DCS-TR-97

May 1981

LINEAR PROGRAMMING VIA DISCRETE L_1 CURVE-FITTING

by

W. L. Steiger

DCS-TR-97

Department of Computer Science
Rutgers University
Hill Center
Busch Campus
New Brunswick, N.J. 08903

Title: Linear Programming via Discrete L_1 Curve-Fitting

Author: W. L. Steiger(1)
Department of Computer Science
Rutgers University
New Brunswick, N.J.

Running Title: Linear Programming via Discrete L_1 Curve-Fitting

Keywords & Phrases: Linear Programming, LAD, LAD Curve-fitting.

AMS(MOS) Classification: Primary: 90C05
Secondary: 65D10, 65D15, 62J05

(1) This work was performed while the author held a visiting fellowship in the Department of Statistics, Institute of Advanced Studies, The Australian National University.

ABSTRACT

A bounded linear programming problem with feasible solutions may be cast as a discrete L_1 curve-fitting problem of the same size. This may be usefully exploited in solving dense LP problems: On the average, a recent L_1 algorithm solves the equivalent L_1 curve fit in far fewer steps, and taking far less time, than that which would be required by the one-phase Simplex method applied to the original problem. The relative advantage increases with problem size and the comparison is even more favorable against the two-phase Simplex method. Finally, the Klee-Minty problems on which the Simplex method is of exponential complexity, seem to be "easy" problems as equivalent L_1 curve fits.

I. INTRODUCTION

One of the most interesting aspects of the Simplex method for linear programming (LP) problems is the mystery as to why it is usually so efficient. Kuhn and Quandt [16] and later Avis and Chvatal [2] have a large body of Monte-Carlo evidence suggesting that the average number of Simplex steps is not far from linear, as a function of the number of constraints. This is borne out as well by a large selection of "real world" examples (see e.g., [11]), and Dantzig [10] has actually proved a statement along these lines, for LP problems that arise according to a certain family of probability distributions.

Klee and Minty [15] have shown that Simplex can be of exponential complexity but given the above evidence, these examples must be, in some real sense, extremely rare. Although Khachian [14] has shown that LP problems are of no more than fourth degree complexity as a function of problem size, there is evidence that his algorithm is less efficient than Simplex, "on the average" (see [12]). One hopes therefore for a method of solving LP problems with both a better worst case behavior than that possessed by Simplex and which is also faster on the average.

Along the latter lines, A. Conn [9], and later R. Bartels [6], have presented a penalty function approach that seems superior, on the average, to the Simplex method. However, these studies give no insight into worst case behavior.

In the present paper a different alternative is considered. LP problems are converted into equivalent discrete L_1 curve fits which are solved directly. The solutions are then translated back to those of the original LP problem. The details of this process are briefly presented in the next section and, to render the treatment self-contained, the L_1 algorithm proposed for this method is also described. In Section 3 the Simplex method is compared to the current technique on randomly generated, dense LP problems. The results indicate that it is more efficient to solve the equivalent L_1 than to use Simplex - in both time and iterations - and that the advantage increases with problem size.

The situation for dense problems has a definite bearing on the LP problems of practical interest, large sparse ones. If, as is the case with Simplex, the L_1 algorithm were implemented to exploit sparsity in large problems, the L_1 equivalent could have practical importance in linear programming. This is the main motivation behind the current work.

As regards worst case behavior, it is interesting that the Klee-Minty type problems are not extreme when solved as equivalent L_1 curve fits. This fact also appears in Section 3. Whether some other type of LP would be "difficult" as an equivalent L_1 curve fit or indeed, whether there exist exponential problems for the better L_1 algorithms, is not known.

II. THE EQUIVALENCE OF L_1 AND LP AND AN L_1 ALGORITHM

There is an intimate relationship between linear programming (LP) problems and discrete L_1 curve fitting. In the latter problem one is given n points $(\underline{x}_i, y_i)$, $\underline{x}_i \in R^k$, the object being to minimize the distance function.

$$(1)\quad f(\underline{c}) = \sum_{i=1}^{n} \left| y_i - \sum_{j=1}^{k} c_j x_{ij} \right| = \sum_{i=1}^{n} \left| y_i - \langle \underline{c}, \underline{x}_i \rangle \right|$$

The optimal $\underline{c}$ describes the k dimensional hyperplane $\{(\underline{x},y): y = \langle \underline{c}, \underline{x} \rangle\}$ in R^{k+1} that fits the n points "best" in the discrete L_1 sense.

It is familiar (see [8] or [19]) that solutions of (1) also solve the primal LP problem

$$(2)\quad \text{minimize} \sum_{i=1}^{n} r_i \quad \text{subject to} \left\{ \begin{array}{l} r_i \geq y_i - \sum_{j=1}^{k} c_j x_{ij} \\ r_i \geq \sum_{j=1}^{k} c_j x_{ij} - y_i \end{array} \right\} \quad i = 1,\ldots,n .$$

By the equivalence of (1) and (2) and the fact that f increases if $|\underline{c}|$ is large enough, (2) is bounded and feasible.

Since (2) has n+k variables, r_i, $i = 1,\ldots,n$, and c_j, $j = 1,\ldots,k$, and 2n constraints, one would not expect the Simplex method applied to (2) to provide an efficient way to solve (1). But by the duality theorem the problem

$$\text{maximize } \left[\sum_{i=1}^{n} y_i u_i - \sum_{i=1}^{n} y_i v_i\right]$$

$$(3) \quad \text{subject to} \begin{cases} u_i + v_i = 1, & i = 1,\ldots,n \\ \sum_{i=1}^{n} x_{ij} u_i - \sum_{i=1}^{n} x_{ij} v_i = 0, & j = 1,\ldots,k \\ u_i \geqslant 0, & i = 1,\ldots,n \\ v_i \geqslant 0 & \end{cases}$$

is dual to (2). Then, regarding the v_i as slacks in $u_i \leqslant 1$ and substituting $v_i = 1 - u_i$, one gets

$$\text{maximize } \sum_{i=1}^{n} y_i u_i$$

$$(4) \quad \text{subject to} \begin{cases} \sum_{i=1}^{n} x_{ij} u_i = 1/2 \sum_{i=1}^{n} x_{ij}, & j = 1,\ldots,k \\ 0 \leqslant u_i \leqslant 1, & i = 1,\ldots,n \end{cases}$$

a bounded variable dual LP problem of reasonable size. From it, the solution to (2), hence (1), may be obtained.

On the other hand an LP problem with bounded, feasible solutions gives rise to an equivalent, discrete L_1 curve fit. Suppose one is given the problem

$$\text{maximize} \quad \sum_{i=1}^{n} d_i x_i$$

$$(5) \qquad \text{subject to} \begin{cases} \sum_{j=1}^{n} a_{ij} x_j = b_i, & i = 1,\dots,m < n \\ x_j \geqslant 0, & j = 1,\dots,n \end{cases}$$

with bounded, feasible solutions. It can be solved as an L_1 fit by transforming it into the form of the problem in (4), as was first noticed, apparently, by C. Witzgoff (preprint, see [5]).

First, since there is $B>0$ such that $|x_i| \leqslant B$, after putting $z_i = x_i/B$, (5) may be written as

$$\text{maximize} \quad \sum_{i=1}^{n} d_i z_i$$

$$(6) \qquad \text{subject to} \begin{cases} \sum_{j=1}^{n} a_{ij} z_j = b_i/B, & i = 1,\dots,m \\ 0 \leqslant z_j \leqslant 1, & j = 1,\dots,n \end{cases}$$

Incorporate a new variable, z_{N+1}, into the problem to obtain

$$\text{maximize} \quad K z_{N+1} + \sum_{i=1}^{n} d_i z_i$$

$$(7) \qquad \text{subject to} \begin{cases} \sum_{j=1}^{n} a_{ij} z_j + t_i z_{n+1} = b_i/B, & i = 1,\dots,m \\ 0 \leqslant z_j \leqslant 1, & j = 1,\dots,N+1 \end{cases}$$

where K and t_i, $i = 1,\dots,m$ are constants. If $K<0$ is chosen small enough, that forces $z_{N+1} = 0$ in the optimal solution of (7), and the extra term $t_i z_{N+1}$ in the m equality constraints will have no effect on the solution, whatever value t_i takes. Thus (7) is equivalent to (5). Finally, if we choose $t_i = 2b_i/B - \sum a_{ij}$ the right-hand side of each equation in (7) is one half the sum of the left-hand side coefficients. This characterizes (7) as a bounded variable LP problem that arises from a discrete L_1 curve fit, as in (4). In fact (7) is obtained from the fitting problem for the points

(8) $\quad \underline{P}_i = (\underline{x}_i, y_i), \quad i = 1,\ldots,N+1$

where, for $i = 1,\ldots,n$, $x_{ij} = a_{ji}$, $j = 1,\ldots,m$, and $y_i = d_i$; $x_{n+1,j} = t_j$, $j = 1,\ldots,m$ and $y_{n+1} = K$. This is easily checked by passing through the steps from (1) - (4) beginning with (8).

If $\underline{c} \varepsilon R^m$ is the optimal fit to the points in (8), at least m of the residuals $y_i - \langle \underline{c}, \underline{x}_i \rangle$ will be zero. The sequence from (1) to (4) shows that (8) is equivalent to the dual of (5), the given LP. By complimentary slackness, when $y_i - \langle \underline{c}, \underline{x}_i \rangle \neq 0$, $x_i = 0$ in the optimal solution to (5). This gives n-m of the x_i's directly. The other m require more work as will be described later.

We can now discuss the potential efficacy of this equivalence. Given (1) one could apply the Simplex method to (4) and just read off the answers. In fact, if Khachian's algorithm were applied to (4), the solution would be obtained in time that is a 4^{th} degree polynomial function of n(2n+k), the size of (4). However this is not the most efficient route to solution.

There are three good, direct methods for LAD curve fits. Barrodale and Roberts (BR) ([3], [4]) devised an algorithm for (1) that is a modification of Simplex on an abbreviated version of (2). It is much faster than applying Simplex directly to (4). Bartels, Conn and Sinclair (BCS) produced another direct algorithm for (1) which was thought (see [5]) to be essentially of the same complexity as that of Barrodale and Roberts. Finally Bloomfield and Steiger [7] (BS) developed a third, competitive method for (1). Recently, Anderson and Steiger have shown (see [1]) that as n increases in (1) the computational cost of BR increases faster than linearly whereas the cost for the BCS algorithm and for the BS algorithm increases linearly. The cost of all algorithms increase linearly in k. They show that the BS algorithm is most efficient in that for fixed k, the complexity of BS and BCS is cn+b and that c is smaller in the case of BS. More specifically, the complexity of BCS and BS is ank+bn+ck+d, and the constant a is smaller for BS. In any case however, it is better to try to solve (1) by some direct method than to exploit the equivalence between LP and discrete L_1 fitting.

The situation is different when solving LP problems where, as the next section will demonstrate, the equivalence of (1) and (5) may be usefully exploited. The idea is to use a fast L_1 algorithm on the equivalent discrete L_1 curve fit for the points in (8) and to translate back the answers to the solution of (5), the given LP.

We use the Bloomfield-Steiger algorithm because it is much more efficient than competitors. For the sake of completeness the main ingredients of the algorithm will now be described.

A key feature is the idea of Boscovitch [13] (or later, in more generality, Singleton [17]) who considered the problem (1) when $k = 1$, when there is only a slope parameter to fit. In this case

$$f(c) = \sum_{i=1}^{n} |y_i - cx_i| = \sum_{i=1}^{n} |x_i||y_i/x_i - c|$$

and the optimal value of c is the weighted median of the ratios y_i/x_i with respect to weights $|x_i|$. It may be obtained by sorting the ratios so that $y_i/x_i \leqslant y_{i+1}/x_{i+1}$ and then finding the smallest integer $p \geqslant 1$ such that

$$\sum_{i=1}^{p} |x_i| \geqslant \sum_{i=1}^{n} |x_i|/2$$

Now $c = y_p/x_p$ is the weighted median, the optimal fit is a line through the origin and the point $(\underline{x}_p, y_p)$, and the p^{th} residual $(y_p - cx_p)$ is zero.

In the general case if r is the rank of $X = (\underline{x}_i) = (x_{ij})$, the optimal fit $\underline{c}$ describes a hyperplane containing at least r of the $(\underline{x}_i, y_i)$ and the corresponding r residuals $y_i - \langle \underline{c}, \underline{x}_i \rangle$ vanish in (1). For simplicity assume $r = k$, that $\underline{x}_{i_1}, \ldots, \underline{x}_{i_k}$ constitutes a linearly independent set of rows of X, and that the current fit $\underline{c}$ has been determined so $y_i = \langle \underline{c}, \underline{x}_i \rangle$, $i = i_1, \ldots, i_k$. The basic iteration step of the algorithm heuristically chooses one of these k points for deletion and replaces it in an optimal way by one of the $n - k$ excluded points.

The new set of k points thus determined is contained in a hyperplane $\underline{c}'$ and the process continues.

Specifically, suppose first that $\underline{x}_q$ has been selected for deletion and write $B = \{i : i \neq q, i = i_1,\ldots,i_k\}$, the set of indices of the remaining k - 1 points. Choose $\underline{d} \in R^k$ so that $\langle \underline{d},\underline{x}_i\rangle = 0$ for i in B, as may be easily done. Then

$$(9) \qquad \underline{g}_t = \underline{c} + t\underline{d}, \quad t \in R$$

describes a one parameter family of hyperplanes containing $(\underline{x}_i, y_i)$, $i \in B$, because $\langle \underline{g}_t,\underline{x}_i\rangle = \langle \underline{c},x_i\rangle + t\langle \underline{d},\underline{x}_i\rangle = y_i$, and when $t = 0$, $\underline{g}_t$ is the current fit. In general,

$$(10) \qquad \begin{aligned} f(\underline{g}_t) &= \sum_{i=1}^{n} |y_i - \langle \underline{g}_t,\underline{x}_i\rangle| \\ &= \sum_{i \notin B} |y_i - \langle \underline{g}_t,\underline{x}_i\rangle| \\ &= \sum_{i \notin B} |y_i - \langle \underline{c},\underline{x}_i\rangle - t\langle \underline{d},\underline{x}_i\rangle| \\ &= \sum_{i \notin B} |w_i - tv_i| \end{aligned}$$

where the notation $w_i = y_i - \langle \underline{c},\underline{x}_i\rangle$ and $v_i = \langle \underline{d},\underline{x}_i\rangle$ has been used. From the foregoing discussion (10) is minimized when $t = \hat{t}$, the weighted median of the ratios w_i/v_i; thus $\hat{t} = w_p/v_p$, say. The new fit is then

$$\underline{c}' = \underline{g}_{\hat{t}} = \underline{c} + \hat{t}\underline{d}$$

which contains $(\underline{x}_i, y_i)$, $i \in B$ and the new point $(\underline{x}_p, y_p)$. Finally the iteration has not decreased the value of f because

$$f(\underline{c}') = f(\underline{g}_{\hat{t}}) < f(\underline{g}_0) = f(\underline{c}).$$

We emphasize that once $\underline{x}_q$ is chosen for deletion, $\underline{d}$ is obtained and then $\underline{c}'$ is determined by a weighted median calculation.

It now remains to describe the choice of the point to be deleted. The method is a heuristic one, also based on weighted medians. Suppose the current fit $\underline{c}$ contains the points $(\underline{x}_i, y_i)$, $i = i_1, \ldots, i_k$. If point i_1 were chosen for deletion, the replacement procedure just outlined would deal with the one parameter family $\underline{g}_t$ in (9), where $\underline{d}$ is orthogonal to the k - 1 points $\underline{x}_i$, $i = i_2, \ldots, i_k$.

It would choose that member $\underline{g}_t$, where t is the weighted median of w_i/v_i with weights $|v_i|$, as defined after (10). This means that t is the median of the distribution function

$$(11) \qquad F(t) = \sum_A |v_i| / \sum_{i=1}^{n} |v_i|, \quad A = \{i : w_i/v_i \leqslant t\}$$

i.e. the smallest t which satisfies $F(t) \geqslant 1/2$. Since the current fit is the value t = 0 in the one parameter family $\underline{g}_t$, $|F(0) - 1/2|$ is the distance - in terms of F - of the current fit from the next fit, assuming $\underline{x}_{i_1}$ is deleted. It purports to capture the relative advantage of replacing $\underline{x}_{i_1}$ optimally. Using $|F(0)-1/2|$ to measure $|g_0 - g_{\hat{t}}|$ has the nice property of eliminating scaling differences between the families $\{g_t\}$ as each $\underline{x}_{i_1},\ldots,\underline{x}_{i_K}$ is considered in turn for deletion.

Actually if $F(0^+) \geqslant 1/2$ and $F(0^-) < 1/2$ $t = 0$ is already the median of F and $\underline{x}_i$ should not be replaced. Excluding this case the criterion then becomes

$$(12)\qquad \begin{array}{lll} 1/2 - F(0) & \text{if} & F(0) \leqslant 1/2 \\ F(0^-) - 1/2 & \text{if} & F(0^-) > 1/2 \end{array}$$

which may be written more succinctly as

$$(13)\quad \Big(\Big|\sum_P |v_i| - \sum_N |v_i|\Big| - \sum_Z |v_i|\Big)/\Big(2\sum_1^n |v_i|\Big)$$

where P is the set of indices for which $w_i/v_i > 0$, N the set where $w_i/v_i < 0$ and Z, the set where $w_i/v_i = 0$.

The criterion in (13) is evaluated for each point $\underline{x}_{i_1},\ldots,\underline{x}_{i_K}$ of the current fit and the point with the largest positive value is selected for deletion. If there are none, the algorithm terminates and the optimal $\underline{c}$ has been obtained.

It may be useful to observe that the criteria in (13) is related to gradients: differentiating (10) with respect to t gives the directional derivative of f in the direction $\underline{d}$ that corresponds to deleting $\underline{x}_q$. At t=0, the right-hand derivative is

$$-\sum_P |v_i| + \sum_N |v_i| + \sum_Z |v_i|$$

and the left-hand derivative is

$$-\sum_P |v_i| + \sum_N |v_i| - \sum_Z |v_i|.$$

The numerator of (13) is thus the maximum of the left-hand derivative and minus the right-hand derivative of f at t=0.

Once the optimal fit to the points in (8) has been obtained, m residuals $y_i - \langle \underline{c}, \underline{x}_i \rangle = 0$. By duality, the values for the corresponding variables x_i in (5), are obtained from the gradients as computed in (13).

The algorithms of BR, BCS and BS all iterate by replacing a point $\underline{x}_q$ in the current basis with an optimal point $\underline{x}_p$ not in the basis. BCS and BS explicitly calculate $\underline{x}_p$ as a weighted median, the former using a heapsort of the ratios w_i/v_i, the latter using a special-purpose weighted median partial quicksort.

All algorithms select the point $\underline{x}_q$ for deletion in a heuristic fashion but only BS uses the weighted median idea embodied in (13). The efficacy of these choices may be reflected in the superior performance of BS [1].

III. COMPUTATIONAL EXPERIENCE

The equivalence between bounded, feasible LP problems and LAD curve fits seems to have potential utility. The Simplex Method was compared to the BS algorithm applied to the equivalent L_1 fits. The comparisons were made on a variety of randomly generated problems and for other cases as well. Initial indications imply that on dense problems, it is better to transform (5) to (8), solve it using BS, and then obtain the solution to (5): fewer iterations are required and the computing time is less than what would be required by Simplex. Although the stability and accuracy of the L_1 equivalent was not compared to that of various Simplex implementations, there is so far no evidence to suggest the presence of any undesirable numerical properties, but large problems have not been studied yet.

First, the Klee-Minty problems are "easy" as equivalent LAD fits. The standard one-phase Simplex algorithm on the problem

$$\text{maximize} \quad \left[\sum_{i=1}^{n} 10^{n-i}x_i\right]$$

$$\text{(14)} \qquad \text{subject to} \left\{ \begin{array}{l} \left(2\sum_{j=1}^{i-1} 10^{i-j}x_j\right) + x_i \leq 100^{i-1} \\ x_i \geq 0 \end{array} \right\} \quad i = 1,\ldots,n$$

would require 2^n-1 iterations using the slack variables as an initial, feasible basis. When cast as a LAD fit, the BS algorithm would always move to the optimal solution in one iteration! This property is shared by the Simplex method using the "largest increase rule" for selecting the entering variable. However problems which cause that algorithm to work exponentially hard are still only "average" as LAD fits.

Next, problems of the form

$$\text{(15)}\qquad \text{maximize } [x_1+\dots+x_n] \quad \text{subject to} \begin{cases} \sum_{j=1}^{n} a_{ij}x_j \leqslant 10{,}000, & i=1,\dots,m \\ x_j \geqslant 0 \end{cases}$$

were generated at random along the line of [2] and [16]. For a single realization, mn random integers a_{ij} were generated. In the previous work they were uniform on [1,1000]. However van Dam and Telgen [18] suggest that as n increases, the faces which correspond to $\underline{Ax} \leqslant \underline{b}$ in the random polytope thus generated, tend to be orthogonal, quite a non-random result. Thus, in the present Monte-Carlo, each a_{ij} is the average of 10 uniformly distributed integers in [1,1000]. Supposedly, the angle between polytope faces will now be more nearly uniformly distributed.

Given an instance of (15) it was first solved by the one-phase Simplex method, starting from an initial feasible basis composed of the slack variables. The number of Simplex steps and the CPU time, net of generating the instance, were recorded.

On the same instance of (15), the equivalent LAD fit, (8) was obtained, solved using BS, and the solutions transformed back to those of (15). The number of LAD iterations was recorded along with the total CPU time for the task, net of the time needed to generate (15).

For a given problem size mn, 10 independent instances were solved and the performance measures were averaged. The results appear in Table 1.

[Insert TABLE 1 Here]

The iteration counts accord closely with those reported by Avis and Chvatal. The LAD advantage seems to increase as n does. Timings have not been included due to the possibility that differences between computers, systems, and actual coding, could alter any results.

However one knows that each Simplex iteration requires on the order of mn algebraic operations. The BS algorithm is operating on a data structure, (8), of size m(n+1) and thus requires on the order of mn+WM(n) operations, WM denoting the complexity of a weighted median calculation, on the average linear in n. Thus timings could be expected to accord with the complexity results presented in Table 1.

This was in fact the case for the experiments in Table 1. Using a simple FORTRAN code for the table form of the Simplex method, the timings corresponding to the results presented in Table 1 give an advantage to LAD of 8% for m=5, n=10 to 30% for m=10, n=50.

In the foregoing experiments an initial feasible solution was presented to the Simplex algorithm. In the general case, the two-phase

Simplex method would be required and would force Simplex to work harder on problems of the same size. It would have no effect on the LAD equivalent.

Thus in a second set of experiments, problems (15) were solved by both the two-phase Simplex method and as LAD equivalents. The former began with a phase-one initial basis composed of artificial variables. The results appear in Table 2.

[Insert TABLE 2 Here]

The LAD results for the new set of problems agree closely with those shown in Table 1 and give a rough idea of the variability of performance statistics such as average iteration counts. If 100 instances had been averaged, the variability would have been reduced.

Clearly less work is done by BS on the LAD equivalent than by the Simplex method. On the larger problems, only 1/3 as many iterations were required.

More Simplex iterations seem to be required in the present study than in the one reported in Bartels [6]. This could be due to the different characteristics of the random polytopes generated in the two experiments, as mentioned earlier. In any case however, the LAD iteration counts in Table 2 are still much smaller than the counts for Bartels' penalty LP method on problems of the same size. For instance on a system with 10 inequalities and 30 variables, 11.5 LAD iterations were required; the penalty LP method took 21 steps ($m=10$ in Table 2 of [6]).

Again, while CPU timings may be misleading, the results for Table 2 accord with a "per iteration" comparison of the algorithms in terms of algebraic operations. The LAD runs were from 1 1/4 to 3 times faster than the two phase Simplex runs.

The computational results support a continued interest in solving LP problems via discrete L_1 curve fitting. In general, it would be interesting to understand - in terms of Simplex steps on a given LP problem - what a single iteration of, say, BS does on the LAD equivalent. This could possibly help explain the apparent efficacy of the latter algorithm. Secondly, there is the suggestion that the ideas in BS, if implemented so as to exploit sparsity, would be useful in large LP codes.

References

[1] Anderson, D. and Steiger, W.L. (1980). "A comparison of Methods for Discrete L_1 Curve Fitting", Department of Computer Science Technical Report 96, Rutgers University

[2] Avis, D. and Chvatal, V. (1978). "Notes on Bland's Pivoting Rule", Mathematical Programming Study, 8, 24-34.

[3] Barrodale, I. and Roberts, F.D.K. (1973). "An Improved Algorithm for Discrete L_1 Linear Approximation", Siam. J. Numer. Anal. 10., 839-848

[4] Barrodale, I. and Roberts, F.D.K. (1974). "Algorithm 478: Solution of an Overdetermined System of Equations in the L_1 Norm", Comm. A.C.M. 14, 319-320

[5] Bartels, R., Conn, A. and Sinclair, James W. (1978). "Minimization Techniques for Piecewise Differentiable Functions: the L_1 Solution to an Overdetermined System, Siam. J. Numer. Anal 15, 224-241

[6] Bartels, R. (1980). "A Penalty Linear Programming Method Using Reduced Gradient Basis-Exchange Techniques", Linear Algebra and Applications 29, 17-32

[7] Bloomfield P. and Steiger, W.L. (1980). "Least Absolute Deviations Curve-Fitting", Siam. J. Scientific and Statistical Comp. 1, 290-301.

[8] Charnes, A., Cooper, W.W. and Ferguson, R.O. (1955). "Optimal Estimation of Executive Compensation by Linear Programming", Mgt. Sci. 1, 138-151

[9] Conn, A.R. (1976). "Linear Programming via a Non-differentiable Penalty Function", Siam. J. Numer. Anal. 13, 145-154

[10] Dantzig, G.B. (1980). "Expected Number of Steps of the Simplex Method for a Linear Program with a Convexity Constraint ", Technical Report SOL 80-3, Dept. of Operations Research, Stanford University

[11] Dantzig, G.B. (1963). Linear Programming and Extensions, Princeton University Press, Princeton, N.J.

[12] Dantzig, G.B. (1980). "Khachians Algorithm: A Comment". Siam. News 13.

[13] Eisenhart, C. (1961). "Boscovitch and the Combination of Observations" In Roger Joseph Boscovitch, L.L. Whyte, Ed. New York: Fordham University Press

[14] Khachian, L.G. (1979). "A Polynomial Algorithm in Linear Programming", Doklady Ahademiia Nauk USSR 244 (5), 1093-96 (translated in Soviet Mathematics Doklady 20, 191-194)

[15] Klee, V. and Minty, G. (1971). "How Good is the Simplex Algorithm?" in Inequalities III. O Shisa, Ed., Academic Press: New York

[16] Kuhn, H. and Quandt, R.E. (1963). "An Experimental Study of the Simplex Method", Proceeding of Symposia in Applied Math., Vol. XV, Amer. Math. Soc., 107-124

[17] Singleton, R.R. (1940). "A Method for Minimizing the Sum of Absolute Values of Deviations", Ann. Math. Stat. 11, 301-310

[18] van Dam, W.B. and Telgen, J. "Randomly Generated Polytopes for Testing Mathematical Programming Algorithms", Report 7929, Econometric Institute, Erasmus Univ.

[19] Wagner, Harvey, M. (1959). "Linear Programming Techniques for Regression Analysis", J. Amer. Stat. Assoc. 54, 206-212

TABLE 1

Comparison of LAD and the one-phase Simplex method in solving the LP problem (15), iteration counts averaged over 10 independent instances.

m		n = 5	n = 10	n = 20	n = 50
5	SIMP	4.6	6.9	8.4	9.9
	LAD	3.1	4.0	5.1	6.7
10	SIMP		10.8	16.3	22.6
	LAD		7.1	10.0	12.7
20	SIMP			24.6	42.6
	LAD			14.8	25.8

TABLE 2

Comparison of LAD and the two-phase Simplex method in solving the LP problem (15), iteration counts averaged over 10 independent instances.

m \ n	5	10	15	20	25	30	40	50	
5	7.7 3.2	11.1 3.9	11.8 4.8	13.7 5.3	13.5 6.4	13.5 6.4	15.4 7.2	14.1 6.8	SIMP LAD
10		21.1 6.4	26.1 8.0	29.6 10.2	29.5 11.0	31.3 11.5	35.7 13.7	39.5 13.3	SIMP LAD
15			36.1 10.1	47.9 12.6	46.8 15.0	45.8 15.6	58.9 19.0	60.0 20.4	SIMP LAD
20				56.8 13.9	68.5 16.1	69.8 18.5	76.3 21.3	81.5 24.8	SIMP LAD