# 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

$$
\begin{equation*}
C a=f \tag{1}
\end{equation*}
$$

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

$$
\begin{equation*}
R(a)=\sum_{i=1}^{n}\left|r_{i}\right| \tag{2}
\end{equation*}
$$

where the residuals

$$
\begin{equation*}
r_{i}=c_{i 1} a_{1}+c_{i 2} a_{2}+\cdots+c_{i m} a_{m}-f_{i}, \quad i=1, \ldots, n \tag{3}
\end{equation*}
$$

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

$$
\begin{equation*}
\text { Maximize } z=\sum_{i=1}^{n} f_{i}\left(b_{i}-1\right) \tag{4a}
\end{equation*}
$$

subject to the constraints

$$
\begin{equation*}
C^{T} b=\sum_{i=1}^{n} C_{i}^{T} \tag{4b}
\end{equation*}
$$

$$
\begin{equation*}
0 \leqslant b_{i} \leqslant 2, \quad i=1, \ldots, n \tag{4c}
\end{equation*}
$$

$C^{T}$ is the transpose of matrix $C, C_{i}^{T}$ is the $i$ th column of $C^{T}$ and the bounded vector $b=\left(b_{i}\right)$.

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

[^0]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 $\left\{b_{i} \mid i \in I(b)\right\}$ 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 $b_{B}=\left\{b_{B_{i}}\right\}, i=1, \ldots, m$.

As usual, the simplex tableau is formed by calculating the nonbasic vectors $y_{j}$ and the parameters $\left\{z_{j}-f_{j}\right\}$, where

$$
\begin{equation*}
y_{j}=B^{-1} C_{j}^{T} \tag{5}
\end{equation*}
$$

and

$$
\begin{equation*}
z_{j}=f_{B}^{T} y_{\dot{r}} \tag{6}
\end{equation*}
$$

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

$$
\begin{equation*}
b_{B}=B^{-1}\left[\sum_{j=1}^{n} C_{j}^{T}-2 \sum_{i \in U(b)} C_{i}^{T}\right]=b_{B_{0}}-2 \sum_{i \in U(b)} y_{i} . \tag{7}
\end{equation*}
$$

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 $2 y_{r}$ 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 $2 y_{j}$ from $b_{B}$.

We also notice, that the process of adding the $2 y_{r}$ and/or subtracting the $2 y_{j}$ 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 $2 y_{r}$ and or subtracting the $2 y_{j}$. (2) Further-
more, the changing of the tableau, may be postponed, until further $2 y_{r}$ have been added to and/or $2 y_{j}$ 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 $k$ th 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 $i$ th row of Tableau 1 and the marginal costs of this tableau. In particular we need to know the values of the parameters $\left\{b_{B_{i}}\right\}$ and the pivot elements $\left\{y_{i r}\right\}$, for the ( $k-1$ ) intermediate tableaux.

We notice that each of the intermediate tableaux is obtained by pivoting over an element $y_{i r}$ 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 $i$ th component of the basic variables and the pivot element in tableaux $t=2, \ldots, k$, are simply $\left(b_{B_{i}} / y_{i, t-1}\right)$ and $\left(y_{i t} / y_{i, t-1}\right)$ 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 $y_{i k}$. Let

$$
\begin{gather*}
\bar{b}_{B_{i}}^{\prime}=b_{B_{i}} \text { and } \bar{y}_{i t}=y_{i t}, \quad t=1,  \tag{8a}\\
\bar{b}_{B_{i}}^{\prime}=b_{B_{i}}^{\prime} / y_{i, t-1} \quad \text { and } \quad \bar{y}_{i t}=y_{i t} / y_{i, t-1}, \quad t=2,3, \ldots, k . \tag{8b}
\end{gather*}
$$

Here $\bar{b}_{B_{i}}^{\prime}$ and $\bar{y}_{i t}$ represent the $i$ th component of the basic variables and the pivot in tableaux $t=1,2, \ldots, k$. The vector $b_{B}^{\prime}$ is $b_{B}$ added to it some $2 y_{r}$ and or subtracted from it some $2 y_{j}$. The element $b_{B_{i}}^{\prime}$ is its $i$ th 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 $\left\{z_{s}-f_{s}\right\}, s=1,2, \ldots, n$, 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 $\tau_{r}=\left(z_{r}-f_{r}\right) / y_{i r}>0, r \notin I(b)$, starting from the smallest one.

If $b_{B_{i}}<0$, then the sequence is given by the columns corresponding to the parameters $\theta_{r}=\left(z_{r}-f_{r}\right) / y_{i r}<0, r \notin I(b)$, 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

$$
\begin{equation*}
b_{B_{i}}=\min \left\{b_{1}, b_{2}\right\}, \tag{9}
\end{equation*}
$$

where $b_{1}=\min _{l}\left\{b_{B_{l}}, b_{B_{l}}<0\right\}$ and $b_{2}=\min _{l}\left\{2-b_{B_{l}}, b_{B_{l}}>2\right\}$. This corresponds to choosing $\left(z_{i}-f_{i}\right)=\min _{l}\left\{z_{l}-f_{l}\right\}$ in the simplex method. Let $C_{r}^{T}$ replace $C_{j}^{T}$ according to one of the steps $3.1^{\prime}-3.4^{\prime}$ below and calculate the new $\bar{b}_{B_{i}}^{\prime}$. If this new $\bar{b}_{B_{i}}^{\prime}$ 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 $\bar{b}_{B_{i}}^{\prime}$ satisfies (4c).

Case $1^{\prime} . \bar{b}_{B_{i}}^{\prime}<0$ and $C_{r}^{T}$ is determined from Theorem 1.
3.1'. If $\bar{y}_{i r}<0$, do not change $b_{B}^{\prime}$, go to 3.5 .
3.2'. If $\bar{y}_{i r}>0$, add $2 y_{r}$ to $b_{B}^{\prime}$ 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 $2^{\prime} . \bar{b}_{B_{i}}^{\prime}>2$ and $C_{r}^{T}$ is determined from Theorem 1.
3.3'. If $\bar{y}_{i r}>0$, subtract $2 y_{j}$ from $b_{B}^{\prime}$ 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 $\bar{y}_{i r}<0$, add $2 y_{r}$ to and subtract $2 y_{j}$ from $b_{B}^{\prime}$. 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 $\bar{b}_{B_{i}}^{\prime}$ from (8). If the answer violates (4c), replace $C_{j}^{T}$ by $C_{r}^{T}$ and go to either Case $1^{\prime}$ or Case $2^{\prime}$ depending on whether the new $\bar{b}_{B_{i}}^{\prime}$ 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 $\left(z_{r}-f_{r}\right) / y_{i r}=0$. 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 $\left(z_{r}-f_{r}\right)$ by a very small number $\delta$ say, and if $b_{r}=2$, we replace $\left(z_{r}-f_{r}\right)$ by $-\delta$. We then recalculate the ratio $\left(z_{r}-f_{r}\right) / y_{i r}$, and it will definitely be either $>$ or $<0$, and the mentioned difficulty is resolved. The quantity $\delta$ 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.

$$
\begin{align*}
-2 a_{1}-2 a_{3} & =6, \\
8 a_{1}+9 a_{2}+17 a_{3} & =6, \\
36 a_{1}+18 a_{2}+54 a_{3} & =-48, \\
-8 a_{1}-8 a_{3} & =24,  \tag{10}\\
21 a_{1}+18 a_{2}+39 a_{3} & =3, \\
12 a_{1}-9 a_{2}+3 a_{3} & =-6, \\
-32 a_{1}-13.5 a_{2}-45.5 a_{3} & =-9 .
\end{align*}
$$

In this example, $C$ is a $7 \times 3$ 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 $\left\{z_{j}-f_{j}\right\}, j=1, \ldots$, 7, calculated from (6) and (7). Column $b_{B_{0}}$ is modified by subtracting $2\left(y_{5}+y_{6}\right)$, 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




By applying (9), to $b_{B}$ in tableau 3, $C_{1}^{T}$ should leave the basis and we have Case $2^{\prime}$ of the algorithm. We have an initial degeneracy, as $\left(z_{r}-f_{r}\right) / y_{i r}=0$ 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 $\left(z_{3}-f_{3}\right)$ and $\left(z_{4}-f_{4}\right)$ by $\delta$, and recalculate the two ratios. The new $\left(z_{3}-f_{3}\right) / y_{13}$ $<0$ and thus $C_{3}^{T}$ cannot enter the basis, but the new $\left(z_{4}-f_{4}\right) / y_{14}>0$ and thus $C_{4}^{T}$
replaces $C_{1}^{T}$ in the basis. Also shown in tableau 3, are the initial values of $\tau_{r}$ 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 $3.3^{\prime}, 2 y_{1}$ is subtracted from $b_{B}$, and a mark is placed above $C_{1}^{T}$. The new $\bar{b}_{B_{1}}^{\prime}=(17.5-2) / 4=15.5 / 4>2$, and we still have Case $2^{\prime}$ 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 $3.4^{\prime}, 2 y_{4}$ is subtracted from and $2 y_{5}$ is added to the previously calculated $b_{B}^{\prime}$. Also a mark is placed over $C_{4}^{T}$ and the mark over $C_{5}^{T}$ is removed. The new $\bar{b}_{B_{1}}^{\prime}=(15.5-8-5) /-2.5=2.5 /-2.5<$ 0 , and we have now Case $1^{\prime}$ 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 $\tau_{r}$. Let $C_{6}^{T}$ enter the basis. The pivot $\bar{y}_{16}=-10 /-2.5>0$, and from $3.2^{\prime}$ we add $2 y_{6}$ to the previously obtained $b_{B}^{\prime}$ and we remove the mark of $C_{6}^{T}$. The new $\bar{b}_{B_{1}}^{\prime}=(2.5-20) /-10=1.75$, 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 $R\left(a^{*}\right)=90, a_{1}^{*}=-0.2, a_{2}^{*}=0.4$ and $a_{3}^{*}=0$. The $\left\{a_{i}^{*}\right\}$ are obtained by solving the 5 th and 6 th 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 $L U$ decomposition," Comm. ACM (Submitted.)
4. I. BARRODALE \& A. YOUNG, "Algorithms for best $L_{1}$ and $L_{\infty}$ 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 $\boldsymbol{l}_{1}$ Approximation, The University of Wisconsin, MRC Tech. Report no. 1172, 1972.
6. I. BARRODALE \& F. D. K. ROBERTS, "An improved algorithm for discrete $\boldsymbol{l}_{1}$ 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 $L U$ 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.

[^0]:    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.

    Copyright © 1975. American Mathematical Society

