JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 56, No. 1, JANUARY 1988

On Solving Three Classes of Nonlinear Programming Problems via Simple Differentiable Penalty Functions[1,2]

N. I. M. GOULD[3]

Communicated by R. A. Tapia

Abstract. We consider the following classes of nonlinear programming problems: the minimization of smooth functions subject to general constraints and simple bounds on the variables; the nonlinear l_1-problem; and the minimax problem. Numerically reliable methods for solving problems in each of these classes, based upon exploiting the structure of the problem in constructing simple differentiable penalty functions, are presented.

Key Words. Quadratic penalty functions, simple bound constraints, l_1-problems, minimax problems.

1. Introduction

In a recent paper (Ref. 1), it has been shown how the nonlinear programming problem

$$\begin{aligned} \text{(NLP)} \quad & \text{minimize } f(x), \\ & \text{subject to } c_i(x) = 0, \quad i \in \mathscr{E}, \\ & c_i(x) \geq 0, \quad i \in \mathscr{I}, \end{aligned}$$

may be solved by a careful use of the quadratic penalty function

$$p_2(x, \mu) = f(x) + 1/2\mu \sum_{i \in \mathscr{E}} c_i(x)^2 + 1/2\mu \sum_{i \in \mathscr{I}} \min(0, c_i(x))^2.$$

[1] This research was made possible by NSERC Grant No. A8442.

[2] The author would like to thank Mrs. J. Selwood of the Department of Combinatorics and Optimization, University of Waterloo, Ontario, Canada for her excellent typesetting.

[3] Senior Scientific Officer, Computer Science and Systems Division, AERE, Harwell, Oxfordshire, England. This work was carried out in the Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada.

0022-3239/88/0100-0089$06.00/0

Although theoretically sound methods for solving NLP via a sequential minimization of $p_2(x, \mu)$ have been known for many years (see Ref. 2 for a detailed history), they have been disregarded as practical methods because of the natural ill-conditioning of $p_2(x, \mu)$ as μ shrinks to zero. This ill-conditioning manifests itself in the increasing condition number of the Hessian matrix of $p_2(x, \mu)$ as μ decreases. As a consequence, a direct application of Newton's method to finding a stationary point of $p_2(x, \mu)$ is likely to produce increasingly inaccurate search directions in the presence of finite precision computer arithmetic (see Refs. 3 and 4).

Recently, a number of authors (Refs. 1, 5, and 6) have shown that, despite the deteriorating condition number of the Hessian matrix of $p_2(x, \mu)$, it is still possible to determine the solution to the exact Newton equations very accurately. This, in turn, has enabled the authors to construct efficient and robust algorithms for solving NLP. In this paper, we shall be concerned with applying the method proposed in Ref. 1 to a number of commonly occurring classes of nonlinear programming problems. We shall be especially interested in using the inherent structure in such problems to enhance the method. In particular, we shall try to construct methods which are more efficient than just simply considering the special problem as an instance of NLP and then applying the method in Ref. 1.

The special problems to be considered here are these: the nonlinear programming problem where the variables are subjected to simple bounds (which may be infinite),

$$\begin{aligned} \text{(BNLP)} \quad & \underset{x \in \mathcal{R}^n}{\text{minimize}} \; f(x), \\ & \text{subject to } c_i(x) = 0, \qquad i \in \mathcal{E}, \\ & \qquad\qquad c_i(x) \geqslant 0, \qquad i \in \mathcal{I}, \\ & \qquad\qquad l_i \leqslant x_i \leqslant u_i, \qquad 1 \leqslant i \leqslant n; \end{aligned}$$

the minimization of a function which may be expressed as a finite sum of absolute values of given functions,

$$\text{(L}_1\text{P)} \quad \underset{x \in \mathcal{R}^n}{\text{minimize}} \sum_{i=1}^{m} |f_i(x)|;$$

and the minimization of the largest of a finite collection of given functions,

$$\text{(MINIMAX)} \quad \underset{x \in \mathcal{R}^n}{\text{minimize}} \; \underset{1 \leqslant i \leqslant m}{\text{maximum}} \; f_i(x).$$

A discussion of each of these problems will be given in the appropriate section of the paper.

The paper is organized as follows. The method given in Ref. 1 is briefly reviewed in Section 1.1. BNLP is considered in Section 2. L_1P is considered

in Section 3, and numerical experiments with the resulting algorithm follow in Section 4. Section 5 contains a description of how MINIMAX can be solved, and numerical results obtained with this proposed method are given in Section 6.

Throughout much of this paper (Sections 1–6), we shall assume that any solution to NLP satisfies the following nondegeneracy assumption:

(NDA) If x^* is a local solution to NLP and if $\mathscr{A} = \{i : i \in \mathscr{J} \cup \mathscr{E}$ and $c_i(x^*) = 0\}$, then the collection of constraint gradients $\nabla_x c_i(x^*)$, $i \in \mathscr{A}$, are linearly independent and the associated Lagrange multipliers are all strictly positive. (See, e.g., Ref. 7).

In Section 7, we discuss the implications of relaxing this assumption and the corresponding modifications to our algorithms. We include this section because degenerate optimal solutions to L_1P and MINIMAX often occur in practice.

Whenever they occur, functions f, f_i, c_i map $\mathscr{R}^n$ into $\mathscr{R}$. The functions are all assumed to be twice continuously differentiable, although the methods described here can normally tolerate some second derivative discontinuities. The $t \times t$ identity matrix will be denoted I_t. If the dimension is clear, I_t will be written as I. The appropriately dimensioned vector e is the vector with each element 1. The ith element of a vector v will often be denoted by v_i. Finally, the vector x^* will indicate a local solution to the problem under consideration.

1.1. Underlying Method. The transformation of nonlinear programming problems into unconstrained minimization problems via penalty functions has proved to be one of the most successful methods for solving NLP. The simplest such penalty function, the quadratic penalty function $p_2(x, \mu)$, is continuously differentiable (and piecewise twice continuously differentiable) and is such that, under normal circumstances, if $x(\mu)$ is a local minimizer of $p_2(x, \mu)$, then $x(\mu) \to x^*$, a local solution of NLP, as $\mu \to 0_+$ (see Ref. 2). This then prompts the following SUMT-type [sequential unconstrained minimization technique (Ref. 2)] method for solving NLP: Given a starting value of μ_{start} (10^{-1}, say), an initial estimate of x^* (x_{start}), and a prescribed smallest value $\mu_{\min}$ (10^{-12}, say), the SUMT algorithm is as follows:

Set $\mu := \mu_{\text{start}}$, $x_0 := x_{\text{start}}$.

Until $\mu < \mu_{\min}$, do these operations:

Minimize $p_2(x, \mu)$ to find $x(\mu)$ starting from x_0.

Let $x_0 := x(\mu)$, $\mu := \mu \times 10^{-2}$.

The unconstrined minimization of $p_2(x, \mu)$, for fixed μ, can be effected by either a line search or trust-region method (see, e.g., Ref. 7). The method proposed in Ref. 1 is of the line search variety and finds a sequence $x^{(k)}$,

so that

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} p^{(k)},$$

where $p^{(k)}$ is a descent direction for $p_2(x, \mu)$ at $x^{(k)}$ and where $\alpha^{(k)}$ is chosen so that $p_2(x^{(k+1)}, \mu)$ is sufficiently smaller than $p_2(x^{(k)}, \mu)$ in the sense of Armijo (Ref. 8).

In order that such an algorithm attains a fast asymptotic convergence rate, it is desirable that $p^{(k)}$ be chosen as the Newton direction whenever possible. Let

$$V(x) = \{i : i \in \mathcal{I},\ c_i(x) \leq 0\} \cup E.$$

At $x = x^{(k)}$, the penalty function may be expressed as

$$f(x) + 1/2\mu \sum_{i \in \mathcal{V}(x^{(k)})} c_i(x)^2 \equiv \bar{p}(x),$$

say. Gould (Ref. 1) takes the point of view that $p^{(k)}$ should be chosen to be the Newton direction for $\bar{p}(x)$ whenever $\nabla_{xx}\bar{p}(x)$ is positive definite, a direction of negative curvature whenever $\nabla_{xx}\bar{p}(x)$ is indefinite, and either a weak solution to the Newton equations or a direction of linear infinite descent whenever $\nabla_{xx}\bar{p}(x)$ is positive semidefinite and singular. The major trouble with such a procedure is that, as $\mu \to 0_+$, $\nabla_{xx}\bar{p}(x)$ becomes increasingly ill-conditioned and a direct realization of the above choices of $p^{(k)}$ increasingly difficult. To overcome these difficulties, we observe that the Newton equations

$$\nabla_{xx}\bar{p}(x^{(k)}) p^{(k)} = -\nabla_x \bar{p}(x^{(k)}) \tag{1}$$

may be rewritten as

$$\begin{bmatrix} \bar{G}(x^{(k)}, \mu) & A(x^{(k)})^T \\ A(x^{(k)}) & -\mu I \end{bmatrix} \begin{bmatrix} p^{(k)} \\ r^{(k)} \end{bmatrix} = -\begin{bmatrix} g^{(k)} \\ c^{(k)} \end{bmatrix}, \tag{2}$$

for some auxiliary vector $r^{(k)}$, where $g^{(k)} = \nabla_x f(x^{(k)})$, $c^{(k)}$ is made up of the $c_i(x^{(k)})$, for all the indices $i \in \mathcal{V}(x^{(k)})$, $A(x^{(k)})$ is the Jacobian matrix of the $c_i(x)$ at x^k, $i \in \mathcal{V}(x^{(k)})$, and

$$\bar{G}(x^{(k)}, \mu) = \nabla_{xx} f(x^{(k)}) - \sum_{i \in \mathcal{V}(x^{(k)})} \lambda_i(x^{(k)}, \mu) \nabla_{xx} c_i(x^{(k)}),$$

with

$$\lambda_i(x, \mu) = -\mu^{-1} c_i(x), \qquad i \in \mathcal{V}(x^{(k)}).$$

Under the nondegeneracy assumption NDA,

$$\lim_{\mu \to 0_+} \lim_{x \to x(\mu)} \lambda_i(x, \mu) = \lambda_i^* > 0, \tag{3}$$

where the λ^* are the unique Lagrange multipliers for NLP. As a consequence,

$$\lim_{\mu \to 0_+} \lim_{x \to x(\mu)} \bar{G}(x, \mu) = \nabla_{xx} L(x^*, \lambda^*), \tag{4}$$

where $L(x, \lambda)$ is the Lagrangian function for NLP. Hence,

$$\lim_{\mu \to 0_+} \lim_{x^{(k)} \to x(\mu)} \begin{bmatrix} \bar{G}(x^{(k)}, \mu) & A(x^{(k)})^T \\ A(x^{(k)}) & -\mu I \end{bmatrix} = \begin{bmatrix} \nabla_{xx} L(x^*, \lambda^*) & A(x^*)^T \\ A(x^*) & 0 \end{bmatrix}. \tag{5}$$

The conditioning of the matrix (5) indicates the sensitivity of NLP to perturbations in the data. For well-conditioned problems, (5) is well conditioned, and hence, for small values of μ, so is the coefficient matrix of (2). Therefore, an accurate solution of (2), unlike (1), may be obtained from a suitable stable factorization [the Bunch–Parlett–Fletcher–Kaufman generalized Cholesky factorization is recommended (Refs. 9–11)].

It is also shown in Ref. 1 how the inertia of the matrices for (1) and (2) are related, and hence when the search direction $p^{(k)}$ obtained from (2) should be used. When $\nabla_{xx} \bar{p}(x^{(k)})$ is not positive definite, it is shown how to construct directions of negative curvature, weak solutions to (2), and directions of linear infinite descent, whichever is appropriate.

Numerical experience with the given algorithm and with an algorithm in which $\bar{G}(x^{(k)}, \mu)$ is approximated [using the symmetric rank-one secant update (Ref. 12)] has been very encouraging.

It is worth noting that (1) may also be rewritten as

$$\begin{bmatrix} \bar{G}(x^{(k)}, \mu) & A(x^{(k)})^T \\ A(x^{(k)}) & -\mu I \end{bmatrix} \begin{bmatrix} p^{(k)} \\ s^{(k)} \end{bmatrix} = -\begin{bmatrix} g^{(k)} - \sum_{i \in \mathcal{T}(x^{(k)})} \lambda_i(x^{(k)}, \mu) \nabla_x c_i(x^{(k)}) \\ 0 \end{bmatrix}, \tag{6}$$

where $s^{(k)}$ is another auxiliary vector. For small values of μ, the right-hand-side of (6) is well behaved [by virtue of (3)] and identical arguments to those given above show that $p^{(k)}$ may be obtained very accurately from (6) when NLP is well conditioned.

2. Problems Including Simple Bounds

In this section, we consider how to solve the nonlinear program BNLP introduced in Section 1. Our approach will be to use the quadratic penalty

function

$$p_2(x, \mu) = f(x) + (1/2\mu) \sum_{i \in \mathscr{E}} c_i(x)^2 + (1/2\mu) \sum_{i \in \mathscr{I}} c_i(x)_-^2$$

$$+ (1/2\mu) \sum_{i=1}^{n} [(x - l_i)_-^2 + (u_i - x_i)_-^2],$$

where we define

$$h(x)_- = \min(0, h(x)).$$

As usual, we will try to locate a local minimizer $x(\mu)$ of $p_2(x, \mu)$ and trace the locus of $x(\mu)$ as $\mu \to 0_+$. However, we shall try to exploit the special nature of the simple bound constraints in calculating a search direction $p^{(k)}$ from our current estimate $x^{(k)}$ of $x(\mu)$.

Define the index sets

$$\mathscr{V}(x) = \mathscr{E} \cup \{i: i \in \mathscr{I} \text{ and } c_i(x) \leqslant 0\},$$

$$\mathscr{L}(x) = \{i: 1 \leqslant i \leqslant n \text{ and } x_i \leqslant l_i\},$$

$$\mathscr{U}(x) = \{i: 1 \leqslant i \leqslant n \text{ and } x_i \geqslant u_i\}.$$

We may then write the penalty function $p_2(x, \mu)$ at $x = x^{(k)}$ as

$$p_2(x, \mu) \equiv \bar{p}(x) = f(x) + (1/2\mu) \sum_{i \in \mathscr{V}(x^{(k)})} c_i(x)^2$$

$$+ (1/2\mu) \left[\sum_{i \in \mathscr{L}(x^{(k)})} (x_i - l_i)^2 + \sum_{i \in \mathscr{U}(x^{(k)})} (x_i - u_i)^2 \right].$$

Let us assume, for simplicity, that $l_i < u_i$ (if $l_i = u_i$, a direct elimination of the variable x_i is possible), and define the vector $b(x)$ so that

$$b_i(x) = \begin{cases} x_i - l_i, & \text{if } i \in \mathscr{L}(x^{(k)}), \\ x_i - u_i, & \text{if } i \in \mathscr{U}(x^{(k)}), \\ 0, & \text{otherwise.} \end{cases}$$

For convenience we shall also assume that the variables have been ordered so that

$$\mathscr{L}(x^{(k)}) \cup \mathscr{U}(x^{(k)}) = \{1, \ldots, \nu\}, \quad \text{where } \nu = |\mathscr{L}(x^{(k)}) \cup \mathscr{U}(x^{(k)}).$$

This reordering is not necessary in practice. In this case,

$$\bar{p}(x) = f(x) + (1/2\mu) \sum_{i \in \mathscr{V}(x^{(k)})} c_i(x)^2 + (1/2\mu) \sum_{i=1}^{\nu} b_i(x)^2.$$

Consider applying Newton's method to finding a zero of the gradient of $\bar{p}(x)$. From $x^{(k)}$, we find a correction $p^{(k)}$ which satisfies

$$\nabla_{xx}\bar{p}(x^{(k)})p^{(k)} = -\nabla_x \bar{p}(x^{(k)}). \tag{7}$$

It is easy to show that $p^{(k)}$ also satisfies the augmented equations [cf. (2)]

$$\begin{bmatrix} \bar{G}_v & \bar{G}_r^T & A_v^T & I_v \\ \bar{G}_r & \bar{G}_s & A_s^T & 0 \\ A_v & A_s & -\mu I_t & 0 \\ I_v & 0 & 0 & -\mu I_v \end{bmatrix} \begin{bmatrix} p_v \\ p_s \\ r_t \\ r_v \end{bmatrix} = - \begin{bmatrix} g_v \\ g_s \\ c_t \\ b_v \end{bmatrix}, \tag{8}$$

where

$$\begin{bmatrix} \bar{G}_v & \bar{G}_r^T \\ \bar{G}_r & \bar{G}_s \end{bmatrix} = \bar{G}(x^{(k)}) \equiv \nabla_{xx} f(x) + \sum_{i \in \mathscr{V}(x^{(k)})} (\mu^{-1} c_i(x^{(k)}) \nabla_{xx} c_i(x^{(k)}),$$

$$(A_v, A_s) = A(x^{(k)}),$$

where $A(x^{(k)})$ is made up of the $t = |\mathscr{V}(x^{(k)})|$ rows $\nabla_x^T c_i(x^{(k)})$, $i \in \mathscr{V}(x^{(k)})$, where

$$\begin{bmatrix} x_v \\ x_s \end{bmatrix} = x^{(k)}, \qquad \begin{bmatrix} p_v \\ p_s \end{bmatrix} = p^{(k)}, \qquad \begin{bmatrix} b_v \\ b_s \end{bmatrix} = b, \qquad \begin{bmatrix} g_v \\ g_s \end{bmatrix} = \nabla_x f(x^{(k)}),$$

and where c_t is a vector made up of entries $c_i(x^{(k)})$, $i \in \mathscr{V}(x^{(k)})$.

Under the nondegeneracy assumption NDA, the usual argument that, for small μ, the coefficient matrix of (8) is a perturbation of the matrix which indicates the sensitivity of BNLP to small perturbations in the problem shows that accurate solutions of (8) will be possible whenever BNLP is well conditioned. However, (8) contains a useful structure. We now indicate how this structure may be exploited.

Let σ be a given positive number. We start by rescaling (8) by multiplying the first row and column by $\mu\sigma$. This yields

$$\begin{bmatrix} \sigma^2\mu^2\bar{G}_v & \mu\sigma\bar{G}_r^T & \sigma\mu A_v^T & \sigma\mu I_v \\ \sigma\mu\bar{G}_r & \bar{G}_s & A_s^T & 0 \\ \sigma\mu A_v & A_s & -\mu I_t & 0 \\ \sigma\mu I_v & 0 & 0 & -\mu I_v \end{bmatrix} \begin{bmatrix} \bar{p}_v \\ p_s \\ r_t \\ r_v \end{bmatrix} = - \begin{bmatrix} \sigma\bar{g}_v \\ g_s \\ c_t \\ \mu\bar{b}_v \end{bmatrix}, \tag{9}$$

where

$$\bar{p}_v = \sigma^{-1}\mu^{-1}p_v, \qquad \bar{b}_v = \mu^{-1}b_v, \qquad \bar{g}_v = \mu g_v.$$

The last row of this block system gives

$$r_v = \sigma\bar{p}_v + \bar{b}_v. \tag{10}$$

If we use (10) to eliminate r_ν from (9), we obtain

$$\begin{bmatrix} \sigma^2\mu(I_\nu+\mu\bar{G}_\nu) & \sigma\mu\bar{G}_r^T & \sigma\mu A_\nu^T \\ \sigma\mu\bar{G}_r & \bar{G}_s & A_s^T \\ \sigma\mu A_\nu & A_s & -\mu I_t \end{bmatrix} \begin{bmatrix} \bar{p}_\nu \\ \bar{p}_s \\ r_t \end{bmatrix} = -\begin{bmatrix} \sigma(\bar{g}_\nu+\mu\bar{b}_\nu) \\ g_s \\ c_t \end{bmatrix}. \tag{11}$$

Finally, rescaling (11) by multiplying the first row and column by $\mu^{-1/2}$, we obtain

$$\begin{bmatrix} \sigma^2(I_\nu+\mu\bar{G}_\nu) & \sigma\mu^{1/2}\bar{G}_r^T & \sigma\mu^{1/2}A_\nu^T \\ \sigma\mu^{1/2}\bar{G}_r & \bar{G}_s & A_s^T \\ \sigma\mu^{1/2}A_\nu & A_s & -\mu I_t \end{bmatrix} \begin{bmatrix} \hat{p}_\nu \\ p_s \\ r_t \end{bmatrix} = -\begin{bmatrix} \hat{g}_\nu+\hat{b}_\nu \\ g_s \\ c_t \end{bmatrix}, \tag{12}$$

where

$$\hat{p}_\nu = \mu^{1/2}\bar{p}_\nu = \sigma\mu^{-1/2}p_\nu,$$

$$\hat{g}_\nu = \sigma\mu^{-1/2}\bar{g}_\nu = \sigma\mu^{1/2}g_\nu,$$

$$\hat{b}_\nu = \sigma\mu^{1/2}\bar{b}_\nu = \sigma\mu^{-1/2}b_\nu.$$

Now, from (3),

$$\lim_{\mu\to 0_+}\lim_{x^{(k)}\to x} \mu^{-1}b_\nu(x^{(k)}) = w^*,$$

positive Lagrange multipliers associated with the simple bound constraints. Hence,

$$\lim_{\mu\to 0_+}\lim_{x^{(k)}\to x(\mu)} \hat{g}_\nu(x^{(k)}) = 0 = \lim_{\mu\to 0_+}\lim_{x^{(k)}\to x(\mu)} \hat{b}_\nu(x^{(k)}).$$

Thus, the right-hand-side of (12) will remain bounded as μ approach zero. We also have, from (4),

$$\lim_{\mu\to 0_+}\lim_{x^{(k)}\to x(\mu)} \begin{bmatrix} \sigma^2(I_\nu+\mu\bar{G}_\nu) & \sigma\mu^{1/2}\bar{G}_r^T & \sigma\mu^{1/2}A_\nu^T \\ \sigma\mu^{1/2}\bar{G}_r & \bar{G}_s & A_s^T \\ \sigma\mu^{1/2}A_\nu & A_s & -\mu I_t \end{bmatrix}$$

$$= \begin{bmatrix} \sigma^2 I_\nu & 0 & 0 \\ 0 & \bar{H}_s & \bar{A}_s^T \\ 0 & \bar{A}_s & 0 \end{bmatrix}, \tag{13}$$

where

$$\begin{bmatrix} \bar{H}_\nu & \bar{H}_r^T \\ \bar{H}_r & \bar{H}_s \end{bmatrix} = \nabla_{xx}L(x^*, \lambda^*),$$

the Hessian matrix of the Lagrangian function associated with BNLP and

$$(\bar{A}_\nu, \bar{A}_s) = A(x^*).$$

We would expect that the submatrix

$$\begin{bmatrix} \bar{H}_s & \bar{A}_s^T \\ \bar{A}_s & 0 \end{bmatrix} \tag{14}$$

normally to be well conditioned, as the conditioning of this matrix indicates the sensitivity of the related nonlinear programming problem

$$\begin{aligned} \text{(BNLP')} \quad & \text{minimize } f(x), \\ & \text{subject to } c_i(x) = 0, \qquad i \in \mathscr{E}, \\ & \qquad\qquad c_i(x) \geq 0, \qquad i \in \mathscr{I}, \\ & \qquad\qquad l_i \leq x_i \leq u_i, \qquad i = 1, \ldots, n, \\ & \qquad\qquad x_i = b_i, \qquad i = 1, \ldots, \nu, \end{aligned}$$

to perturbations in the data. Under assumption NDA, the local solution x^* to BNLP is also a solution to BNLP'. Thus, if BNLP is well conditioned, so is BNLP', and therefore so is (14). Furthermore, the matrix (13) will also be well conditioned if σ is suitably chosen. Specifically, if σ^2 is chosen as a good approximation to the 2-norm of (14), the condition number of (13) will be close to that of (14).

To summarize, we may obtain an accurate solution to (7) by solving (12) and recovering

$$p_\nu = \mu^{1/2}\sigma^{-1}\hat{p}_\nu.$$

The system of equations (12) is smaller than (8). Moreover, the coefficient matrix of (12) is structurally identical to the leading 3×3 blocks of the matrix for (8). Hence, reducing the size of the system solved has not increased the storage requirements above that needed to maintain

$$\begin{bmatrix} \bar{G}(x, \mu) & A(x)^T \\ A(x) & -\mu I_t \end{bmatrix};$$

the simple bound constraints have not increased the dimension of the linear system to be solved.

The remaining issues relating to solving BNLP are concerned with whether we should be solving (8) at all. We would normally only choose $p^{(k)}$ from the Newton equations if $\nabla_{xx}\bar{p}(x^{(k)})$ were positive definite. We relate the inertia of $\nabla_{xx}\bar{p}(x^{(k)})$ to that of the coefficient matrix of (12) as follows.

Definition 2.1. The *inertia* of a real symmetric matrix M is the triple $\text{In}(M) = (m_+, m_-, m_0)$, where m_+, m_-, m_0 are the numbers of positive, negative, and zero eigenvalues of M.

Theorem 2.1. Let K be the coefficient matrix of (12), and let $t = |V(x^{(k)})|$. Then,

$$\text{In}(K) = \text{In}(\nabla_{xx}\bar{p}(x^{(k)})) + (0, t, 0).$$

Proof. Let $\bar{K}$ be the coefficient matrix of (8). Then (Ref. 1, Theorem 3.2),

$$\text{In}(\bar{K}) = \text{In}(\nabla_{xx}\bar{p}(x^{(k)})) + (0, \nu + t, 0). \tag{15}$$

By Sylvester's law of inertia, the inertia of $\bar{K}$ and of the coefficient matrix $\hat{K}$ of (9) are identical. Performing a block pivot on the last block of $\hat{K}$ and using Ref. 1, lemma 3.1, we have

$$\text{In}(\bar{K}) = \text{In}(\hat{K}) = (0, \nu, 0) + In(\bar{K}), \tag{16}$$

where $\bar{K}$ is the coefficient matrix of (11). Again, Sylvester's law ensures that the inertia of $\bar{K}$ and K are identical. Therefore, from (15) and (16), we have

$$\begin{aligned}\text{In}(K) &= \text{In}(\bar{K}) = \text{In}(\bar{K}) - (0, \nu, 0)\\ &= \text{In}(\nabla_{xx}\bar{p}(x^{(k)})) + (0, \nu + t, 0) - (0, \nu, 0).\end{aligned}$$ □

Corollary 2.1. Let K be as in Theorem 2.1, and let $\text{In}(K) = (k_+, k_-, k_0)$. Then:

(i) $\nabla_{xx}\bar{p}(x^{(k)})$ is positive definite if and only if $k_0 = 0$ and $k_- = t$;

(ii) $\nabla_{xx}\bar{p}(x^{(k)})$ is positive definite but singular if and only if $k_- = t$ and $k_0 > 0$;

(iii) $\nabla_{xx}\bar{p}(x^{(k)})$ is indefinite if and only if $k_- > t$.

Proof. It follows directly from Theorem 2.1. □

Corollary 2.1 indicates that we should only find $p^{(k)}$ from (12) when $k_0 = 0$ and $k_- = t$. When $k_- > t$, we should locate a direction of negative curvature. When $k_- = t$ and $k_0 > 0$, we should either find a weak solution of (12) or a direction of linear infinite descent. The latter two possibilities are mutually exclusive and occur whenever (12) is or is not compatible. The mechanics of establishing which one of the possibilities of Corollary 2.1 is satisfied at $x^{(k)}$ and of locating the alternative search directions in cases (ii) and (iii) may all be accomplished if a generalized Cholesky factorization of K is known (see Section 1.1). The details are essentially identical to those described in Ref. 1, Section 3 and will not be repeated here.

We note that the reduction in the size of (12) from (8) is not as large as that normally associated with projected Lagrangian methods for nonlinear programming (see, e.g., Ref. 7); such systems typically involve $n-\nu+t$, rather than our $n+t$ equations and variables. Such a reduction is only possible in the limit as $\mu \to 0_+$ with our method as the coefficient matrix approaches (13) and the determination of $\hat{p}_\nu$ becomes trivial.

One possibility which we have not considered here is that (12) might be solved approximately by finding

$$\hat{p}_\nu = -(\hat{g}_r + \hat{b}_r)/\sigma^2,$$

and then recovering

$$\begin{bmatrix} \bar{G}_s & A_s^T \\ A_s & -\mu I_t \end{bmatrix} \begin{bmatrix} p_s \\ r_t \end{bmatrix} = -\begin{bmatrix} g_s + \sigma\mu^{1/2}\bar{G}_r\hat{p}_\nu \\ c_t + \sigma\mu^{1/2}A_\nu\hat{p}_\nu \end{bmatrix}. \tag{17}$$

Such a method might also incorporate iterative refinement to try to satisfy (12) more accurately. The advantage of this scheme is that the system (17) may be considerably smaller than (12) when ν is large. We leave this for future research.

3. Minimizing a Sum of Absolute Values

We turn to the second class of problems under consideration, the following problem:

$$(\mathrm{L_1P}) \quad \underset{x\in\mathcal{R}^n}{\text{minimize}}\ F_1(x), \text{ where } F_1(x) = \sum_{i=1}^{m} |f_i(x)|.$$

Such problems arise naturally in a number of different areas; in particular, robust methods for data fitting are frequently of this form. For obvious reasons, this problem is often known as the l_1-problem. Our particular concern will be with $\mathrm{L_1P}$ when some or all of the functions $f_i(x)$ are nonlinear. There are many good algorithms for the linear l_1-problem, most of which are based upon linear programming techniques (see, for a survey, Ref. 13), but relatively few for the nonlinear problem. Before we describe our approach, we shall briefly review existing methods.

3.1. Existing Algorithms. The algorithm of Osborne and Watson (Ref. 14) proceeds by linearizing the nonlinear functions $f_i(x)$ and by solving a sequence of linear l_1-problems. As might be expected, the convergence of such a method is often relatively slow. Improvements on this idea are proposed by Mclean and Watson (Ref. 15) in which the above method is used until the set of functions $f_i(x)$ which are thought to be zero at the

solution of L_1P are identified. The algorithm then switches to a variant of Newton's method in which the sum of the nonzero functions is minimized subject to the zero functions remaining zero. A similar method, in which approximate (rather than exact) second derivatives are used in the Newton stage of the algorithm is given by Hald and Madsen (Ref. 16). Bartels and Conn (Ref. 17) propose a method in which second derivatives are used in both phases of the above algorithm.

A second class of methods is based upon the observation that L_1P may be reformulated as the nonlinear programming problem

$$(\mathrm{NLP_1P}) \quad \underset{x\in\mathcal{R}^n,\, u\in\mathcal{R}^m}{\text{minimize}} \; u_i,$$

$$\text{subject to } u_i - f_i(x) \geq 0, \qquad i = 1, \ldots, m,$$

$$u_i + f_i(x) \geq 0, \qquad i = 1, \ldots, m.$$

El-Attar et al (Ref. 18) propose that $NLPL_1P$ be solved by minimizing a sequence of differentiable penalty functions. They recognize the ill-conditioning introduced by their approach and try to avoid it by an extrapolation process. Murray and Overton (Ref. 13) suggest using a projected Lagrangian method based upon $NLPL_1P$ which exploits the special structure of the problem. In order to globalize the algorithm, they argue that, as $F_1(x)$ is the natural merit function for L_1P, there is no need to introduce an auxiliary penalty function to force global convergence, merely to use $F_1(x)$ and to maintain feasibility at each iteration. However, we agree with Conn (Ref. 19), who believes that it is a mistake to try to globalize a locally convergent method for nonlinear programming by choosing a merit function solely on the basis that the search direction for the local method is a descent direction for the merit function. Conn argues that it is the merit function which should be chosen first and the search direction subsequently constructed to minimize local approximations of the given merit function. Many of the problems associated with the transition from the global to the asymptotic convergence phases [the Maratos effect (Ref. 20)] of projected Lagrangian algorithms for nonlinear programming (see, e.g., Ref. 7) are directly attributable to imposing a merit function upon a search direction. It is unclear that these same problems could not affect the convergence of the algorithm described in Ref. 13. The previously mentioned method of Bartels and Conn (Ref. 17) is a realization of Conn's philosophy in which a direct minimization of $F_1(x)$ is attempted.

3.2. New Method. Our method for solving L_1P is to proceed via $NLPL_1P$. One obvious approach is to solve $NLPL_1P$ by means of a sequential

minimization of the quadratic penalty function

$$\bar{p}(x, u, \bar{\mu}) = \sum_{i=1}^{m} u_i + (1/2\bar{\mu}) \sum_{i=1}^{m} (u_i - f_i(x))_-^2 + (u_i + f_i(x))_-^2.$$

It is easy to show that, in such an approach, it is possible to take special account of the auxiliary variables u_i, $i = 1, \ldots, m$. However, we propose that the auxiliary variables should be explicitly removed from $\bar{p}(x, u, \bar{\mu})$ and a simpler penalty function $p(x, \mu)$ obtained. We proceed as follows.

Suppose we fix x and $\bar{\mu}$. We observe that

$$\bar{p}(x, u, \bar{\mu}) = \sum_{i=1}^{m} \Theta_i(x, u_i, \bar{\mu}), \tag{18}$$

where

$$\Theta_i(x, u_i, \bar{\mu}) = u_i + \{(u_i - f_i(x))_-^2 + (u_i + f_i(x))_-^2\}/2\bar{\mu}.$$

For fixed $\bar{\mu}$, our aim is to minimize $\bar{p}(x, u, \bar{\mu})$. If we knew which value of x minimized $\bar{p}(x, u, \bar{\mu})$, it would be easy to find the corresponding u, as each of the $\Theta_i(x, u, \bar{\mu})$ is a piecewise quadratic one-dimensional function of u_i. We therefore define

$$p(x, \bar{\mu}) = \min_{u \in \mathcal{R}^m} \bar{p}(x, u, \bar{\mu}), \tag{19}$$

and observe that the minimizer of $\bar{p}(x, \bar{\mu})$ over x is the same as the minimizer of $p(x, u, \bar{\mu})$ over x and u. Let

$$\bar{u}(x, \bar{\mu}) = \arg \min_{u} \bar{p}(x, u, \bar{\mu}).$$

Then,

$$\bar{u}_i(x, \bar{\mu}) = \arg \min_{u_i} \Theta_i(x, u_i, \bar{\mu}).$$

It is straightforward to show that

$$\bar{u}_i(x, \bar{\mu}) = \begin{cases} |f_i(x)| - \bar{\mu}, & \text{if } \bar{\mu}/2 < |f_i(x)|, \\ -\bar{\mu}/2, & \text{if } |f_i(x)| \leq \bar{\mu}/2. \end{cases}$$

This then leads to

$$\Theta_i(x, \bar{\mu}_i(x, \bar{\mu}), \bar{\mu}) = \begin{cases} |f_i(x)| - \bar{\mu}/2, & \text{if } \bar{\mu}/2 < |f_i(x)|, \\ \bar{\mu}^{-1} f_i(x) - \bar{\mu}/4, & \text{if } |f_i(x)| \leq \bar{\mu}/2. \end{cases} \tag{20}$$

If we define index sets

$$\bar{\mathcal{I}}_+ = \{i: f_i(x) > \bar{\mu}/2\},$$

$$\bar{\mathcal{I}}_- = \{i: f_i(x) < -\bar{\mu}/2\},$$

$$\bar{\mathcal{I}}_0 = \{i: -\bar{\mu}/2 \leq f_i(x) \leq \bar{\mu}/2\},$$

and combine (18), (19), (20), we obtain

$$p(x,\bar{\mu}) = \sum_{i\in\bar{\mathcal{F}}_+} (f_i(x) - \bar{\mu}/2) + \sum_{i\in\bar{\mathcal{F}}_-} (-f_i(x) - \bar{\mu}/2) + \bar{\mu}^{-1} \sum_{i\in\bar{\mathcal{F}}_0} (f_i(x)^2 - \bar{\mu}^2/4).$$

Finally, if we let $\mu = \bar{\mu}/2$, we have

$$p(x,\mu) = \sum_{i\in\mathcal{J}_+} (f_i(x) - \mu) + \sum_{i\in\mathcal{J}_-} (-f_i(x) - \mu) + (1/2\mu) \sum_{i\in\mathcal{J}_0} (f_i(x)^2 - \mu^2), \tag{21}$$

where

$$\mathcal{J}_+ \equiv \mathcal{J}_+(x,\mu) = \{i\colon f_i(x) > \mu\},$$

$$\mathcal{J}_- \equiv \mathcal{J}_-(x,\mu) = \{i\colon f_i(x) < -\mu\},$$

$$\mathcal{J}_0 \equiv \mathcal{J}_0(x,\mu) = \{i\colon -\mu \leqslant f_i(x) \leqslant \mu\}.$$

Our proposal is thus that L_1P be solved by a sequential minimization of $p(x,\mu)$ as $\mu \to 0_+$. It is interesting to observe that (21) bears a strong resemblance to the loss functions used to obtain robust regression estimates (Ref. 21). We have not explored this relationship, but feel this may be worthwhile.

3.3. Derivatives of $p(x,\mu)$. We obtain the following derivatives:

$$\nabla_x p(x,\mu) = A_+(x)^T e_+ - A_-(x)^T e_- + A_0^T \lambda_0(x,\mu),$$

$$\nabla_{xx} p(x,\mu) = \bar{G}(x,\mu) + \mu^{-1} A_0^T(x) A_0(x),$$

where A_+, A_-, A_0 are made up of the rows $\nabla_x^T f_i(x)$ for $i \in \mathcal{J}_+$, $\mathcal{J}_-$, $\mathcal{J}_0$, respectively; e_+ and e_- are appropriately dimensioned vectors of 1's; $\lambda_0(x,\mu)$ is made up of entries

$$\lambda_i(x,\mu) = \mu^{-1} f_i(x), \qquad i \in \mathcal{J}_0;$$

and

$$\bar{G}(x,\mu) = \sum_{i\in\mathcal{J}_+} \nabla_{xx} f_i(x) \sum_{i\in\mathcal{J}_-} \nabla_{xx} f_i(x) + \sum_{i\in\mathcal{J}_0} \lambda_i(x,\mu) \nabla_{xx} f_i(x).$$

If we extend the definition of $\lambda_0(x,\mu)$ to $\lambda(x,\mu)$, where

$$\lambda_i(x,\mu) = 1, \qquad i \in \mathcal{J}_+,$$

$$\lambda_i(x,\mu) = -1, \qquad i \in \mathcal{J}_-,$$

$$\lambda_i(x,\mu) = \mu^{-1} f_i(x), \qquad i \in \mathcal{J}_0,$$

the derivatives may be written in the slightly more compact form

$$\nabla_x p(x, \mu) = A(x)^T \lambda(x, \mu),$$
$$\nabla_{xx} p(x, \mu) = \bar{G}(x, \mu) + \mu^{-1} A_0(x)^T A_0(x),$$

where

$$\bar{G}(x, \mu) = \sum_{i=1}^{m} \lambda_i(x, \mu) \nabla_{xx} f_i(x).$$

By definition,

$$-1 \leq \lambda_i(x, \mu) \leq 1.$$

As the functions $f_i(x)$ are twice continuously differentiable, both $\nabla_x p(x, \mu)$ and $\bar{G}(x, \mu)$ are continuous for all $\mu > 0$. Furthermore, by definition of $x(\mu)$,

$$A(x(\mu))\lambda(x(\mu), \mu) = 0.$$

Therefore, if

$$\lim_{\mu \to 0} \lim_{x \to x(\mu)} \lambda_i(x, \mu) = \lambda_i^*,$$

the Lagrange multipliers λ_i^* satisfy the first-order optimality conditions

(i) $-1 \leq \lambda_i^* \leq 1$,
(ii) $A(x^*)^T \lambda^* = 0$,

for L_1P (see, e.g., Ref. 18). Moreover,

$$\lim_{\mu \to 0_+} \lim_{x \to x(\mu)} \bar{G}(x, \mu) = \nabla_{xx} L(x^*, \lambda^*),$$

where

$$L(x, \lambda) = \sum_{i=1}^{m} \lambda_i f_i(x).$$

Notice that the above statements imply that, if

$$\lim_{\mu \to 0} \lim_{x \to x(\mu)} x = x^* \quad \text{and} \quad \lim_{\mu \to 0} \lim_{x \to x(\mu)} \lambda_i(x, \mu) = \lambda_i^*, \tag{22}$$

then x^* is a first-order stationary point for L_1P. This is a stronger conclusion than normally attainable for penalty function methods for general nonlinear programs where convergence to nonfeasible points is possible. We note that (22) is satisfied if we assume that the gradients $\nabla_x f_i(x^*)$, $i \in \mathcal{I}_0$, are linearly independent (Ref. 22). Moreover, if μ_k is any sequence such that $\mu_k \to 0_+$ and

$$\lim_{\mu_k \to 0_+} \lim_{x \to x(\mu_k)} \lambda_i(x, \mu_k) = \lambda_i^*,$$

for some λ_i^*, the same conclusion is true. That at least one such sequence exists follows from the boundedness of $\lambda_i(x, \mu)$.

3.4. Minimizing $p(x, \mu)$. As $p(x, \mu)$ is piecewise twice continuously differentiable, the obvious way of generating a search direction $p^{(k)}$ from an estimate $x^{(k)}$ of $x(\mu)$ is to apply Newton's method and its variants to $p(x, \mu)$. The normal difficulty, the increasing ill-conditioning of $\nabla_{xx}p(x, \mu)$ as μ decreases, is present. Therefore, instead of calculating $p^{(k)}$ from the Newton equations

$$\nabla_{xx}p(x^{(k)}, \mu)p^{(k)} = -\nabla_x p(x^{(k)}, \mu),$$

we obtain it from

$$\begin{bmatrix} \bar{G}(x^{(k)}, \mu) & A_0(x^{(k)})^T \\ A_0(x^{(k)}) & -\mu I_t \end{bmatrix} \begin{bmatrix} p^{(k)} \\ r^{(k)} \end{bmatrix} = -\begin{bmatrix} A_+(x^{(k)})^T e_+ - A_-(x^{(k)})^T e_- \\ f_0(x^{(k)}) \end{bmatrix}, \tag{23}$$

or from

$$\begin{bmatrix} \bar{G}(x^{(k)}, \mu) & A_0(x^{(k)})^T \\ A_0(x^{(k)}) & -\mu I_t \end{bmatrix} \begin{bmatrix} p^{(k)} \\ s^{(k)} \end{bmatrix} = -\begin{bmatrix} A_+(x^{(k)})^T e_+ - A_-(x^{(k)})^T e_- + A_0(x^{(k)})^T \lambda_0(x^{(k)}, \mu) \\ 0 \end{bmatrix}, \tag{24}$$

where $t = |I_0|$ and f_0 is made up of entries $f_i(x^{(k)})$, $i \in \mathcal{J}_0$.

For convenience, let us make the following nondegeneracy assumption:

(NDA2) The gradients $\nabla_x f_i(x^*)$, $i \in \mathcal{J}_0$, are linearly independent and the associated Lagrange multipliers satisfy $-1 < \lambda_i^* < 1$, for all $i \in \mathcal{J}_0$.

We shall consider relaxing this assumption in Section 7.

The usual arguments indicate that, under NDA2, either (23) or (24) allow an accurate determination of $p^{(k)}$. Furthermore, we have the following theorem.

Theorem 3.1. Let K be the coefficient matrix of (23). Let $In(K) = (k_+, k_-, k_0)$, and let $t = |I_0|$. Then:

(i) $\nabla_{xx}p(x^{(k)}, \mu)$ is positive definite if and only if $k_0 = 0$ and $k_- = t$;

(ii) $\nabla_{xx}p(x^{(k)}, \mu)$ is positive semidefinite but singular if and only if $k_0 > 0$ and $k_- = t$;

(iii) $\nabla_{xp}(x^{(k)}, \mu)$ is indefinite if and only if $k_- > t$.

Proof. The proof is identical to Theorem 3.2 and its corollaries in Ref. 1. □

As usual, we should only calculate $p^{(k)}$ from (23) when (i) above is satisfied. The calculation of directions of infinite descent or weak solutions to (23) or (24) in cases (ii) and (iii) is identical to that described in Section 3 of Ref. 1.

3.5. Practicalities. The obvious way to use the preceding theoretical development in an algorithm to solve L_1P is to incorporate the function (21) in the SUMT framework described in Section 1.1. Such an apporach should clearly succeed. However, there is a drawback.

Recall the definition of $\mathcal{I}_0$,

$$\mathcal{I}_0(x, \mu) = \{i: -\mu \leq f_i(x) \leq \mu\}.$$

For fixed μ, the minimization of $p(x, \mu)$ is implicitly trying to find $\mathcal{I}_0(x(\mu), \mu)$. Under NDA2, if μ is small enough, there is a neighborhood of $x(\mu)$ in which $p(x, \mu)$ is twice continuously differentiable. In such a neighborhood, Newton's method is normally very successful in determining $x(\mu)$. Moreover, in this neighborhood,

$$\mathcal{I}_0(x(\mu), \mu) = \mathcal{I}_0(x^*, 0),$$

and thus $\mathcal{I}_0(x(\mu), \mu)$ predicts which of the $f_i(x)$ are zero at the solution.

Let us suppose that we have found $x(\mu)$ for some fixed μ, and we now reduce μ to $\bar{\mu}$ (in our description of SUMT, $\bar{\mu} = 10^{-2} \times \mu$). The next iteration of SUMT starts at $x = x(\mu)$. It is unreasonable to expect that

$$\mathcal{I}_0(x(\mu), \bar{\mu}) = \mathcal{I}_0(x(\mu), \mu).$$

Therefore, between $x(\mu)$ and $x(\bar{\mu})$, there will often be second derivative discontinuity of $p(x, \mu)$. Such a discontinuity is likely to slow the convergence of x to $x(\mu)$; note that the asymptotic convergence rate should still be quadratic. We take the point of view that the aim of our algorithm is to identify $\mathcal{I}_0(x^*, 0)$ by way of $\mathcal{I}_0(x(\mu), \mu)$ and that we should exploit the fact that $\mathcal{I}_0(x(\mu), \mu)$ should give a prediction of $\mathcal{I}_0(x(\bar{\mu}), \bar{\mu})$. To this end, let $\mathcal{I}$ be a given subset of $\{1, \ldots, m\}$. Define the auxiliary function

$$p(x, \mu, \mathcal{I}) = \sum_{i \in \mathcal{I}_+ \setminus \mathcal{I}} (f_i(x) - \mu) + \sum_{i \in \mathcal{I}_- \setminus \mathcal{I}} (-f_i(x) - \mu) + (1/2\mu) \sum_{i \in \mathcal{I}_0 \cup \mathcal{I}} (f_i(x)^2 - \mu^2), \tag{25}$$

where the set $\mathcal{I}_+$, $\mathcal{I}_-$, $\mathcal{I}_0$ are as given in (21). We note that

$$p(x, \mu, \varnothing) = p(x, \mu).$$

Furthermore, let $x(\mu, \mathcal{I})$ be a local minimizer of $p(x, \mu, \mathcal{I})$. We propose that L_1P is solved by the following scheme:

Set $\mu := \mu_{\text{start}}$, $x_0 := x_{\text{start}}$, $\mathcal{I} = \varnothing$.

Until $\mu < \mu_{\min}$, do these operations:

Find $x(\mu, \mathcal{I})$ and $\mathcal{I}_0(x(\mu, \mathcal{I}), \mu)$ by minimizing $p(x, \mu, \mathcal{I})$ starting at x_0.

Let $x_0 := x(\mu, \mathcal{I})$.

If $\mathcal{I} \subseteq \mathcal{I}_0(x_0, \mu)$, minimize $p(x, \mu)$ starting at x_0 to find $x(\mu)$ (*); let $x_0 := x(\mu)$.

Let $\mathscr{I} := \mathscr{I}_0(x_0, \mu)$, $\mu := \mu \times 10^{-2}$.

The idea behind this scheme is that, if

$$\mathscr{I}_0(x(\mu), \mu) = \mathscr{I}_0(x(\bar{\mu}), \bar{\mu}),$$

$p(x, \bar{\mu}, \mathscr{I}_0(x(\mu), \mu))$ is twice continuously differentiable in a neighborhood of $x(\bar{\mu})$ which includes $x(\mu)$ and will have a local minimizer at $x(\bar{\mu})$. The part of the algorithm indicated by (*) is merely to allow for incorrect choices of $\mathscr{I}_0(x(\bar{\mu}), \bar{\mu})$. The minimization of $p(x, \mu, \mathscr{I})$ may be accomplished in essentially the same fashion as described in Section 3.4, the only differences being that the definitions of the gradients and Hessian matrices in Section 3.3 must allow for the differences between the summation indices in (21) and (25).

3.6. Discussion. We discuss briefly the dominant storage costs and the approximate cost per iteration of our algorithm. Suppose that $A_0(x^{(k)})$ is $t \times n$. The principal storage requirement is the space needed to store the matrix

$$\begin{bmatrix} \bar{G}(x^{(k)}, \mu) & A_0(x^{(k)})^T \\ A_0(x^{(k)}) & -\mu \mathscr{I}_t \end{bmatrix} \tag{26}$$

and its symmetric factorization. Using symmetry and overwriting the factors of (26) on (26), the dominant storage cost is $(n+t)^2/2$ locations. As we do not usually know how large t may be, $t \leqslant n$, we must normally reserve $2n^2$ locations.

The dominant cost per iteration (excluding the often considerable costs of evaluating problem functions, common to all methods) is that required to factor (26). The factorization is dominated by $(n+t)^3/6$ flops. Here, a flop is a floating point multiplication or division. Hence, the dominant cost may be as large as $4n^3/3$ flops.

As a comparison, the methods of Bartels and Conn (Ref. 17) and Murray and Overton (Ref. 13) maintain the matrix $A_0(x^{(k)})$, its LQ factorization $A_0(x^{(k)}) = (L, 0)Q^T$, where Q is $n \times n$ and L is $t \times t$, and the matrix $Z^T H Z$, where H is an approximation to the Hessian of the Lagrangian function and Z^T includes the last $n-t$ rows of Q^T.

If we assume that $Z^T H Z$ is maintained without requiring H and the factor L is overwritten on $A_0(x^{(k)})$, the dominant storage requirements of such a method is $(n-t)^2/2 + tn + n^2 = 3n^2/2 + t^2/2$. If H is maintained, this cost increases to $3n^2/2 + tn$. Thus, once again, we must allow $2n^2$ locations (with H, $5n^2/2$ locations).

The dominant cost per iteration is that required to factorize $A_0(x^{(k)})$ and, possibly, to form and factor $Z^T HZ$. Depending upon which method is used to factor $A_0(x^{(k)})$ (Householder's or Givens' method are the usual candidates; see, e.g., Ref. 23), the factorization requires roughly between $t^2(n-t/3)$ and $2t^2(n-t/3)$ flops. If $Z^T HZ$ is formed and factored, an extra $n^2(n-t)+(n-t)^3/6$ flops are used. Once again, the dominant cost may be as high as $4n^3/3$ flops.

3.7. Comments. In general, it is necessary that we let μ approach zero if we wish to find a local solution to L_1P. However, it is not always the case.

Firstly, suppose that $\mathcal{I}_0(x(\bar{\mu}), \bar{\mu})$ is empty for some value $\bar{\mu}$ of μ. Then, it is easy to see that

$$x(\mu) = x(\bar{\mu}), \qquad \text{for all } \mu \leq \bar{\mu},$$

and hence

$$x^* = x(\bar{\mu}).$$

Secondly, suppose that

$$\mathcal{I}_+(x(\bar{\mu}), \bar{\mu}) = \varnothing = \mathcal{I}_-(x(\bar{\mu}), \bar{\mu}),$$

for some value $\bar{\mu}$ of μ, and that

$$\mu \equiv \max_{1 \leq i \leq m} f_i(x(\bar{\mu})).$$

Then,

$$x(\mu) = x(\bar{\mu}), \qquad \text{for all } \underline{\mu} \leq \mu \leq \bar{\mu}.$$

In particular, if $\underline{\mu} = 0$,

$$x^* = x(\bar{\mu}).$$

In these two cases, $p(x, \mu)$ is therefore an exact penalty function.

The second case, with $\underline{\mu} = 0$, has important consequences for solving sets of nonlinear equations, i.e., in trying to find a point x^* for which

$$f_i(x^*) = 0, \qquad \text{for all } 1 \leq i \leq n.$$

A common approach to such problems is to define a function $h: \mathcal{R}^n \to \mathcal{R}_+$ such that $h(0) = 0$ and to solve the given nonlinear equations by minimizing $h(f(x))$. The major advantage of such an approach is that progress toward the solution can be measured by the size of $h(f(x))$; this allows one to construct globally convergent methods for solving nonlinear equations. The most popular choice for $h(x)$ is $\|x\|_2^2$; another possibility is $\|x\|_1$. We advocate the use of $p(x, \mu)$, with decreasing values of μ if necessary. Such

a choice can be viewed as a compromise between the differentiability of $\|x\|_2^2$ and the robustness of $\|x\|_1$; our approach combines both virtues. If $x(\mu) \to x^*$, at which $f_i(x^*) = 0$, $1 \leq i \leq n$, it is normally that case that $\mathcal{I}_0(x(\mu), \mu) = \{1, \ldots, n\}$, for all μ smaller than some threshold $\bar{\mu}$. In this case, $f_i(x(\bar{\mu})) = 0$, $1 \leq i \leq n$, and we need not reduce μ any further.

Our second comment is that all of the preceding theory can be extended to the constrained l_1-problem,

$$\begin{aligned} (\mathrm{CL_1P}) \quad & \underset{x R^n}{\text{minimize}}\ F_1(x), \\ & \text{subject to } c_i(x) = 0, \qquad i \in \mathcal{E}, \\ & \qquad\qquad c_i(x) \geq 0, \qquad i \in \mathcal{I}, \end{aligned}$$

by the obvious sequential minimization of the penalty function

$$\begin{aligned} p(x, \mu) = & \sum_{i \in \mathcal{I}_+} (f_i(x) - \mu) + \sum_{i \in \mathcal{I}_-} (-f_i(x) - \mu)' \\ & + (1/2\mu) \sum_{i \in \mathcal{I}_0} (f_i(x)^2 - \mu^2) \\ & + (1/2\mu) \sum_{i \in \mathcal{E}} c_i(x)^2 + (1/2\mu) \sum_{i \in \mathcal{I}} c_i(x)_-^2. \end{aligned}$$

Finally, we have only considered solving $\mathrm{NLPL_1P}$ by using the quadratic penalty function. This is by no means the only simple differentiable penalty function which may be used to solve the l_1-problem. For instance, the appropriate variant of the logarithmic barrier function (see, Ref. 2) for $\mathrm{L_1P}$ is the function

$$\begin{aligned} p_B(x, \mu) = & \sum_{i=1}^{m} (\sqrt{(\mu^2 + f_i(x)^2)} \\ & - \mu \log_e (\mu + \sqrt{(\mu^2 + f_i(x)^2)})) + m\mu(1 - \log_e 2\mu). \end{aligned}$$

This function may be derived in essentially the same way that $p(x, \mu)$ was obtained in Section 3.2. Although this function appears rather complicated, it has a significant advantage in that it is twice continuously differentiable at all points at which the $f_i(x)$ are. This may be important when the soluton to $\mathrm{L_1P}$ is degenerate (see Section 7).

4. Testing

In this section, we give results of some preliminary numerical testing of the algorithm outlined in Section 3. Our aim has not been to present the results of a highly polished code, but merely to indicate that our approach

is promising. We have implemented the scheme given in Section 3.5 with $\mu_{\text{start}} = 10^{-1}$ and $\mu_{\min} = 10^{-8}$. The problems we have solved and the starting values x_{start} we use are as follows:

Problem 4.1. Example 3 in Ref. 16.
Starting point $x_{\text{start}} = (0.25, 0.39, 0.415, 0.39)^T$.

Problem 4.2. Example 2 in Ref. 16.
Starting point $x_{\text{start}} = (3.0, 1.0)^T$.

Problem 4.3. Example 1 in Ref. 16.
Starting point $x_{\text{start}} = (1.0, 1.0, 1.0)^T$.

Problem 4.4. Example 6 in Ref. 16.
Starting point $x_{\text{start}} = (2.0, 2.0, 7.0, 0.0, -2.0, 1.0)^T$.

Problem 4.5. Rosenbrock's function.
$f_1(x) = 10.0\ (x_2 - x_1^2)$, $f_2(x) = 1.0 - x_1$.
Starting point $x_{\text{start}} = (-1.2, 1.0)^T$.

Problem 4.6. Davidon 2, see Ref. 24.
Starting point $x_{\text{start}} = (25.0, 5.0, -5.0, -1.0)^T$.

Problem 4.7. Constrained distance problem.
$f_i(x) = \sqrt{((y_i - x_1)^2 + (z_i - x_2)^2)}$, $1 \leq i \leq 64$,
$y_i = 2.0 \sin(\pi i/32.0)$, $z_i = 2.0 \cos(\pi i/32.0)$.
The variables are constrained so that $c_1(x) = (x_1 + 3.0)^2 + x_2^2 = 0$.
Starting point $x_{\text{start}} = (1.0, 1.0)^T$.

Problem 4.8. Problem 4.3 with a constraint.
The objective function in Problem 4.3 is constrained so that $c_1(x) = x_1^2 + x_2^2 + x_3^2 - 1.0 = 0$.
Starting point $x_{\text{start}} = (1.0, 1.0, 1.0)^T$.

Problem 4.9. Problem 4.6 with constraints.
The objective function in Problem 4.6 is constrained so that
$c^1(x) = x_1^2 + 2.0(x_2 - 0.25e^{4.0})^2 + x_3^2 + (x_4 - \cot(4.0))^2 = 0$,
$c_2(x) = x_1 + x_3 = 0$.
Starting point $x_{\text{start}} = (25.0, 0.5, -5.0, -1.0)^T$.

The results are summarized in Tables 1 and 2. In Table 2, n, m, c give the numbers of variables, functions, and constraints (if any) for the problem; a gives the number of functions in the set $\mathcal{I}_0(x^*, 0)$; and k is the number of iterations required by the algorithm.

The number of iterations required by the method is equal to the number of times the derivatives are evaluated. Our implementation uses a rather crude backtracking line search. Consequently the number of function evaluations made is sometimes considerably more than the number of iterations.

Table 1. Solutions obtained by the algorithm.

Problem	Minimum value obtained	Minimizer obtained
4.1	3.87680×10^{-2}	$(0.19337, 0.19377, 0.10893, 0.13973)^T$
4.2	1.0	$(1.1 \times 10^{-11}, 2.2362 \times 10^{-4})^T$
4.3	7.89423	$(0.53596, 0.0, 0.03192)^T$
4.4	5.59813×10^{-1}	$(2.2407, 1.8577, 6.7700, -1.6449, 0.1659, 0.7423)^T$
4.5	6.7×10^{-18}	$(1.0, 1.0)^T$
4.6	903.23433	$(-10.224, 11.908, -0.4581, 0.5803)^T$
4.7	162.94190	$(-2.0, -2 \times 10^{-12})^T$
4.8	8.95605	$(0.98923, -0.9786, 0.10873)^T$
4.9	4836.29362	$(-2.1 \times 10^{-3}, 13.644, 2.1 \times 10^{-3}, 0.864)^T$

As the solution is approached, the Newton stepsize $\alpha^{(k)} = 1.0$ is normally taken, and one function evaluation per iteration is required. For test problems 4.5 and 4.6, it was not necessary to reduce μ below 10^{-1} for the reasons described in Section 3.7. In all cases, the asymptotic convergence rate for a fixed value of μ was observed to be quadratic. Typically, three or four iterations were required to move from $x(\mu)$ to $x(\bar{\mu})$ when $\bar{\mu} = \mu \times 10^{-2}$. We believe that the results are promising; however, we feel that a more sophisticated line search would considerably enhance the behavior of the algorithm.

5. Minimax Problems

The final class of problems that we shall consider is the minimization of the largest of a finite set of functions,

$$\text{(MINIMAX)} \quad \underset{x \in \mathcal{R}^n}{\text{minimize}}\ F_m(x), \text{ where } F_m(x) = \underset{1 \leq i \leq m}{\text{maximum}}\ f_i(x).$$

Table 2. Performance of the algorithm on test problems.

Problem	n	m	c	a	k
4.1	4	11	—	4	45
4.2	2	3	—	2	36
4.3	3	6	—	1	18
4.4	6	51	—	6	66
4.5	2	2	—	2	41
4.6	4	20	—	0	1
4.7	2	64	1	1	18
4.8	3	6	1	3	20
4.9	4	20	2	1	42

Such problems are normally known as minimax problems; an important special case is the discrete Chebyshev approximation of data. Again, we shall be concerned particularly with minimax problems for which some or all of the functions $f_i(x)$ are nonlinear. The linear case has been studied extensively, and many good algorithms for solving such problems are available (see, for a survey, Ref. 25). There are relatively few methods for the nonlinear case (a good survey is given in Ref. 26), and we start by reviewing some of them.

5.1. Existing Algorithms. The development of algorithms for solving MINIMAX parallels that for solving L_1P. A number of algorithms (Refs. 24, 27, 28) are based upon two-phase strategies which start by solving sequences of linearized versions of the nonlinear problem. This continues until those $f_i(x)$ which are considered to be equal to $\max_{1\leq i\leq m} f_i(x)$ at the solution are identified after which a suitable sequence of nonlinear equations are solved by quasi-Newton methods. The method of Conn (Ref. 29) solves MINIMAX directly taking into account the nondifferentiable nature of $F_m(x)$.

A second approach to the problem is to observe that MINIMAX is equivalent to the following nonlinear programming problem

$$\text{(NLPMM)} \quad \underset{x\in\mathcal{R}^n,\, u\in\mathcal{R}}{\text{minimize}}\ u,$$

$$\text{subject to } u - f_i(x) \geq 0, \qquad i = 1, \ldots, m.$$

Han (Ref. 30) proposes that NLPMM be solved by applying his nonlinear programming method (Ref. 31) to NLPMM. The methods of Murray and Overton (Ref. 26) and Han (Ref. 32) apply projected Lagrangian methods to NLPMM and exploit the structure of NLPMM to solve MINIMAX. Both methods impose $F_m(x)$ as a merit function for the minimization, but differ in the way constraints are used in the projected Lagrangian formulation.

5.2. New Method. We tackle MINIMAX by way of NLPMM. An obvious way of solving NLPMM is by means of a sequential minimization of the quadratic penalty function

$$\bar{p}(x, u, \mu) = u + (1/2\mu) \sum_{i=1}^{m} (u - f_i(x))_-^2.$$

Such an approach can easily take special account of the auxiliary variables u. However, we prefer to remove u directly from $\bar{p}(x, u, \mu)$ and to construct a simpler penalty function $p(x, \mu)$.

We observe that, if we fix x and μ, $\bar{p}(x, u, \mu)$ is a piecewise quadratic function of the single variable u. Thus, if x and μ are given, it is easy to find u to minimize $\bar{p}(x, u, \mu)$. We therefore define

$$p(x, \mu) = \min_{u \in \mathcal{R}} \bar{p}(x, u, \mu), \tag{27}$$

and observe that the minimizer of $p(x, \mu)$ over x is the same as the minimizer of $\bar{p}(x, u, \mu)$ over x and u. For given x, suppose the $f_i(x)$ have been ordered so that

$$f_{k_1} \geqslant f_{k_2}(x) \geqslant \cdots \geqslant f_{k_m}(x). \tag{28}$$

Define

$$\Theta(u) \equiv \bar{p}(x, u, \mu), \qquad \text{for fixed } x, \mu,$$

and let

$$\bar{u}(x, \mu) = \arg \min_u \Theta(u).$$

It is clear that $\Theta(u)$ is strictly convex in the region $u \leqslant f_{k_j}(x)$ and the minimizer $\bar{u}(x, \mu)$ lies in this region.

For convenience, introduce

$$f_{k_{m+1}}(x) = -\infty.$$

Let the region Ω_j be

$$\Omega_j = \{u: f_{k_{j+1}}(x) < u \leqslant f_{k_j}(x)\}, \qquad j = 1, \ldots, m.$$

In the region Ω_j,

$$\Theta(u) = \Theta_j(u) = u + (1/2\mu) \sum_{i=1}^{j} (u - f_{k_i}(x))^2.$$

Let

$$\bar{u}_j = \arg \min \Theta_j(u).$$

It is then easy to show that

$$\bar{u}_j = \left[\sum_{i=1}^{j} f_{k_i}(x) - \mu \right] \Big/ j$$

and that $\bar{u}_j$ lies in Ω, if and only if

$$f_{k_{j+1}}(x) < \left[\sum_{i=1}^{j} f_{k_i}(x) - \mu \right] \Big/ j \leqslant f_{k_j}(x).$$

As $\Theta(u)$ is stictly convex when $u \leqslant f_1(x)$,

$$\bar{u}(x, \mu) = \left[\sum_{i=1}^{j} f_{k_i}(x) - \mu \right] \Big/ j, \tag{29}$$

where $j \equiv j(x, \mu)$ is the smallest integer such that

$$f_{k_{j+1}}(x) < \left[\sum_{i=1}^{j} f_{k_i}(x) - \mu \right] \Big/ j. \tag{30}$$

In this case, (27) and (29) combine to give

$$p(x, \mu) = (1/j) \sum_{i=1}^{j} f_{k_i}(x) + \left[\sum_{i=1}^{j} f_{k_i}(x)^2 - (1/j)\left(\sum_{i=1}^{j} f_{k_i}(x) \right)^2 \right] \Big/ 2\mu - \mu/2j. \tag{31}$$

It is clear that $p(x, \mu)$ is one times, and piecewise twice, continuously differentiable. Our proposal is then that MINIMAX be solved by a sequential minimization of $p(x, \mu)$ as $\mu \to 0_+$.

5.3. Derivatives of $p(x, \mu)$. We obtain the following derivatives from (31):

$$\nabla_x p(x, \mu) = (1/j) A_j(x)^T e + \mu^{-1} A_j(x)^T W_j f(x), \tag{32}$$

$$\nabla_{xx} p(x, \mu) = \bar{G}(x, \mu) + \mu^{-1} A_j(x)^T W_j A_j(x), \tag{33}$$

where $A_j(x)$ is made up of the rows $\nabla_x f_{k_i}(x)^T$, $1 \leqslant i \leqslant j$; $f(x)$ is made up of entries $f_{k_i}(x)$, $1 \leqslant i \leqslant j$; W_j is the projection matrix of rank $j-1$,

$$W_j = I_j - ee^T/j;$$

and

$$\bar{G}(x, \mu) = (1/j) \sum_{i=1}^{j} \nabla_{xx} f_{k_i}(x) + \mu^{-1} \sum_{i=1}^{j} w_i(x) \nabla_{xx} f_{k_i}(x),$$

where $w_i(x)$ is the ith element of $W_j f(x)$, i.e.,

$$w_i(x) = f_{k_i}(x) - \sum_{k=1}^{j} f_{k_k}(x)/j. \tag{34}$$

In order to investigate these derivatives further, we need the following lemma.

Lemma 5.1. The quantities $w_i(x)$ are bounded by

$$-1/j \leq w_i(x)\mu^{-1} \leq 1,$$

for all x, all $\mu > 0$, and all $1 \leq i \leq j$.

Proof. See the Appendix. □

Define scalars $\lambda_i(x, \mu)$ such that

$$\lambda_i(x, \mu) = \begin{cases} 1/j + w_i(x)\mu^{-1}, & \text{if } 1 \leq i \leq j, \\ 0, & \text{otherwise.} \end{cases} \tag{35}$$

It is easy to show that these λ_i are continuous functions of x. With such a definition, (32) and (33) may be rewritten as

$$\nabla_x p(x, \mu) = \sum_{i=1}^{j} \lambda_i(x, \mu) \nabla_x f_{k_i}(x),$$

$$\nabla_{xx} p(x, \mu) = \bar{G}(x, \mu) + \mu^{-1} A_j(x)^T W_j A_j(x),$$

where

$$\bar{G}(x, \mu) = \sum_{i=1}^{j} \lambda_i(x, \mu) \nabla_{xx} f_{k_i}(x).$$

Therefore, both $\nabla_x p(x, \mu)$ and $\bar{G}(x, \mu)$ are continuous [provided the $f_i(x)$ are twice continuously differentiable]. If we sum (35), we obtain

$$\sum_{i=1}^{m} \lambda_i(x, \mu) = 1.$$

Furthermore, Lemma 5.1 gives

$$\lambda_i(x, \mu) \geq 0, \qquad 1 \leq i \leq m.$$

Finally, by definition of $x(\mu)$,

$$\lim_{\mu \to 0_+} \lim_{x \to x(\mu)} \sum_{i=1}^{j} \lambda_i(x, \mu) \nabla_x f_i(x) = 0.$$

Therefore, if

$$\lim_{\mu \to 0_+} \lim_{x \to x(\mu)} \lambda_i(x, \mu) = \lambda_i^*, \qquad 1 \leq i \leq m, \tag{36}$$

the Lagrange multipliers λ_i^* satisfy the first order optimality conditions

$$\begin{aligned} &\text{(i)} \quad \sum_{i=1}^{j} \lambda_i^* = 1, \quad \lambda_i^* \geq 0, \qquad 1 \leq i \leq m, \\ &\text{(ii)} \quad \sum_{i=1}^{j} \lambda_i^* \nabla_x f_{k_i}(x^*) = 0, \end{aligned} \tag{37}$$

for MINIMAX (see, e.g., Ref. 33). Moreover,

$$\lim_{\mu \to 0_+} \lim_{x \to x(\mu)} \bar{G}(x, \mu) = \nabla_{xx} L(x^*, \lambda^*), \tag{38}$$

where

$$L(x, \lambda) = \sum_{i=1}^{m} \lambda_i f_{k_i}(x).$$

Hence, if

$$x^* = \lim_{\mu \to 0} \lim_{x \to x(\mu)} x$$

and (36) is satisfied, (37) indicate that x^* is a first-order stationary point for MINIMAX; once again, this is a stronger statement than is normally possible when penalty function methods are applied to nonlinear programming problems. We note that (36) is satisfied if the gradients $\nabla_x f_{k_i}(x^*)$, $1 \leq i \leq j$, span a space of dimension $j-1$. This assumption is often known as the Haar condition. Moreover, if μ_k is any sequence tending to zero such that

$$\lim_{k \to \infty} \lim_{x \to x(\mu_k)} \lambda(x, \mu_k) = \lambda_i^*, \qquad 1 \leq i \leq m,$$

for some λ_i^*, the same conclusion is true. As $\lambda(x, \mu)$ is bounded (Lemma 5.1), there is at least one such sequence.

5.4. Minimizing $p(x, \mu)$. As with the penalty function for L_1P, we advocate using Newton's method and its variants to minimize $p(x, \mu)$. The usual problem of increasing ill-conditioning of $\nabla_{xx} p(x, \mu)$ would arise if Newton's method were applied directly. This time there is a further complication. From (33),

$$\nabla_{xx} p(x, \mu) = \bar{G}(x, \mu) + \mu^{-1} A_j(x)^T W_j A_j(x),$$

where, as usual, $\bar{G}(x, \mu)$ is well behaved [in the light of (38)]. The additional problem is that $A_j(x)^T W_j A_j(x)$ is of rank at most $j-1$. To simplify matters, let us make the following nondegeneracy assumption:

(NDA3) The gradients $\nabla_x f_{k_i}(x^*)$, $1 \leq i \leq j$, span a space of dimension $j-1$ and the associated unique Lagrange multipliers λ_i^* are nonzero, i.e., $\lambda_i^* > 0$, $1 \leq i \leq j$.

We shall consider the implications of relaxing this condition in Section 7. Under NDA3, if x is sufficiently close to x^*, $A_j(x)^T W_j A_j(x)$ is positive semidefinite and of rank $j-1$. Let $\bar{A}_j(x)$ be any $(j-1) \times n$ matrix such that

$$\bar{A}_j(x)^T \bar{A}_j(x) = A_j(x)^T W_j A_j(x). \tag{39}$$

Then, the Newton equations, giving a correction $p^{(k)}$ to $x^{(k)}$,

$$\nabla_{xx} p(x^{(k)}, \mu) p^{(k)} = -\nabla_x p(x^{(k)}, \mu),$$

can be written as either

$$\begin{bmatrix} \bar{G}(x^{(k)}, \mu) & \bar{A}_j(x^{(k)})^T \\ \bar{A}_j(x^{(k)}) & -\mu I_{j-1} \end{bmatrix} \begin{bmatrix} p^{(k)} \\ r^{(k)} \end{bmatrix} = -\begin{bmatrix} A_j(x^{(k)})^T e/j \\ \bar{f}(x^{(k)}) \end{bmatrix}, \tag{40}$$

or

$$\begin{bmatrix} \bar{G}(x^{(k)}, \mu) & \bar{A}_j(x^{(k)})^T \\ \bar{A}_j(x^{(k)}) & -\mu I_{j-1} \end{bmatrix} \begin{bmatrix} p^{(k)} \\ s^{(k)} \end{bmatrix}$$

$$= -\begin{bmatrix} \sum_{i=1}^{j} \lambda_i(x^{(k)}, \mu) \nabla_x f_{k_i}(x^{(k)}) \end{bmatrix}, \tag{41}$$

where $\bar{f}(x^{(k)})$ is any vector satisfying

$$\bar{A}_j(x^{(k)})^T \bar{f}(x^{(k)}) = A_j(x^{(k)})^T W_j f(x^{(k)}). \tag{42}$$

Under NDA3, the normal arguments indicate that, if MINIMAX is well conditioned, either (40) or (41) may be used to determine $p^{(k)}$ accurately. In order to find $\bar{A}_j(x)$ satisfying (39), we propose that Cholesky's method with interchanges (symmetric LU decomposition with interchanges) be used to find

$$\bar{A}_j(x)^T = P \begin{bmatrix} L \\ l^T \end{bmatrix}, \tag{43}$$

where P is a permutation matrix, L is $(j-1)\times(j-1)$ lower triangular, and l is a $j-1$ vector (see, e.g., Ref. 23). A suitable vector $\bar{f}(x^{(k)})$ may then be found by solving

$$L\bar{f}(x^{(k)}) = \hat{f}(x^{(k)}),$$

where $\hat{f}(x^{(k)})$ comprises the first $j-1$ entries of $P^T A_j(x^{(k)})^T W_j f(x^{(k)})$. However, in view of the extra work involved in finding $\bar{f}(x^{(k)})$, (41) may be preferred to (40).

In order to determine when either (40) or (41) is appropriate, we have the following theorem.

Theorem 5.1. Let K be the coefficient matrix of (41), and let $In(K) = (k_+, k_-, k_0)$. Then:

(i) $\nabla_{xx} p(x^{(k)}, \mu)$ is positive definite if and only if $k_0 = 0$ and $k_- = j-1$;

(ii) $\nabla_{xx} p(x^{(k)}, \mu)$ is positive semidefinite but singular if and only if $k_0 > 0$ and $k_- = j-1$;

(iii) $\nabla_{xx} p(x^{(k)}, \mu)$ is indefinite if and only if $k_- \geq j$.

Proof. The proof is identical to Theorem 3.2 and its corollaries in Ref. 1. □

We should only calculate $p^{(k)}$ from (40) or (41) when (i) is satisfied. In cases (ii) or (iii), weak solutions to (40) or (41) or directions of infinite descent can be calculated in the manner described in Ref. 1.

The storage and iteration costs for our method are essentially the same as those described in Section 3.5 for the algorithm for L_1P. The cost per iteration is inceased slightly by about $2n^2 \min(n, j)/3$ multiplications, because of the need to assemble $A_j^T W_j A_j$ and find $\bar{A}_j$. These figures are roughly the same as for good existing algorithms.

5.5. Practicalities. The problems associated with a straightforward sequential minimization of $p(x, \mu)$ described in Section 3.5 for L_1P (the introduction of second derivative discontinuities between the starting value and the minimizer of the penalty function when μ is reduced) equally apply to MINIMAX. Our remedy is essentially the same.

Let

$$\mathscr{I}_0(x, \mu) = \{k_1, \ldots, k_j\},$$

where j is defined by (30). Suppose that $\mathscr{I}_1$ is a given subset of $\{1, \ldots, m\}$ of cardinality j_1. Let

$$\mathscr{I}_2 = \{1, \ldots, m\} \backslash \mathscr{I}_1$$

have cardinality j_2; and suppose that for a given x, the elements of $\mathscr{I}_2$ are written as $\{k'_1, \ldots, k'_{j_2}\}$, where

$$f_{k'_{i+1}}(x) \leqslant f_{k'_i}(x).$$

Further, let $j' \equiv j'(x, \mu, \mathscr{I}_1)$ be the smallest integer such that

$$f_{k'_{j'+1}} < \left[\sum_{i \in \mathscr{I}_1} f_i(x) + \sum_{i=0}^{j'} f_{k'_i}(x) - \mu \right] / (j_1 + j'),$$

if $j_1 > 0$, or $j' = j$, if $j_1 = 0$. Finally, let

$$\mathscr{I}_0(x, \mu, \mathscr{I}_1) = \mathscr{I}_1 \cup \{k'_1, \ldots, k'_{j'}\}$$

have cardinality j_0. Then we define the auxiliary function

$$p(x, \mu, \mathscr{I}_1) = \sum_{i \in \mathscr{I}_0(x, \mu, \mathscr{I}_1)} f_i(x)/j_0 - \mu/2j_0 + \left(\sum_{i \in \mathscr{I}_0(x, \mu, \mathscr{I}_1)} f_i(x)^2 - (1/j_0) \left[\sum_{i \in \mathscr{I}_0(x, \mu, \mathscr{I}_1)} f_i(x) \right]^2 \right) / 2\mu. \tag{44}$$

Notice that $p(x, \mu, \varnothing) = p(x, \mu)$. The important property of $p(x, \mu, \mathcal{I}_1)$ is that, as $\mathcal{I}_1 \subseteq \mathcal{I}_0(x, \mu, \mathcal{I}_1)$, there can be no second derivative discontinuities of $p(x, \mu, \mathcal{I}_1)$, due to the functions $f_i(x)$, $i \in \mathcal{I}_1$. We would normally expect that the set $\mathcal{I}_0(x, \mu)$ will remain fixed for all μ sufficiently small (at least for nondegenerate problems). Therefore, if we fix $\mathcal{I}_1 = \mathcal{I}_0(x(\bar{\mu}), \bar{\mu})$ for some sufficiently small $\bar{\mu}$, we would expect that $p(x, \mu) = p(x, \mu, \mathcal{I}_1)$, for all x in a neighborhood of $x(\mu)$ and all $\mu \leq \bar{\mu}$. Consequently, the minimizer $x(\mu, \mathcal{I}_1)$ of $p(x, \mu, \mathcal{I}_1)$ and $x(\mu)$ should be identical. Furthermore, $p(x, \mu, \mathcal{I}_1)$ should be twice continuously differentiable in a neighborhood including both $x(\mu)$ and $x(\bar{\mu})$ if we have correctly identified $\mathcal{I}_1$. This motivates the following algorithm:

Set $\mu := \mu_{\text{start}}$, $x := x_{\text{start}}$, $\mathcal{I}_1 = \varnothing$.
Until $\mu < \mu_{\min}$, do these operations:
Minimize $p(x, \mu, \mathcal{I}_1)$ starting at x_0 to find $x(\mu, \mathcal{I}_1)$ and $\mathcal{I}_0(x(\mu, \mathcal{I}_1), \mu)$.
Let $x_0 := x(\mu, \mathcal{I}_1)$.
If $\mathcal{I}_1 \neq \mathcal{I}_0(x(\mu, \mathcal{I}_1), \mu)$, minimize $p(x, \mu)$ starting at x_0 to find $x(\mu)$ (*); let $x_0 := x(\mu)$.
Let $\mathcal{I}_1 := \mathcal{I}_0(x_0, \mu)$, $u := \mu \times 10^{-2}$.

The part of the algorithm marked (*) is to allow for an incorrect identification of $\mathcal{I}_0(x(\mu), \mu)$. The minimization of $p(x, \mu, \mathcal{I}_1)$ may be accomplished in essentially the same manner described in Section 5.4. The summation indices in (44) are different from those in (31); these differences carry over into the derivatives (32) and (33) in the obvious way.

5.6. Comments. A commonly occurring minimax problem is to minimize the infinity-norm of a vector valued function,

$$(L_\infty P) \quad \underset{x \in \mathcal{R}^n}{\text{minimize}} \ \underset{1 \leq i \leq m}{\text{maximum}} \ |f_i(x)|.$$

Clearly this may be written as follows:

$$(L_\infty P) \quad \underset{x \in \mathcal{R}^n}{\text{minimize}} \ \underset{1 \leq i \leq m}{\text{maximum}} \ \{f_i(x), -f_i(x)\}$$

and solved by the methods of the preceding sections. However, it is also possible to derive a special penalty function specifically for this problem which takes account of its structure.

Suppose, for fixed x and μ, that the functions $f_i(x)$ are ordered so that

$$|f_{k_{i+1}}(x)| \leq |f_{k_i}(x)|.$$

Define

$$l_j = \begin{cases} |f_{k_{j+1}}(x)|, & \text{if } 1 \leq j \leq m-1, \\ -|f_{k_{2m-j}}(x)|, & \text{if } m \leq j \leq 2m-1, \\ -\infty, & \text{if } j = 2m, \end{cases}$$

$$u_j = \begin{cases} \left(\sum_{i=1}^{j} |f_{k_i}(x)| - \mu \right) \Big/ j, & \text{if } 1 \le j \le m-1, \\ \left(\sum_{i=1}^{2m-j} |f_{k_i}(x)| - \mu \right) \Big/ j, & \text{if } m \le j \le 2m. \end{cases}$$

Now, pick

$$j \equiv j(x, \mu), \qquad 1 \le j \le 2m,$$

such that j is the smallest integer for which $l_j < u_j$. Then, the appropriate version of the quadratic penalty function for $L_\infty P$ is

$$p(x, \mu) = \begin{cases} \sum_{i=1}^{j} |f_{k_i}(x)|/j - \mu/2j \\ + \left[\sum_{i=1}^{j} f_{k_i}(x)^2 - 1/j \left[\sum_{i=1}^{j} |f_{k_i}(x)| \right]^2 \right] / 2\mu, \\ \qquad \text{if } 1 \le j \le m, \\ \sum_{i=1}^{2m-j} |f_{k_i}(x)|/j - \mu/2j + \sum_{i=1}^{m} f_{k_i}(x)^2/\mu \\ - \left[\sum_{i=1}^{2m-j} f_{k_i}(x)^2 + \left[\sum_{i=1}^{2m-j} |f_{k_i}(x)| \right]^2 \Big/ j \right] \Big/ 2\mu, \\ \qquad \text{if } m+1 \le j \le 2m. \end{cases}$$

This function may be derived using the techniques of Section 5.2. Notice that this function is differentiable; for, although there are terms involving $|f_{k_i}(x)|$, the choice of j excludes any $f_i(x)$ for which $f_i(x) = 0$.

Once again, constrained MINIMAX problems present no difficulties; the constraints are introduced into the penalty function in an obvious way. However, in contrast to $L_1 P$, we have been unable to derive differentiable functions for MINIMAX based upon barrier functions which exploit the structure of NLMMP.

6. Testing

An implementation of the algorithm described in Section 5 has enabled us to conduct preliminary numerical experiments. The problem data given

Table 3. Solutions obtained by the algorithm.

Problem	Minimum value obtained	Minimizer obtained
4.1	8.08444×10^{-3}	$(0.18463, 0.10521, 0.01197, 0.11179)^T$
4.2	6.16432×10^{-1}	$(0.45330, -0.94659)^T$
4.3	3.59971	$(0.32826, 0.0, 0.13132)^T$
4.4	3.49049×10^{-2}	$(1.1759, 1.8993, 6.9482, -1{,}6503, 0.1457, 0.5170)^T$
4.5	0.0	$(1.0, 1.0)^T$
4.6	115.70643	$(-12.244, 14.022, -0.451, -0.011)^T$
4.7	3.99999	$(1.20, 1.5 \times 10^{-13})^T$
4.8	4.16140	$(0.97778, 5.3 \times 10^{-22}, 0.20965)^T$
4.9	485.39904	$(-3.912 \times 10^{-3}, 13.639, 3.912 \times 10^{-3}, 0.8634)^T$

in Section 4 has been used to construct test examples of the form

$$\underset{x \in \mathscr{R}^n}{\text{minimize}} \ \underset{1 \le i \le m}{\text{maximum}} \ |f_i(m)|,$$

subject to the given constraints, if any. The results of our testing are given in Tables 3 and 4. The starting values and values of μ_{start} and $\mu_{\min}$ are those given in Section 4.

The comments, with regards to how these results should have been interpreted, given in Section 4, are equally relevant here. Once again, we feel that these results are encouraging and indicate that algorithm is a useful technique for solving minimax problems. In Table 3, n, m, c, k are as described in Section 4; a gives the number of functions thought to be equal to $\max|f_i|$ at the solution.

Table 4. Performance of the algorithm on test problems.

Problem	n	m	c	a	k
4.1	4	11	—	5	43
4.2	2	3	—	2	16
4.3	3	6	—	2	16
4.4	6	51	—	7	69
4.5	2	2	—	2	26
4.6	4	20	—	3	26
4.7	2	64	1	1	24
4.8	3	6	1	2	24
4.9	4	20	2	1	37

7. Degenerate Problems

We shall discuss degeneracy with respect to the problem NLP. Any comments that we make will apply to all of the problems that we have considered in this paper. We say that a nonlinear program is *degenerate* if the local solution that we encounter does not satisfy the nondegeneracy assumption NDA.

Degeneracy has two effects on solving NLP by way of the quadratic penalty function. Firstly, the matrix (5) is singular if the constraint gradients $\nabla_x c_i(x^*)$, $i \in \mathscr{A}$, are linearly dependent. This implies that (2) may be badly conditioned for small μ and the corresponding numerical solution prone to errors. Secondly, the presence of zero Lagrange multipliers indicates that the set $\mathscr{V}(x)$ of violated constraints may not settle down as μ approaches zero. Consequently, the second derivative matrix $\nabla_{xx} p(x, \mu)$ may be discontinuous in any neighborhood of $x(\mu)$ and of x^*. This can impede the convergence rate of the method, even for fixed value of μ.

The first problem may often be avoided in a satisfactory fashion. The second defect is more problematical. In order to proceed, we need to isolate the most frequently occurring cause of degeneracy.

Definition 7.1. We say that the local solution x^* of NLP is *strongly degenerate* if the constraint gradients $\nabla_x c_i(x)$, $i \in \mathscr{A}$, are linearly dependent for all x in some open neighborhood of x^*.

Strong degeneracy usually arises when $|\mathscr{A}| > n$. This frequently happens in l_1-problems and minimax problems. It manifests itself, when using the quadratic penalty function, in the second term of the second derivative matrix

$$\nabla_{xx} \bar{p}(x^{(k)}) = \bar{G}(x^{(k)}, \mu) + \mu^{-1} A(x^{(k)})^T A(x^{(k)}).$$

If $A(x^{(k)})$ is rank deficient and μ is small, (2) is ill-conditioned. This may be avoided using the technique already mentioned in Section 5.4. That is, find a matrix $\bar{A}$ such that

$$\bar{A}^T \bar{A} = A(x^{(k)})^T A(x^{(k)}),$$

and so that $\bar{A}$ is full rank. This may be accomplished using a symmetric LU decomposition with interchanges as indicated in (43). Then, (2) can be replaced by

$$\begin{bmatrix} \bar{G}(x^{(k)}, \mu) & \bar{A}^T \\ \bar{A} & -\mu I \end{bmatrix} \begin{bmatrix} p^{(k)} \\ \bar{r}^{(k)} \end{bmatrix} = -\begin{bmatrix} g^{(k)} \\ \bar{c} \end{bmatrix}, \tag{45}$$

or by

$$\begin{bmatrix} \bar{G}(x^{(k)}, \mu) & \bar{A}^T \\ \bar{A} & -\mu I \end{bmatrix} \begin{bmatrix} p^{(k)} \\ \bar{s}^{(k)} \end{bmatrix} = -\begin{bmatrix} \bar{g} \\ 0 \end{bmatrix}, \tag{46}$$

where

$$\bar{g} = g^{(k)} - \sum_{i \in \mathcal{V}(x^{(k)})} \lambda_i(x^{(k)}, \mu) \nabla_x c_i(x^{(k)})$$

and $\bar{c}$ is any vector satisfying

$$\bar{A}^T \bar{c} = A(x^{(k)})^T c(x^{(k)}).$$

The formulation (46) may be preferred, as it avoids the need to find $\bar{c}$. We do not necessarily limit the use of (45) or (46) to cases when μ is small. These approaches have the additional benefit that (45) or (46) have dimension at most $2n$. The formulation (2) may have as many as $m+n$ equations. For data fitting l_1-problems and minimax problems, where m is frequently much larger than n, this may be important.

It is not clear as to the best way of dealing with the second derivative discontinuities in $p(x, \mu)$ which arise because of the degeneracy. One possibility is to use an active set strategy to predict which constraints are active (i.e., have the value zero) at x^*. Suppose that the constraints indexed by $\bar{\mathcal{A}}$ are predicted to be active at x^*. Clearly, $\mathcal{E} \subseteq \bar{\mathcal{A}}$. Then, we might try to solve NLP by a sequential minimization of the auxiliary penalty function

$$\bar{p}(x, \mu, \bar{\mathcal{A}}) = f(x) + (1/2\mu) \sum_{i \in \bar{\mathcal{A}}} c_i(x)^2 + (1/2\mu) \sum_{i \in \mathcal{F} \setminus \bar{\mathcal{A}}} c_i(x)_-^2.$$

Using the Lagrange multiplier estimates generated in the minimization, it may be possible to correct for incorrect choices of $\bar{\mathcal{A}}$. However, such Lagrange multiplier estimates are not unique, and the difficulties associated with verifying optimality normally associated with degeneracy under these circumstances will now apply (see Ref. 7). We have not pursued this further.

A second way of dealing with such discontinuities is to avoid them altogether by using a different penalty function. For instance, the logarithmic barrier function for inequality constraints is twice continuously differentiable. The difficulty is, of course, the need to find and maintain feasible points for the nonlinear program. This is trivial for L_1P and MINIMAX, and it may turn out that barrier functions are preferable for these problems. We are currently pursuing this idea.

8. Appendix

Proof of Lemma 5.1. For simplicity and without losing generality, assume $k_i = i$, $1 \le i \le j$. We first establish the lower bound. From (28) and (30),

$$(1/j)\left[\sum_{k=1}^{j} f_k(x) - \mu\right] \le f_j(x) \le f_{j-1}(x) \le \cdots \le f_1(x).$$

Hence, for any $1 \leq i \leq j$,

$$\sum_{k=1}^{j} f_k(x) - \mu \leq j f_i(x).$$

Rearranging, we obtain

$$\left[j f_i(x) - \sum_{k=1}^{j} f_k(x) \right] \mu^{-1} \geq -1.$$

Combining with (34), we obtain

$$w_i(x) \mu^{-1} \geq -1/j,$$

as required.

To establish the other part of the inequality, observe that the minimality of j in (30) implies that

$$f_1(x) \geq \cdots \geq f_i(x) \geq f_{i+1}(x) \geq \left[\sum_{k=1}^{i} f_k(x) - \mu \right] \Big/ i, \qquad 1 \leq i \leq j-1. \tag{47}$$

Assume inductively that

$$f_1(x) - \mu \leq f_k(x), \qquad k = 1, \ldots, i \leq j.$$

This is trivially true for $k = 1, 2$. Then, from (47) and the inductive hypothesis,

$$f_{i+1}(x) \geq \left(\sum_{k=1}^{i} f_k(x) - \mu \right) \Big/ i = \left(f_1(x) + \sum_{k=2}^{i} f_k(x) - \mu \right) \Big/ i$$

$$\geq \left(f_1(x) + \sum_{k=2}^{i} (f_1(x) - \mu) - \mu \right) \Big/ i = f_1(x) - \mu.$$

Thus,

$$f_1(x) - \mu \leq f_i(x), \qquad 1 \leq i \leq j.$$

Summing these inequalities, we obtain

$$j f_1(x) - j\mu \leq \sum_{k=1}^{j} f_k(x),$$

i.e.,

$$f_1(x) - \sum_{i=1}^{j} f_i(x)/j \leq \mu. \tag{48}$$

But as

$$f_j(x) \leq f_{j-1}(x) \leq \cdots \leq f_1(x),$$

(48) gives

$$f_i(x) - \sum_{i=1}^{j} f_i(x)/j \leq \mu, \qquad 1 \leq i \leq j,$$

which establishes that

$$\mu^{-1} w_i(x) \leq 1, \qquad 1 \leq i \leq j. \qquad \square$$

References

1. GOULD, N. I. M., *On the Accurate Determination of Search Directions for Simple Differentiable Penalty Functions*, IMA Journal of Numerical Analysis, Vol. 6, pp. 357–372, 1986.
2. FIACCO, A. V., and MCCORMICK, G. P., *Nonlinear Programming: Sequential Unconstrained Minimization Technique*, John Wiley and Sons, New York, New York, 1968.
3. MURRAY, W., *Analytical Expressions for Eigenvalues and Eigenvectors of the Hessian Matrices of Barrier and Penalty Functions*, Journal of Optimization Theory and Applications, Vol. 7, pp. 189–196, 1971.
4. LOOTSMA, F. A., *Hessian Matrices of Penalty Functions for Solving Constrained Optimization Problems*, Phillips Research Reports, Vol. 24, pp. 322–331, 1969.
5. GERENSCHER, L., *A Second-Order Technique for the Solution of Nonlinear Optimization Problems*, Colloquia Mathematica Societis Janos Bolya 12, Progress in Operations Research, Eger, Hungary, 1976.
6. BROYDEN, C. G., and ATTIA, N. F., *A Smooth Sequential Penalty Function Method for Solving Nonlinear Programming Problems*, Paper Presented at the IFIP Conference, Copenhagen, Denmark, 1983.
7. GILL, P. E., MURRAY, W., and WRIGHT, M. H., *Practical Optimization*, Academic Press, London, England, 1981.
8. ARMIJO, L., *Minimization of Functions Having Continuous Partial Derivatives*, Pacific Journal of Mathematics, Vol. 16, pp. 1–3, 1966.
9. BUNCH, J. R., and PARLETT, B. N., *Direct Methods for Solving Symmetric Indefinite Systems of Linear Equations*, SIAM Journal on Numerical Analysis, Vol. 8, pp. 639–655, 1971.
10. FLETCHER, R., *Factorizing Symmetric Indefinite Matrices*, Linear Algebra and Its Applications, Vol. 14, pp. 257–272, 1976.
11. BUNCH, J. R., and KAUFMAN, L. C., *Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Equations*, Mathematics of Computation, Vol. 31, pp. 163–179, 1979.

12. BROYDEN, C. G., *Quasi-Newton Methods and Their Application to Function Minimization*, Mathematics of Computation, Vol. 21, pp. 368–381, 1967.
13. MURRAY, W., and OVERTON, M. L., *A Projected Lagrangian Algorithm for Nonlinear l_1-Optimization*, SIAM Journal on Scientific and Statistical Computing, Vol. 2, pp. 207–224, 1981.
14. OSBORNE, M. R., and WATSON, G. A., *On an Algorithm for Discrete Nonlinear l_1-Approximation*, Computer Journal, Vol. 14, pp. 184–188, 1971.
15. MCLEAN, R. A., and WATSON, G. A., *Numerical Methods for Discrete l_1-Approximation*, Numerical Methods of Approximation Theory, Vol. 5, pp. 169–180, 1980.
16. HALD, J., and MADSEN, K., *Combined LP and Quasi-Newton Methods for Nonlinear l_1-Optimization*, SIAM Journal on Numerical Analysis, Vol. 22, pp. 60–80, 1985.
17. BARTELS, R. H., and CONN, A. R., *An Approach to Nonlinear l_1-Data Fitting*, Numerical Analysis Proceedings, Cocoyoc, Mexico, 1981; Edited by A. Dold and B. Eckmann, Springer-Verlag, New York, New York, pp. 48–58, 1982.
18. EL-ATTAR, R. A., VIDYASAGAR, M., and DUTTA, S. R. K., *An Algorithm for l_1-Norm Minimization with Application to Nonlinear l_1-Approximation*, SIAM Journal on Numerical Analysis, Vol. 16, pp. 70–86, 1979.
19. CONN, A. R., *Nonlinear Programming, Exact Penalty Functions, and Projection Techniques for Nonsmooth Functions*, Numerical Optimization 1984, Edited by P. T. Boggs, R. H. Byrd, and R. B. Schnabel, SIAM, Philadelphia, Pennsylvania, pp. 3–25, 1985.
20. MARATOS, N., *Exact Penalty Function Algorithms for Finite-Dimensional and Control Optimization Problems*, University of London, PhD Thesis, 1978.
21. HUBER, P. J., *Robust Statistics*, John Wiley and Sons, New York, New York, 1981.
22. RYAN, D. M., *Penalty and Barrier Functions*, Numerical Methods for Constrained Optimization, Edited by P. E. Gill and W. Murray, Academic Press, London, England, 1974.
23. GOLUB, G. H., and VAN LOAN, C. F., *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 1983.
24. WATSON, G. A., *The Minimax Solution of an Overdetermined Set of Nonlinear Equations*, Journal of the Institute of Mathematics and Its Applications, Vol. 23, pp 167–180, 1979.
25. BRANNIGAN, M., *Discrete Chebyshev Approximation with Linear Constraints*, SIAM Journal on Numerical Analysis, Vol. 22, pp. 1–15, 1985.
26. MURRAY, W., and OVERTON, M. L., *A Projected Lagrangian Algorithm for Nonlinear Minimax Optimization*, SIAM Journal on Scientific and Statistical Computing, Vol. 3, pp. 345–370, 1980.
27. HETTICH, R., *A Newton Method for Chebyshev Approximation*, Approximation Theory, Edited by A. Dold and B. Eckmann, Bonn, Germany, 1976.
28. HALD, J., and MADSEN, K., *A Two-Stage Algorithm for Minimax Optimization*, Report No. NI-78-11, Institute for Numerical Analysis, Technical University of Denmark, 1978.

29. CONN, A. R., *An Efficient Second-Order Method to Solve the Minimax Problem*, Research Report No. CORR 79-5, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, Canada, 1979.
30. HAN, S. P., *On the Validity of a Nonlinear Programming Method for Solving Minimax Problems*, Report No. 1981, Mathematics Research Center, University of Wisconsin, Madison, Wisconsin, 1978.
31. HAN, S. P., *A Globally Convergent Method for Nonlinear Programming*, Journal of Optimization Theory and Applications, Vol. 22, pp. 297–307, 1977.
32. HAN, S. P., *Variable-Metric Methods for Minimizing a Class of Nondifferentiable Penalty Functions*, Report, Department of Computer Science, Cornell University, 1977.
33. DEM'YANOV, V. F., and MALOZEMOV, V. N., *Introduction to Minimax*, John Wiley and Sons, New York, New York, 1974.