambda\} \\ &\geq \min\{-(k_\lambda - \omega_\lambda(y))^\alpha, 0\} + \lambda = \lambda,\end{aligned}$$

where $y \in X$ satisfies $\omega_\lambda(y) = k_\lambda$. Letting $\lambda \longrightarrow \lambda_{\alpha,f}$, we obtain

$$\lambda_{\alpha,f} \leq \sup_{x\in X} f^{h_\alpha h_\alpha}(x).$$

To see the other inequality, let $\lambda < \sup_{x\in X} f^{h_\alpha h_\alpha}(x)$. By Theorem 4.3, there exist $\omega_\lambda \in X^*$ and $k_\lambda \in \mathbb{R}$ such that

$$-(k_\lambda - \omega_\lambda(x))^\alpha + \lambda \leq f(x)$$

for any $x \in \dot{S}_\lambda(f)$. We can suppose $\omega_\lambda \neq 0$ (since every constant function can be expressed as a supremum of nonconstant elementary functions). Thus, $\lambda \leq \lambda_{\alpha,f}$ and, letting $\lambda \longrightarrow \sup_{x\in X} f^{h_\alpha h_\alpha}(x)$, we get

$$\lambda_{\alpha,f} \geq \sup_{x\in X} f^{h_\alpha h_\alpha}(x),$$

which we wanted to prove. □

THEOREM 4.5. *Let* $f : X \longrightarrow \overline{\mathbb{R}}$. *Then*

$$f^{h_\alpha h_\alpha} = \min\{f_{\bar{q}}, \lambda_{\alpha,f}\}.$$

Moreover, $f^{h_\alpha h_\alpha} = f_{\bar{q}}$ *if and only if* $f_{\bar{q}}$ *has property* (H_α).

Proof. We define $g = \min\{f_{\bar{q}}, \lambda_{\alpha,f}\}$. This function is quasi-convex and l.s.c., and minorizes f. Since $f^{h_\alpha h_\alpha} \leq f_{\bar{q}}$ and $f^{h_\alpha h_\alpha} \leq \sup_{x\in X} f^{h_\alpha h_\alpha}(x) = \lambda_{\alpha,f}$ (see the preceding lemma), we have $f^{h_\alpha h_\alpha} \leq g \leq \lambda_{\alpha,f}$ and, therefore, we deduce

$$f^{h_\alpha h_\alpha} \leq g^{h_\alpha h_\alpha} \leq g \leq \lambda_{\alpha,f}.$$

Taking supremum, we obtain

$$\sup_{x\in X} f^{h_\alpha h_\alpha}(x) \leq \sup_{x\in X} g^{h_\alpha h_\alpha}(x) \leq \sup_{x\in X} g(x) \leq \lambda_{\alpha,f}.$$

By Lemma 4.4 and Theorem 4.3, this implies $g \in \Delta_\alpha(X)$. Since g is a minorant of f, we have $g \leq f^{h_\alpha h_\alpha}$, which concludes the demonstration of the first assertion.

Let us now see the second one. If $f^{h_\alpha h_\alpha} = f_{\bar{q}}$, then

$$\sup_{x\in X} f_{\bar{q}}(x) = \sup_{x\in X} f^{h_\alpha h_\alpha}(x) = \sup_{x\in X} f^{h_\alpha h_\alpha h_\alpha h_\alpha}(x) = \sup_{x\in X} (f_{\bar{q}})^{h_\alpha h_\alpha}(x),$$

whence $f_{\bar{q}}$ has property (H_α). Conversely, if $f_{\bar{q}}$ has property (H_α), then, by Theorem 4.3, $f_{\bar{q}} \in \Delta_\alpha(X)$; thus, we have $f_{\bar{q}} = (f_{\bar{q}})^{h_\alpha h_\alpha} \leq f^{h_\alpha h_\alpha}$, and since $f^{h_\alpha h_\alpha}$ is l.s.c. and quasi-convex and minorizes f, one deduces $f^{h_\alpha h_\alpha} \leq f_{\bar{q}}$, from which we obtain $f^{h_\alpha h_\alpha} = f_{\bar{q}}$, and hence the equivalence in the statement. □

From Theorem 4.5 one easily deduces that, given $f : X \longrightarrow \overline{\mathbb{R}}$ and $x_0 \in X$, the equivalence

$$f^{h_\alpha h_\alpha}(x_0) = f(x_0) \Longleftrightarrow f_{\bar{q}}(x_0) = f(x_0) \leq \lambda_{\alpha,f}$$

holds.

We also observe that $(f_{\bar{q}})^{h_\alpha h_\alpha} \leq f^{h_\alpha h_\alpha} \leq f_{\bar{q}}$ for arbitrary $f : X \longrightarrow \overline{\mathbb{R}}$, from which we deduce

$$(f_{\bar{q}})^{h_\alpha h_\alpha} = f^{h_\alpha h_\alpha}.$$

Using this equality and Lemma 4.4, one obtains

$$\lambda_{\alpha,f_{\bar{q}}} = \sup_{x\in X} (f_{\bar{q}})^{h_\alpha h_\alpha}(x) = \sup_{x\in X} f^{h_\alpha h_\alpha}(x) = \lambda_{\alpha,f}.$$

As a consequence, the following implication holds:

$$f \text{ has property } (H_\alpha) \Longrightarrow f_{\bar{q}} \text{ has property } (H_\alpha).$$

The converse statement is not true as, e.g., the function $f : \mathbb{R} \longrightarrow \mathbb{R}$ defined by $f(x) = 0$ if $x \neq 0$ and $f(0) = 1$ shows.

The preceding considerations remain true when one replaces $f_{\bar{q}}$ by f_q or $\bar{f}$, since one also has $f^{h_\alpha h_\alpha} \leq f_q \leq f$ and $f^{h_\alpha h_\alpha} \leq \bar{f} \leq f$.

COROLLARY 4.6. *If $f : X \longrightarrow \overline{\mathbb{R}}$ is bounded below, then $f^{h_\alpha h_\alpha} = f_{\bar{q}}$.*

Proof. Since f is bounded below, so is $f_{\bar{q}}$. Thus, $f_{\bar{q}}$ satisfies condition (4) of Theorem 4.3 and, therefore, $f_{\bar{q}}$ has property (H_α). The conclusion follows from Theorem 4.5. □

Corollary 4.6 is equivalent to saying that every bounded-below l.s.c. quasi-convex function has property (H_α).

If $f : X \longrightarrow \overline{\mathbb{R}}$ is quasi-convex, we have that $f^{h_\alpha h_\alpha} = \min\{\bar{f}, \lambda_{\alpha,f}\}$ by Theorem 4.5, since, in this case, $f_{\bar{q}} = \bar{f}$ (see [3, Cor. 3, p. 112]).

Now we will give an expression for the second h_α-conjugate of any function.

PROPOSITION 4.7. *Let $f : X \longrightarrow \overline{\mathbb{R}}$. Then, for every $x_0 \in X$ one has*

$$f^{h_\alpha h_\alpha}(x_0) = \sup_{\omega\in X^*} \inf_{x\in X} \max\{(\omega(x_0 - x))^\alpha + f(x), f(x)\}.$$

Proof. By the definitions of h_α-conjugates, one has

$$\begin{aligned}
f^{h_\alpha h_\alpha}(x_0) &= \sup_{(\omega,k)\in D}\{\min\{-(k-\omega(x_0))^\alpha,0\}+k-f^{h_\alpha}(\omega,k)\}\\
&= \sup_{(\omega,k)\in D}\{\min\{-(k-\omega(x_0))^\alpha,0\}+k\\
&\qquad\qquad -\sup_{x\in X}\{\min\{-(k-\omega(x))^\alpha,0\}+k-f(x)\}\}\\
&= \sup_{(\omega,k)\in D}\inf_{x\in X}\{\min\{-(k-\omega(x_0))^\alpha,0\}\\
&\qquad\qquad -\min\{-(k-\omega(x))^\alpha,0\}+f(x)\}.
\end{aligned}$$

It only remains to see that, for all $x \in X$, the expression

$$\min\{-(k-\omega(x_0))^\alpha,0\}-\min\{-(k-\omega(x))^\alpha,0\}$$

takes its maximum, as a function of k, at $k=\omega(x_0)$, i.e., that

$$\begin{aligned}
&\min\{-(k-\omega(x_0))^\alpha,0\}-\min\{-(k-\omega(x))^\alpha,0\}\\
&\qquad \le \max\{(\omega(x_0-x))^\alpha,0\}.
\end{aligned} \tag{4.3}$$

If $k \le \omega(x)$, this inequality is true because its first member is less than or equal to 0. If $k \ge \omega(x)$ and $\omega(x_0)$ is not between these two quantities, (4.3) is satisfied obviously because t^α is a nondecreasing function. Finally, if $k \ge \omega(x_0) \ge \omega(x)$, inequality (4.3) immediately follows from Lemma 4.1. □

The formula in the preceding proposition gives an expression for the l.s.c. quasiconvex hull of a function f when, for example, f is bounded below (see Theorem 4.5 and Corollary 4.6).

The following results show the relation existing between α-lower subgradients and h_α-subgradients in the sense of Balder (see §1).

PROPOSITION 4.8. *Let $f : X \longrightarrow \overline{\mathbb{R}}$ and $x_0 \in X$ be such that $f(x_0) \in \mathbb{R}$. Then, for any $\omega \in X^*$, the following statements are equivalent:*

(1) *there exists $k \in \mathbb{R}$ such that $(\omega, k) \in \partial_{h_\alpha} f(x_0)$;*

(2) $(\omega, \omega(x_0)) \in \partial_{h_\alpha} f(x_0)$;

(3) $\omega \in \partial_\alpha^- f(x_0)$.

Proof. (1) $\Longrightarrow$ (3). Let $(\omega,k) \in \partial_{h_\alpha} f(x_0)$ and let $x \in X$ be such that $f(x) < f(x_0)$. We have

$$\begin{aligned}
0 &> f(x)-f(x_0)\\
&\ge \min\{-(k-\omega(x))^\alpha,0\}-\min\{-(k-\omega(x_0))^\alpha,0\}\\
&\ge \min\{-(k-\omega(x))^\alpha,0\};
\end{aligned} \tag{4.4}$$

thus

$$f(x)-f(x_0) \ge -(k-\omega(x))^\alpha. \tag{4.5}$$

We will distinguish two cases: $k \le \omega(x_0)$ and $k > \omega(x_0)$.

If $k \le \omega(x_0)$, then $(\omega(x_0-x))^\alpha \ge (k-\omega(x))^\alpha$. Combining this with (4.5), we get

$$f(x)-f(x_0) \ge -(\omega(x_0-x))^\alpha,$$

which was to be proved.

If, instead, $k > \omega(x_0)$, by (4.4) we have

$$\begin{aligned} f(x) - f(x_0) &\geq \min\{-(k - \omega(x))^\alpha, 0\} + (k - \omega(x_0))^\alpha \\ &= -(k - \omega(x))^\alpha + (k - \omega(x_0))^\alpha, \end{aligned}$$

the last equality being a consequence of the fact that $f(x) < f(x_0)$. For the same reason, we deduce

$$\omega(x_0 - x) > 0$$

and, by Lemma 4.1 (applied to $a = k - \omega(x_0)$ and $b = \omega(x_0 - x)$),

$$f(x) - f(x_0) \geq -(\omega(x_0 - x))^\alpha.$$

Thus, $\omega \in \partial_\alpha^- f(x_0)$.

(3) $\Longrightarrow$ (2). Let $x \in X$. If $f(x) < f(x_0)$, then, by $\omega \in \partial_\alpha^- f(x_0)$, we can write

$$\begin{aligned} f(x) &\geq f(x_0) - (\omega(x_0 - x))^\alpha \\ &\geq f(x_0) + \min\{-(\omega(x_0 - x))^\alpha, 0\}. \end{aligned}$$

If $f(x) \geq f(x_0)$, one has

$$f(x) \geq f(x_0) \geq f(x_0) + \min\{-(\omega(x_0 - x))^\alpha, 0\}.$$

Since $h_\alpha(x, (\omega, \omega(x_0))) - h_\alpha(x_0, (\omega, \omega(x_0))) = \min\{-(\omega(x_0 - x))^\alpha, 0\}$, we have seen that $(\omega, \omega(x_0)) \in \partial_{h_\alpha} f(x_0)$.

The implication (2) $\Longrightarrow$ (1) is obvious. □

COROLLARY 4.9. *Let $f : X \longrightarrow \overline{\mathbb{R}}$ and $x_0 \in X$ be such that $f(x_0) \in \mathbb{R}$. Then $\partial_\alpha^- f(x_0)$ is the projection of $\partial_{h_\alpha} f(x_0)$ onto X^*.*

COROLLARY 4.10. *Let $f : X \longrightarrow \overline{\mathbb{R}}$ and $x_0 \in X$ be such that $f(x_0) \in \mathbb{R}$. Then f is α-l.s.d. at x_0 if and only if f is h_α-subdifferentiable at x_0.*

COROLLARY 4.11. *Let $f : X \longrightarrow \overline{\mathbb{R}}$ and $x_0 \in X$ be such that $f(x_0) \in \mathbb{R}$. Then*

(1) $\partial_\alpha^- f(x_0) \neq \emptyset \Longrightarrow f(x_0) = f^{h_\alpha h_\alpha}(x_0)$;

(2) $f(x_0) = f^{h_\alpha h_\alpha}(x_0) \Longrightarrow \partial_\alpha^- f(x_0) = \partial_\alpha^- f^{h_\alpha h_\alpha}(x_0)$.

Proof. The proof is an immediate consequence of Corollary 4.9 and properties (1.2) and (1.3). □

According to the first statement in the preceding corollary, any α-l.s.d. function belongs to $\Delta_\alpha(X)$. Unfortunately, there are functions in $\Delta_\alpha(X)$ which, however, fail to be α-l.s.d. at some points. Indeed, $f : \mathbb{R} \longrightarrow \mathbb{R}$ given by

$$f(x) = \begin{cases} -(1 - x^2)^{\alpha/2} & \text{if } x \in [-1, 1] \ , \\ 0 & \text{otherwise,} \end{cases}$$

is not α-l.s.d. at 1 and at -1, and nevertheless it belongs to $\Delta_\alpha(X)$, because it is quasi-convex, continuous, and bounded below (notice that, by Corollary 4.6, $\Delta_\alpha(X)$ contains all bounded-below l.s.c. quasi-convex functions).

COROLLARY 4.12. *Let $f : X \longrightarrow \overline{\mathbb{R}}$ and $x_0 \in X$ be such that $f(x_0) \in \mathbb{R}$, and let $\omega \in X^*$. Then*

$$\omega \in \partial_\alpha^- f(x_0) \Longleftrightarrow f(x_0) + f^{h_\alpha}(\omega, \omega(x_0)) = \omega(x_0).$$

Proof. By Proposition 4.8, we know that $\omega \in \partial_\alpha^- f(x_0)$ if and only if $(\omega, \omega(x_0)) \in \partial_{h_\alpha} f(x_0)$. Applying property (1.1), we see that the latter relation is equivalent to

$$f(x_0) + f^{h_\alpha}(\omega, \omega(x_0)) = h_\alpha(x_0, (\omega, \omega(x_0))).$$

To obtain the desired equivalence, it only remains to observe that

$$h_\alpha(x_0, (\omega, \omega(x_0))) = \omega(x_0). \qquad \square$$

We close this section by showing the monotonic dependence on α of $\Delta_\alpha(X)$ and hence of $f^{h_\alpha h_\alpha}$ for every $f : X \longrightarrow \overline{\mathbb{R}}$.

PROPOSITION 4.13. *Let $\alpha, \beta \in (0,1]$ be such that $\alpha < \beta$. Then*

$$\Delta_\alpha(X) \subsetneq \Delta_\beta(X),$$

and hence

$$f^{h_\alpha h_\alpha} \leq f^{h_\beta h_\beta}$$

for any $f : X \longrightarrow \overline{\mathbb{R}}$.

Proof. Clearly, to prove the inclusion $\Delta_\alpha(X) \subset \Delta_\beta(X)$, it suffices to demonstrate that any elementary function $\min\{-(k-\omega)^\alpha, 0\} + \mu$ in $\Delta_\alpha(X)$, with $k, \mu \in \mathbb{R}$ and $\omega \in X^*$, belongs to $\Delta_\beta(X)$. Since such an elementary function is quasi-convex and continuous, in view of Theorem 4.3 (implication (4) $\Longrightarrow$ (1)), it is enough to show that for every $\lambda < \mu$ there exist $\omega_\lambda \in X^*$ and $k_\lambda \in \mathbb{R}$ such that

$$-(k_\lambda - \omega_\lambda)^\beta + \lambda \leq -(k-\omega)^\alpha + \mu.$$

But it is easy to see that, choosing $k_\lambda = N_\lambda^{1/\beta} k$ and $\omega_\lambda = N_\lambda^{1/\beta}\omega$, N_λ being any upper bound of the function (defined on the set of strictly positive real numbers) $t \longrightarrow t^{\alpha-\beta} - (\mu - \lambda)t^{-\beta}$, the required inequality is satisfied.

Since $f^{h_\alpha h_\alpha}$ and $f^{h_\beta h_\beta}$ are the greatest minorants of f in $\Delta_\alpha(X)$ and $\Delta_\beta(X)$, respectively, the inequality $f^{h_\alpha h_\alpha} \leq f^{h_\beta h_\beta}$ immediately follows from the inclusion we have just proved.

To show that $\Delta_\alpha(X) \neq \Delta_\beta(X)$, take any $\omega \in X^* \backslash \{0\}$ and consider the elementary function, with respect to h_β, $\min\{-(-\omega)^\beta, 0\}$. Of course, it belongs to $\Delta_\beta(X)$. Should this function belong to $\Delta_\alpha(X)$, one could find, according to Theorem 4.3 (implication (1) $\Longrightarrow$ (4)), $\omega_1 \in X^*$ and $k_1 \in \mathbb{R}$ such that

$$\min\{-(k_1 - \omega_1)^\alpha, 0\} - 1 \leq \min\{-(-\omega)^\beta, 0\}.$$

Then, taking $\bar{x} \in X$ with $\omega(\bar{x}) = -1$, we should obtain

$$\min\{-(k_1 - \eta\omega_1(\bar{x}))^\alpha, 0\} - 1 \leq \min\{-\eta^\beta, 0\}$$

for all $\eta \in \mathbb{R}$. But, for large enough $\eta > 0$, this inequality does not hold. $\square$

In fact, the above proof shows that

$$\bigcup_{\alpha<\beta} \Delta_\alpha(X) \subsetneq \Delta_\beta(X)$$

for all $\beta \in (0,1]$. On the other hand, by a slight modification of the argument employed to show that the function $f = \min\{-(-\omega)^\beta, 0\}$ belongs to $\Delta_\beta(X) \setminus \Delta_\alpha(X)$, for all $\alpha \in (0,\beta)$, one can see that f admits no finite elementary minorant with respect to h_α, whence $f^{h_\alpha h_\alpha} \equiv -\infty$, while $f^{h_\beta h_\beta} = f$.

5. α-Hölder quasi-convex functions and their suprema. In this section, X will be a normed space as in §3. We will use the notation of §3; $\mathcal{B}^*(0;N)$ will denote the closed ball in X^* with radius $N > 0$ and center at the origin.

Let $D_N = \mathcal{B}^*(0;N) \times \mathbb{R}, \alpha \in (0,1]$, and let $h_{\alpha,N} : X \times D_N \longrightarrow \mathbb{R}$ be the coupling function defined by

$$h_{\alpha,N}(x,(\omega,k)) = \min\{-(k-\omega(x))^\alpha, 0\} + k.$$

Therefore, $h_{\alpha,N} = h_{\alpha|X\times D_N}$, where h_α is the coupling function of the preceding section. The $h_{\alpha,N}$-conjugates of $f : X \longrightarrow \overline{\mathbb{R}}$ and $g : D_N \longrightarrow \overline{\mathbb{R}}$ are $f^{h_{\alpha,N}} : D_N \longrightarrow \overline{\mathbb{R}}$ and $g^{h_{\alpha,N}} : X \longrightarrow \overline{\mathbb{R}}$ defined by means of

$$f^{h_{\alpha,N}}(\omega,k) = \sup_{x\in X}\{\min\{-(k-\omega(x))^\alpha, 0\} + k - f(x)\}$$

and

$$g^{h_{\alpha,N}}(x) = \sup_{\omega\in\mathcal{B}^*(0;N),k\in\mathbb{R}}\{\min\{-(k-\omega(x))^\alpha, 0\} + k - g(\omega,k)\}.$$

The elementary functions on X with respect to $h_{\alpha,N}$ are of the form

$$\min\{-(k-\omega)^\alpha, 0\} + \mu \quad \text{with } \omega \in \mathcal{B}^*(0;N), \quad k \in \mathbb{R}, \quad \text{and} \quad \mu \in \overline{\mathbb{R}}.$$

It is easy to see that they are quasi-affine. We will denote the set of their suprema by $B_N^\alpha(X)$. As an immediate consequence of the next proposition, taking into account that the elementary functions are quasi-affine, it follows that $B_N^\alpha(X) \setminus \{\pm\infty\}$ is contained in the set of quasi-convex functions that are α-Hölder with constant N^α. We shall prove below (see Theorem 5.4) that the converse inclusion is also true.

PROPOSITION 5.1. *The functions* $f = \min\{-(k-\omega)^\alpha, 0\} + \mu$, *with* $\omega \in X^* \setminus \{0\}$, $k \in \mathbb{R}$, *and* $\mu \in \mathbb{R}$, *are α-Hölder with constant* $(\| \omega \|^*)^\alpha$; *this is the smallest α-Hölder constant for* f.

Proof. Let $x_0, x \in X$. We have, according to (4.3),

$$\begin{aligned} f(x_0) - f(x) &= \min\{-(k-\omega(x_0))^\alpha, 0\} - \min\{-(k-\omega(x))^\alpha, 0\} \\ &\le \max\{(\omega(x_0 - x))^\alpha, 0\} \le (\| \omega \|^*)^\alpha \| x_0 - x \|^\alpha. \end{aligned}$$

Let us suppose that f is α-Hölder with constant $N < (\| \omega \|^*)^\alpha$. Take x_0, x, and ϵ such that $\omega(x_0) = k$, $\omega(x) = k-1$, and $0 < \epsilon < N^{-1/\alpha} - (1/\| \omega \|^*)$. Let $x_\epsilon \in X$ be such that

$$\| x - x_\epsilon \| \le \inf\{\| x - y \| \mid \omega(y) = \omega(x_0)\} + \epsilon \quad \text{and} \quad \omega(x_\epsilon) = \omega(x_0).$$

Using (3.1), we can write the preceding inequality as

$$\| x - x_\epsilon \| \le \frac{1}{\| \omega \|^*}\omega(x_0 - x) + \epsilon = \frac{1}{\| \omega \|^*} + \epsilon.$$

Then we have

$$1 = |f(x) - f(x_\epsilon)| \le N\| x - x_\epsilon \|^\alpha \le N\left(\frac{1}{\| \omega \|^*} + \epsilon\right)^\alpha < 1,$$

which is absurd. □

The following proposition states the dependence of the sets $B_N^\alpha(X)$ with respect to N and their relation with $\Delta_\alpha(X)$ (see §4). By $s(F)$, F being a family of extended real valued functions, we denote the set of pointwise suprema of subfamilies of F.

PROPOSITION 5.2.

$$(1) \qquad \bigcup_{0<N'<N} B_{N'}^\alpha(X) \subsetneq s\left(\bigcup_{0<N'<N} B_{N'}^\alpha(X)\right) \subsetneq B_N^\alpha(X), \qquad (N>0),$$

$$(2) \qquad B_N^\alpha(X) = \bigcap_{N'>N} B_{N'}^\alpha(X), \qquad (N>0),$$

$$(3) \qquad \bigcup_{N>0} B_N^\alpha(X) \subsetneq \Delta_\alpha(X) = s\left(\bigcup_{N>0} B_N^\alpha(X)\right).$$

Proof. (1) The first inclusion is obvious. To see that it is a strict inclusion, consider the function

$$g(x) = \max\{\min\{-(-\omega(x))^\alpha, 0\}, -1\},$$

with $\| \omega \|^* = N$. For each $\lambda \in (0,1)$, we define

$$g_\lambda(x) = \max\{\min\{-(-\lambda[\omega(x)+1]+1)^\alpha, 0\}, -1\}.$$

It is easy to show that g_λ depends nondecreasingly on λ and $\sup_{0<\lambda<1} g_\lambda = g$. Since $g_\lambda \in B_{\lambda N}^\alpha(X)$, we have

$$g \in s\left(\bigcup_{0<N'<N} B_{N'}^\alpha(X)\right).$$

However, $g \notin \bigcup_{0<N'<N} B_{N'}^\alpha(X)$, since the same reasoning showing that no $N' < (\| \omega \|^*)^\alpha$ is an α-Hölder constant for the function f of Proposition 5.1 applies here (notice that $g = \max\{f, -1\}$, when f corresponds to $k = \mu = 0$, and that g and f coincide at x_0 and x, the points appearing in the proof of Proposition 5.1).

The second inclusion is immediate, since $B_N^\alpha(X)$ is closed under supremum. To see that it is strict take f as above. Then $f \in B_N^\alpha(X)$, obviously, but, using Proposition 5.1, one can prove that f admits no elementary α-Hölder minorant with constant $N' < N^\alpha$.

(2) Inclusion $\subset$ is immediate, while $\supset$ will be a consequence of equivalence (1) $\Longleftrightarrow$ (2) in Theorem 5.4.

The inclusions in (3) follow easily from the definition of the elementary functions with respect to h_α and $h_{\alpha,N}$. To prove the strictness of the first one, let $f : X \longrightarrow \mathbb{R}$ be the function defined by $f(x) = \| x \|^\beta$, with $\beta \neq \alpha$. Since f is quasi-convex, continuous, and bounded below, it belongs to $\Delta_\alpha(X)$ (by Corollary 4.6). On the other hand, this function is not α-Hölder and, therefore, $f \notin \bigcup_{N>0} B_N^\alpha(X)$ (see our comments preceding Proposition 5.1). □

In spite of Proposition 4.13, there is no monotonic dependence on α of the sets $B_N^\alpha(X)$ and, therefore, of the second conjugates $f^{h_{\alpha,N}h_{\alpha,N}}$. Indeed, take $\omega \in X^*$ with $\| \omega \|^* = N$ and let $f = \min\{-(-\omega)^\beta, 0\}$, with $\beta \in (0,1]$. This is an elementary function in $B_N^\beta(X)$ which does not belong to $\Delta_\alpha(X)$, and hence to $B_N^\alpha(X)$, for any $\alpha \in (0,\beta)$ (see the proof of Proposition 4.13). Let $\alpha \in (\beta, 1]$ and take any $\bar{x} \in X$ with $\omega(\bar{x}) = -1$. If we had $f \in B_N^\alpha(X)$, then, since, as we observed before Proposition 5.1, f is α-Hölder with constant N^α for all $\lambda > 0$, we should obtain

$$\lambda^\beta = |f(\lambda\bar{x}) - f(0)| \leq N^\alpha \| \lambda\bar{x} \|^\alpha = N^\alpha \| \bar{x} \|^\alpha \lambda^\alpha,$$

whence $\lambda^{\beta-\alpha} \leq N^\alpha \| \bar{x} \|^\alpha$. But this inequality does not hold when λ is too small. Thus, we have shown that the family $\{B_N^\alpha(X)\}_{\alpha\in(0,1]}$ consists of pairwise incomparable (with respect to inclusion) sets. Even more, for any $\beta \in (0,1]$, one has

$$B_N^\beta(X) \not\subset \bigcup_{\alpha\in(0,1]\setminus\{\beta\}} B_N^\alpha(X).$$

We recall (see §1) that a function $f : X \longrightarrow \overline{\mathbb{R}}$ is *$h_{\alpha,N}$-subdifferentiable at* $x_0 \in X$ if $f(x_0) \in \mathbb{R}$ and there exists $(\omega, k) \in D_N$ such that

$$f(x) - f(x_0) \geq \min\{-(k-\omega(x))^\alpha, 0\} - \min\{-(k-\omega(x_0))^\alpha, 0\}$$

for every $x \in X$; (ω, k) is then an *$h_{\alpha,N}$-subgradient of f at* x_0. The set of all $h_{\alpha,N}$-subgradients of f at x_0 is called the *$h_{\alpha,N}$-subdifferential of f at* x_0 and will be denoted by $\partial_{h_{\alpha,N}} f(x_0)$.

Since $D = \bigcup_{N>0} D_N$ and $h_{\alpha,N} = h_{\alpha|X\times D_N}$, the following relations hold.

PROPOSITION 5.3. *Let $f : X \longrightarrow \overline{\mathbb{R}}$ and let $x_0 \in X$. Then*

(1) $f^{h_{\alpha,N}} = f^{h_\alpha}{}_{|D_N}$;

(2) $f^{h_\alpha h_\alpha} = \sup_{N>0} f^{h_{\alpha,N} h_{\alpha,N}}$;

(3) $\partial_{h_{\alpha,N}} f(x_0) = \partial_{h_\alpha} f(x_0) \cap D_N$;

(4) $\partial_{h_\alpha} f(x_0) = \bigcup_{N>0} \partial_{h_{\alpha,N}} f(x_0)$.

The functions that belong to $B_N^\alpha(X)$ are characterized in the next theorem.

THEOREM 5.4. *Let $f : X \longrightarrow \overline{\mathbb{R}}$. Then the following statements are equivalent:*

(1) $f \in B_N^\alpha(X)$;

(2) *either $f(X) \subset \mathbb{R}$ and f is quasi-convex and α-Hölder with constant N^α or $f \equiv \pm\infty$;*

(3) *either f is α-b.l.s.d. with α-b.l.s.d. bound N or $f \equiv \pm\infty$.*

Proof. (1) $\Longrightarrow$ (2). If $f \in B_N^\alpha(X)$ is finite at some point, since it is a supremum of finite elementary functions, which are quasi-affine and α-Hölder with constant N^α, f must be finite everywhere, quasi-convex, and α-Hölder with the same constant. If f is not finite at any point and it takes the value $-\infty$ at some point, then $-\infty$ is its only elementary minorant, whence $f \equiv -\infty$.

The implication (2) $\Longrightarrow$ (3) is essentially Theorem 3.2.

(3) $\Longrightarrow$ (1). It is obvious that the functions $\pm\infty$ belong to $B_N^\alpha(X)$. Let f be an α-b.l.s.d. function, $x_0 \in X$, and take $\omega \in \partial_\alpha^- f(x_0) \cap \mathcal{B}^*(0; N)$; then, by Propositions 4.8 and 5.3, we have $(\omega, \omega(x_0)) \in \partial_{h_\alpha} f(x_0) \cap D_N = \partial_{h_{\alpha,N}} f(x_0)$. Therefore, $\partial_{h_{\alpha,N}} f(x_0) \neq \emptyset$ for every $x_0 \in X$. From this we deduce, by implication (1.2), that $f^{h_{\alpha,N} h_{\alpha,N}}(x_0) = f(x_0)$ for any $x_0 \in X$, that is, $f \in B_N^\alpha(X)$, which concludes the proof. □

From the preceding theorem, we deduce that the functions defined on a convex set that are α-b.l.s.d. are those which have an α-Hölder quasi-convex extension to the whole space X.

PROPOSITION 5.5. *Let $K \subset X$ be a nonempty convex set, $f : K \longrightarrow \mathbb{R}$, and $N > 0$. Then the following statements are equivalent:*

(1) *f is α-b.l.s.d. with α-b.l.s.d. bound N;*

(2) *there exists $g : X \longrightarrow \mathbb{R}$ quasi-convex and α-Hölder with constant N^α that extends f.*

Proof. (2) $\Longrightarrow$ (1). By Theorem 5.4, g is α-b.l.s.d. with α-b.l.s.d. bound N and, thus, $f = g_{|K}$ has the same property.

(1) $\Longrightarrow$ (2). Let $\hat{f}$ be the extension of f to the whole space X such that $\hat{f}_{|X\setminus K} \equiv +\infty$. Let $x_0 \in K$. Since $\partial_\alpha^- \hat{f}(x_0) = \partial_\alpha^- f(x_0)$ and f is α-b.l.s.d. with α-b.l.s.d. bound N, we have

$$\partial_\alpha^- \hat{f}(x_0) \cap \mathcal{B}^*(0; N) \neq \emptyset.$$

By Propositions 5.3 and 4.8, we deduce $\partial_{h_{\alpha,N}} \hat{f}(x_0) \neq \emptyset$ for every $x_0 \in K$. Therefore, from implication (1.2), it follows that $(\hat{f}^{h_{\alpha,N}h_{\alpha,N}})_{|K} = \hat{f}_{|K}$. Since $\hat{f}$ is an extension of f, we have $f = (\hat{f}^{h_{\alpha,N}h_{\alpha,N}})_{|K}$. This function is finite and $\hat{f}^{h_{\alpha,N}h_{\alpha,N}} \in B_N^\alpha(X)$; thus, by Theorem 5.4, $\hat{f}^{h_{\alpha,N}h_{\alpha,N}}$ is quasi-convex and α-Hölder with constant N^α. Hence we can take $g = \hat{f}^{h_{\alpha,N}h_{\alpha,N}}$. □

For the second $h_{\alpha,N}$-conjugate of a function one has the following expression.

PROPOSITION 5.6. *Let $f : X \longrightarrow \overline{\mathbb{R}}$. For every $x_0 \in X$, we have*

$$f^{h_{\alpha,N}h_{\alpha,N}}(x_0) = \max_{\omega \in \mathcal{B}^*(0;N)} \inf_{x \in X} \max\{(\omega(x_0 - x))^\alpha + f(x), f(x)\}.$$

Proof. By a method similar to that of Proposition 4.7, we obtain the formula in the statement with supremum instead of maximum. We will prove that this supremum is attained, when $f^{h_{\alpha,N}h_{\alpha,N}}$ is finite, at any $\omega_0 \in \partial_\alpha^- f^{h_{\alpha,N}h_{\alpha,N}}(x_0) \cap \mathcal{B}^*(0; N)$ (this set is nonempty by Theorem 5.4); when $f^{h_{\alpha,N}h_{\alpha,N}}$ is not finite, it is attained at every $\omega \in \mathcal{B}^*(0; N)$, as we will show. In the first case, let $\omega_0 \in \partial_\alpha^- f^{h_{\alpha,N}h_{\alpha,N}}(x_0) \cap \mathcal{B}^*(0; N)$. For every $x \in \dot{S}_{f^{h_{\alpha,N}h_{\alpha,N}}(x_0)}(f^{h_{\alpha,N}h_{\alpha,N}})$, we have

$$f^{h_{\alpha,N}h_{\alpha,N}}(x) \geq f^{h_{\alpha,N}h_{\alpha,N}}(x_0) - (\omega_0(x_0 - x))^\alpha;$$

thus, for any $x \in X$, we can write

$$\begin{aligned} f^{h_{\alpha,N}h_{\alpha,N}}(x) &\geq \min\left\{f^{h_{\alpha,N}h_{\alpha,N}}(x_0), f^{h_{\alpha,N}h_{\alpha,N}}(x_0) - (\omega_0(x_0 - x))^\alpha\right\} \\ &= f^{h_{\alpha,N}h_{\alpha,N}}(x_0) + \min\{0, -(\omega_0(x_0 - x))^\alpha\} \end{aligned}$$

or

$$\max\{(\omega_0(x_0 - x))^\alpha + f^{h_{\alpha,N}h_{\alpha,N}}(x), f^{h_{\alpha,N}h_{\alpha,N}}(x)\} \geq f^{h_{\alpha,N}h_{\alpha,N}}(x_0).$$

Therefore,

$$\begin{aligned} f^{h_{\alpha,N}h_{\alpha,N}}(x_0) &= \sup_{\omega \in \mathcal{B}^*(0;N)} \inf_{x \in X} \max\{(\omega(x_0 - x))^\alpha + f(x), f(x)\} \\ &\geq \inf_{x \in X} \max\{(\omega_0(x_0 - x))^\alpha + f(x), f(x)\} \\ &\geq \inf_{x \in X} \max\{(\omega_0(x_0 - x))^\alpha + f^{h_{\alpha,N}h_{\alpha,N}}(x), f^{h_{\alpha,N}h_{\alpha,N}}(x)\} \\ &\geq f^{h_{\alpha,N}h_{\alpha,N}}(x_0). \end{aligned}$$

Hence,

$$f^{h_{\alpha,N}h_{\alpha,N}}(x_0) = \inf_{x \in X} \max\{(\omega_0(x_0 - x))^\alpha + f^{h_{\alpha,N}h_{\alpha,N}}(x), f^{h_{\alpha,N}h_{\alpha,N}}(x)\}.$$

If $f^{h_{\alpha,N}h_{\alpha,N}}$ is not finite, by Proposition 5.4, we have either $f^{h_{\alpha,N}h_{\alpha,N}} \equiv +\infty$ or $f^{h_{\alpha,N}h_{\alpha,N}} \equiv -\infty$. In both cases, at every $\omega \in \mathcal{B}^*(0;N)$ the maximum is attained (note that $f^{h_{\alpha,N}h_{\alpha,N}} \equiv +\infty$ only if $f \equiv +\infty$). □

The next proposition states that, in the class of α-Hölder functions with constant N, the Fréchet-differentiable quasi-affine functions are supremal generators.

PROPOSITION 5.7. *Let $f : X \longrightarrow \mathbb{R}$. Then, the following statements are equivalent:*

(1) *f is quasi-convex and α-Hölder with constant N;*

(2) *f can be expressed as a supremum of Fréchet-differentiable quasi-affine functions that are α-Hölder with constant N.*

Proof. The implication (2) $\Longrightarrow$ (1) is immediate. To prove (1) $\Longrightarrow$ (2), in view of Theorem 5.4, implication (2) $\Longrightarrow$ (1), we may assume, without loss of generality, that $f = \min\{-(k-\omega)^\alpha, 0\} + \mu$, with $\omega \in \mathcal{B}^*(0; N^{1/\alpha})$ and $k, \mu \in \mathbb{R}$. Then $f = \sup_{\nu>0} j_{\alpha,k,\mu,\nu} \circ \omega$, where $j_{\alpha,k,\mu,\nu} : \mathbb{R} \longrightarrow \mathbb{R}$ is defined by

$$j_{\alpha,k,\mu,\nu}(t) = \begin{cases} -(k-t)^\alpha + \mu & \text{if } t \leq k - \nu, \\ -\dfrac{\nu^{\alpha+1}}{\alpha(t-k+\nu)+\nu} + \mu & \text{if } t \geq k - \nu. \end{cases}$$

Since $j_{\alpha,k,\mu,\nu}$ is differentiable and α-Hölder with constant 1, we obtain (2). □

If $f : X \longrightarrow \mathbb{R}$ is quasi-convex and α-Hölder with constant N, then, by using Proposition 5.6, we can explicitly construct a family of quasi-affine functions, α-Hölder with constant N, whose supremum is f. Namely, since $f = f^{h_{\alpha,M}h_{\alpha,M}}$, with $M = N^{1/\alpha}$, one has $f = \sup_{\omega \in \mathcal{B}^*(0;M)} \varphi_\omega$, where

$$\varphi_\omega(x_0) = \inf_{x \in X} \max\{(\omega(x_0 - x))^\alpha + f(x), f(x)\} \qquad (x_0 \in X).$$

On the other hand, since

$$f = \sup_{\omega \in \mathcal{B}^*(0;M), k \in \mathbb{R}} \left\{ \min\{-(k-\omega)^\alpha, 0\} + k - f^{h_{\alpha,M}}(\omega, k) \right\},$$

the proof of Proposition 5.7 provides a method for obtaining an explicit expression for f of the type stated in (2) of Proposition 5.7.

From the preceding proposition we deduce a new description of the set $\Delta_\alpha(X)$ for a normed space X.

THEOREM 5.8. *Let $f : X \longrightarrow \overline{\mathbb{R}}$. The following statements are equivalent:*

(1) *$f \in \Delta_\alpha(X)$;*

(2) *f can be expressed as a supremum of α-Hölder quasi-convex functions;*

(3) *f can be expressed as a supremum of α-Hölder Fréchet-differentiable quasi-affine functions.*

Proof. Implications (1) $\Longrightarrow$ (2) and (3) $\Longrightarrow$ (1) are immediate consequences of the equality appearing in (3) of Proposition 5.2, while (2) $\Longrightarrow$ (3) follows from Proposition 5.7. □

For $f \in \Delta_\alpha(X)$, explicit representations of the forms stated in (2) and (3) of the preceding theorem can be obtained by straightforward modifications of the methods described above.

The necessary and sufficient conditions given in Theorem 5.8 for a function $f : X \longrightarrow \overline{\mathbb{R}}$ to belong to $\Delta_\alpha(X)$ are difficult to check in practice. The aim of our next results is to exhibit simpler conditions that are either necessary or sufficient.

We say that a function $f : X \longrightarrow \overline{\mathbb{R}}$ *satisfies the α-growth condition* if there exists some real number $M > 0$ such that $f + M\| \cdot \|^{\alpha}$ is bounded below. When $\alpha = 2$ this is the quadratic growth condition of Rockafellar [12, p. 273]. For f to satisfy the α-growth condition it is necessary that

$$\liminf_{\| x \| \to +\infty} \frac{f(x)}{\| x \|^{\alpha}} > -\infty.$$

This is also sufficient if f is l.s.c., but not in general, as we can see by taking $f : \mathbb{R} \longrightarrow \mathbb{R}$ given by $f(x) = -1/|x|$ if $x \neq 0$ and $f(0) = 0$.

Observing that Proposition 3.1 and Corollary 3.6 in [7] are also valid for functions defined on a normed space, we get the following result.

LEMMA 5.9. *Let $f : X \longrightarrow \overline{\mathbb{R}}$ and $x_0 \in X$ be such that $f(x_0) > -\infty$. Then f is l.s.c. at x_0 and satisfies the α-growth condition if and only if*

$$f(x_0) = \sup_{N>0} \inf_{x \in X} \{f(x) + N\| x - x_0 \|^{\alpha}\}.$$

PROPOSITION 5.10. *Let $f \in \Delta_{\alpha}(X)$ be such that $f \not\equiv -\infty$. Then f is quasi-convex and l.s.c. and satisfies the α-growth condition.*

Proof. If $f \in \Delta_{\alpha}(X)$ is finite at some point, there exist $\omega \in X^*$, $k \in \mathbb{R}$ and $\mu \in \mathbb{R}$, such that $f \geq \min\{-(k - \omega)^{\alpha}, 0\} + \mu$, and, since this latter function satisfies the α-growth condition, f has the same property. □

PROPOSITION 5.11. *Let $f : X \longrightarrow \overline{\mathbb{R}}$ be a l.s.c. quasi-convex function that satisfies the α-growth condition. If for every $(x_0, \lambda) \in (X \times \mathbb{R}) \setminus \operatorname{epi} f$ there exist $\theta \in X^*$ and $k > 0$ such that*

$$\theta(d) \geq k\| d \|, \qquad d \in x_0 - \dot{S}_{\lambda}(f),$$

then $f \in \Delta_{\alpha}(X)$.

Proof. We will apply Lemma 4.2. Let $(x_0, \lambda) \in (X \times \mathbb{R}) \setminus \operatorname{epi} f$. By Lemma 5.9, there is some $N > 0$ such that $f(x) + N\| x - x_0 \|^{\alpha} \geq \lambda$ for every $x \in X$. Take θ and k as in the hypothesis. Every $x \in \dot{S}_{\lambda}(f)$ satisfies

$$\lambda - f(x) \leq N\| x - x_0 \|^{\alpha} \leq \frac{N}{k^{\alpha}}(\theta(x_0 - x))^{\alpha}.$$

Therefore, $\omega = (N^{1/\alpha}/k)\theta$ satisfies the condition in Lemma 4.2. □

Given a set $K \subset X$, we will denote, for any $\epsilon > 0$,

$$\Phi_{\epsilon}(K) = \bigcup_{\substack{H \text{ closed hyperplane} \\ H \cap K \neq \emptyset}} \Pi_{H,\epsilon}(K),$$

where $\Pi_{H,\epsilon}(K) = \{y \in H \,|\, d(x, y) \leq d(x, H) + \epsilon \text{ for some } x \in K\}$, with $d(x, y) = \| x - y \|$ and $d(x, H) = \inf_{y \in H} d(x, y)$.

PROPOSITION 5.12. *Let $f : X \longrightarrow \overline{\mathbb{R}}$ and $x_0 \in X$ be such that $f(x_0) \in \mathbb{R}$. If there exists some $\epsilon > 0$ for which f is α-Hölder on $\Phi_{\epsilon}(\dot{S}_{f(x_0)}(f) \cup \{x_0\})$, then*

$$\operatorname{cone} \partial_{\alpha}^{-} f(x_0) = Tf(x_0) = \partial^{*} f(x_0).$$

Proof. By some results of Crouzeix [2, Prop. 12, 13, 13′, pp. 42–43], we know that $Tf(x_0) \subset \partial^* f(x_0)$ and that these sets are convex cones; therefore, by Proposition 2.6, we have

$$\operatorname{cone}\partial_\alpha^- f(x_0) \subset Tf(x_0) \subset \partial^* f(x_0).$$

It only remains to prove that $\partial^* f(x_0) \subset \operatorname{cone}\partial_\alpha^- f(x_0)$. It is immediate if $f(x_0) = \min_{x\in X} f(x)$, since in this case $\partial^* f(x_0) = X^* = \partial_\alpha^- f(x_0)$, by Proposition 13′ in [2, p. 43] and our Proposition 2.2.

Suppose $f(x_0) > \inf_{x\in X} f(x)$; then, by the mentioned results, we have $0 \notin \partial^* f(x_0)$. Let $\omega \in \partial^* f(x_0)$ and take $x \in X$ such that $f(x) < f(x_0)$. Without loss of generality, we assume that $\| \omega \|^* = 1$. Then, by definition of $\partial^* f(x_0)$, we have $\omega(x) < \omega(x_0)$.

Let $x_\epsilon \in \Pi_{H,\epsilon}(K)$, where H is the closed hyperplane consisting of those $y \in X$ such that $\omega(y) = \omega(x_0)$, and let N be a Hölder constant for f on $\Phi_\epsilon(\dot{S}_{f(x_0)}(f) \cup \{x_0\})$. We have

$$f(x_\epsilon) \geq f(x_0),$$

since $\omega(x_\epsilon) = \omega(x_0)$ and $\omega \in \partial^* f(x_0)$; therefore, $\omega(x - x_\epsilon) = \omega(x - x_0) < 0$. Using (3.1), we can write $\| x - x_\epsilon \| \leq \omega(x_0 - x) + \epsilon$. Hence,

$$\begin{aligned} f(x) - f(x_0) &\geq f(x) - f(x_\epsilon) \geq -N\| x - x_\epsilon \|^\alpha \geq -N(\omega(x_0 - x) + \epsilon)^\alpha \\ &= -[N^{1/\alpha}(\omega(x_0 - x) + \epsilon)]^\alpha. \end{aligned}$$

This inequality remains true when we replace ϵ by any $\epsilon' \in (0, \epsilon)$, since such ϵ' also satisfies the condition imposed on ϵ in the statement. Thus

$$f(x) - f(x_0) \geq -[N^{1/\alpha}\omega(x_0 - x)]^\alpha.$$

Hence, $N^{1/\alpha}\omega \in \partial_\alpha^- f(x_0)$ and, therefore, $\omega \in \operatorname{cone}\partial_\alpha^- f(x_0)$. □

6. Applications to optimization theory.

In this section we will apply the theory of the preceding ones to obtain duality results in optimization.

Let X be an arbitrary set, Y be a locally convex space with dual Y^*, and $\phi : X \times Y \longrightarrow \overline{\mathbb{R}}$. We consider the family of optimization problems $\inf_{x\in X} \phi(x, y)$ depending on a parameter $y \in Y$. The unperturbed primal problem (P) will be the one corresponding to the perturbation parameter $y = \underline{0}$, that is, $\inf_{x\in X} \phi(x, 0)$. We will denote by p the perturbation function $p : Y \longrightarrow \overline{\mathbb{R}}$ which assigns to each y the optimal value of the perturbed problem associated to it. The dual problem of (P) corresponding to the family of perturbations is

$$\sup_{(\theta,k)\in Y^*\times\mathbb{R}} \left\{ h_\alpha(0, (\theta, k)) - p^{h_\alpha}(\theta, k) \right\}, \qquad (D_\alpha).$$

Evidently the optimal value of the dual problem (D_α) is $p^{h_\alpha h_\alpha}(0)$ and, in consequence, as $p^{h_\alpha h_\alpha}(0) \leq p(0)$, we have weak duality. Moreover, by Proposition 4.13, the duality gap $p(0) - p^{h_\alpha h_\alpha}(0)$ decreases when α increases.

Using the expression of the second h_α-conjugate that we obtained in Proposition 4.7, we get

$$\begin{aligned} p^{h_\alpha h_\alpha}(0) &= \sup_{\theta\in Y^*} \inf_{y\in Y} \{p(y) + \max\{(-\theta(y))^\alpha, 0\}\} \\ &= \sup_{\theta\in Y^*} \inf_{y\in Y} \left\{ \inf_{x\in X} \phi(x, y) + \max\{(-\theta(y))^\alpha, 0\} \right\} \\ &= \sup_{\theta\in Y^*} \inf_{y\in Y, x\in X} \{\phi(x, y) + \max\{(-\theta(y))^\alpha, 0\}\}. \end{aligned}$$

From this, we deduce that the dual problem consists, equivalently, in finding

$$\sup_{\theta\in Y^*}\ \inf_{y\in Y, x\in X}\{\phi(x,y)+\max\{(-\theta(y))^\alpha,0\}\}, \qquad (D'_\alpha).$$

There is no duality gap if and only if $p(0)=p^{h_\alpha h_\alpha}(0)$. In this formulation of the dual problem, the objective function does not depend on k and, by Proposition 4.8, one can prove that the set of optimal solutions is $\partial_\alpha^- p^{h_\alpha h_\alpha}(0)$. The following strong duality theorem can be easily proved.

PROPOSITION 6.1. *Problem (D'_α) has an optimal solution and there is no duality gap if and only if p is α-l.s.d. at* 0.

When ϕ is defined by

$$\phi(x,y)=\begin{cases} f(x) & \text{if } g(x)+y\le 0,\\ +\infty & \text{otherwise,}\end{cases}$$

where $f: X\longrightarrow \mathbb{R}$, $g: X\longrightarrow Y$, and the order relation $\le$ in Y is defined by means of a closed convex cone $K\subset Y$, that is, $y_1\le y_2$ if $y_2-y_1\in K$, then we have

$$\begin{aligned} p^{h_\alpha h_\alpha}(0) &= \sup_{\theta\in Y^*}\inf_{y\in Y}\left\{\inf_{x\in X}\{f(x)\,|\,g(x)+y\le 0\}+\max\{(-\theta(y))^\alpha,0\}\right\}\\ &= \sup_{\theta\in Y^*}\inf_{x\in X}\left\{\inf_{y\in Y}\{f(x)+\max\{(-\theta(y))^\alpha,0\}\}\,|\,g(x)+y\le 0\right\}.\end{aligned}$$

On the other hand, given $\theta\in Y^*$,

$$\begin{aligned}&\inf_{y\in Y}\{f(x)+\max\{(-\theta(y))^\alpha,0\}\}\,|\,g(x)+y\le 0\}\\ &\quad=\begin{cases} f(x)+\max\{(\theta(g(x)))^\alpha,0\} & \text{if } \theta\ge 0,\\ f(x) & \text{if } \theta\not\ge 0,\end{cases}\end{aligned}$$

where, as usual, $\theta\ge 0$ means that $\theta(y)\ge 0$ for all $y\in K$. Therefore,

$$p^{h_\alpha h_\alpha}(0)=\sup_{\theta\ge 0}\inf_{x\in X}\{f(x)+\max\{(\theta(g(x)))^\alpha,0\}$$

and thus the dual problem (D'_α) can be written, equivalently, as

$$\sup_{\theta\ge 0}\inf_{x\in X}\{f(x)+\max\{(\theta(g(x)))^\alpha,0\}, \qquad (D''_\alpha).$$

This dual problem is equivalent to (i.e., has the same optimal value as) that of Crouzeix [2] for quasi-convex problems, if, for example, f is bounded below, since in this case we have $p^{h_\alpha h_\alpha}=p_{\bar{q}}$ (see Corollary 4.6).

We conclude this section by presenting a necessary and sufficient Kuhn–Tucker-type optimality condition for quasi-convex mathematical programming problems in terms of α-lower subgradients. Our result is based on the following proposition, which provides a calculus rule for the α-lower subdifferential of the maximum of a finite number of functions. We recall that $f:\mathbb{R}^n\longrightarrow\mathbb{R}$ is strictly quasi-convex [6] if for all $x,y\in\mathbb{R}^n$ with $f(x)\ne f(y)$ and all $\lambda\in(0,1)$ one has $f((1-\lambda)x+\lambda y)<\max\{f(x),f(y)\}$.

PROPOSITION 6.2. *Consider $f_i:\mathbb{R}^n\longrightarrow\mathbb{R}, i=1,\dots,p$, α-Hölder strictly quasi-convex functions and let $f=\max_{i=1,\dots,p} f_i$ and $x_0\in\mathbb{R}^n$ be such that $f(x_0)>\inf_{x\in\mathbb{R}^n} f(x)$. Then one has*

$$\overline{\text{co}}\bigcup_{i\,|\,f_i(x_0)=f(x_0)}\partial_\alpha^- f_i(x_0)\subset\partial_\alpha^- f(x_0)\subset\text{co cone}\bigcup_{i\,|\,f_i(x_0)=f(x_0)}\partial_\alpha^- f_i(x_0).$$

Proof. The first inclusion can be proved as a simple exercise (and requires no assumption on the f_i's). Using Proposition 2.6, the inclusion of the tangential in the quasi-subdifferential, Proposition 15 in [2, p. 81], the fact that quasi-subdifferentials are convex cones (see [5, Thm. 6, p. 442]), and Proposition 5.12, we obtain the following inclusions:

$$\begin{aligned}\partial_\alpha^- f(x_0) \subset Tf(x_0) \subset \partial^* f(x_0) &= \sum_{i \mid f_i(x_0)=f(x_0)} \partial^* f_i(x_0) \\ &\subset \sum_{i \mid f_i(x_0)=f(x_0)} (\partial^* f_i(x_0) \cup \{0\}) \\ &= \operatorname{co}\left(\bigcup_{i \mid f_i(x_0)=f(x_0)} \partial^* f_i(x_0) \cup \{0\}\right) \\ &= \operatorname{co}\left(\bigcup_{i \mid f_i(x_0)=f(x_0)} \operatorname{cone} \partial_\alpha^- f_i(x_0) \cup \{0\}\right) \\ &= \operatorname{co}\left(\operatorname{cone} \bigcup_{i \mid f_i(x_0)=f(x_0)} \partial_\alpha^- f_i(x_0) \cup \{0\}\right) \\ &= \left(\operatorname{co}\operatorname{cone} \bigcup_{i \mid f_i(x_0)=f(x_0)} \partial_\alpha^- f_i(x_0)\right) \cup \{0\};\end{aligned}$$

since $0 \notin \partial_\alpha^- f(x_0)$ (see Proposition 2.2), we obtain the second inclusion in the statement. □

In the preceding proposition the hypothesis that x_0 is not a minimum of f cannot be suppressed, as we can see by taking $f_1, f_2 : \mathbb{R}^2 \longrightarrow \mathbb{R}$ defined by

$$f_1(x, y) = \min\{-(-x)^\alpha, 0\}$$

and

$$f_2(x, y) = \min\{-x^\alpha, 0\}.$$

We have that $f(x, y) = \max\{f_1(x, y), f_2(x, y)\} = 0$ for all x, y. It is easy to see that these functions satisfy the hypotheses of Proposition 6.2. Take $x_0 = (0, 0)$; one can prove that

$$\partial_\alpha^- f_1(x_0) = \{(x^*, 0) \mid x^* \geq 1\}$$

and

$$\partial_\alpha^- f_2(x_0) = \{(x^*, 0) \mid x^* \leq -1\}.$$

In consequence,

$$\operatorname{co}\operatorname{cone} \bigcup_{i \mid f_i(x_0)=f(x_0)} \partial_\alpha^- f_i(x_0) = \operatorname{co}\operatorname{cone}\left(\partial_\alpha^- f_1(x_0) \cup \partial_\alpha^- f_2(x_0)\right) = \mathbb{R} \times \{0\},$$

while $\partial_\alpha^- f(x_0) = \mathbb{R}^2$, since x_0 is a minimum of f (see Proposition 2.2).

In general, the first inclusion in Proposition 6.2 is not an equality. For $\alpha = 1$, this was proved by Martínez-Legaz [7, p. 220]. For $\alpha \in (0,1)$, take the functions $f_1, f_2 : \mathbb{R} \longrightarrow \mathbb{R}$ defined by $f_1(x) = |x|^\alpha$ and $f_2 \equiv 1$. Then, we have that $\partial_\alpha^- f_1(2) = [1, +\infty)$, but $\partial_\alpha^- f(2) = \left[(2^\alpha - 1)^{1/\alpha}, +\infty\right)$.

We say that the functions $g_i : \mathbb{R}^n \longrightarrow \mathbb{R}, i = 1, \ldots, m$, satisfy Slater's condition if there exists $\tilde{x} \in \mathbb{R}^n$ such that $g_i(\tilde{x}) < 0$ for every $i = 1, \ldots, m$.

PROPOSITION 6.3. *Let $f, g_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ with $i = 1, \ldots, m$ be α-Hölder strictly quasi-convex functions and let $x_0 \in \mathbb{R}^n$ be such that $g_i(x_0) \leq 0$ $(i = 1, \ldots, m)$. If $g_1, \ldots, g_m$ satisfy Slater's condition, then x_0 is optimal for*

$$(P) \qquad \inf\{f(x) \mid g_i(x) \leq 0 \ \ (i = 1, \ldots, m)\}$$

if and only if there are nonnegative numbers $\lambda_i, i = 1, \ldots, m$, such that $\lambda_i g_i(x_0) = 0$ for $i = 1, \ldots, m$ and $0 \in \partial_\alpha^- f(x_0) + \sum_{i=1}^m \lambda_i \partial_\alpha^- g_i(x_0)$.

Proof. First, suppose the existence of the λ_i's. Then, clearly,

$$0 \in \operatorname{co}\left(\partial_\alpha^- f(x_0) \cup \bigcup_{i \in I(x_0)} \partial_\alpha^- g_i(x_0)\right),$$

where $I(x_0) = \{i \in \{1, \ldots, m\} | g_i(x_0) = 0\}$. Hence, by Proposition 6.2, $0 \in \partial_\alpha^- v(x_0)$, $v : \mathbb{R}^n \longrightarrow \mathbb{R}$ being the function defined by

$$v(x) = \max\{f(x) - f(x_0), g_1(x), \ldots, g_m(x)\}.$$

Therefore, in view of Proposition 2.2, x_0 is a global minimum of v. Let x be a feasible point for (P). For small enough $t > 0$, the point $x_t = (1-t)x + t\tilde{x}$ satisfies $g_i(x_t) < 0$ $(i = 1, \ldots, m)$ and $v(x_t) \geq v(x_0) = 0$. This implies that $f(x_t) \geq f(x_0)$. By letting $t \to 0^+$, we get $f(x) \geq f(x_0)$. This proves that x_0 is optimal for (P).

Conversely, if x_0 is an optimal solution to problem (P), then clearly $v(x_0) = \min_{x \in \mathbb{R}^n} v(x)$ and thus, by Propositions 2.2 and 6.2,

$$0 \in \partial_\alpha^- v(x_0) \subset \operatorname{co} \operatorname{cone}\left(\partial_\alpha^- f(x_0) \cup \bigcup_{i \in I(x_0)} \partial_\alpha^- g_i(x_0)\right).$$

Therefore, there exist $\lambda_i' \geq 0$, $i \in I(x_0) \cup \{0\}$, with $\lambda_0' + \sum_{i \in I(x_0)} \lambda_i' = 1$, such that $0 \in \lambda_0' \partial_\alpha^- f(x_0) + \sum_{i \in I(x_0)} \lambda_i' \partial_\alpha^- g_i(x_0)$. If $\lambda_0' = 0$, then we should have, by Proposition 6.2,

$$0 \in \sum_{i \in I(x_0)} \lambda_i' \partial_\alpha^- g_i(x_0) \subset \operatorname{co} \bigcup_{i \in I(x_0)} \partial_\alpha^- g_i(x_0) \subset \partial_\alpha^- \left(\max_{i \in I(x_0)} g_i\right)(x_0)$$

and, using Proposition 2.2, we should deduce

$$0 = \max_{i \in I(x_0)} g_i(x_0) = \min_{x \in \mathbb{R}^n} \max_{i \in I(x_0)} g_i(x) \leq \max_{i \in I(x_0)} g_i(\tilde{x}) < 0,$$

which is absurd. We conclude that λ_0' is greater than 0 and hence, taking $\lambda_i = \lambda_i'/\lambda_0'$ $(i \in I(x_0))$ and $\lambda_i = 0$ $(i \in \{1, \ldots, m\} \setminus I(x_0))$, the required Kuhn–Tucker-type conditions are satisfied. □

As a referee pointed out to us, because the minimization of $f(x)$ is equivalent to minimizing $g(x) = \exp(f(x))$, one could apply the exponential transformation to optimization problems before analyzing them by our methods. The advantage of this approach lies in the fact that all relevant functions become bounded from below, which make them more likely to be α-l.s.d. or to belong to $\Delta_\alpha(X)$. On the other hand, if f is α-l.s.d. at some point $x_0 \in X$, so is g. Indeed, one can easily check that, for any $\omega \in \partial_\alpha^- f(x_0)$, one has $(g(x_0))^{1/\alpha}\omega \in \partial_\alpha^- g(x_0)$. Furthermore, by Corollary 4.6, the following nice equivalence holds: $g \in \Delta_\alpha(X)$ if and only if f is quasi-convex and l.s.c. According to the same result, the l.s.c. quasi-convex hull of an arbitrary function $f : X \longrightarrow \overline{\mathbb{R}}$ satisfies $f_{\bar{q}} = \ln(\exp f)^{h_\alpha h_\alpha}$. Therefore, Proposition 4.7 yields the following formula for the l.s.c. quasi-convex hull of any (i.e., not necessarily bounded from below) function $f : X \longrightarrow \overline{\mathbb{R}}$,

$$f_{\bar{q}}(x_0) = \sup_{\omega \in X^*} \inf_{x \in X} \max\{\ln[(\omega(x_0 - x))^\alpha + \exp(f(x))], f(x)\}.$$

REFERENCES

[1] E. J. BALDER, *An extension of duality-stability relations to nonconvex problems*, SIAM J. Control Optim., 15 (1977), pp. 329–343.

[2] J. P. CROUZEIX, *Contributions a l'étude des fonctions quasiconvexes*, Ph.D. thesis, Université de Clermont-Ferrand II, Clermont-Ferrand, France, 1977.

[3] ———, *Continuity and differentiability properties of quasiconvex functions on* $\mathbb{R}^n$, in Generalized Concavity in Optimization and Economics (Proceedings, Internat. Conf. Vancouver, 1980), Academic Press, New York, 1981, pp. 109–130.

[4] I. EKELAND AND R. TEMAM, *Analyse convexe et problèmes variationnels*, Dunod-Gautier Villars, Paris, 1974.

[5] H. P. GREENBERG AND W. P. PIERSKALLA, *Quasiconjugate function and surrogate duality*, Cahiers Centre Études Rech. Opér., 15 (1973), pp. 437–448.

[6] O. L. MANGASARIAN, *Nonlinear Programming*, McGraw Hill, New York, 1969.

[7] J. E. MARTÍNEZ-LEGAZ, *On lower subdifferentiable functions*, in Trends in Mathematical Optimization (Proceedings, Internat. Conf. Irsee, 1986), Birkhäuser-Verlag, Boston, 1988, pp. 197–232.

[8] ———, *Quasiconvex duality theory by generalized conjugation methods*, Optimization, 19 (1988), pp. 603–652.

[9] J. J. MOREAU, *Inf-convolution, sous-additivité, convexité des fonctions numériques*, J. Math. Pures Appl., 49 (1970), pp. 109–154.

[10] J. P. PENOT AND M. VOLLE, *Another duality scheme for quasiconvex problems*, in Trends in Mathematical Optimization (Proceedings, Internat. Conf. Irsee, 1986), Birkhäuser Verlag, Boston, 1988, pp. 259–275.

[11] F. PLASTRIA, *Lower subdifferentiable functions and their minimization by cutting planes*, J. Optim. Theory Appl., 46 (1985), pp. 37–53.

[12] R. T. ROCKAFELLAR, *Augmented lagrange multiplier functions and duality in nonconvex programming*, SIAM J. Control Optim., 12 (1974), pp. 268–285.

[13] I. SINGER, *Best Approximation in Normed Linear Spaces by Elements of Linear Subspaces*, Publ. House Acad. Soc. Rep. Romania and Springer-Verlag, New York, 1970.

SIAM J. OPTIMIZATION
Vol. 3, No. 4, pp. 826–842, November 1993

009

A GLOBAL OPTIMIZATION ALGORITHM FOR CONCAVE QUADRATIC PROGRAMMING PROBLEMS*

IMMANUEL M. BOMZE[†] AND GABRIELE DANNINGER[†]

Abstract. Using a global optimality criterion for concave quadratic problems due to Hiriart–Urruty and Lemaréchal, the authors present an algorithm which manages to "escape" from a local solution and lead towards the global one. In addition, this procedure recognizes the unsolvability of the problem (due to unboundedness of the objective function on the feasible region) and also generates an improving feasible direction even if the starting point does not satisfy the Karush/Kuhn/Tucker conditions. As a key subroutine, a recursive procedure will be used which determines whether or not a given symmetric $n \times n$-matrix is copositive, i.e., yields a quadratic form that is positive on a given polyhedral cone. Both this subroutine and the main body of the algorithm frequently employ the simplex method, while all other operations are elementary.

Key words. copositive matrices, convex maximization problem, global optimality conditions

AMS subject classifications. 90C20, 90C30, 65K05

1. Introduction.

Nonconvex quadratic problems consist of minimizing a nonconvex quadratic function over a polyhedron in n-dimensional Euclidean space $\mathbb{R}^n$. They arise in different fields of applications from combinatorial optimization to database problems and VLSI design. The solution of problems of this type is, from the perspective of worst-case complexity, NP-hard; even checking whether a given feasible point is a local solution is also NP-hard [13], [15]. As pointed out by Pardalos in [14], there is in general no local criterion for global optimality. However, the example chosen by the latter author to illustrate this observation,

$$(1.1) \qquad -\sum_{j=1}^{n}(c_j x_j + x_j^2) \to \min, \quad -1 \leq x_i \leq 1, \quad 1 \leq i \leq n,$$

is not quite appropriate since it involves a concave objective function. As we shall show in the sequel, for this kind of problem there is a criterion for global optimality of a feasible point which may be viewed as a local one, and which can be exploited in a global optimization algorithm which avoids being trapped in the domain of attraction of a local solution. Note that for $c_j > 0$ small enough, problem (1.1) has 3^n Karush/Kuhn/Tucker points and 2^n local minima.

The present paper is organized as follows: after reformulating the global optimality criterion due to [10], we arrive at the key subproblem to determine whether or not a given symmetric $n \times n$-matrix is copositive, i.e., yields a quadratic form that is positive on a given polyhedral cone. In §2 we attack this problem by a recursive procedure that reduces the problem dimension. We then focus in §3 on an algorithm for detecting copositivity, and in §4 employ this routine in a global optimization procedure.

Consider a quadratic minimization problem with a concave objective function, or equivalently, the problem

$$(1.2) \qquad \tfrac{1}{2}x^TQx + c^Tx \to \max, \qquad Ax \leq b,$$

*Received by the editors May 28, 1991; accepted for publication (in revised form) July 17, 1992.

[†] Institut für Statistik, Operations Research, und Computerverfahren, Universität Wien, A-1010 Wien, Austria.

where x^T denotes the transpose of an $n \times 1$-vector $x \in \mathbb{R}^n$; Q is a symmetric, positive semidefinite $n \times n$-matrix; $c \in \mathbb{R}^n$; A is an $m \times n$-matrix; and $b \in \mathbb{R}^m$. The algorithm proposed in this paper is based on the characterization of global solutions $\overline{x}$ of (1.2) given by [10]; see also [9]. We shall now briefly describe this approach. Concise proofs as well as calculations and examples can be found in [5].

In [10], Hiriart-Urruty and Lemaréchal start with the observation that a feasible point $\overline{x} \in M$ is a global solution to (1.2) if and only if

$$\partial_\varepsilon g(\overline{x}) \subseteq N_\varepsilon(M, \overline{x}) \quad \text{for all } \varepsilon > 0 . \tag{1.3}$$

Here $g(x) = \frac{1}{2}x^TQx + c^Tx$ is the objective function,

$$\partial_\varepsilon g(\overline{x}) := \{y \in \mathbb{R}^n : g(x) - g(\overline{x}) \geq y^T(x - \overline{x}) - \varepsilon \text{ for all } x \in \mathbb{R}^n\}$$

is the ε-subdifferential of g at $\overline{x}$, while $M = \{x \in \mathbb{R}^n : Ax \leq b\}$ denotes the set of feasible points of (1.2) and

$$N_\varepsilon(M, \overline{x}) := \{y \in \mathbb{R}^n : y^T(x - \overline{x}) \leq \varepsilon \text{ for all } x \in M\}$$

is the set of ε-normal directions to M at $\overline{x}$. Since both $S(\varepsilon) = \partial_\varepsilon g(\overline{x})$ and $N(\varepsilon) = N_\varepsilon(M, \overline{x})$ are convex sets, the inclusion in (1.3) holds if and only if

$$\sigma_{S(\varepsilon)}(d) \leq \sigma_{N(\varepsilon)}(d) \quad \text{for all directions } d \in \mathbb{R}^n , \tag{1.4}$$

where for a set $Y \subseteq \mathbb{R}^n$, we denote by $\sigma_Y(d) = \sup\{d^Ty : y \in Y\}$ the support functional of Y. Now the optimality characterization (1.3) holds true for general convex g and M. However, relation (1.4) is in general not very helpful. Here we can exploit the simple structure of g and M to make relation (1.4) more explicit with the help of the identities

$$\sigma_{S(\varepsilon)}(d) = d^T\overline{y} \quad \text{with } \overline{y} := \begin{cases} Q\overline{x} + c + \sqrt{\frac{2\varepsilon}{d^TQd}}Qd, & \text{if } d^TQd > 0, \\ Q\overline{x} + c, & \text{otherwise} \end{cases} \tag{1.5}$$

(observe that in any case $d^T\overline{y} = d^T(Q\overline{x} + c) + \sqrt{2\varepsilon\, d^TQd}$ holds), as well as

$$\sigma_{N(\varepsilon)}(d) = \varepsilon z(d) \quad \text{with } z(d) := \begin{cases} \max\big[\{0\} \cup \{(Ad)_i/u_i : i \notin I\}\big], & \text{if } d \in \Gamma, \\ +\infty, & \text{otherwise.} \end{cases} \tag{1.6}$$

Here, we denote by $I = I(\overline{x}) := \{i \in \{1, \ldots, m\} : (A\overline{x})_i = b_i\}$ the set of binding constraints at $\overline{x}$; by $u_i := b_i - (A\overline{x})_i > 0$ the slack variables at $\overline{x}$, $i \notin I$; and the tangential cone of M at $\overline{x}$ by

$$\Gamma := \{d \in \mathbb{R}^n : (Ad)_i \leq 0 \text{ for all } i \in I\} .$$

Now it is easy to see that (1.4) is equivalent to

$$f_d(\delta) = \delta^2 z(d) - \delta\sqrt{2d^TQd} - d^T(Q\overline{x} + c) \geq 0 \quad \text{for all } d \in \mathbb{R}^n , \tag{1.7}$$

where $\delta = \sqrt{\varepsilon}$. Note that $z(d) \geq 0$ always and thus f_d is convex. So instead of (1.3) we shall check in the sequel the inequality $f_d(\delta) \geq 0$ for all $\delta \geq 0$, where $d \in \mathbb{R}^n$ is fixed, but arbitrary. According to (1.6), the relation $f_d(\delta) \geq 0$ is clearly satisfied for all $\delta \geq 0$ if $d \notin \Gamma$, so we only have to investigate directions d belonging to the tangential cone, as one would expect. In case of $z(d) > 0$, the function f_d attains its minimal value at $\delta^* := \sqrt{2d^TQd}/2z(d) > 0$, so that we have only to check $f_d(\delta^*) = -(1/2z(d))d^TQd - d^T(Q\overline{x} + c) \geq 0$, which can be rephrased as

$$-d^TQd - 2d^T(Q\overline{x} + c)z(d) \geq 0\,, \tag{1.8}$$

which also has to hold if $z(d) = 0$, since in this case f_d is affine and thus must have a nonnegative slope in order to be nonnegative for arbitrarily large δ. If we now denote by

$$\begin{aligned}\Gamma_i :=&\{d \in \Gamma : (Ad)_i \geq 0 \text{ and } u_j(Ad)_i \geq u_i(Ad)_j \text{ for all } j \notin I\}\\ =&\left\{d \in \Gamma : z(d) = \frac{(Ad)_i}{u_i}\right\}, \quad i \notin I,\end{aligned} \tag{1.9}$$

and also

$$\begin{aligned}\Gamma_0 :=&\{d \in \Gamma : (Ad)_i \leq 0 \text{ for all } i \notin I\}\\ =&\{d \in \Gamma : z(d) = 0\} = \{d \in \mathbb{R}^n : Ad \leq o\}\,,\end{aligned} \tag{1.10}$$

condition (1.8) can further be reformulated into the conditions

$$d^TQ_id \geq 0 \quad \text{for all } d \in \Gamma_i \text{ and all } i \in \{0, \dots, m\} \setminus I\,, \tag{1.11}$$

where the symmetric $n \times n$-matrices Q_i are defined by

$$Q_i := \begin{cases} -Q, & \text{if } i = 0,\\ B_i - u_iQ, & \text{otherwise,}\end{cases} \tag{1.12}$$

and

$$B_i := -a_i(Q\overline{x} + c)^T - (Q\overline{x} + c)(a_i)^T, \tag{1.13}$$

where $(a_i)^T$ denotes the ith row of A.

Conditions (1.11) alone do not suffice to ensure validity of (1.3) and hence global optimality of $\overline{x}$. Indeed, in case of $z(d) = 0$, where f_d is an affine function, not only the slope of f_d has to be nonnegative to ensure $f_d(\delta) \geq 0$ for all $\delta \geq 0$. In addition, the relation $f_d(0) \geq 0$ has to hold in order to guarantee $f_d(\delta) \geq 0$ also for small values of δ. Now observe that the condition

$$0 \leq f_d(0) = -d^T(Q\overline{x} + c) = -d^T\nabla g(\overline{x}) \quad \text{for all } d \in \Gamma$$

exactly corresponds to the Karush/Kuhn/Tucker conditions. Hence for a Karush/Kuhn/Tucker point $\overline{x}$ the weaker condition

$$d^T(Q\overline{x} + c) \leq 0 \quad \text{for all } d \in \Gamma_0 \tag{1.14}$$

is automatically satisfied. So (1.11) and (1.14) together ensure (1.3) and hence global optimality, but the latter can be ignored if $\overline{x}$ is a Karush/Kuhn/Tucker point. Therefore, let us now consider the problem of determining whether or not

$$d^TQd \geq 0 \quad \text{holds for all } d \in \Gamma\,, \tag{1.15}$$

where Q is a symmetric $n \times n$-matrix and $\Gamma \subseteq \mathbb{R}^n$ is defined by a set of (homogeneous) linear constraints, i.e., Γ is a polyhedral cone. If (1.15) pertains, the matrix Q is said to be "Γ-copositive." Of course (1.15) trivially holds if Q is positive semidefinite, or if Γ is trivial, i.e., $\Gamma = \{o\}$, but in general neither of these properties is shared by the matrices Q_i and the cones Γ_i occurring in the optimality criterion (1.11).

Hence the subsequent sections will be devoted to an algorithm which both detects copositivity and returns a direction $d \in \Gamma$ with $d^TQd < 0$ if (1.15) is invalid. This direction d will then be used in a procedure which essentially helps to "escape" from a local solution $\overline{x}$ of (1.2), and leads towards the global one. In addition, this procedure recognizes the unsolvability of the problem (due to unboundedness of the objective function on the feasible region) and also generates an improving feasible direction d even if the starting point $\overline{x}$ does not satisfy the Karush/Kuhn/Tucker conditions.

2. The recursive structure of copositivity. Consider a polyhedral cone $\Gamma = \{x \in \mathbb{R}^n : Dx \geq o\}$, where D is an $m \times n$ matrix with rows $d_1{}^T, \dots, d_m{}^T$, and suppose we want to know whether or not a given symmetric $n \times n$ matrix Q is Γ-copositive or not. Essentially following [1], in the sequel we shall reduce this question to the investigation of (strict) Γ_ν-copositivity of certain $(n-1) \times (n-1)$ matrices Q_ν, where $\Gamma_\nu \subseteq \mathbb{R}^{n-1}$ are suitably defined polyhedral cones, $1 \leq \nu \leq l$. At first let us "decompose" the relation $x \in \Gamma$; to this end we need some notation: keep $i \in \{1, \dots, n\}$ fixed and let

$$\begin{aligned} J_+(i) &:= \{j \in \{1, \dots, m\} : d_{ji} > 0\}, \\ J_0(i) &:= \{j \in \{1, \dots, m\} : d_{ji} = 0\}, \\ J_-(i) &:= \{j \in \{1, \dots, m\} : d_{ji} < 0\}. \end{aligned}$$

For $x = [x_j]_{1 \leq j \leq n} \in \mathbb{R}^n$ define $y := [x_j]_{j \neq i} \in \mathbb{R}^{n-1}$ as well as

$$\alpha_j(y) := \begin{cases} -\frac{1}{d_{ji}} \sum_{k \neq i} d_{jk} x_k, & \text{if } j \in J_-(i) \cup J_+(i), \\ \sum_{k \neq i} d_{jk} x_k, & \text{if } j \in J_0(i). \end{cases} \tag{2.1}$$

Then $x \in \Gamma$ if and only if

$$\alpha_j(y) \leq x_i \leq \alpha_k(y), \text{ all } j \in J_+(i), \text{ all } k \in J_-(i), \quad \text{and} \quad \alpha_j(y) \geq 0, \text{ all } j \in J_0(i).$$

Note that

$$\begin{aligned} \Theta_0 := \{y \in \mathbb{R}^{n-1} : \alpha_j(y) \geq 0, \text{all } j \in J_0(i)\,, \\ \alpha_r(y) \leq \alpha_s(y), \text{all } (r,s) \in J_+(i) \times J_-(i)\} \end{aligned} \tag{2.2}$$

as well as

$$\begin{aligned} \Theta_r &:= \{y \in \Theta_0 : \alpha_r(y) = \max\nolimits_{j \in J_+(i)} \alpha_j(y)\}, \quad \text{and} \\ \Theta_s &:= \{y \in \Theta_0 : \alpha_s(y) = \min\nolimits_{k \in J_-(i)} \alpha_k(y)\} \end{aligned} \tag{2.3}$$

are polyhedral cones for any $r \in J_+(i)$, $s \in J_-(i)$. It will prove useful to distinguish the following four cases ($\emptyset$ denoting the empty set):

(a) $J_-(i) \neq \emptyset$ and $J_+(i) \neq \emptyset$,
(b) $J_-(i) = \emptyset$, but $J_+(i) \neq \emptyset$,
(c) $J_-(i) \neq \emptyset$, but $J_+(i) = \emptyset$, and
(d) $J_-(i) = J_+(i) = \emptyset$.

Next we establish the key relation for recursive dimensional reduction of copositivity: denote by $p := [q_{ij}]_{j \neq i} \in \mathbb{R}^{n-1}$ and by B the symmetric $(n-1) \times (n-1)$ matrix obtained by deleting the ith row and the ith column in Q. Furthermore, define for $y \in \mathbb{R}^{n-1}$

$$f(\lambda|y) := q_{ii}\lambda^2 + 2\lambda p^T y + y^T B y, \quad \lambda \in \mathbb{R}.$$

If $q_{ii} > 0$, then the quadratic function $f(.|y)$ has a global minimum at

$$\lambda_0(y) := -\frac{p^T y}{q_{ii}}. \tag{2.4}$$

Now Q is Γ-copositive if and only if

$$x^T Q x = f(x_i|y) \geq 0 \quad \text{for all}\, x \in \Gamma,$$

which holds if and only if

$$\begin{aligned}
&\text{in case (a), } f(\lambda|y) \geq 0, \text{ all}\, \lambda \in [\alpha_r(y), \alpha_s(y)], \text{ whenever}\, y \in \Theta_r \cap \Theta_s;\\
&\text{in case (b), } f(\lambda|y) \geq 0, \text{ all}\, \lambda \geq \alpha_r(y), \text{ whenever}\, y \in \Theta_r;\\
&\text{in case (c), } f(\lambda|y) \geq 0, \text{ all}\, \lambda \leq \alpha_s(y), \text{ whenever}\, y \in \Theta_s;\\
&\text{in case (d), } f(\lambda|y) \geq 0, \text{ all}\, \lambda \in \mathbb{R}, \text{ whenever}\, y \in \Theta_0.
\end{aligned}$$

It now remains to show that the positivity conditions on f specified above can be reformulated into equivalent copositivity conditions on $\mathbb{R}^{n-1}$. This is possible since $f(\beta(y)|y)$ is a quadratic form in y for any linear functional β of y. Note that in case (a) of a bounded interval $\mathcal{I}$ for interesting values of λ, it suffices for concave functions $f(.|y)$ to check its values at both boundary points of $\mathcal{I}$ (case (a2)). In the strictly convex cases (a1), (b1), (c1), and (d1), the function $f(.|y)$ attains its minimum over $\mathcal{I}$ either at a boundary point of $\mathcal{I}$ or at $\lambda_0(y)$ if the latter belongs to $\mathcal{I}$. To deal with cases (b2), (c2), and (d2) below, where $q_{ii} = 0$ and hence $f(.|y)$ is an affine function on an unbounded interval, we have to check both slope and value at $\lambda = 0$ of $f(.|y)$. For this reason, we introduce the following cones:

$$\Theta_0^\star := \{z \in \mathbb{R}^{n-1} : z^T y \geq 0 \text{ for all } y \in \Theta_0\} \quad \text{and} \quad \Theta_0^\perp := \Theta_0^\star \cap -\Theta_0^\star.$$

Dealing with the cases (a), (b), (c), and (d) above separately, we arrive at the following equivalences.

(a) Q is Γ-copositive if and only if for all $(r, s) \in J_+(i) \times J_-(i)$, either

(a1) $q_{ii} > 0$ and

$$\begin{aligned}
f(\lambda_0(y)|y) &\geq 0, \quad \text{if}\, \alpha_r(y) \leq \lambda_0(y) \leq \alpha_s(y),\\
f(\alpha_r(y)|y) &\geq 0, \quad \text{if}\, \lambda_0(y) \leq \alpha_r(y),\\
f(\alpha_s(y)|y) &\geq 0, \quad \text{if}\, \lambda_0(y) \geq \alpha_s(y),
\end{aligned}$$

whenever $y \in \Theta_r \cap \Theta_s$, or

(a2) $q_{ii} \leq 0$ and

$$f(\alpha_r(y)|y) \geq 0 \quad \text{as well as}\; f(\alpha_s(y)|y) \geq 0,$$

whenever $y \in \Theta_r \cap \Theta_s$.

(b) Q is Γ-copositive if and only if for all $r \in J_+(i)$, either

(b1) $q_{ii} > 0$ and

$$f(\max\{\lambda_0(y), \alpha_r(y)\}|y) \geq 0, \quad \text{whenever}\; y \in \Theta_r,$$

TABLE 1

Definition of Γ_ν, β_ν, *and* Q_ν *in cases* (a1)–(d2); *the cones* Θ_j *are as in* (2.2) *and* (2.3), *while the functionals* α_j *and* λ_0 *are defined in* (2.1) *and* (2.4), *respectively.*

Case	Γ_ν	β_ν	Q_ν
(a1)	$\{y \in \Theta_r \cap \Theta_s : \alpha_r(y) \leq \lambda_0(y) \leq \alpha_s(y)\}$	λ_0	C
	$\{y \in \Theta_r \cap \Theta_s : \lambda_0(y) \leq \alpha_r(y)\}$	α_r	F_r
	$\{y \in \Theta_r \cap \Theta_s : \alpha_s(y) \leq \lambda_0(y)\}$	α_s	F_s
(a2)	$\Theta_r \cap \Theta_s$	α_r and α_s	F_r and F_s
(b1)	$\{y \in \Theta_r : \lambda_0(y) \leq \alpha_r(y)\}$	α_r	F_r
	$\{y \in \Theta_r : \alpha_r(y) \leq \lambda_0(y)\}$	λ_0	C
(b2)	Θ_r	α_r	F_r
(c1)	$\{y \in \Theta_s : \lambda_0(y) \leq \alpha_s(y)\}$	λ_0	C
	$\{y \in \Theta_s : \alpha_s(y) \leq \lambda_0(y)\}$	α_s	F_s
(c2)	Θ_s	α_s	F_s
(d1)	Θ_0	λ_0	C
(d2)	Θ_0	0	B

or

(b2) $q_{ii} = 0$, $p \in \Theta_0^\star$, and

$$f(\alpha_r(y)|y) \geq 0, \quad \text{whenever } y \in \Theta_r.$$

(c) Q is Γ-copositive if and only if for all $s \in J_-(i)$, either

(c1) $q_{ii} > 0$ and

$$f(\min\{\lambda_0(y), \alpha_s(y)\}|y) \geq 0, \quad \text{whenever } y \in \Theta_s,$$

or

(c2) $q_{ii} = 0$, $-p \in \Theta_0^\star$, and

$$f(\alpha_s(y)|y) \geq 0, \quad \text{whenever } y \in \Theta_s.$$

(d) Q is Γ-copositive if and only if either

(d1) $q_{ii} > 0$ and

$$f(\lambda_0(y)|y) \geq 0, \quad \text{whenever } y \in \Theta_0,$$

or

(d2) $q_{ii} = 0$, $p \in \Theta_0^\perp$, and

$$f(0|y) \geq 0, \quad \text{whenever } y \in \Theta_0\,.$$

The recursive structure described above enables us to reduce the question of Γ-copositivity of an $n \times n$-matrix Q to the investigation of Γ_ν-copositivity of $(n-1) \times (n-1)$ matrices Q_ν, $1 \leq \nu \leq l$. For the sake of transparency, let us specify these quantities in Table 1 below, where also the linear functionals $\beta_\nu(y)$ can be found, which appear as an argument of $f(.|y)$ in the inequalities above. We also rescale the resulting quadratic form $f(\beta_\nu(y)|y)$ by a suitable positive constant. So note that for $\beta_\nu(y) = 0$, we have $f(0|y) = y^TQ_\nu y$, where

$$Q_\nu = B := [q_{jk}]_{j\neq i, k\neq i},$$

while for $\beta_\nu(y) = \lambda_0(y)$, we get $q_{ii}f(\lambda_0(y)|y) = q_{ii}y^TBy - (p^Ty)^2 = y^TQ_\nu y$, where

$$Q_\nu = C := [q_{ii}q_{jk} - q_{ij}q_{ik}]_{j\neq i, k\neq i},$$

and, finally, for $\beta_\nu(y) = \alpha_r(y)$, we arrive at $d_{ri}^2 f(\alpha_r(y)|y) = y^TQ_\nu y$, where

$$Q_\nu = F_r := [d_{rj}d_{rk}q_{ii} - d_{ri}d_{rj}q_{ik} - d_{ri}d_{rk}q_{ij} + d_{ri}d_{ri}q_{jk}]_{j\neq i, k\neq i}.$$

3. An algorithm for checking copositivity. To settle the question of copositivity for given Q and Γ, several procedures have been devised, e.g., in [7], [12], [11], [8], or [4]. But to our knowledge, until now the following algorithm seems to be the only exact and finite one which performs both tasks in an easy and straightforward manner:

- check Γ-copositivity of Q by a routine, where the complexity of the calculations can be reduced in an adaptive, data-driven way;
- in case of a negative answer, generate a direction $d \in \Gamma$ satisfying $d^TQd < 0$.

The recursive structure described in the preceding section can be visualized by a tree, the root being the original Q and Γ, and the leaves corresponding to the one-dimensional problems where Γ is an interval of $\mathbb{R}$ and Q is a number, the sign of which is essential if $\Gamma \neq \{0\}$: Q is Γ-copositive if and only if $Q \geq 0$. The internal nodes are created during the recursion, and are labeled by the cones Γ_ν and functionals β_ν resulting from Table 1. If one of these cones is trivial, or if we have solved the one-dimensional problem at a leaf, we can proceed to traverse the tree as described below in the module TRAVERSE.

Procedure COPOS(node):

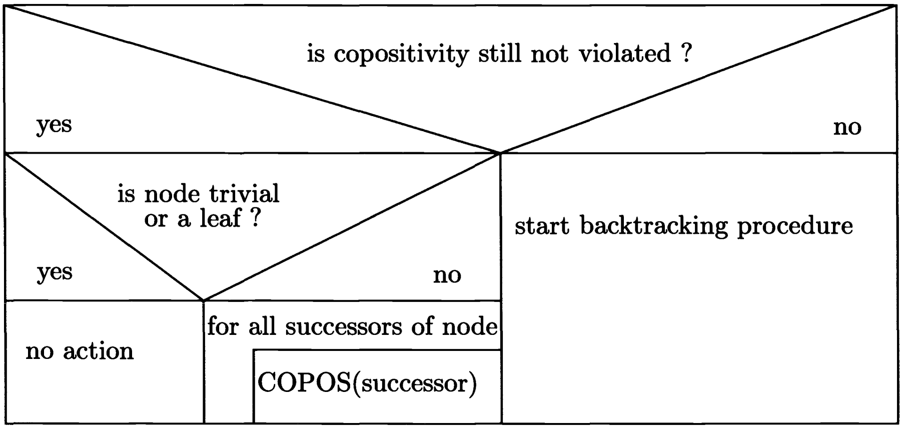

is copositivity still not violated ?
yes
no
is node trivial
or a leaf ?
yes
no
no action
for all successors of node
COPOS(successor)
start backtracking procedure

FIG. 1

For transparency, we depict the recursive structure of copositivity, checking in the structogram of the procedure COPOS (Fig. 1), a node being characterized by the quadruplet $(\Gamma_\nu, Q_\nu; i, \beta_\nu)$ (cf. Table 1 and Fig. 2 below). The procedure COPOS is of course started at the root corresponding to Γ and Q.

As can be seen in Fig. 1, we divide the whole algorithm into two parts: the "forward part" (steps 0–7 below) for recursive generation of nodes which we describe in detail, encapsulating some of the necessary routines in modular form as specified below; and a backtracking procedure (step 8) which, in case of a negative answer, uses the information attached to the nodes generated so far to return a direction $d \in \Gamma$ with $d^TQd < 0$.

- Module TRAVERSE: Starting from the current node, go to one of its neighbours (i.e., another successor of the predecessor); if there are none, return to the preceding level and investigate the neighbours of the predecessing node, and so on; if no nodes are left to investigate, stop: the Boolean variable cop keeps its initialized value true, and thus a positive answer is returned.

- Module TRIVIAL: To check whether or not $\Gamma = \{o\}$, we may use the following procedure.

First solve the linear program LP1 (with $x = x^+ - x^-, x^\pm \geq o$):

$$\begin{aligned} z_1 = \sum_{i=1}^n x_i^- &\to \max, \\ Dx^+ - Dx^- &\geq o, \\ \sum_{i=1}^n x_i^+ + \sum_{i=1}^n x_i^- &\leq 1, \\ x^+, x^- &\geq o. \end{aligned}$$

Since $[\frac{1}{2n}, \ldots, \frac{1}{2n}; \frac{1}{2n}, \ldots, \frac{1}{2n}]^T \in \mathbb{R}^{2n}$ is a feasible vector with objective value $\frac{1}{2}$, the optimal objective value z_1^* of LP1 is never less than $\frac{1}{2}$. If z_1^* exceeds $\frac{1}{2}$, the procedure stops, returning a vector $d^* = x^{*,+} - x^{*,-} \neq o$ belonging to Γ and so $\Gamma \neq \{o\}$. Otherwise, if $z_1^* = \frac{1}{2}$, we solve the linear program LP2:

$$\begin{aligned} z_2 = \sum_{i=1}^n x_i^+ &\to \max, \\ Dx^+ - Dx^- &\geq o, \\ \sum_{i=1}^n x_i^+ + \sum_{i=1}^n x_i^- &\leq 1, \\ x^+, x^- &\geq o. \end{aligned}$$

If the optimal value $z_2^* > \frac{1}{2}$, the procedure ends as above. If not, we conclude $\sum_{i=1}^n x_i^+ = \sum_{i=1}^n x_i^-$, and hence $\sum_{i=1}^n x_i = 0$, for all feasible $x \in \Gamma$. Choosing an arbitrary coordinate, e.g., $x_n = -\sum_{i=1}^{n-1} x_i$, we repeat the above procedure reduced by one dimension, replacing D with the transformed matrix $D_n = [d_{ij} - d_{in}]_{1 \leq i \leq m, 1 \leq j \leq n-1}$. In fact, we have obtained a dimensional reduction similar to that described in the previous section, but with only one successor cone: one then has also to replace Q with $Q_n = [q_{ij} - q_{in} - q_{nj} + q_{nn}]_{1 \leq i,j \leq n-1}$. Furthermore, we have to store $i = n$ and $\beta_\nu(y) = -\sum_{i=1}^{n-1} y_i$ for the backtracking procedure described below (see step 8).

- Module CHOOSE: To choose i, the number of the coordinate to be eliminated during dimensional reduction (cf. §2), observe that case (a) yields at most $3j_+j_-$ successor cones (where we denote the cardinality of $J_\times(i)$ by $j_\times$ for $\times = +, 0, -$), while cases (b) or (c) yield at most $2j_+$, or $2j_-$, respectively; finally, case (d) yields only one. So an optimal choice of i would first search for all i which satisfy case (d), and check $q_{ii} > 0$, or $q_{ii} = 0$ and $\pm p \in \Theta_0^*$ (this will be done by the following module CHECKSIGNS). If no i satisfying (d) exists, then proceed similarly to search for all i that yield cases (b) or (c), and then choose from among those an i with maximal j_0. Finally, if all i yield case (a), then choose an i such that j_+j_- is minimal and j_0 is maximal. Store the selected index i permanently, attaching it to the current node.

- Module CHECKSIGNS: this routine will be used only for the cases (b), (c), and (d). It first checks the sign of q_{ii}; and then, conditional on this result, performs one of the steps described below. For shortness, we only treat case (b). The other cases can be dealt with analogously:

* if $q_{ii} > 0$, nothing happens;

* if $q_{ii} < 0$, then $d = e_i \in \Gamma$ satisfies $d^TQd = q_{ii} < 0$ (here and in the sequel we denote by e_i the ith column of the identity matrix); put `cop = false`;

* if $q_{ii} = 0$, we check the relation $p \in \Theta_0^*$ by an application of the simplex method to

$$p^Ty \to \min, \qquad y \in \Theta_0,$$

which can be stopped whenever a point $y \in \Theta_0$ is generated with $p^Ty < 0$. In this case, put `cop = false` and construct from y, again, a direction $d \in \Gamma$ with $d^TQd < 0$:

$$d_j := \begin{cases} y_j, & \text{if } j \neq i, \\ \max\left[\left\{-\frac{y^TBy}{2p^Ty}\right\} \cup \{\alpha_r(y) : r \in J_+(i)\}\right] + 1, & \text{if } j = i. \end{cases}$$

Now it is possible to describe the main body of the algorithm (although steps 2 and 3 below would be carried out simultaneously for efficiency, we segregated them for the sake of transparency):

0. Initialize `cop = true`;
1. call TRIVIAL; if $\Gamma = \{o\}$ then call TRAVERSE and repeat this step;
2. else (Γ is nontrivial) call CHOOSE, thus selecting an index i, attach i to the current node, and check which of the cases (a), (b), (c), or (d) pertain;
3. if cases (b), (c), or (d) pertain, call CHECKSIGNS; if `cop = false`, then attach the obtained direction $d \in \Gamma$ with $d^TQd < 0$ to the current node and go to step 8; else (still `cop = true`)
4. determine a successor of the current node corresponding to (Γ_ν, β_ν), and also calculate Q_ν from Table 1;
5. if the order of Q_ν exceeds one, replace Q by Q_ν, Γ by Γ_ν, and go to step 1; else we are at leaf level:
6. call TRIVIAL for Γ_ν;
   6a. if $Q_\nu < 0$ and $\Gamma_\nu \neq \{o\}$, then determine whether $(-\infty, 0] \subseteq \Gamma_\nu$ or $\Gamma_\nu = [0, +\infty)$; then either $y = -1$ or $y = 1$ belongs to Γ_ν; in either case, attach y to the node (Γ_ν, β_ν), put `cop = false` and go to step 8, where y will be processed in the backtracking procedure;
   6b. else ($Q_\nu \geq 0$ or $\Gamma_\nu = \{o\}$) call TRAVERSE and go to step 4;
7. if `cop = true`, then return the positive answer "Q is Γ-copositive" and stop.
8. Perform the recursive backtracking procedure described below.

Remark. Since in every step of the recursion an optimal choice of i (in the sense of minimal number of possible successors of the current node) is performed, it seems that the proposed procedure has some advantages compared to that described in [11], which requires Slater's condition $D^{-1}(\operatorname{int} \mathbb{R}^m_+) \neq \emptyset$ to hold in addition. Even for detecting $\mathbb{R}^n_+$-copositivity, the algorithm described above might need less computational effort than the minorant and/or determinant criteria in [8] or [12]. Above all, the proposed procedure has the advantage that in case of a negative answer ("Q is not Γ-copositive"), a direction $d \in \Gamma$ with $d^TQd < 0$ is easily obtained by the following backtracking algorithm. To the best of our knowledge, no other copositivity procedure is able to produce such a direction.

In case of a negative stop where `cop = false`, we leave the forward part of the algorithm at a node $(\Gamma_\nu, \beta_\nu, Q_\nu)$ with a corresponding vector $y \in \Gamma_\nu$ with $y^TQ_\nu y < 0$

(note that in the above description of step 3, the quantities d, Γ and Q play the role of y, Γ_ν, and Q_ν, respectively). Now there is a (unique) path leading to the root, where all nodes of the path are labeled with the permanently stored indices i; all interior nodes are also labeled with the functionals β_ν. The backtracking step from a node with information y and β_ν to its predecessor node with information i can now be described very easily: it consists of the augmentation of y with the ith coordinate given by $\beta_\nu(y)$, i.e., of the transformation

$$y \mapsto x \quad \text{with } x_j := \begin{cases} y_j, & \text{if } j \neq i \quad \text{(i.e., if } y_j \text{ is defined),} \\ \beta_\nu(y), & \text{if } j = i. \end{cases}$$

In the notation of §2, the definition of β_ν in Table 1 implies that $x \in \Gamma$, provided $y \in \Gamma_\nu$, and also that

$$x^TQx = y^TQ_\nu y < 0\,.$$

Performing these transformations recursively, we thus arrive at the top level, i.e., at the root corresponding to the original problem. There we obtain the direction d with the desired properties by putting $d := x$.

Example 1. Let $n = 3$, $m = 5$, and consider

$$Q = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad D = \begin{bmatrix} -2 & 0 & 5 & 0 & -2 \\ -1 & -5 & 0 & 5 & 1 \\ 2 & 4 & 3 & 4 & 2 \end{bmatrix}^T .$$

First iteration:

1. TRIVIAL yields $\Gamma \neq \{o\}$;
2. CHOOSE yields $i = 3$ because of $J_-(3) = J_0(3) = \emptyset$; since $q_{33} > 0$, case (b1) pertains;
3. CHECKSIGNS does nothing and still `cop = true`; $p = o$, and hence $\lambda_0(y) = 0$;
4. according to Table 1, we start generating the successor cone determined by $\beta_1(y) = \alpha_1(y) = \max_j \alpha_j(y)$ and by $\alpha_1(y) \geq \lambda_0(y)$, obtaining (after removing redundant inequalities) $\Gamma_1 = \{y \in \mathbb{R}^2 : D_1 y \geq o\}$ with
$$D_1 = \begin{bmatrix} 4 & -3 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad Q_1 = F_1 = \begin{bmatrix} 0 & 2 \\ 2 & -3 \end{bmatrix};$$
5. since the order of Q_1 exceeds one, we arrive at the second iteration.

Second iteration:

1. TRIVIAL yields $\Gamma_1 \neq \{o\}$;
2. CHOOSE yields $i = 1$ because of $J_-(1) = \emptyset$; since $(Q_1)_{11} = 0$, case (b2) pertains;
3. since $\Theta_0 = [0, +\infty)$, CHECKSIGNS confirms that $p = 2 \in \Theta_0^*$ and hence still `cop = true`;
4. according to Table 1, we obtain $\Gamma_1' = \Theta_1 = \Theta_0$ and $\beta_1'(z) = \alpha_1'(z) = \frac{3}{4}z$, as well as $Q_1' = 0$;
5. now we are at leaf level; hence we proceed to the next step;
6. $\Gamma_1' \neq \{o\}$ and $Q_1' = 0$; since this current node has no neighbour (cf. Table 1), TRAVERSE yields the next successor cone of Γ, which is characterized by $\beta_1(y) = \lambda_0(y) \geq \alpha_1(y) = \max_j \alpha_j(y)$; since the current dimension now is two, we return to step 1, starting the third iteration.

Third iteration:

1. as is easy to see, the current cone is trivial; hence we further TRAVERSE the problem tree, generating the next successor (Γ_2, Q_2) of the root (Γ, Q), where Γ_2 is determined by $\beta_2(y) = \alpha_2(y) = \max_j \alpha_j(y)$ and by $\alpha_2(y) \geq \lambda_0(y)$, obtaining (after removing redundant inequalities) $\Gamma_2 = \{y \in \mathbb{R}^2 : D_2 y \geq o\}$ with
$$D_2 = \begin{bmatrix} -4 & 3 \\ 4 & 3 \end{bmatrix} \quad \text{and} \quad Q_2 = F_2 = \begin{bmatrix} -16 & 0 \\ 0 & 9 \end{bmatrix};$$
repeating step 1, we see $\Gamma_2 \neq \{o\}$, and hence proceed to the following;
2. CHOOSE yields $i = 2$, since $J_-(2) = J_0(2) = \emptyset$; because of $(Q_2)_{22} = 9 > 0$ we obtain case (b1) with four potential successors;
3. CHECKSIGNS does nothing and still `cop = true`; $p = 0$, and hence $\lambda_0(z) = 0$;
4. according to Table 1, we obtain $\Gamma_2' = \Theta_1 = [0, +\infty)$ and $\beta_2'(z) = \alpha_1''(z) = \frac{4}{3}z$, as well as $Q_2' = 144 > 0$;
5. now we are at leaf level; hence we proceed to the next step;

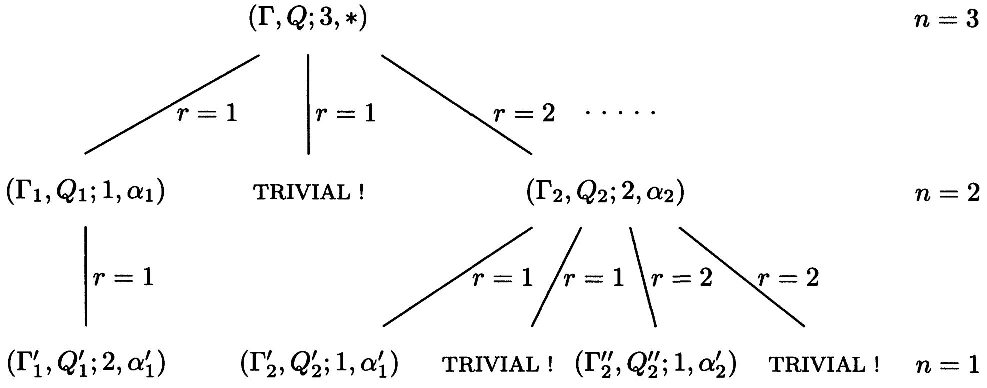

$(\Gamma, Q; 3, *)$
$n = 3$
$r = 1$
$r = 1$
$r = 2$
· · · · ·
$(\Gamma_1, Q_1; 1, \alpha_1)$
TRIVIAL !
$(\Gamma_2, Q_2; 2, \alpha_2)$
$n = 2$
$r = 1$
$r = 1$
$r = 1$
$r = 2$
$r = 2$
$(\Gamma_1', Q_1'; 2, \alpha_1')$
$(\Gamma_2', Q_2'; 1, \alpha_1')$
TRIVIAL !
$(\Gamma_2'', Q_2''; 1, \alpha_2')$
TRIVIAL !
$n = 1$

FIG. 2

6. $\Gamma_2' \neq \{o\}$; now case 6b. pertains so that we TRAVERSE to the neighbour cone characterized by $\beta_2'(z) = \lambda_0(z) \geq \alpha_1(z) \geq \alpha_2(z)$, cf. Table 1; step 5 now leads us again to step 6, which establishes triviality of this cone; hence again step 6b. is in force, TRAVERSE gives us the next neighbour $\Gamma_2'' = (-\infty, 0]$, $\alpha_2'(z) = -\frac{4}{3}z$, and $Q_2'' = 0$, we return via step 5 again to step 6b., which generates a new neighbouring cone characterized by $\beta_2'(z) = \lambda_0(z) \geq \alpha_2(z) \geq \alpha_1(z)$; after returning again via step 5 to step 6, TRIVIAL yields triviality of this last leaf-level cone, so that the next nontrivial cone generated by TRAVERSE leads us via step 5 again to step 1, where the next iteration begins, etc. See Fig. 2, where the nodes are labeled with $(\Gamma_\nu, Q_\nu; i, \beta_\nu)$, and irrelevant entries are symbolized by an asterisk *. Proceeding further, we obtain that Q is Γ-copositive (see Ex. 2 in [1]).

Example 2. Keep D from the previous example, but replace Q with
$$\tilde{Q} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Then the first iteration of the algorithm is exactly the same as in Example 1, with the exception that step 4 yields

$$\tilde{Q}_1 = \begin{bmatrix} -2 & 2 \\ 2 & -3 \end{bmatrix}$$

instead of Q_1; the second iteration now reads

1. $\Gamma_1 \neq \{o\}$;
2. CHOOSE yields $i = 1$ as in Example 1, and case (b) pertains;
3. since $(\tilde{Q}_1)_{11} = -2 < 0$, CHECKSIGNS puts `cop = false`, and $y = e_1 \in \Gamma_1$ satisfies $y^T\tilde{Q}_1 y = -2 < 0$; the next step is therefore
8. the backtracking procedure described above yields $d = x \in \Gamma$ with $x_1 = y_1 = 1$; $x_2 = y_2 = 0$; and $x_3 = \alpha_1(y) = 1$ (recall that the current node is attached with $\beta_\nu(y) = \alpha_1(y) = [1, \frac{1}{2}]y$, while its predecessor (the root) has the attached index $i = 3$ (cf. the upper part of the leftmost branch in Fig. 2)). Indeed, we have $d^T\tilde{Q}d = -1$.

Remark. The cone Γ is pointed, i.e., $\Gamma \cap -\Gamma = \{o\}$, if and only if the kernel $\{x \in \mathbb{R}^n : Dx = o\}$ of D is trivial. Then for no $i \in \{1, \dots, n\}$ and no stage of the recursion, case (d) can occur. For unpointed Γ, it may pay to obtain case (d) in the recursive procedure by performing a coordinate change, where the new basis $\{b_1, \dots, b_n\}$ is such that $\{b_1, \dots, b_k\}$ forms a basis of the kernel of D. Denote by S the matrix with columns b_j, $1 \leq j \leq n$. Then Q is Γ-copositive if and only if $Q' = S^TQS$ is Γ'-copositive, where $\Gamma' = S^{-1}(\Gamma) = \{z \in \mathbb{R}^n : DSz \geq o\}$. Now one may proceed as follows (cf. Theorem 8 in [2]): for the first k stages of recursion, select $i \in \{1, \dots, k\}$ as above; if a point $x' \in \Gamma'$ is obtained such that $(x')^TQ'x' < 0$, then stop: $x = Sx'$ satisfies $x \in \Gamma$ and $x^TQx < 0$. If, however, no negative stop occurs during these first k stages, the dimensionality of the problem is reduced to $n - k$ by a recursion procedure where the tree degenerates to a chain, so that there is only one resulting matrix, and only one resulting cone which is now pointed, and one can proceed as above, ignoring case (d).

4. Global optimization procedure.

4. Global optimization procedure. Suppose we are given a vertex $\bar{x} \in M$ of the feasible set (obtained, e.g., by Phase I of the simplex method). Then there are three possible cases:

(i) $\bar{x}$ fails to satisfy (1.14); or

(ii) $\bar{x}$ satisfies (1.14), but (1.11) is not fulfilled; or

(iii) both (1.11) and (1.14) are met, and hence $\bar{x}$ is a global solution to (1.2).

Property (1.14) can, again, be checked by examining the linear program

$$(Q\bar{x} + c)^Td \to \max, \qquad d \in \Gamma_0,$$

which either has optimal value zero, or is unbounded. In the former case (1.14) is satisfied, while in the latter also our original problem is unbounded: indeed, any direction $d \in \Gamma_0$ with $(Q\bar{x} + c)^Td > 0$ satisfies $\bar{x} + \lambda d \in M$ for all $\lambda \geq 0$, as well as

$$g(\bar{x} + \lambda d) - g(\bar{x}) = \frac{\lambda^2}{2} d^TQd + \lambda d^T(Q\bar{x} + c) \geq \lambda d^T(Q\bar{x} + c) \to \infty \quad \text{as } \lambda \to \infty.$$

There is another unboundedness condition which is independent of the current vertex $\bar{x}$: if $Q_0 = -Q$ is not Γ_0-copositive, i.e., if there is a direction $d \in \Gamma_0$ with $d^TQd > 0$, then as above, $\bar{x} + \lambda d \in M$ for all $\lambda \geq 0$, as well as

$$g(\bar{x} + \lambda d) - g(\bar{x}) = \frac{\lambda^2}{2} d^TQd + \lambda d^T(Q\bar{x} + c) \to \infty \quad \text{as } \lambda \to \infty\,.$$

Since this check has to be done only once, we shall incorporate it into the initialization step.

In case (ii), the copositivity algorithm proposed in the preceding section generates a direction $\overline{d} \in \Gamma_i$ with $\overline{d}^T Q_i \overline{d} < 0$ for some $i \in \{1, \ldots, n\} \setminus I$ (the case $i = 0$ has already been settled in the initialization step). This means $f_{\overline{d}}(\overline{\delta}) < 0$ for some $\overline{\delta} > 0$. A straightforward argument using (1.7) shows that we can choose $\overline{\delta}$ as follows:

$$(4.1) \qquad \overline{\delta} := \begin{cases} \max\left\{1, -\frac{\overline{d}^T(Q\overline{x}+c)}{\sqrt{\overline{d}^T Q \overline{d}}}\right\}, & \text{if } z(\overline{d}) = 0 \text{ (and hence } \overline{d}^T Q \overline{d} > 0\text{)}, \\ \frac{1}{2}\sqrt{\frac{\overline{d}^T(Q\overline{x}+c)}{z(\overline{d})}}, & \text{if } z(\overline{d}) > 0, \text{ but } \overline{d}^T Q \overline{d} = 0, \\ \frac{\sqrt{2\overline{d}^T Q \overline{d}}}{2z(\overline{d})}, & \text{otherwise.} \end{cases}$$

Now if $\overline{\varepsilon} = \overline{\delta}^2$, this means

$$(4.2) \qquad \overline{d}^T \overline{y} = \sigma_{S(\overline{\varepsilon})}(\overline{d}) > \sigma_{N(\overline{\varepsilon})}(\overline{d}),$$

where $\overline{y} \in S(\overline{\varepsilon}) = \partial_{\overline{\varepsilon}} g(\overline{x})$ is defined as in (1.5) for $\varepsilon = \overline{\varepsilon}$ and $d = \overline{d}$. But (4.2) implies that $\overline{y}$ does not belong to the normal cone $N(\overline{\varepsilon}) = N_{\overline{\varepsilon}}(M, \overline{x})$, whence there is a feasible point $\tilde{x} \in M$ such that

$$(4.3) \qquad \overline{y}^T(\tilde{x} - \overline{x}) > \overline{\varepsilon},$$

and therefore

$$(4.4) \qquad g(\tilde{x}) - g(\overline{x}) \geq \overline{y}^T(\tilde{x} - \overline{x}) - \overline{\varepsilon} > 0$$

results due to the definition of $\partial g_{\overline{\varepsilon}}(\overline{x})$. Note that any $x \in M$ satisfying (4.3) instead of $\tilde{x}$ also fulfills (4.4). If we thus solve the linear program

$$\overline{y}^T x \to \max, \qquad x \in M,$$

by the simplex method, we certainly arrive at some vertex v of M with the property $\overline{y}^T(v - \overline{x}) \geq \overline{y}^T(\tilde{x} - \overline{x}) > \overline{\varepsilon}$; consequently, this vertex yields a higher objective value $g(v) > g(\overline{x})$. We thus escaped from $\overline{x}$ even if the latter were a local solution (see Examples 3 and 4 below).

Let us now recapitulate the procedure (for the sake of lucidity, we do not distinguish between vertices of M and those of the feasible set in standard form, i.e., including slack variables).

0. Generate a feasible vertex $\overline{x}$ of $M = \{x \in \mathbb{R}^n : Ax \leq b\}$ (if there is none, stop: the problem is infeasible). Check whether $Q_0 = -Q$ is Γ_0-copositive, where $\Gamma_0 = \{d \in \mathbb{R}^n : Ad \leq o\}$; if the answer is negative, stop: the problem is unbounded.
1. If the linear program

$$(4.5) \qquad (Q\overline{x} + c)^T d \to \max, \qquad d \in \Gamma_0,$$

is unbounded, stop: the problem is unbounded; otherwise (1.14) is satisfied.

2. Determine $I = \{i : (A\bar{x})_i = b_i\}$, the tangential cone $\Gamma = \{d \in \mathbb{R}^n : (Ad)_i \leq 0 \text{ for all } i \in I\}$, and calculate the slack variables $u_i = b_i - (A\bar{x})_i$ for $i \notin I$. For all $i \in \{1, \dots, n\} \setminus I$, check whether or not Q_i is Γ_i-copositive, where Γ_i is given by (1.9) and Q_i is calculated from (1.12) and (1.13).
3. If for all $i \in \{1, \dots, n\} \setminus I$ the matrix Q_i is Γ_i-copositive, stop: $\bar{x}$ is a global solution of the problem (1.2). Else there is some $i \in \{1, \dots, n\} \setminus I$ such that a direction $\bar{d} \in \Gamma_i$ is generated with $\bar{d}^T Q_i \bar{d} < 0$; then calculate $\bar{\varepsilon} = \bar{\delta}^2$ from (4.1) and define $\bar{y} \in S(\bar{\varepsilon})$ as in (1.5). Solve the linear problem

$$\bar{y}^T x \to \max, \qquad x \in M, \tag{4.6}$$

by the simplex method starting at the vertex $\bar{x}$. Along the path of vertices v generated by this procedure, record their objective values $g(v)$ and pick that vertex, say $\bar{v}_i$, with the largest one. Replace $\bar{x}$ with $\bar{v}_i$ and go to step 1 (one may also repeat this step for all i such that Q_i is not Γ_i-copositive; then define $\bar{v}$ to be the vertex satisfying $g(\bar{v}) = \max_i g(\bar{v}_i)$, and replace $\bar{x}$ with $\bar{v}$).

5. Examples and conclusion.

Example 3. Consider again problem (1.1). Here $Q = 2I_n$, where I_n denotes the $n \times n$-identity matrix, $A = \left[\begin{smallmatrix} I_n \\ -I_n \end{smallmatrix}\right]$, and $b = [1, \dots, 1]^T \in \mathbb{R}^{2n}$. Also, we assume $0 < c_j < 1$ for all $j \in \{1, \dots, n\}$.

First observe that due to definition (1.10) we have $\Gamma_0 = \{o\}$. Let us start with a vertex $\bar{x}$ satisfying $\bar{x}_i = -1$. Then $i \notin I$, and

$$\Gamma_i = \{d \in \mathbb{R}^n : d_i \geq d_k \geq 0 \text{ if } \bar{x}_k = -1,\ d_i \geq -d_k \geq 0 \text{ if } \bar{x}_k = 1\},$$

so that $\bar{d} = e_i \in \Gamma_i$ satisfies

$$\bar{d}^T Q_i \bar{d} = -2c_i < 0$$

(this direction $\bar{d}$ would also be generated by the algorithm described in §3). Now $z(\bar{d}) = \frac{1}{2}$; $\bar{d}^T Q \bar{d} = 2$; $\bar{\delta} = 2$; $\bar{\varepsilon} = 4$; and hence $\bar{y} = 2\bar{x} + c + 4e_i$. The corresponding linear program

$$\bar{y}^T x = \sum_{j \neq i} (2\bar{x}_j + c_j)x_j + (c_i + 2)x_i \to \max, \quad -1 \leq x_i \leq 1, \quad 1 \leq i \leq n,$$

has the solution $\bar{v}_i = \bar{x} + 2e_i$, which is obtained by pivoting once from the starting point $\bar{x}$. Thus the global solution x^* satisfying $x_i^* = 1$ for all i is reached after $k_-(\bar{x}) \leq n$ steps, where $k_-(\bar{x})$ denotes the number of negative coordinates of $\bar{x}$.

Example 4. Consider a problem of the form (1.2) with

$$Q = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad A = \begin{bmatrix} 2 & 1 \\ -4 & 1 \\ -3 & -1 \\ 3 & -2 \end{bmatrix}, \quad \text{and} \quad b = \begin{bmatrix} 4 \\ 4 \\ 3 \\ 6 \end{bmatrix}.$$

The feasible polyhedron M is depicted in Fig. 3.

1.0. Starting at the vertex $\bar{x} = \left[\begin{smallmatrix} 0 \\ -3 \end{smallmatrix}\right]$, which clearly is a local solution, we proceed as in the algorithm described above. Since $\Gamma_0 = \{o\}$, $Q_0 = -Q$ is Γ_0-copositive.

1.1. Calculating $Q\bar{x} + c = \left[\begin{smallmatrix} 0 \\ -6 \end{smallmatrix}\right]$, we solve the linear program $-6d_2 \to \max$, such that $d \in \Gamma_0 = \{o\}$, giving us the finite solution $d = o$. So condition (1.14) is satisfied.

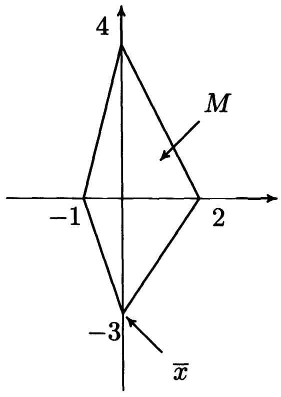


FIG. 3

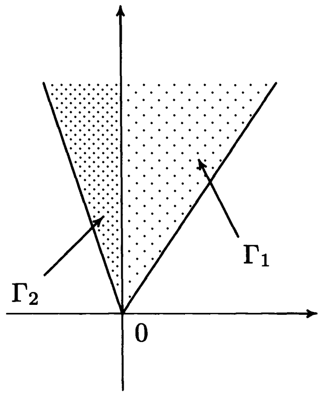


FIG. 4

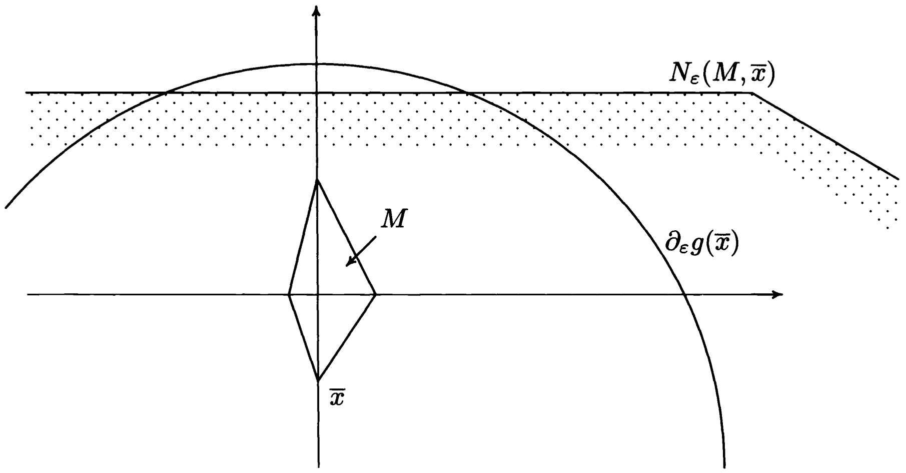


FIG. 5

1.2. Determine the index set $I(\bar{x}) = \{3,4\}$ and the tangential cone $\Gamma(\bar{x}) = \{d \in \mathbb{R}^2 : d_2 \geq -3d_1; d_2 \geq \frac{3}{2}d_1\}$. The slack variables have the value $u_1 = u_2 = 7$, the new subcones are of the following form: $\Gamma_1 = \{d \in \Gamma(\bar{x}) : d_1 \geq 0; \quad d_2 \geq -2d_1\} = \{d \in \mathbb{R}^2 : d_1 \geq 0; \quad d_2 \geq \frac{3}{2}d_1\}$, and $\Gamma_2 = \{d \in \Gamma(\bar{x}) : d_1 \leq 0; \quad d_2 \geq 4d_1\} = \{d \in \mathbb{R}^2 : d_1 \leq 0; \quad d_2 \geq -3d_1\}$ (see Fig. 4).

The corresponding matrices are

$$Q_1 = \begin{bmatrix} -14 & 12 \\ 12 & -2 \end{bmatrix} \quad \text{and} \quad Q_2 = \begin{bmatrix} -14 & -24 \\ -24 & -2 \end{bmatrix}.$$

Summarizing (1.1) and (1.2), we obtain the following: The vector $\bar{d} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ is an extremal ray of both Γ_i and satisfies $\bar{d}^T Q_i \bar{d} = -2 < 0$. Note that this direction $\bar{d}$ would result also from an application of the copositivity procedure described in §3. Hence the vertex $\bar{x}$ is not a global optimum of the problem. This can also be verified directly by criterion (1.3), for example, with $\varepsilon = 49$ (cf. step 1.3 below). As one can easily see, $\partial_\varepsilon g(\bar{x})$ is not included in the set $N_\varepsilon(M; \bar{x})$, and so condition (1.3) is not fulfilled (see Fig. 5).

In the next step let us determine an improving direction.

1.3. From (4.1) we calculate $\bar{\varepsilon} = \bar{\delta}^2 = 2\bar{d}^T Q\bar{d}/4z^2(\bar{d}) = 49$, where $\bar{d} = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$ and $\bar{d}^T Q\bar{d} = 2$ with $z(\bar{d}) = \max[\{0\} \cup \{\frac{1}{7}, \frac{1}{7}\}] = \frac{1}{7}$, according to (1.6). Using (1.5), we obtain

$$\bar{y} = Q\bar{x} + c + \sqrt{\frac{2\bar{\varepsilon}}{\bar{d}^T Q\bar{d}}}Q\bar{d} = \begin{bmatrix}0\\8\end{bmatrix}.$$

Then we solve the linear program (4.6):

$$8x_2 \longrightarrow \max, \qquad x \in M,$$

to obtain the next candidate $x^* = \left[\begin{smallmatrix}0\\4\end{smallmatrix}\right]$, the opposite vertex of M. Returning to our algorithm (step 1), we now check if x^* is a global solution.

2.1. We calculate $Qx^* + c = \left[\begin{smallmatrix}0\\8\end{smallmatrix}\right]$ and the corresponding linear program (4.5) again has the finite solution $d = o$.

2.2. The new index set is $I(x^*) = \{1, 2\}$ and the tangential cone $\Gamma(x^*) = \{d \in \mathbb{R}^2 : d_2 \leq -2d_1;\ \ d_2 \leq 4d_1\}$. The slack variables have the values $u_3 = 7$ and $u_4 = 14$. The new subcones are of the following form: $\Gamma_3 = \{d \in \Gamma(x^*) : d_1 \leq 0;\ \ d_2 \leq -3d_1\} = \{d \in \mathbb{R}^2 : d_1 \leq 0;\ \ d_2 \leq 4d_1\}$, and $\Gamma_4 = \{d \in \Gamma(x^*) : d_1 \geq 0, d_2 \leq \frac{3}{2}d_1\} = \{d \in \mathbb{R}^2 : d_1 \geq 0;\ \ d_2 \leq -2d_1\}$ (see Fig. 6; compare to Fig. 3).

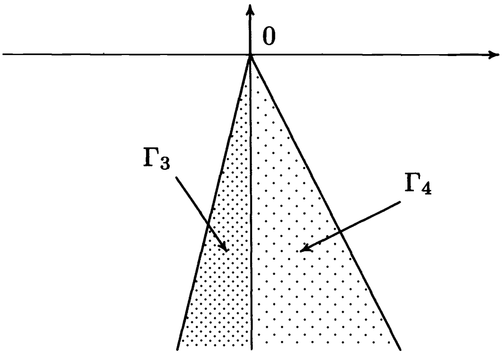


FIG. 6

The corresponding matrices are

$$Q_3 = \begin{bmatrix}-14 & 24\\ 24 & 2\end{bmatrix} \quad \text{and} \quad Q_4 = \begin{bmatrix}-14 & -24\\ -24 & 18\end{bmatrix}.$$

2.3. Since Q_3 is Γ_3-copositive and Q_4 is Γ_4-copositive, as is easy to show, the vertex x^* is the global solution of our original problem. Note that all vertices of the polyhedron M are local solutions. Nevertheless, the algorithm described above does not go from one vertex to an adjacent one. Instead, an improving direction is found, which enables us to skip inefficient adjacent vertices and go directly to the global solution.

As mentioned already in the introduction, problem (1.2) is NP-hard from the worst-case complexity point of view. However, the approach of [10] shows that there is no essential difference between the complexities of checking local versus global optimality [6], despite the fears expressed by [13].

Concerning the proposed procedure, its frequent use of the simplex algorithm, together with data-driven optimal selection in order to reduce recursional complexity in the copositivity algorithm, suggests the hope that in the average case (cf. [3]) the computational costs can be held within reasonable limits. Although the final release of the implementation is not yet finished, numerical experiments yield quite encouraging results.

Acknowledegments. The authors want to thank K. A. Fröschl, W. Gutjahr, M. Hamala, and P. Meravý for stimulating discussion and valuable suggestions, as well as J.-B. Hiriart-Urruty for his kind support.

REFERENCES

[1] I. M. BOMZE, *Remarks on the recursive structure of copositivity*, J. Inform. Optim. Sci., 8 (1987), pp. 243–260.

[2] ———, *Detecting all evolutionarily stable strategies*, J. Optim. Theory Appl., 75 (1992), pp. 313–329.

[3] K. H. BORGWARDT, *The Simplex Method–A Probabilistic Analysis*, Springer-Verlag, Berlin, 1987.

[4] G. DANNINGER, *A recursive algorithm for determining* (*strict*) *copositivity of a symmetric matrix*, in Methods of Operations Research, Vol. 62, U. Rieder, P. Gessner, A. Peyerimhoff, and F. J. Radermacher, eds., Hain, Meisenheim, 1990, pp. 45–52.

[5] G. DANNINGER, *The role of copositivity in optimality criteria for non-convex optimization problems*, J. Optim. Theory Appl., 75 (1992), pp. 535–558.

[6] G. DANNINGER AND I. M. BOMZE, *Using copositivity for global optimality criteria in concave quadratic programming problems*, Math Programming, Ser. A., to appear.

[7] P. H. DIANANDA, *On non-negative forms in real variables some or all of which are non-negative*, Proc. Cambridge Philos. Soc., 58 (1967), pp. 17–25.

[8] K. P. HADELER, *On copositive matrices*, Linear Algebra Appl., 49 (1983), pp. 79–89.

[9] J.-B. HIRIART-URRUTY, *From convex optimization to nonconvex optimization, Part* I: *Necessary and sufficient conditions for global optimality*, in Nonsmooth Optimization and Related Topics, F. H. Clarke, V. F. Demyanov, and F. Gianessi, eds., Plenum Press, New York, 1989, pp. 219–239.

[10] J.-B. HIRIART-URRUTY AND C. LEMARÉCHAL, *Testing necessary and sufficient conditions for global optimality in the problem of maximizing a convex quadratic function over a convex polyhedron*, Preliminary report, Seminar of Numerical Analysis, University Paul Sabatier, Toulouse, 1990.

[11] D. H. MARTIN, *Finite criteria for conditional definiteness of quadratic forms*, Linear Algebra Appl., 39 (1981), pp. 9–21.

[12] T. S. MOTZKIN, *Signs of minors*, in Inequalities, Vol. 1., O. Shisha, ed., Academic Press, New York, 1967, pp. 225–240.

[13] K. G. MURTY AND S. N. KABADI, *Some* NP-*complete problems in quadratic and linear programming*, Math. Programming, 39 (1987), pp. 117–129.

[14] P. M. PARDALOS, *Polynomial time algorithms for some classes of constrained nonconvex quadratic problems*, Optimization, 21 (1990), pp. 843–853.

[15] P. M. PARDALOS AND G. SCHNITGER, *Checking local optimality in constrained quadratic programming is* NP-*hard*, Oper. Res. Lett., 7 (1988), pp. 33–35.

SIAM J. OPTIMIZATION
Vol. 3, No. 4, pp. 843–860, November 1993

A QUADRATICALLY CONVERGENT POLYNOMIAL ALGORITHM FOR SOLVING ENTROPY OPTIMIZATION PROBLEMS*

FLORIAN POTRA[†] AND YINYU YE[†]

Abstract. A potential reduction algorithm is developed for solving entropy optimization problems. It is shown that the algorithm generates an ϵ-optimal solution within at most $O(\sqrt{n}|\log \epsilon|)$ iterations, where, as usual, n is the number of nonnegative variables, and each iteration solves a system of linear equations. Under a computable criterion, the algorithm is tuned to the pure Newton method in a manner that leads to quadratic convergence while maintaining primal feasibility at each step. A stopping criterion is derived which ensures that the objective function approaches its optimal value within any prescribed tolerance. This applies for all entropy optimization problems having interior optimal solutions.

Key words. entropy function, potential reduction algorithm, polynomiality, quadratic convergence

AMS subject classifications. 65K05, 90C25

1. Introduction. In this paper, we consider the separable convex nonlinear optimization problem with linear constraints:

$$\text{(EP)} \qquad \begin{array}{ll} \text{minimize} & f(x) = \sum_{j=1}^{n} f_j(x_j), \\ \text{subject to} & x \in \Omega_p = \{x \in R^n : Ax = b, x \geq 0\}. \end{array}$$

Here $A \in R^{m \times n}, b \in R^m$, and superscript ^{T} denotes the transpose operation. We assume that f_j is a real function $f_j : (0, \infty) \to R$ for all j, and

$$\text{(A1)} \qquad \Omega_p \text{ has a nonempty interior.}$$

A dual problem of (EP) is

$$\text{(ED)} \qquad \begin{array}{ll} \text{maximize} & g(x,y) = b^T y - (x^T \nabla f(x) - f(x)), \\ \text{subject to} & (x,y) \in \Omega_d = \{(x,y) : x \in \Omega_p, \nabla f(x) - A^T y \geq 0\}. \end{array}$$

As usual, we use s to denote the slack vector $\nabla f(x) - A^T y$. We further assume that

$$\text{(A2)} \qquad \Omega_d \text{ has a nonempty interior.}$$

Based on the two assumptions and the Karush–Kuhn–Tucker conditions, x^* is an optimal solution if and only if the following three optimality conditions hold.

1. Primal feasibility: $x^* \in \Omega_p$;
2. Dual feasibility: There exists y^*, such that $(x^*, y^*) \in \Omega_d$;
3. Complementary slackness: $X^*(\nabla f(x^*) - A^T y^*) = 0$.

Here, the upper-case letter X designates the diagonal matrix whose entries are the components of the vector x, e is the vector of all ones, and $\|\cdot\|$ (without subscript) denotes the l_2 norm.

We call the problem an entropy optimization problem if for all j, f_j is an entropy function defined as

$$\text{(1.0)} \qquad f_j''(x_j) = \mu_j(x_j^{d_j} + q_j)$$

* Received by the editors December 7, 1990; accepted for publication (in revised form) July 10, 1992.

[†] University of Iowa, Iowa City, Iowa 52242. The second author's research was supported in part by National Science Foundation grant DDM-8922636 and the Iowa Business School Summer grant.

for some fixed numbers $\mu_j \geq 0, q_j \geq 0$, and $d_j \in R$. This includes the real functions $x^2, x\log(x), -\sqrt{x}, \log(x), 1/x$, etc.

The entropy optimization problem is one of the most popular convex nonlinear programs. Its applications include system equilibrium [3], image reconstruction [1], [5], and transportation distribution [17], among many others. For the entropy optimization problem, computational complexity must be treated in a slightly different manner than for linear programming (LP), and such notions as polynomiality are meaningful in an extended sense only. At each step of a primal-dual interior-point method for solving (EP), a pair $(x^k, y^k) \in \mathrm{Int}(\Omega)$ is produced and the algorithm is terminated when

$$\delta_k < \epsilon, \tag{1.1}$$

where δ_k is equal to the primal-dual gap $(x^k)^T s^k$ $(s^k = \nabla f(x^k) - A^T y^k)$, or to some other bound on the distance of $f(x^k)$ to the optimum. If

$$\lim_{k\to\infty} \delta_k = 0, \tag{1.2}$$

then, by definition, there is an integer $K(\epsilon)$ such that (1.1) is satisfied for all $k \geq K(\epsilon)$. So far, the best complexity results for interior-point methods have been obtained by proving that it is possible to take $K(\epsilon) = O(\sqrt{n}|\log\epsilon|)$.

For LP with integer data it is known that if (1.1) is satisfied with $\epsilon = 2^{-L}$, where L is the length of a binary coding of the data, then by a rounding procedure requiring $O(n^3)$ arithmetic operations it is possible to obtain an exact solution. Such an algorithm is commonly called an $O(\sqrt{n}L)$-iteration algorithm. Each iteration of the LP interior-point algorithm requires $O(n^3)$ arithmetic operations, so that the overall complexity is $O(n^{3.5}L)$. By some special update procedure this can be reduced to $O(n^3L)$, which is the best complexity for LP known so far. An algorithm is called *polynomial* if it requires at most $O(n^tL^q)$ (where t and q are positive integers) arithmetic operations to find an exact solution.

For nonlinear programming the notion of polynomiality described above does not make sense because, in general, we cannot obtain an exact solution of the problem using a finite number of arithmetic operations. However, if there is $p > 0$ and $q > 0$ such that for any $\epsilon > 0$ there is an integer

$$K(\epsilon) = O(n^p|\log\epsilon|^q) \tag{1.3}$$

such that (1.1) is satisfied for all $k \geq K(\epsilon)$, then we say that we have an $O(n^p|\log\epsilon|^q)$-iteration algorithm. Moreover, if each iteration of the algorithm requires $O(n^r), r > 0$, arithmetic operations, then we say that we have a polynomial algorithm.

Several iterative algorithms for the entropy optimization problem have been developed in the past (see, e.g., [20]). Although performing well in practice, they are not polynomial in the sense described above. In the area of interior-point algorithms, this problem has been considered by Ye in [18], where he reported some encouraging numerical results. However, he gave no theoretical complexity result. Several researchers (den Hertog, Roos, and Terlaky [4], Jarre [6], Monteiro and Adler [11], and Nesterov and Nemirovsky [12]) developed sufficient conditions to analyze some convex programs solvable with polynomial complexity. Clearly, our entropy functions satisfy conditions (a) and (c) of Monteiro and Adler. But some of them, including some popular entropy functions like x log (x) and $-\sqrt{x}$, do not satisfy their condition (b).

The main difficulty for nonlinear optimization is that each iteration of interior-point algorithms faces a system of nonlinear equations rather than linear equations as in the linear or quadratic case. Recently, Kortanek, Potra, and Ye [9] analyzed a conceptual algorithm for solving general linearly constrained convex programs. They proposed two schemes: a potential reduction scheme and a path-following one. More importantly, they established a bound for the residual of the system of nonlinear equations such that the polynomiality of linear programming is retained. More recently, Zhu [24] showed that for a class of optimization problems, which contains the entropy optimization problem, one Newton step will achieve such an accuracy if the so-called centering condition is enforced. Therefore, the path-following algorithm [7] can solve these problems with polynomial complexity.

The goal of this paper is twofold. First, we will develop a polynomial potential reduction algorithm for entropy optimization problems. As we know, the linear convergence ratio for the path-following algorithm is bounded from above and below so that the effective speed of convergence is of the same order as the guaranteed theoretical rate. This limits large improvement in every iteration. On the other hand, the potential reduction algorithm is a function-driven algorithm. We can use a line search procedure at each step to obtain additional improvement without destroying its global convergence. This feature is especially useful if the optimization problem is nonlinear, where a line search is commonly used.

Second, while maintaining polynomiality we would also like to develop an algorithm with fast local convergence. As we mentioned before, for linear and quadratic programming with integral data, after the distance ϵ to the optimal value becomes small enough, one can always terminate the algorithm by using a rounding procedure to obtain an exact solution in finite time. However, this finite termination does not work for nonlinear optimization. Therefore, fast local convergence is especially important in nonlinear optimization.

An interior-point algorithm with local superlinear convergence for some optimization problems has been developed by Coleman and Li [2]. Unfortunately, their algorithm does not have polynomiality. The local convergence behavior of some interior-point algorithms has been studied by Zhang, Tapia, and Dennis [22] and Zhang, Tapia, and Potra [23], where several conditions are developed to characterize local superlinear convergence. Zhang and Tapia [21] recently developed an algorithm for linear programming with polynomiality and local quadratic convergence under the assumption of nondegeneracy, or polynomiality and superlinear convergence under the assumption of the convergence of the iteration sequence.

Motivated by all of these results, we develop an interior-point algorithm for solving the entropy optimization problem. First, it is a function-driven polynomial algorithm. The algorithm iteratively generates a feasible pair (x^k, s^k) with

$$\frac{(x^k)^T s^k}{(x^0)^T s^0} \leq \hat{\epsilon}$$

in $O(\sqrt{n}|\log \hat{\epsilon}|)$ iterations, where (x^0, s^0) is an initial feasible pair. The search direction is, as usual, a combination of a centering direction and a descent direction. Second, we develop a computable criterion under which the pure Newton method (without centering) can be applied for the rest of the iterative process, thus giving local quadratic convergence. That is, the additional accuracy ϵ can be obtained in $O(\log|\log(\epsilon)|)$ Newton steps. We show that this faster local convergence is guaranteed for all entropy optimization problems possessing interior optimal solutions, e.g., $x^2, x\log(x), -\sqrt{x}, \log(x), 1/x$, etc.

2. A polynomial potential reduction algorithm. For simplicity and without loss of generality, assume that

$$f_j''(x_j) = x_j^d \quad \text{for all } j.$$

One can easily verify that our results hold for the general entropy function defined in (1.0). Let $|h_j/x_j| \le \alpha < 1$ for some $h_j \in R$ and $x_j > 0$. Then for some $0 < t < 1$

$$\begin{aligned} |x_j(f_j'(x_j+h_j) - f_j'(x_j) - f_j''(x_j)h_j)| &= |\tfrac{1}{2}x_j f_j'''(x_j+th_j)h_j^2| \\ &= \tfrac{1}{2}|x_j d(x_j+th_j)^{d-1}h_j^2| \\ &= \tfrac{1}{2}|d||x_j^d(1+th_j/x_j)^{d-1}h_j^2| \\ &\le \tfrac{1}{2}\bar{d}(\alpha)h_j f_j''(x_j)h_j \end{aligned}$$

and

$$\begin{aligned} |x_j(f_j''(x_j+h_j) - f_j''(x_j))x_j| &= |x_j f_j'''(x_j+th_j)h_j x_j| \\ &= |x_j d(x_j+th_j)^{d-1}h_j x_j| \\ &= |d||x_j^d(1+th_j/x_j)^{d-1}x_j^2||h_j/x_j| \\ &\le \alpha\bar{d}(\alpha)x_j f_j''(x_j)x_j, \end{aligned}$$

where

$$\bar{d}(\alpha) = \max\{1, |d|\max((1+\alpha)^{d-1}, (1-\alpha)^{d-1})\}.$$

Note that for any $0 < \beta \le \alpha < 1, \bar{d}(\beta)$ is a monotone decreasing function, bounded from above by $\bar{d}(\alpha)$ and below by 1. In the following we simply use $\bar{d}$ to denote $\bar{d}(\alpha)$, and choose α such that

$$\alpha\bar{d}(\alpha) = \tfrac{1}{2}.$$

The above relations imply that for $h \in R^n, x > 0 \in R^n$, and $\|X^{-1}h\|_\infty \le \alpha$, we have

$$\|X(\nabla f(x+h) - \nabla f(x) - \nabla^2 f(x)h)\| \le \tfrac{1}{2}\bar{d}h^T\nabla^2 f(x)h \tag{2.0}$$

and

$$(1-\alpha\bar{d})\|X\nabla^2 f(x)X\| \le \|X\nabla^2 f(x+h)X\| \le (1+\alpha\bar{d})\|X\nabla^2 f(x)X\|. \tag{2.1}$$

If both $X\nabla^2 f(x+h)X$ and $X\nabla^2 f(x)X$ are invertible, we also have

$$\begin{aligned} \left(\frac{1}{1+\alpha\bar{d}}\right)\|(X\nabla^2 f(x)X)^{-1}\| &\le \|(X\nabla^2 f(x+h)X)^{-1}\| \\ &\le \left(\frac{1}{1-\alpha\bar{d}}\right)\|(X\nabla^2 f(x)X)^{-1}\|. \end{aligned} \tag{2.2}$$

Now consider the primal-dual potential function [16], [19]

$$\phi(x,s) = \rho\log(x^Ts) - \sum_{j=1}^n \log(x_j s_j),$$

where $\rho \ge n + \sqrt{n}$. This potential function can also be written as

$$\phi(x,s) = (\rho - n)\log(x^Ts) - \sum_{j=1}^n \log\left(\frac{x_j s_j}{x^Ts}\right).$$

The inequality between the geometric and arithmetic means yields

$$-\sum_{j=1}^{n} \log\left(\frac{x_j s_j}{x^T s}\right) \geq n \log n.$$

Hence

$$(\rho - n)\log(x^T s) \leq \phi(x, s) - n \log n.$$

This inequality tells the exact amount, $-(\rho - n)|\log \epsilon|$, by which ϕ should be reduced to obtain

$$x^T s \leq \epsilon.$$

Given $0 < x^k \in \Omega_p$ and $s^k = \nabla f(x^k) - A^T y^k > 0$, we solve according to [9] the following system of nonlinear equations for Δx and Δy:

$$X^k \Delta s + S^k \Delta x = \theta\left(\frac{(x^k)^T s^k}{\rho} e - X^k S^k e\right) = \theta p^k, \tag{2.3a}$$

$$A\Delta x = 0 \quad \text{and} \quad \Delta s = \nabla f(x^k + \Delta x) - \nabla f(x^k) - A^T \Delta y, \tag{2.3b}$$

where

$$p^k = \frac{(x^k)^T s^k}{\rho} e - X^k S^k e.$$

Let

$$x^{k+1} = x^k + \Delta x, \quad y^{k+1} = y^k + \Delta y, \quad \text{and} \quad s^{k+1} = \nabla f(x^{k+1}) - A^T y^{k+1}.$$

Then, choosing

$$\theta = \frac{\beta \min_{1 \leq j \leq n}\left(\sqrt{x_j^k s_j^k}\right)}{\|(X^k S^k)^{-1/2} p^k\|} \tag{2.4}$$

for some $0 < \beta < 1$, we have

$$\phi(x^{k+1}, s^{k+1}) \leq \phi(x^k, s^k) - \gamma$$

for a constant $\gamma > 0$.

It turns out that system (2.3a) does not need to be solved exactly. If the norm of the residual term

$$\|z^k\| = \|X^k \Delta s + S^k \Delta x - \theta p^k\| \leq \zeta \beta \min(x_j^k s_j^k)$$

for some $0 < \zeta = O(\beta) < 1$, then the potential function will still be reduced by a constant for a suitable constant β. More precisely, we have the following proposition.

PROPOSITION 1. *Let $n + \sqrt{n} \leq \rho \leq 2n$ and let Δx and Δy satisfy* (2.3b) *together with*

$$X^k \Delta s + S^k \Delta x = \theta\left(\frac{(x^k)^T s^k}{\rho} e - X^k S^k e\right) + z^k = \theta p^k + z^k,$$

where θ is defined in (2.4). *If*

$$\|z^k\| \le C\beta^2 \min(x_j^k s_j^k)$$

for some constant C, then by choosing $\beta > 0$ such that

$$\beta(1+C\beta) \le \tfrac{1}{2} \quad \textit{and} \quad 1 - C\beta \ge 0,$$

we have

$$\phi(x^{k+1}, s^{k+1}) \le \phi(x^k, s^k) - \gamma$$

where

$$\gamma = (-\sqrt{3}/2)\beta(1-C\beta) + \beta^2(1+C\beta)^2.$$

The proof of the above proposition is almost identical to the proof of Theorem 3.2 of [9] and is presented in the Appendix. In Potra and Ye [15] a similar result is proved for more general nonlinear complementarity problems.

In this paper we show that the above condition on the residual can be achieved by applying one Newton step, i.e., by solving the system of linear equations

$$X^k(\nabla^2 f(x^k)\Delta x - A^T\Delta y) + S^k\Delta x = \theta p^k, \tag{2.5a}$$

$$A\Delta x = 0. \tag{2.5b}$$

The following lemma is a direct result from Kojima, Mizuno, and Yoshise [8] or Pardalos, Ye, and Han [13] for convex quadratic programming.

LEMMA 1. *Let Δx and Δy be the solution of system* (2.5). *Then*

$$\|(X^k)^{-1}\Delta x\| \le \beta.$$

Now we have the following lemma.

LEMMA 2. *Let Δx and Δy be the solution of system* (2.5). *Then*

$$\|z^k\| \le \frac{\overline{d}\beta^2 \min(x_j^k s_j^k)}{8}.$$

Proof. It can be verified that

$$z^k = X^k(\nabla f(x^k + \Delta x) - \nabla f(x^k) - \nabla^2 f(x^k)\Delta x).$$

Thus from (2.0),

$$\|z^k\| \le \tfrac{1}{2}\overline{d}\Delta x^T \nabla^2 f(x^k)\Delta x.$$

Let $D = (X^k)^{-1/2}(S^k)^{1/2}$. Then, from (2.4) and (2.5), we have

$$\begin{aligned}\Delta x^T \nabla^2 f(x^k)\Delta x &= \Delta x^T D(X^k S^k)^{-1/2}\theta p^k - \|D\Delta x\|^2 \\ &\le \|\theta(X^k S^k)^{-1/2} p^k\| \, \|D\Delta x\| - \|D\Delta x\|^2 \\ &\le \|\theta(X^k S^k)^{-1/2} p^k\|^2/4 \\ &= \theta^2 \|(X^k S^k)^{-1/2} p^k\|^2/4 = \beta^2 \min(x_j^k s_j^k)/4.\end{aligned}$$

Thus

$$\|z^k\| \leq \frac{\bar{d}\beta^2 \min(x_j^k s_j^k)}{8}. \qquad \square$$

The procedure can be stated as follows.

PROCEDURE 1.

```
Given $Ax^0 = b, x^0 > 0$ and $s^0 = \nabla f(x^0) - A^T y^0 > 0$;
set $k = 0$;
    while $(x^k)^T s^k \geq \epsilon$ do
        begin
            compute $\Delta x$ and $\Delta y$ of (2.5);
            $x^{k+1} = x^k + \Delta x$;
            $y^{k+1} = y^k + \Delta y$;
            $s^{k+1} = \nabla f(x^{k+1}) - A^T y^{k+1}$;
            $\delta_{k+1} = (x^{k+1})^T s^{k+1}$;
            $k = k + 1$;
        end.
```

THEOREM 1. *Let $n+\sqrt{n} \leq \rho \leq 2n$. Then $x^k > 0$ and $s^k > 0$, and they are feasible for* (EP) *and* (ED). *Moreover, Procedure* 1 *terminates in at most $O(\phi(x^0, s^0) + (\rho - n)|\log \epsilon|)$ iterations.*

Later we will show how to generate an initial feasible point (x^0, s^0) such that $\phi(x^0, s^0)$ is bounded by $(\rho - n)|\log(x^0)^T s^0|$. In practice, a step size can be selected based on the line search

$$\bar{\eta} = \arg\min_{\eta \geq 0} \phi(x_k + \eta\Delta x, s^k + \eta\Delta s).$$

Then we set

$$x^{k+1} = x^k + \bar{\eta}\Delta x \quad \text{and} \quad y^{k+1} + \bar{\eta}\Delta y.$$

Also, ρ can be chosen as an integer greater than $O(n)$. For quadratic programming, Pardalos, Ye, and Han [13] found that $\rho \in (n^{1.5}, n^2)$ seems to give the best numerical performance.

3. Local quadratic convergence. In this section we develop a computable criterion under which the pure Newton method can be applied so that quadratic convergence for the entropy optimization problem is guaranteed. At some $\bar{k}$, suppose we have

$$\|(X^{\bar{k}}\nabla^2 f(x^{\bar{k}})X^{\bar{k}})^{-1}\| \, \|X^{\bar{k}}s^{\bar{k}}\| < 1. \tag{3.0}$$

For simplicity and without loss of generality assume that $\bar{k} = 0$. We now move to the pure Newton method for $\nabla f(x) - A^T y = 0, Ax = b$, by solving the linear system

$$\nabla^2 f(x^0)\Delta x - A^T\Delta y = -s^0 \quad \text{and} \quad A\Delta x = 0, \tag{3.1}$$

and by letting $x^1 = x^0 + \Delta x, y^1 = y^0 + \Delta y$, and $s^1 = \nabla f(x^1) - A^T y^1$. We now show the following quadratic convergence theorem.

THEOREM 2. *Let* $\alpha = 1/2\bar{d} \le \frac{1}{2}$, *and define*

$$\hat{d} = \bar{d}/((1-\alpha)^2(1-\alpha\bar{d})).$$

Then, for any $\beta \le \alpha$, *the inequality*

$$\hat{d}\|(X^0\nabla^2 f(x^0)X^0)^{-1}\|\,\|X^0 s^0\| \le \beta \tag{3.2}$$

implies

$$x^1 > 0, \qquad \|(X^0)^{-1}(x^1 - x^0)\| \le \beta$$

and

$$\hat{d}\|(X^1\nabla^2 f(x^1)X^1)^{-1}\|\,\|X^1 s^1\| \le \beta^2. \tag{3.3}$$

Before proving Theorem 2, we introduce the following lemma.

LEMMA 3. *Let* Δx *and* Δy *be the solution of system* (3.1) *and let*

$$\hat{d}\|(X^0\nabla^2 f(x^0)X^0)^{-1}\|\,\|X^0 s^0\| \le \beta.$$

Then

$$\|(X^0)^{-1}\Delta x\| \le \beta$$

and

$$\Delta x^T\nabla^2 f(x^0)\Delta x \le \|(X^0\nabla^2 f(x^0)X^0)^{-1}\|\,\|X^0 s^0\|^2.$$

Proof. Noting that $\hat{d} \ge \bar{d} \ge 1$, from condition (3.2) we have

$$\|(X^0\nabla^2 f(x^0)X^0)^{-1}\|\,\|X^0 s^0\| \le \beta.$$

From system (3.1),

$$\Delta x^T\nabla^2 f(x^0)\Delta x = -\Delta x^T s^0.$$

Hence

$$\Delta x^T\nabla^2 f(x^0)\Delta x = \|\Delta x^T s^0\| = \|\Delta x^T (X^0)^{-1}X^0 s^0\| \le \|(X^0)^{-1}\Delta x\|\,\|X^0 s^0\|.$$

Thus

$$\begin{aligned}\|(X^0)^{-1}\Delta x\|^2 &= \|(X^0\nabla^2 f(x^0)X^0)^{-1/2}(X^0\nabla^2 f(x^0)X^0)^{1/2}(X^0)^{-1}\Delta x\|^2\\ &\le \|(X^0\nabla^2 f(x^0)X^0)^{-1}\|\,\|(X^0\nabla^2 f(x^0)X^0)^{1/2}(X^0)^{-1}\Delta x\|^2\\ &= \|(X^0\nabla^2 f(x^0)X^0)^{-1}\|\Delta x^T\nabla^2 f(x^0)\Delta x\\ &\le \|(X^0\nabla^2 f(x^0)X^0)^{-1}\|\,\|(X^0)^{-1}\Delta x\|\,\|X^0 s^0\|.\end{aligned}$$

Therefore,

$$\|(X^0)^{-1}\Delta x\| \le \|(X^0\nabla^2 f(x^0)x^0)^{-1}\|\,\|X^0 s^0\| \le \beta,$$

and furthermore

$$\Delta x^T\nabla^2 f(x^0)\Delta x \le \|(X^0)^{-1}\Delta x\|\,\|X^0 s^0\| \le \|(X^0\nabla^2 f(x^0)X^0)^{-1}\|\|X^0 s^0\|^2. \qquad \square$$

Proof of Theorem 2. Using (2.0) and Lemma 3, we can write successively

$$\begin{aligned}\|X^1s^1\| &= \|(X^0+\Delta X)(\nabla f(x^0+\Delta x) - A^T(y^0+\Delta y))\| \\ &= \|(X^0+\Delta X)(\nabla f(x^0+\Delta x) - \nabla f(x^0) - \nabla^2 f(x^0)\Delta x)\| \\ &= \|(I+(X^0)^{-1}\Delta X)X^0(\nabla f(x^0+\Delta x) - \nabla f(x^0) - \nabla^2 f(x^0)\Delta x)\| \\ &\le \|I+(X^0)^{-1}\Delta X\|\,\|X^0(\nabla f(x^0+\Delta x) - \nabla f(x^0) - \nabla^2 f(x^0)\Delta x)\| \\ &\le (1+\beta)\|X^0(\nabla f(x^0+\Delta x) - \nabla f(x^0) - \nabla^2 f(x^0)\Delta x)\| \\ &\le (1+\beta)\tfrac{1}{2}\bar{d}(\Delta x^T\nabla^2 f(x^0)\Delta x) \\ &\le \bar{d}\|(X^0\nabla^2 f(x^0)X^0)^{-1}\|\,\|X^0s^0\|^2.\end{aligned}$$

On the other hand, from Lemma 3 and (2.2),

$$\begin{aligned}\|(X^1\nabla^2 f(x^1)X^1)^{-1}\| &= \|(X^1)^{-1}X^0(X^0\nabla^2 f(x^1)X^0)^{-1}X^0(X^1)^{-1}\| \\ &\le \|(X^0\nabla^2 f(x^1)X^0)^{-1}\|\,\|X^0(X^1)^{-1}\|^2 \\ &\le \frac{1}{(1-\alpha)^2(1-\alpha\bar{d})}\|(X^0\nabla^2 f(x^0)X^0)^{-1}\|.\end{aligned} \tag{3.4}$$

Thus, we have

$$\begin{aligned}\hat{d}\|(X^1\nabla^2 f(x^1)X^1)^{-1}\|\,\|X^1s^1\| &\le \hat{d}\frac{\bar{d}}{(1-\alpha)(1-\alpha\bar{d})}\|(X^0\nabla^2 f(x^0)X^0)^{-1}\|^2\|X^0s^0\|^2 \\ &\le \hat{d}^2\|(X^0\nabla^2 f(x^0)X^0)^{-1}\|^2\|X^0s^0\|^2 \\ &\le \beta^2. \qquad \square\end{aligned}$$

Now we describe the Newton procedure.

PROCEDURE 2.

```
Given $Ax^0 = b, x^0 > 0$ and $s^0 = \nabla f(x^0) - A^Ty^0 > 0$;
let $x^0$ and $s^0$ satisfy (3.2);
set $k = 0$;
    while $\|(X^k\nabla^2 f(x^k)X^k)^{-1}\|\,\|X^ks^k\| \ge \epsilon$
    do
        begin
            compute $\Delta x$ and $\Delta y$ in (3.1) at $x^k$ and $s^k$;
            $x^{k+1} = x^k + \Delta x$;
            $y^{k+1} = y^k + \Delta y$;
            $s^{k+1} = \nabla f(x^{k+1}) - A^Ty^{k+1}$;
            $k = k + 1$;
        end.
```

Since $\hat{d}$ depends only on d and α, Procedure 2 generates a sequence of $x^k > 0(Ax^k = b)$ and s^k such that $\|(X^k\nabla^2 f(x^k)X^k)^{-1}\|\,\|X^ks^k\|$ converges to zero quadratically. In the case $d = -1$, we have

$$\|s^k\| = \|(X^k)^{-1}X^ks^k\| \le \|(X^k)^{-1}\|\,\|X^ks^k\| = \|(X^k\nabla^2 f(x^k)X^k)^{-1}\|\,\|X^ks^k\|.$$

This implies that $\|s^k\|$ also converges to zero quadratically. We will see that this is true for arbitrary d.

Now let us elaborate a bit more on condition (3.2). Note that in our case

$$\|(X\nabla^2 f(x)X)^{-1}\| = \frac{1}{\min(X\nabla^2 f(x)x)} = \frac{1}{\min(X^{d+2}e)}.$$

The condition can be further written as

$$\|X^k(\nabla f(x^k) - A^T y^k)\| \leq \gamma \min(X^{d+2}e)$$

for some $\gamma < 1$. We note that for $d \neq -1$

$$\nabla f(x^k) = \frac{1}{d+1}(X^k)^{d+1}e + c$$

for a fixed vector c. Then

$$\begin{aligned} \|X^k(\nabla f(x^k) - A^T y^k)\| &= \left\| \frac{1}{d+1}(X^k)^{d+2}e + X^k(c - A^T y^k) \right\| \\ &\leq \gamma \min(X^{d+2}e). \end{aligned} \tag{3.5}$$

Thus, for $d < -1$ and γ small enough,

$$X^k(c - A^T y^k) > 0$$

so that

$$(c - A^T y^k) > 0.$$

In fact, if $d = -2$, then condition (3.5) becomes

$$\| -e + X^k(c - A^T y^k)\| \leq \gamma,$$

which is precisely the so-called centering condition for LP when $\sum_j$ log (x_j) is used as the barrier function. Therefore, the condition can be viewed as a "centering" condition for LP when $\sum_j f_j(x)$ with $d < -1$ is used as the barrier function. In these cases, the Newton method will generate a sequence of x^k and $c^T - A^T y^k$ such that

$$Ax^k = b, \quad x^k > 0, \quad \text{and} \quad c^T - A^T y^k > 0.$$

In general, we can prove the following lemma for Procedure 2.

LEMMA 4. *Let $\{x^k\}$ be the sequence generated by Procedure 2. Then, for all k,*

$$\|(X^k)^{-1}X^0\| \leq 3 \quad \textit{and} \quad \|(X^0)^{-1}X^k\| \leq 3.$$

Proof. Note that from Theorem 2 we have

$$1 - \beta^{2^k} \leq \|(X^k)^{-1}X^{k+1}\| \leq 1 + \beta^{2^k}$$

and

$$1/(1 + \beta^{2^k}) \leq \|(X^{k+1})^{-1}X^k\| \leq 1/(1 - \beta^{2^k}).$$

Now using the inequality

$$\log(1 - \zeta) \geq -\zeta - \frac{\zeta^2}{2(1 - \zeta)}$$

for $0 \le \zeta < 1$, and noting $\beta^2 \le \frac{1}{4}$, we have

$$\begin{aligned}\sum_{t=1}^{\infty} \log(1-\beta^{2^t}) &\ge \sum_{t=1}^{\infty}\left(-\beta^{2^t} - \frac{2}{3}\beta^{2^{t+1}}\right)\\ &\ge -\beta^2 - \frac{5}{3}\sum_{t=1}^{\infty}(\beta^4)^t\\ &= -\beta^2 - \frac{5}{3}\frac{\beta^4}{1-\beta^4}\\ &\ge -\frac{1}{4} - \left(\frac{5}{3}\right)\left(\frac{1}{15}\right) = -\frac{13}{36}.\end{aligned}$$

Hence

$$\prod_{t=0}^{\infty}(1-\beta^{2^t}) \ge (1-\beta)\exp\left(-\frac{13}{36}\right) \ge \frac{1}{2}\exp\left(-\frac{13}{36}\right) > \frac{1}{3}.$$

Thus

$$\begin{aligned}\|(X^k)^{-1}X^0\| &= \|(X^k)^{-1}X^{k-1}(X^{k-1})^{-1}\cdots X^1(X^1)^{-1}X^0\|\\ &= \|(X^k)^{-1}X^{k-1}\|\cdots\|(X^1)^{-1}X^0\|\\ &\le \prod_{t=0}^{k-1}(1/(1-\beta^{2^t})) \le 3.\end{aligned}$$

Also

$$\begin{aligned}\|X^k(X^0)^{-1}\| &= \|X^k(X^{k-1})^{-1}X^{k-1}\cdots(X^1)^{-1}X^1(X^0)^{-1}\|\\ &= \|X^k(X^{k-1})^{-1}\|\cdots\|X^1(X^0)^{-1}\|\\ &\le \prod_{t=0}^{k-1}(1+\beta^{2^t})\\ &\le \prod_{t=0}^{k-1}(1/(1-\beta^{2^t})) \le 3.\end{aligned}$$

Therefore, we have the desired result. □

Lemma 4 indicates that Procedure 2 generates points x^k that are relatively close to the starting point x^0. Now we are ready to prove our main result.

THEOREM 3. *Procedure* 2 *generates a sequence of feasible points that converges quadratically to the optimal solution.*

Proof. From Lemma 4 and Theorem 2,

$$\begin{aligned}\|s^k\| &= \|(X^k)^{-1}X^k s^k\|\\ &= \|\nabla^2 f(x^k)X^k(X^k\nabla^2 f(x^k)X^k)^{-1}X^k s^k\|\\ &\le \|\nabla^2 f(x^k)X^k\|\,\|(X^k\nabla^2 f(x^k)X^k)^{-1}\|\,\|X^k s^k\|\\ &= \|(X^k)^{d+1}\|\,\|(X^k\nabla^2 f(x^k)X^k)^{-1}\|\,\|X^k s^k\|\\ &\le C\|(X^0)^{d+1}\|\,\|(X^k\nabla^2 f(x^k)X^k)^{-1}\|\,\|X^k s^k\|\\ &\le C\|(X^0)^{d+1}\|\beta^{2^k}\end{aligned}$$

for some constant $C = 3^{|d|+1}/\hat{d}$. Similarly, we can derive

$$\|X^k s^k\| \le 3C\|(X^0)^{d+2}\|\beta^{2^k} = 3C\|X^0\nabla^2 f(x^0)X^0\|\beta^{2^k}. \tag{3.6}$$

By letting $k \to \infty$, we see that $\|X^k s^k\|$ converges to zero quadratically, while x^k remains as a positive feasible point due to Lemma 4. Since $\|(X^k)^{-1}\|$ is bounded from above, we also see that $\|s^k\|$ converges to zero quadratically.

On the other hand, from Lemma 4 and Theorem 2

$$\begin{aligned}\|x^{k+1} - x^k\| &= \|X^k(X^k)^{-1}(x^{k+1} - x^k)\| \\ &\le \|X^k\|\,\|(X^k)^{-1}(x^{k+1} - x^k)\| \\ &\le \|X^k\|\beta^{2^k} \\ &\le \|X^0\|\,\|(X^0)^{-1}X^k\|\beta^{2^k} \\ &\le 3\|X^0\|\beta^{2^k},\end{aligned}$$

which indicates that $\{x^k\}$ is a Cauchy sequence and therefore it must converge to the optimal solution x^*. Since

$$\begin{aligned}\|x^k - x^*\| &\le \sum_{j=0}^{\infty}\|x^{k+j} - x^{k+j+1}\| \\ &\le 3\|X^0\|\sum_{j=0}^{\infty}\beta^{2^{k+j}} \\ &\le 3\|X^0\|\sum_{j=0}^{\infty}(\beta^{2^k}\beta^j) \\ &= 3\|X^0\|\beta^{2^k}\sum_{j=0}^{\infty}\beta^j \\ &\le 6\|X^0\|\beta^{2^k},\end{aligned}$$

x^k converges to x^* quadratically. □

Essentially, we have proved the R-quadratic convergence of $\{s^k\}, \{X^k s^k\}$, and $\{x^k\}$. With a little more effort one can prove that these sequences are Q-quadratically convergent. For the difference between R-order and Q-order of convergence, see Potra [14].

4. Further complexity analysis. Let the entropy optimization problem have the interior optimal solution x^* and let the Hessian $\nabla^2 f(x)$ be positive definite in the interior of the feasibility region. Now we provide a bound on the number of iterations needed by Procedure 1 to satisfy condition (3.2).

LEMMA 5. *Let $\hat{x}$ and $\hat{s}$ be generated from Procedure* 1, *and let*

$$\hat{x}^T\hat{s} < \delta^2 \le \frac{(\alpha^2/2)(1 - \alpha\bar{d})}{\|(X^*\nabla^2 f(x^*)X^*)^{-1}\|}.$$

Then

$$\|(X^*)^{-1}(\hat{x} - x^*)\| < \alpha.$$

Proof. The proof is by contradiction. From the primal-dual theory, we must have

$$f(\hat{x}) - f(x^*) < \frac{(\alpha^2/2)(1-\alpha\bar{d})}{\|(X^*\nabla^2 f(x^*)X^*)^{-1}\|}.$$

To obtain a contradiction let us assume that

$$\|(X^*)^{-1}(\hat{x} - x^*)\| \geq \alpha.$$

Now consider the minimization problem:

$$\begin{array}{ll} \text{minimize} & f(x), \\ \text{subject to} & Ax = b, \quad x > 0, \quad \text{and} \quad \|(X^*)^{-1}(x - x^*)\| \geq \alpha. \end{array}$$

Since the function is convex and $x^* > 0$ is outside of the feasibility region, the minimal solution $\overline{x}$ of the above problem must be on the boundary of the ellipsoid constraint. In other words,

$$\|(X^*)^{-1}(\overline{x} - x^*)\| = \alpha.$$

However, using $\nabla f(x^*)(\overline{x} - x^*) = 0$ and (2.1), we have

$$\begin{aligned} f(\hat{x}) - f(x^*) &\geq f(\overline{x}) - f(x^*) \\ &= \int_0^1 \nabla f(x^* + t(\overline{x} - x^*))^T(\overline{x} - x^*)dt \\ &= \int_0^1 (\nabla f(x^* + t(\overline{x} - x^*)) - \nabla f(x^*))^T(\overline{x} - x^*)dt \\ &= \int_0^1 \int_0^1 t(\overline{x} - x^*)^T \nabla^2 f(x^* + ts(\overline{x} - x^*))(\overline{x} - x^*)dt\,ds \\ &\geq (1 - \alpha\bar{d}) \int_0^1 \int_0^1 t(\overline{x} - x^*)^T \nabla^2 f(x^*)(\overline{x} - x^*)dt\,ds \\ &\geq \frac{1}{2}(1 - \alpha\bar{d})(\overline{x} - x^*)^T \nabla^2 f(x^*)(\overline{x} - x^*) \\ &= \frac{1}{2}(1 - \alpha\bar{d})(\overline{x} - x^*)^T (X^*)^{-1} X^* \nabla^2 f(x^*) X^* (X^*)^{-1}(\overline{x} - x^*) \\ &\geq \frac{1}{2}(1 - \alpha\bar{d})\|(X^*)^{-1}(\overline{x} - x^*)\|^2 / \|(X^*\nabla^2 f(x^*)X^*)^{-1}\| \\ &= \frac{\alpha^2}{2}(1 - \alpha\bar{d}) / \|(X^*\nabla^2 f(x^*)X^*)^{-1}\|, \end{aligned}$$

which is a contradiction. □

Based on Lemma 5 and (3.4), we have

$$\|(X^k\nabla^2 f(x^k)X^k)^{-1}\| \leq \frac{1}{(1-\alpha)^2(1-\alpha\bar{d})}\|(X^*\nabla^2 f(x^*)X^*)^{-1}\|.$$

Thus

$$\|(X^*\nabla^2 f(x^*)X^*)^{-1}\|\,\|X^k s^k\| \leq \|(X^*\nabla^2 f(x^*)X^*)^{-1}\|(x^k)^T s^k \leq \delta^2$$

implies that

$$\|(X^k\nabla^2 f(x^k)X^k)^{-1}\|\;\|X^k s^k\| \le \delta^2/((1-\alpha)^2(1-\alpha\bar{d})),$$

which gives condition (3.2) for some constant δ. Therefore, we can terminate Procedure 1 when

$$(x^k)^T s^k \le \delta^2/\|(X^*\nabla^2 f(x^*)X^*)^{-1}\|. \tag{4.0}$$

Then we rename (x^k, s^k) as (x^0, s^0) and start Procedure 2.

From Lemma 5, we have

$$1-\alpha \le \|(X^*)^{-1}X^0\| \le 1+\alpha \tag{4.1a}$$

or

$$1/(1+\alpha) \le \|(X^0)^{-1}X^*\| \le 1/(1-\alpha). \tag{4.1b}$$

Since x^k is feasible, $f(x)$ is convex, and $\alpha \le \frac{1}{2}$, from Lemma 4, Theorem 3, (2.1), (3.5), and (4.1),

$$\begin{aligned}
f(x^k) - f(x^*) \le \nabla f(x^k)^T(x^k - x^*) &= (s^k)^T(x^k - x^*)\\
&= (s^k)^T X^k (X^k)^{-1}(x^k - x^*)\\
&\le \|(X^k)^{-1}(x^k - x^*)\|\;\|X^k s^k\|\\
&= \|e - (X^k)^{-1}x^*\|\;\|X^k s^k\|\\
&\le (\sqrt{n} + \|(X^k)^{-1}x^*\|)\|X^k s^k\|\\
&= (\sqrt{n} + \|(X^k)^{-1}X^0(X^0)^{-1}X^* e\|)\|X^k s^k\|\\
&\le (\sqrt{n} + \|(X^k)^{-1}X^0\|\;\|(X^0)^{-1}X^*\|\;\|e\|)\|X^k s^k\|\\
&\le (\sqrt{n} + 3\sqrt{n}/(1-\alpha))\|X^k s^k\|\\
&\le 7\sqrt{n}\|X^k s^k\|\\
&\le 7\sqrt{n}C\|X^0\nabla^2 f(x^0)X^0\|\beta^{2^{k-1}}\\
&\le 7\sqrt{n}\overline{C}\|X^*\nabla^2 f(x^*)X^*\|\beta^{2^{k-1}}
\end{aligned}$$

for some constant $\overline{C}$. This indicates that the accuracy $f(x) - f(x^*) < \epsilon$ can be obtained in log $(\log(\sqrt{n}\|X^*\nabla^2 f(x^*)X^*\|/\epsilon))$ Newton steps.

Overall, we have the following complexity result.

THEOREM 4. *Let the entropy optimization problem have the interior optimal solution x^* and the Hessian $\nabla^2 f(x)$ be positive definite in the interior of the feasibility region. Then Procedure* 1 *generates x and s satisfying* (4.0) *in at most*

$$O(\phi(x^0, s^0) + \sqrt{n}(1 + |\log\|(X^*\nabla^2 f(x^*)X^*)^{-1}\|\,|))$$

iterations. Then Procedure 2 *can be applied to generate a feasible point x with*

$$f(x) - f(x^*) \le \epsilon \tag{4.2}$$

in at most $\log(\log(\sqrt{n}\|X^*\nabla^2 f(x^*)X^*\|/\epsilon))$ *iterations.*

In particular, if $d = 2$, then we have at most $O(\phi(x^0, s^0) + \sqrt{n})$ iterations for Procedure 1 and $\log(\log(\sqrt{n}/\epsilon))$ iterations for Procedure 2.

Our algorithm can start from any interior points x^0 and y^0 without jeopardizing its convergence for Procedure 1. From a theoretical point of view, the generation of an initial solution whose primal-dual potential function value is polynomially bounded is the same as for linear or quadratic programming. For example, we can generate the approximate analytic center x^0 for Ω_p, i.e.,

$$Ax^0 = b, \quad x^0 > 0, \quad \text{and} \quad \|X^0(-A^T\overline{y}) - e\| \leq 0.1$$

for some $\overline{y}$. Then we can choose some $\zeta > 0$ such that

$$\|X^0(\nabla f(x^0) - A^T y^0) - ((x^0)^T s^0/n)e\| \leq .5(x^0)^T s^0/n,$$

where $s^0 = \nabla f(x^0) - A^T y^0$ and $y^0 = \zeta\overline{y}$. Now the initial potential value

$$\phi(x^0, s^0) \leq \sqrt{n}\log(x^0)^T s^0 + 1.$$

In practice, a combined Phase I and Phase II approach has been developed for linear programming (see, e.g., [10]). It can be also used for the entropy optimization problem.

5. Final remark. In this paper we have restricted ourselves to entropy optimization problems. However, all the proofs carry over for large classes of problems. The polynomiality results of §2 can be proved for all functions f satisfying condition (2.0), which is the so-called scaled Lipschitz condition in [24]. In general, this condition can be written as: for $h \in R^n, x > 0 \in R^n$, and $\|X^{-1}h\|_\infty \leq \alpha$,

$$\|X(\nabla f(x+h) - \nabla f(x) - \nabla^2 f(x)h)\| \leq \psi(\alpha)h^T\nabla^2 f(x)h,$$

where $\psi(\alpha)$ is a monotone increasing function $\psi : (0,1) \to (0,\infty)$.

Appendix. Here we restate and prove Proposition 1 which is used in §2.

PROPOSITION 1. *Let $n + \sqrt{n} \leq \rho \leq 2n$ and let Δx and Δy satisfy*

$$X^k\Delta s + S^k\Delta x = \theta\left(\frac{(x^k)^T s^k}{\rho}e - X^k S^k e\right) + z^k = \theta p^k + z^k,$$

$$A\Delta x = 0 \quad \textit{and} \quad \Delta s = \nabla f(x^k + \Delta x) - \nabla f(x^k) - A^T\Delta y,$$

where

$$\theta = \frac{\beta \min_{1\leq j\leq n}\left(\sqrt{x_j^k s_j^k}\right)}{\|(X^k S^k)^{-1/2}p^k\|}$$

and

$$\|z^k\| \leq C\beta^2 \min(x_j^k s_j^k) \tag{A.0}$$

for some constant C. Let

$$x^{k+1} = x^k + \Delta x, \quad y^{k+1} = y^k + \Delta y \quad \textit{and} \quad s^{k+1} = s^k + \Delta s = \nabla f(x^{k+1}) - A^T y^{k+1}.$$

Then choosing $\beta > 0$ such that

$$\beta(1 + C\beta) \leq \tfrac{1}{2} \quad \textit{and} \quad 1 - C\beta \geq 0, \tag{A.1}$$

we have

$$\phi(x^{k+1}, s^{k+1}) \leq \phi(x^k, s^k) - \gamma,$$

where

$$\gamma = (-\sqrt{3}/2)\beta(1 - C\beta) + \beta^2(1 + C\beta)^2.$$

Proof. To simplify formulae, we drop the index k, and let $x^+ = x + \Delta x, y^+ = y + \Delta y$, and $s^+ = s + \Delta s = \nabla f(x^+) - A^T y^+$. Let $D = (XS)^{1/2}$. Note that

$$\begin{aligned} \|D^{-1}\| &= 1/\min(\sqrt{x_j s_j}), \\ \theta &= \frac{\beta}{\|D^{-1}\|\,\|D^{-1}p\|}, \end{aligned} \tag{A.2}$$

and

$$D^{-1}p = \frac{x^T s}{\rho} D^{-1}e - De. \tag{A.3}$$

The following standard result is frequently used in interior-point algorithms (see, e.g., [18]). If

$$\max(\|X^{-1}\Delta x\|_\infty, \|S^{-1}\Delta s\|_\infty) \le \tfrac{1}{2}, \tag{A.4}$$

then

$$\begin{aligned} \phi(x^+, s^+) - \phi(x, s) \le{}& \frac{\rho}{x^T s}(x^T \Delta s + s^T \Delta x + \Delta x^T \Delta s) \\ &- e^T(X^{-1}\Delta x + S^{-1}\Delta s) + \delta^2, \end{aligned} \tag{A.5}$$

where

$$\delta^2 = \|X^{-1}\Delta x\|^2 + \|S^{-1}\Delta s\|^2.$$

First, we have

$$X\Delta s + S\Delta x = \theta p + z = \theta(p + q), \tag{A.6}$$

where $q = \frac{z}{\theta}$. From (A.0) and (A.2)

$$\|D^{-1}q\| \le \|D^{-1}\|\,\|q\| = \frac{\|D^{-1}\|}{\theta}\|z\| \le \frac{\|D^{-1}\|}{\theta}\frac{C\beta^2}{\|D^{-1}\|^2} = C\beta\|D^{-1}p\|. \tag{A.7}$$

Thus

$$\begin{aligned} \delta^2 &= \|D^{-2}X\Delta s\|^2 + \|D^{-2}S\Delta x\|^2 \\ &\le \|D^{-1}\|^2(\|D^{-1}X\Delta s\|^2 + \|D^{-1}S\Delta x\|^2) \\ &= \|D^{-1}\|^2(\|D^{-1}(X\Delta s + S\Delta x)\|^2 - 2\Delta x^T \Delta s) \\ &= \|D^{-1}\|^2(\theta^2\|D^{-1}(p + q)\|^2 - 2\Delta x^T \Delta s) \quad \text{from (A.6)} \\ &\le \|D^{-1}\|^2(\theta^2(\|D^{-1}p\| + \|D^{-1}q\|)^2 - 2\Delta x^T \Delta s) \\ &\le \|D^{-1}\|^2(\theta^2\|D^{-1}p\|^2(1 + C\beta)^2 - 2\Delta x^T \Delta s) \quad \text{from (A.7)} \\ &= \theta^2\|D^{-1}\|^2\|D^{-1}p\|^2(1 + C\beta)^2 - 2\|D^{-1}\|^2\Delta x^T \Delta s \\ &= \beta^2(1 + C\beta)^2 - 2\|D^{-1}\|^2\Delta x^T \Delta s \quad \text{from (A.2).} \end{aligned} \tag{A.8}$$

The convexity of $f(\,.\,)$ implies

$$\Delta x^T \Delta s \ge 0.$$

Therefore, from (A.1) and (A.8), we have

$$\delta \le \beta(1 + C\beta) \le \tfrac{1}{2}.$$

This also implies that the condition (A.4) holds for inequality (A.5).

Now we have

$$
\begin{aligned}
&\frac{\rho}{x^T s}(x^T \Delta s + s^T \Delta x) - e^T(X^{-1}\Delta x + S^{-1}\Delta s) \\
&= \frac{\rho}{x^T s} e^T \left((X\Delta s + S\Delta x) - \frac{x^T s}{\rho} D^{-2}(X\Delta s + S\Delta x) \right) \\
&= \frac{\rho}{x^T s} e^T \left(D - \frac{x^T s}{\rho} D^{-1} \right) D^{-1}(X\Delta s + S\Delta x) \\
&= -\frac{\rho}{x^T s} p^T D^{-1} D^{-1}(X\Delta s + S\Delta x) \quad \text{from (A.3)} \\
&= -\frac{\rho\theta}{x^T s} p^T D^{-1} D^{-1}(p + q) \quad \text{from (A.6)} \\
\text{(A.9)} \qquad &= -\frac{\rho\theta}{x^T s}(\|D^{-1}p\|^2 + p^T D^{-1} D^{-1} q) \\
&\le -\frac{\rho\theta}{x^T s}(\|D^{-1}p\|^2 - \|D^{-1}p\|\,\|D^{-1}q\|) \\
&\le \frac{\rho\theta}{x^T s}(\|D^{-1}p\|^2 - C\beta\|D^{-1}p\|^2) \quad \text{from (A.7)} \\
&= -\frac{\rho\theta}{x^T s}(1 - C\beta)\|D^{-1}p\|^2 \\
&\le -\frac{\sqrt{3}}{2}(1 - C\beta)\theta\|D^{-1}\|\,\|D^{-1}p\| \\
&= -\frac{\sqrt{3}}{2}(1 - C\beta)\beta \quad \text{from (A.2),}
\end{aligned}
$$

where the last inequality holds since

$$\frac{\rho}{x^T s}\|D^{-1}p\| = \left\| D^{-1}e - \frac{\rho}{\|De\|^2} De \right\| \ge \frac{\sqrt{3}}{2}\|D^{-1}\|$$

for $\rho \ge n + \sqrt{n}$ (see [8] and [13]).

Finally, we have from (A.5), (A.8), and (A.9)

$$
\begin{aligned}
&\phi(x^+, s^+) - \phi(x, s) \\
&\le -\frac{\sqrt{3}}{2}(1 - C\beta)\beta + \frac{\rho}{x^T s}\Delta x^T \Delta s + \delta^2 \\
\text{(A.10)} \qquad &\le -\frac{\sqrt{3}}{2}\beta(1 - C\beta) + \beta^2(1 + C\beta)^2 + \left(\frac{\rho}{x^T s} - 2\|D^{-1}\|^2 \right) \Delta x^T \Delta s \\
&= -\frac{\sqrt{3}}{2}\beta(1 - C\beta) + \beta^2(1 + C\beta)^2 + \left(\frac{\rho}{x^T s} - \frac{2}{\min(x_j s_j)} \right) \Delta x^T \Delta s.
\end{aligned}
$$

Since $x^T s \ge n$ min $(x_j s_j)$, for any $\rho \le 2n$ we have

$$\frac{2}{\min(x_j s_j)} \ge \frac{2n}{x^T s} \ge \frac{\rho}{x^T s},$$

which indicates that the last term in (A.10) is nonpositive. Hence

$$\phi(x^+, s^+) - \phi(x, s) \le -\frac{\sqrt{3}}{2}\beta(1 - C\beta) + \beta^2(1 + C\beta)^2.$$

Acknowledgment. The authors would like to thank Richard Tapia whose suggestions led to an improved presentation of their results.

REFERENCES

[1] Y. Censor, T. Elfving, and G. T. Herman, *Methods for entropy maximization with applications in image processing*, in Proc. 3rd Scandinavian Conf. on Image Analysis, P. Johansen and P. W. Becker, eds., Chartwell-Bratt, Lund, Sweden, 1983, pp. 296–300.

[2] T. Coleman and Y. Li, *A quadratically-convergent algorithm for the linear programming problems with lower and upper bounds*, Tech. Rep., Dept. of Computer Science, Cornell Univ., Ithaca, NY, 1990.

[3] G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963.

[4] D. den Hertog, C. Roos, and T. Terlaky, *On the classical logarithmic barrier function method for a class of smooth convex programming problems*, Rep. 90-28, Faculty of Technical Mathematics and Informatics, Delft Univ. of Technology, the Netherlands, 1990.

[5] G. T. Herman, *A relaxation method for reconstructing objects from noisy X-rays*, Math. Programming, 8(1975), pp. 1–19.

[6] F. Jarre, *On the Complexity of a Numerical Algorithm for Solving Smooth Convex Programs by Following a Central Path*, Manuscript, University of Wurzburg, West Germany, 1988.

[7] M. Kojima, S. Mizuno, and A. Yoshise, *A polynomial-time algorithm for a class of linear complementarity problems*, Math. Programming, 44(1989), pp. 1–26.

[8] ———, *An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems*, Math. Programming, 50(1991), pp. 331–342.

[9] K. O. Kortanek, F. Potra, and Y. Ye, *On some efficient interior point methods for nonlinear convex programming*, Linear Algebra Appl., 152(1991), pp. 169–189.

[10] I. J. Lustig, R. Marsten, and D. F. Shanno, *Computational experience with a primal-dual interior point methodfor linear programming*, Rep. SOR-89-17, Dept. of Civil Engineering and Operations Research, Princeton Univ., Princeton, NJ, 1989.

[11] R. C. Monteiro and I. Adler, *An extension of Karmarkar type algorithm to a class of convex separable programming problems with global rate of convergence*, Math. Oper. Res., 15(1990), pp. 408–422.

[12] Y. E. Nesterov and A. S. Nemirovsky, *Self-Concordant Functions and Polynomial Time Methods in Convex Programming*, Centr. Econ. and Math. Inst., USSR Acad. Sci., USSR, 1989.

[13] P. M. Pardalos, Y. Ye, and C.-G. Han, *Algorithms for the solution of quadr7atic knapsack problems*, Linear Algebra Appl. 152(1991), pp. 69–91.

[14] F. A. Potra, *On Q-order and R-order of convergence*, J. Optim. Theory Appl., 63(1989), pp. 415–431.

[15] F. A. Potra and Y. Ye, *Interior-point methods for nonlinear complementarity problems*, Working Paper No. 15, Dept. of Mathematics, Univ. of Iowa, Iowa City, IA, 1991.

[16] M. J. Todd and Y. Ye, *A centered projective algorithm for linear programming*, Math. Oper. Res., 15(1990), pp. 508–529.

[17] D. S. Wong, *Maximum likehood, entropy maximization, and geometric programming approaches to the calibration of trip distribution models*, Transportation Res., Part B., 15(1981), pp. 329–343.

[18] Y. Ye, *Interior Algorithms for Linear, Quadratic, and Linearly Constrained Convex Programming*, Ph.D. thesis, Dept. of Engineering-Economic Systems, Stanford Univ., CA, 1987.

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

[20] S. A. Zenios, *Matrix balancing on a massively parallel connection machine*, ORSA J. Comput., 2(1990), pp. 112–125.

[21] Y. Zhang and R. A. Tapia, *A superlinearly convergent polynomial primal-dual interior-point algorithm for linear programming*, TR90-40, Dept. of Mathematical Sciences, Rice Univ., Houston, TX, 1990; SIAM J. Optimization, 3(1993), pp. 118–133.

[22] Y. Zhang, R. A. Tapia, and J. E. Dennis, *On the superlinear and quadratic convergence of primal-dual interior point linear programming algorithms*, TR90-6, Dept. of Mathematical Sciences, Rice Univ., Houston, TX, 1990; SIAM J. on Optimization, 2(1992), pp. 304–323.

[23] Y. Zhang, R. A. Tapia, and F. Potra, *On the superlinear convergence of interior-point algorithms for a general class of problems*, TR90-9, Dept. of Mathematical Sciences, Rice Univ., Houston, TX, 1990; SIAM J. on Optimization, 3(1993), pp. 413–422.

[24] J. Zhu, *A path following algorithm for a class of convex programming problems*, Working Paper No. 90-14, College of Business Administration, Univ. of Iowa, Iowa City, IA, 1990.

SIAM J. OPTIMIZATION
Vol. 3, No. 4, pp. 861–867, November 1993

011

ON MIZUNO'S RANK-ONE UPDATING ALGORITHM FOR LINEAR PROGRAMMING*

ROBERT A. BOSCH†

Abstract. Recently, Mizuno devised a linear programming algorithm that performs at most one rank-one update of a certain matrix per iteration. Mizuno's algorithm is generalized by a variant that allows any fixed number of updates per iteration. This variant also makes use of an explicit Goldstein–Armijo condition to safeguard linesearches of the potential function, compared to Mizuno's implicit use of such a condition in his analysis. The variant's complexity is $O([mn + m^2]nL)$ operations, which is the same as that of the original.

Key words. linear programming, interior point algorithms, potential function, modified method, rank-one updates

AMS subject classification. 90C05

1. Introduction. Since Karmarkar [7], the technique of partial updating has been used to lower the complexity of interior point algorithms for linear programming. Karmarkar's original projective algorithm requires $O(nL)$ iterations and $O([mn+m^2n]nL)$ total arithmetic operations; $O(m^2n)$ per iteration for solving weighted least squares problems and $O(mn)$ per iteration for everything else. (Here n is the number of variables, m is the number of constraints, and L is the bit size of a standard form problem with integer data.) Karmarkar's "modified" algorithm uses a rank-one update technique to solve the weighted least squares problems. On average, only $O(\sqrt{n})$ rank-one updates, each requiring $O(m^2)$ arithmetic operations, are needed per iteration. As a result, a total of $O([m^2\sqrt{n}]nL)$ arithmetic operations are expended on weighted least squares problems. The iteration count is the same order as that of the original, but the overall complexity is lowered to $O([mn + m^2\sqrt{n}]nL)$.

Karmarkar's two algorithms are potential reduction algorithms, and as such, the faster they reduce the potential function, the faster they converge. In Karmarkar's original algorithm, a linesearch of the potential function is permissible; incorporating a linesearch does not violate any aspect of the complexity analysis. In the modified algorithm, a simple linesearch is precluded by Karmarkar's complexity analysis. Linesearches safeguarded by a Goldstein–Armijo rule are permissible, however, as demonstrated in Anstreicher [1].

Partial updating has since been applied to other algorithms. Gonzaga [5], Vaidya [14], Monteiro and Adler [13], Kojima, Mizuno, and Yoshise [8], and Ye [15] all incorporate the technique. (Each of the above algorithms has an $O(\sqrt{n}L)$ iteration count. The first four listed are path-following algorithms. Ye's algorithm is a potential reduction algorithm.) Anstreicher and Bosch [2] incorporate the safeguarded linesearch of Anstreicher [1] into Ye's algorithm, and Mizuno [11] presents other partial updating variants of Ye's algorithm. Bosch and Anstreicher [4] apply partial updating and Anstreicher's safeguarded linesearch to the $O(\sqrt{n}L)$ potential reduction algorithm of Kojima, Mizuno, and Yoshise [8], and den Hertog, Roos, and Vial [6] do the same within the context of long-step path following.

All of the partial updating algorithms mentioned thus far perform, on average, $O(\sqrt{n})$ rank-one updates per iteration. Mizuno [12] presented a modified version of Ye's algorithm that performs at most *one* rank-one update per iteration. However, Mizuno's algorithm has an $O(nL)$ iteration count. The overall complexity of the method is $O([mn + m^2]nL)$.

In this paper we examine Mizuno's algorithm and generalize it. We present a variant of Mizuno's algorithm that allows for up to any fixed number of updates per iteration. In addition, our variant utilizes the safeguarded linesearch of Anstreicher [1] in place of Mizuno's use of fixed stepsizes. We assume that the reader is familiar with Mizuno [12], so we omit the details

* Received by the editors September 20, 1991; accepted for publication (in revised form) August 12, 1992.

† Department of Mathematics, Oberlin College, Oberlin, Ohio 44074.

of a number of features of our variant that are identical to features of Mizuno's algorithm (the entire dual step, for instance). Also, many of our proofs are very similar in structure to Mizuno's proofs. To enhance the readability of this paper, we present the proofs in full, even though doing so necessitates the repetition of a number of arguments originally presented in Mizuno [12].

Throughout the paper, we use the notation of Mizuno [12] as much as possible. We let $\| \cdot \|$ and $\| \cdot \|_1$ denote the l_2-norm and l_1-norm, respectively. For any vector $v \in \Re^n$, $V = \text{diag}(v)$ denotes the diagonal matrix whose ith diagonal entry is v_i.

2. The algorithm. Both Mizuno's algorithm and our variant of it work directly on a standard form linear program and its dual:

$$\text{(P)} \qquad \begin{array}{rl} \min & c^{\mathrm{T}}x \\ & Ax = b, \\ & x \geq 0, \end{array}$$

$$\text{(D)} \qquad \begin{array}{rl} \max & b^{\mathrm{T}}y \\ & A^{\mathrm{T}}y + z = c, \\ & z \geq 0, \end{array}$$

where A is an $m \times n$ matrix with independent rows. Both (P) and (D) are assumed to have nonempty relative interiors. Each algorithm produces a primal interior point x^j, its approximation $\hat{x}^j$, a dual interior point (y^j, z^j), and an inverse matrix $B_j = (A\hat{X}_j^2A^{\mathrm{T}})^{-1}$ on each iteration. (An alternate approach is to maintain a factorization of $A\hat{X}_j^2A^{\mathrm{T}}$.) To keep the primal interior point and its approximation close to one another, each index i is forced to satisfy

$$\text{(1)} \qquad \frac{\hat{x}_i^j}{\rho} \leq x_i^j \leq \rho\hat{x}_i^j,$$

where $\rho > 1$ is a fixed number. Whenever the primal interior point changes, (1) must be checked. If (1) fails to hold for u of the n indices, then those u entries of the approximate point must be altered. In addition, the inverse matrix (or factorization) must be modified. This can be accomplished by performing u successive rank-one modifications of the current inverse matrix (or factorization). Each such rank-one modification, or update, requires $O(m^2)$ arithmetic operations.

Each of the algorithms measures progress in (P) and (D) via the primal-dual potential function

$$\phi(x, z) = (n + \sqrt{n})\ln(x^{\mathrm{T}}z) - \sum_{i=1}^{n} \ln(x_i z_i) - n\ln(n).$$

The algorithms are initially provided with interior points x^0 and (y^0, z^0) for which $\phi(x^0, z^0) = 0(\sqrt{n}L)$. The algorithms are terminated when $\phi(x^j, z^j) \leq -2\sqrt{n}L$. (Given such solutions, a standard "rounding" procedure will produce exact optimal basic solutions to (P) and (D) in $O(mn^2)$ operations.) On iteration j, the algorithms try to reduce $\phi(\cdot, \cdot)$ by taking either a dual step or a primal step. A dual step involves moving from (y^j, z^j) to (y', z'), where

$$y' = y^j + \frac{(x^j)^{\mathrm{T}}z^j}{n + \sqrt{n}} B_j A\hat{X}_j^2 \nabla_x \phi(x^j, z^j), \qquad z' = c - A^{\mathrm{T}}y'.$$

A primal step takes the form

$$x(\beta) = x^j - \beta \hat{X}_j p^j,$$

where p^j is the projection of the vector $\hat{X}_j \nabla_x \phi(x^j, z^j)$ onto the nullspace of $A\hat{X}_j$. In our variant, we force the steplength β to satisfy a Goldstein–Armijo condition

$$\phi(x(\beta), z^j) - \phi(x^j, z^j) \leq \frac{1}{2}\beta \frac{d}{d\beta}\bigg|_{\beta=0} \phi(x(\beta), z^j),$$

which, since $\frac{d}{d\beta}|_{\beta=0}\phi(x(\beta), z^j) = -\|p^j\|^2$, may be written as

$$\text{(GA)} \qquad \phi(x(\beta), z^j) - \phi(x^j, z^j) \leq -\tfrac{1}{2}\beta\|p^j\|^2.$$

In Mizuno's algorithm, three sets are formed: I, the set of primal iterations on which one update is performed; J, the set of primal iterations on which zero updates are performed; and K, the set of dual iterations. Here, we let I_u^P denote the set of primal iterations on which precisely u updates are performed. We let I^D be the set of dual iterations and I^P be the set of primal iterations (hence $I^P = I_0^P \cup I_1^P \cup \cdots \cup I_{\overline{u}}^P$, where $\overline{u}$ is the maximum number of updates we allow per iteration). In addition, we let U_j denote the set of indices that correspond to updates on iteration j. We now describe our algorithm.

ALGORITHM.

Input. A, b, c, m, and n; $\rho > 1$; $1 \leq \overline{u} \leq n$; a primal interior point x^0 and a dual interior point (y^0, z^0) for which $\phi(x^0, z^0) = O(\sqrt{n}L)$.

Step 1. Let $j = 0, \hat{x}^0 = x^0$, and $I^P \cup I^D = \emptyset$. Compute the matrix $B_0 = (A\hat{X}_0^2 A^\top)^{-1}$.

Step 2. If $\phi(x^j, z^j) \leq -2\sqrt{n}L$ then stop.

Step 3. Compute y', z', and p^j.

Step 4. If $\|p^j\| \leq (4\rho)^{-1}$ then go to Step 7. Otherwise, compute

$$\alpha_i = \max\left\{\alpha : \frac{\hat{x}_i^j}{\rho} \leq x_i(\alpha) \leq \rho\hat{x}_i^j\right\}$$

for each i. Find the smallest of the α_i, β', and the $\overline{u}$th smallest, $\beta_{\max}$. These two values may be computed in $O(n)$ arithmetic operations (see Blum et al. [3] for details). If condition (GA) holds with $\beta = \beta'$ go to Step 5. Otherwise, go to Step 6.

Step 5. Let $(y^{j+1}, z^{j+1}) = (y^j, z^j)$ and

$$\beta_j = \text{argmin}\{\phi(x(\beta), z^j) : \beta' \leq \beta \leq \beta_{\max}, \text{(GA) holds}, x(\beta) > 0\}.$$

Let $x^{j+1} = x(\beta_j)$. Let U_j be the set of all indices i for which either $x_i^{j+1} < \hat{x}_i^j/\rho$ or $x_i^{j+1} > \rho\hat{x}_i^j$. If $|U_j| < \overline{u}$, augment U_j by adding to it up to $\overline{u} - |U_j|$ indices i for which either $x_i^{j+1} = \hat{x}_i^j/\rho$ or $x_i^{j+1} = \rho\hat{x}_i^j$. Let $I_{|U_j|}^P \leftarrow I_{|U_j|}^P \cup \{j\}$. Let

$$\hat{x}_i^{j+1} = \begin{cases} \hat{x}_i^j & \text{if } i \notin U_j, \\ x_i^{j+1} & \text{if } i \in U_j. \end{cases}$$

Compute the matrix $B_{j+1} = (A\hat{X}_{j+1}^2 A^\top)^{-1}$ from B_j by means of $|U_j|$ rank-one updates. Increase j by one and go to Step 2.

Step 6. Let $(y^{j+1}, z^{j+1}) = (y^j, z^j)$ and

$$\beta_j = \text{argmin}\{\phi(x(\beta), z^j) : 0 \leq \beta \leq \beta', \text{(GA) holds}, x(\beta) > 0\}.$$

Let $x^{j+1} = x(\beta_j), U_j = \emptyset, I_0^P \leftarrow I_0^P \cup \{j\}, \hat{x}^{j+1} = \hat{x}^j$, and $B_{j+1} = B_j$. Increase j by one and go to Step 2.

Step 7. Let $(y^{j+1}, z^{j+1}) = (y', z'), x^{j+1} = x^j, U_j = \emptyset, I^D \leftarrow I^D \cup \{j\}, \hat{x}^{j+1} = \hat{x}^j$, and $B_{j+1} = B_j$. Increase j by one and go to Step 2.

Steps 5 and 6 are the primal steps. Note that if Step 5 was executed on iteration j, then at least one update was performed on that iteration. If Step 6 was executed instead, then no updates were performed on that iteration. Consequently, if $j \in I_0^P$, then condition (GA) did not hold with $\beta = \beta'$. (See Step 4.)

Note that Mizuno's algorithm uses $\overline{u} = 1$. Also, instead of checking the Goldstein–Armijo condition in Step 4, Mizuno's algorithm checks if $\beta' < \rho^{-4}$. Finally, Mizuno's algorithm uses fixed stepsizes of β' and ρ^{-4} in Steps 5 and 6, respectively, in place of the linesearches. Although Mizuno never *explicitly* uses the Goldstein–Armijo condition, an important step in his analysis is his argument that the fixed stepsizes β' and ρ^{-4} satisfy the condition (see the first part of Mizuno's Lemma 3).

3. Complexity. In this section we provide worst-case bounds for the number of updates, iterations, and arithmetic operations required by our variant of Mizuno's algorithm. Theorem 4 is the main result of the paper. Proposition 1, which we give without proof, is the first part of Lemma 1 of Mizuno [12].

PROPOSITION 1. *For each fixed $x^j > 0$ and $z^j > 0$, we have*

$$\phi(x, z^j) - \phi(x^j, z^j) \leq \nabla_x \phi(x^j, z^j)^{\mathrm{T}}(x - x^j) + \tfrac{1}{2}\sigma_j(x)\|X_j^{-1}(x - x^j)\|^2,$$

for each $x > 0$, where $\sigma_j(x) = \max\{1, x_1^j/x_1, x_2^j/x_2, \ldots, x_n^j/x_n\}$.

LEMMA 2. *Fix $\rho \geq 1$. Assume that $\|p^j\| \geq (4\rho)^{-1}$. If $0 \leq \beta \leq (5\rho^3\|p^j\|)^{-1}$, then condition* (GA) *holds. Furthermore, if $\beta = (5\rho^3\|p^j\|)^{-1}$, then $\phi(x(\beta), z^j) - \phi(x^j, z^j) \leq -(40\rho^4)^{-1}$.*

Proof. Let $0 \leq \beta \leq (5\rho^3\|p^j\|)^{-1}$. By Proposition 1, (1), and the definitions of $x(\beta)$ and p^j,

$$\phi(x(\beta), z^j) - \phi(x^j, z^j) \leq -\beta\|p^j\|^2 + \tfrac{1}{2}\sigma_j(x(\beta))\beta^2\rho^2\|p^j\|^2.$$

If $p_i^j \geq 0$, then the choice of β, (1), and the assumption that $\rho \geq 1$ imply that

$$\frac{x_i(\beta)}{x_i^j} = 1 - \frac{\beta\hat{x}_i^j p_i^j}{x_i^j} \geq 1 - \frac{\hat{x}_i^j p_i^j}{5\rho^3 x_i^j\|p^j\|} \geq 1 - \frac{1}{5\rho^2} \geq \frac{4}{5}.$$

If $p_i^j < 0$, then $x_i(\beta)/x_i^j > 1$. Hence $\sigma_j(x(\beta)) \leq \frac{5}{4}$. We thus obtain

$$\phi(x(\beta), z^j) - \phi(x^j, z^j) \leq (-1 + \tfrac{5}{8}\rho^2\beta)\beta\|p^j\|^2. \tag{2}$$

The choice of β and $\|p^j\| \geq (4\rho)^{-1}$ imply that $\frac{5}{8}\rho^2\beta \leq (8\rho\|p^j\|)^{-1} \leq \frac{1}{2}$. By applying this last inequality to (2) we obtain condition (GA). Now let $\beta = (5\rho^3\|p^j\|)^{-1}$. From condition (GA), we obtain

$$\phi(x(\beta), z^j) - \phi(x^j, z^j) \leq -\frac{\|p^j\|}{10\rho^3}.$$

The second part of the lemma follows from $\|p^j\| \geq (4\rho)^{-1}$. □

LEMMA 3. *We have*

$$\phi(x^{j+1}, z^{j+1}) - \phi(x^j, z^j) \leq -\tfrac{1}{2}\beta_j\|p^j\|^2 \quad \textit{for each } j \in I^P$$

and

$$\phi(x^{j+1}, z^{j+1}) - \phi(x^j, z^j) \leq -\frac{1}{40\rho^4} \quad \textit{for each } j \in I_0^P \cup I^D.$$

Proof. The first part of the lemma follows from the fact that condition (GA) holds for each $j \in I^P$. (If $j \in I_0^P$, then $(5\rho^3\|p^j\|)^{-1} < \beta'$. Otherwise, Lemma 2 would imply that condition (GA) holds with $\beta = \beta'$, which cannot happen when $j \in I_0^P$.) For $j \in I_0^P$, the second part of the lemma follows from Lemma 2. (By Lemma 2, if $(5\rho^3\|p^j\|)^{-1}$ were used as the steplength, condition (GA) would hold and the potential function would be reduced by at least $(40\rho^4)^{-1}$. The actual steplength, $\beta_j = \operatorname{argmin}\{\phi(x(\beta), z^j) : 0 \leq \beta \leq \beta', \text{(GA) holds}, x(\beta) > 0\}$, must reduce the potential function by at least as much.) For $j \in I^D$, the second part of the lemma follows from Lemma 3 of Mizuno [12]. □

THEOREM 4. *The total number of updates performed by the algorithm proposed in the previous section is* $\sum_{u=1}^{\overline{u}} u|I_u^P| = O(nL)$. *The number of iterations required is* $|I^P \cup I^D| = O(nL)$. *The total number of arithmetic operations expended is* $O([mn + m^2]nL)$.

Proof. Assume that the algorithm is terminated at the beginning of iteration $k+1$, so that $\phi(x^{k+1}, z^{k+1}) \leq -2\sqrt{n}L$ and $\phi(x^k, z^k) > -2\sqrt{n}L$. Then , since $\phi(x^0, z^0) = O(\sqrt{n}L)$, we have

$$\phi(x^0, z^0) - \phi(x^k, z^k) = O(\sqrt{n}L). \tag{3}$$

By Lemma 3, $\phi(\cdot, \cdot)$ is reduced by at least a positive constant independent of n and L on each iteration $j \in I_0^P \cup I^D$. From this and (3), we obtain $|I_0^P| = O(\sqrt{n}L)$ and $|I^D| = O(\sqrt{n}L)$. Also by Lemma 3, we have that

$$\begin{aligned}\phi(x^k, z^k) - \phi(x^0, z^0) &= \sum_{j=0}^{k-1}(\phi(x^{j+1}, z^{j+1}) - \phi(x^j, z^j)) \\ &\leq -\left(\sum_{j \in I^P \setminus \{k\}} \frac{1}{2}\beta_j\|p^j\|^2 + \sum_{j \in I^D \setminus \{k\}} \frac{1}{40\rho^4}\right).\end{aligned}$$

From the above inequality and (3), we obtain

$$\sum_{j \in I^P \setminus \{k\}} \beta_j\|p^j\|^2 = O(\sqrt{n}L). \tag{4}$$

As in Mizuno [12], let $\delta = 1 - 1/\rho$ and let $d_j = \|\hat{X}_j^{-1}(x^j - \hat{x}^j)\|_1$ for each j. Note that

$$d_{j+1} = d_j \quad \text{for each } j \in I^D. \tag{5}$$

Also, if $j \in I_u^P$ then

$$\begin{aligned}d_{j+1} &= \sum_{i \notin U_j}\left|\frac{x_i^{j+1}}{\hat{x}_i^j} - 1\right| \\ &= \|\hat{X}_j^{-1}(x^{j+1} - \hat{x}^j)\|_1 - \sum_{i \in U_j}\left|\frac{x_i^{j+1}}{\hat{x}_i^j} - 1\right| \\ &\leq \|\hat{X}_j^{-1}(x^{j+1} - x^j)\|_1 + \|\hat{X}_j^{-1}(x^j - \hat{x}^j)\|_1 - \delta u.\end{aligned}$$

The above inequality may be written as

$$\delta u \le d_j - d_{j+1} + \beta_j \|p^j\|_1 \quad \text{for each } j \in I_u^P. \tag{6}$$

By (5) and (6), we have

$$\begin{aligned}\sum_{u=1}^{\overline{u}} \sum_{j\in I_u^P\setminus\{k\}} \delta u &\le \sum_{j\in I^P\setminus\{k\}} (d_j - d_{j+1} + \beta_j\|p^j\|_1) + \sum_{j\in I^D\setminus\{k\}} (d_j - d_{j+1})\\ &= \sum_{j=0}^{k-1}(d_j - d_{j+1}) + \sum_{j\in I^P\setminus\{k\}} \beta_j\|p^j\|_1\\ &\le d_0 - d_k + \sum_{j\in I^P\setminus\{k\}} \beta_j\sqrt{n}\|p^j\|\\ &\le \sum_{j\in I^P\setminus\{k\}} 4\rho\beta_j\sqrt{n}\|p^j\|^2,\end{aligned}$$

where the last inequality follows from $d_0 = 0, d_k \ge 0$, and $\|p^j\| \ge (4\rho)^{-1}$. Therefore, since at most $\overline{u}$ updates are performed on iteration k, we obtain

$$\sum_{u=1}^{\overline{u}} u|I_u^P| \le \left(\sum_{u=1}^{\overline{u}} \sum_{j\in I_u^P\setminus\{k\}} u\right) + \overline{u} \le \left(\left(\frac{4\rho}{\delta}\right)\sqrt{n} \sum_{j\in I^P\setminus\{k\}} \beta_j\|p^j\|^2\right) + \overline{u}.$$

Consequently, by (4) and $\overline{u} \le n$, we have $\sum_{u=1}^{\overline{u}} u|I_u^P| = O(nL)$. Note that this implies that $\sum_{u=1}^{\overline{u}} |I_u^P| = O(nL)$. Since $|I_0^P| = O(\sqrt{n}L)$ and $|I^D| = O(\sqrt{n}L)$, it follows that $|I^P \cup I^D| = O(nL)$. Finally, note that $O(m^2)$ arithmetic operations are expended per update, and that $O(mn)$ arithmetic operations are expended on nonupdating matters per iteration. Thus, the complexity of the algorithm is $O([mn + m^2]nL)$. □

Theorem 4 has an additional implication, namely, that the number of primal iterations on which precisely u updates are performed is $O([n/u]L)$, where $1 \le u \le \overline{u}$. For instance, if the algorithm is run with $\overline{u} = n$, then the number of iterations on which n updates are performed is $O(L)$.

4. Conclusions.

The algorithm presented in §2 is very similar to Mizuno's rank-one updating algorithm. Although it allows for up to any fixed number $\overline{u}$ of updates per iteration, instead of just $\overline{u} = 1$, it achieves the same worst-case update count, iteration count, and complexity. In addition, it incorporates linesearches of the potential function, safeguarded by a Goldstein–Armijo condition.

Acknowledgment. The author would like to thank Kurt Anstreicher for helpful comments and discussions.

REFERENCES

[1] K. M. Anstreicher, *A standard form variant, and safeguarded linesearch, for the modified Karmarkar algorithm*, Math. Programming, 47(1990), pp. 337–351.

[2] K. M. Anstreicher and R. A. Bosch, *Long steps in a $O(n^3L)$ algorithm for linear programming*, Math. Programming, 54(1992), pp. 251–265.

[3] M. Blum, R. W. Floyd, V. R. Pratt, R. L. Rivest, and R. E. Tarjan, *Time bounds for selection*, J. Comput. System Sci., 7(1973), pp. 448–461.

[4] R. A. Bosch and K. M. Anstreicher, *On Partial Updating in a Potential Reduction Linear Programming Algorithm of Kojima, Mizuno, and Yoshise*, Dept. of Operations Research, Yale Univ., New Haven, CT, 1990; Algorithmica, 9(1993), pp. 184–197.

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

[6] D. den Hertog, C. Roos, and J.-Ph. Vial, *A complexity reduction for the long-step path-following algorithm for linear programming*, SIAM J. Optim., 2(1992), pp. 71–87.

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

[8] M. Kojima, S. Mizuno, and A. Yoshise, *An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems,* Math. Programming, 50(1991), pp. 331–342.

[9] ———, *A polynomial-time algorithm for a class of linear complementarity problems*, Math. Programming, 44(1989), pp. 1–26.

[10] S. Mehrotra, *Deferred Rank One Updates in $O(n^3L)$ Interior Point Algorithms,* Tech. Rep. 90-11, Dept. of Industrial Engineering and Management Sciences, Northwestern Univ., Evanston, IL, 1990.

[11] S. Mizuno, *$O(n^\rho L)$ Iteration $O(n^3L)$ Potential Reduction Algorithms for Linear Programming,* Tech. Report 22, Dept. of Management Science and Engineering, Tokyo Institute of Technology, Tokyo, Japan, 1989.

[12] ———, *A rank one updating interior algorithm for linear programming*, The Arabian J. Sci. Engrg., 15(1990), pp. 671–677.

[13] R. C. Monteiro and I. Adler, *Interior path following primal-dual algorithms. Part II: Convex quadratic programming*, Math. Programming, 44(1989), pp. 43–66.

[14] P. M. Vaidya, *An algorithm for linear programming which requires $O(((m+n)n^2 + (m+n)^{1.5}n)L)$ arithmetic operations*, Math. Programming, 47(1990), pp. 175–201.

[15] Y. Ye, *A $O(n^3L)$ potential reduction algorithm for linear programming*, Math. Programming, 50(1991), pp. 239–258.

SIAM J. OPTIMIZATION
Vol. 3, No. 4, pp. 868–881, November 1993

ACCELERATED STOCHASTIC APPROXIMATION*

BERNARD DELYON† AND ANATOLI JUDITSKY†

Abstract. A technique to accelerate convergence of stochastic approximation algorithms is studied. It is based on Kesten's idea of equalization of the gain coefficient for the Robbins–Monro algorithm. Convergence with probability 1 is proved for the multidimensional analog of the Kesten accelerated stochastic approximation algorithm. Asymptotic normality of the delivered estimates is also shown. Results of numerical simulations are presented that demonstrate the efficiency of the acceleration procedure.

Key words. stochastic approximation, accelerated algorithms, optimal algorithms

AMS subject classifications. 62L20, 93B30

1. Introduction. Let us consider the problem of searching for the stationary point x^* of the vector field $\varphi(x) : R^N \to R^N$. The observations y_t of the function $\varphi(\cdot)$ are available at any point $x_{t-1} \in R^N$ and contains random disturbance ξ_t:

$$y_t = \varphi(x_{t-1}) + \xi_t. \tag{1}$$

The problem is to find x^* under the assumption that a unique solution exists.

The method of stochastic approximation (SA) (which takes its origin from [10]) is well studied for this problem. To obtain a sequence of estimates of the solution x^*, the following recursive procedure is used:

$$x_t = x_{t-1} - \gamma_t y_t, \tag{2}$$

where γ_t is a gain coefficient and x_0 is an arbitrary fixed point in R^N. In the study of this algorithm the main focus of attention was the asymptotic analysis of the method when $\gamma_t = \gamma t^{-1}$. For this case conditions have been obtained under which almost sure convergence and asymptotic normality take place (see [6] and [15]). Asymptotically optimal versions (algorithms that ensure the highest asymptotic rate of convergence) of that method have also been developed in the works of Venter [14], Fabian [3], and Polyak and Tsypkin [9].

On the other hand, nonasymptotic properties of SA algorithms are the main focus of the interest in applications. Unfortunately, as is well known to engineers (see the discussion in [13]), asymptotically optimal methods behave badly in finite time: the choice of the gain γt^{-1} is too "cautious" if the disturbance ξ_t is small with respect to the initial error $x_0 - x^*$. Several heuristic procedures have been suggested in order to accelerate convergence in a finite time interval (see, for instance, [13, Chap. 5])[1].

In particular, the accelerated SA procedure has been studied in the work of Kesten [4] for the one-dimensional case. It is based on the idea that frequent changes of the sign of the difference $x_t - x_{t-1} = \gamma_t y_t$ indicate that the estimates are close to the real solution and are significantly disturbed by noise, whereas few fluctuations of the sign indicate that x_t is still far from x^*. In fact, the number s_t of changes of the sign of y_i for $i = 1, \dots, t-1$ constitutes a new time scale. According to this scale, small values of s_t mean that large gains γ_t (in other words, large magnitudes of correction) should be used at the tth step and, in turn, large values of s_t mean that the procedure has "reached" its asymptotic region and $\gamma_t = \gamma t^{-1}$ should be used. Almost sure convergence of that procedure has been proved.

* Received by the editors on June 17, 1991; accepted for publication (in revised form) on July 25, 1992.

† Institut de Recherche en Informatique et Systèmes Aléatoires/Institut National de Recherche en Informatique et en Automatique, Campus de Beaulieu, 35042, Rennes, France.

[1] As noted in [4], the investigation of this problem had been suggested by Robbins in his first works on SA.

The principal issue of this paper is a result of the almost sure convergence of the multidimensional analog of Kesten's algorithm. Based on that in §3 we obtain conditions for asymptotic normality for the accelerated version of the usual SA procedure. In §4 we study Kesten-like modification of the Ruppert–Polyak (see [8] and [12]) SA algorithm. Section 5 contains results of numerical simulations.

2. Kesten's algorithm. In order to obtain the estimates x_t of x^* we use the following algorithm:

$$x_t = x_{t-1} - \gamma_t y_t, \qquad x_0 \in R^N, \tag{3}$$

where the scalar gain γ_t is defined by the equations

$$s_{t+1} = s_t + I(y_t^T y_{t-1} < 0), \tag{4}$$

$$\gamma_{t+1} = \gamma(s_{t+1}) \tag{5}$$

(here $\gamma(t)$ is a deterministic sequence).

We suppose that we have a probability space $(\Omega, \mathcal{F}, P)$ with an increasing family of σ-fields $\mathcal{F}_t = \sigma(x_0, \xi_1, \dots, \xi_t)$. Let us consider the following assumptions on the problem.

ASSUMPTION 1. ξ_t is a sequence of random variables such that the conditional distribution $P_x(d\xi)$ of ξ_t, knowing the past, depends only on $x_{t-1} = x$. Furthermore, $E(\xi_t | x_{t-1}) = 0$ and for some $S_M, E(\xi_t \xi_t^T | x_{t-1}) \leq S_M$. The measures P_x satisfy

$$\lim_{x \to x^*} \|P_x - P_{x^*}\| = 0, \tag{6}$$

where $\|\cdot\|$ denotes the total variation. Moreover, for any hyperplane H containing the origin, $P_{x^*}(H) = 0$. For any $R > 0$ and $\delta > 0$,

$$\min_{|x|<R} P_x(|\xi| \leq \delta) > 0. \tag{7}$$

ASSUMPTION 2. $\varphi(x)$ is a continuous function of x. There exists $\rho > 0$ such that for any $\gamma^* \leq \rho$ and any starting point x_0, the deterministic sequence

$$x_{t+1} = x_t - \gamma^* \varphi(x_t) \tag{8}$$

converges to x^*. There exists a function $V(x) : R^N \to R^+$, positive $\beta, \bar{R}$, and a matrix $M > 0$ such that

$$\begin{aligned} &V(x^*) = 0, \\ &\nabla^2 V(x) \leq M \quad \text{for all } x, \\ &\varphi(x)^T \nabla V(x) \geq \frac{\rho}{2}(\varphi(x)^T M \varphi(x) + tr(S_M M)) + \beta \end{aligned}$$

for any x such that $|x - x^*| \geq \bar{R}$. Moreover,

$$\varphi(x)^T \nabla V(x) > 0 \quad \text{for any } x \neq x^*.$$

ASSUMPTION 3. The gain coefficient $\gamma(n) > 0$ satisfies

$$\sup_n \gamma(n) \leq \rho, \quad \sum_{n=1}^{\infty} \gamma(n) = \infty, \quad \sum_{n=1}^{\infty} \gamma^2(n) < \infty.$$

Note that Assumption 1 implies that

$$\lim_{x \longrightarrow x^*} E_x \xi\xi^T = E_{x^*}\xi\xi^T = S(x^*) \tag{9}$$

exists, and there is $v > 0$ such that $S(x^*) > vI$ (here E_x denotes the expectation with respect to P_x). Denote $P^* = P_{x^*} \otimes P_{x^*}$.

Comment. We present here an example of the procedure when the conditions stated above are satisfied. Let us consider the following *nonlinear* algorithm for estimating x^*:

$$x_t = x_{t-1} - \gamma_t f(y_t).$$

Here $f(x) : R^N \longrightarrow R^N$ is a nonlinear function. We can rewrite this algorithm in a form similar to (3):

$$x_t = x_{t-1} - \gamma_t \psi(x_{t-1}) - \gamma_t \xi_t,$$

where $\psi(x) = E_x f(\varphi(x) + \xi)$ and $\zeta_t = f(\varphi(x) + \xi) - \psi(x)$. Suppose that Assumption 1 is satisfied. We require that $|f(x)| \leq K_0(1 + |x|)$ and $f(x)$ is continuous. This implies that the functions ψ and $\chi(x) = E_x \zeta_t \zeta_t^T$ are correctly defined and there is K_1 such that $|\chi(x)| \leq K_1$. Furthermore, if the distribution P_x of ξ is absolutely continuous, then Assumption 1 with respect to ζ_t holds true. Given some additional assumptions on f, Assumption 2 can also be verified.

Moreover, one can study the case of nonadditive disturbances (when $y_t = \varphi(x_{t-1}, \xi_t)$; see [1]) in the same way.

THEOREM 1. *Let Assumptions* 1–3 *hold. Then the process defined by* (3)–(5) *satisfies*

$$x_t \longrightarrow x^* \quad a.s.,$$
$$\lim_{t \to \infty} \frac{s_t}{t} - P^*(\xi_1^T \xi_2 < 0) \longrightarrow 0 \quad a.s.$$

Comment. A result similar to the first proposition of Theorem 1 has been stated for the one-dimensional case in [4]. The second proposition of the theorem states that the new time scale, defined by (4), is asymptotically equivalent (up to a coefficient) to the original scale.

Assumption 3 is typical when dealing with stochastic approximation algorithms. Assumption 2 is specific to the Kesten algorithm. It guarantees the stability of the Markov chain, defined by (3) when $\gamma_t \equiv \gamma$. As we shall see later, it ensures a certain regularity in the increase of s_t.

Note that we cannot directly utilize classical results on almost sure convergence of SA procedures (see, for instance, [5, Thm. 2.3.3] and [1, Thm. 2.5.1]). Indeed, the conditions of these results demand, at least, that $\gamma_t \longrightarrow 0$ as $t \longrightarrow \infty$, which is not obvious for the algorithm under consideration.

Proof of Theorem 1. In the proofs of the theorems let us adopt the following conventions: we denote by K, δ, and α generic positive constants. All relations between random variables are supposed to be true almost surely.

An outline of the proof is as follows. We show first the positive recurrency of the process x_t in some vicinity of x^*, where the disturbance ξ_t forces s_t to increase regularly, so that $\gamma_t \longrightarrow 0$. Next we prove that (x_t) visits any neighborhood of x^* infinitely often. We conclude the proof by showing that x_t escapes an arbitrary neighborhood of x^* only a finite number of times.

For the sake of simplicity let us put $x^* = 0$. The following lemma will be used to prove that s_t tends to infinity; it is actually slightly stronger than we need.

LEMMA 1. *For any starting point* $z_0 \in R^N$, *and any* $\gamma^* \leq \rho$, *the Markov chain* z_t *resulting from the equation*

$$z_{t+1} = z_t - \gamma^*(\varphi(z_t) + \xi_{t+1}) \tag{10}$$

satisfies

(i) $P(z_t \in B(\bar{R})$ *infinitely often*$) = 1$, *where* $B(\bar{R})$ *is a ball* $\{|x| < \bar{R}\}$ *and* $\bar{R}$ *is defined as in Assumption 2.*

(ii) *There exist* $\epsilon > 0$ *and* n_0 *such that*

$$P(z_{n_0}^T z_{n_0+1} < 0|z_0) > \epsilon$$

for any $z_0 \in B(\bar{R})$.

Proof. Put $V_t = V(z_t)$ and $\varphi_t = \varphi(z_t)$. As z_t satisfies (10), we have

$$V_{t+1} \leq V_t - \gamma^*\varphi_t^T \nabla V_t - \gamma^* \xi_t^T \nabla V_t + \gamma^{*2}(\varphi_t + \xi_t)^T M(\varphi_t + \xi_t)/2,$$

and from Assumptions 1 and 2

$$\begin{aligned} E(V_{t+1}|z_t) &\leq V_t - \gamma^*\varphi_t^T \nabla V_t + \gamma^{*2}(\varphi_t^T M \varphi_t + tr(S_M M))/2 \\ &\leq V_t - \gamma^*\varphi_t^T \nabla V_t + \gamma^{*2}(\varphi_t^T \nabla V_t - \beta)/\rho + KI(|z_t| < \bar{R}) \\ &\leq V_t - \gamma^{*2}\beta/\rho + KI(|z_t| < \bar{R}). \end{aligned} \tag{11}$$

Define a stopping time $\nu = \inf\{t \geq 1 : |z_t| \leq \bar{R}\}$. Then we derive from (11)

$$\begin{aligned} 0 &\leq EV_{t+1}I(t < \nu) \\ &\leq EV_t I(t < \nu) - \gamma^{*2}\beta/\rho EI(t < \nu) \\ &\leq EV_t I(t-1 < \nu) - \gamma^{*2}\beta/\rho EI(t < \nu) \\ &\leq V_0 + K - \gamma^{*2}\beta/\rho E\left(\sum_{i=0}^{t} I(i < \nu)\right) \\ &= V_0 + K - E\nu\gamma^{*2}\beta/\rho. \end{aligned}$$

Thanks to Assumption 2, $V_0 \leq K|z_0|^2$. Thus

$$E\nu \leq K(|z_0|^2 + 1)\rho/(\gamma^{*2}\beta) \quad \text{and} \quad \nu < \infty \quad \text{a.s.}$$

Hence

$$P(z_t \in B(\bar{R}) \text{ i.o.}) = 1. \tag{12}$$

Note that

$$z_0^T z_1 = |z_0|^2 - \gamma^* z_0^T \varphi_0 - \gamma^* z_0^T \xi_1$$

and the distribution $P_0(d\xi)$ is nondegenerate. Thus the continuity of the $\phi(\cdot)$ along with condition (6) implies the existence of $\delta_1 > 0$ and $\epsilon_1 > 0$ such that

$$P(z_0^T z_1 < 0|z_0) > \epsilon_1 \tag{13}$$

for any $z_0 \in B(\delta_1)$. On the other hand, from the convergence of the deterministic counterpart (8) of the algorithm, condition (7), and, again, the continuity of $\phi(\cdot)$, we obtain that for any $\delta_1 > 0$ there exist n_0 and $\epsilon_2 > 0$ such that

$$P(|z_{n_0}| \leq \delta_1) \geq \epsilon_2 \tag{14}$$

for any $z_0 \in B(\bar{R})$. Hence we get from (14) and (13) that

$$P(z_{n_0}^T z_{n_0+1} < 0|z_0) > \epsilon_2\epsilon_1 = \epsilon$$

as soon as $z_0 \in B(\bar{R})$. □

LEMMA 2. $s_t \to \infty$ *almost surely.*

Proof. For any integer s^*

$$\begin{aligned} P(\lim_{t\to\infty} s_t = s^*) &\leq \sum_t P(s_i = s^* \text{ for any } i > t) \\ &= \sum_t E(P(s_i = s^* \text{ for any } i > t|\mathcal{F}_t)). \end{aligned}$$

It follows from the strong Markov property that the conditional to the $\mathcal{F}_t$ law of the process (γ_{t+i}, x_{t+i}) if s_{t+i} remains equal to s^* coincides with the law Q_{x_t} of the Markov chain given by (10) with $\gamma^* = (s^*)^{-1}$ and starting point $z_0 = x_t$. Consequently,

$$P(\lim_{t\to\infty} s_t = s^*) \leq \sum_t E(Q_{x_t}(z_i^T z_{i-1} \geq 0 \text{ for all } i \geq 1)). \tag{15}$$

However, by standard manipulations (see [2, Problem 9, Chap. 5.6]), we get from Lemma 1 (ii) that for any $z_0 \in R^N$

$$\{z_i^T z_{i-1} < 0 \text{ i.o.}\} \supset \{z_i \in B(\bar{R}) \text{ i.o.}\} \text{ a.s.}$$

Hence by Lemma 1 (i), we conclude that $Q_{z_0}(z_i^T z_{i-1} < 0 \text{ infinitely often}) = 1$ for any z_0. This implies

$$Q_x(x_i^T z_{i-1} \geq 0 \quad \text{for all } i \geq 1) = 0 \quad \text{for any } x$$

and consequently $P(\lim_{t\to\infty} s_t = s^*) = 0$ for any s^*. □

LEMMA 3. *For any* $\epsilon > 0, x_t \in B(\epsilon)$ *infinitely often* (*in other words*, x_t *visits any neighborhood of zero infinitely often*).

Proof. Define for any $\gamma^* > 0$ the stopping times

$$\begin{aligned} \sigma &= \inf(t : \gamma_t \leq \gamma^*), \\ \tau &= \inf(t \geq \sigma : |x_t| < \epsilon). \end{aligned}$$

We have from Assumption 2 that for γ^* small enough for all $|x| > \epsilon$ and $\gamma \leq \gamma^*$

$$\begin{aligned} &\varphi(x)^T\nabla V(x) - \frac{\gamma}{2}(\varphi(x)^T M\varphi(x) + trMS_M) \\ &\geq \Big(\varphi(x)^T\nabla V(x) - \frac{\gamma}{2}(\varphi(x)^T M\varphi(x) + trMS_M)\Big) I(|x| > \bar{R}) \\ &\quad + \varphi(x)^T\nabla V(x) I(\epsilon < |x| \leq \bar{R}) - \gamma K \geq \delta(\epsilon) \end{aligned}$$

with $\delta(\epsilon) > 0$. Thus we obtain from (3) for all $t > \sigma$ (since $\{t > \sigma\}$ is $\mathcal{F}_{t-1}$ measurable)

$$\begin{aligned} E(V_t I(t \leq \tau)|\mathcal{F}_{t-1}) &\leq I(t-1 \leq \tau)E(V_t|\mathcal{F}_{t-1}) \\ &\leq V_{t-1}I(t-1 \leq \tau) - \gamma_t\varphi(x_{t-1})^T\nabla V_{t-1}I(t-1 \leq \tau) \\ &\quad + \frac{\gamma_t^2}{2}(\varphi(x_{t-1})^T M\varphi(x_{t-1}) + trMS_M)I(t-1 \leq \tau) \\ &\leq V_{t-1}I(t-1 \leq \tau) - \gamma_t\delta(\epsilon)I(t-1 \leq \tau), \end{aligned} \tag{16}$$

with $\delta(\epsilon) > 0$. Hence, taking expectation with respect to $\mathcal{F}_\sigma$ and summing up to τ, we obtain from (16)

$$\delta(\epsilon)E\left(\sum_{i=\sigma+1}^{\tau} \gamma_i|\mathcal{F}_\sigma\right) \leq V_\sigma.$$

Due to Lemma 2, $\sigma < \infty$ and hence $V_\sigma < \infty$. It is clear that

$$E\left(\sum_{i=\sigma+1}^{\tau} \gamma(i)|\mathcal{F}_\sigma\right) \leq E\left(\sum_{i=\sigma+1}^{\tau} \gamma_i|\mathcal{F}_\sigma\right) < \infty.$$

From the fact that $\sum_{i=1}^{\infty} \gamma(i) = \infty$, we conclude that $\tau < \infty$. □

For $\epsilon > 0$ small enough, let us define the stopping times

$$\begin{aligned} \tau &= \inf(t : V(x_t) > \epsilon), \\ \sigma_k &= \inf(t : s_t = k). \end{aligned} \tag{17}$$

LEMMA 4. *There exists $\delta_v > 0$ such that if $V(x_0) \leq \epsilon/2$ then $P(\tau < \infty) \leq K(\epsilon)\sum_{i=1}^{\infty} \gamma(i)^2$ for any $\epsilon < \delta_v$.*

Proof. Let us choose δ_v such that $|\varphi(x)| < \delta_1$ if $V(x) < \delta_v$ (δ_1 has been defined in (14)). From (3) we obtain by Assumption 2,

$$\begin{aligned} P(\tau < \infty) &\leq P\Bigg(-\sum_{i=0}^{\tau-1} \gamma_{i+1}\varphi(x_i)^T\nabla V(x_i) - \sum_{i=0}^{\tau-1} \gamma_{i+1}\nabla V(x_i)^T\xi_{i+1} \\ &\qquad + K\sum_{i=0}^{\tau-1} \gamma_{i+1}^2(|\varphi(x_i)|^2 + |\xi_{i+1}|^2) \geq \epsilon/2\Bigg) \\ &\leq P\Bigg(\left|\sum_{i=0}^{\tau-1} \gamma_{i+1}\nabla V(x_i)^T\xi_{i+1}\right| \\ &\qquad + K\sum_{i=0}^{\tau-1} \gamma_{i+1}^2(|\varphi(x_i)|^2 + |\xi_{i+1}|^2) \geq \epsilon/2\Bigg) \\ &\leq P\left(\left|\sum_{i=0}^{\tau-1} \gamma_{i+1}\nabla V(x_i)^T\xi_{i+1}\right| \geq \epsilon/4\right) \end{aligned} \tag{18}$$

$$+P\left(\sum_{i=0}^{\tau-1} \gamma_{i+1}^2(|\varphi(x_i)|^2 + |\xi_{i+1}|^2) \geq K\epsilon/4\right) = I_1 + I_2. \tag{19}$$

Define the martingale

$$M_t = \sum_{i=1}^{t\wedge\tau} \gamma_i\nabla V(x_{i-1})^T\xi_i$$

(where $t \wedge \tau = \min(t, \tau)$). Then by the Doob inequality, we have

$$I_1 \leq P(\sup_t |M_t| \geq \epsilon/4) \leq \frac{32}{\epsilon^2}E\left(\sum_{i=1}^{\infty} \gamma_i^2|\nabla V(x_{i-1})|^2|\xi_i|^2 I(i \leq \tau)\right)$$

$$\leq K/\epsilon^2 E\left(\sum_{i=1}^{\infty}\gamma_i^2 \sum_{|x|\leq\epsilon} |\nabla V(x)|^2 E(|\xi_i|^2 I(i\leq\tau)|\mathcal{F}_{i-1})\right)$$
$$\leq K(\epsilon)\sum_{i=1}^{\infty} E\gamma_i^2 I(i\leq\tau).$$

In an analogous way we get

$$I_2 \leq K'(\epsilon)\sum_{i=1}^{\infty} E\gamma_i^2 I(i\leq\tau).$$

Hence

$$P(\tau<\infty)\leq K\sum_{i=1}^{\infty} E\gamma_i^2 I(i\leq\tau). \tag{20}$$

Now we will show that we can substitute $\gamma(i)$ for γ_i in (20). When x_t is close to zero the noise ξ_t forces the s_t to increase regularly. Indeed, due to Assumption 1 there is $\mu > 0$ such that for all x small enough

$$\mu = \max_v P_x((\xi+u)^T v < 0), \qquad |u| < \delta_1. \tag{21}$$

Since the function $\varphi(\cdot)$ is continuous, we conclude from (21) that

$$\mu \leq \max_v Px((\xi+\varphi(x))^T v < 0)$$

as soon as $V(x) < \delta_v$. In other words,

$$P((\xi_t+\varphi(x_{t-1}))^T(\xi_{t-1}+\varphi(x_{t-2})) < 0|\mathcal{F}_{t-1}) \geq \mu$$

for $|x_{t-1}|$ such that $V(x_{t-1}) < \delta_v$. Hence

$$P(\{s_{t+1}-s_t>0\}\cap\{t\leq\tau\}|\mathcal{F}_{t-1}) \geq \mu I(t\leq\tau). \tag{22}$$

Define $\nu_k = \min(\sigma_k,\tau)$. Then we have

$$\sum_{i=1}^{\tau}\gamma_i^2 = \sum_{i=1}^{\tau}\gamma^2(s_i) \leq \sum_{k=0}^{\infty}\gamma^2(k)(\nu_{k+1}-\nu_k).$$

Next, for any $n\geq 0$ we obtain from (22)

$$\begin{aligned}&P(\nu_{k+1}-\nu_k\geq n|\mathcal{F}_{\nu_k})\\ &\quad = P(\{\text{there is no change of the gain coefficient on } n \text{ steps}\}\cap\{\nu_k+n\leq\tau\}|\mathcal{F}_{\nu_k})\\ &\quad \leq (1-\mu)^n,\end{aligned}$$

which implies that $E(\nu_{k+1}-\nu_k)\leq\mu^{-1}$. Therefore,

$$E\sum_{i=1}^{\tau}\gamma_i^2 \leq \sum_{k=0}^{\infty}\gamma^2(k)E(\nu_{k+1}-\nu_k) \leq \frac{1}{\mu}\sum_{k=0}^{\infty}\gamma^2(k);$$

hence

$$P(\tau < \infty) \le \frac{K(\epsilon)}{\mu}\sum_{k=0}^{\infty}\gamma^2(k). \qquad \square$$

LEMMA 5. $x_t \to 0$ *almost surely.*

Proof. Denote

$$A = \{|x_t| > \epsilon \text{ i.o.}\}. \tag{23}$$

Define the stopping time $\tau_k = \inf(t \ge \sigma_k : x_t \in B(\epsilon/2))$ with σ_k defined in (17). From Lemmas 2 and 3, we have that the sequence τ_k is strictly increasing and finite. The Markov property then implies that for all k

$$P(A) = P(A \cap \{\tau_k < \infty\}) \le E(I(\tau_k < \infty)P_{\tau_k}(A)) \le K(\epsilon)\sum_{i=k}^{\infty}\gamma_i^2.$$

Thus $P(A^c) = 1$. Due to the arbitrary choice of ϵ in (23), we obtain the desired proposition. $\square$

The objective of the following proposition is to obtain an estimate of the speed of convergence of s_t/t to its limit.

PROPOSITION 1. $s_t/t \to P^*(\xi_1^T\xi_2 < 0)$ *almost surely.*

Proof. Put

$$\Psi_x(a) = \max_{|u|=1} P_x(|u^T\xi| \le a).$$

Note that $\Psi_x(a)$ is the highest probability of a stripe of width $2a$ "centered in 0" under the conditional law of ξ. We use the decomposition

$$s_t = \sum_{i=1}^{t-1} s_{i+1} - s_i - I(\xi_i^T\xi_{i-1} < 0) + \sum_{i=1}^{t-1} I(\xi_i^T\xi_{i-1} < 0),$$

and setting $\varphi_i = \varphi(x_i)$, we obtain the following bound for the first term:

$$\begin{aligned}
&|s_{t+1} - s_t - I(\xi_t^T\xi_{t-1} < 0)|\\
&\quad = |I((\varphi_{t-1} + \xi_t, \varphi_{t-2} + \xi_{t-1}) < 0) - I(\xi_t^T\xi_{t-1} < 0)|\\
&\quad \le |I((\varphi_{t-1} + \xi_t, \varphi_{t-2} + \xi_{t-1}) < 0) - I((\xi_t, \varphi_{t-2} + \xi_{t-1}) < 0)|\\
&\qquad + |I((\xi_t, \varphi_{t-2} + \xi_{t-1}) < 0) - I(\xi_t^T\xi_{t-1} < 0)|\\
&\quad \le I(|(\xi_t, \varphi_{t-2} + \xi_{t-1})| < |(\varphi_{t-1}, \varphi_{t-2} + \xi_{t-1})|) + I(|\xi_t^T\xi_{t-1}| < |\xi_t^T\varphi_{t-2}|).\\
&\quad = u_t + v_t.
\end{aligned}$$

From the Neveu martingale theorem [7], we have

$$\begin{aligned}
\sum_{i=1}^{t-1} u_i &= \sum_{i=1}^{t-1} u_i - E(u_i|\mathcal{F}_{i-1}) + \sum_{i=1}^{t-1} E(u_i|\mathcal{F}_{i-1})\\
&= o(t^{1/2+\alpha}) + \sum_{i=1}^{t-1} \Psi_{x_{i-1}}(|\varphi_{i-1}|).
\end{aligned}$$

Let us estimate

$$E(v_t|\mathcal{F}_{t-2}) = E(P_{x_{t-1}} \otimes P_{x_{t-2}}(|\xi_t^T \xi_{t-1}| < |\xi_t^T \varphi_{t-2}|)|\mathcal{F}_{t-2}).$$

Substituting the law P_0 for $P_{x_{t-1}}$, we get

$$\begin{aligned} E(v_t|\mathcal{F}_{t-2}) &\le P_0 \otimes P_{x_{t-2}}(|\xi^T \xi_{t-1}| < |\xi^T \varphi_{t-2}|) + E(\|P_{x_{t-1}} - P_0\||\mathcal{F}_{t-2}) \\ &\le \Psi_{x_{t-2}}(|\varphi_{t-2}|) + (E(\|P_{x_{t-1}} - P_0\||\mathcal{F}_{t-2}) - \|P_{x_{t-1}} - P_0\|) + \|P_{x_{t-1}} - P_0\|. \end{aligned}$$

Using again the Neveu theorem, we obtain

$$\begin{aligned} \sum_{i=1}^{t-1} v_i &= \sum_{i=1}^{t-1} v_i - E(v_i|\mathcal{F}_{i-1}) + \sum_{i=2}^{t-1} E(v_i|\mathcal{F}_{i-1}) - E(v_i|\mathcal{F}_{i-2}) + \sum_{i=2}^{t-1} E(v_i|\mathcal{F}_{i-2}) + Ev_1 \\ &= o(t^{1/2+\alpha}) + \sum_{i=1}^{t-2} \Psi_{x_{i-1}}(|\varphi_{i-1}|) + \|P_{x_{i-1}} - P_0\|. \end{aligned}$$

Summing up, we have

$$\begin{aligned} s_t &= o(t^{1/2+\alpha}) + 2\sum_{i=1}^{t-1} \Psi_{x_{i-1}}(|\varphi_{t-1}|) \\ &\quad + \sum_{i=1}^{t-2} \|P_{x_{i-1}} - P_0\| + \sum_{i=1}^{t-1} I(\xi_i^T \xi_{i-1} < 0) \\ &= o(t^{1/2+\alpha}) + 2\sum_{i=1}^{t-1} \Psi_0(|\varphi_{i-1}|) + 3\sum_{i=1}^{t-1} \|P_{x_{i-1}} - P_0\| + tP^*(\xi_1^T \xi_2 < 0). \end{aligned}$$

Note that since $x_t \to 0$ and the function $\varphi(\cdot)$ is continuous, we derive that

$$\frac{1}{t}\sum_{i=1}^{t-1} \Psi_0(|\varphi(x_{i-1})|) + \frac{1}{t}\sum_{i=1}^{t-1} \|P_{x_{i-1}} - P_0\| \to 0.$$

Hence $s_t/t \to P^*(\xi_1^T \xi_2 < 0)$. □

3. Asymptotic normality of the SA procedure. Consider algorithm (3)–(5) with the special choice of the gain sequence: $\gamma(t) = \gamma t^{-1}$. Denote $\zeta^{-1} = P^*(\xi_1^T \xi_2 < 0)$. We shall show that the accelerated algorithm is asymptotically equivalent to the usual SA procedure with the gain $\gamma_t = \gamma\zeta t^{-1}$. Let us consider the following assumptions.

ASSUMPTION 2′. Assumption 2 holds. Moreover,

$$|\varphi(x) - \varphi'(x^*)(x - x^*)| = o(|x - x^*|).$$

The matrix $I/2 - \gamma\zeta\nabla\varphi(x^*)$ is Hurwitz, i.e., has all strictly negative eigenvalues.

ASSUMPTION 3′. $\gamma(t) = \gamma t^{-1}$ with $\gamma < \rho$ for ρ defined in Assumption 2.

THEOREM 2. *Let assumptions* 1′–3′ *hold. Then*

$$\begin{aligned} &x_t \to x^* a.s., \\ &\sqrt{t}(x_t - x^*) \xrightarrow{D} \mathcal{N}(0, V), \end{aligned}$$

where matrix V is a unique positive definite solution of the Lyapunov equation

$$\left(\gamma\zeta\nabla\varphi(x^*) - \frac{I}{2}\right)V + V\left(\gamma\zeta\nabla\varphi(x^*) - \frac{I}{2}\right)^T = (\gamma\zeta)^2 S(x^*). \tag{24}$$

In other words, normalized errors of algorithm (3)–(5) are asymptotically normal with zero mean and covariance matrix V.

Proof. Put $x^* = 0$. Note that as soon as all conditions of Theorem 1 hold

$$x_t \longrightarrow 0, \qquad s_t - \zeta^{-1}t \longrightarrow 0,$$

which means that $t\gamma_t - \zeta\gamma \longrightarrow 0$. The following simple lemma will be useful in further developments.

LEMMA 6. *Let P (v_t) be a random sequence of real numbers, such that $v_t \longrightarrow 0$ almost surely as $t \longrightarrow \infty$. Then there exists a deterministic sequence (a_t) such that*

$$a_t \longrightarrow 0 \quad \textit{and} \quad v_t/a_t \longrightarrow 0\, a.s.$$

Proof. Let us construct the sequence $w_t = \max\{|v_i|, i \geq t\}$. Obviously, (w_t) is decreasing and $w_t \overset{P}{\longrightarrow} 0$. Thus there exists a sequence (ϵ_t) such that $\epsilon_t > 0, \epsilon_t \longrightarrow 0$, and $P(w_t > \epsilon_t) < \epsilon_t$ as $t \longrightarrow \infty$. So, $w_t/\sqrt{\epsilon_t} \overset{P}{\longrightarrow} 0$. This means that there is a subsequence t_k of times such that $w_{t_k}/\sqrt{\epsilon_{t_k}} \longrightarrow 0$ almost surely. Let us define a sequence (a_j) in the following way:

$$a_j = \sqrt{\epsilon_{t_k}} \quad \text{for } t_k \leq j < t_{k+1}.$$

Then we have for all $j \geq 1$

$$|v_j|/a_j \leq w_j/a_j \leq w_{t_k}/\sqrt{\epsilon_{t_k}} \longrightarrow 0 \quad \text{as } j \longrightarrow \infty. \qquad \square$$

Theorem 1, along with Lemma 6, yields that there exists a sequence (a_t) of nonrandom positive numbers such that

$$a_t \longrightarrow 0 \quad \text{and} \quad (\gamma\zeta - t\gamma_t)/a_t \longrightarrow 0, \qquad x_t/a_t \longrightarrow 0 \text{ a.s.} \tag{25}$$

Let us define the stopping times

$$\tau_R = \inf\{t : |\gamma\zeta - t\gamma_t| \geq R|a_t|\}, \qquad \sigma_R = \inf\{t : |x_t| \geq R|a_t|\} \tag{26}$$

for $\alpha > 0$ and $\nu = \min(\tau_R, \sigma_R)$. From Lemma 6 and (25) we conclude that for any $\epsilon > 0$ one can choose $R < \infty$ such that

$$P(\nu = \infty) \geq 1 - \epsilon. \tag{27}$$

Consider along with the process (3)–(5) a new linearized process z_t, which is defined by the equation

$$z_t = z_{t-1} - \frac{\gamma\zeta}{t}(\varphi'(0)z_{t-1} + \xi_t), \qquad z_0 = x_0. \tag{28}$$

Asymptotic properties of this process have been completely studied. For example, all of the conditions of the Nevel'son–Khasminskij theorem [6] are satisfied; thus

$$\begin{aligned} &z_t t^{1/2-\alpha} \longrightarrow 0 \quad \text{for all } \alpha > 0 \quad \text{and} \quad E|z_t|^2 \leq \tfrac{K}{t}, \\ &\sqrt{t}z_t \overset{D}{\longrightarrow} \mathcal{N}(0, V), \end{aligned} \tag{29}$$

where the matrix V is defined in (24). Hence to prove the assertion of the theorem, it suffices to show asymptotic equivalence of the processes (x_t) and (z_t).

Denote $\Delta_t = x_t - z_t$.

PROPOSITION 2. $\sqrt{t}\Delta_t \xrightarrow{P} 0$.

Proof. For Δ_t, we have from (3) and (28)

$$\begin{aligned} \Delta_t &= \Delta_{t-1} - \frac{\gamma\zeta}{t}\varphi'(0)\Delta_{t-1} + \left(\frac{\gamma\zeta}{t} - \gamma_t\right)\varphi'(0)x_{t-1} \\ &\quad + \gamma_t(\varphi'(0)x_{t-1} - \varphi(x_{t-1})) + \left(\frac{\gamma\zeta}{t} - \gamma_t\right)\xi_t \end{aligned} \tag{30}$$

$$= \Delta_{t-1} - \frac{\gamma\zeta}{t}\varphi'(0)\Delta_{t-1} + |x_{t-1}|\frac{u_{t-1}}{t} + \left(\frac{\gamma\zeta}{t} - \gamma_t\right)\xi_t, \tag{31}$$

where u_t is an $\mathcal{F}_t$ measurable random variable satisfying

$$|u_t| \le \max\left\{|\varphi'(0)|R\mathrm{a}_t, (\gamma\zeta + R\mathrm{a}_t) \sup_{|x|\le R\mathrm{a}_t} (|\phi(x) - \phi'(0)x|/|x|)\right\} \overset{\text{def}}{=} b_t.$$

Note that $\lim_{t\to\infty} b_t = 0$. From Assumption $2'$ and the Lyapunov theorem, we conclude that there is a solution $A = A^T > 0$ of the Lyapunov equation

$$\left(\frac{I}{2} - \gamma\zeta\varphi'(0)\right)A + A\left(\frac{I}{2} - \gamma\zeta\varphi'(0)\right)^T = -I.$$

Thus we obtain

$$\gamma\zeta(A^T\varphi'(0) + \varphi'(0)^T A) \ge (1+\beta)A \tag{32}$$

for some $\beta > 0$. Let us put $V_t = \Delta_t^T A \Delta_t$. Using the inequality

$$(a+b+c+d)^2 \le a^2 + 3(b^2+c^2+d^2) + 2a(b+c+d),$$

we obtain from (31) for any $t \le \nu$

$$\begin{aligned} V_t &\le V_{t-1} + 3|A|(\gamma^2\zeta^2 t^{-2}|\varphi'(0)|^2|\Delta_{t-1}|^2 + |x_{t-1}|^2|u_{t-1}|^2 t^{-2} + R^2 a_t^2 t^{-2}|\xi_t|^2) \\ &\quad + t^{-1}(-(1+\beta)V_{t-1} + 2|\Delta_{t-1}||x_{t-1}||u_{t-1}| + 2(\gamma\zeta - t\gamma_t)\xi_t^T A\Delta_{t-1}) \\ &\le V_{t-1} + K(t^{-2}V_{t-1} + a_{t-1}t^{-2} + a_{t-1}t^{-2}|\xi_t|^2) \\ &\quad + t^{-1}(-(1+\beta)V_{t-1} + 4(|\Delta_{t-1}|^2 + |z_{t-1}|^2)b_{t-1} + 2(\gamma\zeta - t\gamma_t)\xi_t^T A\Delta_{t-1}) \\ &\le V_{t-1}\left(1 - \frac{1+\beta/2}{t}\right) + Ka_{t-1}t^{-2} + a_{t-1}t^{-2}|\xi_t|^2 \\ &\quad + 2|z_{t-1}|^2 b_{t-1}/t + 2(\gamma\zeta/t - \gamma_t)\xi_t^T A\Delta_{t-1} \end{aligned}$$

if t is large enough. And now, taking expectations on both sides, we obtain

$$EV_t I(t<\nu) \le EV_t I(t-1<\nu) \le \left(1 - \frac{1+\beta/2}{t}\right) EV_{t-1}I(t-1<\nu) + o(t^{-2}).$$

Therefore, we get for $W_t = tV_t I(t<\nu)$

$$E(W_t|\mathcal{F}_{t-1}) \le \left(I - \frac{\beta/2}{t-1}\right)W_{t-1} + o(t^{-1}).$$

Hence $EW_t \to 0$ and $\sqrt{t}\Delta_t I(t<\nu) \xrightarrow{P} 0$ for any value of R. Due to the arbitrary choice of ϵ in (27) we obtain the desired proposition. □

4. Algorithm with averaging of trajectories. Let us consider the Polyak–Ruppert algorithm [8], [12] for the stochastic approximation problem:

$$\left\{ \begin{array}{l} x_t = x_{t-1} - \gamma_t y_t, \\ \bar{x}_t = \frac{1}{t}\sum_{i=0}^{t-1} x_i, \qquad x_0 \in R^N, \end{array} \right. \tag{33}$$

$$y_t = \varphi(x_{t-1}) + \xi_t$$

with the sequence of scalar gain coefficients γ_t defined by (4) and (5).

The first equation of (33) along with (4) and (5) constitutes an accelerated stochastic approximation algorithm that is analogous to that considered in §2. The averaging in (33) ensures the asymptotical optimality of the method (see [8] for details). We impose the following assumptions:

ASSUMPTION 5. There exists a function $U(x) : R^N \to R^+$ such that for some $\kappa > 0, \alpha > 0, \epsilon > 0, L > 0$ and any $x, y \in R^N$, the following conditions hold:

$$\begin{array}{l} U(x) \geq \alpha |x|^2, \\ |\nabla U(x) - \nabla U(y)| \leq L|x-y|, \\ U(x^*) = 0, \qquad \nabla U(x)^T \varphi(x) > 0 \quad \text{for } x \neq x^*, \\ \nabla U(x)^T \varphi(x) \geq \kappa U \quad \text{for } |x - x^*| < \epsilon. \end{array}$$

ASSUMPTION 6. There exists a matrix $\varphi'(x^*) > 0$ and $K_\varphi < \infty, 0 < \lambda \leq 1$ such that

$$|\varphi(x) - \varphi'(x^*)(x - x^*)| \leq K_\varphi |x - x^*|^{1+\lambda}.$$

ASSUMPTION 7. $\gamma(t) = \gamma t^{-\mu}$ with $\gamma > 0$ and $(1+\lambda)^{-1} < \mu < 1$.

Comment. In fact, Assumptions 2 and 5 declare the existence of two Lyapunov functions for the system. The probe function U in condition 5 describes the local properties of the function $\varphi(\cdot)$ in the neighborhood of x^*, and V declared in Assumption 2 is, in turn, a "global" one that guarantees the global stability of the system.

THEOREM 3. *If conditions* 1–7 *are satisfied then*

$$\begin{array}{l} \bar{x}_t \to x^* a.s., \\ \sqrt{t}(\bar{x}_t - x^*) \xrightarrow{D} \mathcal{N}(0, V), \end{array}$$

where

$$V = \varphi'(x^*)^{-1} S(x^*)(\varphi'(x^*)^{-1})^T.$$

Proof. We will verify the assumptions of Theorem 2 in [8]. Assumptions 1, 5, and 6 ensure that conditions 3.1–3.4 of Theorem 2 in [8] hold. It suffices to show that

$$\sum_{i=1}^{\infty} \gamma_t^{(1+\lambda)/2} t^{-1/2} < \infty. \tag{34}$$

Note that all of the conditions of Theorem 1 are satisfied; thus

$$\frac{s_t}{t} - \zeta^{-1} \to 0.$$

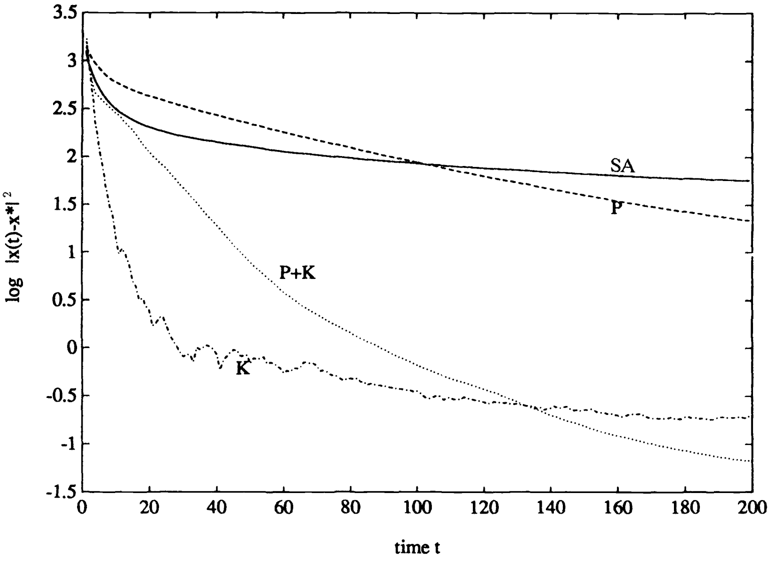


FIG. 1

This means that there are $\alpha > 0$ and $t_\alpha < \infty$ such that $s_t \geq \alpha t$ for $t \geq t_\alpha$. Thus we obtain by Assumption 7

$$\begin{aligned}\sum_{i=1}^{\infty} \gamma_i^{(1+\lambda)/2} i^{-1/2} &= \sum_{i=1}^{t_\alpha} \gamma_i^{(1+\lambda)/2} i^{-1/2} + \sum_{i=t_\alpha+1}^{\infty} \gamma_i^{(1+\lambda)/2} i^{-1/2} \\ &\leq K + \sum_{i=t_\alpha+1}^{\infty} (i\alpha)^{-\mu(1+\lambda)/2} i^{-1/2} \\ &\leq K + K \sum_{i=t_{\alpha+1}}^{\infty} i^{-1+\alpha'}\end{aligned}$$

for some $\alpha' > 0$. Hence the series (34) is summable. □

5. Numerical examples. Consider a stochastic approximation problem for the vector field in R^2

$$\varphi(x) = \left(\frac{x_1 - x_1^*}{1 + \sqrt{|x - x^*|}}, \frac{8(x_2 - x_2^*)}{1 + \sqrt{|x - x^*|}} \right)^T$$

with disturbances $\xi_t \in R^2$ that are independent and identically distributed Gaussian random variables with zero mean and covariance

$$S(x^*) = \begin{bmatrix} 1.0 & 0 \\ 0 & 1.0 \end{bmatrix}.$$

The initial error is $x_0 - x^* = (20, 20)^T$.

The trajectories of the *logarithm* of the error variance averaged by 10 samples for the ordinary SA algorithm, the Polyak–Ruppert algorithm (P), and their accelerated versions (K and P + K, respectively) are presented in Fig. 1. First we compare algorithm (3)–(5) with $\gamma(t) = t^{-1}$ to the ordinary stochastic approximation algorithm

$$x_t = x_{t-1} - \frac{\gamma}{t}(\varphi(x_{t-1}) + \xi_t). \tag{35}$$

In this example the accelerated algorithm (K) significantly outperforms the ordinary one (SA). Next we can compare this behavior to that of the Ruppert–Polyak algorithm. ($\gamma(t) = t^{-0.6}$ was arbitrarily chosen for the first equation of the Ruppert–Polyak method (33).) We can see that the Ruppert–Polyak algorithm (P) and its Kesten-like modification (P + K) asymptotically outperform their ordinary counterparts (algorithms without averaging of the trajectories).

REFERENCES

[1] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations,* Springer-Verlag, Berlin, 1990.

[2] L. Breiman, *Probability,* Addison-Wesley, Reading, MA, 1968.

[3] V. Fabian, *On asymptotically efficient recursive estimation,* Ann. Statist., 6(1978), pp. 854–866.

[4] H. Kesten, *Accelerated stochastic approximation,* Ann. Math. Statist., 29(1958), pp. 41–59.

[5] H. J. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems,* Springer-Verlag, New York, 1978.

[6] M. B. Nevel'son and R. Z. Khas'minskij, *Stochastic Approximation and Recursive Estimation,* American Mathematical Society, Providence, RI, 1973.

[7] J. Neveu, *Martingales á Temps Discret,* Masson, Paris, 1972.

[8] B. T. Polyak and A. B. Juditsky, *Acceleration of stochastic approximation by averaging,* SIAM, J. Control Optim., 29(1991), pp. 838–855.

[9] B. T. Polyak and Ya. Z. Tsypkin, *Adaptive estimation algorithms* (*convergence, optimality, stability*), Automation and Remote Control, 40 (1980), pp. 378–389.

[10] H. Robbins and S. Monro, *A stochastic approximation method,* Ann. Math. Statist., 22(1951), pp. 400–407.

[11] H. Robbins and D. Siegmund, *A convergence theorem for nonnegative almost supermartingales and some applications,* in Optimizing Method in Statistics, J. S. Rustaji, ed., Academic Press, New York, pp. 233–257.

[12] D. Ruppert, *Efficient Estimators from a Slowly Convergent Robbins–Monro Process,* Tech. Rep. N. 781, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1985.

[13] Ya. Z. Tsypkin, *Foundations of Informational Theory of Identification,* Nauka, Moscow, 1984. (In Russian.)

[14] J. H. Venter, *An extension of the Robbins–Monro procedure,* Ann. Math. Statist., 38(1967), pp. 181–190.

[15] M. Wasan, *Stochastic Approximation,* Cambridge University Press, London, 1970.

SIAM J. OPTIMIZATION
Vol. 3, No. 4, pp. 882–900, November 1993

PARALLEL PROJECTED AGGREGATION METHODS FOR SOLVING THE CONVEX FEASIBILITY PROBLEM*

UBALDO GARCÍA-PALOMARES†

Abstract. Convergence conditions are established for new sequential and parallel projected aggregation methods (PAMs) that find a feasible point of a large system of convex inequalities and linear equations. To formulate a multiprocessor method suitable for solving a nonstructured convex system, block iterative methods are used and all system constraints are simultaneously processed. Each processor is assigned the task of finding closer points to one block subsystem, so that at every iteration each processor proposes a point closer (in some norm) to a group of the system constraints, and a head processor combines the proposals and generates a point closer to the original system. These parallel versions appear amenable to multiprocessing. Numerical results are reported that give hints on how to code these methods in a multiprocessor environment.

Key words. convex systems, convex feasibility problem, parallel processing, projected aggregation methods

AMS subject classifications. 52A41, 52A40, 65A05

1. Introduction.

We consider the convex feasibility problem of finding a real vector of n components in a set S defined as

$$S := \left\{ x \in R^n | x \in \bigcap_{i=1}^{p} S_i \right\}, \tag{1.1a}$$

where

$$S_i := \begin{cases} \{x \in R^n | g^i(x) \le b^i\} & \text{if } i = 1, \dots, m, \\ \{x \in R^n | g^i(x) = b^i\} & \text{if } i = m+1, \dots, p, \end{cases} \tag{1.1b}$$

$g^i(x), i = 1, \dots, m$ are convex subdifferentiable functions and $g^i(x) := a_i * x, i = m + 1, \dots, p$ are linear functions. We always assume that the first m constraints defining the set S are convex (or linear) inequalities, and that the last p–m constraints are linear equalities.

Our concern is to propose sequential and parallel methods to solve the convex feasibility problem, when both the number of constraints p and the number of variables n are large. We mainly report on mathematical, not experimental, results. Conclusive computational assessments of the methods presented here require the solution of the convex feasibility problem on different multiprocessor architectures, a subject that remains open for future research.

A good review of iterative methods for solving (1.1) can be found in [5], [6], and references therein. Many of these methods coincide when they solve a system of linear (in)equalities, and can be considered as an outgrowth of the wealth of research stemming from work by Cimmino [10] and Kaczmarz [26], who proposed iterative algorithms for solving a linear system of equations by cyclically projecting on the hyperplane defined by one equation. Agmon [1] and Motzkin and Schoenberg [28] used the same approach for solving a system of linear inequalities. They proposed an algorithm that successively projects on the supporting hyperplane of the convex set defined by one inequality. Apparently these algorithms are robust but sometimes very slow, and many researchers have analyzed block iterative versions to improve convergence [3], [4], [6], [13], [14], [17]–[20], [22], [24], [27], [33].

Our objective is to apply and analyze the projected aggregation methods (PAMs) in the solution of the convex feasibility problem. These methods generate a new iterate as the (under/over) projection of the previous iterate on a hyperplane defined by a suitable aggregation of linearizations of appropriate constraints. To our knowledge, Householder and Bauer [24]

* Received by the editors June 26, 1991; accepted for publication (in revised form) August 25, 1992.

† Universidad Simón Bolívar, Departmento de Procesos y Sistemas, Apartado 89000, Caracas, 1086 A, Venezuela.

were the first researchers who proposed a PAM for solving a linear system of equations, but it seems that their method went unnoticed. García-Palomares [20] gave conditions to enable parallel processing for PAMs on structured linear systems and extended those results for systems of convex inequalities [19]. In this paper we complement these and previous results on PAMs [17], [18] and apply these methods to the problem of finding a feasible point of a convex system defined by the intersection of convex subdifferentiable inequalities and linear equalities. We show that the proposed algorithms for solving this convex feasibility problem inherit most of the properties of the PAM for the linear case, and fit nicely into a parallel processing procedure for solving structured convex systems. In addition, we show that under appropriate conditions our algorithms become a particular instance of an underlying scheme proposed by Aharoni, Berman, and Censor [2], who use projections onto hyperplanes that separate an iterate from the convex system.

We also present novel methods that allow a high degree of parallelism even for nonstructured convex systems. Instead of trying to find a feasible point to a large system directly, we split it into smaller subsystems and devise a block-iterative projection method. We then assign one processor to each block. To ensure convergence to a feasible point of the large system, a head processor reconciles all processors' actions. We present some preliminary numerical experiments for linear systems with the mere purpose of providing some ways of coding these methods in a multiprocessing environment.

In our notation lowercase Greek (un)subscripted letters denote real numbers, lowercase Latin (un)subscripted letters denote vectors, and uppercase Latin unsubscripted letters denote matrices or sets: a^i is the ith component of the vector a; $a * b$ is the inner product of vectors a, b; A^T is the transpose of the matrix A; $\partial g(.)$ denotes a subgradient of the function $g(.)$; $v_+ \in R^p_+$, represents a vector point (or a vector function) whose p components are given as

$$v^i_+ = \begin{cases} \max\{0, v^i\} & \text{if } i = 1, \dots, m, \\ |v^i| & \text{if } i = m+1, \dots, p. \end{cases}$$

We denote $\sigma(x, S)$ as a distance from the point $x \in R^n$ to the set S. We assume that $\sigma(x, S)$ satisfies the triangle inequality, that is, $\sigma(x, S) \leq \sigma(x, y) + \sigma(y, S)$ for any $x, y \in R^n$. An infinite sequence will be represented by $\{.\}_{k=0}^{\infty}$, but to simplify the notation $\{.\} \to 0$ means that the sequence $\{.\}_{k=0}^{\infty}$ converges to 0. The rest of the notation is rather standard and we hope that it can be understood from the context.

The paper is organized as follows. Section 2 describes some variants of the sequential PAMs that include practical ways to aggregate constraints and to reduce the computational work required. Section 3 describes variations of PAMs for solving structured systems. We also introduce novel methods that are quite amenable to parallel processing. Section 4 shows preliminary numerical experiments with some methods presented here.

2. PAMs.

Householder [24, p. 100] gives the following iterative PAM for solving the problem of finding $z \in (x \in R^n | Ax = b)$, where A is a nonsingular real $n \times n$ matrix, and $b \in R^n$.

```
$k = 0$.
Choose $x_0$
Until convergence do
    Choose some no-null vector $u_k \in R^n$
        $x_{k+1} = x_k - \frac{u_k * (Ax_k - b)}{A^T u_k * A^T u_k} A^T u_k$
        $k = k + 1$
```

```
End Do
```

Householder obtained that

$$\|x_{k+1} - z\|^2 = \|x_k - z\|^2 - \frac{\left(u_k * (Ax_k - b)\right)^2}{A^T u_k * A^T u_k}, \tag{2.1}$$

and proved that the choice of $\{u_k\}_{k=0}^{\infty}$ is crucial in determining the rate of convergence of the algorithm. Specifically, he obtained that

$$\|x_{k+1} - z\| / \|x_k - z\| \leq \frac{(\kappa - \kappa^{-1})^2}{(\kappa + \kappa^{-1})^2} \quad \text{if } u_k = Ax_k - b, \tag{2.2a}$$

$$\|x_{k+1} - z\| / \|x_k - z\| \leq 1 - \frac{1}{n\kappa^2} \quad \text{if } u_k = \max_{1 \leq i \leq n} |(Ax_k - b)^i|, \tag{2.2b}$$

where κ is the condition number of A. Below, we state general convergence conditions (mainly on $(u_k)_{k=0}^{\infty}$) for the more general convex feasibility problem (1.1). Surprisingly, we obtain formulas similar to (2.1).

Let $g(x) \in R^p$ be a vector of p components whose ith component is $g^i(x)$. Let $A(x)^T$ be the $n \times p$ matrix

$$A(x)^T := (\partial g^i(x), \ldots, \partial g^p(x)).$$

With a minor abuse of notation and when no confusion is possible, we denote $A := A(x)$. Given $0 < \eta \leq 1$ and a symmetric real positive definite $n \times n$ matrix M, the PAM's iterative procedure is as follows.

```
GENERAL SCHEME FOR PAMs. Choose $x \in R^n$.
Iteration
    Until convergence do
        Choose $u \in R^p$ to ensure convergence ((C3)–(C5) below)
(2.3a)      $d := -M^{-1}A^T u$
(2.3b)      $\lambda := \frac{u * (g(x) - b)}{d * Md}$
(2.3c)      $\eta \leq \omega \leq 2 - \eta$
(2.3d)      $x = x + \omega\lambda d$
    end do
End of the scheme for PAM
```

In the remainder, $\{x_k, u_k, \omega_k, d_k, \lambda_k\}_{k=0}^{\infty}$ is a sequence generated by the iteration loop of the general scheme for PAMs and we assume that conditions (C1)–(C6) given next hold, where ρ is a scalar that generally depends on the initial estimate x_0, and where $0 < \eta \leq 1$.

(C1) $S \neq \emptyset$

(C2) $\partial g^i(.), i = 1, \ldots, p$, are uniformly bounded in

$$X_0 := \left\{x \in R^n \mid \|x - z\|_M^2 \leq \rho^2, z \in X\right\},$$

where

$$X := \left\{x \in S \mid \|x_0 - x\|_M^2 \leq \rho^2\right\}.$$

(C3) $\{u_k\}_{k=0}^{\infty}$ is a uniformly bounded sequence.
(C4) $\forall_k : u_k^i \geq 0$ for $i \leq m$.
(C5) $\forall_k : u_k * (g(x_k) - b) \geq 0$.
(C6) $\forall_k : 0 < \eta \leq \omega_k \leq 2 - \eta$.

It is easy to deduce that the expressions for d and λ, given by (2.3a) and (2.3b), respectively, are nothing but the explicit evaluations of the projection of x on the hyperplane $H(u, x)$ defined by (2.4) below; therefore, PAMs are methods where the new point is the (under/over) projection of the point x on the hyperplane $H(u, x)$ defined by the aggregation of the linearized constraints, namely,

$$H(u,x) := \left\{ y \in R^n \middle| \sum_i u^i(g^i(x) + \partial g^i(x) * (y - x)) = \sum_i u^i b^i \right\} \tag{2.4a}$$

or, equivalently,

$$H(u,x) := \{y \in R^n | u * (g(x) + A(x)(y - x)) = u * b\}. \tag{2.4b}$$

At the kth iteration PAMs solve the easy problem

$$\min_{y \in H(u_k, x_k)} \|y - x_k\|_M^2, \tag{2.5}$$

where x_k is the estimate of some feasible point, u_k are weights that satisfy (C3)–(C5), $\|y - x_k\|_M^2 := (y - x_k) * M(y - x_k)$, and M is a symmetric strictly positive definite $n \times n$ matrix.

If $\mathcal{P}x_k$ is the solution to 2.5, the next estimate x_{k+1} is given as

$$x_{k+1} = x_k + \omega_k(\mathcal{P}x_k - x_k) \quad \text{for } \eta \leq \omega_k \leq 2 - \eta, \tag{2.6}$$

and, as we shall prove,

$$\|x_{k+1} - z\|_M^2 \leq \|x_k - z\|_M^2 - \omega_k(2 - \omega_k)\|\mathcal{P}x_k - x_k\|_M^2 \quad \text{for all } z \in X.$$

It is pertinent to point out [20] that the PAM is an underlying scheme for well-known methods (Craig's conjugate gradient [11], Polyak [31], Oettli [29], and others) that differ from each other in the choice of the sequence $\{u_k\}_{k=0}^{\infty}$. Likewise, block-iterative procedures, which have been successfully used for solving large linear systems (see [3], [4], and references therein), belong to the family of PAMs [18] and their convergence is easily deduced from [19, Thm. 3.2]. Illustrative applications and experiments on linear systems reveal that convergence improves with a suitable choice of the sequence of weights $\{u_k\}_{k=0}^{\infty}$, which aggregates several constraints at every iteration; even finite termination can be achieved in some cases [18]. In Fig. 1 the dashed line shows the well-known zigzag effect of Agmon [1] and Motzkin and Schoenberg's [28] projection method (AMS) when the set S is defined by the intersection of two inequalities that shape a small angle wedge. The method forces the alternate projection on the supporting hyperplanes that originates the zigzag. The PAM prevents the zigzagging by projecting on the hyperplane H. We should note that the primal-dual projection method due to Spingarn [32] possesses a natural acceleration feature through the use of "dual" variables when the iterates get caught in small angles; however, for large problems the amount of storage required is prohibitive. It needs to keep p dual vectors in R^n.

Another significant feature of the PAM for the convex feasibility problem is that the projection 2.5 is performed on a hyperplane $H(u, x)$, as opposed to previous methods that converge if the projection is performed on the convex sets $S_i, i = 1, \ldots, p$ [21], or if boundary points of the convex sets are computed [2], [5].

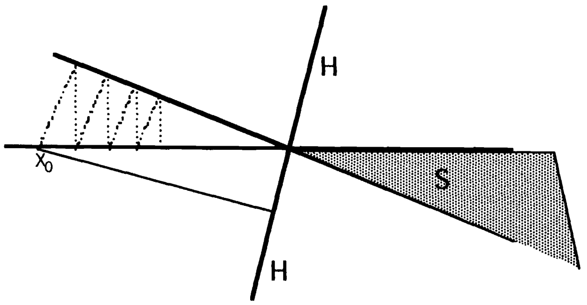


FIG. 1. AMS *projection versus aggregation.*

We now turn our attention to the convergence of PAMs.

Condition (C1), the nonemptiness of S, implies that given any initial point x_0, there exists some ρ^2 (generally dependent upon x_0) such that

$$X := \{x \in S | \; \|x_0 - x\|_M^2 \leq \rho^2\} \neq \emptyset.$$

In our convergence analysis we look at PAMs as methods of variational type. If $X \neq \emptyset$, PAMs generate a Fejer minimizing sequence with respect to the set X of the quadratic functional $f : R^n \times X \rightarrow R$ defined as

$$(2.7) \qquad f(x,z) = \|x - z\|_M^2 = (x - z) * M(x - z), \qquad x \in R^n, \quad z \in X.$$

We now show that for one iteration of the PAM given by (2.3), we obtain that $f(x_{k+1}, z) \leq f(x_k, z)$ for all $z \in X$; that is, the next iterate x_{k+1} is closer to all $z \in X$ than the previous iterate x_k. We start with the following lemma.

LEMMA 2.1. *Let $x \in R^n$ and $u \in R^p$. If $u^i \geq 0$ for $i \leq m$, if $u * (g(x) - b) \geq 0$, if $0 < \omega < 2$, if $d = -M^{-1}A^T u$, if*

$$\lambda = \frac{u * (g(x) - b)}{d * Md} \geq 0 \quad \textit{and} \quad x_1 = x + \omega\lambda d,$$

then

$$\|x_1 - z\|_M^2 \leq \|x - z\|_M^2 - \omega(2 - \omega)\lambda u * (g(x) - b) \quad \textit{for all } z \in X.$$

Proof. Let $z \in X$. By straightforward algebraic manipulation, we have

$$\begin{aligned} \|x_1 - z\|_M^2 &= \|x - \omega\lambda M^{-1}A^T u - z\|_M^2 \\ &= \|x - z\|_M^2 - 2\omega\lambda A^T u * (x - z) + \omega^2\lambda^2 d * Md \\ &= \|x - z\|_M^2 + 2\omega\lambda u * (g(x) + A(z - x)) - 2\omega\lambda u * g(x) + \omega^2\lambda^2 d * Md. \end{aligned}$$

As $z \in S$ we deduce by convexity and by the definitions of λ, u that

$$\begin{aligned} \|x_1 - z\|_M^2 &\leq \|x - z\|_M^2 + 2\omega\lambda u * g(z) - 2\omega\lambda u * g(x) + \omega^2\lambda^2 d * Md \\ \text{(2.8a)} \qquad &\leq \|x - z\|_M^2 - 2\omega\lambda u * (g(x) - b) + \omega^2\lambda^2 d * Md \\ &= \|x - z\|_M^2 - \omega(2 - \omega)\lambda u * (g(x) - b), \end{aligned}$$

which shows the validity of the lemma.

The last inequality can be rewritten as

$$\begin{aligned}
&(2.8\text{b}) && \|x_1 - z\|_M^2 \leq \|x - z\|_M^2 - \omega(2-\omega)\lambda^2 d * Md \\
&(2.8\text{c}) && \qquad = \|x - z\|_M^2 - \omega(2-\omega)\lambda^2 A^T u * M^{-1} A^T u \\
&(2.8\text{d}) && \qquad = \|x - z\|_M^2 - \frac{\omega(2-\omega)}{\omega^2}\|x_1 - x\|_M^2 \\
&(2.8\text{e}) && \qquad = \|x - z\|_M^2 - \omega(2-\omega)\|\mathcal{P}_x - x\|_M^2.
\end{aligned}$$

Relation (2.8a) for the convex feasibility problem resembles relation (2.1) derived by Householder [23] for a linear system of equations; therefore, we should expect that the choice of $\{u_k\}_{k=0}^{\infty}$ will influence the rate of convergence of the PAM. Some suggestions for $\{u_k\}_{k=0}^{\infty}$ are given by [19], [30], and [33]. Lemma 2.2 below gives an interpretation of the PAM when the system lacks inequalities ($m = 0$). In this case, given $x \in R^n$ and $d \in R^n$, the PAM will locate the minimum of $f(.,z)$, starting at x, and along the direction d. In other words, the PAM will locate the closest point to z.

LEMMA 2.2. *If* $m = 0$, *that is,* $S := \{x \in R^n | Ax = b\}$, *then given* $x \in R^n, d = -M^{-1}A^T u$, *and* $z \in S$

$$\arg\min_{\lambda} \|x + \lambda d - z\|_M^2 = \frac{u * (g(x) - b)}{d * Md}.$$

Proof. Let $x \in R^n$ and $d \in R^n$ be given and define

$$\vartheta(\lambda) := \|x + \lambda d - z\|_M^2 = \|x - z\|_M^2 + 2\lambda d * M(x - z) + \lambda^2 d * Md,$$

so that $\vartheta(\lambda)$ is a convex function and $\vartheta'(\lambda) = 0$ if and only if $\lambda = -(d * M(x - z))/d * Md$. The conclusion follows whenever $d = -M^{-1}A^T u$.

LEMMA 2.3. *Given* x_0 *let* $\{x_k\}_{k=0}^{\infty}$ *be the sequence generated by the* PAM, *and let*

$$\begin{aligned}
X &:= \left\{x \in S \,\middle|\, \|x_0 - x\|_M^2 \leq \rho^2\right\} \neq \emptyset, \\
X_0 &:= \left\{x \in R^n \,\middle|\, \|x - z\|_M^2 \leq \rho^2, z \in X\right\}.
\end{aligned}$$

If (C1)–(C2) *hold, then* $\{x_k\}_{k=0}^{\infty} \subseteq X_0$ *and* X_0 *is a compact convex set.*

Proof. By assumption, $x_0 \in X_0$. Besides, Lemma 2.1 shows that $\|x_{k+1} - z\|_M^2 \leq \|x_k - z\|_M^2$; thus, by induction, $\{x_k\}_{k=0}^{\infty} \subseteq X_0$. The compactness and the convexity of X_0 are obvious from its definition.

THEOREM 2.1 (convergence theorem). *Let* $\{x_k, u_k, \omega_k, d_k, \lambda_k\}_{k=0}^{\infty}$ *be generated by the* PAM. *If* (C1)–(C6) *hold, if* $\{x_k\}_{k=0}^{\infty} \notin S$, *and*

$$(2.9\text{a}) \qquad \{u_k * (g(x_k) - b)\} \to 0 \Rightarrow \{\sigma(x_k, S)\} \to 0,$$

or

$$(2.9\text{b}) \qquad \forall k : u_k * (g(x_k) - b) \geq \alpha\sigma(x_k, S) \quad \textit{for some } \alpha > 0,$$

then $\{\|x_{k+1} - x_k\|\} \to 0, \{\sigma(x_k, S)\} \to 0$, *and* $\{x_k\} \to \hat{x} \in S$.

Proof. Since (2.9b) $\Rightarrow$ (2.9a), we assume that (2.9a) holds. Let $z \in X$. The sequence $\{\|x_k - z\|_M^2\}_{k=0}^{\infty}$ is nonnegative and by (2.8) monotonically decreasing; therefore, it converges and

$$\{\omega_k(2 - \omega_k)\lambda_k u_k * \|(g(x_k) - b)\} \to 0;$$

TABLE 1

Typical choices of u_k. (The sign of u_k^i is chosen such that $u_k^i(g(x_k) - b)^i \geq 0$.)

$\sigma(x_k, S)$	u_k
$\|(g(x_k) - b)_+\|_\infty$	$u_k^i = \begin{cases} \pm 1 & \text{if } (g(x_k) - b)_+^i = \sigma(x_k, S), \\ 0 & \text{otherwise.} \end{cases}$
$\|(g(x_k) - b)_+\|_1$	$u_k^i = \begin{cases} \pm 1 & \text{if } (g(x_k) - b)_+^i > 0, \\ 0 & \text{otherwise.} \end{cases}$
$\|(g(x_k) - b)_+\|_2^2$	$u_k^i = \pm(g(x_k) - b)_+^i$.

therefore, by (C6) and the definition of $\{\lambda_k\}_{k=0}^\infty$, we obtain that

$$\left\{ \frac{(u_k * (g(x_k) - b))^2}{d_k * Md_k} \right\} \to 0. \tag{2.10}$$

By construction, we also have that

$$\|x_{k+1} - x_k\|_M^2 = \omega_k^2 \lambda_k^2 d_k * Md_k = \omega_k^2 \frac{(u_k * (g(x_k) - b))^2}{d_k * Md_k},$$

so we immediately conclude by (C6) and (2.10) that $\{\|x_{k+1} - x_k\|_M^2\} \to 0$, or equivalently, $\{\|x_{k+1} - x_k\|\} \to 0$. Moreover, from (C2) and (C3) we deduce that $\{u_k * (g(x_k) - b)\} \to 0$, and (2.9) implies $\{\sigma(x_k, S)\} \to 0$. Since $\{x_k\}_{k=0}^\infty$ is a Fejer sequence, we conclude by continuity that $\{x_k\}_{k=0}^\infty$ converges to some $\hat{x} \in X \subseteq S$.

It is known [2] that the sequence $\{x_k\}_{k=0}^\infty$ generated by (2.6) converges if the projection $\mathcal{P}x_k$ is performed on any hyperplane that separates S and a ball of center x_k and radius $\delta\sigma(x_k, S)$ with $0 < \delta \leq 1$. We now prove that under assumption (2.9b), the hyperplane $H(u_k, x_k)$ is a separation hyperplane. In this case, PAMs become a special instance of the (δ, η) algorithm.

LEMMA 2.4. *Under conditions* (C1)–(C6) *and under assumption* (2.9b), *the sequence* $\{x_k\}_{k=0}^\infty$ *converges to some point in* S.

Proof. Let $x \in B(x_k, \delta\sigma(x_k, S))$, a ball centered at x_k, and radius $\delta\sigma(x_k, S)$. Let $\beta = \sup_{x \in X_0} \|A(x)\|$ and $\gamma = \sup \|u_k\|$.

We have

$$u_k * (g(x_k) + A(x_k)(x - x_k)) \geq u_k * b + \alpha\sigma(x_k, S) - \beta\gamma\delta\sigma(x_k, S) > u_k * b \tag{2.11}$$

for δ sufficiently small.

On the other hand, if $x \in S$, then

$$u_k * (g(x_k) + A(x_k)(x - x_k)) \leq u_k * g(x) \leq u_k * b.$$

The last inequality and (2.11) show the separation property of $H(u_k, x_k)$ and the convergence follows from [2, Thm. 1]. When the sequence $\{u_k\}_{k=0}^\infty$ is chosen such that $u_k * (g(x_k) - b)$ becomes a predefined distance $\sigma(x_k, S)$, we identify several known methods for solving linear systems as instances of PAMs. In the convex case we can choose $\{u_k\}_{k=0}^\infty$ with the same feature, and the theoretical convergence results are clear from Theorem 2.1. Table 1 shows different definitions of $\sigma(x_k, S)$ and typical choices of u_k.

All of the choices for u_k depicted in Table 1 aggregate violated inequalities. These choices present at least two drawbacks: the possibility of zigzag and of costly evaluations of the distance function $\sigma(x_k, S)$ at all iterations. To prevent zigzag and hopefully enable a better performance of the algorithm, it should be advantageous to aggregate nonviolated inequalities. It is evident that we can aggregate any number of constraints, that is, $u_k^i \neq 0$ for several values of i, as long as (2.9) and conditions (C3)–(C6) are fulfilled. To prevent the costly evaluation of the distance function, we split the system into subsystems. We prove that convergence is preserved under similar conditions if at every iteration we only evaluate the distance function to a particular subsystem. Theorem 2.2 below and succeeding remarks give an outline on how to implement these ideas.

Hereafter, we assume that $P_1, \ldots, P_q$ are q index sets that exhibit a row splitting of the system S, and that $Y_1, \ldots, Y_q$ are the induced subsystems; that is,

$$\bigcup_{i=1}^{q} P_i = P := \{1, \ldots, p\}, \qquad Y_i := \bigcap_{j \in P_i} S_j, \quad i = 1, \ldots, q.$$

Following Censor and Lent [8], we define a control $\{i(k)\}_{k=0}^{\infty}$ to be almost cyclic on $\{1, \ldots, q\}$ if $1 \leq i(k) \leq q$ for all $k \geq 0$, and if

$$\forall j, \quad 1 \leq j \leq q \quad \exists(\text{a finite } t)\ \forall k \text{ such that} \quad i(k') = j \quad \text{for } k' \leq k + t.$$

For any given k, and a control $i(k)$, let u_k be a (bounded) vector with the following properties:

$$u_k^j \geq 0 \quad \text{for } j \leq m, \tag{2.12a}$$

$$u_k^j = 0 \quad \text{if } (g(x_k) - b)^j < -(\alpha/p)\sigma(x_k, Y_{i(k)}), \tag{2.12b}$$

$$\sum_{i \in P_{i(k)}} u_k^j (g(x_k) - b)^j \geq \alpha\sigma(x_k, Y_{i(k)}). \tag{2.12c}$$

THEOREM 2.2 *Let* $\cup_{i=1}^{q} P_i = P := \{1, \ldots, p\}$, *let* $\{i(k)\}_{k=0}^{\infty}$ *be almost cyclic on* $\{1, \ldots, q\}$, *and let* $\{u_k\}_{k=0}^{\infty}$ *be a set of (uniformly bounded) vectors chosen by* (2.12). *If* $\{x_k\}_{k=0}^{\infty}$ *is the sequence generated by the* PAM *and if* (C1)–(C3) *and* (C6) *hold, then* $\{x_k\}_{k=0}^{\infty}$ *converges to some* $\hat{x} \in X$.

Proof. It is evident from (2.12) that $\{u_k\}_{k=0}^{\infty}$ satisfies condition (C4). Condition (C5) also holds because

$$u_k * (g(x_k) - b) \geq \left[\alpha - \frac{p-1}{p}\alpha\right] \sigma(x_k, Y_{i(k)}) = (\alpha/p)\sigma(x_k, Y_{i(k)}) \geq 0. \tag{2.13}$$

If we mimic the convergence proof of Theorem 2.1, we deduce from (C1)–(C6) that

$$\{u_k * (g(x_k) - b)\} \to 0, \tag{2.14a}$$

$$\{\|x_{k+1} - x_k\|\} \to 0. \tag{2.14b}$$

Also, by definition,

$$0 \leq \sigma\left(x_k, Y_{i(k')}\right) \leq \sigma\left(x_k, x_{k'}\right) + \sigma(x_{k'}, Y_{i(k')})\,. \tag{2.15}$$

Let $j = i(k)$. By hypothesis, given any $r \in \{1, \ldots, q\}$ there exists a finite $t, k + t \geq k' > k$ such that $r = i(k')$. From (2.14b) we obtain that $\sigma(x_k, x_{k'}) \to 0$. From (2.13) and (2.14a) we also deduce that $\sigma(x_{k'}, Y_{i(k')}) \to 0$. Therefore, from (2.15) we obtain that $\sigma(x_k, Y_{i(k')}) \to 0$, that is, $\sigma(x_k, Y_r) \to 0$; but since r was arbitrary we infer that

$$\lim_{k \to \infty} \sigma(x_k, Y_r) = 0 \quad \text{for } r = 1, \ldots, q.$$

By continuity arguments we infer that $\sigma(x_k, S) \to 0$, which shows that any accumulation point of $\{x_k\}_{k=0}^{\infty}$ belongs to S. But a Fejer sequence has a unique accumulation point $\hat{x}$; therefore, $\{x_k\} \to \hat{x} \in X$. The proof is complete.

Before we end this section, it is useful to state the following remarks.

Remark 2.1. The theorem remains valid with no modification even if $\sigma\left(x_k, Y_{i(k)}\right) = 0$ for several values of k.

Remark 2.2. A particular case of the theorem is the cyclic subgradient projections method due to [8], where $Y_i = S_i := \{x \in R^n | x \text{ satisfies the } i\text{th constraint}\}, i(k) = k \bmod(p) + 1$, and

$$u_k^j = \begin{cases} 0 & \text{if } j \neq i(k), \\ 1 & \text{if } j = i(k), \quad j \leq m, \\ \operatorname{sgn}(g(x_k) - b)^j & \text{otherwise.} \end{cases}$$

If S is a linear system, this choice for $\{u_k\}_{k=0}^{\infty}$ gives rise to the well-known method of Agmon [1] and Motzkin and Schoenberg [28].

Remark 2.3. We can aggregate all the constraints; for instance, we may choose u_k^j for $j \neq i(k)$ as

$$u_k^j = \frac{-(\alpha/p)\sigma\left(x_k, Y_{i(k)}\right)}{(g(x_k) - b)^j} \quad \text{if} \begin{cases} \text{either } (g(x_k) - b)^j < 0 \quad \text{and} \quad j \leq m, \\ \text{or } j > m. \end{cases}$$

To ensure uniform boundedness, we reject the value given by the previous equality when a pre-established bound for u_k^j is exceeded. It is not prudent, however, to enforce the aggregation of all the constraints. It seems intuitively relevant to aggregate only those constraints that are violated or nearly satisfied at x_k. It also seems convenient to enforce that

$$u_k^j (g(x_k) - b)^j \geq 0 \quad \text{for } j > m.$$

Remark 2.4. Convergence of the block-splitting approach can be proved if $\{u_k\}_{k=0}^{\infty}$ is defined as follows:

$$\forall_k : u_k^j \geq 0 \quad \text{for } j \leq m, \tag{2.16a}$$

$$\forall_k : u_k * (g(x_k) - b) \geq 0, \tag{2.16b}$$

$$\{u_k * (g(x_k) - b)\} \to 0 \Rightarrow \{\sigma(x_k, Y_{i(k)})\} \to 0. \tag{2.16c}$$

The definition of $\{u_k\}_{k=0}^{\infty}$ by (2.11) was merely a convenient way to explicitly state a scheme to aggregate violated as well as nonviolated constraints.

Remark 2.5. Theorem 2.2 is valid for any block-splitting (not necessarily a partition) $P_1, \ldots, P_q$ with the property

$$\bigcup_{i=1}^{q} P_i = P := \{1, \ldots, p\}.$$

It is therefore permissible that $P_i = P_j, i \neq j$, which means that we can apply a finite number of consecutive iterations of the PAM to the same subsystem.

Remark 2.6. If we look carefully at the proof of Theorem 2.2, we observe that a dynamic block choice is allowed without impairing convergence. Let $\{P_k\}_{k=0}^{\infty}$ be a sequence of blocks; we say that $\{P_k\}_{k=0}^{\infty}$ is almost cyclic by row if

$$\begin{array}{lll} \forall k, & \emptyset \neq P_k \subseteq P = \{1, \ldots, p\}, & \text{and} \\ \forall j, & 1 \leq j \leq p, \quad \exists(\text{a finite } t)\ \forall k \text{ such that} & j \in P_{k'} \quad \text{for } k' \leq k + t. \end{array}$$

In our numerical experiments with PAMs we make use of this dynamic block choice. For completeness we state the convergence proposition and sketch its proof.

PROPOSITION 2.1. *Let $\{P_k\}_{k=0}^{\infty}$ be a sequence of blocks, which is almost cyclic by a row, and let $\{Y_k\}_{k=0}^{\infty}$ be the sequence of its respective induced subsystems. Let* (C1)–(C6) *hold. Assume that*

$$\{u_k * (g(x_k) - b)\} \rightarrow 0 \Rightarrow \{\sigma(x_k, Y_k)\} \rightarrow 0.$$

If $\{x_k, u_k, \omega_k, d_k, \lambda_k\}_{k=0}^{\infty}$ is the sequence generated by the PAM, *and $\{x_k\}_{k=0}^{\infty} \notin S$, then*

$$\{\|x_{k+1} - x_k\|\} \rightarrow 0, \quad \{\sigma(x_k, S)\} \rightarrow 0 \quad \text{and} \quad \{x_k\} \rightarrow \hat{x} \in S.$$

Proof. As in Theorem 2.1, we deduce that

$$\{u_k * (g(x_k) - b)\} \rightarrow 0, \qquad \{\|x_{k+1} - x_k\|\} \rightarrow 0.$$

Given k, let $j \in P_k$. Given any $r \in \{1, \ldots, p\}$ take $k + t \geq k' \geq k$, such that $r \in P_{k'}$. By definition,

$$0 \leq \sigma(x_k, S_r) \leq \sigma(x_k, x_{k'}) + \sigma(x_{k'}, S_r).$$

Since $\sigma(x_{k'}, Y_{k'}) \rightarrow 0$ and $r \in P_{k'}$, then $\sigma(x_{k'}, S_r) \rightarrow 0$. From the previous inequality $\sigma(x_k, S_r) \rightarrow 0$. Because r was arbitrary, we have

$$\sigma(x_k, S_r) \rightarrow 0 \quad \text{for } r = 1, \ldots, p,$$

and by continuity we obtain the convergence result.

Remark 2.7. We require neither (2.16a) nor (2.16b) to prove convergence of the PAM for solving a system of linear equations. The basic assumption will be

$$\text{(2.16c)} \qquad \{u_k * (g(x_k) - b)\} \rightarrow 0 \Rightarrow \left\{\sigma\left(x_k, Y_{i(k)}\right)\right\} \rightarrow 0.$$

Therefore, it is not convenient to treat equalities as two inequalities.

3. Parallel algorithms. In this section we describe and analyze two types of PAMs that are useful in a multiprocessor environment: structured PAMs (SPAMs), that are well suited for the solution of structured and sparse systems [19], [20], and parallel PAMs (PPAMs) that exhibit per se a high degree of parallelism and are therefore suitable for the solution of dense and nonstructured systems.

To develop SPAMs we need the following definition.

Uncoupled subsystems. Assume that the index sets $P_1, \ldots, P_q$ define a row block, that is, $P_i \cap P_j = \emptyset$, and $\cup_{i=1}^{q} P_i \subseteq P := \{1, \ldots, p\}$. The subsystems $Y_r := \cap_{i \in P_r} S_i$ and $Y_t := \cap_{i \in P_t} S_i$ are uncoupled if

$$\text{(3.1a)} \qquad i \in P_r, \qquad j \in P_t,$$

$$\text{(3.1b)} \qquad a_i * M^{\prime} a_j \begin{cases} \leq 0 & \text{if } i \leq m, \quad j \leq m, \\ = 0 & \text{otherwise,} \end{cases}$$

where a_i is the ith row of $A(x)$.

Let $P_1, \ldots, P_q$ be index sets of uncoupled subsystems $Y_1, \ldots, Y_q$. Given $x_k \in R^n$, let $u_{ki}, i = 1, \ldots, q$, be q (bounded) vectors in R^p that satisfy (3.2) below. To simplify the notation we substitute u_i for u_{ki} as follows:

$$\text{(3.2a)} \qquad u_i^j \geq 0 \quad \text{if } j \leq m \quad \text{and} \quad j \in P_i,$$

$$\text{(3.2b)} \qquad u_i^j = 0 \quad \text{if } j \notin P_i,$$

$$\text{(3.2c)} \qquad u_i * (g(x_k) - b) \geq 0,$$

$$\text{(3.2d)} \qquad \{u_i * (g(x_k) - b)\} \rightarrow 0 \Rightarrow \{\sigma(x_k, Y_i)\} \rightarrow 0.$$

LEMMA 3.1. *Given $x \in R^n, z \in X$, and $Q = \{1, \dots, q\}$, let $P_1, \dots, P_q$ be uncoupled subsystems, let* (3.2) *hold, and let $x_1 = x + \sum_{i \in Q} \omega_i \lambda_i d_i$ with*

$$0 < \eta \leq \omega_i \leq 2 - \eta, \quad d_i = -M^{-1}A^T u_i \quad and \quad \lambda_i = \frac{u_i * (g(x) - b)}{d_i * M d_i} \quad for\ i \in Q.$$

Under these conditions

$$\|x_1 - z\|_M^2 \leq \|x - z\|_M^2 - \sum_{i \in Q} \omega_i (2 - \omega_i) \lambda_i u_i * (g(x) - b), \tag{3.3a}$$

$$\|x_1 - z\|_M^2 \leq \|x - z\|_M^2 - \sum_{i \in Q} \omega_i (2 - \omega_i) \lambda_i^2 d_i * M d_i. \tag{3.3b}$$

Proof. By definition, (3.3a) and (3.3b) are equivalent, so we only prove (3.3a).

$$\begin{aligned} \|x_1 - z\|_M^2 &= \left\| x - \sum_{i \in Q} \omega_i \lambda_i M^{-1} A^T u_i - z \right\|_M^2 \\ &= \|x - z\|_M^2 - \sum_{i \in Q} (2\omega_i \lambda_i A^T u_i * (x - z) - \omega_i^2 \lambda_i^2 A^T u_i * M^{-1} A^T u_i) \\ &\quad + \sum_{i,j \in Q, i \neq j} 2\omega_i \lambda_i \omega_j \lambda_j A^T u_i * M^{-1} A^T u_j. \end{aligned}$$

The last term is nonpositive by (3.1) and (3.2); therefore,

$$\|x_1 - z\|_M^2 \leq \|x - z\|_M^2 - \sum_{i \in Q} (2\omega_i \lambda_i A^T u_i * (x - z) - \omega_i^2 \lambda_i^2 d_i * M d_i).$$

Now we follow Lemma 2.1 to deduce that

$$\|x_1 - z\|_M^2 \leq \|x - z\|_M^2 - \sum_{i \in Q} \omega_i (2 - \omega_i) \lambda_i u_i * (g(x) - b).$$

Lemma 3.1 and (3.2d) allow us to state the following proposition.

PROPOSITION 3.1. *Let $\{Q_k\}_{k=0}^{\infty}$ be a block sequence which is almost cyclic by row, where for any k, Q_k is either the union of index sets of uncoupled subsystems or the index set of a coupled subsystem. Let $\{Y_k\}_{k=0}^{\infty}$ be the sequence of its respective induced subsystems. Let* (C1)–(C6) *hold. Assume that* (3.2) *holds for any set of uncoupled subsystems, and* (2.9a) *holds for any block Q_k of coupled subsystems.*

If $\{x_k, u_k, \omega_k, d_k, \lambda_k\}_{k=0}^{\infty}$ is the sequence generated by PAMs, *and $\{x_k\}_{k=0}^{\infty} \notin S$, then*

$$\{\|x_{k+1} - x_k\|\} \to 0, \quad \{\sigma(x_k, S)\} \to 0 \quad and \quad \{x_k\} \to \hat{x} \in S.$$

Proof. We follow Theorem 2.2 (or Proposition 2.1) to prove that (2.9a) holds when Q_k is the union set of uncoupled subsystems, and the result follows, using the same theorem once more.

Because of this proposition, we can have individual processors working simultaneously on uncoupled subsystems. The results generated by each processor are then added up (see Fig. 2). It is pertinent to point out that after performing q sequential projection steps of PAMs, that is, q iterations, we obtain by (2.8a) that

$$\|x_q - z\|_M^2 \leq \|x - z\|_M^2 - \sum_{i \in Q} \omega_i (2 - \omega_i) \lambda_i u_i * (g(x_i) - b), \quad Q - \{0, \dots, q-1\}, \tag{3.4}$$

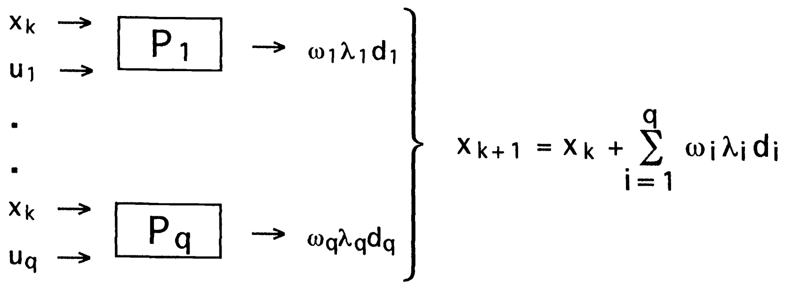


FIG. 2. *For uncoupled subsystems, q processors work simultaneously.*

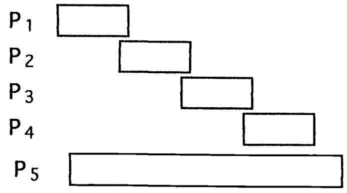


FIG. 3. *Block angular structure.*

which resembles (3.3); yet, they have a sensible practical difference. The subgradients $\partial g(.)$ must be computed q times to generate x_q by (3.4), whereas inequality (3.3) holds after performing q simultaneous projections and one addition of weighted directions. This means that the subgradients are computed only once while generating x_{k+1}. The rest of the work required to generate x_q in a sequential process is almost the same as the work required to generate x_{k+1} in a parallel process.

Summarizing, we can always use parallel processing on uncoupled portions of the system and sequential processing on the rest of the system. For example, we may partition the p rows of the block angular system shown in Fig. 3 in blocks $P_1, \ldots, P_5$, and define the following block sequence: $Q_1 = \{P_1 \cup P_2 \cup P_3 \cup P_4\}, Q_2 = P_5$. We generate x_{k+1} working simultaneously on blocks P_1, P_2, P_3, and P_4; then we generate x_{k+2} working on block P_5. Another common system possesses the staircase structure given in Fig. 4. We define $Q_1 = \{P_1 \cup P_3\}$ and $Q_2 = \{P_2 \cup P_4\}$ and generate x_{k+1} working simultaneously on P_1 and P_3, and x_{k+2} working simultaneously on P_2 and P_4.

The PPAMs to be described next are quite useful for solving large, dense, unstructured, convex systems in a multiprocessor architecture. If we have q processors $p_1, \ldots, p_q$ that can work simultaneously, we may split the linearized system $Ax = b$ into subsystems $A_1x = b_1, \ldots, A_qx = b_q$, not necessarily disjoint. Given the estimate x, the processor p_i generates ω_i, λ_i, u_i, and d_i under the general scheme for PAMs for solving $A_ix = b_i$; thus each processor p_i generates a weighted direction $\omega_i\lambda_i d_i$ aimed at finding a closer point to the subsystem $A_ix = b_i$. We are essentially using a "divide and conquer" approach, where the task of finding a feasible point of a large and nonstructured system S is simplified into the lenient tasks of solving the feasibility problem of q small subsystems. The weighted directions generated $\omega_i\lambda_i d_i, i = 1, \ldots, q$, are proposed to a coordinating processor which defines the

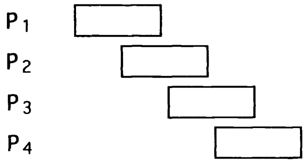


FIG. 4. *Staircase structure.*

direction $\underline{d} := \sum_{i=1}^{q} \omega_i \lambda_i d_i$ and generates $\underline{\omega}$, $\underline{\lambda}$, and $\underline{u}$ to get the minimum along $\underline{d}$ of the function $f(.,z)$ defined by (2.7); that is, we get the closest point to the set X starting at $\underline{x}$ and along the direction $\underline{d}$. It is rather trivial to generate $\underline{\omega}$, $\underline{\lambda}$, and $\underline{u}$; indeed,

$$\underline{d} := \sum_{i=1}^{q} \omega_i \lambda_i d_i = -\sum_{i=1}^{q} \omega_i \lambda_i M^{-1} A^T u_i = -M^{-1} A^T \underline{u},$$

where $\underline{u} = \sum_{i=1}^{q} \omega_i \lambda_i u_i$. If $\underline{\lambda}$ is given by (2.3b) and $0 < \eta \leq \underline{\omega} \leq 2 - \eta$, all the conditions of Lemma 2.1 hold; therefore,

$$\text{(3.5)} \quad \|\underline{x} + \underline{\omega\lambda d} - z\|_M^2 \leq \|\underline{x} - z\|_M^2 - \underline{\omega}(2 - \underline{\omega})\underline{\lambda u} * (g(\underline{x}) - b) \quad \text{for all } z \in X,$$

and convergence can be proved similarly. For the sake of completeness, let us state the general framework for the PPAMs and sketch their proof of convergence.

```
PPAM ALGORITHM.
$P_1, \ldots, P_q$ := Index sets describing the row block splitting with the property $\cup_{i=1}^{q} P_i = \{1, \ldots, p\}$
$Y_i := \cap_{j \in P_i} S_j, i = 1, \ldots, q$
$\sigma(.,.)$ := a predefined point to set distance
$0 < \eta \leq 1, 0 < \varepsilon$
Let $k = 0$
Choose $\underline{x}_k \in R^n$ as an estimate of a feasible point of $S$
While $\sigma(\underline{x}_k, S) > \varepsilon$ do
Step 1
    For $i = 1, \ldots, q$ do in parallel
        Define $u_{ki} \in R^p$ satisfying (3.2)
        Compute $\omega_{ki}, \lambda_{ki}$ and $d_{ki}$ according to Lemma 3.1.
    End For
Step 2
    Let $\underline{u}_k = \sum_{i=1}^{q} \omega_{ki} \lambda_{ki} u_{ki}$
    Define $0 < \eta \leq \underline{\omega}_k \leq 2 - \eta$
    Let $d_k = -M^{-1} A^T \underline{u}$
    Let $\underline{\lambda}_k = (\underline{u}_k * (g(\underline{x}_k) - b))/(\underline{d}_k * M\underline{d}_k)$
    Let $\underline{x}_k = \underline{x}_k + \underline{\omega}_k \underline{\lambda}_k \underline{d}_k$
    Let $k = k + 1$
End While
End of PPAM Algorithm
```

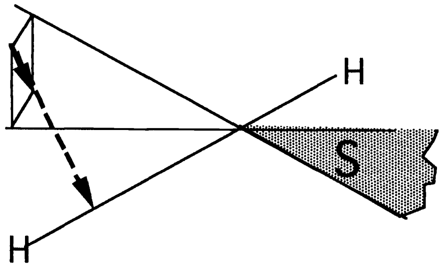


FIG. 5. *Cimmino vs "closest."* → *Cimmino is "shorter" than* ⇢ *closest.*

We let $\{\underline{x}_k, \underline{u}_k, \underline{\omega}_k, \underline{d}_k, \lambda_k\}_{k=0}^{\infty}$ be the sequence generated by the PPAM above, and formulate the following proposition.

PROPOSITION 3.2. *If* (C1) *and* (C2) *hold, then* $\{\underline{x}_k\} \to \hat{x} \in X$.

Proof. Equation (3.5) holds by assumption; therefore,

$$\{\underline{\omega}_k(2-\underline{\omega}_k)\lambda_k\underline{u}_k * (g(\underline{x}_k)-b)\} \to 0,$$

which implies

$$\{\underline{u}_k * (g(\underline{x}_k)-b)\} \to 0. \tag{3.6}$$

We prove convergence if we show that (3.6) implies $\sigma(\underline{x}_k, S) \to 0$. This follows because

$$\underline{u}_k * (g(\underline{x}_k)-b) = \sum_{i=1}^{q} \omega_{ki}\lambda_{ki}u_{ki} * (g(\underline{x}_k)-b),$$

and (3.6) implies in turn, using (C2), (3.2), and Lemma 3.1,

$$\begin{array}{ll} \{\lambda_{ki}u_{ki} * (g(\underline{x}_k)-b)\} \to 0 & \text{for } i=1,\dots,q, \\ \{u_{ki} * (g(\underline{x}_k)-b)\} \to 0 & \text{for } i=1,\dots,q, \\ \{\sigma(\underline{x}_k, Y_i)\} \to 0 & \text{for } i=1,\dots,q, \\ \{\sigma(\underline{x}_k, S)\} \to 0, & \end{array}$$

where the last implication follows by continuity and $S = Y_1 \cap \cdots \cap Y_q$. A remarkable feature of the PPAM is that the new direction $\underline{d}$ is not a convex combination of the projected directions, as it is required to be by Cimmino [10] and recent researchers (Elfving [14], Censor and Elfving [7], De Pierro and Iusem [12], Flåm [16], Bramley and Sameh [3] and Yang and Murty [33]). Figure 5 gives insight into the improvement that can occur in badly conditioned systems. Cimmino generates a "short" distance, and so has slow convergence, because the iterates remain caught in the region of small angle. PPAMs, on the other hand, project on the hyperplane $H(\underline{u}, \underline{x})$ to locate the closest point to X along the same direction, escaping from that region. Numerical experiments carried out by Bramley and Sameh [3] show that a variant of Cimmino's method is not the best algorithm for solving a system of equations derived from discretization of elliptic partial differential equations.

Getting "the minimum along the line" can be used in sequential algorithms for solving linear systems. After we generate the weighted directions $\omega_{k+1}, \lambda_{k+1}, d_{k+1}, i = 1, \dots, q$, we

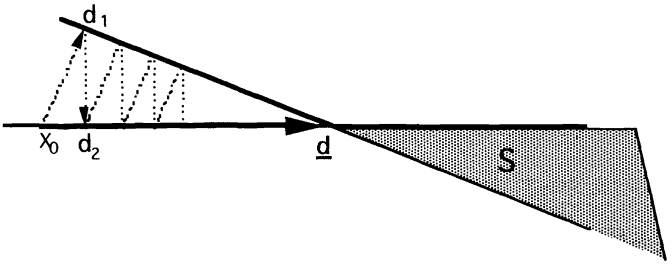


FIG. 6. *d is the direction of search for a minimum.*

make a correction step and find the minimum value of $f(., z)$ along the combined direction $\underline{d}$ as before. Figure 6 illustrates the behavior of a sequential method, which suggests the search of a minimum along $\underline{d}$ after we project on all blocks.

Remark 3.1. If each processor proposes a direction $-M^{-1}A^Tu_i, i = 1, \ldots, q$, with u_i satisfying (3.2), the convergence proof still holds if the head processor generates u, d, λ, ω, and x_{k+1} as follows:

$$u = \sum_{i=1}^{q} u_i,$$
$$d = -M^{-1}A^T \sum_{i=1}^{q} u_i = -M^{-1}A^T u,$$
$$\lambda = \frac{u * (g(x_k) - b)}{d * Md}, \qquad \eta \le \omega \le 2 - \eta,$$
$$x_{k+1} = x_k + \omega\lambda d.$$

Remark 3.2. We recall that Cimmino defines a direction as a linear combination of the directions proposed by the q processors, namely,

$$d = \sum_{i=1}^{q} \mu_i \lambda_i d_i \quad \text{with } \mu_i \ge 0 \quad \text{and} \quad \sum_{i=1}^{q} \mu_i = 1,$$
$$x_{k+1} = x_k + \omega d, \qquad \eta \le \omega \le 2 - \eta.$$

4. Numerical experiments. In this section we report some preliminary numerical results. The main purpose of our experiments is to compare the behavior of parallel versions of the PAM with some sequential versions and to give some hints on ways for coding PPAMs on a multiprocessor environment. We carried out all the numerical tests on a Macintosh SE/30 and coded all programs in the compiled Microsoft Quick Basic version 1.0.

We solved the system $Ax \le b$ with 500 inequalities and 50 unknowns, where A was a matrix of constant integer values randomly generated between -10 and $+10$ and $b = Ae + (1.\text{E} - 07)e$ to ensure feasibility of $e = (1, \ldots, 1)$. The starting point was the same random integer vector with all of its components between -5 and 5, M was the identity matrix, and $\omega = 1$. The problem generated was not particularly easy to handle; in fact, a subset of 100 constraints and 20 variables could not be solved by either the Hildreth method as reported by

TABLE 2
Sequential versus parallel procedures.

Sequential	Parallel
x is given	x is given
$P := \{1,\dots,p\} = P_1 \cup P_2 \cup \dots \cup P_q$	$P := \{1,\dots,p\} = P_1 \cup P_2 \cup \dots \cup P_q$
While $\sigma(x,S) > \varepsilon$ **do**	**While** $\sigma(x,S) > \varepsilon$ **do**
Step 1	**Step 1**
Let $w = x$	**For** $i = 1,\dots,q$ **do IN PARALLEL**
Let $d = 0$	Choose u_i
Let $u = 0$	**Let** $(\omega_i, \lambda_i, d_i)$ be generated by one or more
For $1 = 1,\dots,q$ **do**	iterations of the PAM on the set defined by P_i
Choose u_i	**End For**
Let $(\omega_i, \lambda_i, d_i)$ be generated by one or more	**Let** $d = 0$
iterations of PAM on the set defined by P_i	**Let** $u = 0$
Let $d = d + \omega_i\lambda_i d_i$	**For** $i = 1,\dots,q$ **do**
Let $u = u + \omega_i\lambda_i u_i$	**Let** $d = d + \omega_i\lambda_i d_i$
Let $x = x + d$	**Let** $u = u + \omega_i\lambda_i d_i$
End For	**End For**
Step 2	**Step 2**
If $A(x)$ is constant then	Compute (ω, λ) by one iteration of the PAM
Compute (ω, λ) by one iteration of the PAM	**Let** $x = x + \omega\lambda d$
Let $x = w + \omega\lambda d$	**End While**
End if	
End While	

Iusem and De Pierro [25], or the projection method of Agmon [1] and Motzkin and Schoenberg [28]. Neither method gave a solution in fewer than 10,000 projections. Nonetheless, convergence was always obtained when more than five constraints were aggregated at every iteration.

Among the myriad algorithms that belong to the PAM, we report in Table 3 the results obtained with the choices of $\{u_k\}$ shown in Table 1 and a dynamic splitting of the index set $P := \{1,\dots,p\}$. To start the kth iteration we are given the index i_k, which represents the first index that may belong to P_k, and an integer r, which represents the maximum number of violated constraints that might be present in P_k. We assembled P_k using the following procedure.

```
PROCEDURE ASSEMBLING $P_k$
Let $P_k = \emptyset$: Card $(P_k) = 0$ (Cardinality of $P_k$).
    total = 0
    $j = i_k$
While Card$(P_k) < r$ and total $< p$ do
    If $(g(x_k) - b)^j_+ > \varepsilon$ then
        $P_k = P_k \cup \{j\}$
        Card$(P_k)$ = Card$(P_k)$ +1
    end if
    total = total +1
    $j = j + 1$
    if $j > p$ then $j = 1$
end while
$i_{k+1} = j$
end of procedure
```

The first column of Table 3 shows the value of r, the second column gives the number of directions to be combined in step 2 of the procedures shown in Table 2, the third column shows

TABLE 3
Sequential versus parallel procedures.

			Number of projecting directions ($Ax \leq b, 500 \times 50$)		
			Norm chosen as $\sigma(x, S)$		
Violated constraints	Combined directions	Version	Infinity	1-norm	2-norm
1	1	X	1898	*	*
	5	sequential	2271	*	*
	5	parallel	2343	*	*
	20	sequential	1981	*	*
	20	parallel	2384	*	*
	100	sequential	1868	*	*
	100	parallel	4013	*	*
5	1	X	725	553	415
	5	sequential	895	675	478
	5	parallel	1043	794	597
	20	sequential	720	577	428
	20	parallel	1237	1043	821
	100	sequential	49[1]	480	363
	100	parallel	100[1]	315[2]	187[2]
20	1	X	484	183	124
	5	sequential	608	216	133
	5	parallel	807	329	244
	20	sequential	529	183	116
	20	parallel	554	37[2]	62[2]
100	1	X	458	56	44
	5	sequential	458	90	69
	5	parallel	764	61[2]	130
500	1	X	445	43	37

*Same values of the ∞-norm.

[1] Best value for the ∞-norm.

[2] Better value for the parallel version compared with the sequential version.

TABLE 4
PPAM algorithm. (Number of projecting directions with 2-norm.)

	Minor	Projecting directions generated	
Rows/Block	iterations	All processors	Head processor
1	1	6673	37
5	1	3016	38
	5	6384	35
20	1	877	35
	5	3057	30
	20	5063	30
100	1	205	35
	5	474	20
	20	1359	17
500	1	74	37
	5	45	8
	20	44	4

the version used, and the last three columns give the number of directions generated when $u_k * (Ax_k - b) = \sigma(x_k, S)$. The 2-norm gave the best results, except for some anomalous cases. The minimum along the line (step 2 of the sequential procedure) did not reduce the number of projecting directions; it was, for this problem and/or the methods chosen, a useless step. The parallel version took more projections, but we should expect good speedup in a multiprocessor environment.

Table 4 gives the results of the same problem using PPAMs, but only with the 2-norm, because this norm returned the best results in the sequential version. The system was partitioned into subsystems with the same number of rows r given in the rows/block column. The minor iterations column shows t, the number of loops of the general scheme of PAMs that the processor p_i must perform. We found that the number of projecting directions always increased with t, which suggests that it is not worthwhile to strive for feasibility at all iterations; however, we should point out that the numerical experiments carried out by García-Palomares [18] reveal that exact projections might decrease the total time spent by the algorithm, as long as the number of rows per block is not too big. If we assume that the number of processors equals p/r, the time for parallel processing seems to improve with the number of processors.

We carried out experiments using (2.12) for $\{u_k\}_{k=0}^{\infty}$, but we noticed no significant changes in the number of projecting directions.

To claim conclusive statements about the practical parallel capabilities of the PAM, numerical experiments must be carried out on different parallel architectures. It is reasonable to expect that we must adapt particular methods to particular architectures.

Acknowledgments. The author is indebted to a referee who suggested the proof of Lemma 2.4, called his attention to the papers by Aharoni, Berman, and Censor [2], Censor and Lent [8], Censor [5], De Pierro and Iusem [12], and Spingarn [32], and gave opinions that improved the paper notably. He thanks another referee who forwarded to him the recent papers by Bramley and Sameh [3], [4], and expresses his gratitude to Dr. Olvi Mangasarian who sent the papers by Aharoni, Berman, and Censor [2] and Censor [5], and to Luis R. Perichi who gave him the papers by De Pierro and Iusem [12] and Spingarn [32]. The author also thanks Dr. Bernardo Feijoo, who read a previous version of this paper and suggested meaningful stylistic changes.

REFERENCES

[1] S. Agmon, *The relaxation method for linear inequalities,* Canad. J. Math., 6(1954), pp. 382–392.

[2] R. Aharoni, A. Berman, and Y. Censor, *An interior points algorithm for the convex feasibility problem,* Adv. in Appl. Math., 4(1983), pp. 479–489.

[3] R. Bramley and A. Sameh, *Domain decomposition for parallel row projection algorithms,* Appl. Numer. Math., 8(1991), pp. 303–315.

[4] R. Bramley and A. Sameh, *Row projection methods for large nonsymmetric linear systems,* SIAM J. Sci. Statist. Comput., 13(1992), pp. 168–193.

[5] Y. Censor, *Iterative methods for the convex feasibility problem,* Ann. Discrete Math., 20(1984), pp. 83–91.

[6] ———, *Parallel application of block-iterative methods in medical imaging and radiation therapy,* Math. Programming, 42(1988), pp. 307–325.

[7] Y. Censor and T. Elfving, *New methods for linear inequalities,* Linear Algebra Appl., 42(1982), pp. 199–211.

[8] Y. Censor and A. Lent, *An iterative row action method for interval convex programming,* J. Optim. Theory Appl. 34(1981), pp. 321–353.

[9] ———, *Cyclic subgradient projections,* Math. Programming, 24(1982), pp. 233–235.

[10] G. Cimmino, *Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari,* Ric. Sci., 16(1938), pp. 326–333.

[11] E. J. Craig, *The n-step iteration procedure,* J. Math. Phys., 34(1955), pp. 65–73.

[12] A. De Pierro and A. Iusem, *A simultaneous projections method for linear inequalities,* Linear Algebra Appl., 64(1985), pp. 243–253.

[13] P. P. B. Eggermont, G. Herman, and A. Lent, *Iterative algorithms for large partitioned linear systems, with applications to image reconstruction,* Linear Algebra Appl., 40(1981), pp. 37–67.

[14] T. Elfving, *Group iterative methods for consistent and inconsistent linear equations,* Numer. Math., 35(1980), pp. 1–12.

[15] I. I. Eremin, *Methods of Fejer approximations in convex programming,* Math. Notes, 3(1968), pp. 217–234.

[16] S. Flåm, *A continuous path to convex feasibility,* in Mathematical Research. Parametric Programming and Related Topics II, Akademie-Verlag 62, Berlin, 1991, pp. 50–70.

[17] I. García, *Alternativas numéricas para la solución de sistemas lineales grandes de igualdades y desigualdades,* Master's thesis, Decanato de postgrado, Universidad Simón Bolívar, Caracas, Venezuela, 1989.

[18] U. M. García-Palomares, *Q-relaxation method for solving a system of linear inequalities,* Tech. Memo ANL/MCS-TM-19, Argonne National Laboratory, Argonne, IL, 1983.

[19] U. M. García-Palomares, *A class of methods for solving large convex systems,* Oper. Res. Lett., 9(1990), pp. 183–187.

[20] ———, *Projected aggregation methods for solving a linear system of equalities and inequalities,* in Mathematical Research, Parametric Programming and Related Topics II, Akademie-Verlag 62, Berlin, 1991, pp. 61–75.

[21] L. G. Gubin, B. T. Polyak, and E. V. Raik, *The method of projections for finding the common point of convex sets,* U.S.S.R. Comput. Math. and Math. Phys., 7(1967), pp. 1–24.

[22] G. Herman and H. Levkowitz, *Initial performance of block-iterative reconstruction algorithms,* in Mathematics and Computer Science of Medical Imaging, M. Viergever and A. Todd-Porkopek, eds., Springer-Verlag, New York, 1987, pp. 305–318.

[23] A. S. Householder, *The Theory of Matrices in Numerical Analysis,* Dover Publications, New York, 1964.

[24] A. S. Householder and F. L. Bauer, *On certain iterative methods for solving linear systems,* Numer. Math., 2(1960), pp. 55–59.

[25] A. N. Iusem and A. R. De Pierro, *A simultaneous iterative method for computing projections on polyhedra,* SIAM J Control Optim., 25(1987), pp. 231–243.

[26] S. Kaczmarz, *Angenäherte Auflösung von Systemen linearer Gleichungen,* Bull. Intern Acad. Polonaise Sci. Lett., 35(1937), pp. 355–357.

[27] C. Kamath and A. Sameh, *A projection method for solving nonsymmetric linear systems on multiprocessors,* Parallel Comput., 9(1989), pp. 291–312.

[28] T. Motzkin and I. Y. Schoenberg, *The relaxation method for linear inequalities,* Canad. J. Math., 6(1954), pp. 393–404.

[29] W. Oettli, *An iterative method, having linear rate of convergence, for solving a pair of dual programs,* Math. Programming, 3(1972), pp. 302–311.

[30] S. O. Oko, *Surrogate methods for linear inequalities,* J. Optim. Theory Appl., 72(1992), pp. 247–268.

[31] B. T. Polyak, *Minimization of unsmooth functionals,* U.S.S.R. Comput. Math. and Math. Phys., 9(1969), pp. 14–25.

[32] J. Spingarn, *A primal dual projection method for solving systems of linear inequalities,* Linear Algebra Appl., 65(1985), pp. 45–62.

[33] K. Yang and K. G. Murty, *New iterative methods for linear inequalities,* J Optim. Theory Appl., 72(1992), pp. 163–185.

SIAM J. OPTIMIZATION
Vol. 3, No. 4, pp. 901–918, November 1993

014

A CENTRAL CUTTING PLANE ALGORITHM FOR CONVEX SEMI-INFINITE PROGRAMMING PROBLEMS*

K. O. KORTANEK† AND HOON NO‡

Abstract. The central cutting plane algorithm for linear semi-infinite programming (SIP) is extended to nonlinear convex SIP of the form min $\{f(x)|x \in H, g(x,t) \leq 0 \text{ all } t \in S\}$. Under differentiability assumptions that are weaker than those employed in superlinearly convergent algorithms, a linear convergence rate is established that has additional important features. These features are the ability to (i) generate a cut from any violated constraint; (ii) invoke efficient constraint-dropping rules for management of linear programming (LP) subproblem size; (iii) provide an efficient grid management scheme to generate cuts and ultimately to test feasibility to a high degree of accuracy, as well as to provide an automatic grid refinement for use in obtaining admissible starting solutions for the nonlinear system of first-order conditions; and, (iv) provide primal and dual (Lagrangian) SIP feasible solutions in a finite number of iterations.

Numerical tests are provided on a collection of problems that have appeared in the literature including some moderately sized problems from complex approximation theory.

Key words. semi-infinite programming, convex programming, central cutting plane, constraint-dropping rules, computational experiments

AMS subject classification. 49D20, 49D39, 49D45, 52A40, 65D15, 65K99, 90C25

1. Introduction. Nonlinear semi-infinite programs. In this paper the following nonlinear semi-infinite program is considered.

PROGRAM D. Find

$$V_D = \min f(x)$$

subject to the constraints

$$g(x,t) \leq 0, \qquad \text{for all } t \in S,$$

$$x \in H.$$

Here the index-set S is fixed, and the following assumptions about the data in Program D are made.

ASSUMPTION 1.1. (i) H is a compact, convex, and nonempty subset of $\mathbf{R}^n$;

(ii) S is a compact and nonempty subset of $\mathbf{R}^m$;

(iii) $f : \mathbf{R}^n \to \mathbf{R}$ is convex and continuously differentiable on H;

(iv) $g : \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}$ is continuous on $H \times S$, $g(\cdot, t)$ is convex for all t and continuously differentiable on H, and $\nabla_x g(x,t)$ is continuous on $H \times S$;

(v) there is an $\hat{x} \in H$ for which $g(\hat{x}, t) < 0$ for all $t \in S$, and which is not optimal for Program D, referred to as a *Slater* point.

Applications of problems of this type are abundant; see [7], [12], [17]–[19], and [22].

In this paper we extend Gribik's [11] linear SIP central cutting plane algorithm to Program D. Gribik's algorithm is an extension of the Elzinga–Moore [5] algorithm for finite convex programming and all of these, including ours, are interior point cutting plane methods, because certain spheres are inscribed within the region determined by all of the cuts generated up to the current iteration. As with the linear case, the algorithm does not depend on a starting

* Received by the editors June 11, 1990; accepted for publication (in revised form) September 23, 1992.

† Department of Management Sciences, University of Iowa, Iowa City, Iowa 52242.

‡ Korea Institute for Defense Analysis, P.O. Box 250, Cheong-Ryang, Seoul 130-650, Korea.

solution and, under appropriate assumptions, converges. Hence it is a global algorithm. We are able to retain the key features of the central cutting plane approach in our extension. These features are the ability to (i) generate a cut from any violated constraint; (ii) invoke efficient constraint-dropping rules for management of convex subproblem size; (iii) provide an efficient grid management scheme to generate cuts and ultimately test feasibility to a high degree of accuracy; (iv) provide primal D and dual Lagrangian feasible solutions in a finite number of iterations, and with arbitrary tolerance (hence convergence to zero) of primal-dual duality gap; and (v) provide a linear convergence rate between primal feasible points.

For the cases of linear and convex semi-infinite programming, Gustafson [14] and Gustafson and Kortanek [16] developed a system of nonlinear equations in the primal and dual variables that are necessary and sufficient for primal and dual *feasible* solutions to be optimal. Typically, Newton-type methods were used to solve these equations. The earliest implementation (of which we are aware) for solving the nonlinear system arising from linear semi-infinite programming with a convex objective function is Fahlander's 1973 Technical Report [6]. Given an adequate starting solution and continuing to solve the nonlinear system provides the location of the local maxima

$$\max g(\bar{x}, t), \qquad t \in S,$$

which greatly reduces the need to use uniform or other highly meshed grids. Recent applications of this approach also appear in Glashoff and Roleff [9] and Tang [31]. An extensive survey of efficient methods for solving general SIP problems that intrinsically use local reduction procedures appears in the recent survey by Hettich and Kortanek [18].

In this paper we present the algorithm in §2 and give a convergence analysis in §3. The study of Lagrangian duality and the convergence rate is done in §4, but under the restriction that the convex set H is to be a polytope. In §5 we test the algorithm on some examples appearing in the literature, which include some complex approximation problems. In addition, we test the combined procedure involving the nonlinear equations problem solver of Brown and Saad [3], taking as starting points the primal-dual feasible points that the algorithm delivers. We then compare the accuracy of the solutions we obtained with those of other methods, including some alternative cutting plane algorithms that have appeared recently. Finally, our conclusions appear in §6.

2. Specification of the algorithm. The assumption of a linear objective function involves no loss of generality. For if the objective function $f(x)$ is convex, then obviously $f(x) - z$ is convex, and Program D is equivalent to the following.

Find

$$\min z$$

subject to the constraints

$$\begin{aligned} &f(x) - z \leq 0, \\ &g(x,t) \leq 0 \quad \text{for all } t \in S, \\ &x \in H, \qquad \min f(x) \leq z \leq \max f(x). \end{aligned}$$

We substitute the scalar variable z with the extended component x_{n+1} of the vector x. More formally, let f^0 denote the original f with $\underset{\sim}{x} = (x_1, \ldots, x_n)$. Redefine f on $R^{n+1}, x = (\underset{\sim}{x}, x_{n+1})$ by $f(x) = f^0(\underset{\sim}{x}) - x_{n+1}$, and simply set $g(x,t) = g(\underset{\sim}{x}, t)$.

One can also specify lower and upper bounds for each component of x, since H is compact. The equivalent problem we consider for algorithmic development is then the following program.

PROGRAM D′. Find

$$V_{D'} = \min x_{n+1}$$

subject to the constraints

$$\begin{aligned} &f(x) \leq 0, \\ &g(x,t) \leq 0 \quad \text{for all } t \in S, \end{aligned}$$

where

$$x \in H \times [\min f(x), \max f(x)] = H' (\text{a subset of } \mathbf{R}^{n+1}).$$

Let $\bar{f}$ be a *strict* upper bound of $V_{D'}$ and let $\|x\|$ be the Euclidean norm; that is,

$$\|x\| = (x^T x)^{1/2} \quad \text{for } x \in \mathbf{R}^{n+1}.$$

Then the algorithm is as follows (where we define Program SD recursively).

Step 0. Choose a constant β in (0,1). Let $\bar{f}$ be strictly greater than $V_{D'}$. Let SD_0 be the program

$$\max \sigma$$

subject to

$$x_{n+1} + \sigma \leq \bar{f}, \qquad x \in H'.$$

Choose $y^{(0)} \in H'$. Let $k = 1$.

Step 1. Let $\left(x^{(k)}, \sigma^{(k)}\right) \in \mathbf{R}^{n+1} \times \mathbf{R}$ be a solution to SD_{k-1}. If $\sigma^{(k)} = 0$, stop. Otherwise go to Step 2.

Step 2. Delete constraints from SD_{k-1} according to either or both of the deletion rules, or do not delete constraints. Call the resulting program SD_{k-1} again.

Step 3. If $x^{(k)}$ is infeasible for the first constraint of Program D′, that is, $f(x^{(k)}) > 0$, go to (ii). Else if $x^{(k)}$ is infeasible for the infinite constraint system of Program D′, that is, $g\left(x^{(k)}, t\right) > 0$ for some $t \in S$, set that t to $t^{(k)}$ and go to (iii). Otherwise go to (i).

(i) Add the constraint

$$x_{n+1} + \sigma \leq x_{n+1}^{(k)}$$

to Program SD_{k-1} and set $y^{(k)} = x^{(k)}$.

(ii) Add the constraint

$$f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)\left(x - x^{(k)}\right) + \left\|\nabla f(x^{(k)})\right\| \sigma \leq 0$$

to Program SD_{k-1} and set $y^{(k)} = y^{(k-1)}$.

(iii) Add the constraint

$$g\left(x^{(k)}, t^{(k)}\right) + \nabla_x g\left(x^{(k)}, t^{(k)}\right)\left(x - x^{(k)}\right) + \left\|\nabla_x g\left(x^{(k)}, t^{(k)}\right)\right\| \sigma \leq 0$$

to Program SD_{k-1} and set $y^{(k)} = y^{(k-1)}$. Note $\nabla_x g\left(x^{(k)}, t^{(k)}\right) \neq 0$.

In either case, call the resulting Program SD_k. Set $k := k + 1$ and return to Step 1.

Deletion Rule 1. Delete the constraint $x_{n+1} + \sigma \le \bar{f}$ or any constraint generated by Step 3(i) in a previous iteration if $x^{(k)}$ is feasible for Program D′.

Deletion Rule 2. Delete a constraint from SD_{k-1} if

(a) the constraint was generated by Step 3(ii) or (iii) at the jth iteration where $j < k$,

(b) $\sigma^{(k)} \le \beta\sigma^{(j)}$,

(c) the constraint was not tight in SD_{k-1} at $\left(x^{(k)}, \sigma^{(k)}\right)$.

In practice, both Deletion Rule 1 and Deletion Rule 2 are applied. We will show here that the algorithm coverges with either rule, both rules, or neither rule used in Step 3.

3. A convergence analysis of the algorithm. The proof of convergence closely parallels the convergence proof given in Gribik [11] for his linear semi-infinite programming algorithm, and in some cases, we have retained some of his original wording and phrases.

LEMMA 3.1 *Using either, both, or neither of the deletion rules, if the algorithm does not terminate, the sequence* $\{\sigma^{(k)}\}_k$ *converges to* 0.

Proof. Since the set of feasible points for Program D′ is compact and nonempty, there exists a point that is optimal for Program D′. Let $\bar{x} \in \mathbf{R}^{n+1}$ be optimal for Program D′; then $(\bar{x}, 0)$ is obviously feasible for SD_k for all k. Hence $\sigma^{(k)} \ge 0$ for all k. Deletion Rules 1 and 2 only drop constraints that are not binding. Therefore, $\sigma^{(k)} \ge \sigma^{(k+1)}$ for all k. Hence

$$\lim_k \sigma^{(k)} = \bar{\sigma} \ge 0, \quad \text{where } \bar{\sigma} \text{ is finite.}$$

Assume that $\bar{\sigma}$ is positive. Then there exists a $\hat{k}$ such that

$$\bar{\sigma} \le \sigma^{(\hat{k})} \le \bar{\sigma}/\beta$$

since $0 < \beta < 1$. Hence for $k \ge j \ge \hat{k}$,

$$\bar{\sigma} \le \sigma^{(k)} \le \sigma^{(j)} \le \bar{\sigma}/\beta.$$

Thus $\beta\sigma^{(j)} \le \sigma^{(k)}$ for all $k \ge j \ge \bar{k}$. Consequently, condition (b) of Deletion Rule 2 is never satisfied for $j \ge \hat{k}$. These constraints are never deleted. Three cases must be considered.

Case 1. $x^{(j)}$ is infeasible for the first constraint of Program D′.

In this case, Step 3(ii) is used to generate a cut, and so for all $k > j$, we must have

$$f\left(x^{(j)}\right) + \nabla f\left(x^{(j)}\right)\left(x^{(k)} - x^{(j)}\right) + \left\|\nabla f\left(x^{(j)}\right)\right\| \sigma^{(k)} \le 0. \tag{3.1}$$

From (3.1), we have

$$\begin{aligned} \left\|\nabla f\left(x^{(j)}\right)\right\| \sigma^{(k)} &\le -f\left(x^{(j)}\right) - \nabla f\left(x^{(j)}\right)\left(x^{(k)} - x^{(j)}\right) \\ &\le -\nabla f\left(x^{(j)}\right)\left(x^{(k)} - x^{(j)}\right) \\ &\le \left\|\nabla f\left(x^{(j)}\right)\right\| \left\|x^{(k)} - x^{(j)}\right\|, \end{aligned} \tag{3.2}$$

where the first inequality is due to $f\left(x^{(j)}\right) > 0$, and the second inequality is due to the Cauchy–Schwarz inequality.

Thus (3.2) implies (since $\nabla f\left(x^{j}\right) \ne 0$)

$$\left\|x^{(k)} - x^{(j)}\right\| \ge \sigma^{(k)} \ge \bar{\sigma} \quad \text{for all } k > j.$$

Case 2. $x^{(j)}$ is infeasible for the continuous constraints of Program D′; that is,

$$\max_{t \in S} g\left(x^{(j)}, t\right) > 0.$$

In this case, Step 3(iii) is used to generate a cut, and so for all $k > j \geq \bar{k}$,

$$\begin{aligned} g\left(x^{(j)}, t^{(j)}\right) + \nabla_x g\left(x^{(j)}, t^{(j)}\right)\left(x^{(k)} - x^{(j)}\right) \\ + \left\|\nabla_x g\left(x^{(j)}, t^{(j)}\right)\right\| \sigma^{(k)} \leq 0. \end{aligned} \tag{3.3}$$

Since $\left(x^{(j)}, t^{(j)}\right)$ is infeasible,

$$g\left(x^{(j)}, t^{(j)}\right) > 0 \quad \text{and} \quad \nabla_x g\left(x^{(j)}, t^{(j)}\right) \neq 0. \tag{3.4}$$

Subtracting (3.4) from (3.3) yields

$$\nabla_x g\left(x^{(j)}, t^{(j)}\right)\left(x^{(k)} - x^{(j)}\right) + \left\|\nabla_x g\left(x^{(j)}, t^{(j)}\right)\right\| \sigma^{(k)} \leq 0.$$

But, by the Cauchy–Schwarz inequality, this implies that

$$\begin{aligned} \left\|\nabla_x g\left(x^{(j)}, t^{(j)}\right)\right\| \sigma^{(k)} &\leq -\nabla_x g\left(x^{(j)}, t^{(j)}\right)\left(x^{(k)} - x^{(j)}\right) \\ &\leq \left\|\nabla_x g\left(x^{(j)}, t^{(j)}\right)\right\| \left\|x^{(k)} - x^{(j)}\right\|. \end{aligned}$$

Consequently, this inequality yields

$$\left\|x^{(k)} - x^{(j)}\right\| \geq \sigma^{(k)} \geq \bar{\sigma} \quad \text{for all } k > j.$$

Case 3. $x^{(j)}$ is feasible for Program D$'$.

In this case,

$$x_{n+1}^{(k)} + \sigma^{(k)} \leq x_{n+1}^{(j)} \quad \text{for all } k > j.$$

Hence this implies that

$$\left\|x_{n+1}^{(k)} - x_{n+1}^{(j)}\right\| \geq \sigma^{(k)} \geq \bar{\sigma} \quad \text{for all } k > j.$$

Therefore, in any case,

$$\left\|x^{(k)} - x^{(j)}\right\| \geq \sigma^{(k)} \geq \bar{\sigma} > 0 \quad \text{for all } k > j \geq \hat{k}.$$

This contradicts the fact that $\left\{x^{(k)}\right\}_k$ must have limit points since H' is a compact set. Thus the sequence $\left\{\sigma^{(k)}\right\}_k$ converges to 0. □

LEMMA 3.2 *If* $\tilde{x}$ *is feasible for Program* D$'$ *and* $f(\tilde{x}) < 0, g(\tilde{x}, t) < 0$ *for all* $t \in S$, *there exists a* $\tilde{\sigma} > 0$ *such that* $(\tilde{x}, \tilde{\sigma})$ *satisfies any cut generated by Step* 3(ii) *or Step* 3(iii).

Proof of Case 1. The cut generated by Step 3(ii) has $f\left(x^{(k)}\right) > 0$. If $x \in H$ and $f(x) > 0$, then by convexity of f, $\nabla f(\tilde{x})(\tilde{x} - x) \leq f(\tilde{x}) - f(x) < 0$, implying $\nabla f(x) \neq 0$. Let us check the conditions that $\tilde{\sigma}_1$ must satisfy. First,

$$\begin{aligned} & f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)\left(\tilde{x} - x^{(k)}\right) + \left\|\nabla f\left(x^{(k)}\right)\right\| \tilde{\sigma}_1 \\ & \leq f\left(x^{(k)}\right) + \left(f(\tilde{x}) - f\left(x^{(k)}\right)\right) + \left\|\nabla f\left(x^{(k)}\right)\right\| \tilde{\sigma}_1 \\ & = f(\tilde{x}) + \left\|\nabla f\left(x^{(k)}\right)\right\| \tilde{\sigma}_1, \end{aligned}$$

and so for all subproblems this constraint is satisfied if

$$\tilde{\sigma}_1 \leq -\frac{f(\tilde{x})}{\left\|\nabla f\left(x^{(k)}\right)\right\|}, \qquad k = 1, 2, \ldots .$$

Setting

$$M_1 = \max\{\|\nabla f(x)\| \, | x \in H - \{x | f(x) \leq 0\}\}$$

gives a finite positive number, and fulfills the requirement that $\tilde{\sigma}_1 \leq -f(\tilde{x})/M_1$.

Proof of Case 2. The cut generated by Step 3(iii) has $g\left(x^{(k)}, t^{(k)}\right) > 0$. From convexity of $g\left(\cdot, t^{(k)}\right)$,

$$\begin{aligned} &g\left(x^{(k)}, t^{(k)}\right) + \nabla_x g\left(x^{(k)}, t^{(k)}\right)\left(\tilde{x} - x^{(k)}\right) + \left\|\nabla_x g\left(x^{(k)}, t^{(k)}\right)\right\| \tilde{\sigma}_2 \\ &\leq g\left(\tilde{x}, t^{(k)}\right) + \left\|\nabla_x g\left(x^{(k)}, t^{(k)}\right)\right\| \tilde{\sigma}_2; \end{aligned}$$

in particular, $\nabla_x g\left(x^{(k)}, t^{(k)}\right) \neq 0$. Thus, by the same arguments as in Case 1, letting

$$M_2 = \max_{t \in S}\left\{\max \|\nabla_x g(x, t)\| \, | x \in H - \left\{x | g\left(x, t^{(k)}\right) \leq 0\right\}\right\}$$

leads to the requirement that $\tilde{\sigma}_2 \leq -g\left(\hat{x}, t^{(k)}\right)/M_2$. Simply take $\tilde{\sigma} = \min\{\tilde{\sigma}_1, \tilde{\sigma}_2\}$, which is positive. □

LEMMA 3.3 *If the algorithm terminates at iteration k^*, $y^{(k^*-1)}$ is feasible for Program* D′. *If the algorithm does not terminate, there exists $\hat{k}$ such that $y^{(k)}$ is feasible for Program* D′ *for $k \geq \hat{k}$.*

Proof. Assume that for all k, $y^{(k)}$ is infeasible for Program D′. Then for each k, $x^{(k)}$ is infeasible for Program D′. Hence Step 3(i) is never used to generate a constraint, and Deletion Rule 1 is never used to drop a constraint. Let $\bar{x}$ be optimal for Program D′ and let $\hat{x}$ be the point in Assumption 1.1(v). Then $x(\alpha) = \alpha\hat{x} + (1-\alpha)\bar{x}$ is feasible for $0 \leq \alpha \leq 1$ and

$$\begin{aligned} &f(x(\alpha)) < 0, \\ &g(x(\alpha), t) < 0 \quad \text{for all } t \in S, \qquad 0 < \alpha \leq 1. \end{aligned} \tag{3.5}$$

Since $\bar{x}_{n+1} < \bar{f}$, we can choose $\tilde{\alpha} \in (0,1)$ sufficiently small such that $x_{n+1}(\tilde{\alpha}) < \bar{f}$. Hence there exists $\sigma' > 0$ such that

$$x_{n+1}(\tilde{\alpha}) + \sigma' \leq \bar{f}.$$

By Lemma 3.2 and (3.5), there exists $\sigma'' > 0$ such that $(x(\tilde{\alpha}), \sigma'')$ satisfies any cut generated by Step 3(ii) or Step 3(iii). Choosing $\sigma^* = \min(\sigma', \sigma'') > 0$, $(x(\tilde{\alpha}), \sigma^*)$ is feasible for SD_k for all k. Thus $\lim \sigma^{(k)} \geq \sigma^* > 0$. This contradicts Lemma 3.1. □

THEOREM 3.1. *If the algorithm terminates at iteration k^*, then $y^{(k^*-1)}$ is optimal for Program* D′. *Otherwise limit points exist to the sequence $\left\{y^{(k)}\right\}_k$ and they are optimal for Program* D′.

Proof. Suppose that the algorithm does not terminate. By Lemmas 3.1 and 3.3 there exists a $\hat{k}$ such that $y^{(k)}$ is feasible for Program D′ for all $k \geq \hat{k}$. Since the feasible region for Program D′ is compact, limit points exist to $\left\{y^{(k)}\right\}_k$ and they are feasible. Let $\bar{y}$ be a limit point and $\left\{y^{(k)}\right\}_k$ now denote a subsequence converging to $\bar{y}$. Suppose that $\bar{y}$ is not optimal.

Let $\bar{x}$ be optimal for Program D$'$ and $\hat{x}$ be the point specified in Assumption 1.1(v). Then, defining $x(\alpha) = \alpha\bar{x} + (1-\alpha)\hat{x}$,

$$\begin{aligned} &f(x(\alpha)) < 0 \quad \text{for } 0 \le \alpha < 1, \\ &g(x(\alpha), t) < 0 \quad \text{for all } t \in S \quad \text{and} \quad 0 \le \alpha < 1, \end{aligned}$$

and

$$x_{n+1}(\alpha) < \bar{y}_{n+1} \quad \text{for} \max\left(\frac{\bar{y}_{n+1} - \hat{x}_{n+1}}{\bar{x}_{n+1} - \hat{x}_{n+1}}, 0\right) \le \alpha < 1.$$

Choose

$$\tilde{\alpha} \in \left(\max\left(\frac{\bar{y}_{n+1} - \hat{x}_{n+1}}{\bar{x}_{n+1} - \hat{x}_{n+1}}, 0\right), 1\right)$$

and set $\sigma' = \bar{y}_{n+1} - x_{n+1}(\tilde{\alpha}) > 0$. Then $(x_{n+1}(\tilde{\alpha}), \sigma')$ satisfies

$$x_{n+1} + \sigma \le \bar{y}_{n+1} < y_{n+1}^{(k)} \quad \text{for } k \ge \hat{k}.$$

By Lemma 3.2, there exists a $\sigma'' > 0$ such that $(x(\tilde{\alpha}), \sigma'')$ satisfies any cut generated by Step 3(ii) or Step 3(iii). Thus $(x(\tilde{\alpha}), \sigma^*)$, where $\sigma^* = \min(\sigma', \sigma'') > 0$ is feasible for SD_k for all k. Hence $\lim_k \sigma^{(k)} \ge \sigma^* > 0$, which contradicts Lemma 3.1.

If the algorithm terminates at iteration k^*, $y^{(k^*-1)}$ is feasible for Program D$'$ by Lemma 3.3. If $y^{(k^*-1)}$ is not optimal, an argument similar to the above shows that $\sigma^{(k^*)} > 0$. Hence the algorithm could not have been terminated. □

4. Dual program and convergence rate.

For the derivation of a convergence rate of the algorithm, and hence for the development of a dual program, we will assume that H' is a polyhedral set defined by a set of linear inequalities; that is,

$$H' = \{x | a_j x \le b_j, j \in J\}, \quad \text{where } J = \{1, 2, \ldots, m\}.$$

We assume that the matrix A whose jth row is denoted a_j has full rank; for example, "box" constraints could be typical.

In this section we shall also denote a limit point of a nonterminating sequence generated by the algorithm by $\bar{x} \in \mathbf{R}^{n+1}$.

Used in the construction of a dual program to D$'$, let $\mathbf{R}^{(S)}$ denote the linear space of all real-valued functions on S having only finite support, termed the generalized finite sequence space of S over $\mathbf{R}$. Thus, if $\xi \in \mathbf{R}^{(S)}$, then the number of elements in the set $\{t | \xi(t) \ne 0\}$ is finite, while $\xi \ge 0$ shall mean pointwise. We consider a Lagrangian form of the dual of D; see Gol'stein [10] and Rockafellar [26].

PROGRAM P.

$$V_P = \sup_{\xi \in \mathbf{R}^{(S)}, \Psi \in \mathbf{R}^n} \left\{ \inf_{x \in \mathbf{R}^n} L(\xi, \Psi, x) \right\},$$

where

$$\begin{aligned} &L(\xi, \Psi, x) = f^0(x) + \sum_t \xi(t) g(x, t) + \sum_j \Psi_j (a_j x - b_j) \\ (4.1) \qquad &\text{subject to} \quad \inf_x L(\xi, \Psi, x) > -\infty, \\ &\text{and } \xi \text{ and } \Psi \ge 0. \end{aligned}$$

For convenience in the analysis to follow, let us repeat program SD_{k-1} as the following linear program and use its duality properties next.

$$(\mathrm{SD}_{k-1}) \qquad \max \sigma$$

subject to the constraints

$$\begin{aligned}
&x_{n+1} + \sigma \le y_{n+1}^{(k-1)},\\
&f\left(x^{(i)}\right) + \nabla f\left(x^{(i)}\right)\left(x - x^{(i)}\right) + \left\|\nabla f\left(x^{(i)}\right)\right\| \sigma \le 0 \quad \text{for } i \in F_k,\\
&g\left(x^{(l)}, t^{(l)}\right) + \nabla_x g\left(x^{(l)}, t^{(l)}\right)\left(x - x^{(l)}\right) + \left\|\nabla_x g(x^{(l)}, t^{(l)}\right\| \sigma \le 0 \quad \text{for } l \in G_k,\\
&a_j x \le b_j \quad \text{for } j \in J,
\end{aligned}$$

where F_k, G_k are the sets of indices of subprogram constraints which were generated from $f(x)$ and $g(x,t)$, respectively.

LEMMA 4.1. *Let* $\left(x^{(k)}, \sigma^{(k)}\right)$ *be optimal for* SD_{k-1} *and let* $\mu_0^{(k)}, \{v_i^{(k)} \mid \text{all } i \in F_k\}$, $\{\mu_l^{(k)} \mid \text{all } l \in G_k\}$, *and* $\{\lambda_j^{(k)} \mid \text{all } j \in J\}$ *be optimal dual variables to* SD_{k-1}. *Let* $\underset{\sim}{x} \in \boldsymbol{R}^n$ *be feasible for* D *and assume* $\mu_0^{(k)} > 0$. *Then*

$$f(\underset{\sim}{x}) \ge y_{n+1}^{(k-1)} - \left(\sigma^{(k)} \big/ \mu_0^{(k)}\right).$$

In this case, the right-hand side is a lower bound for V_D.

Proof. It follows from LP duality that

$$\mu_0^{(k)} = \sum_i v_i^{(k)}, \tag{4.2}$$

$$\mu_0^{(k)} + \sum_{i \in F_k} \nu_i^{(k)} \left\|\nabla f\left(x^{(i)}\right)\right\| + \sum_{l \in G_k} \mu_l^{(k)} \left\|\nabla_x g\left(x^{(l)}, t^{(l)}\right)\right\| = 1, \tag{4.3}$$

and

$$\begin{aligned}
\sigma^{(k)} \ge \sigma &- \mu_0^{(k)}(x_{n+1} + \sigma - y_{n+1}^{(k-1)})\\
&- \textstyle\sum_{i \in F_k} \nu_i^{(k)} \left(f(x^{(i)})\right.\\
&\qquad \left. + \nabla f\left(x^{(i)}\right)(x - x^{(i)}) + \left\|\nabla f(x^{(i)})\right\| \sigma\right)\\
&- \textstyle\sum_{l \in G_k} \mu_l^{(k)}(g(x^{(l)}, t^{(l)}) + \nabla_x g\left(x^{(l)}, t^{(l)}\right)(x - x^{(l)})\\
&\qquad + \left\|\nabla_x g\left(x^{(l)}, t^{(l)}\right)\right\| \sigma) - \textstyle\sum_{j \in J} \lambda_j^{(k)}(a_j x - b_j)\\
&\text{for all } x \in \mathbf{R}^{n+1} \text{and all } \sigma \in \mathbf{R}.
\end{aligned} \tag{4.4}$$

The next step will be to convert (4.4) into a Lagrangian type inequality by using convexity, i.e.,

$$\begin{aligned}
f(x) &\ge f\left(x^{(i)}\right) + \nabla f\left(x^{(i)}\right)\left(x - x^{(i)}\right),\\
g\left(x, t^{(l)}\right) &\ge g\left(x^{(l)}, t^{(l)}\right) + \nabla_x g\left(x^{(l)}, t^{(l)}\right)\left(x - x^{(l)}\right),
\end{aligned}$$

and simplifying via (4.2) and (4.3); in particular, σ vanishes from the right side of (4.4):

$$\begin{aligned}
\sigma^{(k)} \ge \mu_0^{(k)} &\left(y_{n-1}^{(k-1)} - f^0(\underset{\sim}{x})\right) - \textstyle\sum_l \mu_l^{(k)} g(\underset{\sim}{x}, t^{(l)})\\
&- \textstyle\sum_j \lambda_j^{(k)}(a_j \underset{\sim}{x} - b_j) \quad \text{for all } \underset{\sim}{x} \in \mathbf{R}^n.
\end{aligned} \tag{4.5}$$

Now for all feasible $\underset{\sim}{x} \in \mathbf{R}^n, g\left(\underset{\sim}{x}, t^{(l)}\right) \le 0$ for all l, and $a_j \underset{\sim}{x} - b_j \le 0$ for all j. Hence $f(\underset{\sim}{x}) \ge y_{n+1}^{(k-1)} - \sigma^{(k)}/\mu_0^{(k)}$. □

LEMMA 4.2. *Any sequences of optimal* LP *dual vectors,* $\{\mu_0^{(k)}\}_k, \{\nu_i^{(k)} | \textit{for all } i \in F_k\}$, $\{\mu_l^{(k)} | \textit{for all } l \in G_k\}_k$, *and* $\{\lambda_j^{(k)} | \textit{for all } j \in J\}_k$, *are bounded. Moreover,* $\{\sum_{l \in G_k} \mu_l^{(k)}\}_k$ *is also bounded.*

Proof. From (4.3) we have $\mu_0^{(k)} \in [0, 1]$, which together with (4.2), shows that $\{\lambda_i^{(k)} |$ all $i \in F_k\}_k$ is bounded. Referring back to a Slater point $\hat{x} \in \mathbf{R}^n$ (Assumption 1.1), set $x' = \left(\hat{x}, f^0(\hat{x})\right)$, and let $\bar{x}$ be D′-optimal. Then, from (4.5) for all k,

$$\sigma^{(k)} \ge -\mu_0^{(k)} \left(f^0(\hat{x}) - y_{n+1}^{(k-1)}\right) - \sum_l \mu_l^{(k)} g\left(\hat{x}, t^{(l)}\right), \tag{4.6}$$

using $f(x') = 0$ and $-\sum_j \lambda_j^{(k)} \left(a_j \bar{x} - b_j\right) \ge 0$. But for all $l \in G_k, g\left(\hat{x}, t^{(l)}\right) \le \max_t g(\hat{x}, t) < 0$. Since $\sigma^{(k)}$ converges to 0 and $\mu_l^{(k)} \in [0, 1]$ for all l, it follows from (4.6) that $\{\sum_{l \in G_k} \mu_l^{(k)}\}_k$ is bounded. In particular, $\{\mu_l^{(k)}\}_k$ is uniformly bounded.

Finally, the boundedness of $\{\lambda_j^{(k)} | j \in J\}_k$ follows from the full rank assumption of the matrix A. □

LEMMA 4.3. *Assume that the algorithm does not terminate, and let* $\underline{\mu}_0 = \lim_k \inf \mu_0^{(k)}$ *and* $\bar{\mu}_0 = \lim_k \sup \mu_0^{(k)}$. *Then* $0 < \underline{\mu}_0 \le \bar{\mu}^0 < +\infty$. *If the algorithm terminates at stage k, then* $\mu_0^{(k)} > 0$.

Proof. Let: $\delta = \max\{\max_{x \in H'} \|\nabla f(x)\|, \max_{t \in S}(\max_{x \in H'} \|\nabla_x g(x, t)\|)\} > 0$. Then from (4.2) and (4.3), for each $k = 1, 2, \ldots,$

$$\sum_l \mu_l^{(k)} \ge \left(1 - \mu_0^{(k)}(1 + \delta)\right) / \delta. \tag{4.7}$$

For a Slater point $\hat{x}$, let τ denote $\max_t g(\hat{x}, t)$, which is negative. We return to (4.6) with the knowledge that $V_D \le y_{n+1}^{(k-1)}$ and using (4.7),

$$\sigma^{(k)} \ge -\mu_0^{(k)}(f^0(\hat{x}) - V_D) - \tau(1 - \mu_0^{(k)}(1 + \delta))/\delta,$$

to immediately obtain

$$\mu_0^{(k)} \ge -\left(\sigma^{(k)} + \frac{\tau}{\delta}\right) \Big/ \left(f^0(\hat{x}) - V_D - \frac{\tau}{\delta}(1 + \delta)\right). \tag{4.8}$$

If the algorithm does not terminate, there is no restriction against writing $\lim_k \mu_0^{(k)} = \underline{\mu}_0$ and $\lim_k \sum_l \mu_l^{(k)} = M, M = \lim_k \sup \sum_l \mu_l^{(k)}$; see (4.7). Since $\lim_k \sigma^k = 0$, (4.8) yields

$$\underline{\mu}_0 \ge \left(-\frac{\tau}{\delta}\right) \Big/ \left(f^0(\hat{x}) - V_D - \frac{\tau}{\delta}(1 + \delta)\right) > 0. \tag{4.9}$$

For finite termination, $\sigma_k = 0$, and so (4.8) permits $\underline{\mu}_0$ on the left side of (4.9) to be replaced with $\mu_0^{(k)}$. □

THEOREM 4.1. *Let* $\mu_0^{(k)}, \{\nu_i^{(k)} | \textit{ all } i \in F_k\}$, *and* $\{\mu_l^{(k)} | \textit{ all } l \in G_k\}$ *be optimal* LP *duals for* SD_{k-1}. *For* $\mu_0^{(k)} \neq 0$ *define*

$$\begin{array}{lll} \xi_l^{(k)}(t) & = \mu_l^{(k)}/\mu_0^{(k)} & \textit{if } t \in \left\{t^{(l)} | l \in G_k\right\}, \\ & = 0 & \textit{otherwise}, \end{array}$$

and

$$\Psi_j^{(k)} = \lambda_j^{(k)}/\mu_0^{(k)} \quad \textit{for } j \in J.$$

Then $\xi(t) = \xi^{(k)}(t)$ *and* $\Psi_j = \Psi_j^{(k)}$ *is feasible for the Lagrangian dual* P. *Moreover,* $\lim_k L\left(\xi^{(k)}, \Psi^{(k)}, \bar{x}\right) = f^0(\bar{x})$, *where now* $\bar{x} \in \mathbf{R}^n$ *denotes an optimal solution to Program* D.

Proof. The result will follow from (4.5) and Lemma 4.3. Since $\mu_0^{(k)} > 0$, we have

$$(4.10) \qquad \inf_{x \in \mathbf{R}^n} \left[f^0(x) + \sum_l \xi_l^{(k)} g(x, t^{(l)}) + \sum_j \Psi_j^{(k)}(a_j x - b_j) \right] \geq y_{n+1}^{(k-1)} - \sigma^{(k)}/\mu_0^{(k)}.$$

Thus the consistency condition (4.1) is satisfied.

Now, given any $\epsilon > 0$, there exists K such that $k \geq K$ implies

$$(4.11) \qquad y_{n+1}^{(k-1)} - \sigma^{(k)}/\mu_0^{(k)} \geq f^0(\bar{x}) - \epsilon,$$

using Lemma 4.3, $\lim_k y_{n+1}^{(k-1)} = f^0(\bar{x})$, and $\lim_k \sigma^{(k)} = 0$. By the standard duality inequality, $f^0(\bar{x})$ is at least as great as the left side of (4.10). Combined with (4.11) we obtain

$$f(\bar{x}) \geq \inf_{x \in \mathbf{R}^n} L\left(\xi^{(k)}, \Psi^{(k)}, x\right) \geq f^0(\bar{x}) - \epsilon \quad \text{for } k \geq K. \qquad \square$$

Remark. Consistency of the Lagrangian dual can also be expressed in differential Wolfe-dual form; see [26].

Let us consider the convergence rate of the algorithm, determined with the help of duality.

THEOREM 4.2. *Between feasible points, the algorithm has a linear rate of convergence.*

Proof. Combining Lemmas 3.3 and 4.3 yields that there exists K such that for $k \geq K$, $\mu_0^{(k)}$ is positive and y_{k-1} is feasible for Program D. Now with respect to an optimal solution $x^{(k)}$ for SD_{k-1}, complementary slackness of its first constraint yields

$$(4.12) \qquad \sigma^{(k)} = y_{n+1}^{(k-1)} - x_{n+1}^{(k)}, \qquad x_{n+1}^{(k)} = f^0(\underline{x}).$$

Setting $\underset{\sim}{x} = \bar{x}$, a D-optimum, in (4.5) yields

$$y_{n+1}^{(k-1)} - x_{n+1}^{(k)} \geq \mu_0^{(k)}\left(y_{n+1}^{(k-1)} - \bar{x}_{n+1}\right) - \sum_l \mu_l^{(k)} g(\bar{x}, t^{(l)}) - \sum_j \lambda_j (a_j \bar{x} - b_j) \geq -\mu_0^{(k)}\left(\bar{x}_{n+1} - y_{n+1}^{(k-1)}\right).$$

Hence there exists $\rho \in (0,1)$ such that for k sufficiently large,

$$(4.13) \qquad y_{n+1}^{(k-1)} - x_{n+1}^{(k)} \geq \rho \underline{\mu}_0 \left(y_{n+1}^{(k-1)} - V_D\right), \qquad \bar{x}_{n+1} = V_D.$$

Now when $x^{(k)}$ becomes feasible for Program D, $y^{(k)} = x^{(k)}$ and from (4.12),

$$y_{n+1}^{(k)} < y_{n+1}^{(k-1)}.$$

Thus, for k sufficiently large, (4.13) becomes (with subtraction and addition of $\bar{x}_{n+1}$)

$$\left(y_{n+1}^{(k-1)} - \bar{x}_{n+1}\right) - \left(y_{n+1}^{(k)} - \bar{x}_{n+1}\right) \geq \rho\bar{\mu}_0 \left(y_{n+1}^{(k-1)} - \bar{x}_{n+1}\right).$$

Hence

$$\frac{y_{n+1}^{(k)} - \bar{x}_{n+1}}{y_{n+1}^{(k-1)} - \bar{x}_{n+1}} \leq 1 - \rho\bar{\mu}_0 \quad \text{for } k \geq \hat{k} \quad \text{and} \quad \rho \in (0,1).$$

Since the objective function value of Program D′ is $\bar{x}_{n+1}$ and $y_{n+1}^{(k)}$ is the objective function value of Program D′ at iteration k, the algorithm has a linear rate of convergence in the objective function value. □

5. Implementation and numerical examples.

5.1. Computational realization of the algorithm. Note that $g\left(x^{(k)}, t\right)$ does not need to satisfy stronger assumptions such as concavity or differentiability. We usually use a discretization method for index set S. That seems reasonable with reference to modeling of semi-infinite programming problems in practice.

The index set S is replaced by a finite subset S_ϵ of S with the density

$$\max_{t\in S}\min_{t'\in S_\epsilon} \|t - t'\| \leq \epsilon.$$

Generally, a uniform fixed grid is used for any cutting plane algorithm to get the most violated cut within one iteration procedure. In the central cutting plane algorithm, however, we can apply a grid refinement procedure within one iteration because we only need to find one of the violated cuts. We initially determine ϵ_0 by 16 uniform intervals and increase these to 46 intervals for ϵ_1 and so on, if we cannot find an infeasible point within one iteration. The smallest ϵ value, say ϵ_l, is predetermined to check the feasibility of the constraints, and the accuracy of the final solution is dependent on the ϵ value, ϵ_l. It is obvious that the feasibility checking procedure is more efficient if we choose $\epsilon_1, \ldots, \epsilon_l$ corresponding to mutually disjoint discretizations. Even though we can get an accurate solution for very small ϵ_l, this is inefficient because it takes much computational time, and because singularity of the constraint matrix of a linear programming subproblem may occur. To get a more accurate solution, we adjoin the following phase to the central cutting algorithm itself.

For Program D the rather well-known nonlinear system which is necessary and sufficient for *feasible* primal and dual solutions to be optimal is of the following form (see [14]–[17] and [19]).

$$\nabla f(x) + \sum_{i=1}^{q} \xi(t_i)\nabla_x g(x,t_i) = 0, \tag{5.1}$$

$$\xi(t_i)g(x,t_i) = 0 \quad \text{for } i = 1,\ldots,q \tag{5.2}$$

and

$$g(x,\cdot) \text{ has a local maximum at each } t_i \quad \text{if } \xi(t_i) > 0, \tag{5.3}$$

where $q \leq n$, and where at most n variables need to be nonzero.

Note that (5.3) corresponds to solving the Karush–Kuhn–Tucker system for max $\{g(x,t)|t \in S\}$; see [19, pp. 96–97]. Much progress has been made recently in solving this system in such a way that t_i can be formed independently and efficiently. Until now, however, additional differentiability assumptions and constraint qualifications have been required, e.g., see the Hettich and Kortanek survey [18]. This system of nonlinear equations has $n + 2q$ variables and $n + 2q$ equations if all of t_i lies in the interior of S.

To avoid including additional conditions arising from the case that a t_i could be a boundary point of S, we have performed experiments with a simple approximation. If a computed t_i value (from the cutting plane algorithm) lies within a fixed tolerance of a boundary point, we reassign to t_i the particular boundary value that is closest to it. Typically, the tolerance chosen was 1.D $-$ 6 to 1.D $-$ 9. Thus the system we solve is

$$\nabla_x g(x,t_i) = 0 \quad \text{only for } t_i \text{ in the interior of } S, \tag{5.4}$$

and the fixing, if necessary, of some t_i at boundary points (a or b in one dimension, or $(a_1,b_1),(a_2,b_2)$ in two dimensions).

In Step 3(i) of the algorithm, if we get the new feasible point $x^{(k)}$ and its dual $\xi(t_1)^{(k)},\dots,\xi(t_q)^{(k)}$, then we solve the system of nonlinear equations (5.1)–(5.3) by an iterative method using $x^{(k)},t_1^{(k)},\dots,t_q^{(k)},\xi(t_1)^{(k)},\dots,\xi(t_q)^{(k)}$ as the starting point. If the iterative method for the system of nonlinear equations (5.1)–(5.3) converges, then we check to see if the results are feasible for the program, in which case they are optimal.

Combining a discretization with the system of nonlinear equations to the algorithm permits the following realization of the algorithm that we have implemented.

Step 0$'$. Choose $\{S_{\epsilon_0},\dots,S_{\epsilon_l}\}$ from S and apply Step 0 of the algorithm.
Step 1$'$. Same as Step 1 of the algorithm.
Step 2$'$. Same as Step 2 of the algorithm.
Step 3$'$. (a) Set $i = 0$.
(b) Check feasibility of $x^{(k)}$ for $t \in S_{\epsilon_i}$.
(c) If $x^{(k)}$ is infeasible, then go to Step 3(ii) or (iii).
(d) If $i < l$, then set $i = i + 1$ and go to (b).
(e) Solve the nonlinear system of equations (5.1)–(5.3) with the current primal and dual $\left(x^{(k)} \text{ and } \lambda^{(k)}\right)$ as a starting point. If the resulting solution is feasible, then stop. Otherwise go to Step 3(i) of the algorithm.

5.2. Implementation details. In Step 0$'$, it is not necessary to specify an initial feasible point. As indicated in [5] and [11], the discretized central cutting plane algorithm will find a feasible point after a finite number of iterations. Thus an initial feasible point was not required in the implementation.

Instead of using $\{S_{\epsilon_0},\dots,S_{\epsilon_l}\}$ in Step 0$'$, we used uniform grid increment methods. In this case all we need are three parameters such as the maximum number of points in a grid, initial number of points, and increments. We typically chose 301 points, starting with 16 points, and incrementing with 30 points.

To solve the linear programming subproblems in Step 1 of the algorithm, LINOP by Hettich et al. was used. Georg and Hettich [8] have shown that the most stable simplex method implementation occurs when the orthogonal Q-matrix is retained and updated through successive iterations. The LINOP program is an application of this concept.

Like most convex cutting plane algorithms, the solution of linear programming subproblems can be accomplished by adding or dropping columns in the dual linear programming subproblems. Since only one column needs to be added per iteration and only inactive cuts are dropped, few pivots are required to solve these dual linear programming subproblems. We, however, did not make this simplification in our current implementation.

To keep the size of the linear programming subproblems within manageable dimensions, all deletion rules were applied in Step 2 of the algorithm.

To solve the nonlinear system equations in Step 3$'$, the recent development of a nonlinear Krylov solver NKSOL by Brown and Saad [3] was used, as just one possible choice made, in part, for convenience. This program uses an inexact Newton method as the basic nonlinear iteration, where the Newton equations are solved only approximately by a linear Krylov iteration coupled with either a linesearch or dogleg global strategy. We applied the software to (5.4) using the approximation for the case of t_i near the boundary of S.

We programmed a FORTRAN code implementation and used double precision. Some BLAS subroutines are included to handle the vector operations. We ran experiments on the VAX 6000-410 under the VMS 5.3 operating system.

5.3. Numerical examples. The following problems were used in numerical experiments of the proposed algorithm. If a starting point was given, then it was used; otherwise, a simple LP solution from Step 0 was used. In all examples, the β used in Deletion Rule 2 was 0.75.

Example 1 (Tichatschke and Nebeling [32]).

$$\begin{aligned}
&f(x) = (x_1 - 2)^2 + (x_2 - 0.2)^2;\\
&g(x,t) = ((5\sin\pi\sqrt{t})/(1+t^2))x_1^2 - x_2;\\
&H = \{x \in \mathbf{R}^2 | -1 \le x_1 \le 1, 0 \le x_2 \le 0.2\};\\
&S = \{t \in \mathbf{R} | 0 \le t \le 8\}.
\end{aligned}$$

Input parameters.
Maximum number of grid points: 301;
Tolerance for the boundary of S: 10^{-2};
Stopping tolerance for NL system equations: 10^{-10}.
Results. Primal solution:

$$x_1 = 0.205236774, \qquad x_2 = 0.2.$$

Dual solution:

$$\begin{aligned}
\xi(t) &= 1.84175707 \quad &&\text{if } t = 0.213412466,\\
&= 0 &&\text{otherwise.}
\end{aligned}$$

Objective function value: 3.22117504;
Norm of NL system equations: 0.2×10^{-12};
Number of iterations: 3;
Number of NL system equations applied: 1;
Elapsed CPU time (seconds): 0.37.

Using 1024 uniform grid points, the ordinary cutting plane solution given in [32] is $(x_1, x_2) = (0.205143, 0.199808)$ after 12 iterations.

Example 2 (Polak and He [23]; Tanaka, Tukushima, and Ibaraki [30]).

$$\begin{aligned}
&f(x) = x_1^2 + x_2^2 + x_3^2,\\
&g(x,t) = x_1 + x_2\exp(x_3 t) + \exp(2t) - 2\sin(4t),\\
&H = \{x \in \mathbf{R}^3 | -2 \le x_1 \le 2, -2 \le x_2 \le 2, -2 \le x_3 \le 2\},\\
&S = \{t \in \mathbf{R} | 0 \le t \le 1\}.
\end{aligned}$$

Input parameters.
Maximum number of grid points: 301,
Tolerance for the boundary of S: 10^{-2},
Stopping tolerance for NL system equations: 10^{-10}.
Results. Primal solution.

$$x_1 = -0.213312578, \quad x_2 = -1.36145045, \quad x_3 = 1.85354733.$$

Dual solution.

$$\begin{aligned} \xi(t) &= 0.426625155 \quad \text{if } t = 1.0, \\ &= 0 \quad \text{otherwise.} \end{aligned}$$

Objective function value: 5.33468728;
Norm of NL system equations: 0.4×10^{-12};
Number of iterations: 27;
Number of NL system equations applied: 9;
Elapsed CPU time (seconds): 2.71.

For Example 2 the set H was artificially constructed. On the other hand, most of the examples used the initial starting point that was specified in the literature.

Example 3 (Tichatschke and Nebeling [32]).

$$\begin{aligned} &f(x) = x_1^2 + x_2^2 \\ &g(x,t) = ((x_1-2)^2 + (x_2-2)^2 - 4)t_1 + (x_1^2 + x_2^2 - 4)t_2, \\ &H = \{x \in \mathbf{R}^2 | 0 \le x_1 \le 2, 0 \le x_2 \le 2\}, \\ &S = \{t \in \mathbf{R}^2 | 0 \le t_1 \le 1, 0 \le t_2 \le 1\}. \end{aligned}$$

Input parameters.
Maximum number of grid points: 16×16;
Tolerance for the boundary of S: 10^{-2},
Stopping tolerance for NL system equations: 10^{-10}.
Results. Primal solution

$$x_1 = 0.585786438, \qquad x_2 = 0.585786438.$$

Dual solution.

$$\begin{aligned} \xi(t) &= 6.62741700 \quad \text{if } t = (0.0625, 0), \\ &= 0 \quad \text{otherwise.} \end{aligned}$$

Objective function value: 0.686291501;
Norm of NL system equations: 0.5×10^{-13};
Number of iterations: 9;
Number of NL system equations applied: 1;
Elapsed CPU time (seconds): 0.30.
The exact solution of this problem is $(x_1, x_2) = (2 - \sqrt{2},\ 2 - \sqrt{2})$.

5.4. Complex approximation. A linear Chebyshev complex approximation problem that we study is the following convex program (CP). Given complex valued functions $u_j, j = 1, \ldots, n$, and f defined on the complex plane **C**, let $r(x,z) = f(z) - \sum_{j=1}^{n} x_j u_j(z)$, where $x, z \in \mathbf{C}$. Let B be a compact set in the complex plane.

TABLE 1

Primal solutions

	Kortanek and No	From [27, Table 6]
$x_1 =$	0.368117039	0.3682810
$x_2 =$	0.888713155	0.8891080
$x_3 =$	−1.98904455	−1.989632
$x_4 =$	−1.98904456	−1.989632
$x_5 =$	2.63132741	2.6317190
$x_6 =$	1.08993150	1.0900940
$\overline{w} =$	1.470768E-02	1.47063E-02

Find

$$\bar{w} = \min_{x \in \mathbf{C}} \max\{|r(x,z)|\,|z \in B\}.$$

As recognized by Barrodale, Delves, and Mason [2] and others, CP is equivalent to a program like Program D in §1.

Convex dual (CD):

$$\begin{aligned}&\min w\\ &\text{s.t.} - w + (\mathrm{Re}(r(x,z)))^2 + (\mathrm{Im}(r(x,z)))^2 \leq 0\\ &\text{for all } z \in B,\end{aligned}$$

where Re and Im denote real and imaginary parts of a complex number.

In the numerical results to follow we omit the dual solution and other outputs, but we compare our primal solutions (to nine digits) to those of others.

Example 4 (Streit [27], Streit and Nuttall [28]).

$$\begin{aligned}&f(z) = \exp(i3t), \quad u_j(z) = \exp(i(j-1)t), \quad j = 1,2,3; \quad i = \sqrt{-1},\\ &B = [0, \pi/4], \quad H = \{x \in R^6 |\, -4 \leq x_j \leq 4, j = 1,2,\ldots,6\}.\\ &x_j \text{ complex variable identified with the real pair } x_{2(j-1)+1}, x_{2(j-1)+2}\\ &\text{for } j = 1,2,3.\end{aligned}$$

Input parameters.

Maximum number of grid points: 301;

Tolerance for the boundary of S: 10^{-10};

Stopping tolerance for NL system equations: 10^{-12}.

Results. See Table 1.

Norm of NL system equations: 1.5×10^{-11};

Number of iterations: 139;

Elapsed CPU time (seconds): 90.63.

Example 5 (Barrodale, Delves, and Mason [2]; Glashoff and Roleff [9]).

$$\begin{aligned}&f(z) = 1/(z-2), \qquad u_j(z) = z^{j-1},\\ &j = 1,2,\ldots,n \quad \text{with } z = \cos t + i \sin t,\\ &t \in B = [0, 2\pi].\end{aligned}$$

TABLE 2

Primal solutions		
	Kortanek and No	From [9]
$x_1 =$	−0.500000000	−0.499999
$x_2 =$	−0.250000000	−0.25001
$x_3 =$	−0.125000000	−0.125001
$x_4 =$	−0.625000000E-01	−0.062499
$x_5 =$	−0.416666666E-01	−0.041667
$\overline{w} =$	2.08333333E-02	0.020833

TABLE 3

Primal solutions		
	Kortanek and No	From [9]
$x_1 =$	−0.500000000	−0.500003
$x_2 =$	−0.250000000	−0.249999
$x_3 =$	−0.125000000	−0.124996
$x_4 =$	−0.625000000E-01	−0.062505
$x_5 =$	−0.312499999E-01	−0.031249
$x_6 =$	−0.156250000E-01	−0.015622
$x_7 =$	−0.104166666E-01	−0.010419
$\overline{w} =$	5.2083333333E-03	0.005208

In these examples all x_j's are real, and $H = \{x \in R^n | -3.1 \leq x_j \leq 3.1, j = 1, 2, \ldots, n\}$. Our computed $\overline{w}$'s are about the same as in [2].

For $n = 5$ in Example 5.

Input parameters.
Maximum number of grid points: 301;
Tolerance for the boundary of S: 1.0×10^{-4};
Stopping tolerance for NL System Equations: 1.0×10^{-12}
Results. See Table 2.
Norm of NL system equations: 5.0×10^{-13};
Number of iterations: 123;
Elapsed CPU time (seconds): 201.30.
For $n = 7$ in Example 5.
Results. See Table 3.
Norm of NL system equations: 5.5×10^{-13};
Number of iterations: 437;
Elapsed CPU time (seconds): 603.62.

Remark. For Example 4, Streit [27] determined a closed form solution given 101 uniformly spaced gridpoints. The solution has OV (objective value) $\bar{w} = 1.47063\text{E-}02$, which Streit recovered in Table 6 [27] and which is given above. Over the continuum, this solution is slightly infeasible. For Example 4, an objective value is reported in Table 3(b) in Reemtsen [25] as

$$\bar{w} = 1.470768026(30)\text{E-}02.$$

This agrees with our solution to a high degree, where we obtained

$$\bar{w} = 1.470768029\text{E-}02.$$

6. Conclusions. The assumptions for the nonlinear SIP in §1 are clearly weaker than those typically found for guaranteeing superlinearly convergent algorithms. When addressing only convex problems, the assumptions that f itself and $g(\cdot, t)$ are convex for all t are the usual ones. A Slater point basically insures bounded Lagrange multipliers in a dual program. However, requiring second-order derivatives in x, together with additional constraint qualifications, leads to more efficient methods of local reduction, particularly in locating the local maxima in (5.3).

In this paper the main differentiability assumptions are that $g(x,t)$ is continuously differentiable on H for each t, and that $\nabla_x g(x,t)$ is continuous on $H \times S$. There is no assumption of differentiability with respect to t or, say, concavity with respect to t (for fixed x).

The cutting plane algorithm here also uses the Slater assumption because it is an interior point algorithm. A favorite description of Gribik [11] for the linear case applies here also with an appropriate linearization. A finite LP problem gives the largest sphere that can be drawn within all of the cuts added so far and the upper bound on the linearized cost function whose center lies in H. Another interior point cutting plane method (see Bahn et al. [1]) uses the concept of an analytic center, which can be defined by means of a logarithmic potential function.

Typical of cutting plane methods including [1], [11], and [13], our algorithm also has linear convergence between primal feasible points (not just between all iterates generated by the algorithm). It also has useful and easily implementable constraint-dropping schemes, just as in the linear case. In our view, a more remarkable result is the attainment of primal and Lagrangian dual feasibility in a finite number of iterations, analogous to the linear SIP case with a generalized finite sequence space dual. This property reinforces the subjective view that a cutting plane method can have an important role in obtaining a good starting solution (as a Phase I method), to which a more efficient method could then be applied when suitable differentiability and constraint qualification assumptions are present.

Acknowledgments. This paper benefitted from considered observations made by two referees. Initially, the duality section was incomplete and the numerical implementation procedures required a clearer presentation.

REFERENCES

[1] O. Bahn, J. L. Goffin, J. P. Vial, and O. DuMerle, *Implementation and Behavior of an Interior Point Cutting Plane Algorithm for Convex Programming: An Application to Geometric Programming*, Dép. d'Economie Commerciale et Industrielle, Univ. de Génève, Switzerland, March 1991.

[2] I. Barrodale, L. M. Delves, and J. C. Mason, *Linear Chebyshev approximation of complex-valued functions*, Math Comp., 32(1978), pp. 853–863.

[3] P. N. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear system of equations*, SIAM J. Sci. Statist. Comput., 11(1990), pp. 450–481.

[4] U. Eckhardt, *Semi-infinite quadratic programming*, OR-Spektrum, 1(1979), pp. 51–55.

[5] J. Elzinga and T. G. Moore, *A central cutting plane algorithm for the convex programming problem*, Math. Programming, 8(1975), pp. 134–145.

[6] K. Fahlander, *Computer Programs for Semi-Infinite Optimization*, TRITA NA-7312, Dept. of Numerical Analysis and Computer Sciences, Royal Institute of Technology, S-10044, Stockholm 70, Sweden, 1973.

[7] A. V. Fiacco and K. O. Kortanek, eds., *Semi-Infinite Programming and Applications*, Lecture Notes in Economics and Mathematical Systems 215, Springer-Verlag, New York, 1981.

[8] K. Georg and R. Hettich, *On the Numerical Stability of the Simplex Algorithm*, Univ. Trier, Germany, 1985; also in the LINOP Software Package.

[9] K. Glashoff and K. Roleff, *A new method for Chebyshev approximation of complex-valued functions*, Math. Comp., 36(1981), pp. 233–239.

[10] E. G. Gol'stein, *Theory of Convex Programming*, Transl. Math. Monographs, American Mathematical Society, Providence, RI, 1972.

[11] P. R. Gribik, *A central cutting plane algorithm for semi-infinite programming problems*, in Semi-Infinite Programming, Lecture Notes in Control and Information Sciences 15, R. Hettich, ed., Springer-Verlag, New York, 1979.

[12] ———, *Selected applications of semi-infinite programming*, in Constructive Approaches to Mathematical Models, C. V. Coffman and G. J. Fix, eds., Academic Press, New York, 1979, pp. 171–188.

[13] P. R. Gribik and D. N. Lee, *A comparison of two central cutting plane algorithms for prototype geometric programming problems*, in Methods of Operations Research 31, W. Oettli and F. Steffens, eds., 1978, Verlag Anton/Hain/Mannheim, Germany, pp. 275–287.

[14] S.-Å. Gustafson, *On the computational solution of a class of generalized moment problems*, SIAM J. Numer. Anal., 7(1970), pp. 343–357.

[15] ———, *A three-phase algorithm for semi-infinite programs*, in Semi-Infinite Programming and Applications, Lecture Notes in Economics and Mathematical Systems 215, A. V. Fiacco and K. O. Kortanek, eds., Springer-Verlag, New York, 1981, pp. 138–157.

[16] S.-Å. Gustafson and K. O. Kortanek, *Numerical treatment of a class of semi-infinite programming problems*, Naval Research Logistics Quart., 20(1973), 477–504.

[17] ———, *Semi-infinite programming and applications*, in Mathematical Programming: the State of the Art 1982, A. Bachem, M. Grotschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983.

[18] R. Hettich and K. O. Kortanek, *Semi-infinite programming: Theory, methods, and applications*, Univ. Trier, Germany, 1992; SIAM Rev., 35(1993), pp. 380–429.

[19] R. Hettich and P. Zencke, *Numerische Methoden der Approximation und Semi-Infiniten Optimierung*, Teubner, Stuttgart, 1985.

[20] K. O. Kortanek, *Vector-supercomputer experiments with the primal affine linear programming scaling algorithm*, SIAM J. Sci. Comput., 14(1993), pp. 279–294.

[21] G. Opfer, *Solving complex approximation problems by semi-infinite optimization techniques: A study on convergence*, Numer. Math., 39(1982), 411–420.

[22] E. Polak, *On the mathematical foundations of nondifferentiable optimization in engineering design*, SIAM Rev., 29(1987), pp. 21–89.

[23] E. Polak and L. He, *A Unified Steerable Phase I–Phase II Method of Feasible Directions for Semi-Infinite Optimization*, Dept. Electrical Engineering and Computer Sciences, Univ. of California, Berkeley, 1990.

[24] R. Reemsten, *Outer Approximation Methods for Semi-infinite Optimization Problems*, Fach. Math., Technische Univ. Berlin, Germany, 1991.

[25] ———, *A cutting plane method for solving minimax problems in the complex plane*, Numer. Algorithms, 2(1992), pp. 409–436.

[26] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[27] R. L. Streit, *Solutions of systems of complex linear equations in the l_∞ norm with constraints on the unknowns*, SIAM J. Sci. Statist. Comput., 7(1986), pp. 132–149.

[28] R. L. Streit and A. H. Nuttall, *A general Chebyshev complex function approximation procedure and an application to beamforming*, J. Acoust. Soc. Amer., 72(1982), pp. 181–190.

[29] ———, *A note on the semi-infinite programming approach to complex approximation*, Math. Comp., 40(1983), pp. 599–605.

[30] Y. Tanaka, M. Tukushima, and T. Ibaraki, *A comparative study of several semi-infinite nonlinear programming algorithms*, European J. Oper. Res., 36(1988), pp. 92–100.

[31] P. T. P. Tang, *A fast algorithm for linear complex Chebysev approximations*, Math. Comp., 51(1988), pp. 721–739.

[32] R. Tichatschke and V. Nebeling, *A cutting-plane method for quadratic semi-infinite programming problems*, Optimization, 19(1988), pp. 803–817.

[33] G. A. Watson, *Numerical methods for Chebyshev approximation of complex-valued functions*, in Algorithms for Approximation II, J. C. Mason and M. G. Cox, eds., Chapman and Hall, London, New York, 1990, pp. 246–264.