AD-A084 284 TEXAS UNIV AT AUSTIN CENTER FOR CYBERNETIC STUDIES F/G 12/1

GENERALIZED NETWORK APPROACHES FOR SOLVING LEAST ABSOLUTE VALUE--ETC(U)

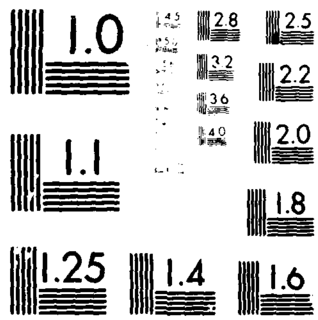
DEC 79 D KLINGMAN, J MOTE N00014-78-C-0222

unclassified CCS-365 NL

1 OF 1

AD
A084284

END
DATE
FILMED
6-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(1) LEVEL II

CENTER FOR CYBERNETIC STUDIES

The University of Texas
Austin, Texas 78712

CCS 365

GENERALIZED NETWORK APPROACHES FOR SOLVING LEAST ABSOLUTE VALUE AND TCHEBYCHEFF REGRESSION PROBLEMS

by

Darwin Klingman*

John Mote**

December 1979

* Professor of Operations Research and Computer Sciences, The University of Texas at Austin, Department of General Business, BEB 608, Austin, TX 78712.

** Systems Analyst, Analysis, Research, & Computation, Inc., P.O. Box 4067, Austin, TX 78765.

This research was supported in part by the Office of Naval Research contract no. N00014-78-C-0222. Reproduction in whole or in part is permitted for any purpose of the United States Government.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
BEB 203E
The University of Texas at Austin
Austin, Texas 78712
(512) 471-1821

1. INTRODUCTION

This paper studies two classes of bivariate regression analysis problems that can be formulated and solved quite efficiently as generalized network flow problems.

Regression analysis techniques are used in many situations to determine the "best" relationship between an independent variable X and a dependent variable Y.

If $\{(x_k, y_k)\}$ is the set of n paired observations of an independent and a dependent variable, then the bivariate regression problem is to determine a linear regression equation of the form:

$$\hat{y}_k = A + Bx_k \tag{1.1}$$

such that the vector of predicted values of the dependent variable is *close* to the vector of observed values. The classical least squares method of regression analysis uses the Euclidian norm to measure the closeness of the vectors of predicted and observed values [9]. Using this norm, the unweighted bivariate least squares regression problem can be stated as:

$$\text{Minimize} \quad \left(\sum_{k=1}^{n} (\hat{y}_k - y_k)^2 \right)^{1/2}$$

subject to:

$$\hat{y}_k = A + Bx_k \text{ for } k = 1, 2, \ldots, n.$$

This classical problem has well known closed-form solutions for the slope, B, and intercept, A, of the regression equation (1.1).

The Euclidian norm is one member of the L_p family of norms. For a specified value of $p \geqq 1$, the unweighted bivariate L_p regression equation is given as the solution to the following problem:

$$\text{Minimize} \quad \left(\sum_{k=1}^{n} |\hat{y}_k - y_k|^p \right)^{1/p} \qquad (1.2)$$

subject to:

$$\hat{y} = A + Bx_k \text{ for } k = 1, 2, \ldots, n$$

The classical least squares, or L_2, regression problem is the only member of the L_p family of regression problems that possesses a closed-form solution. In general, a nonlinear optimization problem, (1.2), must be solved in order to determine the optimal L_p regression equation. Since this is often an impractical means of solving a problem, most of the members of the L_p family of regression problems, other than the classical least squares problem, have been ignored in the literature. Two notable exceptions are the L_1, or least absolute value, regression problem and the L_∞, or Tchebycheff, regression problem.

Historically, most of the proposed solution procedures for the L_1 (multiple) regression problem have been based on the efficient simplex algorithm of linear programming. However, the specific implementations of the basic algorithm have been quite varied. Charnes, Cooper, and Ferguson [8] were the first to show that a primal formulation of the L_1 problem could be solved with the primal simplex algorithm. Later, Wagner [16] suggested that it would be more efficient to solve a dual formulation of the problem since the size of the working basis would be greatly reduced. This conjecture was widely accepted until Barrodale and Roberts [6] showed that a specialized primal simplex algorithm could be used to capitalize on the underlying structure of the primal problem. McCormick and Sposito [11] proposed an improved Barrodale and Roberts algorithm which uses the least squares regression problem to speed convergence. Abdelmalek [1, 2] suggested that a dual simplex algorithm could be used to solve a dual formulation of the L_1 problem. The equivalence of the Barrodale and Roberts approach and the Abdelmalek approach was demonstrated by Armstrong and Godfrey [3]. A specialized version of the Barrodale and Roberts procedure was developed by Armstrong and Kung [5] to solve the unweighted bivariate L_1 regression problem. Their approach is currently regarded as the most efficient one for this class of problems.

Like the L_1 problem, most of the suggested solution procedures for the L_∞ (multiple) regression problem have been based on the efficient simplex algorithm [4, 7, 14, 15].

This paper further investigates the bivariate L_1 and L_∞ regression problems. Specifically, it is shown that both problems are equivalent to generalized network problems with very special network topologies. As a result, solution algorithms that exploit the network topologies are developed. Computer implementations of both algorithms are shown to be very efficient and highly robust. This paper also explores the underlying relationship between the bivariate L_1 and L_∞ regression problems. It is shown that the L_∞ problem is a *relaxation* of the L_1 problem, and as such it should be easier to solve. This conjecture is supported by the computer testing of the two network-based algorithms.

2. WEIGHTED BIVARIATE L_1 REGRESSION ANALYSIS

Given a set of paired observations, $\{(x_k,y_k)\}$, and a set of n positive weights, $\{w_k\}$, the weighted bivariate L_1 regression problem is to determine a linear regression equation that minimizes the weighted sum of the absolute deviations (also residuals or errors). Formally stated, the problem is:

$$\text{Minimize} \quad \sum_{k=1}^{n} w_k \mid \hat{y}_k - y_k \mid \tag{2.1}$$

subject to:

$$\hat{y}_k = A + Bx_k \text{ for } k = 1, 2, \ldots, n$$

It is well known that this problem can be formulated as a linear programming problem with n + 2 structural variables and 2n constraints. This is accomplished by introducing a deviation variable, D_k, and two constraints, $\hat{y}_k - y_k \leq D_k$ and $-(\hat{y}_k - y_k) \leq D_k$, for each of the n pairs of observations. Since each weight, w_k, is assumed to be positive, D_k corresponds to the absolute deviation associated with the k^{th} data point. That is,

$$D_k = |\hat{y}_k - y_k|.$$

The complete linear programming formulation of the weighted bivariate L_1 regression problem is given by:

$$\text{Minimize} \quad \sum_{k=1}^{n} w_k D_k \tag{2.2}$$

subject to:

$$-A - x_k B + D_k \geq - y_k \text{ for } k = 1, 2, \ldots, n \tag{2.3}$$

$$A + x_k B + D_k \geq y_k \text{ for } k = 1, 2, \ldots, n \tag{2.4}$$

All structural variables in this primal linear programming problem are treated as unrestricted variables, even though constraints (2.3) and (2.4) insure that each deviation variable, D_k, is non-negative. The transformation of problem (2.2)-(2.4) to its dual linear programming problem is simplified if each deviation variable is considered as an unrestricted variable.

In order to carry out the transformation of problem (2.2)-(2.4) to its dual problem, it is necessary to introduce a dual variable for each primal constraint. Let α_k be the dual variable associated with the k^{th} primal constraint, (2.3), and let β_k be the dual variable associated with the $(n + k)^{th}$ primal constraint, (2.4). Then the dual of problem (2.2)-(2.4) is:

$$\text{Maximize} \; -\sum_{k=1}^{n} y_k \alpha_k + \sum_{k=1}^{n} y_k \beta_k \tag{2.5}$$

subject to:

$$-\sum_{k=1}^{n} \alpha_k + \sum_{k=1}^{n} \beta_k = 0 \tag{2.6}$$

$$-\sum_{k=1}^{n} x_k \alpha_k + \sum_{k=1}^{n} x_k \beta_k = 0 \tag{2.7}$$

$$\alpha_k + \beta_k = W_k \text{ for } k = 1, 2, \ldots, n \tag{2.8}$$

$$\alpha_k \geqq 0, \ \beta_k \geqq 0 \tag{2.9}$$

The β_k variables can be eliminated from the dual linear programming problem by using equations (2.8). This is why each of the deviation variables in the primal problem were treated as unrestricted variables.

The following problem is obtained from problem (2.5)-(2.9) by eliminating the β_k variables, scaling equations (2.5) and (2.7) by -1/2, and scaling equation (2.6) by 1/2.

$$\text{Minimize} \quad \sum_{k=1}^{n} y_k \alpha_k - \frac{\sum_{k=1}^{n} w_k y_k}{2} \tag{2.10}$$

subject to:

$$-\sum_{k=1}^{n} \alpha_k = \frac{-\sum_{k=1}^{n} w_k}{2} \tag{2.11}$$

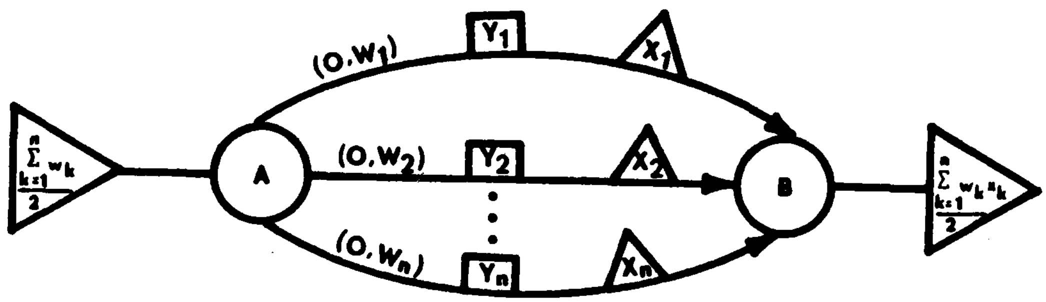
$$\sum_{k=1}^{n} x_k \alpha_k = \frac{\sum_{k=1}^{n} w_k x_k}{2} \tag{2.12}$$

$$0 \leqq \alpha_k \leqq w_k \tag{2.13}$$

This formulation of the dual problem is a capacitated generalized network problem with two nodes and n arcs. Figure 1 illustrates the special network topology of this problem.

FIGURE 1

L_1 GENERALIZED NETWORK



For reasons that will become apparent, the node associated with constraint (2.11) will be labeled A and the node associated with constraint (2.12) will be labeled B. Node A has a net supply of $\sum_{k=1}^{n} w_k/2$ units and node B has a net demand of $\sum_{k=1}^{n} w_k x_k/2$ units. It is interesting to note that the net gain through the network is equal to the weighted mean of observations of the independent variable

$$\frac{\sum_{k=1}^{n} w_k x_k}{\sum_{k=1}^{N} w_k}$$

All arcs for this network problem are directed from node A to node B. The k^{th} arc has a lower bound of zero, upper bound of w_k, multiplier of x_k, and objective function coefficient of y_k.

At this point it will be assumed that $x_i \neq x_j$ for some i and j. Otherwise, the regression problem is meaningless. This assumption is equivalent to assuming that the constraint matrix for the generalized network problem, (2.10)-(2.13), has full row rank.

The generalized network problem always has a finite optimal solution. This is true since $\alpha_k = w_k/2$ is always a feasible solution, and constraints (2.13) bound the feasible region. Since problem (2.10)-(2.13) always has a finite optimal solution, duality theory ensures that the original problem, (2.1), also possesses a finite optimal solution.

Since problem (2.10)-(2.13) is a capacitated generalized network problem, it can be solved using a specialized network-based revised simplex algorithm.

Every basis matrix for the generalized network problem has the form:

$$B = \begin{bmatrix} -1 & -1 \\ x_{k_1} & x_{k_2} \end{bmatrix}$$

where k_1 and k_2 are the indices of the two basic arcs. The notation B for the basis matrix should not be confused for the label of the node associated with constraint (2.12). Since B must have a rank of two, $x_{k_1} \neq x_{k_2}$. The basis inverse is

$$B^{-1} = \left(\frac{1}{x_{k_1} - x_{k_2}}\right) \begin{bmatrix} x_{k_2} & 1 \\ -x_{k_1} & -1 \end{bmatrix}.$$

Let U be an index set of arcs that are non-basic at their upper bounds. Let

$$b_A = \frac{-\sum_{k=1}^{n} w_k}{2} + \sum_{k \in U} w_k$$

be the net demand (negative supply) at node A after the non-basic arcs have been taken into consideration. Similarly, let

$$b_B = \frac{\sum_{k=1}^{n} w_k x_k}{2} - \sum_{k \in U} w_k x_k$$

be the net demand at node B.

The values of the flows on the two basic arcs are given by:

$$f = B^{-1} b \qquad (2.14)$$

where

$$f = \begin{pmatrix} \alpha_{k_1} \\ \alpha_{k_2} \end{pmatrix} \text{ and } b = \begin{pmatrix} b_A \\ b_B \end{pmatrix}.$$

A network representation of the basis is not needed since (2.14) has the closed-form solution:

$$\alpha_{k_1} = \frac{b_B + b_A x_{k_2}}{x_{k_1} - x_{k_2}}$$

$$(2.15)$$

$$\alpha_{k_2} = \frac{b_B + b_A x_{k_1}}{x_{k_2} - x_{k_1}}$$

A basis is feasible if (2.15) satisfies constraints (2.13) for arcs k_1 and k_2. The selection of an initial feasible basis is greatly simplified by adding an artificial arc to the generalized network problem. It is convenient to number the artificial arc 0. Like the n real arcs, the artificial arc is directed from node A to node B.

Let U be the index set of arcs such that $b_A < 0$. U is simply the set of real arcs that are initially selected to be non-basic at their upper bounds. U can either be initialized to be empty or a heuristic rule can be used to select the arcs assigned to U.

Given an initial choice of U, the multiplier for the artificial arc is defined as

$$x_o = \frac{-b_B}{b_A} .$$

The artificial arc is given an infinite objective function coefficient in order to insure that its flow in the optimal solution is zero.

Using this artificial arc, the choice of an initial feasible basis is very simple. Let $k_1 = 0$ denote that the artificial arc is the first basic arc. For the second basic arc, any arc $k_2 \notin U$ can be selected provided that $x_{k_1} \neq x_{k_2}$. The flows on these two basic arcs are given by (2.15). Due to the specific choice of x_o that was made, the flows on the two basic arcs are simply $\alpha_{k_1} = -b_A$ and $\alpha_{k_2} = 0$.

The specialized revised simplex algorithm for solving generalized network problems uses node potentials in order to simplify the selection of entering arcs. The node potentials are defined as

$$\pi = c_B B^{-1} \qquad (2.16)$$

where $\pi = (\pi_A, \pi_B)$ and $c_B = (y_{k_1}, y_{k_2})$. A network representation of the basis is not needed since (2.16) has the closed-form solution :

$$\pi_A = \frac{x_{k_2} y_{k_1} - x_{k_1} y_{k_2}}{x_{k_1} - x_{k_2}} \tag{2.17}$$

$$\pi_B = \frac{y_{k_1} - y_{k_2}}{x_{k_1} - x_{k_2}}$$

By applying duality theory to the generalized network problem, (2.10)-(2.13), a given basic feasible solution is an optimal solution if

$$-\pi_A + x_k \pi_B \leqq y_k \quad \text{if } \alpha_k = 0$$
$$\tag{2.18}$$
$$-\pi_A + x_k \pi_B \geqq y_k \quad \text{if } \alpha_k = w_k$$

If for some arc k_e, either

$$-\pi_A + x_{k_e} \pi_B > y_{k_e} \quad \text{and } \alpha_{k_e} = 0 \tag{2.19}$$

or

$$-\pi_A + x_{k_e} \pi_B < y_{k_e} \quad \text{and } \alpha_{k_e} = w_{k_e} \tag{2.20}$$

then the current solution is not optimal and arc k_e is a candidate to enter the basis (i.e., pivot eligible).

After selecting an arc to enter the basis the specialized revised simplex algorithm for the generalized network problem must compute the representation of the entering arc in terms of the current basis. The representation of arc k_e is given by

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = B^{-1}\begin{pmatrix} -1 \\ x_{k_e} \end{pmatrix} = \begin{pmatrix} \dfrac{x_{k_e} - x_{k_2}}{x_{k_1} - x_{k_2}} \\[2ex] \dfrac{x_{k_1} - x_{k_e}}{x_{k_1} - x_{k_2}} \end{pmatrix}$$

The simplex algorithm uses this representation to perform the standard minimum ratio test. Specifically, if arc k_e is initially non-basic with zero flow, then the minimum ratio is given by

$$\Delta = \min\left\{ w_{k_e},\ \min_i\left\{ \frac{\alpha_{k_i}}{Y_i} \;\middle|\; Y_i > 0 \right\},\ \min_i\left\{ \frac{w_{k_i} - \alpha_{k_i}}{-Y_i} \;\middle|\; Y_i < 0 \right\} \right\}.$$

If arc k_e is initially non-basic at its upper bound, then the minimum ratio is given by

$$\Delta = \min\left\{ w_{k_e},\ \min_i\left\{ \frac{\alpha_{k_i}}{-Y_i} \;\middle|\; Y_i < 0 \right\},\ \min_i\left\{ \frac{w_{k_i} - \alpha_{k_i}}{Y_i} \;\middle|\; Y_i > 0 \right\} \right\}.$$

If $\Delta = w_{k_e}$, then arc k_e remains non-basic but it is placed at its opposite bound. Otherwise, arc k_e becomes a basic arc and the arc that yielded the minimum ratio is made non-basic at its appropriate bound.

The status of the α_k variables in the optimal solution to the generalized network problem, (2.10)-(2.13), provides some useful insight into the relationship between the original problem data, $\{(x_k, y_k)\}$, and the optimal regression equation. From duality theory, it is known that the basic variables in the optimal solution to the dual problem correspond to the binding constraints in the primal problem. This means that the optimal regression line passes directly through the two data points corresponding to the optimal basis of the generalized network problem. This result can be used to derive the formulas for the slope and intercept of the optimal regression equation. The slope of the line through points (x_{k_1}, y_{k_1}) and (x_{k_2}, y_{k_2}) is:

$$B = \frac{y_{k_1} - y_{k_2}}{x_{k_1} - x_{k_2}} = \pi_B \tag{2.21}$$

and the intercept of the line is:

$$A = y_{k_1} - Bx_{k_1} = \frac{x_{k_1} y_{k_1} - x_{k_2} y_{k_1}}{x_{k_1} - x_{k_2}} = -\pi_A \tag{2.22}$$

It is this relationship between the node potentials and the optimal regression equation that motivated the choice of A and B as the node names.

Using (2.21) and (2.22), the optimality conditions (2.18), become:

$$A + Bx_k \leq y_k \quad \text{if } \alpha_k = 0$$

$$A + Bx_k \geq y_k \quad \text{if } \alpha_k = w_k.$$

In other words, all observations that lie above (below) the optimal regression line have dual variables that are non-basic at zero (upper bound).

This relationship between the sign of the error term and the non-basic status of the arcs for problem (2.10)-(2.13) can be used to develop heuristics for the initial selection of the set U.

This network-based simplex algorithm has been coded in FORTRAN and tested on a CDC 6600. Computational testing of this code and a state-of-the-art non-network code are presented in Section 5. This testing indicates the superiority of the network approach.

3. BIVARIATE L_∞ REGRESSION ANALYSIS

Given a set of n paired observations, $\{(x_k, y_k)\}$, the bivariate L_∞ regression problem is to determine a linear regression equation that minimizes the largest absolute deviation. Formally stated, the problem is

$$\begin{aligned} &\text{Minimize} \quad \max_k \left| \hat{y}_k - y_k \right| \\ &\text{subject to:} \\ &\qquad \hat{y}_k = A + Bx_k \quad \text{for } k = 1, 2, \ldots, n \end{aligned} \tag{3.1}$$

This problem can be formulated as a linear programming problem with three structural variables and 2n constraints. The formulation is

$$\text{Minimize} \quad D \tag{3.2}$$

subject to:

$$-A - x_k B + D \geq -y_k \quad \text{for } k = 1, 2, \ldots, n \tag{3.3}$$

$$A + x_k B + D \geq y_k \quad \text{for } k = 1, 2, \ldots, n \tag{3.4}$$

All three structural variables are treated as unrestricted variables even though constraints (3.3) and (3.4) ensure that the deviation variable, D, is non-negative. Like the L_1 case, the treatment of D as an unrestricted variable simplifies the transformation of problem (3.2)-(3.4) to its dual linear programming problem. In order to carry out this transformation it is necessary to introduce a dual variable for each primal constraint. Let α_k be the dual variable associated with the k^{th} primal constraint, (3.3), and let β_k be the dual variable associated with the $(n + k)^{th}$ primal constraint, (3.4). The dual of problem (3.2)-(3.4) is

$$\text{Maximize} \quad -\sum_{k=1}^{n} y_k \alpha_k + \sum_{k=1}^{n} y_k \beta_k \tag{3.5}$$

subject to:

$$-\sum_{k=1}^{n} \alpha_k + \sum_{k=1}^{n} \beta_k = 0 \tag{3.6}$$

$$-\sum_{k=1}^{n} x_k \alpha_k + \sum_{k=1}^{n} x_k \beta_k = 0 \tag{3.7}$$

$$\sum_{k=1}^{n} \alpha_k + \sum_{k=1}^{n} \beta_k = 1 \qquad (3.8)$$

$$\alpha_k \geqq 0, \ \beta_k \geqq 0 \text{ for } k = 1, 2, \ldots, n \qquad (3.9)$$

The following formulation is equivalent to (3.5)-(3.9) and therefore can be used to solve the original problem.

$$\text{Minimize} \quad \sum_{k=1}^{n} y_k \alpha_k + \sum_{k=1}^{n} \left(\frac{-y_k}{x_k}\right) \beta_k \qquad (3.10)$$

subject to:

$$-\sum_{k=1}^{n} \alpha_k = -1/2 \qquad (3.11)$$

$$\sum_{k=1}^{n} x_k \alpha_k - \sum_{k=1}^{n} \beta_k = 0 \qquad (3.12)$$

$$\sum_{k=1}^{n} \left(\frac{1}{x_k}\right) \beta_k = 1/2 \qquad (3.13)$$

$$\alpha_k \geqq 0, \ \beta_k \geqq 0 \quad \text{for } k = 1, 2, \ldots, n \qquad (3.14)$$

This new formulation is obtained by replacing equation (3.6) with one-half times equation (3.6) less one-half times equation (3.8), replacing equation (3.8) with one-half times equation (3.6) plus one-half times equation (3.8), scaling the objective function (3.5) by -1/2, and scaling each β_k variable by $1/x_k$. The scaling of each β_k implicitly assumes that each x_k is positive. It will be shown later that this is not a critical assumption.

The new formulation of the dual problem, (3.10)-(3.14), is an uncapacitated generalized network problem with three nodes and 2n arcs. Figure 2 illustrates the special network topology of this problem.

The nodes corresponding to constraints (3.11), (3.12), and (3.13) will be labeled A, B, and D, respectively. Node A has a net supply of 1/2 units and node D has a net demand of 1/2 units.

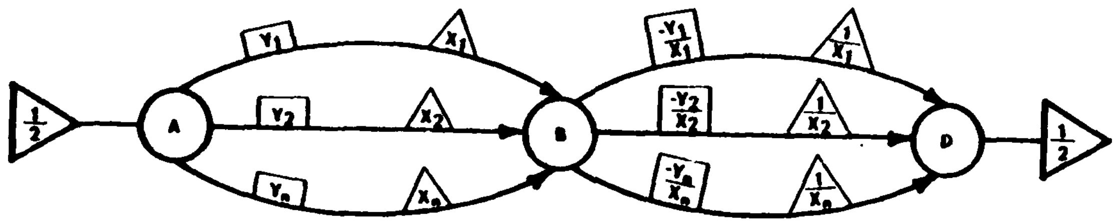
The arcs corresponding to the α_k variables are directed from node A to node B. The multipliers on these arcs are x_k and the objective function coefficients are y_k. The arcs corresponding to the β_k variables are directed from node B to node D. The multipliers on these arcs are $1/x_k$ and the objective function coefficients are $-y_k/x_k$. All 2n arcs have lower bounds of zero.

At this point it will be assumed that $x_k \neq x_j$ for some i and j. Otherwise, the regression problem is meaningless. This assumption is equivalent to assuming that the constraint matrix for the generalized network problem, (3.10)-(3.14), has full row rank.

The generalized network problem always has a finite optimal solution. This is true since the network is acyclic and $\alpha_1 = 1/2$, $\beta_1 = x_1/2$, and $\alpha_k = \beta_k = 0$ for $k > 1$ is always a feasible solution.

FIGURE 2

L_∞ GENERALIZED NETWORK



Since problem (3.10)-(3.14) is a generalized network problem, it can be solved using a specialized revised simplex algorithm. Since the problem is uncapacitated, at most three arcs will have non-zero flows when a simplex approach is used to solve the problem. That is, only the three arcs associated with a given basis can have non-zero flows. All non-basic arcs must have zero flow.

Due to the special topology of the generalized network problem, only two basis matrix structures are possible. For obvious reasons these two structures will be referred to as type α and type β bases.

The type α basis consists of two α_k arcs (k_1 and k_3) and one β_k arc (k_2). The type α basis matrix,

$$B = \begin{bmatrix} -1 & 0 & -1 \\ x_{k_1} & -1 & x_{k_3} \\ 0 & \frac{1}{x_{k_2}} & 0 \end{bmatrix} ,$$

has full row rank only if $x_{k_1} \neq x_{k_3}$.

The type β basis consists of two β_k arcs (k_1 and k_3) and one α_k arc (k_2). The type β basis matrix,

$$B = \begin{bmatrix} 0 & -1 & 0 \\ -1 & x_{k_2} & -1 \\ \frac{1}{x_{k_1}} & 0 & \frac{1}{x_{k_3}} \end{bmatrix} ,$$

has full rank only if $x_{k_1} \neq x_{k_3}$.

Since B must have full rank, it will be assumed that the basic arcs are numbered such that $x_{k_1} > x_{k_3}$.

The values of the flows on the three basic arcs are given by

$$f = B^{-1}b \tag{3.15}$$

where $b = \begin{pmatrix} -1/2 \\ 0 \\ 1/2 \end{pmatrix}$,

and $f = \begin{pmatrix} \alpha_{k_1} \\ \beta_{k_2} \\ \alpha_{k_3} \end{pmatrix}$ if B is type α

or $f = \begin{pmatrix} \beta_{k_1} \\ \alpha_{k_2} \\ \beta_{k_3} \end{pmatrix}$ if B is type β.

For a type α basis, the solution to (3.15) is given by:

$$\alpha_{k_1} = \frac{x_{k_2} - x_{k_3}}{2(x_{k_1} - x_{k_3})}$$

$$\beta_{k_2} = \frac{x_{k_2}}{2}$$

$$\alpha_{k_3} = \frac{x_{k_1} - x_{k_2}}{2(x_{k_1} - x_{k_3})}$$

Since α_{k_1} must be non-negative in order for B to be a feasible basis, and since $x_{k_1} > x_{k_3}$, it is necessary for $x_{k_2} \geq x_{k_3}$. Similarly, since α_{k_3} must be non-negative, $x_{k_1} \geq x_{k_2}$. So B is a feasible type α basis if and only if $x_{k_1} \geq x_{k_2} \geq x_{k_3}$ and $x_{k_1} > x_{k_3}$.

For a type β basis, the solution to (3.15) is given by:

$$\beta_{k_1} = \frac{x_{k_1}(x_{k_2} - x_{k_3})}{2(x_{k_1} - x_{k_3})}$$

$$\alpha_{k_2} = 1/2$$

$$\beta_{k_3} = \frac{x_{k_3}(x_{k_1} - x_{k_2})}{2(x_{k_1} - x_{k_3})}$$

Since β_{k_1} and β_{k_3} must be non-negative, B is a feasible type β basis if and only if $x_{k_1} \geq x_{k_2} \geq x_{k_3}$ and $x_{k_1} > x_{k_3}$.

The node potentials associated with a given basis are given by:

$$\pi = c_B B^{-1} \tag{3.16}$$

where $\pi = (\pi_A, \pi_B, \pi_D)$,

and $c_B = \left(y_{k_1}, \frac{-y_{k_2}}{x_{k_2}}, y_{k_3}\right)$ if B is type α

or $$c_B = \left(\frac{-y_{k_1}}{x_{k_1}}, y_{k_2}, \frac{-y_{k_3}}{x_{k_3}} \right) \text{ if B is type } \beta.$$

For a type α basis, the solution to (3.16) is:

$$\pi_B = \frac{y_{k_1} - y_{k_3}}{x_{k_1} - x_{k_3}}$$

$$\pi_A = x_{k_1}\pi_B - y_{k_1} = x_{k_3}\pi_B - y_{k_3}$$

$$\pi_D = x_{k_2}\pi_B - y_{k_2}.$$

For a type β basis, the solution to (3.16) is:

$$\pi_B = \frac{y_{k_1} - y_{k_3}}{x_{k_1} - x_{k_3}}$$

$$\pi_A = x_{k_2}\pi_B - y_{k_2}$$

$$\pi_D = x_{k_1}\pi_B - y_{k_1} = x_{k_3}\pi_B - y_{k_3}.$$

A network representation of the basis is not needed since all primal and dual variables have closed-form solutions.

The value of the objective function associated with a given basis is

$$z = c_B f.$$

For a type α basis

$$z = \frac{y_{k_1}(x_{k_2} - x_{k_3}) - y_{k_2}(x_{k_1} - x_{k_3}) + y_{k_3}(x_{k_1} - x_{k_2})}{2(x_{k_1} - x_{k_3})}$$

and for a type β basis

$$z = \frac{-y_{k_1}(x_{k_2} - x_{k_3}) + y_{k_2}(x_{k_1} - x_{k_3}) - y_{k_3}(x_{k_1} - x_{k_2})}{2(x_{k_1} - x_{k_3})} .$$

The determination of an initial feasible basis is trivial for problem (3.10)-(3.14). It is only necessary to select three observations, k_1, k_2, and k_3, such that $x_{k_1} \geq x_{k_2} \geq x_{k_3}$ and $x_{k_1} > x_{k_3}$. Let

$$\theta = y_{k_1}(x_{k_2} - x_{k_3}) - y_{k_2}(x_{k_1} - x_{k_3}) + y_{k_3}(x_{k_1} - x_{k_2}).$$

If $\theta \leq 0$, let α_{k_1}, β_{k_2} and α_{k_3} form the initial type α basis. Otherwise, let β_{k_1}, α_{k_2} and β_{k_3} form the initial type β basis.

Given a feasible basis, it is necessary to examine the non-basic arcs in order to either select an entering arc or to verify optimality. From the application of duality theory to the generalized network problem, (3.10)-(3.14), a basis is optimal if

$$\pi_D \leq x_k\pi_B - y_k \leq \pi_A \quad \text{for } k = 1, 2, \ldots, n.$$

If $x_k\pi_B - y_k > \pi_A$, then α_k is a candidate to enter the basis. If $x_k\pi_B - y_k < \pi_D$, then β_k is a candidate to enter the basis. The *most* pivot eligible arc can easily be identified by solving the following two subproblems.

$$\max_k z^{\alpha} = x_k\pi_B - y_k \tag{3.17}$$

$$\min_k z^{\beta} = x_k\pi_B - y_k \tag{3.18}$$

If $z^{\alpha} = \pi_A$ and $z^{\beta} = \pi_D$, then the current solution is optimal. If $z^{\alpha} - \pi_A \geq \pi_D - z^{\beta}$, then the α_{k_e} corresponding to the observation that solved (3.17) is the most pivot eligible arc. Otherwise, the β_{k_e} corresponding to the observation that solved (3.18) is the most pivot eligible arc.

After selecting an entering arc, the standard simplex minimum ratio test can be used to determine the leaving arc. However, due to the special structure of the generalized network problem, the minimum ratio test can be performed entirely with logical, instead of arithmetic, operations.

In order to illustrate how the logical operations were derived, assume that the current basis is type α. If α_{k_e} is selected to enter the basis, then either α_{k_1} or α_{k_3} must leave the basis since a basis consisting of three α_k variables is not feasible.

Since $x_{k_1} \geqq x_{k_2} \geqq x_{k_3}$, α_{k_1} must leave the basis if $x_{k_e} > x_{k_2}$. This is the only selection that leaves x_{k_2} in the middle. Likewise, if $x_{k_e} < x_{k_2}$, then α_{k_3} must leave the basis. If $x_{k_e} = x_{k_2}$, then either α_{k_1} or α_{k_3} can leave the basis.

Now consider the case where the initial basis is type α and β_{k_e} is selected as the entering arc. There are three possibilities to consider.

First, if $x_{k_e} > x_{k_1}$, then α_{k_3} must leave the basis. In order to maintain $x_{k_1} \geqq x_{k_2} \geqq x_{k_3}$, the arcs are renumbered as follows

$$k_3 = k_2$$

$$k_2 = k_1$$

$$k_1 = k_e.$$

Second, if $x_{k_1} \geqq x_{k_e} \geqq x_{k_3}$, then β_{k_2} must leave the basis.

Finally, if $x_{k_3} > x_{k_e}$, then α_{k_1} must leave the basis. The arcs are renumbered as follows

$$k_1 = k_2$$

$$k_2 = k_3$$

$$k_3 = k_e.$$

In the first and third cases, the new basis is type β. In the second case, the new basis is still type α.

A similar analysis can be done for the case where the initial basis is type β.

After solving the generalized network problem, it is necessary to translate the optimal solution to the dual problem, (3.10)-(3.14), back into an optimal solution for the original L_∞ regression problem, (3.1). By simply reversing the steps of the transformation of the dual linear programming problem, the following intercept and slope of the L_∞ regression equation are obtained

$$A = \frac{-(\pi_A + \pi_D)}{2}$$

$$B = \pi_B$$

The maximum absolute error is

$$\max_k \left| \hat{y}_k - y_k \right| = \frac{\pi_A - \pi_D}{2} .$$

As stated earlier, it is assumed that all x_k are positive. This is not a critical assumption since the coordinate system can be translated by adding a constant, c, to each x_k, such that $x_k + c$ is positive for all observations. After solving the transformed problem, the correct intercept can be obtained by adding c times the slope to the intercept.

This network-based simplex algorithm has been coded in FORTRAN and tested on a CDC 6600. Computational testing is presented in Section 5.

4. RELATIONSHIP BETWEEN THE L_1 AND L_∞ DUAL PROBLEMS

In this section an underlying relationship between the unweighted L_1 and L_∞ regression problems will be considered. This relationship is best illustrated by examining the dual linear programming formulations of the problems.

The dual linear programming formulation of the weighted L_1 regression problem was developed in Secion 2. The unweighted, or equal weighted, L_1 regression problem can be obtained by setting $w_k = 1/n$ for each observation. The unweighted L_1 regression problem is therefore equivalent to

$$\text{Maximize} \quad -\sum_{k=1}^{n} y_k\alpha_k + \sum_{k=1}^{n} y_k\beta_k \qquad (4.1)$$

$$\text{subject to:} \quad -\sum_{k=1}^{n} \alpha_k + \sum_{k=1}^{n} \beta_k = 0 \qquad (4.2)$$

$$-\sum_{k=1}^{n} x_k\alpha_k + \sum_{k=1}^{n} x_k\beta_k = 0 \qquad (4.3)$$

$$\alpha_k + \beta_k = 1/n \text{ for } k = 1, 2, \ldots, n \qquad (4.4)$$

$$\alpha_k \geq 0,\ \beta_k \geq 0 \text{ for } k = 1, 2, \ldots, n \qquad (4.5)$$

In Section 3 the dual linear programming formulation of the unweighted L_∞ regression problem was shown to be:

$$\text{Maximize} \quad -\sum_{k=1}^{n} y_k \alpha_k + \sum_{k=1}^{n} y_k \beta_k \tag{4.6}$$

subject to:

$$-\sum_{k=1}^{n} \alpha_k + \sum_{k=1}^{n} \beta_k = 0 \tag{4.7}$$

$$-\sum_{k=1}^{n} x_k \alpha_k + \sum_{k=1}^{n} x_k \beta_k = 0 \tag{4.8}$$

$$\sum_{k=1}^{n} \alpha_k + \sum_{k=1}^{n} \beta_k = 1 \tag{4.9}$$

$$\alpha_k \geq 0, \ \beta_k \geq 0 \text{ for } k = 1, 2, \ldots, n \tag{4.10}$$

Clearly, the dual L_∞ problem, (4.6)-(4.10), is a relaxation of the dual L_1 problem, (4.1)-(4.5). This relaxation is obtained by simply summing the n constraints of form (4.4). This yields a constraint of form (4.9). Since the dual L_∞ problem is a relaxation of the dual L_1 problem, this implies that the dual L_∞ problem should be easier to solve than the corresponding dual L_1 problem. The computational results presented in Section 5 strongly support this hypothesis since the network-based L_∞ code runs one and a half to three times faster than the network-based L_1 computer code.

5. COMPUTATIONAL TESTING

The performance statistics for a network-based weighted L_1 regression computer code and a network-based unweighted L_∞ regression computer code are presented in this section. In addition.

for purposes of comparison, the solution times for the state-of-the-art, non-network, unweighted L_1 regression computer code of Armstrong and Kung [5] are also reported.

All test problems were generated by adding an error term to a fixed linear relationship between the independent and dependent variables. The observations of the independent variables as well as the error terms were randomly sampled from pre-specified uniform distributions. Computational testing, that is not presented here, indicates that none of the solution algorithms is adversely affected by either the range of the independent variable, the magnitude of the error terms, or the fixed relationship between the independent and dependent variables.

The non-network code of [5] uses a specialized dual simplex algorithm to solve a dual linear programming formulation of the unweighted L_1 regression problem. The efficiency of this approach is credited to the fact that it uses a technique referred to as *multiple pivoting*. Armstrong and Kung have demonstrated that their code is from ten to one hundred times faster than the code of Sadovski [12], which is based on the algorithm of Edgeworth [10]. Furthermore, it has been demonstrated by Sposito [13] that the Sadovski code may fail to converge.

A discussion of the development of the weighted L_1 regression code will be presented next. This is following by a computational comparison of this network-based code to the state-of-the-art code of Armstrong and Kung. Then a discussion is given on the impact of the magnitude and range of the weights on the weighted L_1 regression code. After presenting the L_1 code, the development and computational testing of the L_∞ regression code will be outlined. This is followed by a comparison of the two new network codes.

Two of the most crucial determinants of the relative efficiency of network-based simplex codes are (1) the choice of the starting basis (i.e., the rule for constructing the initial basic feasible solution) and (2) the choice of the pivot strategy (i.e., the rule for selecting entering arcs). As expected, both of these aspects have major ramifications for the solution efficiency of the weighted L_1 regression code. Two start rules were developed and tested as well as two standard pivot rules. A brief discussion of each follows.

In Section 2 it was shown that an initial feasible (artificial) basis can be constructed by (1) selecting a set of arcs U to be non-basic at their upper bounds, (2) constructing a basic artificial arc that transfers the remaining units of supply from node A to node B, and (3) selecting a real arc to complete the full row rank basis. The most important step of this start procedure is the initialization of the set U. The first start rule tested simply initializes U as the empty set. The second start rule that was tested was motivated by an observation based on duality theory. This observation is that $\hat{y}_k \geq y_k$ ($\hat{y}_k \leq y_k$) if the optimal network solution has $\alpha_k = 0$ ($\alpha_k = w_k$). The heuristic start rule based on this observation is referred to as the advanced L_2 start. Quite simply, the heuristic is to use the weighted least squares regression equation to assign arcs to the set U. That is, U initially contains those arcs associated with positive least squares residuals. The solution times (in c.p.u. seconds) and the required pivots for both start rules are presented in Table 1. All testing was done on The University of Texas' CDC 6600 using the MNF FORTRAN compiler. In order to reduce the effect of

TABLE 1

START RULE

	EMPTY U		ADVANCED L_2	
NUMBER OF OBSERVATIONS	SOLUTION TIME	PIVOTS	SOLUTION TIME	PIVOTS
100	.174	102	.016	8
200	.671	203	.049	11
300	1.474	301	.148	29
400	2.604	400	.166	23
500	4.025	498	.291	32

*LONE with most eligible pivot rule
Mean times and pivots reported

measurement error, all solution times and pivots that are reported are actually averages for numerous test problems.

It is clear from Table 1 that the advanced L_2 start is far superior to the simple empty U start. It is important to note that the time required to determine the weighted least squares regression equation is included in the reported times in Table 1 for the advanced L_2 start code, but is not included for the empty U start since it is not used by the code.

The second aspect of the network-based L_1 regression code that was studied was the choice of pivot rule. Numerous rules have been suggested in the network literature for the selection of entering arcs. Two of the most straightforward rules are to pivot on (1) the most pivot eligible arc, and (2) the first pivot eligible arc found. The philosophy underlying these rules is that the most eligible rule requires a lot of effort to identify the entering arc, but is expected to produce the fewest total pivots. On the other hand, the first eligible rule requires little effort to identify an entering arc, but is expected to require more total pivots. Both pivot strategies were tested in order to determine the relative trade-off between the amount of work required and the number of pivots performed. Table 2 presents the solution times and pivots for these rules. Again, the reported statistics reflect the averages of numerous test problems.

The impact of the pivot rule is clear. The first eligible rule requires from two to four times as many pivots as the

TABLE 2

PIVOT STRATEGY RULE

	MOST PIVOT ELIGIBLE		FIRST PIVOT ELIGIBLE	
NUMBER OF OBSERVATIONS	SOLUTION TIME	PIVOTS	SOLUTION TIME	PIVOTS
50	.016	11	.016	28
100	.016	8	.016	16
150	.032	9	.015	31
200	.049	11	.022	33
250	.114	24	.053	60
300	.148	29	.063	72
350	.179	28	.065	61
400	.166	23	.088	88
450	.300	37	.102	107
500	.291	32	.108	123

*LONE with L_2 advanced start

Mean times and pivots reported

most eligible rule. However, the first eligible rule is consistently faster than the most eligible rule.

Based on the computational testing of the start rules and the pivot rules, the best network-based weighted L_1 regression code, referred to as LONE, uses the advanced L_2 start and the first eligible pivot rule. In order to properly assess the performance of this network approach, the fastest non-network L_1 code known was acquired. This code, known as SIMLP, is based on the dual simplex algorithm. Unlike LONE, SIMLP can only solve unweighted L_1 regression problems. Both codes require four n length data arrays, but a streamlined unweighted version of LONE could be developed that uses only three n length data arrays. The solution speed of such a code would also improve since the operations using the weights would be streamlined.

Table 3 presents the computational results for LONE and SIMLP on a large set of unweighted test problems. The indication is that LONE is roughly twice as fast as the state-of-the-art code SIMLP. This supports the recent findings that network simplex algorithms are computationally superior to their non-network counterparts.

Limited computational testing of LONE was carried out in order to measure the impact on its performance capabilities as the magnitude and range of the weights are varied. Table 4 reports the average solution times and pivots for a number of test problems with 300 data observations. For each of these problems, the

TABLE 3

LONE VS. SIMLP

	LONE		SIMLP	
NUMBER OF OBSERVATIONS	SOLUTION TIME	PIVOTS	SOLUTION TIME	MULTIPLE PIVOTS
50	.016	28	.010	3
100	.016	16	.023	4
150	.015	31	.038	6
200	.022	33	.049	4
250	.053	60	.076	4
300	.063	72	.095	5
350	.065	61	.129	5
400	.088	88	.139	4
450	.102	107	.180	5
500	.108	123	.213	6

*LONE with L_2 advanced start and first eligible pivot rule
Mean times and pivots reported

TABLE 4

WEIGHTED PROBLEMS

MINIMUM WEIGHT	MAXIMUM WEIGHT	SOLUTION TIME	PIVOTS
1	1	.064	76
1	2	.064	70
1	5	.059	74
1	10	.057	63

*300 observations
Mean times and pivots reported
LONE with L_2 start and first eligible pivot

weights were uniformly distributed between a specified upper and lower bound. This testing indicates that LONE is insensitive to the variation in the weights.

A brief summary of the development and computational testing of the network-based unweighted L_∞ regression code is presented next. As is the case with all network simplex codes, the most critical algorithmic decisions made during the development of the L_∞ code dealt with the choice of the initial basis and the entering arc selection rule. The most efficient implementation, referred to as LINF, constructs its initial basis from the first three observations, (x_{k_1}, y_{k_1}), (x_{k_2}, y_{k_2}), and (x_{k_3}, y_{k_3}), such that $x_{k_1} \neq x_{k_3}$. This code uses the most eligible rule to select the entering arcs. LINF only uses two n length data arrays.

The solution times and the number of pivots performed by LINF are reported for a large set of unweighted test problems in Table 5. For purposes of comparison, the table also provides the results for the network-based L_1 code on the same problems. As hypothesized in Section 4, the L_∞ problem is indeed easier to solve than the corresponding L_1 problem. This is due to the fact that the L_∞ dual linear programming problem can be interpreted as a relaxation of the L_1 dual linear programming problem.

TABLE 5

LINF VS. LONE

NUMBER OF OBSERVATIONS	LINF SOLUTION TIME	LINF PIVOTS	LONE SOLUTION TIME	LONE PIVOTS
50	.004	3	.016	28
100	.008	4	.016	16
150	.010	5	.015	31
200	.015	4	.022	33
250	.026	5	.053	60
300	.022	5	.063	72
350	.035	6	.065	61
400	.050	6	.088	88
450	.046	6	.102	107
500	.043	6	.108	123

REFERENCES

1. N. N. Abdelmalek, "An Efficient Method for the Discrete Linear L_1 Approximation Problem," *Mathematics of Computation*, 29 (1975) 844-850.

2. N. N. Abdelmalek, "On the Discrete Linear L_1 Approximation and L_1 Solutions of Overdetermined Linear Equations," *Journal of Approximation Theory*, 11 (1974) 38-53.

3. R. D. Armstrong and J. P. Godfrey, "A Comment on Two Linear Programming Algorithms for the Linear Discrete L_1 Norm Problem," Working Paper, The University of Texas at Austin.

4. R. D. Armstrong and D. S. Kung, "A Dual Method for Discrete Chebychev Curve Fitting," Working Paper 78-9, Graduate School of Business, The University of Texas at Austin, February 1978.

5. R. D. Armstrong and M. T. Kung, "Least Absolute Value Estimates for a Simple Linear Regression Problem," *Applied Statistics*, 27, 3 (1978) 363-366.

6. I. Barrodale and F. D. Roberts, "An Improved Algorithm for Discrete L_1 Linear Approximation," *SIAM Journal of Numerical Analysis*, 10 (1973) 839-848.

7. I. Barrodale and A. Young, "Algorithms for Best L_1 and L_∞ Linear Approximation on a Discrete Set," *Numerische Mathematik*, 8 (1966) 295-306.

8. A. Charnes, W. W. Cooper, and R. Ferguson, "Optimal Estimation of Executive Compensation by Linear Programming," *Management Science*, 2 (1955) 138-151.

9. N. Draper and H. Smith, *Applied Regression Analysis*, John Wiley and Sons, Inc., New York, 1966.

10. F. Y. Edgeworth, "On a New Method of Reducing Observations Relating to Several Quantities," *Phil. Mag.* (5th Ser.), 25 (1888) 184-191.

11. G. McCormick and V. Sposito, "Using the L_2-Estimation in the L_1-Estimation," *SIAM Journal of Numerical Analysis*, 13 (1976) 337-343.

12. A. Sadovski, "L_1-Norm Fit of a Straight Line," *Applied Statistics*, 23 (1974) 244-248.

13. V. Sposito, "A Remark on Algorithm AS74. L_1-Norm Fit of a Straight Line," *Applied Statistics*, 25, 96-97.

14. E. Stiefel, "Note on Jordan Elimination, Linear Programming, and Tchebyscheff Approximation," *Numerische Mathematik*, 2 (1960) 1-17.

15. E. Stiefel, "Methods--Old and New--For Solving the Tchebycheff Approximation Problem," *SIAM Journla of Numerical Analysis*, 1 (1964) 164-176.

16. H. Wagner, *Principles of Operations Research*, Prentice-Hall, Englewood Cliffs, New Jersey, 1969.

-8

DTIC