Linear Programming via Least Squares

Achiya Dax
Hydrological Service
P.O.B. 6381
Jerusalem, 91060, Israel

Submitted by Hans Schneider

ABSTRACT

The paper suggests a new implementation of the active set method for solving linear programming problems. The proposed method is based on the observation that the search direction can be obtained via the solution of a linear least squares subproblem. It is shown that the steepest descent direction can be computed by solving the same least squares subproblem but with simple bounds on the variables. This direction is used to prevent cycling at degenerate dead points. Numerical experiments illustrate the feasibility of the new approach.

1. INTRODUCTION

This paper outlines an active set method for solving the problem

$$\begin{aligned} &\text{minimize} && \mathbf{g}^T\mathbf{x} \\ &\text{subject to} && \mathbf{a}_i^T\mathbf{x} = b_i \quad \text{for } i \in E \\ &\text{and} && \mathbf{a}_i^T x \geqslant b_i \quad \text{for } i \in C, \end{aligned} \tag{1.1}$$

where E and C are finite index sets, $\mathbf{x} \in \mathbb{R}^n$ is the vector of unknowns, $\mathbf{g}$ and $\mathbf{a}_i$ are given vectors in $\mathbb{R}^n$, and b_i are given real numbers.

A comprehensive up-to-date survey of methods for solving this problem can be found in Fletcher (1981) or Osborne (1985). The proposed algorithm can be viewed as a modification of the nonsimplex method of Gill and

655 Avenue of the Americas, New York, NY 10010 0024-3795/88/$3.50

Murray (1973), which enables the handling of degenerate and underdetermined problems. The new feature which characterizes our method is that the search direction is obtained by solving a linear least squares problem. The motivation behind this approach is based on constructive optimality conditions that avoid the traditional assumption of nondegeneracy.

Being an active set method, the new algorithm is related to the methods of Conn (1976) and Bartels (1980). However, there are significant differences between the two approaches. That is, the new method is not a penalty function method and preserves feasibility. Also, it handles degeneracy in a different way.

2. MOTIVATION AND OPTIMALITY CONDITIONS

We shall start by introducing some necessary terminology. Let $\mathbf{x} \in \mathbb{R}^n$ be a feasible point, i.e., $\mathbf{a}_i^T\mathbf{x} = b_i$ for $i \in E$ and $\mathbf{a}_i^T\mathbf{x} \geqslant b_i$ for $i \in C$. Define

$$C^* = \left\{ i \middle| i \in C \text{ and } \mathbf{a}_i^T\mathbf{x} = b_i \right\}$$

and

$$N = E \cup C^*.$$

Then N contains the indices of the *active* constraints. The number of active constraints is denoted by t. The active constraint matrix A is a $t \times n$ matrix whose rows are $\mathbf{a}_i^T$, $i \in N$. The order of the rows does not matter. Thus it is assumed here, for simplicity, that $N = \{1, \dots, t\}$.

A vector $\mathbf{u} \in \mathbb{R}^n$ is said to be a feasible direction at $\mathbf{x}$ if it satisfies

$$\mathbf{a}_i^T\mathbf{u} = 0 \quad \text{for } i \in E, \quad \text{and} \quad \mathbf{a}_i^T\mathbf{u} \geqslant 0 \quad \text{for } i \in C^*. \tag{2.1}$$

If, in addition to (2.1), $\mathbf{u}$ satisfies

$$\mathbf{g}^T\mathbf{u} < 0, \tag{2.2}$$

then it is called a feasible descent direction. The following theorem provides a simple way to obtain such a direction.

THEOREM 1. *Let* $\tilde{\mathbf{y}} \in \mathbb{R}^t$ *solve the problem*

$$\text{minimize} \quad \|A^T\mathbf{y} - \mathbf{g}\|_2^2, \tag{2.3}$$

and let $\mathbf{r} = A^T\tilde{\mathbf{y}} - \mathbf{g}$ *denote the corresponding residual vector. If* $\mathbf{r} \neq \mathbf{0}$, *it is a feasible descent direction.*

Proof. The basic properties of the linear least squares problem imply that $A\mathbf{r} = \mathbf{0}$. This means that $\mathbf{r}$ is a feasible direction and

$$\mathbf{g}^T\mathbf{r} = \left(A^T\tilde{\mathbf{y}} - \mathbf{r}\right)^T\mathbf{r} = \tilde{\mathbf{y}}^TA\mathbf{r} - \mathbf{r}^T\mathbf{r} = -\mathbf{r}^T\mathbf{r} < 0. \qquad \blacksquare$$

If $\mathbf{r} = \mathbf{0}$, then $\mathbf{x}$ is called a *dead* point. In this case the question whether there exists a feasible descent direction at $\mathbf{x}$ is answered by imposing certain bounds on the components of $\mathbf{y}$.

THEOREM 2. *Let* $\mathbf{y}^* \in \mathbb{R}^t$ *solve the problem*

$$\text{minimize} \quad \|A^T\mathbf{y} - \mathbf{g}\|_2^2 \tag{2.4}$$

$$\text{subject to} \quad y_i \geq 0 \quad \textit{for } i \in C^*,$$

and let $\mathbf{r} = A^T\mathbf{y}^* - \mathbf{g}$ *denote the corresponding residual vector. If* $\mathbf{r} = \mathbf{0}$, *then* $\mathbf{x}$ *solves* (1.1). *Otherwise* $\mathbf{r}$ *is a feasible descent direction.*

Proof. If $\mathbf{r} = \mathbf{0}$ and $\mathbf{u}$ is a feasible direction, then

$$\mathbf{g}^T\mathbf{u} = \left(A^T\mathbf{y}^*\right)^T\mathbf{u} = \sum_{i \in N} y_i^*\left(\mathbf{a}_i^T\mathbf{u}\right) = \sum_{i \in C^*} y_i^*\left(\mathbf{a}_i^T\mathbf{u}\right) \geq 0,$$

which means that $\mathbf{x}$ solves (1.1).

The proof of the second claim is obtained by considering the one parameter functions

$$f_i(\vartheta) = \left\| A^T(\mathbf{y}^* + \vartheta\mathbf{e}_i) - \mathbf{g} \right\|_2^2 = \|\vartheta\mathbf{a}_i + \mathbf{r}\|_2^2,$$

$i = 1, \ldots, t$, where $\mathbf{e}_i$ denotes the ith column of the $t \times t$ unit matrix. If $i \in E$, then $\vartheta = 0$ minimizes $f_i(\vartheta)$. Similarly, for $i \in C^*$ this point minimizes $f_i(\vartheta)$

subject to the bound $y_i^* + \vartheta \geqslant 0$. The necessary conditions for minimizing $f_i(\vartheta)$ imply, therefore,

$$\mathbf{a}_i^T\mathbf{r} = 0 \quad \text{when} \quad i \in E,$$

$$\mathbf{a}_i^T\mathbf{r} = 0 \quad \text{when} \quad i \in C^* \text{ and } y_i^* > 0,$$

$$\mathbf{a}_i^T\mathbf{r} \geqslant 0 \quad \text{when} \quad i \in C^* \text{ and } y_i^* = 0.$$

These relations indicate that $\mathbf{r}$ is a feasible direction,

$$(\mathbf{y}^*)^T A\mathbf{r} = 0,$$

and

$$\mathbf{g}^T\mathbf{r} = \left(A^T\mathbf{y}^* - \mathbf{r}\right)^T\mathbf{r} = -\mathbf{r}^T\mathbf{r} < 0. \qquad \blacksquare$$

The above proof provides a constructive way to derive Farkas' lemma and other related theorems of the alternative. For a detailed discussion of this topic see Dax (1985). The following result indicates that $\mathbf{r}$ points at the steepest descent direction.

COROLLARY 1. *If* $\mathbf{r} \neq \mathbf{0}$, *then* $\mathbf{r}/\|\mathbf{r}\|_2$ *solves the problem*

$$\text{minimize} \quad \mathbf{g}^T\mathbf{u} \tag{2.5}$$

$$\text{subject to} \quad \mathbf{a}_i^T\mathbf{u} = 0 \quad \textit{for } i \in E,$$

$$\mathbf{a}_i^T\mathbf{u} \geqslant 0 \quad \textit{for } i \in C^*,$$

$$\textit{and} \quad \mathbf{u}^T\mathbf{u} = 1.$$

The proof of this corollary is obtained by verifying that the Kuhn-Tucker optimality conditions are satisfied. The next corollary may avoid the need to solve (2.4) in order to ensure feasibility.

COROLLARY 2. *Let* $\tilde{\mathbf{y}}$ *solve* (2.3). *If* $\mathbf{g} = A^T\tilde{\mathbf{y}}$ *and* $\tilde{\mathbf{y}}$ *satisfies the bounds*

$$y_i \geqslant 0 \qquad \textit{for} \quad i \in C^*, \tag{2.6}$$

then $\mathbf{x}$ *solves* (1.1).

If the rows of A are linearly dependent, then $\mathbf{x}$ is called *degenerate*. Otherwise, when the rows of A are linearly independent, $\mathbf{x}$ is called *nondegenerate*. The assumption of nondegeneracy provides an alternative way to compute a feasible descent direction at dead points.

THEOREM 3. *Let $\mathbf{x}$ be a nondegenerate dead point such that $\tilde{\mathbf{y}}$, the unique solution of* (2.3), *violates* (2.6). *Let $i \in C^*$ be an index such that $\tilde{y}_i < 0$. Let $\hat{A}$ denote the $(t-1)\times n$ matrix which is obtained from A by deleting its ith row. Let $\hat{\mathbf{y}} \in \mathbb{R}^{t-1}$ solve the problem*

$$\text{minimize} \quad \|\hat{A}^T\mathbf{y} - \mathbf{g}\|_2^2, \tag{2.7}$$

and let $\mathbf{r} = \hat{A}^T\hat{\mathbf{y}} - \mathbf{g}$ denote the corresponding residual vector. Then $\mathbf{r}$ is a feasible descent direction.

Proof. The possibility $\mathbf{r} = \mathbf{0}$ is excluded by the following argument. If $\mathbf{r} = \mathbf{0}$, then the relation $\mathbf{g} = A^T\tilde{\mathbf{y}} = \hat{A}^T\hat{\mathbf{y}}$ contradicts the nondegeneracy assumption. It is also clear that $\hat{A}\mathbf{r} = \mathbf{0}$, which gives

$$\mathbf{g}^T\mathbf{r} = \left(\hat{A}^T\hat{\mathbf{y}} - \mathbf{r}\right)^T\mathbf{r} = -\mathbf{r}^T\mathbf{r} < 0.$$

Similarly, the equality $\mathbf{g} = A^T\tilde{\mathbf{y}}$ implies

$$\mathbf{a}_i = \frac{\mathbf{g} - \sum_{j\in\hat{N}}\tilde{y}_j\mathbf{a}_j}{\tilde{y}_i},$$

where $\hat{N} = N - \{i\}$. Therefore

$$\mathbf{a}_i^T\mathbf{r} = \frac{\mathbf{g}^T\mathbf{r}}{\tilde{y}_i} = -\frac{\mathbf{r}^T\mathbf{r}}{\tilde{y}_i} > 0. \qquad \blacksquare$$

3. MOVING AWAY FROM A DEAD POINT

The discussion in the previous section suggests the following alternatives. In both cases it is assumed that $\mathbf{g} = A^T\mathbf{y}$ while (2.6) is violated.

STRATEGY A. Solve (2.4) and use $\mathbf{r}$ as a search direction.

STRATEGY B. If there is a clear indication of degeneracy (e.g. $t > n$), apply Strategy A. Otherwise, when $\mathbf{x}$ is assumed to be a nondegenerate dead point, solve (2.7) and use $\mathbf{r}$ as a search direction. If, however, the nondegeneracy assumption is false, and, $\mathbf{r} = \mathbf{0}$, then use Strategy A.

The motivation behind Strategy B lies in the observation that solving (2.7) is likely to require less work than solving (2.4). Furthermore, if (2.4) is solved, then it is possible that many constraints are deleted from the active set at one step. On the other hand, there are many problems in which it is known in advance that the optimal point is at a vertex of the feasible region, i.e. a point where n linearly independent constraints are satisfied exactly. In such a case the deletion of constraints is likely to be followed by at least the same number of "adding" iterations, which implies that deleting one constraint at a time is a better strategy.

4. THE ALGORITHM

The basic iteration of the proposed active set method is composed of the following three steps.

Step 1: Compute a search direction. Let $\mathbf{x}$ denote the current feasible point. Solve (2.3) for $\tilde{\mathbf{y}}$ and compute $\mathbf{r}$. If $\mathbf{r} \neq \mathbf{0}$, set $\mathbf{u} = \mathbf{r}$ and skip to step 3.

Step 2: Moving away from a dead point (Strategy A). If $\tilde{\mathbf{y}}$ satisfies (2.6), terminate. In this case $\mathbf{x}$ solves (1.1). Otherwise solve (2.4) for $\mathbf{y}^*$ and compute $\mathbf{r}$. If $\mathbf{r} = \mathbf{0}$, terminate. In this case $\mathbf{x}$ solves (1.1). Otherwise set $\mathbf{u} = \mathbf{r}$.

Step 3: The line search. Set the new point to $\mathbf{x} + \lambda\mathbf{u}$, where λ is the largest positive number that keeps this point feasible, i.e.

$$\lambda = \min\left\{ -\frac{\mathbf{a}_i^T\mathbf{x} - b_i}{\mathbf{a}_i^T\mathbf{u}} \,\middle|\, i \in C - C^* \text{ and } \mathbf{a}_i^T\mathbf{u} < 0 \right\}.$$

If the above set is empty, terminate. In this case the objective function is not bounded below in the feasible region.

The implementation of Strategy B is achieved by inserting the corresponding changes in step 2. The finite termination of the algorithm is a consequence of the following properties:

(a) The objective function is strictly decreasing at each iteration.

(b) Each time that $\mathbf{x}$ is not a dead point the number of active constraints increases. Therefore, since there are a finite number of inequality constraints,

m say, it is not possible to have a succession of more than m iterations in which step 2 is skipped.

(c) Each time that step 2 is executed, the current point is a minimizer of the objective function on the linear manifold

$$\left\{ \mathbf{z} \middle| \mathbf{a}_i^T \mathbf{z} = b_i \text{ for all } i \in N \right\}.$$

(d) There are a finite number of such manifolds.

A straightforward way to implement the algorithm for small dense problems is to solve (2.3) and (2.7) via the QR factorization of A^T and $\hat{A}^T$. The updating of this factorization when a row is added to or deleted from A is explained in Gill and Murray (1973). In our case it is advantageous to order the rows of A so that the first rows correspond to equality constraints. This way the QR factorization of these rows remains unchanged throughout the minimization process. Also, as a by-product, this factorization may help us to exclude redundant equality constraints. In the next section we show that the algorithm for solving (2.4) may apply the same factorization scheme as the main algorithm.

5. THE BOUNDED LINEAR LEAST SQUARES PROBLEM

This section describes an active set method for solving (2.4) whose basic iteration is similar to that of the main algorithm. The search direction is obtained by solving a modified form of (2.3), using the same factorization scheme as in the main algorithm. This way, the factorization of A^T continues and the algorithm is easily incorporated into the main one.

Let $\mathbf{y} \in \mathbb{R}^t$ satisfy the constraints (2.6). Then the ith variable of $\mathbf{y}$ is said to be *bounded* if $i \in C^*$ and $y_i = 0$. Otherwise it is called *free*. The point $\mathbf{y}$ defines a diagonal matrix $D = \operatorname{diag}\{d_1, \ldots, d_t\}$ by the following rule: If the ith component of $\mathbf{y}$ is bounded, $d_i = 0$; otherwise $d_i = 1$. The number of free components is denoted by s, and the $s \times n$ matrix whose rows correspond to free components is denoted by $\bar{A}$. The basic iteration of the proposed method is composed of the following three steps.

Step 1. Compute a search direction. Let $\mathbf{y}$ denote the current feasible point, and let $\mathbf{r} = A^T\mathbf{y} - \mathbf{g}$ denote the current residual vector. The search direction, $\mathbf{v}$, is obtained by solving the problem

$$\begin{aligned} &\text{minimize} \quad S(\mathbf{v}) = \tfrac{1}{2}\|A^T\mathbf{v} - \mathbf{r}\|_2^2 \\ &\text{subject to} \quad D\mathbf{v} = \mathbf{v}. \end{aligned} \tag{5.1}$$

Clearly, if $\mathbf{r} = \mathbf{0}$ or $s = 0$, then there is no need to solve (5.1). In this case continue to step 2. If the resulting solution satisfies $S(\mathrm{v}) < \frac{1}{2}\|\mathbf{r}\|_2^2$, then skip to step 3; otherwise continue to step 2. The aim of the last precaution is to ensure that the objective function will decrease at each iteration.

Step 2: Moving away from a dead point. Compute $\mathbf{h} = A\mathbf{r}$, the gradient vector of the objective function at the current point. Then $\mathbf{h}$ is used to obtain a further vector, $\mathbf{h}^* \in \mathbb{R}^t$, by the following rule: If y_i is bounded and $h_i > 0$, set $h_i^* = 0$. Otherwise set $h_i^* = h_i$. The vector $\mathbf{h}^*$ enables us to test whether the Kuhn-Tucker optimality conditions hold at y. If $\mathbf{h}^* = 0$, then y solves (2.4) and the algorithm terminates. Otherwise a feasible descent direction, v, is obtained as follows: Compute an index j such that

$$|h_j^*| = \max\{|h_i^*|,\ i = 1,\dots,t\},$$

and set

$$\mathrm{v} = -\frac{h_j}{\mathbf{a}_j^T\mathbf{a}_j}\mathbf{e}_j, \tag{5.2}$$

where $\mathbf{e}_j$ denotes the jth column of the $t \times t$ unit matrix.

Step 3: The line search. Set the new point to $\mathrm{y} + \rho\mathrm{v}$ where ρ is the largest number in the interval $[0,1]$ that keeps this point feasible. If $\rho = 1$ and v was computed by (5.1), then the next iteration should start at step 2.

The finite termination of the above algorithm is proved by standard arguments. The solution of (5.1) is carried out by using the QR factorization of $\bar{A}^T$ to solve the unconstrained problem

$$\underset{w \in \mathbb{R}^s}{\text{minimize}}\|\bar{A}^T w - \mathbf{r}\|_2^2. \tag{5.3}$$

The updating of this factorization, when a row is added to or deleted from $\bar{A}$, can be done exactly as in the main algorithm.

The initial point, y, is obtained from $\tilde{\mathrm{y}}$ by the following rule: If $i \in C^*$ and $\tilde{y}_i < 0$, set $y_i = 0$. Otherwise set $y_i = \tilde{y}_i$. This way the initial $\bar{A}$ is obtained from A by deleting those rows for which $\tilde{y}_i$ violates its bound. Hence the QR factorization of $\bar{A}^T$ is easily obtained from that of A^T. Similarly, the final matrix $\bar{A}$ provides the new matrix A for the main algorithm, and the factorization of this matrix is available.

6. FURTHER REMARKS

If the rows of A are linearly dependent, then $\tilde{\mathbf{y}}$, the solution of (2.3), is not unique. In such a case we apply the following strategy: Let ρ denote the rank of A. Then the last $t - \rho$ components of $\tilde{\mathbf{y}}$ are set to be zero. The reason for this rule is the assumption that the QR factorization transforms the first ρ columns of A^T into a $n \times \rho$ upper triangular matrix whose rank is ρ. Hence the above rule fixes $\tilde{\mathbf{y}}$ in a definite way. This strategy is advantageous when $t > n$, since the QR factorization can be restricted to the first n columns of A^T. A similar strategy is applied in the solution of (5.3). This way the first ρ rows of A form the *working set*, while the solution of (2.4) provides an effective way for determining the right working set at degenerate dead points.

For large sparse problems the solution of (2.3) and (2.4) should be done by methods which are able to take advantage of the special structure of these problems. This is demonstrated in Dax (1986a), where a problem analogous to (2.4) is solved by a relaxation technique.

The computation of a feasible initial point for the main algorithm can be done by a similar algorithm (see Dax, 1985). The only differences are that here the objective function is

$$F(\mathbf{x}) = \sum_{i \in V} \left(b_i - \mathbf{a}_i^T \mathbf{x} \right),$$

and

$$\mathbf{g} = -\sum_{i \in V} \mathbf{a}_i,$$

where

$$V = \left\{ i \mid i \in C \text{ and } \mathbf{a}_i^T \mathbf{x} < b_i \right\}.$$

7. NUMERICAL RESULTS

This section presents some preliminary trials with the new algorithm. The first type of test problems consists of "random" problems that have the following structure. The components of $\mathbf{g}$ are random numbers from the interval $[-1, 1]$. (The random number generator is of uniform distribution.) The components of the equality constraints are generated in the same way,

TABLE 1
RESULTS OF "RANDOM" TEST PROBLEMS

	Strategy A			Strategy B		
m	Major iterations	Solving (2.4)	Minor iterations	Major iterations	Solving (2.4)	Minor iterations
5	15.9	0.6	1.3	15.9	0	0
10	16.4	0.9	1.3	16.5	0	0
20	18.2	1.6	3.8	18.9	0	0
40	22.4	2.8	12.6	26.4	0	0

while $b_i = \sum_{j=1}^{n} a_{ij}$ for $i \in E$. The inequality constraints include $2n$ simple bounds of the form $-10 \leqslant x_j \leqslant 10$, $j = 1, \ldots, n$, and m "random" constraints whose components (including b_i) are random numbers from the interval $[-1, 1]$. The starting point is $\mathbf{e} = (1, 1, \ldots, 1)^T \in \mathbb{R}^n$. In order to ensure that this point is feasible, we check the "random" inequalities and multiply by -1 those which are violated at this point. All the experiments were done with $n = 20$ and five equality constraints. The number of "random" inequalities takes the values $m = 5, 10, 20, 40$. For each value of m we have generated and solved ten different problems of this type. The results of the "random" test problems are presented in Table 1. The figures in this table are, therefore, average numbers. The columns headed "Major iterations" give the number of iterations made by the main algorithm. The columns headed "Solving (2.4)" provide the number of major iterations in which problem (2.4) was solved. The columns headed "Minor iterations" give the overall number of iterations made by the active set method for solving (2.4).

The second type of test problems consists of "degenerate" problems. These problems are obtained from the "random" problems by adding $m(m-1)/2$ redundant constraints. Let $\mathbf{a}_i^T \mathbf{x} \geqslant b_i$, $i = 1, \ldots, m$, denote the

TABLE 2
RESULTS OF "DEGENERATE" TEST PROBLEMS

	Strategy A			Strategy B		
m	Major iterations	Solving (2.4)	Minor iterations	Major iterations	Solving (2.4)	Minor iterations
5	16.0	1.4	3.8	16.0	1.4	3.8
6	16.3	1.7	4.5	16.3	1.7	4.5
7	15.9	1.7	5.1	15.9	1.7	5.1
8	16.1	1.7	8.7	16.1	1.7	8.7

"random" inequality constraints. Then the redundant constraints have the form $(\mathbf{a}_i + \mathbf{a}_j)^T\mathbf{x} \geqslant b_i + b_j$, where $1 \leqslant i < j \leqslant m$. In these problems m takes the values $m = 5, 6, 7, 8$. As before, for each value of m we have generated and solved ten different problems of this type. The results of the "degenerate" test problems are given in Table 2.

The comparison of Strategies A and B reveals that the second strategy is advantageous in solving "random" test problems, while for "degenerate" problems there is no difference between the two options.

8. CONCLUDING REMARKS

The results of our experiments are quite encouraging. It seems that the new method compares favorably with other methods. It is illustrated that the number of "minor" iterations needed to solve (2.4) is usually small. The maximum number that was recorded is 8, while the average number is about 4. This indicates that the proposed algorithm for solving (2.4) provides an efficient way to resolve degeneracy.

Another feature that distinguishes the new method is that there is no need to transform the constraints into a standard form. Moreover, the number of constraints need not exceed the number of variables, and the current feasible point need not be a vertex. In fact, the computational effort per iteration is reduced as the number of active constraints decreases. Therefore the method is especially attractive for problems with few constraints.

The potential value of the new approach lies in the solution of large sparse problems where the least squares subproblems can be solved efficiently by iterative methods such as relaxation or conjugate gradients.

The ability to compute the steepest descent direction gives rise to the steepest descent method for solving (1.1). This idea is investigated in Dax (1987).

Finally we note that similar methods have been constructed for the analogous l_1 and l_∞ problems (see Dax 1986b, 1986c, respectively). The present algorithm can be viewed, therefore, as a special case of a general method for minimizing polyhedral convex functions subject to linear constraints.

REFERENCES

Bartels, R. H. 1980. A penalty linear programming method using reduced-gradient basis-exchange techniques, *Linear Algebra Appl.* 29:17–32.

Conn, A. R. 1976. Linear programming via a nondifferential penalty function, *SIAM J. Numer. Anal.* 13:145–154.

Dax, A. 1985. The Computation of Descent Directions at Degenerate Points, Tech. Rep., Hydrological Service of Israel.

Dax, A. 1986a. An efficient algorithm for solving the rectilinear multifacility location problem, *IMA J. Numer. Anal.* 6:343–355.

Dax, A. 1986b. The l_1 Solution of Linear Equations Subject to Linear Constraints, Tech. Rep., Hydrological Service of Israel, (To appear in *SIAM J. on Sci. and Stat. Comp.*).

Dax, A. 1986c. The minimax solution of linear equations subject to linear constraints, Tech. Rep., Hydrological Service of Israel, (To appear in *IMA J. Numer. Anal.*).

Dax, A. 1987. The steepest descent method for minimizing polyhedral convex functions, Tech. Rep., Hydrological Service of Israel.

Fletcher, R. 1981. *Practical Methods of Optimization, Vol 2: Constrained Optimization*, Wiley, Chichester.

Gill, P. E. and Murray, W. 1973. A numerically stable form of the simplex algorithm, *Linear Algebra Appl.* 7:99–138.

Osborne, M. R. 1985. *Finite Algorithms in Optimization and Data Analysis*, Wiley, Chichester, 1985.

Received 22 December 1986; final manuscript accepted 21 April 1988