NASA-CR-172,382

NASA Contractor Report 172382

ICASE REPORT NO. 84-25

NASA-CR-172382
19840021434

ICASE

A NEW MINMAX ALGORITHM

Avi Vardi

Contract No. NAS1-15810
June 1984

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

A NEW MINMAX ALGORITHM

Avi Vardi*
RAFAEL, Armament Development Authority
Haifa, Israel

ABSTRACT

The paper deals with the minimax problem $\min_{x \in \mathbb{R}^n} \max_{i=1,\cdots,m} f_i(x)$. We work with its equivalent representation $\min t$ s.t. $f_i(x) - t \leq 0$ for all i. For this problem we design a new active set strategy in which there are three types of functions: active, semi-active, and non-active. This technique will help in preventing zigzagging which often occurs when an active set strategy is used. Some of the inequality constraints are handled with slack variables. Also a trust region strategy is used in which at each iteration there is a sphere around the current point in which the local approximation of the function is trusted. The algorithm suggested in the paper was implemented into a successful computer program. Numerical results are provided.

*This research 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, VA 23665.

INTRODUCTION

This paper deals with the minimax problem $\min_{x \in \mathbb{R}^n} \max_{i=1,\cdots,m} f_i(x)$ where f_i, $i=1,\cdots,m$ are real valued functions defined on $\mathbb{R}^n$. We begin by transforming the problem into an equivalent inequality constrained minimization problem $\min t$ s.t. $f_i(x) - t \leq 0$ for all i, $i=1,\cdots,m$. For this problem we suggest a new active set strategy in which there are three types of functions: nonactive, semi-active and active and these sets play a different role in our algorithm. The active ones are treated as equality constraints; the semi-active ones are assigned slack variables so that they can be treated as equalities too. The introduction of semi-active functions may help prevent the possibility of zigzagging that sometimes occurs in algorithms that use active set strategy.

At the end we solve an equality constrained minimization problem for which we design a trust region algorithm that takes into advantage the special structure of the problem. In this algorithm we have at every iteration a sphere of radius r, in which the local model that is used to approximate the functions is trusted.

Section 2 contains the basic model with all the necessary notation as well as the introduction of the new active set strategy. In Section 3 we give a description of the trust region strategy in unconstrained minimization and in constrained minimization. We suggest the use of the trust region for the minimax problem in Section 4. In Section 5 we discuss our numerical implementation of the algorithm and in Section 6 we give the numerical results of six problems taken from the literature with various starting points.

2. THE BASIC MODEL

Consider m real valued functions $f_1,\cdots,f_m$ defined on $\mathbb{R}^n$. We are interested in solving the problem

$$\text{(P1)} \qquad \min_{x \in \mathbb{R}^n} \max_{i=1,\cdots,m} f_i(x)$$

(P1) is equivalent to (P2):

$$\text{(P2)} \qquad \min_{x,t} \; t$$
$$\text{subject to } \; f_i(x) - t \leq 0, \; i=1,\cdots,m.$$

By introducing m slack variables $w_1,\cdots,w_m$ we obtain another equivalent problem:

$$\text{(P3)} \qquad \min_{x,t,w} \; t$$
$$\text{subject to } \; f_i(x) - t + \tfrac{1}{2} w_i^2 = 0, \; i=1,\cdots,m.$$

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 advantage of the structure of the problem. We demonstrate that the addition of m variables in (P3) does not result in additional work and provides a good way of dealing with the inequality constraints in (P2).

The following notation will be used:

$$F = \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix} \qquad \nabla F = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_1} \\ \vdots & & \vdots \\ \frac{\partial f_1}{\partial x_n} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix} .$$

Associated with each index j we have a Lagrange multiplier v_j. Thus we obtain the Lagrangian function

$$L(x,t,v,w) = t + \sum v_i\left(f_i(x) - t + \tfrac{1}{2} w_i^2\right).$$

The gradient of L is

$$\nabla L = \nabla_{x,t,w,v} L = \begin{pmatrix} \nabla F v & \\ 1 - \Sigma v_i & \\ (v_i w_i) \quad , & i=1,\cdots,m \\ \left(f_i(x) - t + \tfrac{1}{2} w_i^2\right), & i=1,\cdots,m \end{pmatrix},$$

and the Hessian matrix of L is

$$\nabla^2 L = \begin{pmatrix} B & 0 & 0 & \nabla F \\ 0 & 0 & 0 & -e^T \\ 0 & 0 & \mathrm{diag}(v_i) & \mathrm{diag}(w_i) \\ \nabla F^T & -e & \mathrm{diag}(w_i) & 0 \end{pmatrix},$$

where $e^T = (1,\cdots,1)$, $B = \Sigma v_i \nabla^2 f_i(x)$.

We now state the necessary conditions for a solution to (P2).

Theorem 2.1. Let $x^* \in \mathbb{R}^n$ and assume that $\begin{pmatrix} \nabla F(x^*) \\ -e^T \end{pmatrix}$ is of full rank. Necessary conditions for a local minimum at x^* (with $t^* = \max f_j(x^*)$) are: there exists $v^* \in \mathbb{R}^m$ such that

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

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

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

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

and for all $z \in \mathbb{R}^m$ such that $\begin{pmatrix} \nabla F(x^*) \\ -e^T \end{pmatrix}^T z = 0$, $z^T \begin{pmatrix} B & 0 \\ 0 & 0 \end{pmatrix} z > 0$.

(The last condition in the theorem will always be satisfied in our algorithm because we use a BFGS matrix update to obtain a symmetric positive-definite approximation to $\Sigma v_i \nabla^2 f_i(x)$. For an appropriate discussion see Powell [5].)

At the solution (x^*, v^*), some components of v^* are zero, and the corresponding functions do not influence the direction of convergence. This leads to a natural division of the functions into active and non-active ones, and to an active-set strategy to determine the active set at each iteration.

We suggest here a division of the functions into three sets: active functions, semi-active functions and non-active functions. These sets are denoted by (I), (II), and (III) respectively. A function f_i becomes active at a certain iteration if $f_i(x) = \max_j f_j(x)$, or if it was semi-active and prevented us from taking a longer step in the previous iteration (i.e., $f_i(x + \Delta\bar{x})$ was greater than $\max_j f_j(x + \Delta x)$ for $\|\Delta\bar{x}\| > \|\Delta x\|$.) It will stay active as long as its Lagrange multiplier remains positive. Set (I) will not contain more than (n+1) functions at any iteration (since at most (n+1) can be active at the solution).

A function f_i becomes semi-active if it has just been dropped from set (I) (in which case its multiplier is reinitialized with a positive value), or if it was non-active and prevented us from taking a longer step in the previous iteration. It will stay semi-active as long as its associated multiplier remains positive. In a neighborhood of the solution we expect to have no semi-active functions.

We now consider the problem

$$\min_{x,t,w} \quad t$$
$$\begin{cases} f_i(x) - t = 0 , & i \text{ active} \\ f_i(x) - t + \tfrac{1}{2} w_i^2 = 0, & i \text{ semi-active.} \end{cases}$$

Let F_1, F_2 denote the function vector for sets (I), (II) respectively and ∇F_1, ∇F_2 their gradients (in columns). Each active function f_i will be associated with Lagrange multiplier v_i and each semi-active function f_j will be associated with multiplier u_j. Now the Lagrangian becomes

$$L(x,t,w,u,v) = t + \sum_{i \,\varepsilon\, (I)} v_i\bigl(f_i(x) - t\bigr) + \sum_{i \,\varepsilon\, (II)} u_i\bigl(f_i(x) - t + \tfrac{1}{2} w_i^2\bigr). \tag{2.1}$$

$$\nabla L = \begin{bmatrix} \nabla f_1 v + \nabla f_2 u & \\ 1 - \Sigma v_i - \Sigma u_i & \\ (u_i w_i) & , \quad i \in (II) \\ (f_i(x) - t) & , \quad i \in (I) \\ (f_i(x) - t + \tfrac{1}{2} w_i^2), & \quad i \in (II) \end{bmatrix} \tag{2.2}$$

$$\nabla^2 L = \begin{bmatrix} B & 0 & 0 & \nabla F_1 & \nabla F_2 \\ 0 & 0 & 0 & -e^T & e^T \\ 0 & 0 & \mathrm{diag}(u_i) & 0 & \mathrm{diag}(w_i) \\ \nabla F_1^T & -e & 0 & 0 & 0 \\ \nabla F_2^T & -e & \mathrm{diag}(w_i) & 0 & 0 \end{bmatrix} \tag{2.3}$$

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

$$\begin{bmatrix} \Delta x \\ \Delta t \\ \Delta w \\ \Delta v \\ \Delta u \end{bmatrix} = -\left(\nabla^2 L\right)^{-1} \nabla L. \tag{2.4}$$

When we multiply by $\nabla^2 L$ from the left and consider the resulting equation by components we obtain:

$$\begin{pmatrix} B & 0 & \Delta x \\ & & \\ 0 & 0 & \Delta t \end{pmatrix} + \begin{pmatrix} \nabla F_1 \\ \\ -e^T \end{pmatrix} v_+ + \begin{pmatrix} \nabla F_2 \\ \\ -e^T \end{pmatrix} u_+ = 0, \tag{2.5}$$

$$u_i \Delta w_i + u_{+i} w_i = 0, \qquad \text{for all} \quad i \in (II), \tag{2.6}$$

$$\begin{pmatrix} \nabla F_1 \\ \\ -e^T \end{pmatrix}^T \begin{pmatrix} \Delta x \\ \\ \Delta t \end{pmatrix} + \left(F_1 - te\right) = 0, \tag{2.7}$$

$$\begin{pmatrix} \nabla F_2 \\ \\ -e^T \end{pmatrix}^T \begin{pmatrix} \nabla x \\ \\ \nabla t \end{pmatrix} + \operatorname{diag}\left(w_i \Delta w_i\right) = 0, \tag{2.8}$$

where

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

In (2.8) we assumed that for all $i \in (II)$ $f_i(x) - t + \frac{1}{2} w_i^2 = 0$ because in each iteration we will take $t = \max_j f_j(x)$ and $\frac{1}{2}\left(w_i\right)^2 = t - f_i(x)$.

We can now eliminate the slack variables from our system:

from (2.6)

$$\Delta w_i = - u_{i_+} \frac{w_i}{u_i},$$

from (2.8)

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

Since

$$w_i^2 = 2\left(\max_j f_j(x) - f_i(x)\right),$$

$$u_+ = \mathrm{diag}\left(u_i/2(\max_j f_j(x) - f_i(x)\right)\begin{pmatrix}\nabla F_2 \\ -e^T\end{pmatrix}^T \begin{pmatrix}\Delta x \\ \Delta t\end{pmatrix}. \qquad (2.9)$$

We can now replace u_+ in (2.5). Define

$$C \equiv \left[\begin{pmatrix}B & 0 \\ 0 & 0\end{pmatrix} + \begin{pmatrix}\nabla F_2 \\ -e^T\end{pmatrix} \mathrm{diag}\left(u_i/2(\max_j f_j(x) - f_i(x)\right)\begin{pmatrix}\nabla F_2 \\ -e^T\end{pmatrix}^T\right]. \qquad (2.10)$$

Then the linear system (2.5) - (2.8) becomes

$$\begin{bmatrix} C & \begin{pmatrix}\nabla F_1 \\ -e^T\end{pmatrix} \\ \begin{pmatrix}\nabla F_1 \\ -e^T\end{pmatrix}^T & 0\end{bmatrix}\begin{pmatrix}\begin{pmatrix}\Delta x \\ \Delta t\end{pmatrix} \\ \Delta v\end{pmatrix} = -\begin{bmatrix}\begin{pmatrix}0 \\ 1\end{pmatrix} + \begin{pmatrix}\nabla F_1 \\ -e^T\end{pmatrix} v \\ F_1 - t1\end{bmatrix}. \qquad (2.11)$$

In this last representation of the system (2.4) the semi-active functions affect only the matrix C.

An iterative algorithm that is based on (2.4) will converge to a solution only if (x^0,t^0,w^0,v^0,u^0) is close enough to the solution. In order to obtain convergence from bad starting points we suggest the use of the trust region strategy applied to problem (P4).

3. THE TRUST REGION STRATEGY

We first describe the trust region in the unconstrained case. The problem is $\min_{x \in \mathbb{R}^n} f(x)$ where $f : \mathbb{R}^n \to \mathbb{R}$. For a step Δx with $\|\Delta x\|$ small enough we have

$$f(x + \Delta x) \approx q(x + \Delta x) \equiv f(x) + \nabla f(x)^T \Delta x + 1/2\, \Delta x^T B \Delta x, \qquad (3.1)$$

where $\nabla f(x)$ is the gradient of f at the point x and B is a symmetric positive definite matrix approximating the Hessian of f at the point x. If we try to minimize the quadratic function q over Δx we will obtain $\Delta x = -B^{-1}\nabla f(x)$, the quasi-Newton step. The quadratic approximation in (3.1) is only valid for $\|\Delta x\|$ small enough. $\Delta x = -B^{-1}\nabla f(x)$ may not always be a good step to take. In a trust region algorithm we assume that at each iteration we have a radius r that was determined at the end of the previous iteration as an estimate on the radius of the ball with a center at x in which the approximation (3.1) can be trusted. Thus we obtain the problem

$$\min_{\Delta x} q(x + \Delta x) = f(x) + \nabla f(x)^T \Delta x + 1/2\, \Delta x^T B \Delta x$$
$$\text{s.t. } \|\Delta x\| \le r.$$

The solution to this problem is $\Delta x = \Delta x(\lambda) = -(B + \lambda I)^{-1}\nabla f(x)$ where

$$\begin{cases} \lambda = 0 & \text{if } \|\Delta x(0)\| \le r \\ \lambda > 0 \text{ is s.t. } \|\Delta x(\lambda)\| = r & \text{otherwise} . \end{cases}$$

When the radius is large enough, $\lambda = 0$ and the full quasi-Newton step is taken. We can prove that in the neighborhood of the solution, when a BFGS matrix update is used to obtain B, $\lambda = 0$ in each iteration resulting in Q-super-linear convergence. When r is very small, $\lambda >> 0$, $\Delta x \approx -\frac{1}{\lambda}\nabla f(x)$, and we obtain a short step in the negative gradient direction. More details on this algorithm including numerical results can be found in Vardi [8].

Consider next the problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad h_i(x) = 0 \quad i=1,\ldots m$$

where $f,h_1,\ldots,h_m : \mathbb{R}^n \to \mathbb{R}$, $m < n$. When we assign a Lagrange multilplier v_i to each of the constraints we can form the Lagrangian function $L(x,v) = f(x) + h(x)^T v$.

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

$$\nabla^2 L(x,v) = \begin{pmatrix} \nabla^2 f(x) + \Sigma v_1 \nabla^2 h_i(x) & \nabla h(x) \\ \nabla h(x)^T & 0 \end{pmatrix}.$$

Let

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

be a symmetric positive definite matrix

$$\overline{B} = \begin{pmatrix} B & \nabla h(x) \\ \nabla h(x)^T & 0 \end{pmatrix}.$$

Then the quasi-Newton step for this problem becomes

$$\begin{pmatrix} \Delta x \\ \Delta v \end{pmatrix} = - \overline{B}^{-1} \nabla L(x,v).$$

or

$$\begin{pmatrix} B & \nabla h(x) \\ \nabla h(x)^T & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta v \end{pmatrix} = - \begin{pmatrix} \nabla_x L \\ h(x) \end{pmatrix} . \tag{3.2}$$

The following quadratic programming problem is equivalent to (3.2): ($\nabla h(x)$ is assumed to be of full rank; B positive definite.)

$$\min_{\Delta x} L(x,v) + \nabla_x L(x,v)^T \Delta x + 1/2\, \Delta x^T B \Delta x \tag{3.3a}$$

$$\text{subject to} \quad h(x) + \nabla h(x)^T \Delta x = 0. \tag{3.3b}$$

To this problem we now add the contraint $\|\Delta x\|_2 \le r$, i.e., we impose a trust region of radius r. Since it is possible that

$$\{\Delta x : h(x) + \nabla h(x)^T \Delta x = 0\} \cap \{\Delta x : \|\Delta x\| \le r\} = \phi,$$

we have to make the following correction: solve at each iteration

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

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

$$\{\Delta x : \alpha h(x) + \nabla h(x)^T \Delta x = 0\} \cap \{\Delta x : \|\Delta x\| \le r\} \ne \phi,$$

and that the norm of the resulting step Δx is a monotonically increasing function of r (so that for each r there will be a unique solution Δx).

Given r we determine λ as follows

$$\begin{cases} \lambda = 0 & \text{if } \|\Delta x(\lambda)\| < r \\ \lambda > 0 \text{ such that } \|\Delta x(\lambda)\| = r & \text{otherwise.} \end{cases}$$

The solution just presented can be written, similarly to (3.2) as

$$\begin{pmatrix} B + \lambda I & \nabla h(x) \\ \nabla h(x)^T & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta v \end{pmatrix} = - \begin{pmatrix} \nabla x L \\ \alpha(\lambda) h(x) \end{pmatrix}. \tag{3.4}$$

When the radius is large enough, $\lambda = 0$ and the full quasi-Newton step is taken. We can prove that in the neighborhood of the solution $\lambda = 0$ in each iteration, resulting in two-step superlinear convergence. When r is very small, $\lambda >> 0$, $v_+ \approx \tilde{v}_+ \equiv -[\nabla h(x)^T \nabla h(x)]^{-1} \nabla h(x)^T \nabla f(x)$ and $\Delta x \approx -\frac{1}{\lambda} \nabla_x L(x, \tilde{v}_+)$.

This algorithm was introduced in Vardi [7]. Proofs of global and local convergence are provided there together with numerical results for several test problems.

4. THE TRUST REGION MODEL FOR THE MINIMAX PROBLEM

We will apply the trust region strategy on the system (2.10). Let mm_1 be the number of active functions in a given iteration. We assume in this section that $\begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix}$ is of full rank. (The case where $\begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix}$ is rank deficient will be discussed in section 5.) Let $Q \begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix} \Pi = \begin{pmatrix} T \\ 0 \end{pmatrix}$ represent a Q-R decomposition of $\begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix}$ where Q is an orthogonal matrix, Π a permutation matrix and T upper triangular. Let $Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$ where Q_1 has

mm_1 rows. The new system becomes then

$$\begin{pmatrix} C + \lambda I & \begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix} \\ \begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix}^T & 0 \end{pmatrix} \begin{pmatrix} \begin{pmatrix} \Delta x \\ \Delta t \end{pmatrix} \\ \Delta v \end{pmatrix} = - \begin{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix} v \\ \alpha(\lambda)\ (F_1 - te) \end{pmatrix} , \qquad (4.1)$$

where

$$\alpha(\lambda) = \min \left\{ 1, \frac{\max\{1, z\}}{\lambda^2} \right\}$$

where

$$z = \frac{\| Q_2 C Q_1^T T^{-T} \Pi^T (F_1 - te) \| \ \| Q_2 \binom{0}{1} \|}{\| T^{-T} \Pi^T (F_1 - te) \|^2} .$$

(When $mm_1 = n+1$ we always have $Q_2\binom{0}{1} = 0$; see remark at Theorem 4.2.) In (4.1), $\lambda = 0$ if the radius is large enough so that the norm of $\binom{\Delta x}{\Delta t}$ (computed for $\lambda = 0$) is less than r; otherwise $\lambda > 0$ is determined so that $\|\binom{\Delta x}{\Delta t}\| = r$. The next theorem summarizes the characteristics of $\binom{\Delta x(\lambda)}{\Delta t(\lambda)}$, $\Delta v(\lambda)$ as a function of λ.

Theorem 4.1. Consider

$$\binom{\Delta x(\lambda)}{\Delta t(\lambda)}, \ \Delta v(\lambda)$$

as defined in (4.1). Then

$$\left\| \binom{\Delta x(\lambda)}{\Delta t(\lambda)} \right\|$$

is monotonically decreasing to zero as a function of λ. When $\lambda \to \infty$,

$$v_+ \to \tilde{v}_+ \equiv \left[\begin{pmatrix}\nabla F_1\\ -e^T\end{pmatrix}^T \begin{pmatrix}\nabla F_1\\ -e^T\end{pmatrix}\right]^{-1} e,$$

and

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

Also when

$$Q_2\binom{0}{1} = 0 \quad \text{and} \quad \lambda \to \infty,$$

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

<u>Proof</u>. The proof of monotonicity of $\left\|\begin{matrix}\Delta x(\lambda)\\ \Delta t(\lambda)\end{matrix}\right\|$ is simple but long and is omitted here. For a proof see Vardi [7].

From (4.1) we have

$$(C + \lambda I)\begin{pmatrix}\Delta x\\ \Delta t\end{pmatrix} + \begin{pmatrix}\nabla F_1\\ -e^T\end{pmatrix} v_+ + \begin{pmatrix}0\\ 1\end{pmatrix} = 0, \tag{4.2}$$

$$\begin{pmatrix}\nabla F_1\\ -e^T\end{pmatrix}^T \begin{pmatrix}\Delta x\\ \Delta t\end{pmatrix} + \alpha(\lambda)(F_1 - te) = 0. \tag{4.3}$$

From (4.2) we have

$$\begin{pmatrix}\Delta x\\ \Delta t\end{pmatrix} = -\ (C + \lambda\ I)^{-1} \left[\begin{pmatrix}\nabla F_1\\ -e^T\end{pmatrix} v_+ + \begin{pmatrix}0\\ 1\end{pmatrix}\right], \tag{4.4}$$

and from (4.3)

$$v_+ = \left[\begin{pmatrix}\nabla F_1 \\ -e^T\end{pmatrix}^T (C + \lambda\ I)^{-1} \begin{pmatrix}\nabla F_1 \\ -e^T\end{pmatrix}\right]^{-1}\left[\alpha(\lambda)(F_1 - te) - \begin{pmatrix}\nabla F_1 \\ -e^T\end{pmatrix}^T (C + \lambda\ I)^{-1} \begin{pmatrix}0 \\ 1\end{pmatrix}\right]. \tag{4.5}$$

When $\lambda \to \infty$,

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

From (4.2)

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

Note that when

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

(using the definition of Q_2)

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

thus

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

Further, careful analysis reveals that when $Q_2\begin{pmatrix}0 \\ 1\end{pmatrix} = 0$ and $\lambda \to \infty$

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

and thus (using (4.2)

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

Theorem 4.1 points out a nice feature of the algorithm: on the limit, when $\lambda \to \infty$, $\binom{\Delta x}{\Delta t}$ and Δv depend solely on information that comes from active functions while semi-active functions 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 $\max_j f_j(x+\Delta x) < \max_j f_j(x)$. If not, we try a shorter radius. The next theorem confirms that there always exists $r > 0$ small enough (or equivalently, because $\left\|\binom{\Delta x(\lambda)}{\Delta t(\lambda)}\right\|$ is monotonically decreasing, $\lambda > 0$ large enough) such that the step is accepted.

Theorem 4.2. Let

$$\binom{\Delta x(\lambda)}{\Delta t(\lambda)},\ \Delta v(\lambda)$$

be computed as in (4.1) and assume

$$\begin{pmatrix} \binom{0}{1} + \binom{\nabla F_1}{-e^T} \tilde{v}_+ \\ F_1 - t\,e \end{pmatrix} \neq 0.$$

Also assume that if

$$Q_2\binom{0}{1} = 0 \quad \text{then} \quad (\tilde{v}_+)_i > 0$$

for all $i \in I$ such that $f_i(x) < \max_j f_j(x)$. Then there exists $\lambda > 0$ large enough such that

$$\max_j f_j\bigl(x + \Delta x(\lambda)\bigr) < \max_j f_j(x) .$$

Remark. When rank $\begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix} = n + 1$, Q_2 does not exist. However from now on we will assume that the case where $Q_2\binom{0}{1} = 0$ includes the case where rank $\begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix} = n + 1$.

Proof. From Taylor's theorem we have

$$\lambda f_i(x + \Delta x) - \lambda f_i(x) = \lambda \nabla f_i(x)^T \Delta x + \lambda \Delta x^T \nabla^2 f_i(\eta)\Delta x \quad \text{for all } i \in (I).$$

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 $Q_2\binom{0}{1} \neq 0$

$$\nabla F_1(x)^T \nabla F_1(x)\tilde{v}_+ = [\nabla F_1(x)^T \nabla F_1(x) \pm ee^T][\nabla F_1(x)^T \nabla F_1(x) + ee^T]^{-1} e$$

$$= \left(1 - e^T[\nabla F_1(x)^T \nabla F_1(x) + ee^T]^{-1} e\right) \underset{\neq}{>} 0.$$

Thus for λ large enough, for all $i \in I$, $f_i(x + \Delta x) < f_i(x)$.

$Q_2\binom{0}{1} = 0$ implies (see Theorem 4.1) that $\begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix}\tilde{v}_+ + \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 0$. In this case we assumed that $(F_1 - t\ e) \neq 0$ and $(\tilde{v}_+)_i > 0$ for all i such that $f_i(x) < \max_j f_j(x)$.

In Theorem 4.1 we have shown that

$$\lambda^2 \Delta t \to -\max\{1,z\}e^T\left[\begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix}^T \begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix}\right]^{-1} (F_1 - te) = -\max\{1,z\}\tilde{v}_+^T(F_1 - te) < 0.$$

Thus for λ large enough $\Delta t < 0$.

$$[f_i(x+\Delta x) - (t+\Delta t)] = [f_i(x)-t] + \begin{pmatrix}\nabla f_i\\ -e^T\end{pmatrix}^T \begin{pmatrix}\Delta x\\ \Delta t\end{pmatrix} + \frac{1}{2}\begin{pmatrix}\Delta x\\ \Delta t\end{pmatrix}^T \begin{pmatrix}\nabla^2 f_i(\eta) & 0\\ 0 & 0\end{pmatrix} \begin{pmatrix}\Delta x\\ \Delta t\end{pmatrix},$$

for all $i \in (II)$. When $\lambda \to \infty$,

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

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

Thus for λ large enough

$$F_1(x + \Delta x) < F_1(x) + \Delta te < F_1(x).$$

This completes the proof.

We can now present the algorithm:

<u>Step 1</u> Start with x^0, $(II) = \phi$, $(I = \{i: f_i(x^0) = \max_j f_j(x^0)\}$, r^0, B^0, $k = -1$.

<u>Step 2</u> Let $k = k+1$, RADINC = 0.

<u>Step 3</u> Take a Q-R decomposition of $\begin{pmatrix}\nabla F_1(x^k)\\ -e^T\end{pmatrix}$ and check whether $Q_2\binom{0}{1} = 0$. If so, compute $\tilde{v}_+$. If for all $i \in I$ such that $f_i(x) < \max_j f_j(x)$ $(\tilde{v}_+)_i > 0$ (or if $Q_2\binom{0}{1} \neq 0$) continue to Step 4.

Otherwise remove from set I the function i_0 such that $f_{i_0}(x) <$ $\max f_j(x)$ and with the minimal $(\tilde{v}_+)_i$, and insert it in set II. Restart step 3.

Step 4 Compute C^k as in (2.10).

Step 5 Find $\lambda^k, \begin{pmatrix} \Delta x^k \\ \Delta t^k \end{pmatrix}, \Delta v^k$ that solve the system (4.1) with the requirements: $\left\| \begin{matrix} \Delta x^k \\ \Delta t^k \end{matrix} \right\| \leq r^k$, $\lambda^k \geq 0$, $\lambda^k \left(\left\| \begin{matrix} \Delta x^k \\ \Delta t^k \end{matrix} \right\| - r^k \right) = 0$. (See Section 5 for computational details.)

Step 6 Check if $\max_j f_j(x^k + \Delta x^k) < \max_j f_j(x^k)$. If so continue to Step 7. If not, if RADINC ≤ 0 half r^k, RADINC = RADINC-1 and go to Step 5. (If RADINC > 0 the step corresponding to the smaller radius has been computed already. Retrieve it and continue to Step 7.)

Step 7 If no new active functions are introduced by the last step and if RADINC ≥ 0 and if $\lambda^k > 0$ let RADINC = RADINC + 1, store the current step, double r^k and go to Step 4. Otherwise continue to step 8.

Step 8 Check for convergence.

Step 9 Compute u^{k+1} by (2.9) and v^{k+1} by (4.5) and according to the signs update (I) and (II). (See Section 2). Compute B^{k+1} (details in section 5).

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

Theorem 4.2 guarantees that in every iteration there will be r^k small enough for which $\max_j f_j(x^{k+1}) < \max_j f_j(x^k)$. We have to justify here the removal of an active constraint as described in Step 3 when $(\tilde{v}_+)_i < 0$ and $Q_2\binom{0}{1} = 0$ (i.e. $mm_1 = n+1$. For a discussion on the possiblility of $Q_2\binom{0}{1} = 0$ when $\begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix}$ is rank deficient see Section 5.)

The necessary conditions at the solution x^* require that

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} \nabla F_1(x^*) \\ -e^T \end{pmatrix} v^* = 0 \quad \text{and} \quad v^* \geq 0.$$

Since $mm_1 = n+1$

$$v^* = -\begin{pmatrix} \nabla F_1(x^*) \\ -e^T \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \tilde{v}_+$$

(because $F_1(x^*) = t^*$). Thus $Q_2\binom{0}{1} = 0$ and $(\tilde{v}_+)_i < 0$ implies that we are converging to a point with a negative multiplier and we therefore need to remove that function from the active list.

5. IMPLEMENTATION

A scaling problem In order to prevent a scaling problem that may be created in $\begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix}$ when $\|\nabla F_1\|$ is much greater or much less than one, it is important to actually replace (P2) by

(P2′)

$$\min \; c\,t$$
$$\text{subject to} \quad f_i(x) - c\,t \leq 0 \qquad i = 1,\cdots,m$$

Now instead of working with the matrix $\begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix}$ we work with the matrix $\begin{pmatrix} \nabla F_1 \\ -ce^T \end{pmatrix}$. In order for this matrix to be well scaled, we update c to $c = \|\nabla F_1(x)\|$ frequently. This modification can be accomplished without difficulties. (We will continue to assume $c = 1$ in the rest of the paper for clarity.)

Initialization of (I) and (II). As ´Step 1´ in the algorithm indicates, at the first iteration usually only one function is active and all the rest are nonactive. We have tried other options where more of the functions were active and semi-active in the first iterations but observed that this strategy does not result in a decrease in the number of iterations. Since these other options required in total more gradient evaluations we decided to use initially one active function.

Solving System (4.1). Using the Q-R decomposition of the matrix $\begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix}$,

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

we can obtain the following expressions:

From (4.3)

$$Q_1 \begin{pmatrix} \Delta x \\ \Delta t \end{pmatrix} = -\alpha(\lambda) T^{-T} \Pi^T (F_1 - t\ e) . \tag{5.1}$$

From (4.2)

$$Q_2 \begin{pmatrix} \Delta x \\ \Delta t \end{pmatrix} = -\left(Q_2 C Q_2^T + \lambda\ I\right)^{-1} Q_2 \left[\begin{pmatrix} 0 \\ 1 \end{pmatrix} + C Q_1^T \left(Q_1 \begin{pmatrix} \Delta x \\ \Delta t \end{pmatrix}\right)\right] \tag{5.2}$$

$$\begin{pmatrix}\Delta x\\ \Delta t\end{pmatrix} = Q_1^T Q_1\begin{pmatrix}\Delta x\\ \Delta t\end{pmatrix} + Q_2^T Q_2\begin{pmatrix}\Delta x\\ \Delta t\end{pmatrix} . \tag{5.3}$$

A Cholesky decomposition of C is performed, $C = LL^T$. (When $mm_2 = 0$ and $C = \begin{bmatrix} B & 0 \\ 0 & 0 \end{bmatrix}$, $L = \begin{bmatrix} L_1 & 0 \\ 0 & 0 \end{bmatrix}$ 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, (P orthogonal, Σ permutation, R upper triangular). Partition P into $P = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$ where P_1 has $n+1-mm_1$ rows. We now have $(Q_2 C Q_2^T + \lambda I) = \Sigma(R^T R + \lambda I)\Sigma^T$ and $Q_2\binom{\Delta x}{\Delta t}$ can be obtained by solving

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

where

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

In order to solve this linear least squares problem, a Q-R decomposition of $\begin{bmatrix} R \\ \lambda^{1/2} I \end{bmatrix}$ is obtained with the use of Givens transformation.

Define $\phi(\lambda) = \|\binom{\Delta x}{\Delta t}\| - r$. For a given λ we can use the above decomposition to compute $\phi(\lambda)$ and $\phi'(\lambda)$. A λ iterative scheme is used to obtain λ such that $\phi(\lambda) = 0$. In this λ-iterative process we usually have a good guess for λ and we take $\lambda^{j+1} = \lambda^j - \frac{\phi(\lambda^j) + r}{r} \frac{\phi(\lambda^j)}{\phi'(\lambda^j)}$. Known upper and lower bounds on the solution are used to help the convergence. It takes an average of less than two λ-iterations to obtain an acceptable solution, (i.e. λ such that $\phi(\lambda) < 0.1 * r$) . For more details on this part of the algorithm see Vardi [7].

Updating B. When moving from the k^{th} iteration to the $(k+1)^{th}$ iteration,

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

has to be updated into

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

Define

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

If $\begin{pmatrix} \Delta x^k \\ \Delta t^k \end{pmatrix}$ is small, it is reasonable to require that $B^{k+1}\Delta x^k = y^k$. Other requirements on B^{k+1} are that it is symmetric, positive definite, close to B^k in some norm and easy to compute. The B.F.G.S. update is often used in which

$$B^{k+1} = \mathrm{BFGS}(B^k, \Delta x^k, u^k) = B^k + \frac{y^k \, y^{kT}}{y^{kT} \, \Delta x^k} - \frac{B^k \, \Delta x^k \, \Delta x^{kT} \, B^k}{\Delta x^{kT} \, B^k \, \Delta x^k} .$$

If B^k is positive definite and $y^{kT} \Delta x^k > 0$, B^{k+1} is positive definite. In our algorithm we use a modification of the BFGS that was suggested by Powell (see Powell [5]):

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

where

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

and

$$\theta = \begin{cases} 1 & \text{if} \quad y^k \Delta x^k > 0.2\ \Delta^{kT} B^k \Delta x^k \\ \dfrac{0.8\ \Delta x^{kT} B^k \Delta x^k}{\Delta x^{kT} B^k \Delta x^k - y^{kT} \Delta x^k} & \text{otherwise} \end{cases}$$

Powell proved that when this update is used in an algorithm for constrained minimization, local two-step superlinear convergence results.

When $\binom{\nabla F_1}{-e^T}$ is rank deficient

When the Q-R decomposition is performed to obtain $Q\binom{\nabla F_1}{-e^T}\Pi = \binom{T}{0}$, we check at each step whether any of the remaining columns have norm larger than (machine ps) $\times T_{11}$. If not, we stop. If ℓ steps were performed, and T is an $\ell \times mm_1$ upper triangular matrix, we consider rank $\binom{\nabla F_1}{-e^T} = \ell$. Partition $Q = \binom{Q_1}{Q_2}$, $\Pi = \Pi_1, \Pi_2$ and $\bar{T} = [T,S]$ where Q_1 has ℓ rows, Π_1 and T have ℓ columns. (5.1) becomes $Q_1\binom{\Delta x}{\Delta t} = -\alpha(\lambda)T^{-T}\Pi_2^T(F_1 - t\ e)$ and all other expressions remain the same. The effect of this is that the active functions that correspond to the remaining columns in $\binom{\nabla F_1}{-e^T}$ are ignored.

When $\binom{\nabla F_1}{-e^T}$ is rank deficient it is possible that $Q_2\binom{0}{1} = 0$ and as the algorithm indicates $\tilde{v}_+$ is checked and active functions with negative $(\tilde{v}_+)_i$ may become semiactive.

6. NUMERICAL TESTS

A computer program based on the algorithm was written and tested on 6 test problems from the optimization literature as well as one large practical problem.

The complete information on these problems is given in this section. In order to check global convergence we added starting points that are much further from the known solution than the suggested starting points. In all problems global convergence was achieved and fast local convergence was observed.

For all problems we used $r^0 = 1$, $B^0 = I$. The stopping criteria were

1. $$\|\Delta x\| < 10^{-10}(\|x\| + 1),$$

2. $$\left\| \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} \nabla F_1 \\ -e^T \end{pmatrix} v_+ \right\| < 10^{-6},$$

and

$$\max_{i \varepsilon I} f_i(x) - \min_{i \varepsilon I} f_i(x) < 10^{-6} \max\{|\max_{i \varepsilon I} f_i(x)|, |\min_{i \varepsilon I} f_i(x)|\}.$$

In each of the problems we tried 3 starting points. We recorded the number of function evaluations and the number of individual gradient evaluations (i.e., gradients of functions in (I) and (II); if, for example, thre are in a certain iterations mm_1 active functions and mm_2 semi-active functions, we count $mm_1 + mm_2$ gradient evaluations.)

We display the count for all iterations at the end of which a change in (I) or (II) occurred.

Test Problem 1

$n = 2,\ m = 3$ $\qquad F_1(x) = x_1^2 + x_2^4$

$$F_2(x) = (2 - x_1)^2 + (2 - x_2)^2$$

$$F_3(x) = 2\ \exp(-x_1 + x_2)$$

solution $x = (1.139037652,\ .8995599384)$

$f_1 = F_2 = 1.952224494$

Starting Point	Iteration No.	No of. Fun. Evaluation	No. of Ind. Grad. Evaluation	max $f_j(x)$	Semi-Active Functions	Active Functions
(1,-.1)	0	1	1	5.41	-	2
	2	6	3	2.170849382	-	2
	7	12	13	1.952224494	-	1,2
(10,-1)	0	1	1	101	-	1
	1	6	2	2.335229368	-	1
	7	12	14	1.952224494	-	1,2
(100,-10)	0	1	1	20000.	-	1
	3	14	4	40.68514215	-	1
	5	17	8	2.501544410	3	1
	7	19	12	2.010184030	-	1,3
	8	20	15	1.964914927	3	1,2
	12	24	23	1.952224494	-	1,2

Test Problem 2

n = 2, m = 3

$F_1 = x_1^4 + x_2^2$

F_2, F_3, as in example 1

$F_1 = F_2 = F_3 = 2$

solution x = (1,1)

Starting Point	Iteration No.	No. of Fun. Evaluation	No. of Ind. Grad. Evaluation	max $f_j(x)$	Semi Active Functions	Active Functions
(1,-.1)	0	1	1	5.41	-	2
	1	2	2	3.181388755	-	2
	2	4	4	2.417247765	-	1,2
	4	6	10	2.004795643	3	1,2
	6	9	16	2.000000000	-	1,2,3
(10,-1)	0	1	1	10001.	-	1
	1	6	2	19.74943380	-	1
	2	8	4	19.36287258	-	1,2
	6	14	16	2.808076541	1	2,3
	10	18	28	2.000000000	-	1,2,3
(100,-10)	0	1	1	100000100.	-	1
	10	20	11	4.436401306	-	1
	13	26	17	2.076370586	-	1,2
	15	31	23	2.000020450	3	1,2
	17	33	29	2.000000000	-	1,2,3

Test Problem 3

n = 4, m = 4

$$F_1(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$

$$F_2(x) = F_1(x) - 10(-x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8)$$

$$F_3(x) = F_1(x) - 10(-x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10)$$

$$F_4(x) = F_1(x) - 10(-2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5)$$

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

Starting Point	Iteration No.	No. of Fun. Evaluation	No. of Ind. Grad. Evaluation	max f (x)	Semi Active Functions	Active Functions
(0,0,0,0)	0	1	1	0	–	1
	2	6	3	-39.35982740	–	1
	3	8	5	-41.13057951	–	1,4
	5	11	11	-41.67511609	2	1,4
	10	16	26	-44.00000000	–	1,2,4
(10,10,10,10)	0	1	1	5960	–	3
	1	6	2	973.8877654	–	3
	2	8	4	91.01083219	–	2,3
	4	12	8	-13.00041510	2	3
	7	20	20	-41.13646654	1,2,3	4
	9	24	26	-43.41889604	2	1,4
	15	31	44	-44.00000000	–	1,2,4
(100,100,100,100)						
	0	1	1	645500.	–	3
	4	18	5	344.6890778	–	3
	8	23	13	-22.66648305	–	3,4
	9	24	16	-27.19599155	–	1,3,4
	10	25	20	-39.22987562	3	1,2,4
	18	34	44	-44.00000000	–	1,2,4

Test Problem 4

$n = 2,\ m = 3$

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

Solution $x = \pm(.4532962370,\ -.9065924741)$

$F_1 = f_3 = .6164324356$

Starting Point	Iteration No.	No. of Fun. Evaluation	No. of Ind. Grad. Evaluation	max f (x)	Semi Active Functions	Active Functions
(3,1)	0	1	1	13.	-	1
	1	4	2	.8682565665	-	1
	2	5	4	.8053242777	-	1,3
	4	8	10	.7272455047	3	1,2
	5	10	13	.6428900754	-	1,2,3
	10	15	28	.6164324356	2	1,3
(30,10)	0	1	1	1300	-	1
	3	11	4	.9923360793	-	1
	12	23	22	.6164324356	-	1,3
(300,100)	0	1	1	130000	-	1
	4	15	5	.9995701115	-	1
	11	24	19	.6164324356	-	1,3

Test Problem 5

$n = 3,\ m = 6$

$F_1(x) = x_1^2 + x_2^2 + x_3^2 - 1$

$F_2(x) = x_1^2 + x_2^2 + (x_3-2)^2$

$F_3(x) = x_1 + x_2 + x_3 - 1$

$F_4(x) = x_1 + x_2 - x_3 + 1$

$F_5(x) = 2x_1^3 + 6x_2^2 + 2(5x_3-x_1+1)^2$

$F_6(x) = x_1^2 - 9x_3$

Solution $x = (.32825995,\ 0,\ .1313200636)$

$F_2 = F_5 = 3.599719300$

Starting Point	Iteration No.	No. of Fun. Evaluation	No. of Ind. Grad. Evaluation	max f (x)	Semi Active Functions	Active Functions
(1,1,1)	0	1	1	58.	-	5
	2	4	3	5.028346958	-	5
	13	26	25	3.599719300	-	2,5
(10,10,10)	0	1	1	5962.	-	5
	5	14	6	4.448679576	-	5
	18	43	32	3.599719300	-	2,5
(100,100,100)	0	1	1	2381602.	-	5
	1	10	2	21172.51683	-	5
	2	12	4	13296.67178	-	1,5
	3	13	6	7506.056041	5	1
	4	14	7	1785.905159	-	1
	5	15	9	848.2530772	-	1,2
	6	16	11	324.5466404	1	2
	7	17	12	213.7819513	-	2
	15	25	28	3.599719300	-	2,5

Test Problem 6

n = 3, m = 30

$$F_j(x) = -y_j + x_1 + \frac{u_j}{v_j x_2 + w_j x_3} \qquad j = 1,\cdots,15$$

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

where

$$u_j = j,\ v_j = 16 - j \quad \text{and} \quad w_j = \min\{u_j, v_j\},$$

$$\vec{y} = (.14,.18,.22,.25,.32,.39,.37,.58,.73,.96,1.34,2.1,4.39)$$

Solutions:

$x = \pm(.05346938776,\ t,\ 3.5 - t)$ $\qquad .5 \leq t \leq 1.5$

$F_9 = F_{23} = F_{30} = .05081632653$ $\qquad$ (or $F_{24} = F_8 = F_{15}$)

$x = \pm(.2831587485,\ -4.8412419079,\ 9.323361111)$

$F_6 = F_{20} = F_{30} = .7602099910$ $\qquad$ (or $F_{21} = F_5 = F_{15}$)

$x = \pm(.02033344564,\ .100795522614,\ 3.3992044739)$

$F_1 = F_2 = F_{23} = F_{30} = .08395226864$ $\qquad$ (or $F_{16} = F_{17} = F_8 = F_{15}$)

Test Problem 6 (continued)

Starting Point	Iteration No.	No. of Fun. Evaluation	No. of Ind. Grad. Evaluation	max f (x)	Semi Active Functions	Active Functions
(1,1,1)	0	1	1	4.11	-	15
	1	3	2	0.7422952056	-	15
	2	4	4	0.3286634371	-	9,15
	3	5	7	0.2339086843	15	9,18
	5	7	15	0.05443554245	15	9,18,30
	7	11	21	0.05115868122	-	9,18,30
	8	12	25	0.05081633724	-	9,18,23,30
	9	13	29	0.05081632653	18	9,23,30
(10,10,10)	0	1	1	9.86625	-	1
	3	8	4	2.417983420	-	1
	9	21	16	1.207116039	-	1,30
	10	23	19	0.8923142963	15	1,30
	11	24	22	0.1098360412	-	1,15,30
	12	27	26	0.08548202468	-	1,9,15,30
	13	29	30	0.07228591795	1,15	9,30
	16	34	39	0.05081632653	-	9,23,30
(100,100,100)						
	0	1	1	99.860625	-	1
	5	13	6	11.35666692	-	1
	16	42	28	1.779837231	-	1,30
	17	43	31	0.6808764481	-	1,9,30
	18	44	34	0.3756488583	1	9,30
	19	47	36	0.1292719191	-	9,30
	20	49	39	0.07580649840	15	9,30
	23	55	48	0.05392409950	-	9,18,30
	24	56	52	0.05081719777	-	9,18,23,30
	25	57	56	0.05081632653	18	9,23,30

The algorithm has also undergone limited testing on the problem of designing an aircraft lateral stability augmentation system found in [9] identified as the deterministic problem at Mach 2.5. By using different starting points it both found the minimum reported in Schy et al. [1981] and located another local minimum. Execution times correspond favorably with those of the algorithm used in Schy et al. [1981]. In the only case where problems formulations (starting points, convergence criteria, etc.) were similar enough to make comparisons valid, the present algorithm executed in less than 1/4 of the time (Giesy [1982]). In these testings convergence to a local minimum and fast local convergence were always achieved.

REFERENCES

[1] M. AVRIEL. Nonlinear Programming: Analysis and Methods. Prentice-Hall, Inc., 1976.

[2] C. CHARALAMBOUS and A. R. CONN. An official method to solve the minimax problem directly. SIAM J. Numer. Anal., 15, No. 1 (1978), pp. 162-187.

[3] S. P. HAN. A globally convergent method for nonlinear program. J. Optim. Theory Appl., 22, No. 3 (1977), pp. 297-309.

[4] W. MURRAY and M. L. OVERTON. A projected Lagrangian algorithm for nonlinear minimax optimization. System Optimization Laboratory Technical Report No. 79-21, 1979.

[5] M. J. D. POWELL. The convergence of variable metric methods for nonlinearly constrained optimization calculations; NLP3; Meyer and Robinson, eds., Academic Press, 1978.

[6] R. A. TAPIA. On the role of slack variables in quasi-Newton methods for constrained optimization. Numerical Optimization of Dynamic Systems, L. C. W. Dixon and G. P. Szego, eds., 1980, pp. 235-246.

[7] A. VARDI. A trust region algorithm for equality constrained minimization: convergence properties and implementation. ICASE Report No. 81-9, February 1981, to appear in SIAM J. Numer. Anal., 1985.

[8] A. VARDI. A trust region algorithm for unconstrained minimization: convergence properties and implementation. ICASE Report No. 80-35, 1980.

[9] A. A. SCHY and D. P. GIESY, Multiobjective insensitive design of airplane control systems with uncertain parameters, AIAA Guidance and Control Conference, Albuquerque, New Mexico, August 19-21, 1981.

1. Report No. NASA CR-172382 ICASE Report No. 84-25	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A New Minmax Algorithm		5. Report Date June 1984
		6. Performing Organization Code
7. Author(s) Avi Vardi		8. Performing Organization Report No. 84-25
9. Performing Organization Name and Address Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665		10. Work Unit No.
		11. Contract or Grant No. NAS1-15810
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		13. Type of Report and Period Covered Contractor Report
		14. Sponsoring Agency Code 505-31-83-01

15. Supplementary Notes

Langley Technical Monitor: Robert H. Tolson
Final Report

16. Abstract

The paper deals with the minimax problem $\min_{x \in \mathbb{R}^n} \max_{i=1,\cdots,m} f_i(x)$. We work with its equivalent representation $\min t$ s.t. $f_i(x) - t \leq 0$ for all i. For this problem we design a new active set strategy in which there are three types of functions: active, semi-active, and non-active. This technique will help in preventing zigzagging which often occurs when an active set strategy is used. Some of the inequality constraints are handled with slack variables. Also a trust region strategy is used in which at each iteration there is a sphere around the current point in which the local approximation of the function is trusted. The algorithm suggested in the paper was implemented into a successful computer program. Numerical results are provided.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
minimization constraints	64 Numerical Analysis 61 Computer Programming & Software Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	37	A03