JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 75, No. 3, DECEMBER 1992

New Minimax Algorithm[1]

A. VARDI[2]

Communicated by R. A. Tapia

Abstract. The purpose of this paper is to suggest a new, efficient algorithm for the minmax problem

$$\min_x \max_i f_i(x), \qquad x \in \Re^n, \qquad i = 1, \ldots, m,$$

where f_i, $i = 1, \ldots, m$, are real-valued functions defined on $\Re^n$. The problem is transformed into an equivalent inequality-constrained minimization problem, min t, s.t. $f_i(x) - t \leq 0$, for all i, $i = 1, \ldots, m$. The algorithm has these features: an active-set strategy with three types of constraints; the use of slack variables to handle inequality constraints; and a trust-region strategy taking advantage of the structure of the problem. Following Tapia, this problem is solved by an active set strategy which uses three types of constraints (called here nonactive, semiactive, and active). Active constraints are treated as equality constraints, while semiactive constraints are treated as inequality constraints and are assigned slack variables. This strategy helps to prevent zigzagging. Numerical results are provided.

Key Words. Minmax problems, constrained optimization, active set strategy, trust region, slack variables.

1. Introduction

The purpose of this paper is to suggest a new, efficient algorithm for the minmax problem

$$\min_x \max_i f_i(x), \qquad x \in \Re^n, \qquad i = 1, \ldots, m,$$

---

[1]This paper was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-15810 while the author was in residence at the Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia.

[2]Assistant Professor, Department of Mathematics and Computer Science, Drexel University, Philadelphia, Pennsylvania.

where f_i, $i=1,\dots,m$, are real-valued functions defined on $\Re^n$. The problem is transformed into an equivalent inequality-constrained minimization problem min t, s.t. $f_i(x)-t\leq 0$, for all i, $i=1,\dots,m$. The algorithm has these features: an active-set strategy with three types of constraints; the use of slack variables to handle inequality constraints; and a trust-region strategy taking advantage of the structure of the problem. Extensive testing of a computer program based on this algorithm indicates that this combination is successful.

For the constrained problem, Tapia (Ref. 1) suggested a strategy of using three categories of inequality constraints. Tapia also renewed interest in the use of slack variables for handling inequality constraints. Following Tapia, we use an active set strategy with three types of constraints: nonactive, semiactive, and active. Semiactive constraints are such that, being far from the solution, there is still uncertainty that they are active. In our algorithm, active constraints are treated as equality constraints and semiactive ones are assigned slack variables and are also treated as equality constraints. Ultimately, we solve an equality-constrained minimization problem. This active set strategy with the use of slack variables may prevent zigzagging which often occurs in the implementation of active set algorithms.

Section 2 contains the basic model and the introduction of the new active set strategy. In Section 3, we give a description of the trust-region strategy in constrained minimization. We suggest the use of the trust-region strategy for the minmax problem in Section 4. In Section 5, we discuss the numerical implementation of the algorithm, and in Section 6 we give numerical results of six standard problems from the literature.

2. Basic Model

Consider m real-valued functions $f_1,\dots,f_m$ defined on $\Re^n$. We are interested in solving the problem

$$\text{(P1)}\quad \min_x \max_i f_i(x), \qquad i=1,\dots,m,$$

which is equivalent to the following problem:

$$\text{(P2)}\quad \min_{x,t} \quad t,$$

$$\text{s.t.}\quad f_i(x)-t\leq 0, \qquad i=1,\dots,m.$$

The following notation will be used:

$$F=\begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix}, \qquad \nabla F=\begin{bmatrix} \partial f_1/\partial x_1 & \dots & \partial f_m/\partial x_1 \\ \vdots & & \vdots \\ \partial f_1/\partial x_n & \dots & \partial f_m/\partial x_n \end{bmatrix}=[\nabla f_1,\dots,\nabla f_m].$$

Consider now problem (P2). Let $x^* \in \Re^n$, and assume that $[\nabla F^T(x^*), -e]^T$ is of full rank. Let $t^* \in \Re$. Further, assume that a first-order and a second-order constraint qualification holds at (x^*, t^*). Necessary conditions for a local minimum of (P2) at (x^*, t^*) are: There exists $v^* \in \Re^m$ such that

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} \nabla F(x^*) \\ -e^T \end{bmatrix} v^* = 0,$$

$$v_i^*(f_i(x^*) - t^*) = 0, \qquad \text{all } i,$$

$$v_i^* \geq 0, \qquad \text{all } i,$$

$$f_i(x^*) - t \leq 0, \qquad \text{all } i,$$

and for all $z \in \Re^{n+1}$ satisfying

$$\begin{bmatrix} \nabla f_i(x^*) \\ -1 \end{bmatrix}^T z = 0,$$

for all i such that $f_i(x^*) - t^* = 0$,

$$z^T \begin{bmatrix} B^* & 0 \\ 0 & 0 \end{bmatrix} z \geq 0,$$

where

$$e^T = (1, \ldots, 1), \qquad B^* = \sum_{i=1}^{m} v_i^* \nabla^2 f_i(x^*).$$

Assume now that, at the solution (x^*, t^*, v^*),

$$v_i^* > 0 \quad \text{or} \quad f_i(x^*) - t^* < 0, \qquad \text{all } i.$$

This strict complementarity assumption is both standard and mild. The constraints which correspond to the positive components of $v^*(f_i(x^*) = t^*)$ are referred to as active constraints. This leads to a natural division of the constraints into active and nonactive ones, and to an active-set strategy.

Being far from the solution, it is difficult to determine which constraints are active at the solution. Tapia (Ref. 1) has an interesting discussion on active-set strategy and concludes that a strategy which allows a certain amount of indecisiveness in the selection of active constraints is needed. Following Tapia, an active-set strategy is used in which the constraints are divided into three sets: set (I) contains active constraints; set (II) contains semiactive constraints; and set (III) contains nonactive constraints. At each iteration, this division is revised. Set (I) will contain no more than $n+1$ constraints at any iteration, since at most $n+1$ can be active at the solution.

Constraints in set (II) are in transition between sets (I) and (III). Set (II) is expected to contain no constraints near the solution.

Constraint i becomes active in the kth iteration if $i\in(\mathrm{II})$ and, for an attempted step Δx,

$$f_i(x^k+\Delta x)>\max f_j(x^k)$$

or if, for the accepted step Δx^k,

$$f_i(x^k+\Delta x^k)=\max f_j(x^k+\Delta x^k).$$

The constraint will stay active as long as its Lagrange multiplier remains positive.

Constraint i becomes semiactive in the kth iteration if $i\in(\mathrm{III})$ and, for an attempted step Δx,

$$f_i(x^k+\Delta x)>\max f_j(x^k)$$

or if

$$i\in(\mathrm{I}),\qquad \text{but } v_i^k<0.$$

The constraint will stay semiactive as long as its Lagrange multiplier remains positive.

We now consider the following problem:

$$\begin{aligned}
(\mathrm{P3})\quad &\min_{x,t,w}\quad t,\\
&\text{s.t.}\quad f_i(x)-t=0, && i\in(\mathrm{I}),\\
&\qquad\;\; f_i(x)-t+(1/2)w_i^2=0, && i\in(\mathrm{II}).
\end{aligned}$$

We have thus transformed our original problem into a problem of equality-constrained minimization for which successful algorithms are available. We use a trust-region approach to solve (P3), while taking into advantage the structure of the problem. Tapia (Ref. 1) demonstrated that the addition of slack variables in (P3) does not result in additional work and provides a good way of dealing with the inequality constraints in (P2).

Let F_1, F_2 denote the function vector for sets (I), (II), respectively; and let ∇F_1, ∇F_2 denote their gradients (in columns). We assign a Lagrange multiplier v_i to each active constraint i and a Lagrange multiplier u_j to each semiactive constraint j. Thus, we obtain the Lagrangian function

$$L(x,t,w,v,u)=t+\sum_{i\in(\mathrm{I})}v_i(f_i(x)-t)+\sum_{i\in(\mathrm{II})}u_i(f_i(x)-t+(1/2)w_i^2).$$

The gradient of L is

$$\nabla L = \begin{bmatrix} \nabla F_1 v + \nabla F_2 u \\ 1 - \sum v_i - \sum u_i \\ (u_i w_i), \qquad \forall i \in (\mathrm{II}) \\ (f_i(x) - t), \qquad \forall i \in (\mathrm{I}) \\ (f_i(x) - t + (1/2) w_i^2), \qquad \forall i \in (\mathrm{II}) \end{bmatrix},$$

and the Hessian matrix of L is

$$\nabla^2 L = \begin{bmatrix} B & 0 & 0 & \nabla F_1 & \nabla F_2 \\ 0 & 0 & 0 & -e^T & -e^T \\ 0 & 0 & \mathrm{diag}(u_i) & 0 & \mathrm{diag}(w_i) \\ \nabla F_1^T & -e & 0 & 0 & 0 \\ \nabla F_2^T & -e & \mathrm{diag}(w_i) & 0 & 0 \end{bmatrix},$$

where

$$B = \sum_{i \in (\mathrm{I})} v_i \nabla^2 f_i(x) + \sum_{i \in (\mathrm{II})} u_i \nabla^2 f_i(x).$$

Assume that we have in a certain iteration x, t, w, v, u. A Newton-type step would then be determined by

$$\begin{bmatrix} \Delta x \\ \Delta t \\ \Delta w \\ \Delta v \\ \Delta u \end{bmatrix} = -(\nabla^2 L)^{-1} \nabla L. \tag{1}$$

We multiply this equation by $\nabla^2 L$ from the left and consider the result by components

$$\begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta t \end{bmatrix} + \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix} v_+ + \begin{bmatrix} \nabla F_2 \\ -e^T \end{bmatrix} u_+ = 0, \tag{2}$$

$$u_i \Delta w_i + u_{+i} w_i = 0, \qquad i \in (\mathrm{II}), \tag{3}$$

$$\begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}^T \begin{bmatrix} \Delta x \\ \Delta t \end{bmatrix} + (F_1 - te) = 0, \tag{4}$$

$$\begin{bmatrix} \nabla F_2 \\ -e^T \end{bmatrix}^T \begin{bmatrix} \Delta x \\ \Delta t \end{bmatrix} + \mathrm{diag}(w_i \Delta w_i) = 0, \tag{5}$$

where

$$v_+ = v + \Delta v, \qquad u_+ = u + \Delta u.$$

In (5), we assumed that, for all $i \in (\mathrm{II})$,

$$f_i(x) - t + (1/2) w_i^2 = 0,$$

because in each iteration we will take

$$t = \max f_j(x) \quad \text{and} \quad (1/2) w_i^2 = t - f_i(x).$$

We can now eliminate the slack variables from our system. From (3), we have

$$\Delta w_i = -u_{i_+}(w_i/u_i);$$

from (5), we have

$$\begin{bmatrix} \nabla F_2 \\ -e^T \end{bmatrix}^T \begin{bmatrix} \Delta x \\ \Delta t \end{bmatrix} - \operatorname{diag}(w_i^2/u_i) u_+ = 0.$$

Since

$$w_i^2 = 2(\max f_j(x) - f_i(x)),$$

we have

$$u_+ = \operatorname{diag}(u_i/2)(\max f_j(x) - f_i(x)) \begin{bmatrix} \nabla F_2 \\ -e^T \end{bmatrix}^T \begin{bmatrix} \Delta x \\ \Delta t \end{bmatrix}. \tag{6}$$

We can now replace u_+ in (2). Define

$$C \equiv \left[\begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} \nabla F_2 \\ -e^T \end{bmatrix} \operatorname{diag}(u_i/2)(\max f_j(x) - f_i(x)) \begin{bmatrix} \nabla F_2 \\ -e^T \end{bmatrix}^T \right]. \tag{7}$$

Then, the linear system (2)–(5) becomes

$$\begin{bmatrix} C & \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix} \\ \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}^T & 0 \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \Delta x \\ \Delta t \end{bmatrix} \\ \Delta v \end{bmatrix} \begin{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix} v \\ (F_1 - te) \end{bmatrix}. \tag{8}$$

In this representation of the system (1), the semiactive constraints affect only the matrix C.

We now suggest a trust-region implementation in order to obtain good local and global convergence properties.

3. Trust-Region Strategy

Consider the problem

$$\min_{x \in \Re^n} \quad f(x),$$
$$\text{s.t.} \quad h_i(x)=0, \qquad i=1,\ldots,m,$$

where $f, h_1, \ldots, h_m : \Re^n \to \Re$, $m<n$. We assign a Lagrange multiplier v_i to each constraint i and form the Lagrangian function $L(x, v)=f(x)+h(x)^T v$. Then,

$$\nabla L(x, v)=\begin{bmatrix} \nabla_x L(x, v) \\ \nabla_v L(x, v) \end{bmatrix}=\begin{bmatrix} \nabla f(x)+\nabla h(x)v \\ h(x) \end{bmatrix},$$

$$\nabla^2 L(x, v)=\begin{bmatrix} \nabla^2 f(x)+\sum v_i \nabla^2 h_i(x) & \nabla h(x) \\ \nabla h(x)^T & 0 \end{bmatrix}.$$

Let

$$B \approx \nabla^2 f(x)+\sum v_i \nabla^2 h_i(x).$$

The quasi-Newton step for the problem is obtained by solving

$$\begin{bmatrix} B & \nabla h(x) \\ \nabla h(x)^T & 0 \end{bmatrix}\begin{bmatrix} \Delta x \\ \Delta v \end{bmatrix}=-\begin{bmatrix} \nabla_x L \\ h(x) \end{bmatrix}. \tag{9}$$

If the matrix B is positive definite (as is the case in this implementation), the following quadratic programming problem is equivalent to (9):

$$\min \quad L(x, v)+\nabla_x L(x, v)^T \Delta x+(1/2)\Delta x^T B \Delta x,$$
$$\text{s.t.} \quad h(x)+\nabla h(x)^T \Delta x=0.$$

The trust-region step is obtained by solving the problem

$$\min \quad L(x, v)+\nabla_x L(x, v)^T \Delta x+(1/2)\Delta x^T B \Delta x,$$
$$\text{s.t.} \quad \alpha h(x)+\nabla h(x)^T \Delta x=0,$$
$$\|\Delta x\|_2 \leq r,$$

where $0 \leq \alpha \leq 1$ depends on r and is determined so that

$$\{\Delta x: \alpha h(x)+\nabla h(x)^T \Delta x=0\} \cap \{\Delta x: \|\Delta x\|_2 \leq r\} \neq \emptyset.$$

The solution can be written, analogously to (9), as

$$\begin{bmatrix} B+\lambda I & \nabla h(x) \\ \nabla h(x)^T & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta v \end{bmatrix} = -\begin{bmatrix} \nabla_x L \\ \alpha(\lambda) h(x) \end{bmatrix};$$

here, $\lambda \geq 0$ is the Lagrange multiplier of the inequality constraint $\|\Delta x\|_2 \leq r$ and depends on r; $0 \leq \alpha(\lambda) \leq 1$. When r is large enough, $\lambda = 0$ and $\alpha(\lambda) = 1$. Details of the algorithm can be found in Vardi (Ref. 2).

4. Trust-Region Model for the Minmax Problem

We apply the trust-region strategy to the system (8). Let m_1, m_2 be the number of active and semiactive constraints in a given iteration. We assume in this section that the matrix $[\nabla F_1^T, -e]^T$ is of full rank. The case where $[\nabla F_1^T, -e]^T$ is rank deficient is discussed in Section 5. Let

$$Q\begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}\Pi = \begin{bmatrix} T \\ 0 \end{bmatrix}$$

represent a Q-R decomposition of the matrix $[\nabla F_1^T, -e]^T$, where Q is an orthogonal matrix, Π a permutation matrix, and T an upper triangular matrix. Let

$$Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix},$$

where Q_1 has m_1 rows. The new system becomes

$$\begin{bmatrix} C+\lambda I & \begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix} \\ \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}^T & 0 \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \Delta x \\ \Delta t \end{bmatrix} \\ \Delta v \end{bmatrix} = -\begin{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix} v \\ \alpha(\lambda)(F_1 - te) \end{bmatrix}, \tag{10}$$

where $\lambda \geq 0$ depends on r, the trust-region radius, and

$$\alpha(\lambda) = \min\{1, \max\{1, z\}/\lambda^2\}, \qquad 0 \leq \alpha(\lambda) \leq 1,$$

and

$$z = \frac{\|Q_2 C Q_1^T T^{-T} \Pi^T (F_1 - te)\|_2 \cdot \left\| Q_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\|_2}{\|T^{-T} \Pi^T (F_1 - te)\|_2^2}.$$

In (10), $\lambda=0$ if the radius is large enough so that the norm of $[\Delta x^T, \Delta t]^T$, computed for $\lambda=0$, is less than r; otherwise, $\lambda>0$ is determined so that the norm of $(\Delta x^T, \Delta t)^T$ is equal to r.

The next theorem summarizes the characteristics of the vectors $[\Delta x^T(\lambda), \Delta t(\lambda)]^T$ and Δv as a function of λ.

Theorem 4.1. Consider the vectors $[\Delta x^T(\lambda), \Delta t(\lambda)]^T$ and Δv as defined in (10). Then, norm$[\Delta x^T(\lambda), \Delta t(\lambda)]^T$ is monotonically decreasing to zero as a function of λ. When $\lambda\to\infty$,

$$v_+ = v+\Delta v(\lambda)\to\tilde{v}_+ \equiv\left[\begin{bmatrix}\nabla F_1\\ -e^T\end{bmatrix}^T\begin{bmatrix}\nabla F_1\\ -e^T\end{bmatrix}\right]^{-1} e \tag{11}$$

and

$$\lambda\begin{bmatrix}\Delta x\\ \Delta t\end{bmatrix}\to-\begin{bmatrix}\nabla F_1\\ -e^T\end{bmatrix}\tilde{v}_+ -\begin{bmatrix}0\\ 1\end{bmatrix}.$$

Also when

$$Q_2\begin{bmatrix}0\\ 1\end{bmatrix}=0\ (n+1 \text{ active constraints}) \text{ and } \lambda\to\infty,$$

there holds

$$\lambda^2\begin{bmatrix}\Delta x\\ \Delta t\end{bmatrix}\to-\max\{1, z\}\begin{bmatrix}\nabla F_1\\ -e^t\end{bmatrix}\left[\begin{bmatrix}\nabla F_1\\ -e^T\end{bmatrix}^T\begin{bmatrix}\nabla F_1\\ -e^T\end{bmatrix}\right]^{-1}(F_1-te).$$

Proof. The proof of monotonicity of norm $[\Delta x^T(\lambda), \Delta t(\lambda)]^T$ appears in Vardi (Ref. 2). From (10), we have

$$(C+\lambda I)\begin{bmatrix}\Delta x\\ \Delta t\end{bmatrix}+\begin{bmatrix}\nabla F_1\\ -e^T\end{bmatrix}v_+ +\begin{bmatrix}0\\ 1\end{bmatrix}=0, \tag{12}$$

$$\begin{bmatrix}\nabla F_1\\ -e^T\end{bmatrix}^T\begin{bmatrix}\Delta x\\ \Delta t\end{bmatrix}+\alpha(\lambda)(F_1-te)=0. \tag{13}$$

From (12), we have

$$\begin{bmatrix}\Delta x\\ \Delta t\end{bmatrix}=-(C+\lambda I)^{-1}\left[\begin{bmatrix}\nabla F_1\\ -e^T\end{bmatrix}v_+ +\begin{bmatrix}0\\ 1\end{bmatrix}\right], \tag{14}$$

and from (13),

$$v_+ = \left[\begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}^T (C+\lambda I)^{-1} \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}\right]^{-1}$$
$$\times \left[\alpha(\lambda)(F_1 - te) - \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}^T (C+\lambda I)^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right]. \tag{15}$$

When $\lambda \to \infty$,

$$v_+ \to \left[\begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}^T \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}\right]^{-1}$$
$$\times \left([\max\{1, z\}/\lambda](F_1 - te) - \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}^T \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) \to \tilde{v}_+ .$$

From (12),

$$\lambda \begin{bmatrix} \Delta x \\ \Delta t \end{bmatrix} \to -\begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix} v_+ - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \to -\begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix} \tilde{v}_+ - \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Note that, when

$$Q_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0,$$

we have

$$\left\{I - \begin{bmatrix} \nabla F_1 \\ e^T \end{bmatrix} \left[\begin{bmatrix} \nabla F_1 \\ e^T \end{bmatrix}^T \begin{bmatrix} \nabla F_1 \\ e^T \end{bmatrix}\right]^{-1} \begin{bmatrix} \nabla F_1 \\ e^T \end{bmatrix}^T\right\} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = Q_2^T Q_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0.$$

Thus,

$$\begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix} \tilde{v}_+ + \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0.$$

Further analysis reveals that, when

$$Q_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0 \quad \text{and} \quad \lambda \to \infty,$$

we have

$$\lambda (v_+ - \tilde{v}_+) \to \max\{1, z\} \left[\begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}^T \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}\right]^{-1} [F_1 - te],$$

and thus [by using (12)],

$$\lambda^2\begin{bmatrix}\Delta x\\ \Delta t\end{bmatrix}\to -\max\{1,z\}\begin{bmatrix}\nabla F_1\\ -e^T\end{bmatrix}\left[\begin{bmatrix}\nabla F_1\\ -e^T\end{bmatrix}^T\begin{bmatrix}\nabla F_1\\ -e^T\end{bmatrix}\right]^{-1}[F_1-te]. \qquad \square$$

Theorem 4.1 points out a nice feature of the algorithm: in the limit, when $\lambda\to\infty$, the vectors $[\Delta x^T, \Delta t]^T$ and Δv depend solely on information that comes from active constraints, while semiactive constraints and the matrix B are ignored.

In the course of the algorithm, we often check whether, for a given r, the resulting step Δx satisfies

$$\max f_j(x+\Delta x)<\max f_j(x).$$

If not, we try a shorter radius. The next theorem confirms that there always exists $r>0$ small enough (or equivalently, because norm$[\Delta x^T(\lambda), \Delta t(\lambda)]^T$ is monotonically decreasing, $\lambda>0$ large enough) such that the step is accepted.

Theorem 4.2. Let the vectors $[\Delta x^T(\lambda), \Delta ty(\lambda)]^T$ and Δv be computed as in (10), and assume that

$$\begin{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix}+\begin{bmatrix}\nabla F_1\\ -e^T\end{bmatrix}\tilde{v}_+\\ F_1-te\end{bmatrix}\neq 0.$$

Also, assume that, if

$$Q_2\begin{bmatrix}0\\ 1\end{bmatrix}=0,$$

then $(\tilde{v}_+)_i>0$, for all $i\in(\mathrm{I})$ such that $f_i(x)<\max f_j(x)$. Then, there exists $\lambda>0$ large enough such that

$$\max f_j(x+\Delta x(\lambda))<\max f_j(x).$$

Remark 4.1. When

$$\operatorname{rank}\begin{bmatrix}\nabla F_1\\ -e^T\end{bmatrix}=n+1,$$

we assume that

$$Q_2\begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0.$$

Proof. From Taylor's theorem, we have

$$\lambda f_i(x+\Delta x) - \lambda f_i(x) = \lambda \nabla f_i(x)^T \Delta x + \lambda \Delta x^T \nabla^2 f_i(\eta) \Delta x, \qquad i \in (\mathrm{I}).$$

When $\lambda \to \infty$,

$$\lambda F_1(x+\Delta x) - \lambda F_1(x) \to -\nabla F_1(x)^T \nabla F_1(x) \tilde{v}_+ .$$

If

$$Q_2\begin{bmatrix} 0 \\ 1 \end{bmatrix} \neq 0,$$

we have

$$\begin{aligned} &\nabla F_1(x)^T \nabla F_1(x) \tilde{v}_+ \\ &= [\nabla F_1(x)^T \nabla F_1(x) \pm ee^T][\nabla F_1(x)^T \nabla F_1(x) + ee^T]^{-1} e \\ &= e\{1 - e^T[\nabla F_1(x)^T \nabla F_1(x) + ee^T]^{-1} e\} > 0. \end{aligned}$$

Thus, for λ large enough,

$$f_i(x+\Delta x) < f_i(x), \qquad i \in (\mathrm{I}).$$

The fact that

$$Q_2\begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0$$

implies that (see Theorem 4.1)

$$\begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix} \tilde{v} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0.$$

In this case, we assumed that

$$F_1 - te \neq 0 \text{ and } (\tilde{v}_+)_i > 0, \qquad \text{for all } i \text{ such that } f_i(x) < \max f_j(x).$$

In Theorem 4.1, we have shown that

$$\lambda^2 \Delta t \to \max\{1, z\}\, e^T \left[\begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}^T \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix} \right]^{-1} (F_1 - te)$$

$$= \max\{1, z\} \tilde{v}_+^T (F_1 - te) < 0.$$

Thus, for λ large enough, $\Delta t < 0$ and

$$f_i(x + \Delta x) - (t + \Delta t)$$

$$= f_i(x) - t + \begin{bmatrix} \nabla f_i \\ -1 \end{bmatrix}^T \begin{bmatrix} \Delta x \\ \Delta t \end{bmatrix}$$

$$+ (1/2) \begin{bmatrix} \Delta x \\ \Delta t \end{bmatrix}^T \begin{bmatrix} \nabla^2 f_i(\eta) & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta t \end{pmatrix}, \qquad i \in (\mathrm{I}).$$

When $\lambda \to \infty$,

$$\lambda^2 \{[F_1(x + \Delta x) - (t + \Delta t)e] - [F_1(x) - te]\}$$

$$\to \max\{1, z\} \begin{bmatrix} \nabla F_1 \\ -e_T \end{bmatrix}^T \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix} \left[\begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}^T \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix} \right]^{-1} (F_1 - te) < 0.$$

Thus, for λ large enough,

$$F_1(x + \Delta x) < F_1(x) + \Delta t\, e < F_1(x).$$

This completes the proof. □

We now present the algorithm.

Algorithm.

Step 1. Start with x^0, r^0, B^0, $(\mathrm{II}) = \varnothing$, $(\mathrm{I}) = \{i : f_i(x^0) = \max f_j(x^0)\}$, $k = -1$.

Step 2. Let $k = k + 1$, $D = 0$. (D stands for the number of radius increases or decreases for each x^k.)

Step 3. Compute a Q-R decomposition of $[\nabla F_1^T(x^k), -e]^T$ and check whether

$$Q_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0.$$

If so, compute $\tilde{v}_+$ by (11). If $(\tilde{v}_+)_i > 0$, for all $i \in (\mathrm{I})$, such that $f_i(x) < \max f_j(x)$, or if

$$Q_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \neq 0,$$

continue to Step 4. Otherwise, move the constraint with the smallest $(\tilde{v}_+)_i$ among constraints satisfying $f_i(x) < \max f_j(x)$ from set (I) to set (II).

Step 4. Compute C^k as in (7).

Step 5. Find λ^k, $[\Delta x^{kT}, \Delta t^k]^T$, Δv^k that solve the system (10) under the constraints

$$\left\| \begin{bmatrix} \Delta x^k \\ \Delta t^k \end{bmatrix} \right\|_2 \leq r^k, \qquad \lambda^k \geq 0, \qquad \lambda^k \left[\left\| \begin{bmatrix} \Delta x^k \\ \Delta t^k \end{bmatrix} \right\|_2 - r^k \right] = 0.$$

For details see Section 5.

Step 6. Check if

$$\max f_j(x^k + \Delta x^k) < \max f_j(x^k).$$

If so, continue to Step 7. If not, if D is nonpositive, then halve r^k, set D to D-1, and go to Step 5. If D is positive, the step corresponding to the smaller radius has already been computed. Retrieve it and continue to Step 7.

Step 7. If no new active constraints are introduced by the last step and if D is nonnegative and λ^k is positive, let D be $D+1$, store the current step, double r^k, and go to Step 4. Otherwise, continue to Step 8.

Step 8. Check for convergence; see Section 6.

Step 9. Compute u^{k+1} by (6) and v^{k+1} by (15) and, according to their signs, update (I) and (II); see Section 2. Compute B^{k+1}; see Section 5.

Step 10. Set x^{k+1} to $x^k + \Delta x^k$. Return to Step 2.

Theorem 4.2 guarantees that, in every iteration, there will be r^k small enough for which

$$\max f_j(x^{k+1}) < \max f_j(x^k).$$

The removal of an active constraint in Step 3 when

$$Q_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0$$

(i.e., $m_1 = n+1$; for a discussion of the possibility that

$$Q_2\begin{bmatrix}0\\1\end{bmatrix}=0$$

when $[\nabla F_1^T, -e]^T$ is rank deficient, see Section 5) is done in order to prevent convergence to a point that is not a solution of our problem.

5. Implementation

Scaling Problem. In order to prevent a scaling problem that may occur in $[\nabla F_1^T, -e]^T$ when $\|\nabla F_1\| \gg 1$, it is important to actually replace (P2) by

$$\text{(P2')}\quad \min\ ct,\qquad \text{s.t.}\quad f_i(x) - ct < 0, \qquad i = 1, \ldots, m.$$

Now, instead of working with the matrix $[\nabla F_1^T, -e]^T$, we work with the matrix $[\nabla F_1^T, -ce]^T$, where $c = \|\nabla F_1(x)\|$. This modification can be accomplished without difficulties. For clarity, we will continue to use $c=1$ in this paper.

Initialization of (I) and (II). As Step 1 in the algorithm indicates, at the first iteration usually only one constraint is active and all the rest are nonactive. Other options were tried with more active constraints in the first iteration, but this one gave the best results.

Solving System (10). Using the Q-R decomposition of the matrix $[\nabla F_1^T, -e]^T$,

$$Q\begin{bmatrix}\nabla F_1\\ -e^T\end{bmatrix}\Pi = \begin{bmatrix}T\\0\end{bmatrix}, \qquad Q = \begin{bmatrix}Q_1\\Q_2\end{bmatrix}, \qquad Q_1 \text{ has } m_1 \text{ rows},$$

we obtain the following expressions from (13) and (12), respectively:

$$Q_1\begin{bmatrix}\Delta x\\ \Delta t\end{bmatrix} = -\alpha(\lambda)T^{-T}\Pi^T(F_1 - te), \tag{16}$$

$$Q_2\begin{bmatrix}\Delta x\\ \Delta t\end{bmatrix} = -[Q_2CQ_2^T + \lambda I]^{-1}Q_2\left[\begin{bmatrix}0\\1\end{bmatrix} + CQ_1^TQ_1\begin{bmatrix}\Delta x\\ \Delta t\end{bmatrix}\right]. \tag{17}$$

Finally,

$$\begin{bmatrix}\Delta x\\ \Delta t\end{bmatrix} = Q_1^TQ_1\begin{bmatrix}\Delta x\\ \Delta t\end{bmatrix} + Q_2^TQ_2\begin{bmatrix}\Delta x\\ \Delta t\end{bmatrix}. \tag{18}$$

A Cholesky decomposition of C is used, $C = LL^T$. When

$$m_2 = 0 \quad \text{and} \quad C = \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix},$$

we have

$$L = \begin{bmatrix} L_1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad \text{where } B = L_1[L_1^T].$$

Define $M = Q_2 L$, and let

$$PM^T\Sigma = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

represent a Q-R decomposition of M^T, with P orthogonal, Σ permutation, R upper triangular. Partition P into

$$P = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix},$$

where P_1 has $n+1-m_1$ rows. We now have

$$(Q_2 C Q_2^T + \lambda I) = \Sigma(R^T R + \lambda I)\Sigma^T,$$

and $Q_2[\Delta x^T, \Delta t]^T$ can be obtained by solving

$$\begin{bmatrix} R \\ \lambda^{1/2} I \end{bmatrix}^T \begin{bmatrix} R \\ \lambda^{1/2} I \end{bmatrix} \left[\Sigma^T Q_2 \begin{bmatrix} \Delta x \\ \Delta t \end{bmatrix} \right] = \begin{bmatrix} R \\ \lambda^{1/2} I \end{bmatrix}^T \begin{bmatrix} b \\ 0 \end{bmatrix},$$

where

$$b = R^{-T} \Sigma^T Q_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} + P_1 L^T Q_1^T Q_1 \begin{bmatrix} \Delta x \\ \Delta t \end{bmatrix}.$$

In order to solve this linear least-squares problem, a Q-R decomposition of $[R^T, \lambda^{1/2} I]^T$ is obtained with the use of Givens transformations. The step is finally computed by (18).

Define

$$\Phi(\lambda) = \left\| \begin{bmatrix} \Delta x \\ \Delta t \end{bmatrix} \right\|_2 - r.$$

For a given λ, we can use the above decomposition to compute $\Phi(\lambda)$ and $\Phi'(\lambda)$. A λ-iterative process is used to obtain λ such that $\Phi(\lambda) = 0$. In this process, we usually start with a good guess for λ and take iteratively

$$\lambda^{j+1} = \lambda^j - [(\Phi(\lambda^j) + r)/r][\Phi(\lambda^j)/\Phi'(\lambda^j)].$$

Known upper and lower bounds on the solution are used in the process. It takes on average less than two λ-iterations to obtain an acceptable solution, i.e., λ such that $|\Phi(\lambda)| \leq 0.1r$. This iterative process is due to Hebden (Ref. 3) and More (Ref. 4). For more details on this part of the algorithm, see Vardi (Ref. 2).

Updating B. In the $(k+1)$th iteration, the matrix

$$B^k \approx \sum_{i\in(\mathrm{I})^k} v_i^k \nabla_{xx}^2 f_i(x^k) + \sum_{i\in(\mathrm{II})^k} u_i^k \nabla_{xx}^2 f_i(x^k)$$

has to be updated to

$$B^{k+1} \approx \sum_{i\in(\mathrm{I})^{k+1}} v_i^{k+1} \nabla_{xx}^2 f_i(x^{k+1}) + \sum_{i\in(\mathrm{II})^{k+1}} u_i^{k+1} \nabla_{xx}^2 f_i(x^{k+1}).$$

Define

$$y^k = [\nabla F_1(x^{k+1}) - \nabla F_1(x^k)]v^{k+1} + [\nabla F_2(x^{k+1}) - \nabla F_2(x^k)]u^{k+1}.$$

The BFGS update is now used,

$$\begin{aligned} B^{k+1} &= \mathrm{BFGS}(B^k, \Delta x^k, y^k) \\ &= B^k + y^k y^{k^T}/y^{k^T}\Delta x^k - B^k \Delta x^k \Delta x^{k^T} B^{k^T}/\Delta x^{k^T} B^k \Delta x^k. \end{aligned}$$

If B^k is positive definite and $y^{k^T}\Delta x^k > 0$, B^{k+1} is positive definite. In our implementation, we use Powell's modification [see Powell (Ref. 5)],

$$B^{k+1} = \mathrm{BFGS}(B^k, \Delta x^k, \bar{y}^k),$$

where

$$\bar{y}^k = \theta y^k + (1-\theta)B^k \Delta x^k,$$

$$\theta = \begin{cases} 1, & \text{if } y^{k^T}\Delta x^k > 0.2\Delta x^{k^T} B^k \Delta x^k, \\ \dfrac{0.8\Delta x^{k^T} B^k \Delta x^k}{(\Delta x^{k^T} B^k \Delta x^k - y^{k^T}\Delta x^k)}, & \text{otherwise.} \end{cases}$$

Rank-Deficient Matrix $[\nabla F_1^T, -e]^T$. When the Q-R decomposition is performed to obtain

$$Q\begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix}\Pi = \begin{bmatrix} T \\ 0 \end{bmatrix},$$

we check at each step to see whether any of the remaining columns has a norm larger than (machine eps) $\times T_{1,1}$. If not, we stop. If l steps were performed, and if T is an $l \times m_1$ upper triangular matrix, we consider

$$\operatorname{rank}\begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix} = l.$$

Partition

$$Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}, \qquad \Pi = [\Pi_1, \Pi_2], \qquad \bar{T} = [T, S],$$

so that Q_1 has l rows, and Π_1 and T have l columns. Equation (16) becomes

$$Q_1 \begin{bmatrix} \Delta x \\ \Delta t \end{bmatrix} = -\alpha(\lambda) T^{-T} \Pi_2^T (F_1 - te),$$

and all other expressions remain unchanged. The effect of this is that the active constraints that correspond to the remaining columns in $[\nabla F_1^T, -e]^T$ are ignored.

When $[\nabla F_1^T, -e]^T$ is rank deficient, it is possible that

$$Q_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0$$

and, as has been indicated, v_+ is checked and active constraints with negative $(\tilde{v}_+)_i$ may become semiactive.

6. Numerical Tests

A computer program based on the algorithm was written and tested on six test problems from the optimization literature. For each problem, two starting points were used, the second one far from the solution. We used $r^0 = 1$, $B^0 = \mathrm{I}$, and the stopping criteria

(i) $\|\Delta x\| \leq 10^{-10}(\|x\| + 1)$,

(ii) $\left\| \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} \nabla F_1 \\ -e^T \end{bmatrix} v_+ \right\| < 10^{-6}.$

The norm of the gradient of the Lagrangian is small.

We recorded the accumulating number of (individual) gradient evaluations. We counted one for each semiactive or active constraint in every iteration.

For the first starting point, we display the count for iterations which had a change in active status of the constraints.

Test Problem 1. Here, $n=2$, $m=3$, and

$$F_1(x)=x_1^2+x_2^4, \qquad F_2(x)=(2-x_1)^2+(2-x_2)^2,$$
$$F_3(x)=2\exp(-x_1+x_2).$$

The solution is

$$x=(1.139037652, 0.8995599384), \qquad F_1=F_2=1.952224494.$$

Starting point	Iteration number	Total number of gradient evaluations	$\max f_j(x)$	Semiactive constraints	Active constraints
(1, −0.1)	0	1	5.41	—	2
	7	13	1.952224494	—	1, 2
(100, −10)	0	1	20000	—	1
	12	23	1.952224494	—	1, 2

Test Problem 2. Here, $n=2$, $m=3$, and

$$F_1=x_1^4+x_2^2, \qquad F_2, F_3 \text{ as in Test Problem 1.}$$

The solution is

$$x=(1, 1), \qquad F_1=F_2=F_3=2.$$

Starting point	Iteration number	Total number of gradient evaluations	$\max f_j(x)$	Semiactive constraints	Active constraints
(1, −0.1)	0	1	5.41	—	2
	2	4	2.417247765	—	1, 2
	6	16	2.0000000000	—	1, 2, 3
	0	1	100000100	—	1
(100, −10)	17	29	2.000000000	—	1, 2, 3

Test Problem 3. Here, $n=4$, $m=4$, and

$$F_1(x)=x_1^2+x_2^2+2x_3^2+x_4^2+5x_1-5x_2-21x_3+7x_4,$$
$$F_2(x)=F_1(x)-10(-x_1^2-x_2^2-x_3^2-x_4^2-x_1+x_2-x_3+x_4+8),$$
$$F_3(x)=F_1(x)-10(-x_1^2-2x_2^2-x_3^2-2x_4^2+x_1+x_4+10),$$
$$F_4(x)=F_1(x)-10(-2x_1^2-x_2^2-x_3^2-2x_1+x_2+x_4+5).$$

The solution is

$$x=(0,1,2,-1), \qquad F_1=F_2=F_4=-44.$$

Starting point	Iteration number	Total number of gradient evaluations	$\max f_j(x)$	Semiactive constraints	Active constraints
(0, 0, 0, 0)	0	1	0	—	1
	3	5	−41.13057951	—	1, 4
	5	11	−41.67511609	2	1, 4
	10	26	−44.00000000	—	1, 2, 4
(100, 100	0	1	645500.	—	3
100, 100)	18	44	−44.00000000	—	1, 2, 4

Test Problem 4. Here, $n=2, m=3$, and

$$F_1(x)=x_1^2+x_2^2+x_1x_2, \qquad F_2(x)=\sin(x_1), \qquad F_3(x)=\cos(x_2).$$

The solution is

$$x=\pm(0.4532962370, -0.9065924741), \qquad F_1=F_3=0.6164324356.$$

Starting point	Iteration number	Total number of gradient evaluations	$\max f_j(x)$	Semiactive constraints	Active constraints
(3, 1)	0	1	13	—	1
	2	4	0.8053242777	—	1, 3
	4	10	0.7272455047	3	1, 2
	5	13	0.6428900754	—	1, 2, 3
	10	28	0.6164324356	2	1, 3
(300, 100)	0	1	130000	—	1
	11	19	0.6164324356	—	1, 3

Test Problem 5. Here, $n=3, m=6$, and

$$F_1(x)=x_1^2+x_2^2+x_3^2-1, \qquad F_4(x)=x_1+x_2-x_3+1,$$
$$F_2(x)=x_1^2+x_2^2+(x_3-2)^2, \qquad F_5(x)=2x_1^3+6x_2^2+2(5x_3-x_1+1)^2,$$
$$F_3(x)=x_1+x_2+x_3-1, \qquad F_6(x)=x_1^2-9x_3.$$

The solution is

$$x=(0.32825995, 0, 0.1313200636), \qquad F_2=F_5=3.599719300.$$

Starting point	Iteration number	Total number of gradient evaluations	$\max f_j(x)$	Semiactive constraints	Active constraints
(1, 1, 1)	0	1	58	—	5
	13	25	3.599719300	—	2, 5
(100, 100, 100)	0	1	2381602	—	5
	15	28	3.599719300	—	2, 5

Test Problem 6. Here, $n=3$, $m=30$, and

$$F_j(x)=-y_j+x_1+u_j/(v_jx_2+w_jx_3), \qquad j=1,\ldots,15,$$

$$F_j(x)=-F_{j-15}(x), \qquad j=16,\ldots,30,$$

where

$u_j=j,\ v_j=16-j, \qquad w_j=\min\{u_j, v_j\},$

$y=(0.14, 0.18, 0.22, 0.25, 0.32, 0.39, 0.37, 0.58, 0.73, 0.96, 1.34, 2.1, 4.39).$

The solution is

$$x=\pm(0.05346938776, t, 3.5-t), \qquad 0.5\leq t\leq 1.5,$$

$$F_9=F_{23}=F_{30}=0.05081632653 \quad \text{or}$$

$$F_8=F_{15}=F_{24}=0.05081632653.$$

Starting point	Iteration number	Total number of gradient evaluations	$\max f_j(x)$	Semiactive constraints	Active constraints
(1, 1, 1)	0	1	4.11	—	15
	2	4	0.3286634371	—	9, 15
	3	7	0.2339086843	15	9, 18
	5	15	0.05443554245	15	9, 18, 30
	7	21	0.05115868122	—	9, 18, 30
	8	25	0.05081633724	—	9, 18, 23, 30
	9	29	0.05081632653	18	9, 23, 30
(100, 100, 100)	0	1	99.860625	—	1
	25	56	0.05081632653	18	9, 23, 30

References

1. TAPIA, R. A., *On the Role of Slack Variables in Quasi-Newton Methods for Constrained Optimization*, Numerical Optimization of Dynamic Systems, Edited by L. C. W. Dixon and G. P. Szego, pp. 235–246, 1980.
2. VARDI, A., *A Trust-Region Algorithm for Equality Constrained Minimization: Convergence Properties and Implementation*, SIAM Journal on Numerical Analysis, Vol. 22, pp. 575–591, 1985.
3. HEBDEN, M. D., *An Algorithm for Minimization Using Exact Second Derivatives*, Technical Paper 515, Atomic Energy Research Establishment, Harwell, England, 1973.
4. MORE, J. J., *The Levenberg–Marquardt Algorithm: Implementation and Theory*, Numerical Analysis, Edited by G. A. Watson, Springer-Verlag, Berlin, Germany, Vol. 630, pp. 105–116, 1978.
5. POWELL, M. J. D., *The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations*, Nonlinear Programming 3, Edited by R. R. Meyer and S. M. Robinson, Academic Press, New York, New York, 1978.
6. AVRIEL, M., *Nonlinear Programming: Analysis and Methods*, Prentice-Hall, Englewood Cliffs, New Jersey, 1976.
7. CHARALAMBOUS, C., and CONN, A. R., *An Efficient Method to Solve the Minimax Problem Directly*, SIAM Journal on Numerical Analysis, Vol. 15, pp. 162–187, 1978.
8. HAN, S. P., *A Globally Convergent Method for Nonlinear Programming*, Journal of Optimization Theory and Applications, Vol. 22, pp. 297–309, 1977.
9. MURRAY, W., and OVERTON, L., *A Projected Lagrangian Algorithm for Nonlinear Minimax Optimization*, Technical Report No. 79-21, Systems Optimization Laboratory, Stanford University, Stanford, California, 1979.
10. VARDI, A., *A Trust-Region Algorithm for Unconstrained Minimization: Convergence Properties and Implementation*, NASA Langley Research Center, Hampton, Virginia, ICASE Report No. 80-35, 1980.