MATHEMATICS OF COMPUTATION, VOLUME 29, NUMBER 131
JULY 1975, PAGES 844–850

An Efficient Method for the Discrete Linear L_1 Approximation Problem

By Nabih N. Abdelmalek

Abstract. An improved dual simplex algorithm for the solution of the discrete linear L_1 approximation problem is described. In this algorithm certain intermediate iterations are skipped. This method is comparable with an improved simplex method due to Barrodale and Roberts, in both speed and number of iterations. It also has the advantage that in case of ill-conditioned problems, the basis matrix can lend itself to triangular factorization and can thus ensure a stable solution. Numerical results are given.

1. Introduction. Consider the overdetermined system of linear equations

$$Ca = f, \tag{1}$$

where C is a given real $n \times m$ constant matrix of rank $k \leqslant m < n$, and f is a given real n-vector. The L_1 solution of (1) is to determine the m-vector a which minimizes the L_1 norm

$$R(a) = \sum_{i=1}^{n} |r_i|, \tag{2}$$

where the residuals

$$r_i = c_{i1}a_1 + c_{i2}a_2 + \cdots + c_{im}a_m - f_i, \qquad i = 1, \ldots, n. \tag{3}$$

Wagner [10] reduced this problem to a linear programming problem in either the primal or the dual forms. The dual form is

$$\text{Maximize } z = \sum_{i=1}^{n} f_i(b_i - 1), \tag{4a}$$

subject to the constraints

$$C^T b = \sum_{i=1}^{n} C_i^T, \tag{4b}$$

$$0 \leqslant b_i \leqslant 2, \qquad i = 1, \ldots, n. \tag{4c}$$

C^T is the transpose of matrix C, C_i^T is the ith column of C^T and the bounded vector $b = (b_i)$.

In [1], a dual simplex algorithm for solving problem (4) is given, where no artificial variables are used. It was also shown that the algorithm in [1] is completely

Received January 2, 1974; revised October 2, 1974.

AMS (MOS) subject classifications (1970). Primary 65D15, 65B99, 65F20; Secondary 90C05, 90C50.

Key words and phrases. Discrete linear L_1 approximation, overdetermined system of linear equations, linear programming, dual simplex algorithm.

equivalent to a modified version of a method due to Usow [9] for solving the discrete linear L_1 approximation problem. Except that one iteration in the latter is equivalent to one or more iterations in the former.

Meanwhile, Barrodale and Roberts [5], [6], [7] described an efficient algorithm for the present problem. This algorithm improves an earlier simplex method [4], for the primal of problem (4). The main improvement is that their algorithm is able to skip certain intermediate iterations and this makes their method a fast one.

In the present work, it is shown that such improvement could be implemented in a natural manner, in our method [1]. Thus the algorithm in [1] is here developed such that certain intermediate iterations are skipped. Hence according to Lemma 7 in [1], this makes each iteration completely equivalent to a corresponding iteration in Usow's algorithm.

2. The Description of The New Method. We shall use most of the notations given in [1]. Let the basis indicator set for b be the index set $I(b) \subset \{1, 2, \ldots, n\}$ with the property that the variables $\{b_i | i \in I(b)\}$ are basic variables. Let also the index sets $L(b)$ and $U(b)$ be indicators for the nonbasic variables b_i, which are respectively at their lower and at their upper bounds. B denotes the basis matrix and the basic variables are denoted by $b_B = \{b_{B_i}\}$, $i = 1, \ldots, m$.

As usual, the simplex tableau is formed by calculating the nonbasic vectors y_j and the parameters $\{z_j - f_j\}$, where

$$y_j = B^{-1} C_j^T \tag{5}$$

and

$$z_j = f_B^T y_j. \tag{6}$$

Hence since some of the nonbasic variables may be at their upper bound (= 2), from (4b),

$$b_B = B^{-1}\left[\sum_{j=1}^{n} C_j^T - 2 \sum_{i \in U(b)} C_i^T\right] = b_{B_0} - 2 \sum_{i \in U(b)} y_i. \tag{7}$$

From now on, b_{B_i} will denote the basic variable under consideration. Let C_j^T be associated with b_{B_i} and let C_r^T be the nonbasic column which replaces C_j^T in the basis. In view of (7), steps 3.1–3.4 of [1], are analyzed as follows. When a nonbasic column C_r^T at its upper bound enters the basis, as in cases 3.2 and 3.4 in [1], it is no more at its upper bound and according to (7), we add $2y_r$ to b_B, or in effect, we add 2 to b_{B_i}. Also when a basic column C_j^T goes from the basis to its upper bound, as in cases 3.3 and 3.4, we subtract $2y_j$ from b_B.

We also notice, that the process of adding the $2y_r$ and/or subtracting the $2y_j$ is done after the simplex tableau has been changed by applying a Gauss-Jordan elimination step.

The modification to the algorithm in [1], consists simply of making the following two changes. (1) Reverse the order of the mentioned two processes. The Gauss-Jordan step may follow the process of adding the $2y_r$ and or subtracting the $2y_j$. (2) Further-more, the changing of the tableau, may be postponed, until further $2y_r$ have been added to and/or $2y_j$ have been subtracted from b_B. That is until some nonbasic columns, each enters and then leaves the basis, as the corresponding b_{B_i} does not yet satisfy (4c). This process continues until the last nonbasic column for which b_{B_i} satisfies (4c), is found. This ensures that the maximum decrease in z in replacing the basic column C_j^T, has been obtained [9], [1].

Let the last nonbasic column which enters the basis for which b_{B_i} satisfies (4c) be the kth one. Let us also assume that we start the procedure of the previous paragraph from a basic solution given by a simplex tableau, tableau 1 say. Then this method suggests that we skip calculating the $(k-1)$ intermediate tableaux, tableaux $t = 2, 3, \ldots, k$, which correspond to the $(k-1)$ nonbasic columns which entered and then left the basis. We calculate only tableau $t = k + 1$.

We show here that all necessary data needed for pursuing this procedure is contained in the ith row of Tableau 1 and the marginal costs of this tableau. In particular we need to know the values of the parameters $\{b_{B_i}\}$ and the pivot elements $\{y_{ir}\}$, for the $(k-1)$ intermediate tableaux.

We notice that each of the intermediate tableaux is obtained by pivoting over an element y_{ir} in row i of the previous tableau. For simplicity, assume that tableaux $t = 2, 3, \ldots, k+1$, are obtained by pivoting over the element in column $(t-1)$ of row i in tableau $(t-1)$. Obviously row i in tableau $t = 2, \ldots, k+1$, is simply obtained by dividing over the pivot element in row i of the previous tableau. By working this out, we see that the ith component of the basic variables and the pivot element in tableaux $t = 2, \ldots, k$, are simply $(b_{B_i}/y_{i,t-1})$ and $(y_{it}/y_{i,t-1})$ respectively. The following fact is also known. The $(k+1)$th tableau is itself the tableau we obtain had we changed tableau 1 once by pivoting over y_{ik}. Let

$$\text{(8a)} \qquad \bar{b}'_{B_i} = b_{B_i} \text{ and } \bar{y}_{it} = y_{it}, \qquad t = 1,$$

$$\text{(8b)} \qquad \bar{b}'_{B_i} = b'_{B_i}/y_{i,t-1} \quad \text{and} \quad \bar{y}_{it} = y_{it}/y_{i,t-1}, \qquad t = 2, 3, \ldots, k.$$

Here $\bar{b}'_{B_i}$ and $\bar{y}_{it}$ represent the ith component of the basic variables and the pivot in tableaux $t = 1, 2, \ldots, k$. The vector b'_B is b_B added to it some $2y_r$ and or subtracted from it some $2y_j$. The element b'_{B_i} is its ith component.

We also need to know the sequence of the k nonbasic columns which enter the basis, and may then leave the basis, in an iteration. This may be determined from the following theorem, whose proof may be established by working out two consecutive tableaux in the method of [1].

THEOREM 1. *Let us start a certain iteration from a basic solution given by tableau* 1 *say. Let* $\{z_s - f_s\}$, $s = 1, 2, \ldots, n$, *be the marginal costs in tableau* 1. *Then the sequence of the nonbasic columns assumed to enter the basis and may then leave the basis is determined as follows.*

If $b_{B_i} > 2$, *the sequence is given by the columns corresponding to the parameters* $\tau_r = (z_r - f_r)/y_{ir} > 0$, $r \notin I(b)$, *starting from the smallest one.*

If $b_{B_i} < 0$, *then the sequence is given by the columns corresponding to the parameters* $\theta_r = (z_r - f_r)/y_{ir} < 0$, $r \notin I(b)$, *starting from the algebraically biggest one.*

The new algorithm modifies step 3 of [1], as follows.

Step 3′: Scan b_{B_l} for $l = 1, 2, \ldots, m$, and consider

$$b_{B_i} = \min\{b_1, b_2\}, \tag{9}$$

where $b_1 = \min_l\{b_{B_l}, b_{B_l} < 0\}$ and $b_2 = \min_l\{2 - b_{B_l}, b_{B_l} > 2\}$. This corresponds to choosing $(z_i - f_i) = \min_l\{z_l - f_l\}$ in the simplex method. Let C_r^T replace C_j^T according to one of the steps 3.1′–3.4′ below and calculate the new $\bar{b}'_{B_i}$. If this new $\bar{b}'_{B_i}$ still violates (4c), this last vector C_r^T leaves the basis and another nonbasic vector enters the basis. This is repeated, a finite number of times until $\bar{b}'_{B_i}$ satisfies (4c).

Case 1′. $\bar{b}'_{B_i} < 0$ and C_r^T is determined from Theorem 1.

3.1′. If $\bar{y}_{ir} < 0$, do not change b'_B, go to 3.5.

3.2′. If $\bar{y}_{ir} > 0$, add $2y_r$ to b'_B and go to 3.5. Remove the mark from column C_r^T to indicate that b_r is no more at its upper bound.

Case 2′. $\bar{b}'_{B_i} > 2$ and C_r^T is determined from Theorem 1.

3.3′. If $\bar{y}_{ir} > 0$, subtract $2y_j$ from b'_B and place a mark over column C_j^T to indicate that it is now at its upper bound. Go to 3.5.

3.4′. If $\bar{y}_{ir} < 0$, add $2y_r$ to and subtract $2y_j$ from b'_B. Remove the mark from column C_r^T and place a mark on column C_j^T. Go to step 3.5.

3.5. Calculate $\bar{b}'_{B_i}$ from (8). If the answer violates (4c), replace C_j^T by C_r^T and go to either Case 1′ or Case 2′ depending on whether the new $\bar{b}'_{B_i}$ is < 0 or > 2, respectively. If the answer satisfies (4c), change the tableau in the usual manner. Go to step 2 in [1].

This constitutes one iteration, which again consists of k steps. The number of skipped iterations would be $(k - 1)$.

3. The Occurrence of Degeneracy. As usual, no provisions are made to resolve degeneracy. However, one difficulty arises from the occurrence of initial degeneracy. That is if there exist one or more nonbasic column r for which $(z_r - f_r)/y_{ir} = 0$. It should be decided whether $\tau_r = 0$ or $\theta_r = 0$. This is resolved as follows. If $b_r = 0$, we suggest to replace $(z_r - f_r)$ by a very small number δ say, and if $b_r = 2$, we replace $(z_r - f_r)$ by $-\delta$. We then recalculate the ratio $(z_r - f_r)/y_{ir}$, and it will definitely be either $>$ or < 0, and the mentioned difficulty is resolved. The quantity δ is of the order of the precision of the computer.

4. Organizing the Computation. The computation may be divided into two parts. In part 1, a numerically stable initial basic solution is obtained. Also in case of rank deficiency, linearly dependent rows of C^T are detected and discarded. Part 2, constitutes the main body of the algorithm. This is explained by the following example. However, in order to illustrate the details of the present method, in this example, part 1 is obtained differently. This point is mentioned later.

Solve the following system of equations in the L_1 norm.

$$
\begin{aligned}
-2a_1 - 2a_3 &= 6,\\
8a_1 + 9a_2 + 17a_3 &= 6,\\
36a_1 + 18a_2 + 54a_3 &= -48,\\
-8a_1 - 8a_3 &= 24,\\
21a_1 + 18a_2 + 39a_3 &= 3,\\
12a_1 - 9a_2 + 3a_3 &= -6,\\
-32a_1 - 13.5a_2 - 45.5a_3 &= -9.
\end{aligned} \tag{10}
$$

In this example, C is a 7×3 matrix of rank 2. The third column equals the sum of the other two. Shown are the initial data for programming problem (4) and the simplex tableaux 2 and 3 for the solution.

Tableau 1 is obtained by pivoting over the first nonzero element in row 1 of C^T and applying a Gauss-Jordan step. Tableau 2 is obtained likewise by pivoting over the first nonzero element in row 2 of tableau 1. It is seen that the third row in tableau 2 consists of zero elements and thus it is discarded from the computation. This ends part 1 with columns 1 and 2 of C^T forming the initial basis and column 3 of C discarded.

Tableau 3 is itself tableau 2 added to it the marginal costs $\{z_j - f_j\}$, $j = 1, \ldots, 7$, calculated from (6) and (7). Column b_{B_0} is modified by subtracting $2(y_5 + y_6)$, of the columns having negative marginal costs. A mark x is placed over each of these two columns. The objective function z which equals $R(a)$ of (2) is calculated from (31) of [1].

Initial Data

f_j	6	6	−48	24	3	−6	−9
$\sum_{j=1}^{7} C_j^T$	C_1^T	C_2^T	C_3^T	C_4^T	C_5^T	C_6^T	C_7^T
35	−2	8	36	−8	21	12	−32
22.5	0	9	18	0	18	−9	−13.5
57.5	−2	17	54	−8	39	3	−45.5

Tableau 2 (Part 1)

b_{B_0}							
−7.5	1	0	−10	4	−2.5	−10	10
2.5	0	1	2	0	2	−1	−1.5
0	0	0	0	0	0	0	0

Tableau 3 (Part 2)

b_B	b'_B					x	x	
$-7.5 + 5 + 20 = 17.5$	−17.5	1	0	−10	4	−2.5	−10	10
$2.5 - 4 + 2 = .5$	2.5	0	1	2	0	2	−1	−1.5
$z = 126$	$-6(z_j - f_j)$	0	0	0	0	−6	−60	60
	τ_r				0	2.4	6	6

By applying (9), to b_B in tableau 3, C_1^T should leave the basis and we have Case 2′ of the algorithm. We have an initial degeneracy, as $(z_r - f_r)/y_{ir} = 0$ for $r = 3$ and $r = 4$. Thus from Section 3 above, since each of $b_3 = 0$ and $b_4 = 0$, we replace each of $(z_3 - f_3)$ and $(z_4 - f_4)$ by δ, and recalculate the two ratios. The new $(z_3 - f_3)/y_{13} < 0$ and thus C_3^T cannot enter the basis, but the new $(z_4 - f_4)/y_{14} > 0$ and thus C_4^T

replaces C_1^T in the basis. Also shown in tableau 3, are the initial values of τ_r for this iteration. Hence according to Theorem 1, the sequence of the columns which enter the basis and may then leave the basis is C_4^T, C_5^T, C_6^T, C_7^T.

From (8a), the pivot $\bar{y}_{14} = 4 > 0$, and according to 3.3′, $2y_1$ is subtracted from b_B, and a mark is placed above C_1^T. The new $\bar{b}'_{B_1} = (17.5 - 2)/4 = 15.5/4 > 2$, and we still have Case 2′ of the algorithm. Now C_4^T leaves the basis and C_5^T enters the basis. Again from (8b), the pivot $\bar{y}_{15} = -2.5/4 < 0$. Hence from 3.4′, $2y_4$ is subtracted from and $2y_5$ is added to the previously calculated b'_B. Also a mark is placed over C_4^T and the mark over C_5^T is removed. The new $\bar{b}'_{B_1} = (15.5 - 8 - 5)/- 2.5 = 2.5/- 2.5 <$ 0, and we have now Case 1′ of the algorithm.

Now C_5^T leaves the basis and either C_6^T or C_7^T enters the basis, as they both have the same initial τ_r. Let C_6^T enter the basis. The pivot $\bar{y}_{16} = -10/- 2.5 > 0$, and from 3.2′ we add $2y_6$ to the previously obtained b'_B and we remove the mark of C_6^T. The new $\bar{b}'_{B_1} = (2.5 - 20)/- 10 = 1.75$, and we thus change the tableau. This ends this iteration which consists of 3 steps.

The same procedure is followed in the next iteration which consists of one step in which C_5^T replaces C_2^T in the basis. The solution in tableau 5 is optimal and feasible. Hence, this example required 4 iterations, 2 in part 1 and 2 in part 2. The final answer of this example is $R(a^*) = 90$, $a_1^* = -0.2$, $a_2^* = 0.4$ and $a_3^* = 0$. The $\{a_i^*\}$ are obtained by solving the 5th and 6th equations in (10) and taking $a_3 = 0$.

It is mentioned earlier that in part 1 of the present algorithm, a numerically stable initial basic solution is obtained. This is done as follows. In part 1, tableau t is obtained by pivoting over the largest element in absolute value in row t of tableau $(t - 1)$. The final result of the problem is obtained much faster.

Again consider the example given by (10). Tableau 1 is obtained by pivoting over the largest element in absolute value in row 1 of C^T. Tableau 2 is obtained likewise by pivoting over the largest element in absolute value in row 2 in tableau 1. Here columns 3 and 6 form the initial basis.

The final tableau is obtained in one more iteration which consists of 1 step in which C_5^T replaces C_3^T in the basis. This is instead of the two iterations which consisted of 4 steps, in the previous solution.

5. Numerical Results and Comments. Over 50 test problems were solved by both the present method [2] and that of Barrodale and Roberts [7]; each is coded in Fortran IV. The results show that the two methods are comparable in both the speed and the number of iterations. The execution time required by both methods differ by less than 25% either way (referee's confirmation).

The examination of the two methods shows that, apart from part 1 of the present method, the computational procedures of the two methods are very similar.

However, in obtaining part 1 of the present method, in tableau t, $t = 1, \ldots, k$, where k is the rank of C, we attempt to obtain the smallest possible, in absolute value, of the initial basic variables b_{B_t}. That is, we attempt to bring these variables near to their values for the optimum solution, $0 \leqslant b_{B_i} \leqslant 2$. Hence we expect to start part 2 with the best possible initial basic solution. While part 2 constitutes the active part of the algorithm, the optimal solution is obtained by the accelerating process of skipping certain intermediate iterations. On the other hand, in the method [5]–[7], the active role of the algorithm starts from tableau 1. This might explain why the two methods are comparable.

Finally, we remark that, unlike in the method of [5]–[7], the basis matrix B has the order of the rank of matrix C. Therefore, if necessary, we can apply to it triangular factorization techniques [8], [3]. This ensures stable solutions for ill-conditioned problems.

6. Acknowledgment. I would like to express my gratitude to Ian Fraser for providing most of the data used in testing this algorithm and for several interesting discussions.

Computation Centre
National Research Council of Canada
Ottawa, Ontario, Canada

1. N. N. ABDELMALEK, "On the discrete linear L_1 approximation and L_1 solutions of overdetermined linear equations," *J. Approximation Theory*, v. 11, 1974, pp. 38–53.

2. N. N. ABDELMALEK, "L_1 solution of overdetermined system of linear equations by a dual simplex method," *Comm. ACM* (Submitted.)

3. N. N. ABDELMALEK, "L_1 solution of overdetermined system of linear equations by a dual simplex method and *LU* decomposition," *Comm. ACM* (Submitted.)

4. I. BARRODALE & A. YOUNG, "Algorithms for best L_1 and L_∞ linear approximations on a discrete set," *Numer. Math.*, v. 8, 1966, pp. 295–306. MR 33 #5096.

5. I. BARRODALE & F. D. K. ROBERTS, *An Improved Algorithm For Discrete l_1 Approximation*, The University of Wisconsin, MRC Tech. Report no. 1172, 1972.

6. I. BARRODALE & F. D. K. ROBERTS, "An improved algorithm for discrete l_1 approximation," *SIAM J. Numer. Anal.*, v. 10, 1973, pp. 839–848.

7. I. BARRODALE & F. D. K. ROBERTS, "Solution of an overdetermined system of equations in the l_1 norm," *Comm. ACM*, v. 17, 1974, pp. 319–920.

8. R. H. BARTELS & G. H. GOLUB, "The simplex method of linear programming using *LU* decomposition," *Comm. ACM*, v. 12, 1969, pp. 266–268. MR 39 #2302.

9. K. H. USOW, "On L_1 approximation II: Computation for discrete functions and discretization effects," *SIAM J. Numer. Anal.*, v. 4, 1967, pp. 233–244. MR 36 #588.

10. H. M. WAGNER, "Linear programming techniques for regression analysis," *J. Amer. Statist. Assoc.*, v. 54, 1959, pp. 206–212. MR 24 #A612.