CENTER FOR CYBERNETIC STUDIES
The University of Texas
Austin, Texas 78712
AN ALGORITHM FOR A LEAST ABSOLUTE VALUE REGRESSION PROBLEM WITH BOUNDS ON THE PARAMETERS

by

R. Armstrong
M. Kung*

June 1980

*York University, Downsview, Ontario, Canada

This research was partly supported by ONR Contracts N00014-75-C-0569 and N00014-80-C-0242 with the Center for Cybernetic Studies, The University of Texas at Austin. Reproduction in whole or in part is permitted for any purpose of the United States Government.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
Business-Economics Building, 203E
The University of Texas at Austin
Austin, Texas 78712
(512) 471-1821
An Algorithm for a Least Absolute Value Regression Problem with Bounds on the Parameters

ABSTRACT

This paper presents a special purpose linear programming algorithm to solve a least absolute value regression problem with upper and lower bounds on the parameters. The algorithm exploits the problem's special structure by maintaining a compact representation of the basis inverse and by allowing for the capability to combine several simplex iterations into one. Computational results with a computer code implementation of the algorithm is given.
1. **Introduction**

Least absolute value (LAV) regression has become popular recently as a robust estimation technique [11]. This has come about as a result of an increasing awareness of the limitations of least squares analysis and the development of efficient algorithms for obtaining LAV estimates [2,5].

This paper considers a restricted least absolute value regression problem of the following form.

\[
\text{Minimize} \quad \sum_{i=1}^{n} \left| y_i - \sum_{j=1}^{m} x_{ij} \beta_j \right|
\]

subject to \( L_j \leq \beta_j \leq U_j \), \( j = 1, 2, \ldots, m \).

For purposes of exposition, it is assumed that all \( \beta_j \)'s have upper and lower bounds. An absence of an upper (lower) bound on a parameter can be handled within the algorithm to be developed here by either assigning an arbitrarily large positive (negative) value for the bound, or by assuming the existence of this pseudo-bound and handling it logically (i.e., no value is actually assigned). The former method is used in our computer code implementation.

Special purpose algorithms for the least squares (LSQ) equivalent of (1) are given by Armstrong and Frome [1] and Waterman [18]. These algorithms were able to deviate from the standard linear programming (LP) approach to solving restricted LSQ problems, commonly called quadratic programming problems, by utilizing the special structure of the constraints. Because
the objective function of (1) is convex, a modification of the algorithms
found in [1] and [18] can also be used to solve the restricted LAV problem.
The major change is that an unrestricted LAV problem must be solved at any stage
rather than an unrestricted LSQ problem. However, inasmuch as the most efficient
solution method for the unrestricted LAV problem utilizes LP and the constraints
of (1) can be handled within the LP framework, it would seem appropriate to
specialize the existing LAV-LP algorithms to solve (1). This is done in the
next section.

2. Algorithm

Charnes, Cooper and Ferguson [8] appear to be the first to have
demonstrated that linear LAV problems can be rewritten as LP problems.
Employing their result here, problem (1) is equivalent to:

\[ \text{(2) Minimize } \sum_{i=1}^{n} (\delta_i^+ + \delta_i^-), \]

subject to

\[ \sum_{j=1}^{m} x_{ij} \beta_j + \delta_i^+ - \delta_i^- = y_i, \quad i = 1, 2, \ldots, n \]

\[ L_j \leq \beta_j \leq U_j, \quad j = 1, 2, \ldots, m \]

\[ \delta_i^+ \geq 0 \quad \text{and} \quad \delta_i^- \geq 0, \quad i = 1, 2, \ldots, n. \]

Davies [10], Barrodale and Roberts [5], and Spyropoulos, Kiountouzis
and Young [17] give closely related special purpose primal simplex algorithms
to solve (2) when bounds on the parameters are not present. These algorithms
currently appear to be the most efficient method to solve the unrestricted LAV problem. Armstrong and Hultz [3] and Barrodale and Roberts [6] develop primal LP algorithms for a LAV problem with arbitrary linear constraints on the parameters. Both algorithms can be applied to problem (2). Armstrong and Hultz [3] solve the restricted LAV problem using techniques from interval linear programming [9] while Barrodale and Roberts [6] utilize a modification of their unconstrained algorithm which was presented in [4]. The algorithm proposed here is a specialization of the method from [3]. It will be demonstrated that considerable computational simplifications are possible when the constraints are of the important special form present in (2).

To aid in the discussion to follow, the constraints of (2) are rewritten in matrix notation as:

\begin{align}
(3) & \quad x_\beta + Ix^+ - Ix^- = \gamma, \\
(4) & \quad L \leq x_\beta \leq U, \\
(5) & \quad \delta^+ \geq 0 \text{ and } \delta^- \geq 0.
\end{align}

The dual problem of (2) is:

\begin{align}
(6) & \quad \text{Maximize} \quad \sum_{i=1}^{n} \pi_i y_i + \sum_{j=1}^{m} w_j u_j + \sum_{j=1}^{m} v_j l_j \\
& \quad \text{subject to} \quad \sum_{i=1}^{n} \pi_i x_{ij} + w_j + v_j = 0, \quad j = 1, 2, \ldots, m \\
& \quad \quad \quad \quad \quad \pi_i \quad \leq 1, \quad i = 1, 2, \ldots, n \\
& \quad \quad \quad \quad \quad \pi_i \quad \geq -1, \quad i = 1, 2, \ldots, n \\
& \quad \quad \quad \quad \quad w_j \quad \leq 0, \quad j = 1, 2, \ldots, m \\
& \quad \quad \quad \quad \quad v_j \quad \geq 0, \quad j = 1, 2, \ldots, m
\end{align}

The dual problem (6) will be used later with the explanation of the algorithm.
From the results presented by Armstrong and Hultz [3], all the information required to execute a primal LP algorithm can efficiently be obtained from a list of indicators and the inverse of an m by m basic matrix. This matrix, $X_B$, can be represented (after row and column interchanges), relative to problem (2), as follows.

$$X_B = \begin{pmatrix} X_F & X_R \\ 0 & I \end{pmatrix}$$

where $X_F$ is an r by r full rank matrix, $X_R$ is an r by m-r matrix, 0 is an m-r by r zero matrix and I is an m-r by m-r identity matrix. This structure assumes that r rows of X and m-r rows corresponding to the bound constraints are in the basis. The following discussion will assume that the rows and columns of the problem have been explicitly reordered to obtain this structure. The computer code utilizes two arrays to implicitly achieve this reordering.

It will be demonstrated that all the computations required in the algorithm can conveniently be performed by inverting only $X_F$, which shall be referred to as the working basis. The inverse of $X_B$ is given by:

$$X_B^{-1} = \begin{pmatrix} X_F^{-1} & -X_F^{-1}X_R \\ 0 & I \end{pmatrix}$$
The solution of (1), \( \mathbf{B} \), can be partitioned into \( \mathbf{B} = \begin{pmatrix} \mathbf{B}_F \\ \mathbf{B}_R \end{pmatrix} \) where \( \mathbf{B}_F \) is the current value of \( \beta_j \), \( j=1,2,\ldots,r \), and \( \mathbf{B}_R \) is the current value of \( \beta_j \), \( j=r+1,r+2,\ldots,m \). From equations (3) and (4), \( \mathbf{B} = \mathbf{X}^{-1} \begin{pmatrix} \mathbf{Y}_B \\ \mathbf{C} \end{pmatrix} \) where \( \mathbf{Y}_B \) are the values of the dependent variable corresponding to the \( r \) rows of \( \mathbf{X} \) in the basis and \( \mathbf{C} \) assumes the value of either \( U \) or \( L \) depending on whether \( \beta_j \), \( j=r+1,r+2,\ldots,m \) is at its upper or lower bound.

\[
\begin{pmatrix} \mathbf{B}_F \\ \mathbf{B}_R \end{pmatrix} = \begin{pmatrix} \mathbf{X}_F^{-1} & -\mathbf{X}_F^{-1}\mathbf{X}_R \\ 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{Y}_B \\ \mathbf{C} \end{pmatrix}
\]

\[
\mathbf{B}_R = \mathbf{C}
\]

\[
\mathbf{B}_F = \left( \mathbf{X}_F^{-1} - \mathbf{X}_F^{-1}\mathbf{X}_R \right) \begin{pmatrix} \mathbf{Y}_B \\ \mathbf{C} \end{pmatrix}
\]

(7) \( \mathbf{B}_F = \mathbf{X}_F^{-1} (\mathbf{Y}_B - \mathbf{X}_R \mathbf{C}_R) \)

By a list of bound indicators, the elements of \( \mathbf{B}_R \) will be assigned to their upper (or lower) bound value.

Define \( \mathbf{IB} \) to be the index set of the rows of \( \mathbf{X} \) in the basis, and \( \mathbf{NB} \) to be the index set of nonbasic rows of \( \mathbf{X} \). The basic and nonbasic rows of \( \mathbf{I} \) will be determined implicitly within the computer code by the value of \( r \) and the current column ordering. The discussion here assumes explicit reordering at each iteration.
The deviations for (2), or the reduced costs for the dual problem (6), are given by:

$$(8) \quad d_i = y_i - x_i^\pi, \quad i = 1, 2, \ldots, n.$$  

The dual variables corresponding to the nonbasic rows are assigned a value depending on the sign of the deviation, that is, $\pi_i = \text{sign}(d_i), \ i \in \text{NB}$. Degeneracy occurs when $d_i = 0, \ i \in \text{NB}$. In the computer code, the initial value assigned to $\pi_i$ when $d_i = 0, \ i \in \text{NB}$, is arbitrarily defined to be +1, and thereafter, the value is determined by the steps of the algorithm. A thorough treatment of the problem of degeneracy can be found elsewhere (see Charnes [7] for example) and this phenomenon will not be examined here.

Define the nonbasic dual variables $w_j$ and $v_j$ to be zero. Define $\tau_B$ to be the vector of the basic dual variables. $\tau_B$ can be partitioned into $\tau_{BF}$ and $\tau_{BR}$; that is,

$$\tau_B = (\tau_{BF}, \tau_{BR})$$

$$\tau_B = (\tau_{B_1}, \tau_{B_2}, \ldots, \tau_{B_r}, \tau_{B_{r+1}}, \tau_{B_{r+2}}, \ldots, \tau_{B_m})$$

where

$$\tau_{BF} = (\tau_{B_1}, \tau_{B_2}, \ldots, \tau_{B_r})$$

$$\tau_{BR} = (\tau_{B_{r+1}}, \tau_{B_{r+2}}, \ldots, \tau_{B_m}).$$

The vector $\tau_{BF}$ consists of the basic dual variables corresponding to the $r$ rows of $X$ in the basis $(X_B)$, and $\tau_{BR}$ consists of the basic dual variables corresponding to the basic bound constraints. The nonbasic dual variables $\pi_i, \ i \in \text{NB}$, are set to
+1 or -1. The nonbasic \( w_j \) and \( v_j \) are zero. Thus, from (6), the current values of the basic dual variables can be obtained from the dual constraints.

\[
\begin{align*}
\text{r} & \sum_{i=1}^{r} \bar{B}_iX_{ij} + \sum_{i \in NB} \bar{\pi}_iX_{ij} = 0, & j &= 1, 2, \ldots, r \\
\text{r} & \sum_{i=1}^{r} \bar{B}_iX_{ij} + \bar{\tau}_B + \sum_{i \in NB} \bar{\pi}_iX_{ij} = 0, & j &= r+1, r+2, \ldots, m \\
-\bar{\tau}_B^T & = -H_B^T
\end{align*}
\]

where \( H_B^j = \sum_{i \in NB} \bar{\pi}_iX_{ij}, j = 1, 2, \ldots, m \)

\[
\bar{\tau}_B = -H_B^X_B^{-1}
\]

\[
\begin{pmatrix}
\bar{\tau}_B^F \\
\bar{\tau}_B^R
\end{pmatrix} = -
\begin{pmatrix}
H_B^T \\
H_B^{TR}
\end{pmatrix}
\begin{bmatrix}
x_F^{-1} & -x_F^{-1}x_R \\
0 & 1
\end{bmatrix}
\]

(9) \( \bar{\tau}_B^F = -H_B^X_F^{-1} \)

(10) \( \bar{\tau}_B^R = -\bar{\tau}_B^X_R - H_B^R \)

Since this is a primal algorithm, the necessary condition for optimality is dual feasibility. The optimality condition for (2) is:

(11) \(-1 \leq \bar{\pi}_i \leq +1, \quad i = 1, 2, \ldots, n,\)

(12) \(\bar{\tau}_F \leq 0, \quad j = 1, 2, \ldots, m.\)

(13) \(\overline{v}_j \geq 0, \quad j = 1, 2, \ldots, m.\)

If the optimality condition is not satisfied, then there exists one or more basic rows where condition (11), (12), or (13) is violated. Define \( \bar{\tau}_{B_k} \) to be the
dual variable most violating a bound restriction. If \( k \leq r \), the \( k \)-th row of \( X_B \) will be chosen to leave the basis. On the other hand, if \( k > r \), the \( k \)-th row of \( I \) will be selected to leave the basis. Define \( \rho = \text{sign}(\tau_{B_k}) \). The value of \( \rho \) indicates if \( \tau_{B_k} \) is to be increased or decreased, \( \rho = -1 \) implies that \( \tau_{B_k} \) is to be increased and \( \rho = +1 \) implies that \( \tau_{B_k} \) is to be decreased.

The simplex algorithm of linear programming maintains at zero the current deviations of all basic constraints except the \( k \)-th. In other words, the \( \delta_i^+ \) and \( \delta_i^- \) of the constraints \( X_i \beta + \delta_i^+ - \delta_i^- = y_i \), \( i \in I_B \), \( i \neq k \), remain zero and \( \beta_j \), \( j > r \), \( j \neq k \) remain fixed at their upper or lower bound. The deviation of the \( k \)-th basic constraint is increased to a value, \( \theta \), and another constraint enters the basis with a deviation of zero. The algorithm determines the value, \( \theta \), to increase the deviation of the \( k \)-th basic constraint while (a) decreasing the objective value (ignoring degeneracy), (b) maintaining primal feasibility and (c) obtaining dual feasibility in the \( k \)-th basic variable of the dual problem.

If \( k > r \), the algorithm guarantees that the bound restriction on \( \beta_k \) is not violated. The maximum change of \( \beta_k \) possible while still maintaining the feasibility of this constraint is \( U_k - L_k \). Thus, an upper bound on the change in the value of \( \beta_k \) is given by \( \theta_1 \).

\[
\theta_1 = \begin{cases} 
U_k - L_k & \text{for } k > r \\
+\infty & \text{for } k \leq r 
\end{cases}
\]
The algorithm next calculates $\Theta_2$, the maximum change possible in the deviation of the $k$-th basic constraint if the primal feasibility in the $\beta_j$, $j=1,2,\ldots,r$ is to be maintained. Define

$$\xi = \begin{cases} 
X^{-1}_F(k) & \text{for } k \leq r \\
-X^{-1}_F X_R(k) & \text{for } k > r 
\end{cases}$$

where $X^{-1}_F(k)$ is the $k$-th column of $X^{-1}$, and $X_R(k)$ is the $k$-th column of $X_R$.

To calculate the maximum change in the deviation in the $k$-th basic constraint while continually decreasing the absolute sum of the deviations, $\Theta_3$, the algorithm utilizes the basis entry tests of Barrodale and Roberts [4]. These tests determine a nonbasic dual variable which is to enter the basis. The procedure is to calculate the ratio values:

$$\Theta_2 = \min \left\{ \frac{(\beta_j - L_j)}{(-\rho \xi_j)} \right\} \text{ for } \rho \xi_j < 0, j=1,2,\ldots,r;$$

$$\Theta_2 = \min \left\{ \frac{(U_j - \beta_j)}{(\rho \xi_j)} \right\} \text{ for } \rho \xi_j > 0, j=1,2,\ldots,r.$$

The dual variable $\pi_0(p)$ is chosen as a candidate to enter the basis by a ranking of the ratios and the following feasibility check on $\tau_B$:

(i) $|\tau_B| - \sum_{i=1}^{p-1} 2|\phi_i| > 1$ for $k \leq r$

(ii) $|\tau_B| - \sum_{i=1}^{p-1} 2|\phi_i| > 0$ for $k > r$
(iii) \(|\tau_{B_k}^{(i)}| - \sum_{i=1}^{p} 2|\phi_0(i)| \leq 1 \) for \( k \leq r \)

and

(iv) \(|\tau_{B_k}^{(i)}| - \sum_{i=1}^{p} 2|\phi_0(i)| \leq 0 \) for \( k > r \)

where \( \frac{d_0(1)}{\phi_0(1)} \leq \frac{d_0(2)}{\phi_0(2)} \leq \ldots \leq \frac{d_0(p-1)}{\phi_0(p-1)} \leq \frac{d_0(p)}{\phi_0(p)} \)

The value of \( \Theta_3 \) equals \( \frac{d_0(p)}{\phi_0(p)} \). This process is implemented in the computer code version of the algorithm with the partial sort process of Armstrong, Frome and Kung [2].

The value of \( \Theta \) is therefore

(17) \( \Theta = \min(\Theta_1, \Theta_2, \Theta_3) \).

In the case of \( \Theta = \Theta_1 \), the working basis does not change. On the other hand, if \( \Theta \neq \Theta_1 \), a new basis is formed and \( X_F^{-1} \) must be updated. In any event, the nonbasic dual variables corresponding to the candidate list of ratios with ratio values less than \( \Theta \) will remain nonbasic but will switch from their current bound to their opposite bound value. The other changes in the primal and dual variables are given as follows.

If \( \Theta = \Theta_1 \), \( \beta_k \) switches bound value and \( X_B \) is not changed. Relating this situation to the dual problem, the nonbasic \( w_j \) enters the basis and the basic \( v_j \) leaves the basis if \( \beta_k \) switches from its lower to upper bound. On the other hand, if \( \beta_k \) switches from its upper to lower bound value, the nonbasic \( v_j \) enters the basis and the basic \( w_j \) leaves the basis.

If \( \Theta = \Theta_2 \), a nonbasic row of I will enter the basis, and \( X_{B_k} \) will leave the basis, where \( X_{B_k} \) is the \( k \)-th row of the basis \( X_B \). In the dual problem, the nonbasic \( w_j \) or \( v_j \) will enter the basis and \( \tau_{B_k} \) will leave the basis.
If $\theta = \theta_3$, a nonbasic row of $X$ will become basic, the $k$-th row of $X_B$ (associated with the basic dual variable, $\tau_{B_k}$) will leave the basis. In terms of the dual problem, the nonbasic $\pi_s$ will become basic.

There are five cases in the updating process of $X^{-1}_F$.

**Case 1:** If $k \leq r$, $\theta = \theta_3$, this is the situation when a row of $X_B$ is leaving the basis and a row of $X$ is entering the basis. The update of $X^{-1}_F$ is the standard simplex pivot.

**Case 2:** If $k \leq r$, $\theta = \theta_2$, this is the situation when a row of $X_B$ is leaving the basis and a row of $I$ is entering the basis. Since $X^{-1}_F$ contains the rows of $X$ in the basis, the dimension of $X^{-1}_F$ will be decreased by 1.

**Case 3:** If $k > r$, $\theta = \theta_1$, $X^{-1}_F$ does not change.

**Case 4:** If $k > r$, $\theta = \theta_3$, this is the situation when a row of $I$ is leaving the basis, and a row of $X$ is entering the basis. The dimension of $X^{-1}_F$ will be increased by 1.

**Case 5:** If $k > r$, $\theta = \theta_2$, this is the situation when a row of $I$ is leaving the basis and a row of $I$ is entering the basis. The dimension of $X^{-1}_F$ does not change.

The algorithm then updates the deviations by

$$d_i + d_{i'} - \rho \delta i (k) \quad i = 1, 2, \ldots, n.$$  \hspace{1cm} (18)

The algorithm also updates the indicators used for re-ordering purposes.

The iterative process continues until the condition $-1 \leq \tau_{B_i} \leq +1$, $i = 1, 2, \ldots, r$ and $\tau_{B_i} \leq 0$ or $\tau_{B_i} \geq 0$, $i = r+1, r+2, \ldots, m$ is satisfied.

### 3. Steps of the Algorithm

In this section a step-by-step statement of the algorithm is outlined.

**Step 1.** The initial values of $\pi_R$ are found from the following:
(i) If \( L_j \leq \beta_j \leq U_j \), the \( j \)-th element of \( \tilde{\beta} \), \( \tilde{\beta}_j \), is set equal to \( L_j \).
(ii) If \( L_j \leq \beta_j < \infty \), set \( \tilde{\beta}_j = L_j \).
(iii) If \( -\infty < \beta_j \leq U_j \), set \( \tilde{\beta}_j = U_j \).

After interchanges, \( \tilde{\beta} = (\tilde{\beta}_F, \tilde{\beta}_R)^T \), where \( \tilde{\beta}_R \) is a vector of values equal to the upper or lower bound values of the restricted parameters.

Step 2. Formulate the initial basis, \( X_B \):

(i) If all the parameters are restricted, \( X_B \) is an \( m \) by \( m \) identity matrix.
(ii) When all the parameters of the problem do not have bound restrictions, the algorithm attempts to form the basis matrix \( X_B \) by choosing a full rank \( X_F \) matrix from the coefficients of the unrestricted parameters. If this procedure fails to create an initial \( X_B \), the algorithm advances to (iii).
(iii) A full rank matrix \( X_B \) is now formed by including in the initial basis as many additional rows from the identity of (4) as is necessary. The \( \beta_j \)'s corresponding to these rows are assigned upper and lower bounds of plus and minus infinity.

Only the values of \( X_F^{-1} \) and the corresponding parameter indices of \( X_F^{-1} \) are used for the remaining computations.

Step 3. Calculate \( \tilde{\beta}_F \) and \( d_i \) based on (7) and (8). Determine the values of \( \tilde{\pi}_i \), \( i \in NB \), based on the sign of \( d_i \).

Step 4. Calculate \( \tau_{BF} \) and \( \tau_{BR} \) based on (9) and (10).

Step 5. Check for the optimality conditions given by (11), (12) and (13).

If these are satisfied, terminate. Otherwise, go to step 6.
Step 6. Determine the value of $\rho$ from the sign of $\tau_{B_k}$ where $\tau_{B_k}$ is the dual variable most violating a bound restriction.

Step 7. Calculate $\Theta_1$, $\Theta_2$, $\Theta_3$, and $\Theta$ from (14), (15), (16) and (17).

Step 8. If $\Theta = \Theta_1$, $\beta_k$ switches to its opposite bound value. Go to step 12.

Step 9. If $\Theta = \Theta_2$, the dimension of $X_F^{-1}$ is decreased by 1 if $k \leq r$. On the other hand, the dimension of $X_F^{-1}$ does not change if $k > r$. Go to step 11.

Step 10. If $\Theta = \Theta_3$, the dimension of $X_F^{-1}$ remains the same if $k \leq r$. If $k > r$, the dimension of $X_F^{-1}$ will be increased by 1. Go to step 11.

Step 11. Update $X_F^{-1}$ and the other basic indicators.

Step 12. Update the deviations based on (18), and the values of $\pi_i$, $i \in NB$, based on the sign of the updated values of $d_i$. Go to step 4.

4. Computational Results

To evaluate the efficiency of the algorithm given here, a FORTRAN version of the algorithm presented here called RESLI was tested against the algorithm INTBND developed by Armstrong and Hultz [3]. The INTBND code solves least absolute value problems with arbitrary linear constraints on the parameters. Both codes employ the revised simplex method of linear programming. The observations for the study have been drawn from various uniform distributions using a random number generator. All the problems were solved by the CDC 6600 in the University of Texas at Austin. The results presented in Table 1 are mean times and iteration counts for a set of 5 problems with the same characteristics.
Table 1

(A comparison between RESL1 and INTBND)

<table>
<thead>
<tr>
<th>Number of parameters</th>
<th>Number of observations</th>
<th>Number of unrestricted parameters</th>
<th>Time (in CPU milliseconds)</th>
<th>Number of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>RESL1</td>
<td>INTBND</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>0</td>
<td>29</td>
<td>56</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>2</td>
<td>75</td>
<td>115</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>0</td>
<td>158</td>
<td>376</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>5</td>
<td>679</td>
<td>968</td>
</tr>
<tr>
<td>15</td>
<td>150</td>
<td>0</td>
<td>323</td>
<td>1004</td>
</tr>
<tr>
<td>15</td>
<td>150</td>
<td>15</td>
<td>2750</td>
<td>5629</td>
</tr>
<tr>
<td>15</td>
<td>200</td>
<td>0</td>
<td>830</td>
<td>1671</td>
</tr>
<tr>
<td>15</td>
<td>200</td>
<td>10</td>
<td>2111</td>
<td>5463</td>
</tr>
<tr>
<td>20</td>
<td>200</td>
<td>0</td>
<td>1383</td>
<td>3016</td>
</tr>
<tr>
<td>20</td>
<td>200</td>
<td>12</td>
<td>6230</td>
<td>9334</td>
</tr>
<tr>
<td>20</td>
<td>250</td>
<td>0</td>
<td>1693</td>
<td>3897</td>
</tr>
<tr>
<td>20</td>
<td>250</td>
<td>14</td>
<td>7565</td>
<td>12566</td>
</tr>
<tr>
<td>25</td>
<td>250</td>
<td>0</td>
<td>1263</td>
<td>5663</td>
</tr>
<tr>
<td>25</td>
<td>250</td>
<td>20</td>
<td>8970</td>
<td>14445</td>
</tr>
<tr>
<td>25</td>
<td>300</td>
<td>0</td>
<td>1311</td>
<td>6086</td>
</tr>
<tr>
<td>25</td>
<td>300</td>
<td>18</td>
<td>14128</td>
<td>18751</td>
</tr>
<tr>
<td>30</td>
<td>300</td>
<td>0</td>
<td>2543</td>
<td>9142</td>
</tr>
<tr>
<td>15</td>
<td>500</td>
<td>10</td>
<td>7644</td>
<td>16872</td>
</tr>
<tr>
<td>15</td>
<td>500</td>
<td>0</td>
<td>1673</td>
<td>4643</td>
</tr>
</tbody>
</table>
Our computational study indicates that the algorithm RESL1 is consistently faster than INTBND on all problem sizes. Also, RESL1 utilizes a reduced basis inverse and requires less computer storage than INTBND.

4. Conclusion

Since the time of Laplace [16], minimizing the sum of absolute deviations has been considered as a criterion for parameter estimation. Recently, LAV estimators have been widely examined as a robust estimation technique [11] and with this interest has come a demand for efficient algorithms to provide LAV estimates for a variety of statistical models. This demand has been satisfied primarily by the specialization of linear programming to take advantage of the distinctive characteristics of the problem.

This paper presents an approach to solve least absolute value problems with bound restrictions on the parameters. The algorithm presented here exploits the structure of the problem to maintain a compact representation of the basis inverse. All computations required in the algorithm can be performed by means of a reduced basis. The algorithm also combines several standard simplex pivots into one. Computational experience with a FORTRAN version of this algorithm compared to a FORTRAN version of the algorithm developed by Armstrong and Hultz [3] indicates its superiority in terms of computer time and storage.
REFERENCES


This paper presents a special purpose linear programming algorithm to solve a least absolute value regression problem with upper and lower bounds on the parameters. The algorithm exploits the problem's special structure by maintaining a compact representation of the basis inverse and by allowing for the capability to combine several simplex iterations into one. Computational results with a computer code implementation of the algorithm is given.
Regression
Least Absolute Value
Upper Bounding