An Efficient Algorithm for Nonlinear Minimax Problems

Andrew R. Conn*
Yuying Li**

TR 90-1104
March 1990

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

---

*Dept. of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, N2L 3G1 CANADA. The research of this author was supported in part by NSERC grant A8639.
**Computer Science Dept., Cornell University, Upson Hall, Ithaca, NY 14853. The research of this author was partially supported by the U.S. Army Research Office through the Mathematical Science Institute, Cornell University.

AN EFFICIENT ALGORITHM FOR NONLINEAR MINIMAX PROBLEMS

ANDREW R. CONN* AND YUYING LI†

Abstract. We present a new method for solving a nonlinear minimax problem. This new algorithm exploits the structure and characterisation of the solution whenever possible. The exploitation is based on the results that have been established in [13]. The algorithm is globally convergent with a superlinear convergence rate. Numerical results indicate the efficacy of the new method.

Key Words. nonlinear Chebyshev approximation

AMS(MOS) subject classification. 41A50, 65D99, 65F20, 65K05

1. Introduction. We want to solve a *discrete nonlinear minimax problem* which is written as

$$\min_{x\in\Re^n} \max_{i\in M} f_i(x), \tag{1.1}$$

where M is a finite index set. We seek to find the minimum value for the *maximum function*

$$\psi(x) = \max_{i\in M} f_i(x).$$

For simplicity, we describe in detail our algorithm in terms of the discrete Chebyshev problem, which is a major class of discrete minimax problems. The extensions required for the problem (1.1) are mentioned. It is clear that a discrete Chebyshev problem,

$$\min_{x\in\Re^n} \max_{1\le i\le m} |f_i(x)|, \tag{1.2}$$

could be regarded as a special case of a general minimax problem (1.1) with

$$M = \{1, 2, \cdots, m, m+1, \cdots, 2m\},$$

$$f_{i+m}(x) = -f_i(x), \qquad i = 1, \cdots, m.$$

In this paper, we are content to find a local minimum of (1.1) and we assume that a local minimum for (1.1) always exists. We also assume that each $f_i(x)$ is twice continuously differentiable.

Numerical methods for the discrete nonlinear Chebyshev/minimax problem are less prolific than for the linear problem. It is well-known that the maximum function, $\psi(x) = \max_{i\in M} f_i(x)$, is not differentiable at kinks that arise whenever $f_i(x) = f_j(x)$, $i, j \in M$, $i \neq j$. Therefore, traditional gradient type methods cannot be applied directly.

As is mentioned in [13] in more detail, the existing methods are essentially based on successive linear programming or nonlinear programming techniques applied to an

* Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada. The research of this author was supported in part by NSERC grant A8639.

† Computer Science Department, Cornell University, Upson Hall, Ithaca, NY, 14853. The research of this author was partially supported by the U.S. Army Research Office through the Mathematical Science Institute, Cornell University.

equivalent nonlinear programming problem. Examples include [1], [12], [17], [19], [20], [21], [24], [26], [28], [32] and [33].

The classical Chebyshev theory provides us with the characterisations for the best linear Chebyshev approximation. These properties uniquely determine a solution in many instances and thus requiring approximations with these special features is likely to result in a more efficient technique. Indeed, such has been the experience with classical Remez algorithms for best continuous/discrete linear Chebyshev approximation (see for example, [31]).

In [13], we have established the characterisation for a solution of a *nonlinear minimax problem.* These characterisations are generalizations of those of the best linear Chebyshev approximation. Hence, forcing the approximate solutions of a nonlinear minimax problem to have these properties will expedite the solution finding process, in particular for those problems arising from the discretisation of continuous approximation problems.

The new approach proposed in this paper is different from the existing methods in that the structure and characterisation of a solution of the minimax problem has been used explicitly.

2. The Model Algorithm. In [13], we have given a characterisation for a solution of a nonlinear minimax problem. We briefly summarise the main result here and the interested reader is recommended to see [13] for the motivation and details.

DEFINITION 2.1. *The vector set* $C = \{\nabla f_{i_j}\}_{j=0}^{l}$ *is called a* **cadre** *if and only if:*

1. $rank([\nabla f_{i_0}, \cdots, \nabla f_{i_l}]) = l$;
2. *for any* $\{\nabla f_{j_1}, \cdots, \nabla f_{j_l}\} \subset C$, $rank([\nabla f_{j_1}, \cdots, \nabla f_{j_l}]) = l$.

It is easy to prove the following.

LEMMA 2.2. *(Lemma 20 in [13])* $C = \{\nabla f_{i_j}\}_{j=0}^{l}$ *is a cadre if and only if*

1. $rank(C) = l$, *and*
2. *there exist multipliers* $\{\lambda_i\}$ *such that*

$$\sum_{j=0}^{l} \lambda_j \nabla f_{i_j} = 0 \quad \textit{and} \quad \lambda_j \neq 0, \qquad j = 0, \cdots, l.$$

We refer to $\{\lambda_j\}$, normalised by $\sum_{j=0}^{l} \lambda_j = 1$, if $\sum_{j=0}^{l} \lambda_j \neq 0$ and $\lambda_0 = 1$ otherwise, as *cadre multipliers.*

DEFINITION 2.3. *The functions* $\{f_{i_j}(x)\}_{j=0}^{l}$ *are said to be locally forming a* **reference set** *of a minimax problem (1.1) if* $C = \{\nabla f_{i_j}\}_{j=0}^{l}$ *is a cadre such that*

1. *The cadre multipliers* $\{\lambda_j\}_{j=0}^{l}$ *satisfy* $\lambda_j > 0$, $j = 0, \cdots, l$;
2. *The functions* $\{f_{i_j}(x)\}_{j=0}^{l}$ *all have the same sign.*

The reference set is further called a **levelled reference set** *if the value of each function is the same, viz.,*

$$f_{i_j}(x) = f_{i_k}(x), \qquad \textit{for any } i_j, i_k \in C.$$

A local minimum of (1.2) can be characterised as follows.

THEOREM 2.4. *(Theorem 31 in [13])*

Suppose x^* is a local minimum for a minimax problem (1.1). Then, there exists a set of $l+1$ functions $\{f_{i_j}(x)\}_{j=0}^{l}$ which is a levelled reference set at x^* on the cadre $C = \{\nabla f_{i_j}(x^*)\}_{j=0}^{l}$ with the maximum deviation.

A reference set is a generalisation of the alternating sign property of a best Chebyshev approximation [13]. Our experience with the numerical methods for linear l_∞ problems [5] has shown that it is very important to computationally exploit the above properties of a solution. The algorithm proposed in this paper is developed using this principle.

The proposed algorithm is a descent method with a line search. The special features of the suggested algorithm, however, are that the search directions always decrease the maximum function and attempt to enforce the characterisation of a solution at the same time. Since a levelled reference set with the maximum deviation characterises a solution to a minimax problem, we attempt to compute the solution by constructing approximate solutions with such properties.

Assume $\mathcal{W} = \{i_0, i_1, \cdots, i_l\}$ is an index set and all the functions in $\mathcal{W}$ form a reference set, not levelled. Denote

$$A = [\nabla f_{i_0} - \nabla f_{i_1}, \cdots, \nabla f_{i_0} - \nabla f_{i_l}],$$

$$\Phi(x)^T = [f_{i_0}(x) - f_{i_1}(x), f_{i_0}(x) - f_{i_2}(x), \cdots f_{i_0}(x) - f_{i_l}(x)],$$

and $i_0 \in \mathcal{A}(x, 0)$. Here, $\mathcal{A}(x, 0)$ denotes the indices of the *active functions* which are the functions achieving the maximum value at the current point x. In other words, $\mathcal{A}(x, 0) = \{i \in M \,|\, \psi(x) = f_i(x)\}$.

In [13], we have shown that it is possible to determine descent directions which attempt to construct a levelled reference set in the neighbourhood of a cadre or reference set. The results are summarised in the following lemmas.

LEMMA 2.5. *Suppose the functions in $\mathcal{W}$ form a reference set which includes all the current active functions. Then, the direction defined from $\mathcal{W}$ by*

$$v = -A(A^T A)^{-1}\Phi(x), \tag{2.1}$$

is a descent direction for all the active functions provided the reference set is not levelled.

If a unit step along v is taken, $\Phi(x) + A^T v = 0$. Thus the functions in $\mathcal{W}$ would all have the same value as the representative function, up to first order.

LEMMA 2.6. *Suppose $C = \{\nabla f_{i_0}, \nabla f_{i_1}, \cdots, \nabla f_{i_l}\}$ is a non-reference-set cadre with cadre multipliers $\{\lambda_j\}_{j=0}^{l}$ summing to one and $f_{i_0}(x)$ achieves the current maximum deviation for (1.1). Then, the direction v defined on $\mathcal{W} = \{i_0, i_1, \cdots, i_l\}$ by*

$$[\nabla f_{i_0} - \sigma_0\sigma_j \nabla f_{i_j}]^T v = -(f_{i_0} - \sigma_0\sigma_j f_{i_j}), \quad i_j \in \mathcal{W}, i_j \neq i_0, \ \sigma_j = sgn(f_{i_j}), \tag{2.2}$$

decreases all the active functions, assuming $\mathcal{W}$ includes all the active functions at x.

We build up cadres using the concept of working sets. As defined in [13], a working set is a function index set which includes all the indices of the current maximum functions. We emphasize, however, that the working set $\mathcal{W}$ is not an active set in general.

The search direction is determined from the working set. If a cadre has not been located, in addition to decreasing the maximum function, the search direction is constructed to level the functions in the working set, when this is possible. The motivation behind this levelling comes from the fact that the structure of the solution requires the error curve to be levelled on the extreme points.

The suggested model algorithm is now outlined.

MODEL ALGORITHM

Step 1 Suppose an initial point x^0 is given. Set $k \leftarrow 0$.

Step 2 [Set Up a Working Set]
The new working set $\mathcal{W}^k$ is determined. Check if there is a cadre $\mathcal{C}^k$ whose indices form a subset of $\mathcal{W}^k$. If there is no such cadre, go to Step 4.

Step 3 [Construct a Levelled Reference Set]
Check reference set conditions. If the cadre corresponds to a reference set, compute a descent direction by levelling the reference set. Otherwise, find a descent direction that attempts to construct a reference set. Go to Step 5.

Step 4 [Descend and Level]
A search direction d^k is found that decreases all the ϵ-active functions and levels the working functions in the working set $\mathcal{W}^k$, if possible.

Step 5 [Line Search]
A line search is performed on $\psi(x)$ along the direction d^k

$$x^{k+1} \leftarrow x^k + \lambda^k d^k; \quad k \leftarrow k+1;$$

Step 6 [Termination]
If optimal, stop. Otherwise, go to Step 2.

The Step 3 of the model algorithm is the major part in which the characterisation of the solution is exploited. In [13], we have indicated that Step 3 is computationally possible. In this paper, we discuss how to construct a working set (§4), how to identify cadres (§3) and how to compute a search direction when there is no cadre (§5). We also present details of the computation, including degeneracy handling (§7). (Following [13], the current point x^k is degenerate if and only if there is a cadre $\mathcal{C} = \{\nabla f_{i_0}, \nabla f_{i_1}, \cdots, \nabla f_{i_l}\}$ such that $\{i_0, i_1, \cdots, i_l\} \subset \mathcal{A}(x^k, 0)$).

3. Identifying Cadres. Given a set of functions $\{f_{i_0}, \cdots, f_{i_l}\}$, we discuss whether there exists a cadre within this set. We divide cadres into two types, depending upon whether

$$\sum_{j=0}^{l} \lambda_j = 1, \quad \text{or} \quad \sum_{j=0}^{l} \lambda_j = 0,$$

where $\{\lambda_j\}_{j=0}^{l}$ are cadre multipliers. The cadre which defines a reference set always belongs to the first type.

It is straightforward to prove the following lemma.

LEMMA 3.1. *Suppose* $\{\nabla f_{i_0} - \nabla f_{i_1}, \cdots, \nabla f_{i_0} - \nabla f_{i_l}\}$ *are linearly independent. Then, the rank of the vector set* $\{\nabla f_{i_0}, \nabla f_{i_1}, \cdots, \nabla f_{i_l}\}$ *is at least* l.

The following lemma gives, under certain assumptions, necessary and sufficient conditions for the existence of a cadre with the sum of cadre multipliers being zero.

LEMMA 3.2. *Suppose* $A = [\nabla f_{i_0} - \nabla f_{i_1}, \cdots, \nabla f_{i_0} - \nabla f_{i_{l-1}}]$ *is of full rank and that* $Z^T \nabla f_{i_0} \neq 0$, *where the columns of* Z *form a basis for the null space of* A^T. *Then, there exists a cadre* $\mathcal{C} \subseteq \{\nabla f_{i_0}, \nabla f_{i_1}, \cdots, \nabla f_{i_l}\}$ *with cadre multipliers summing to zero if and only if* $[\nabla f_{i_0} - \nabla f_{i_1}, \cdots, \nabla f_{i_0} - \nabla f_{i_l}]$ *is rank deficient.*

Proof. Suppose $\mathcal{C} = \{\nabla f_{k_0}, \cdots, \nabla f_{k_\nu}\}$ is a cadre and $\{k_0, k_1, \cdots, k_\nu\} \subseteq \{i_0, i_1, \cdots, i_l\}$ with

$$\sum_{j=0}^{\nu} \lambda_j \nabla f_{k_j} = 0, \quad \sum_{j=0}^{\nu} \lambda_j = 0, \quad \lambda_j \neq 0, \ j = 0, \ldots, \nu.$$

Then it is obvious that

$$\sum_{j=0}^{\nu} \lambda_j (\nabla f_{i_0} - \nabla f_{k_j}) = 0. \tag{3.1}$$

From (3.1) and the assumption that $\{\nabla f_{i_0} - \nabla f_{i_1}, \cdots, \nabla f_{i_0} - \nabla f_{i_{l-1}}\}$ are linearly independent, we know that $i_l \in \{k_0, \cdots, k_\nu\}$. Hence, $\lambda_l \neq 0$ and we have

$$(\nabla f_{i_0} - \nabla f_{i_l}) = \sum_{j=1}^{l-1} \hat{\lambda}_j (\nabla f_{i_0} - \nabla f_{i_j}),$$

after padding with zeros if necessary. On the other hand, if we assume that $\{\nabla f_{i_0} - \nabla f_{i_1}, \cdots, \nabla f_{i_0} - \nabla f_{i_{l-1}}\}$ are linearly independent and $\{\nabla f_{i_0} - \nabla f_{i_1}, \cdots, \nabla f_{i_0} - \nabla f_{i_{l-1}}, \nabla f_{i_0} - \nabla f_{i_l}\}$ are linearly dependent, we have

$$\nabla f_{i_0} - \nabla f_{i_l} = \sum_{j=1}^{l-1} \hat{\lambda}_j (\nabla f_{i_0} - \nabla f_{i_j}). \tag{3.2}$$

From the assumption that A is full-rank and Lemma 3.1, we have that

$$\operatorname{rank}(\{\nabla f_{i_0}, \nabla f_{i_1}, \cdots, \nabla f_{i_{l-1}}\}) \geq l - 1.$$

Moreover, from $Z^T \nabla f_{i_0} \neq 0$, and the argument that follows, we can conclude that

$$\operatorname{rank}(\{\nabla f_{i_0}, \nabla f_{i_1}, \cdots, \nabla f_{i_{l-1}}\}) = l. \tag{3.3}$$

The above is true because, if $\{\nabla f_{i_0}, \nabla f_{i_1}, \cdots, \nabla f_{i_{l-1}}\}$ are linearly dependent, then there exist $\{\lambda_j\}$ which are not all zero such that

$$\sum_{j=0}^{l-1} \lambda_j \nabla f_{i_j} = 0.$$

If $\sum_{j=0}^{l-1} \lambda_j \neq 0$, without loss of generality, we can assume $\sum_{j=0}^{l-1} \lambda_j = 1$. Thus, $\lambda_0 = 1 - \sum_{j=1}^{l-1} \lambda_j$. Hence

$$\nabla f_{i_0} = \sum_{j=1}^{l-1} \lambda_j (\nabla f_{i_0} - \nabla f_{i_j}).$$

We conclude that $Z^T \nabla f_{i_0} = 0$ which is a contradiction.

If $\sum_{j=0}^{l-1} \lambda_j = 0$, we have $\lambda_0 = -\sum_{j=1}^{l-1} \lambda_j$. Hence

$$\sum_{j=1}^{l-1} \lambda_j (\nabla f_{i_0} - \nabla f_{i_j}) = 0$$

which is again a contradiction to the assumption that A is full rank.

Thus, using (3.2), we obtain

$$\sum_{j=0}^{l} \hat{\lambda}_j \nabla f_{i_j} = 0, \quad \text{and} \quad \sum_{j=0}^{l} \hat{\lambda}_j = 0, \tag{3.4}$$

where $\hat{\lambda}_0 = 1 - \sum_{j=1}^{l-1} \hat{\lambda}_j, \hat{\lambda}_l = -1$.

Define $\mathcal{C} = \{ \nabla f_{i_j} \mid \hat{\lambda}_j \neq 0 \ , \ j = 0, \ldots, \nu \}$. Using (3.4),

$$\text{rank}(\mathcal{C}) \leq |\mathcal{C}| - 1.$$

From (3.3), we know that

$$\text{rank}(\mathcal{C}) \geq |\mathcal{C}| - 1.$$

Hence

$$\text{rank}(\mathcal{C}) = |\mathcal{C}| - 1. \tag{3.5}$$

Moreover

$$\hat{\lambda}_j \neq 0, \ \nabla f_{i_j} \in \mathcal{C}, \ \text{with} \sum_{\nabla f_{i_j} \in \mathcal{C}} \hat{\lambda}_j = 0.$$

Using Lemma 2.2, $\mathcal{C}$ is a cadre with the sum of the cadre multipliers being zero. □

Now, we present a lemma which tells us how to identify cadres with cadre multipliers summing to one.

LEMMA 3.3. *Suppose* $\{\nabla f_{i_0} - \nabla f_{i_1}, \cdots, \nabla f_{i_0} - \nabla f_{i_l}\}$ *are linearly independent. Then there exists a cadre* $\mathcal{C} \subseteq \{\nabla f_{i_0}, \nabla f_{i_1}, \cdots, \nabla f_{i_l}\}$ *with cadre multipliers summing to one if and only if the orthogonal projection* $Z^T \nabla f_{i_0}$ *is zero, where*

$$A = [\nabla f_{i_0} - \nabla f_{i_1}, \cdots, \nabla f_{i_0} - \nabla f_{i_l}], \ Z^T A = 0.$$

Proof. Since $\{\nabla f_{i_0} - \nabla f_{i_1}, \cdots, \nabla f_{i_0} - \nabla f_{i_l}\}$ are linearly independent, using Lemma 3.1,

$$\text{rank}(\{\nabla f_{i_0}, \nabla f_{i_1}, \cdots, \nabla f_{i_l}\}) \geq l. \tag{3.6}$$

The orthogonal projection of ∇f_{i_0} on the null space of A^T is $Z^T \nabla f_{i_0}$. The vector $Z^T \nabla f_{i_0}$ is zero if and only if there exist $\{\lambda_j\}_{j=0}^{l}$ such that

$$\lambda_0 \nabla f_{i_0} + \sum_{j=1}^{l} \lambda_j \nabla f_{i_j} = 0, \qquad \sum_{j=0}^{l} \lambda_j = 1. \tag{3.7}$$

Suppose (3.7) is satisfied. From (3.6) and (3.7), $\text{rank}(\{\nabla f_{i_0}, \nabla f_{i_1}, \cdots, \nabla f_{i_l}\}) = l$. Let $\mathcal{C} = \{\nabla f_{i_j} \mid \lambda_j \neq 0, j = 0, 1, \ldots, l\}$. Then, as in the argument for (3.5), $\mathcal{C}$ has rank $|\mathcal{C}| - 1$. From Lemma 2.2, $\mathcal{C}$ is a cadre. Moreover, the sum of the cadre multipliers is one.

On the other hand if there is a cadre $\mathcal{C} \subseteq \{\nabla f_{i_0}, \nabla f_{i_1}, \cdots, \nabla f_{i_l}\}$ with cadre multipliers summing to one, then, following Lemma 2.2, there exist $\{\lambda_j\}$ such that (3.7) holds and then, $Z^T \nabla f_{i_0} = 0$.

The lemma has been proved. □

With both Lemmas 3.2 and 3.3, we know whether there exists a cadre.

4. Establishment of the Working Set. A working set is a function index set which is used to determine the current descent direction. Since we want the search direction to decrease all the ϵ-active functions, this working set $\mathcal{W}^k$ is chosen to include all the ϵ-active functions at the current point x^k. Nonetheless, there is flexibility in constructing such a set. We have chosen to build up the working set by selecting the functions that are maximum through iterations. This is motivated by the fact that it is the extreme points that are important in determining the best approximation for a Chebyshev approximation problem. Thus, we require that

$$\mathcal{W}^k \subseteq \mathcal{W}^{k-1} \cup \mathcal{A}(x^k, \epsilon). \tag{4.1}$$

Moreover, the current ϵ-active functions are given priority over the old working functions in forming the new working set.

However, since adjustment of the functions in the working set is necessary when the current working set is not approaching a reference set, we use $\hat{\mathcal{W}}^k$ to denote the set after possible modification and the rules for changing the set will be described precisely later. Hence, in general, we require

$$\mathcal{W}^k \subseteq \hat{\mathcal{W}}^{k-1} \cup \mathcal{A}(x^k, \epsilon). \tag{4.2}$$

Assume, at the kth iteration, that a *representative function* $f_\mu(x)$, which can be any function $f_\mu(x)$ such that $\mu \in \mathcal{A}(x^k, \epsilon)$, is selected. Suppose $\mathcal{W}^k = \{\mu, i_1, \cdots, i_l\}$. The following Jacobian matrix corresponding to $\mathcal{W}^k$

$$A^k = [\nabla f_\mu - \nabla f_{i_1}, \cdots, \nabla f_\mu - \nabla f_{i_l}] \tag{4.3}$$

is required, *numerically* to have full rank where the columns of Z are an orthnormal basis for the null space of A^{k^T}. More specifically, our implementation accounts for this numerical rank. Conceptually it is equivalent to having some tolerance on the smallest singular value of A^k.

In implementation, we consider the projection $Z^T \nabla f_\mu$ numerically zero if

$$\|Z^T \nabla f_\mu(x^k)\| \leq \tau_c^k$$

where τ_c^k is a small positive constant. Hence, if we identify cadres according to Lemma 3.3, we have a *near* cadre.

Since we need the QR decomposition (see, for example [18], Chapter 6) of the matrix A^k in computing the direction (see § 5), we build up the current working set $\mathcal{W}^k$ as follows.

CONSTRUCT $\mathcal{W}^k$:

Step 1 Set $Q \leftarrow I_{n \times n}$, $\mathcal{W}^k \leftarrow \{\mu\}$, where $\mu \in \mathcal{A}(x^k, \epsilon)$. $t \leftarrow 0$.

Step 2 If $\mathcal{A}(x^k,\epsilon)\setminus\mathcal{W}^k=\emptyset$, go to Step 3. Otherwise, let Q_2 be the last $n-t$ columns of Q and $j\in\mathcal{A}(x^k,\epsilon)\setminus\mathcal{W}^k$. If $\|Q_2{}^T(\nabla f_\mu-\nabla f_j)\|\le\tau_0$, go to Step 2. Otherwise, go to Step 4.

Step 3 If $\hat{\mathcal{W}}^{k-1}\setminus\mathcal{W}^k=\emptyset$, stop. Otherwise, let Q_2 be the last $n-t$ columns of Q. If $\|Q_2^T\nabla f_\mu\|\le\tau_c^k$, stop. Let $j\in\hat{\mathcal{W}}^{k-1}\setminus\mathcal{W}^k$. If $\|Q_2{}^T(\nabla f_\mu-\nabla f_j)\|\le\tau_0$, go to Step 3. Otherwise, continue.

Step 4 Let $a=\nabla f_\mu-\nabla f_j$. Add the column a to A^k and update Q and R accordingly. Set:

$$A^k\leftarrow[A^k,a],\quad \mathcal{W}^k\leftarrow\mathcal{W}^k\cup\{j\},\quad t\leftarrow t+1.$$

Go to Step 2.

Thus, the working set is the largest subset of $\hat{\mathcal{W}}^{k-1}\cup\mathcal{A}(x^k,\epsilon)$ (largest in the sense of the corresponding Jacobian matrix A^k being full rank), where the indices of the current ϵ-active functions have been entered preferentially.

Following the procedure of constructing a working set, it is clear that, if the current point is nondegenerate and there is no cadre with cadre multipliers summing to zero, the Jacobian corresponding to all the ϵ-active functions is of full rank. Therefore

$$\mathcal{A}(x^k,\epsilon)\subseteq\mathcal{W}^k.$$

Moreover, if $\|Z^T\nabla f_\mu\|\le\tau_c^k$, where $Z=Q_2$ for some Q, then a cadre (or a near cadre) with cadre multipliers summing to one is found.

5. Determining the Search Direction. Assume the working set at the current point x_c is

$$\mathcal{W}(x_c)=\{i_0,\cdots,i_l\},\ \text{and}\ \mu=i_0.$$

The desired search direction, in addition to being one of descent, attempts to enforce the characterisation of a solution.

Before a cadre with multipliers adding to one is located, we would like the search direction to decrease all the active functions and level all the working functions, if possible. It is clear that $d=x-x_c$, where x attempts to solve

$$\min_{x\in\mathcal{R}^n} f_\mu(x)$$

(5.1) subject to

$$f_\mu(x)-f_{i_j}(x)=0,\qquad i_j\in\mathcal{W}(x_c),\ i_j\neq\mu,$$

is the required direction. Note that μ is in fact a function of x_c and we use it to denote the current representative function as long as no confusion arises.

Dropping the subscript on x_c to simplify the description, one may approximate (5.1) as follows:

$$\min_{d\in\mathcal{R}^n}\nabla f_\mu(x)^Td+\frac{1}{2}d^TGd$$

(5.2) subject to

$$\Phi(x)+A^Td=0,$$

where

$$A=[\nabla f_\mu(x)-\nabla f_{i_1}(x),\nabla f_\mu(x)-\nabla f_{i_2}(x),\cdots,\nabla f_\mu(x)-\nabla f_{i_l}(x)],$$
$$\Phi(x)=[f_\mu(x)-f_{i_1}(x),f_\mu(x)-f_{i_2}(x),\cdots,f_\mu(x)-f_{i_l}(x)]^T,$$

and G is a matrix such that Z^TGZ is positive definite, where the columns of Z form an orthonormal basis for the null space of A^T.

When close to a stationary point, Z^TGZ is chosen to contain the curvature information of the working functions in the null space of A^T (see §6 for details).

From the construction of the working set $\mathcal{W}(x)$, we know that A is of full rank. Following [10], the solution to (5.2) may be written as

$$\begin{aligned} d &= \hat{h} + v, \\ \hat{h} &= -Z(Z^TGZ)^{-1}Z^T(\nabla f_\mu(x) + Gv), \\ v &= -A(A^TA)^{-1}\Phi(x). \end{aligned}$$

It has been suggested in [10] that one could ignore the computation of Z^TGv altogether without significantly effecting the rate of convergence. In this case, an approximate solution to (5.2) can be written as

$$d = h + v,$$

where

$$\text{(5.3)} \qquad \begin{cases} h = -ZB^{-1}Z^T(\nabla f_\mu(x)), \\ v = -A(A^TA)^{-1}\Phi(x), \end{cases}$$

and

$$B = Z^TGZ.$$

It is clear that h is in the null space of A^T while v is in the range space of A. The direction in the null space of A^T will be called the *horizontal direction* and the direction in the range space of A will be called the *vertical direction.* We also point out that, given $\mathcal{W}$, Z and B, the value of h and v is independent of the choice of μ (see [14] for details).

We now prove that a nonzero horizontal direction h is a descending direction for all the functions in $\mathcal{W}$.

LEMMA 5.1. *Assume $\mathcal{W}$ is the working set that defines the search direction. Assume further that B is positive definite. Then, assuming there is no cadre $\mathcal{C} = \{\nabla f_{i_0}, \cdots, \nabla f_{i_l}\}$, with the cadre multipliers summing to one, such that $\{i_0, \cdots, i_l\} \subseteq \mathcal{W}$, the horizontal direction decreases all the functions in $\mathcal{W}$ equally (up to the first order); otherwise, the horizontal direction h defined from $\mathcal{W}$ is zero.*

Proof. The horizontal direction defined in (5.3) is

$$h = -ZB^{-1}Z^T(\nabla f_\mu(x)), \quad \mu = i_0,$$

where $Z^TZ = I_{n-l}$, $A^TZ = 0$. Since B is positive definite and

$$h^T\nabla f_\mu(x) = -(Z^T\nabla f_\mu(x))^TB^{-1}(Z^T\nabla f_\mu(x)),$$

it follows that

$$h^T\nabla f_\mu(x) < 0, \qquad \text{iff } Z^T\nabla f_\mu \neq 0.$$

Since there is no cadre $\mathcal{C} = \{\nabla f_{i_0}, \cdots, \nabla f_{i_l}\}$, with the cadre multipliers summing to one such that $\{i_0, \cdots, i_l\} \subseteq \mathcal{W}$, we have, from the definition of $\mathcal{W}$ and Lemma 3.3, $Z^T\nabla f_\mu \neq 0$ and h is a descent direction for the representative function $f_\mu(x)$.

Furthermore, since

$$A^T h = 0, \quad \text{and} \quad \nabla f_{i_j}{}^T h = \nabla f_\mu{}^T h, \quad i_j \in \mathcal{W},$$

any function in the working set $\mathcal{W}$ will be decreased by the same amount (up to first order) as the representative function f_μ.

On the other hand, assuming there exists a cadre with cadre multipliers summing to one, by Lemma 3.3, the result follows. □

In conclusion, the horizontal direction h is a projection of the negative gradient of the representative function onto the null space of A^T. It is always a descent direction as long as $\mathcal{W}$ is not a cadre with cadre multipliers summing to one. As a descent direction, it decreases the functions in the working set by the same amount (up to first order). The horizontal direction h defined on the cadre with the cadre multipliers summing to one is always zero.

5.1. No Cadre. If there is no cadre with cadre multipliers summing to one, we have $h \neq 0$.

If there exists a cadre with multipliers summing to zero, the cadre does not correspond to a reference set. In this case, although v (see §5.1 below) corresponds to levelling, we emphasize decreasing the maximum function. Thus, we simply take $d^k = h^k$.

When a cadre is not located, vertical directions are descent directions in most cases since the working set attempts to approximate a reference set. In this case, we perform the levelling process, i.e, set the search direction $d = v + h$. In the case that the vertical direction is ascending, the vertical direction is discarded and the horizontal direction alone is taken as the search direction, specifically we define

$$d^k = \begin{cases} h^k + v^k & \text{if } \nabla f_\mu{}^T v^k < 0 \\ h^k & \text{otherwise.} \end{cases} \tag{5.4}$$

Our numerical experience shows that an ascent vertical direction is a rare occurrence. This may be explained by the fact that the working set is constructed to approach a reference set. In the event that ascend does occur, we consider this as an indication that the working set is not approaching a reference set. This may be caused by some function which will not be maximum being included in $\mathcal{W}^k$. Thus, the next working set will not always include all the functions of the current working set, instead, we define

$$\begin{cases} \hat{\mathcal{W}}^k \leftarrow \mathcal{W}^k \setminus I^+, & \text{if } \nabla f_\mu^T v^k \geq 0, \text{ where} \\ I^+ = \begin{cases} \{j_0\} & \text{if } \mathcal{A}(x^k, \epsilon) \subset \mathcal{W}^k \text{ and } f_\mu - f_{j_0} = \max_{j \in \mathcal{W}^k}(f_\mu - f_j); \\ \emptyset & \text{otherwise.} \end{cases} \end{cases} \tag{5.5}$$

5.2. Levelling a Reference Set. If the functions in the working set form a reference set, the vertical direction v^k defined by (5.3) attempts to level the functions in the working set while the horizontal direction h^k (again defined by (5.3)) makes the gradients approach an *exact* cadre. From Lemma 2.5, v^k is a descent direction. Thus, $d^k = h^k + v^k$ is a descent direction (note that h^k is a descent direction).

5.3. Constructing a Reference Set. Suppose a cadre with multipliers summing to one has been located within the working set. As we see in the following Lemma 5.2, moving along v, which is defined by (2.2)

$$[\nabla f_\mu - \sigma_0 \sigma_j \nabla f_{i_j}]^T v = -(f_\mu - \sigma_0 \sigma_j f_{i_j}), \qquad i_j \in \mathcal{W},\ i_j \neq i_0,$$

attempts to construct a reference set. Furthermore, v is a descending direction for the maximum function. Equivalently, we can write (2.2) as

$$\begin{cases} \hat{A}v = -\hat{\Phi} \text{ where} \\ \hat{A} = [\nabla f_\mu - \sigma_0\sigma_1\nabla f_{i_1}, \cdots, \nabla f_\mu - \sigma_0\sigma_l\nabla f_{i_l}] \\ \hat{\Phi} = [f_\mu - \sigma_0\sigma_1 f_{i_1}, \cdots, f_\mu - \sigma_0\sigma_l f_{i_l}]^T. \end{cases} \tag{5.6}$$

We also modify the working set for the next iteration as follows. The cadre multipliers associated with the functions in the working set are used to construct the working set for the next iteration. The functions with positive multipliers are considered to be the functions which should be in the working set, i.e., the correct functions. For the functions with negative multipliers, we would like to put its negative function into the working set. However, because of nonlinearity and the fact that the cadre and reference set are both local properties, we prefer not to do so. Instead, the functions with negative multipliers are simply deleted from the working set, since the functions corresponding to negative multipliers will no longer remain ϵ-active when the direction v is taken and the multipliers sum to one. Thus we define

$$\hat{\mathcal{W}}^k \leftarrow \mathcal{W}^k \setminus \{ i_j \mid \lambda_j < 0 \}. \tag{5.7}$$

The multipliers are thus used as a means to construct the working set and more than one functions may be removed.

In [13], we have been able to prove the following.

LEMMA 5.2. *Suppose $\mathcal{W} = \{\mu, i_1, \cdots, i_l\}$ consists only of indices of the currently active functions. Assume further that $C = \{\nabla f_\mu, \nabla f_{i_1}, \cdots, \nabla f_{i_l}\}$ is a cadre. Assume the direction v is determined from $\mathcal{W}$ as in (2.2). Then:*

1. *all the active functions with negative multipliers will be decreased more rapidly than all the other active functions if the cadre multipliers sum to one, i.e. $\sum_{j=0}^{l} \lambda_j = 1$;*
2. *all the active functions are decreased equally (up to first order) provided the cadre multipliers sum to zero, i.e., $\sum_{j=0}^{l} \lambda_j = 0$.*

Thus, if the functions in the working set are all active and the multipliers sum to one, moving along the vertical direction initially decreases all the functions with the negative multipliers faster (up to first order) than those with positive multipliers. This corresponds to (possibly multiple) dropping of active functions for the equivalent nonlinear programming problem.

Now, consider a general nonlinear minimax problem written as

$$\min_{x \in \Re^n} \max_{i \in \{1, \cdots, m\}} f_i(x).$$

The search direction can be computed in the exactly same way except that the reference set, after a cadre has been located, could not be established as before. Since there exists no negative function of a given function, the vertical direction that determines which active functions should be dropped is not defined. Thus we now discuss how the definition of the vertical direction is modified for the general minimax problem.

If the current maximum deviation $\psi(x^k)$ is positive, we assume that for any given $f_i(x)$, there exists an imaginary $f_{i+m}(x) = -f_i(x)$. The working set $\mathcal{W}^k$ is chosen such that

$$-\psi(x^k) < f_{i_j}(x^k) \leq \psi(x^k), \quad \text{for any } i_j \in \mathcal{W}^k.$$

Hence locally we can treat the problem as a Chebyshev problem and the vertical direction, defined as for the Chebyshev problem, is a descent direction.

If the current maximum deviation $\psi(x)$ is nonpositive, we define a descent direction in a way similar to a general nonlinear programming approach [12]. In this case, if there exists some cadre multiplier which is negative, we simply remove the corresponding function from the working set and update the projection matrix and recompute the search direction from the new projector. Under the assumption of linear independence, this will give a descent direction [12].

6. Approximation of the Hessian. In order to obtain a horizontal descent direction at each iteration, B^k, an $(n-l)\times(n-l)$ matrix, is assumed to be sufficiently definite.

For problems whose solutions are on a smooth valley, i.e., the number of active functions is less than $n+1$, the second order information from the nonlinear active functions becomes significant for the fast final convergence of the algorithm. When close to x^*, B^k should be a good approximation to the projected Lagrangian Hessian, $Z^{k^T}G^kZ^k$, where $G^k = \sum_{j=0}^{l}\lambda_j^k\nabla^2 f_{i_j}(x^k)$, the columns of Z^k form a basis for the null space of A^{k^T}, and $\lambda_j{}^k$ is an approximation to the Lagrangian multipliers (which are defined by the first order optimality conditions of the equivalent nonlinear programming problem, see for example, [13]).

If we assume the second order sufficiency conditions hold at x^* and let λ^k be a good approximation to the cadre multipliers, λ^*, at *a solution* x^*, (which are equal to the Lagrangian multipliers at a solution), then the matrix $Z^{k^T}G^kZ^k$, for x^k sufficiently close to x^*, is positive definite, as follows from continuity arguments.

The first order method, for example of [9], solves the problem whose solution is at a vertex (i.e. with $n+1$ linear independent activities) with a fast asymptotic rate of convergence since, once the correct activities are determined, one is merely using Newton's method (or a quasi–Newton method) to determine the unique intersection of these activities, with the corresponding quadratic (or superlinear) rate of convergence. First order directions are usually good descent directions when we are far away from a stationary point and the computation of a first order direction is cheaper than a second order direction. We choose to use the first order direction if it gives a good improvement in the sense of constructing reference sets.

Computationally, we consider that the first order direction fails to improve the establishment of reference sets when the working set has not been changed for γ consecutive iterations (this may be a result of having the correct set but in this case it is reasonable to want to accelerate convergence by using a second order direction). We arbitrarily set $\gamma = 3$ in our implementation. When failure occurs, we use the second order information of the representative function or of all the working functions, depending on how close we are to a stationary point of the subproblem.

Let *ibase* denote the number of consecutive iterations for which the working set remains unchanged. Suppose ρ is a small positive constant used to measure the closeness to a stationary point. The matrix G^k is set up as follows:

$$G^k \begin{cases} \approx \nabla^2 f_\mu(x^k) & \text{if } ibase \geq \gamma \text{ and } \|Z^{k^T}\nabla f_\mu\| > \rho, \\ \approx \sum_{j=0}^{l}\lambda_j^k\nabla^2 f_{i_j}(x^k) & \text{if } ibase \geq \gamma \text{ and } \|Z^{k^T}\nabla f_\mu\| \leq \rho, \\ = I & \text{otherwise,} \end{cases} \tag{6.1}$$

where $\lambda_j{}^k$ is an approximation to the Lagrangian multipliers. We note that when

$\|Z^{k^T}\nabla f_\mu\| \leq \rho$, it is reasonable to expect a suitable approximation to the Lagrangian multipliers.

Also, when $G^k = I$, the search direction is a first order direction.

In our algorithm, we use a quasi–Newton method to update the projected Hessian. Suppose Z^k is the orthogonal matrix such that $Z^{k^T}A^k = 0$, where A^k is defined as in (4.3). In the implementation, we have used the extended BFGS updating given below. Finite differences are used to initialise the approximation.

Extended BFGS Updating

$$B^{k+1} = B^k - \frac{1}{s_r^{k^T} B^k s_r^k} B^k s_r^k s_r^{k^T} B^k + \frac{1}{y_r^{k^T} s_r^k},$$

where

$$s_r^k = Z^{k+1^T}(x^{k+1} - x^k),$$
$$y_r^k = Z^{k+1^T}\nabla f_\mu(x^{k+1}) - Z^{k^T}\nabla f_\mu(x^k).$$

Assume B^k is positive definite. Then B^{k+1} remains positive definite if $s_r^{k^T} y_r^k > 0$. For unconstrained minimization, this condition is ensured by a line search. For constrained minimization, however, it can not be satisfied in general. We have chosen to skip the update if the above condition is not satisfied.

7. Degeneracy. For a discrete Chebyshev problem, degeneracy handling is an important part of a useful algorithm. This is because, for example in the linear case, it is not unusual for many residuals to achieve the maximum deviation. In this section, we discuss the handling of degeneracy in our algorithm.

Denote

$$\mathcal{W}^k = \{\mu, i_1, \cdots, i_l\}, \quad A^k = [\nabla f_\mu - \nabla f_{i_1}, \cdots, \nabla f_\mu - \nabla f_{i_l}].$$

If x^k is a degenerate point, the following difficulty may occur. There is more than one cadre $\mathcal{C} = \{\nabla f_\mu, \nabla f_{i_1}, \cdots, \nabla f_{i_l}\}$ satisfying $\mathcal{W}^k \subset \mathcal{A}(x^k, 0)$. Thus it may not be possible to define a search direction such that it decreases the functions in all the cadres, although we know how to define a descending direction on one cadre.

If we consider the cadres which correspond to subsets of active functions, then there can be three types of degenerate points:

Type A there only exist cadres with cadre multipliers summing to zero;
Type B there exists a unique cadre and its cadre multipliers sum to one;
Type C there exists more than one cadre and at least one with cadre multipliers summing to one.

A point x^* is a stationary point if and only if there exists at least one reference set consisting of active functions only.

For the degenerate points of Type A, there can not be any reference set consisting of only the active functions. This is because, for any reference set, each of the corresponding cadre multipliers is positive and the sum of them is one. Thus, the current point cannot be optimal. For this type of degeneracy, the horizontal direction h defined on the current working set decreases all the ϵ-active functions, up to first order, by the same amount.

For the degenerate points of Type B, it is possible that a reference set exists within the active set. If there is such a reference set, then the current point is already a stationary point. Otherwise, since there exists a unique cadre, the vertical direction v defined on the cadre by (2.2) attempts to construct a levelled reference set. Moreover, other maximum functions not in the cadre can also be decreased at the same time. Since we identify cadres by a tolerance of τ^k, the (numerical) degeneracy identified depends on the tightness of ${\tau_c}^k$. Thus, when degeneracy is encountered, we reduce it by

$$ {\tau_c}^{k+1} \leftarrow \frac{{\tau_c}^k}{2}. \tag{7.1} $$

For the degenerate points of Type C, we do not know a direct way of defining a descent direction. Following a similar approach to [6], we solve the least squares problem:

$$ \min_{\theta \in \Re^{l+1}} \| \sum_{j=0}^{l} \theta_j \nabla f_{i_j} \|_2 $$

$$ \text{subject to} \tag{7.2} $$

$$ \sum_{j=0}^{l} \theta_j = 1, \quad \theta_j \geq 0, \quad j = 0, \cdots, l. $$

Assume $\{\lambda_j^k\}$ is the solution to (7.2). Analogous to the proof in [6], d^k defined by

$$ d^k = -\sum_j \lambda_j^k \nabla f_{i_j} \tag{7.3} $$

is a descent direction unless $d^k = 0$, in which we are optimal. Moreover, it is not difficult to prove that (7.2) can be solved via a least squares problem with only simple nonnegativity constraints [14].

8. Summary of the Algorithm. Now we give a more detailed description of the algorithm.

Initialization: Suppose an initial point x^0 is given. Set $k \leftarrow 1$, $\hat{\mathcal{W}}^0 \leftarrow \emptyset$.

Step 1 [QR Decomposition]
Find the working set $\mathcal{W}^k \subseteq \hat{\mathcal{W}}^{k-1} \cup \mathcal{A}(x^k, \epsilon)$, Jacobian A^k and its QR decomposition. Assume the columns of Z^k form a basis for the null space of ${A^k}^T$.
If $\mathcal{A}(x^k, \epsilon) \subseteq \mathcal{W}^k$ and $\|{Z^k}^T \nabla f_\mu\| \leq {\tau_c}^k$, go to Step 2;
If $\mathcal{A}(x^k, \epsilon) \subseteq \mathcal{W}^k$ and $\|{Z^k}^T \nabla f_\mu\| > {\tau_c}^k$, go to Step 3;
If $\mathcal{A}(x^k, \epsilon) \not\subseteq \mathcal{W}^k$ and $\|{Z^k}^T \nabla f_\mu\| > {\tau_c}^k$, go to Step 4;
If $\mathcal{A}(x^k, \epsilon) \not\subseteq \mathcal{W}^k$ and $\|{Z^k}^T \nabla f_\mu\| \leq {\tau_c}^k$, go to Step 5;

Step 2 [Cadre "Found" with $\sum_{i \in \mathcal{C}} \lambda_i = 1$] If $\mathcal{W}^k$ is a reference set, obtain $B^k = {Z^k}^T G^k Z^k$, where G^k is defined as in (6.1); Compute the horizontal direction h^k and the vertical direction v^k from (5.3); Set the search direction $d^k = h^k + v^k$ and $\hat{\mathcal{W}}^k \leftarrow \mathcal{W}^k$.
Otherwise, compute the vertical direction according to (5.6) and set $\hat{\mathcal{W}}^k$ using (5.7). Modify ${\tau_c}^k$ by (7.1). Set $d^k = v^k$. Go to Step 6.

Step 3 [Cadre not Found]
Obtain $B^k = Z^{k^T} G^k Z^k$, where G^k is defined as in (6.1). Compute the horizontal direction h^k and the vertical direction v^k from (5.3). Compute the search direction d^k using (5.4). Set up $\hat{\mathcal{W}}^k$ according to (5.5). Go to Step 6.

Step 4 [Cadre "Found" with $\sum_{i\in\mathcal{C}} \lambda_i = 0$]
Compute $d^k = -Z^k Z^{k^T} \nabla f_\mu^k$. $\hat{\mathcal{W}}^k \leftarrow \mathcal{W}^k$. Go to Step 6.

Step 5 [More than One Cadre and at Least One with $\sum_{i\in\mathcal{C}} \lambda_i = 1$]
Compute the search direction d^k using (7.3). Obtain $\hat{\mathcal{W}}^k$ from (5.7). Set $\tau_c^{k+1} \leftarrow \frac{\tau_c^k}{2}$.

Step 6 [Line Search]
Perform a safeguarded line search. Set $k \leftarrow k+1$. If $\|d^k\|_2 < \tau_s$ and $\mathcal{W}^k$ includes a levelled reference set, stop. Otherwise, go to Step 1.□

We use quotes around "found" to emphasize that τ_c^k is nonzero. The safeguards and details of the line search are given in [14].

9. Numerical Testing. In this section, we compare the new algorithm with four other typical methods: [7], [12], [20], [33].

The numerical results are for both minimax problems and discrete Chebyshev problems all written in the form:

$$\min_{x\in\Re^n} \max_{i\in M} f_i(x). \tag{9.1}$$

The Method of Conn. The method of [12] basically applies the active set strategy of nonlinear programming to the equivalent form of a minimax problem. It is a globally convergent algorithm with a superlinear convergence rate.

At each iteration, an equality constrained quadratic programming subproblem is solved to determine the search direction. The subproblem is established upon all the current ϵ-active functions. The finite difference of the derivatives is used to approximate the second order information.

This approach essentially corresponds to the sequential equality constrained quadratic programming (EQP) approach for nonlinear programming problems, using projected Hessians. However, once the search direction is determined, the line search is done directly on the non-differentiable maximum function $\psi(x)$.

Although numerical results for general nonlinear minimax problems have been relatively fewer than for linear problems, to date, the available numerical results seem to indicate that the following method ([20]) which is a combination of a linear programming (LP) approach and a quasi–Newton method for a nonlinear system of equations, works well on most types of minimax problems.

The Method of Hald and Madsen. At each iteration of the first stage, the method of [20] requires an exact solution to a constrained linear minimax problem

$$\min_{d\in\Re^n} \max_{i\in[M]} \{f_i(x^k) + \nabla f_i(x^k)^T d\}$$

subject to

$$\|d\|_\infty \le \Lambda^k$$

in order to find the search direction. A trust region method has been incorporated to ensure convergence.

If a solution is suspected to go through a smooth valley, i.e., the number of active functions at the solution is less than $n + 1$, a switch to a second stage is made within which a nonlinear system of equations established by the Kuhn–Tucker conditions for the active functions is solved by some quasi–Newton method.

The entire Lagrangian Hessian is approximated by some modified secant updates. It is possible for the maximum $\psi(x)$ to be increased. A return to the first stage might be necessary.

Under certain conditions, the method of [20] is globally convergent with a quadratic or superlinear final convergence depending upon whether or not a quasi–Newton method is involved.

The first stage of the method essentially corresponds to a sequential linear programming approach (SLP), stabilized via a trust region, for nonlinear programming problems.

The Method of Womersley and Fletcher. The method of [33] is similar to that of [12]. It is a descent method which uses an active set strategy, a nonsmooth line search and a quasi–Newton approximation to the projected Hessian of the Lagrangian function.

Global convergence of the algorithm has been proved. Under certain conditions, superlinear convergence occurs.

Like that of [12], this method could be considered as belonging to the class of sequential equality constrained quadratic programming (EQP) approaches.

The Method of Charalambous. In the approach of [7], the original minimax problem is defined as a modified least pth objective function which under certain conditions has the same optimum as the original problem.

9.1. Computational Costs Comparison. At each iteration, the methods of [12] and [33] and the new algorithm require the computation of a search direction by solving an equality constrained quadratic programming (EQP) or an equality constrained linear programming (ELP). Comparable nonsmooth line searches have been used in the methods of Conn and Womersley and Fletcher, whereas Hald and Madsen used the trust region method. For our new algorithm, determining a cadre and dropping one working function, when a non-reference-set cadre is found, requires no extra work compared with the methods of [12] and [33]. When more than one working function is dropped, an equivalent number of QR updates are required. Since these functions should be dropped and function evaluation is more expensive than a single QR update, in general, this extra work is well justified. The amount of computation per iteration required by the above three methods is roughly $O(n^3)$, assuming the functions are indeed nonlinear.

The amount of work required by each iteration of [7] is roughly the same as performing a quasi–Newton step for an unconstrained function. It involves solving a linear system of equations of dimension $n \times n$. Hence, the amount of work per iteration is also comparable with the aforementioned three methods.

At each iteration of [20], in stage one, a linear programming problem of size at least $n \times |M|$ is solved up to optimality. At each iteration of stage two, if it is ever entered, the computation required is similar to the methods of [12] etc. However, in general, most of the iterations are spent in stage one.

Loosely speaking, comparison of computational costs of one iteration of the new

algorithm and that of [20] is similar to the comparison between one iteration of EQP and IQP methods.

A solution of EQP can be obtained by solving two linear systems of equations. The size of each linear system is at most n. A solution for IQP, however, usually requires iterative methods (i.e. inner iterations). Although the number of iterations are bounded by the number of unknowns and constraints, it is potentially very large and it could even become prohibitive for a discretised Chebyshev problem because the number of the constraints of its associated IQP can be much larger than those of usual nonlinear programming problems.

Therefore, considering the amount of work required per iteration, the method of [20] is considerably more expensive than the others.

For nonlinear programming problems, the advantage of the IQP approach compared to EQP, however, has been the iterative search for the correct active set. It can be shown that in a neighborhood of the solution, under certain conditions, some IQP subproblems will make the correct choice of active set, in the sense that the set of active linear constraints at the solution of the QP is equivalent to the set of active nonlinear constraints active at the solution of the original nonlinear programming problem ([27]).

Likewise, one would expect that the advantage of the method of [20] over that of [12], [33] and the new algorithm is similar to that of the successive IQP method over the successive EQP approach for programming problems, namely, it can identify the correct active set faster. This probably is true for the methods of [12] and [33]. The new algorithm, however, is not an active set method. It can also identify the correct active set quickly. It achieves this not by an iterative search but by recognising the structure of the optimum and constructively building up the reference set. Through exploiting the structure of the Chebyshev problem and minimax problem, we are able to retain the advantages of both the EQP approach and the IQP approach.

Finally, we remark that for a degenerate point of Type A or B, there is no extra work required compared with that for a nondegenerate point. For a degenerate point of Type C, we must solve a least squares problem with nonnegativity constraints.

9.2. Numerical Results. We present some limited numerical results in this section.

For our numerical testing, the constants required by the algorithm are set as

$$\tau_c^0 = 0.05, \quad \tau_0 = 10^{-12}, \quad \tau_s = \frac{1}{2}10^{-5}, \quad \rho = 0.5.$$

The algorithm terminates when the following three conditions are satisfied

1. $\|d^k\|_2 \leq \tau_s$;
2. $\mathcal{W}^k \subseteq \mathcal{A}(x^k, \epsilon)$;
3. $\lambda_j^k \geq 0$, for all $j \in \mathcal{W}^k$.

Thus, at termination, there exists, approximately, a levelled reference set with the maximum deviation.

The test problems include both nonlinear minimax problems and nonlinear Chebyshev problems.

We implicitly write a nonlinear Chebyshev problem

$$\min_{1 \leq i \leq m} |f_i(x)|,$$

in the general minimax form

$$\min_{1\leq i\leq 2m} f_i(x),$$

where $f_{i+m}(x) = -f_i(x)$, for $i = 1, \cdots, m$.

Consider the following nonlinear programming problem:

$$\min_{x\in\Re^n} F(x)$$

subject to

$$g_i(x) \geq 0, \quad i = 2, \cdots, m,$$

and the minimax problem:

$$\min_{x\in\Re^n} \max_{1\leq i\leq m} f_i(x)$$

subject to

$$f_1(x) = F(x),$$
$$f_i(x) = F(x) - \alpha_i g_i(x), \quad 2 \leq i \leq m,$$

where

$$\alpha_i \geq 0, \quad 2 \leq i \leq m.$$

It is straightforward to show that (see [2]) that for sufficiently large α_i, the optimum of the minimax problem coincides with that of the nonlinear programming problem.

We have tested some nonlinear programming problems through the above transformation. The α parameter is set as

$$\alpha_i = 10.0, \quad 2 \leq i \leq m,$$

which we know, a priori, is sufficiently large.

We have listed the results for the following minimax testing problems (their references are also indicated): Charalambous and Bandler 1 [8], Charalambous and Bandler 2 [8], Freudenstein and Roth [32], Colville Problem 2 [11], Barrodale, Powell and Roberts [4], Wong 1, Wong 2 and Wong 3 [9], Rosen and Suzuki [29], Rosenbrock [30], Transmission Problems [3], Davidon [15], Enzyme [22], El Attar [16], Hettich [32], Bard [32], Watson [32] and Osborne [32]. The starting points used are the same as that specified in the references.

The results for the problems Davidon, Enzyme, El Attar and Hettich, under the column [20], are taken from [25] which describes essentially the same method as that of Hald and Madsen.

In Table 1, we report the number of function evaluations required by our new algorithm, under the column NM. For each problem, we have used the nomenclature of the cited reference. The results of other methods, using a comparable stopping tolerance, are listed for comparison where available.

The column under the column *nact* indicates the number of maximum functions at the solution.

The Rosenbrock problem is degenerate at the solution. The Watson problem is degenerate at the starting point. The Watson problem with $n = 20$ is also degenerate at the solution obtained. For the other test problems, numerical degeneracy does not occur.

PROBLEMS	n	m	nact	NM	HM	CN	WF	CL
					[20]	[12]	[33]	[7]
CHARALAMBOUS & BANDLER 1	2	3	2	11	11[b]	18	12	
CHARALAMBOUS & BANDLER 2	2	3	3	6	11[b]	8	6	
FREUDENSTEIN & ROTH	2	2	2	11	15[b]			
COLVILLE.2	15	21	12	49	41[b]	275	80	413
BARRODALE,POWELL et al	5	21	5	21	10		38	
WONG1.1	7	5	3	25	23	106	53	107
WONG1.2	7	5	3	33	29	77	37	
WONG2	10	9	7	24	27 [a]			120
WONG3	20	18	13	33	49[b]			318
ROSEN & SUZUKI	4	4	3	12	18	64	37	66
ROSENBROCK	2	2	4	31	21			
TRANSMISSION 1	6	11	4	52	21	67		78
TRANSMISSION 2	6	11	4	25	46	80		
DAVIDON	4	20	3	20	27			
ENZYME	4	22	5	11	18			
EL ATTAR	6	51	7	25	12			
HETTICH	4	5	4	11	19[b]			
BARD	3	15	3	10	9[b]			
MADSEN	2	3	2	17	13[b]			
WATSON6	6	31	7	24	12[b]			
WATSON20	20	31	39	22	39[b]			
OSBORNE	5	33	5	10	31[b]			

[a] The results are obtained by using the codes in [20].
[b] The algorithm stopped because of round off error without obtaining a solution.

TABLE 1
Number of Function Evaluations: BFGS Updates

The reported results use extended BFGS updates. Similar results were obtained using exact derivatives. From the limited numerical results, we observe that, compared with [7], [12] and [33], the overall number of function evaluations required by the new algorithm is much less. We also recall that the amount of computation required by the aforementioned four methods are comparable. Hence, the new method appears to be more efficient than that of [7], [12] and [33].

The only method that seems to be competitive with the new algorithm is that of [20]. The number of function evaluations required by these two methods is comparable. However, we recall that the amount of computation required per iteration demanded by the method of [20] is significantly more than the proposed method. Thus, our new method still appears to be preferable.

We have also tested our new algorithm on a real application problem. The problem has 80 functions, in terms of a general minimax problem, with 40 variables. The number of activities at the solution is 39 (out of 80). Our algorithm solved it successfully in 50 function evaluations while the method of [20] failed to locate a solution.

10. Summary. The algorithm presented is a globally convergent algorithm with superlinear convergence rate [23]. It has been developed based on the principle that a minimax problem, in particular the Chebyshev problem, has special properties that can be computationally exploited in both the linear and nonlinear cases.

In [13], we have given a characterisation for a nonlinear minimax problem that can be computationally exploited. In this paper, we present the algorithm which profits from this exploitation.

We point out that it is possible for the Maratos effect to occur for the new algorithm as presently implemented. However, we have not experienced this effect for our numerical testings. Moreover, the algorithm can be slightly modified to guarantee that there is no Maratos effect for any problem. We only need to re–evaluate the functions at the point $x^k + h^k$ and compute the vertical direction using the updated values, when one is close to a stationary point (see [14] for more details).

Finally, we point out that the algorithm can be extended to solve the constrained minimax problem. (see [14] for more details).

REFERENCES

[1] D. H. ANDERSON AND M. R. OSBORNE, *Discrete, nonlinear approximations in polyhedral norms: a Levenberg-like algorithm*, Numerische Mathematik, 28 (1977), pp. 167–170.

[2] J. W. BANDLER AND C. CHARALAMBOUS, *Nonlinear programming using minimax techniques*, J. Optimization Theory Appl., 13 (1974), pp. 607–619.

[3] J. W. BANDLER AND P. A. MCDOBALD, *Optimization of microwave networks by razor research*, IEEE Trans. Microwave Theory Tech., 17 (1969), pp. 552–562.

[4] I. BARRODALE, M. J. D. POWELL, AND F. D. K. ROBERTS, *The differential correction algorithm for rational l_∞-approximation*, SIAM J. Num. Anal., 7 (1972), pp. 493–504.

[5] R. H. BARTELS, A. R. CONN, AND Y. LI, *Primal methods are better than dual methods for solving overdetermined linear systems in the l_∞ sense?*, SIAM J. Numer. Anal., 26 (1989), pp. 693–726.

[6] S. BUSOVAČA, *Handling degeneracy in a nonlinear l_1 algorithm*, Tech. Rep. Tech. Rept. CS-85-34, Univ. of Waterloo, Dept. of Computer Science, Univ. of Waterloo, Waterloo, Ontario N2L 3G1, 1985.

[7] C. CHARALAMBOUS, *Acceleration of the least pth algorithm for minimax optimization*, Tech. Rep. 28-0-28067, University of Waterloo, Department of System Design, 1977.

[8] C. CHARALAMBOUS AND J. W. BANDLER, *Nonlinear minimax optimisation as a sequence of least pth optimization with finite values of p*, International Journal of System Science, 7 (1976), pp. 377–394.

[9] C. CHARALAMBOUS AND A. R. CONN, *An efficient method to solve the minimax problem directly*, SIAM J. Numer. Anal., 15 (1978), pp. 162–187.

[10] T. F. COLEMAN AND A. R. CONN, *Nonlinear programming via an exact penalty function: Asymptotic analysis*, Math. Prog., 24 (1982), pp. 123–136.

[11] A. R. COLVILLE, *A comparative study on nonlinear programming codes*, Tech. Rep. 320-2949, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 1968.

[12] A. R. CONN, *An efficient second order method to solve the (constrained) minimax problem*, Tech. Rep. CORR 79-5, Dept. of Combinatorics and Optimization, University of Waterloo, Ontario, Canada, 1979.

[13] A. R. CONN AND Y. LI, *Structure and characterization of discrete chebyshev problems*, Tech. Rep. CS-88-39, Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada, 1988.

[14] A. R. CONN AND Y. LI, *An approach to nonlinear l_∞ approximation*, Tech. Rep. CS-89-21, Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada, 1989.

[15] W. C. DAVIDON, *A new least squares algorithm*, J. Opt. Theory and Applics., 18 (1976), pp. 187–198.

[16] EL-ATTAR, M. VIDYASAGAR, AND S. R. K. DUTTA, *An algorithm for l_1-approximation*, SIAM J. Num. Anal., 16 (1979), pp. 70–86.

[17] R. FLETCHER, *A model algorithm for composite nondifferentiable optimization problems*, Math. Prog. Study, 17 (1982), pp. 67–76.

[18] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, 1983.

[19] N. I. M. GOULD, *On solving three classes of nonlinear programming problems via simple differentiable penalty functions*, J. Opt. Theory and Applics., 56 (1988), pp. 89–126.

[20] J. HALD AND K. MADSEN, *Combined LP and Quasi-Newton methods for minimax optimization*, Math. Prog., 20 (1981), pp. 49–62.

[21] S. P. HAN, *Variable metric methods for minimizing a class of nondifferentiable functions*, Math. Prog., 20 (1981), pp. 1–13.

[22] J. S. KOWALIK AND M. R. OSBORNE, *Methods for Unconstrained Optimization*, American Elsevier Publishing Co., 1968.

[23] Y. LI, *An Efficient Algorithm for Nonlinear Minimax Problems*, PhD thesis, University of Waterloo, 1988.

[24] K. MADSEN, *Minimax solution of nonlinear equations without calculating derivatives*, Mathematical Programming Study, 3 (1975), pp. 110–126.

[25] ———, *Minimization of Nonlinear Approximation Functions*, PhD thesis, Technical University of Denmark, 1985.

[26] W. MURRAY AND M. L. OVERTON, *A projected Lagrangian algorithm for nonlinear minimax optimization*, SIAM. J. Sci. Stat. Comput., 1 (1980), pp. 345–370.

[27] W. MURRAY AND M. H. WRIGHT, *Computation of the search direction in constrained optimization algorithms*, Math. Prog. Study, 16 (1982), pp. 62–83.

[28] M. R. OSBORNE AND G. A. WATSON, *An algorithm for minimax approximation in the nonlinear case*, Computing J., 12 (1968), pp. 63–68.
[29] J. B. ROSEN AND S. SUZUKI, *Construction of the nonlinear programming test problems*, Communications of the Association of Computing Machinery, 8 (1965), p. 113.
[30] H. H. ROSENBROCK, *An automatic method for finding the greatest or least value of a function*, Comp. J., 3 (1960), pp. 175–184.
[31] E. L. STIEFEL, *Numerical methods of Tchebycheff approximation*, in Proceedings of a symposium, math. research center, Univ. of Wisconsin Press, 1959.
[32] G. A. WATSON, *The minimax solution of an overdetermined system of non-linear equations*, J. Inst. Math. Appl., 23 (1979), pp. 167–180.
[33] R. S. WOMERSLEY AND R. FLETCHER, *An algorithm for composite nonsmooth optimization problems*, J. Opt. Theory and Applis., 48 (1986), pp. 493–523.