JOURNAL OF COMPUTER AND SYSTEM SCIENCES: 1, 213-226 (1967)

Generalized Upper Bounding Techniques*

G. B. DANTZIG† AND R. M. VAN SLYKE‡

Operations Research Center, University of California, Berkeley, California 94720

Received July 12, 1966

ABSTRACT

A variant of the revised simplex method is given for solving linear programs with $M + L$ equations, L of which have the property that each variable has at most one nonzero coefficient in them. Special cases include transportation problems, programs with upper bounded variables, assignment and weighted distribution problems. The algorithm described uses a working basis of M rows for pivoting, pricing, and inversion which for large L can result in a substantial reduction of computation. This working basis is only $M \times M$ and is a further reduction of the size found in an earlier version.

I. INTRODUCTION AND NOTATION

The application of linear programming to large systems inevitably leads to programs with special structure. One such structure arising frequently in distribution, production scheduling and optimal control problems is a linear program in which each variable has at most one nonzero coefficient in the last L equations which is nonnegative, and the last L constant terms are positive. This is the problem we study here. See also [*1*]. In Section IV we indicate the necessary modifications to handle negative coefficients in the last L equations. By normalizing the variables and multiplying the equations by constants, we can assume without loss of generality that all nonzero coefficients and constants in the last L equations are 1's [See Eq. (1)].

The lth *set of variables or columns*, S_l, will refer (depending on context) to those variables or columns corresponding to the columns of coefficients in (1) with 1 as their $M + l$th component. S_0, *the* 0th *set*, is the set corresponding to columns with zeros for the $M + 1$st through $M + L$th coefficients.

* This research has been partially supported by the Office of Naval Research under Contract Nonr-222(83) with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

† Present address: Stanford University (Operations Research House), Stanford California.

‡ Also with the Department of Electrical Engineering, University of California, Berkeley.

571/1/3-1

Max x_0 subject to

$$
\begin{array}{l}
A_1^0 x_0 + A_1^1 x_1 + \cdots + A_1^{n_0} x_{n_0} + A_1^{n_0+1} x_{n_0+1} + \cdots + A_1^{n_1} x_{n_1} + A_1^{n_1+1} x_{n_1+1} + \cdots + A_1^{n_2} x_{n_2} + \cdots + A_1^{n_{L-1}+1} x_{n_{L-1}+1} + \cdots + A_1^N x_N = b_1 \\
A_2^0 x_0 + A_2^1 x_1 + \cdots + A_2^{n_0} x_{n_0} + A_2^{n_0+1} x_{n_0+1} + \cdots + A_2^{n_1} x_{n_1} + A_2^{n_1+1} x_{n_1+1} + \cdots + A_2^{n_2} x_{n_2} + \cdots + A_2^{n_{L-1}+1} x_{n_{L-1}+1} + \cdots + A_2^N x_N = b_2 \\
\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
A_M^0 x_0 + A_M^1 x_1 + \cdots + A_M^{n_0} x_{n_0} + A_M^{n_0+1} x_{n_0+1} + \cdots + A_M^{n_1} x_{n_1} + A_M^{n_1+1} x_{n_1+1} + \cdots + A_M^{n_2} x_{n_2} + \cdots + A_M^{n_{L-1}+1} x_{n_{L-1}+1} + \cdots + A_M^N x_N = b_M \\
\qquad\qquad x_{n_0+1} + \cdots + x_{n_1} = 1 \\
\qquad\qquad\qquad\qquad x_{n_1+1} + \cdots + x_{n_2} = 1 \\
\qquad\qquad\qquad\qquad\qquad\qquad \ddots \qquad\qquad \vdots \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad x_{n_{L-1}+1} + \cdots + x_N = 1.
\end{array}
\tag{1}
$$

We assume that the system (1) is of full rank and denote by $[\mathbf{A}^{j_1},\ldots, \mathbf{A}^{j_{M+L}}]$, a basis for the system. We always assume $\mathbf{A}^{j_1} = \mathbf{A}^0$, the coefficient of the variable to be optimized. Bold-face type is used to differentiate coefficient vectors with all $M + L$ components from the reduced vectors of the first M coefficients, which are in light-face type. Individual components A_i^j of these two different types of vectors will not be bold-faced since they differ only in the *number* of their components.

THEOREM 1. *At least one variable from each set S_l is basic, $l = 0,\ldots, L$.*

Proof. Since our system is assumed to be full rank, if $[\mathbf{A}^{j_1},\ldots, \mathbf{A}^{j_{M+L}}]$ is a basis any $M + L$ vector $\mathbf{b}$ can be expressed as a linear combination of the columns of basis. In particular, if the $M + l$th, $l = 1,\ldots, L$, component of $\mathbf{b}$ is nonzero at least one of the basic columns must have a nonzero element in this component and, therefore, belong to $S_l \cdot A^0 = A^{j_1} \in S_0$.

THEOREM 2. *The number of sets containing two or more basic variables is at most $M - 1$.*

Proof. Of the $M + L$ basic variables, $L + 1$ of them are in different sets by Theorem 1. This leaves at most $M - 1$ basic variables to compose sets with more than one basic variable.

The sets containing two or more basic variables plus the set S_0 are called *essential* sets. An essential set for one basis may become an unessential one in the next.

In the next section we outline the method; in the following we formalize it as an algorithm. In the last section we indicate some extensions and, finally, in the Appendix the method is carried out on an example.

II. THE METHOD

Given a feasible basis[1] $\{\mathbf{A}^{j_1},\ldots, \mathbf{A}^{j_{M+L}}\}$, we assume we have selected for each S_l, $l = 1,\ldots, L$ one basic variable x_{k_l} to be the *key variable*. A^{k_l} is said to be the *key column*.

[1] Obtaining a first feasible solution is accomplished using this method with a Phase I setup as in the usual simplex method.

S_0 has no key column. We then consider the system obtained by subtracting the key columns from every other column in their respective sets [in (2) we assume for simplicity that the key variable was the first one in each set]. In this modified system

$$\begin{array}{l}
A^0 y_0 + \cdots + A^{n_0} y_{n_0} + A^{n_0+1} y_{n_0+1} + (A^{n_0+2} - A^{n_0+1}) y_{n_0+2} + \cdots + (A^{n_1} - A^{n_0+1}) y_{n_1} + \cdots + A^{n_{L-1}+1} y_{n_{L-1}+1} + \cdots + (A^{n_L} - A^{n_{L-1}+1}) y_{n_L} = b \\
\qquad\qquad 1 \cdot y_{n_0+1} \quad 0 \; \text{-----------} \; 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = 1 \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \ddots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad 1 \cdot y_{n_{L-1}+1} \; \text{--------} \; 0 \qquad\qquad = 1,
\end{array} \tag{2}$$

where

$$y_j = x_j , \qquad j = 0,\ldots, n_0 ,$$

$$y_{n_i+1} = \sum_{j=n_i+2}^{n_{i+1}} x_j , \qquad i = 1,\ldots, L-1,$$

$$y_{n_i+j} = x_{n_i+j} , \qquad i = 1,\ldots, L-1;\ j = 2,\ldots, n_{i+1} .$$

the value of the key variables corresponding to any feasible solution must clearly be one. We treat these variables as we would variables at upper bound in an upper bounded variables algorithm for the revised simplex method and subtract their coefficients from the right-hand side. We then introduce the following notation: if $A^j \in S_l$ we let

$$\begin{aligned} D^{k_l} &= A^{k_l}, \\ D^j &= A^j - A^{k_l}, \qquad j \neq k_l , \\ d &= b - \sum_{i=1}^{L} D^{k_l} = b - \sum A^{k_l}. \end{aligned} \tag{3}$$

We then can consider D^j for $j = k_l$ (key) to be absent from the system. The *working basis*, B, is given by $B = \{D^j | A^j \text{ is basic and not key}\}$. Since there are exactly L key columns it is clear that B has M columns. We assume $B^1 = A^0$ corresponds to the coefficient of the variable to be optimized. We define the *derived system* to be

$$\sum y_j D^j = d, \tag{4}$$

and it is easy to prove.

THEOREM 3. *B is a basis for* (4).

Proof. Suppose $\sum \lambda_j B^j = 0$. Since B^j differs from $\mathbf{B}^j$ [the same column considered in the system depicted in (2)] by only 0 components $\sum \lambda_j \mathbf{B}^j = 0$. But this implies that the $\mathbf{B}^j$ plus the key columns are linearly dependent since the $\mathbf{B}^j$ by themselves are linearly dependent. On the other hand, this set is obtained from a (nonsingular) basis by subtraction of columns from within the set which does not reduce the rank, yielding a contradiction.

By Theorem 2 there exist at most $M - 1$ sets with more than one basic variable. These sets and S_0 are the only sets which contain members of B and will be referred to as the *essential* sets.

Thus, with each feasible basis for the original system (1), we have associated a set of L key variables and a basis for the derived system. We now show that we can carry out the steps of the simplex method using just the inverse B^{-1} of B, the reduced basis, and the corresponding basic solution of the derived system (4).

The first step is to obtain a set of prices for (1). Let us denote by $\pi = (\pi_1, \ldots, \pi_M)$ the prices on the first M equations and $\mu = (\mu_1, \ldots, \mu_L)$ the prices on the last L. These prices are determined uniquely by the condition that

$$(\pi, \mu)\mathbf{A}^0 = (\pi, \mu)\mathbf{A}^{j_1} = 1,$$

$$(\pi, \mu)\mathbf{A}^{j_i} = 0, \qquad i = 2, \ldots, M + L.$$

Let $\hat{\pi} = (B^{-1})_1$, the first row of the inverse of the working basis B. It has the property that

$$\hat{\pi}B^1 = \hat{\pi}A^0 = 1,$$

$$\hat{\pi}B^j = 0, \qquad j = 2, \ldots, M;$$

i.e., $\hat{\pi}$ is a set of prices for (4). To extend $\hat{\pi}$ to a set of prices for (2) is trivial we simply set

$$\hat{\mu}_l = - \hat{\pi}A^{k_l}, \qquad l = 1, \ldots, L. \tag{5}$$

Now for basic columns A^{j_i},

$$(\hat{\pi}, \hat{\mu})\mathbf{A}^{j_i} = (\hat{\pi}, \hat{\mu})\mathbf{A}^{k_l} = 0 \quad \text{if} \quad A^{j_i} \quad \text{is key}$$

or if A^{j_i} is not key,

$$\begin{aligned}(\hat{\pi}, \hat{\mu})\mathbf{A}^{j_i} &= (\hat{\pi}, \hat{\mu})(\mathbf{B}^i + \mathbf{A}^k) \quad \text{for some } k \\ &= (\hat{\pi}, \hat{\mu})\mathbf{B}^i + (\hat{\pi}, \hat{\mu})\mathbf{A}^k \\ &= 0 + 0.\end{aligned}$$

Thus $(\hat{\pi}, \hat{\mu})$ is a set of prices for the original system (1).

Using these prices we can "price out" the columns of (1) to find the next column to enter the basis. Using the usual simplex criterion, the incoming column A^s would be chosen by

$$\Delta_s = (\pi, \mu)\,\mathbf{A}^s = \min_j(\pi, \mu)\,\mathbf{A}^j = \min_j \Delta_j$$

where

$$\Delta_j = \sum \pi_i A_i^j + \mu_l \quad \text{for} \quad A^j \in S_l .$$

Suppose $A^s \in S_\sigma$. If $\Delta_s \geqslant 0$, we have an optimal basic feasible solution and we're done; otherwise, we bring $\mathbf{A}^s$ into the basis. To do this, we must express $\mathbf{A}^s$ and $\mathbf{b}$ in terms of the current basis *for* (1). If we let

$$\bar{D}^s = B^{-1}D^s = B^{-1}(A^s - A^{k_\sigma}),$$

then

$$(A^s - A^{k_\sigma}) = \sum_{i=1}^{M} \bar{D}_i^s B^i = \sum \bar{D}_i^s (A^{\eta_i} - A^{\nu_i}), \tag{6}$$

where η_i indicates the column number in (2) corresponding to the ith column of the working basis and ν_i denotes the column number of the corresponding key variable.

We denote the representation of $\mathbf{A}^s$ in terms of the current basis by $\bar{A}_i^s$; that is,

$$\mathbf{A}^s = \sum_{i=1}^{M+L} \bar{A}_i^s \mathbf{A}^{j_i}.$$

From (6) we see

$$\bar{A}_i^s = \begin{cases} 1 - \sum\limits_{\nu_t = k_\sigma} \bar{D}_t^s & \text{if} \quad A^{j_i} = A^{k_\sigma}, \\ \bar{D}_t^s & \text{if} \quad A^{j_i} = A^{\eta_t} \quad \text{for some } t, \\ -\sum\limits_{\nu_t = j_i} \bar{D}_t^s & \text{if} \quad A^{j_i} = A^{\nu_t} \quad \text{for some } t, \end{cases} \tag{7}$$

The current values for the variables in the basis $\bar{b}_i$ are given either by updating the values of the previous iteration in the usual way or recomputed in a similar way to $\bar{A}_j^s$ above. That is,
Let

$$\bar{d} = (\bar{d}_1, ..., \bar{d}_M) \quad \text{be given by}$$

$$\bar{d} = B^{-1}\left(b - \sum A^{k_l}\right) = B^{-1}d; \tag{8}$$

then

$$\left(b - \sum A^{k_l}\right) = \sum \bar{d}_i B^i = \sum \bar{d}_i (A^{\eta_i} - A^{\nu_i})$$

and, as in (7), the $\bar{b}_i$ are given by

$$\bar{b}_i = \begin{cases} 1 - \sum_{\nu_t = j_i} \bar{d}_t & \text{if} \quad A^{j_i} \quad \text{is key} \\ \bar{d}_t & \text{if} \quad A^{j_i} = A^{\eta_t} \quad \text{for some } t \end{cases} \tag{9}$$

Finding the variable to leave the basis is accomplished in exactly the same way as in the ordinary simplex method. Let

$$\theta \triangleq \frac{\bar{b}_r}{\bar{A}_r{}^s} \triangleq \min_{\bar{A}_i{}^s > 0} \frac{\bar{b}_i}{\bar{A}_i{}^s}, \qquad i = 1, \ldots, M + L, \tag{10}$$

where we, require that $\bar{A}_r{}^s > 0$. Let us assume $A^{j_r} \in S_\rho$. Three cases can occur in the updating process:

(a) If S_σ is not essential and $A^{j_r} \in S_\sigma$; i.e., the outgoing variable is the key variable in S_σ then B remains unchanged, and A^s simply replaces A^{j_r} as the key variable in S_σ. This requires the updating of $\bar{d}$ which is accomplished as follows[2]:

$$\begin{aligned} \bar{d} :&= B^{-1}\Big(b - \sum_{l \neq \sigma} A^{k_l} - A^{k_\sigma} + A^{j_r} - A^s\Big) \\ &= B^{-1}\Big(b - \sum_{l \neq \sigma} A^{k_l} - A^{k_\sigma}\Big) - B^{-1}(A^s - A^{j_r}) \\ &= \bar{d} - B^{-1}(A^s - A^{j_r}) \\ &= \bar{d} - \bar{D}^s. \end{aligned} \tag{11}$$

Observing that $A^{j_r} = A^{k_\sigma}$, we see that this is easy to compute since we already have $\bar{d}$ and the second term was generated in determining the $\bar{A}_i{}^s$.

(b) If A^{j_r} is not a key variable, then we update B^{-1} simply by pivoting on the column $\bar{D}^s$ on the row which $A^{j_r} - A^{k_\rho}$ occupies in the working basis. In symbols $B^{-1} := PB^{-1}$ where P is the matrix which performs the pivot. $\bar{d}$ is updated by applying this pivot to the old $\bar{d}$.

(c) If $A^{j_r} \in S_\rho$ is a key variable in an essential set, we must first change the key variable in S_ρ. To change the inverse of the working basis B, we consider all columns of B of the form $A^j - A^{j_r}$ there must be at least one such since after A^s enters the basis S_ρ must contain a basic variable (Theorem 1). One of these columns, call it A^k, is to become the key variable. To get the new working basis $\bar{B}$ from the old one B we wish to multiply the column (in B) $A^k - A^{j_r}$ by -1 to obtain $A^{j_r} - A^k$ and we wish to subtract $A^k - A^{j_r}$ from every other column of the form $A^j - A^{j_r}$ for $j \neq k$ to obtain $A^j - A^k$. That is (see footnote 2),

[2] The symbol " $:=$ " does not indicate equality but rather that the expression on the right replaces or (updates) the variable on the left.

$B := BT$ where T is of the form

$$T = \begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ 0 \cdots 0 & -1 \cdots & -1 & -1 & -1 \cdots & -1 & 0 \cdots 0 \\ & & & & 1 & & & \\ & & & & & \ddots & & \\ & & & & & & 1 \end{bmatrix}. \tag{12}$$

The -1's occur in the columns corresponding to $A^j \in S_\rho$, and the row corresponds to the new key variable A^k.

$$B^{-1} := T^{-1}B^{-1}$$

and it is easily verified that $T^{-1} = T$, since applying the process twice replaces A^{j_r} as the key variable. The values for $\bar{d}$ are updated by applying T^{-1}. Now with the new key variable in S_ρ, we simply apply the process outlined in (b).

With our updated B, y, and key variables, we are now ready to make another iteration. If the inverse of the working-basis is expressed in product form we have

$$B^{-1} = \prod^{t} T^t,$$

where each T^t is either of the form (12) or (13) below, the latter resulting from a pivot on the rth element of the column $(\alpha_1, ..., \alpha_m)^T$,

$$P = \begin{bmatrix} 1 & & & -\alpha_1/\alpha_r & & & \\ & \ddots & & \vdots & & & \\ & & 1 & & & & \\ & & & -\alpha_{r-1}/\alpha_r & & & \\ & & & 1/\alpha_r & & & \\ & & & & 1 & & \\ & & & \vdots & & \ddots & \\ & & & -\alpha_m/\alpha_r & & & 1 \end{bmatrix}. \tag{13}$$

As Orchard-Hays has remarked to one of the authors, we can, if we wish, express each transformation of the form (12) as a sequence of transformation of the form (13). Suppose we wish to express a matrix T in terms of transformations of the form (13), where -1's appear in columns h_0, h_1,..., h_i, and suppose that the -1 in column h_0

lies on the diagonal; in other words, all the -1's are the h_0th row. Let $P(r, s)$ denote the pivot matrix (13) with $\alpha_j = 0\, j \neq r$, or s, $\alpha_r = 1$ and $\alpha_s = -1$. When multiplying on the left, this matrix has the effect of subtracting the rth row from the sth row. Finally, let $P(r, r)$ be the matrix with all plus ones on the diagonal except in the rth diagonal element which is -1. Every other element is zero. When multiplying on the *right*, this has the effect of multiplying the rth column by -1. It is then easy to see that

$$T = P(h_0, h_k)\, P(h_0, h_{k-1}) \cdots P(h_0, h_1)\, P(h_0, h_0).$$

III. Description of Algorithm

Referring to Fig. 1, the algorithm takes place in the following steps:

(1) We assume we enter the algorithm with the inverse B^{-1} of the working basis, the value $\bar{d}$ of the appropriate basic solution of the derived system (4) and the set of key variables. To get this initial solution, the usual phase 1 procedure can be carried out in the obvious way.

(2) Let $\pi_i = (B^{-1})_1{}^i$ for $i = 1,..., M$ and for each set $S_l (l \neq 0)$.

Let $\mu_l = -\sum_{i=1}^{M} \pi_i A_i^{k_l}$, where A^{k_l} is the key column in S_l.

Let $\Delta_j = \sum_{i=1}^{M} \pi_i A_i{}^j + \mu_l$ for $A^j \in S_l$.

Let $\Delta_s = \min \Delta_j$ and suppose $A^s \in S_\sigma$.

If $\Delta_s \geqslant 0$, we go to Step (3); otherwise, skip to (4).

(3) Terminate; we have an optimal solution.

(4) Find $\bar{D}^s = B^{-1} D^s = B^{-1}(A^s - A^{k_\sigma})$, $\bar{\mathbf{A}}^s$ [by means of Eq. (7)], and $\bar{\mathbf{b}}$. [by means of Eq. (9)]. Use the usual simplex decision rule [Eq. (10)] to find the variable to be dropped, A^{j_r}, and suppose $A^{j_r} \in S_\rho$. If A^{j_r} is key, go to step (5); if not S_ρ is essential, go to (6); if $\rho = \sigma$, go to step (7).

(5) We pivot with respect to $\bar{D}^s$ in the row corresponding to D^{j_r} in B^{-1} and update $\bar{d}$ by applying the pivot transform to it. We then return to step (1) for another iteration.

(6) Make some basic column, say A^k, $k \neq j_r$ in set S_ρ key instead of A^{j_r}. Update B^{-1} by applying a column transformation of the form (12) and update B^{-1} by $B^{-1} := T^{-1} B^{-1}$. $\bar{d}$ is updated by $\bar{d} := T^{-1}\bar{d}$. We then can go to step (5).

(7) Make A^s key instead of A^{j_r} and update $\bar{d}$ by $\bar{d} := \bar{d} - \bar{D}^s$. Return to step (1).

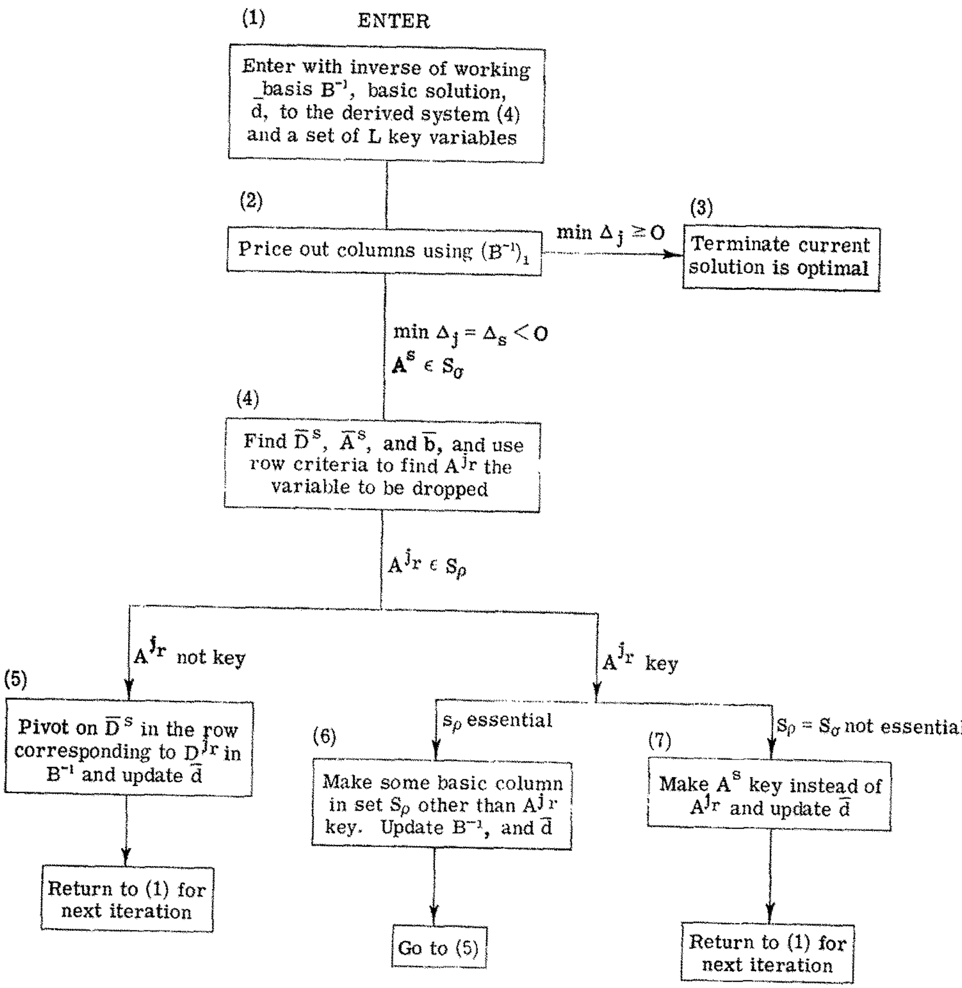

FIG. 1. Flow diagram of the generalized upper-bound algorithm.

IV. EXTENSIONS AND COMPUTATIONAL REMARKS

When negative coefficients appear in the last L equations, the algorithm is changed in a quite obvious way. We can assume without loss of generality that the coefficients in the last L equations are $+1$ or -1 and the last L right-hand side components are $+1$. Theorems 1, 2, and 3 still hold, and we can require that each key column have a $+1$ in the last L equations since clearly each set must have such a column which is basic. In the pricing process, if the column A^j to be priced has a negative coefficient in the last L components, the appropriate μ is subtracted rather than added to πA^j. To form the difference columns D^j, the key column, is added to columns with a -1 rather than subtracted and appropriate modifications in equations (7) and (9) must be made to reflect this. Other than these slight modifications, the algorithm proceeds exactly as before.

The algorithm has been implemented by James Bigelow in collaboration with Mike Kasatkin of the Crown Zellarbach Corporation on the IBM 7094 computer by modifying the M–3 linear programming system. The implementation was particularly easy because it required only a slight modification of features usually present in current large-scale linear programming codes. In particular, the separable-programming logic was modified to handle the partition of the variables into sets and the system used to indicate the state of a variable was a slight generalization of that used in the upper-bounded variable algorithm. This code will handle problems with $M \leqslant 100$, $L \leqslant 5{,}000$ and $N \leqslant 30{,}000$. A test problem with $N = 2813$, $L = 780$, and $M = 39$ was solved in 15 minutes. The iterations were carried out at a rate of 50 per minute. Solution time is to be contrasted with an estimated running time of 150 minutes using an efficient general linear programming code a decrease in running time by a factor of 10.

A further generalization of this method to the case of block staircase structure in the last L equations has been carried out by Kaul [*2*]. Similar approaches have also been investigated by Bennett [*3*], Rosen [*4*], and Charnes and Lemke [*5*].[3]

Example: Consider the following example with $M = 3$. We seek to maximize x_0.

	S_0	S_1			S_2	S_3	S_4		S_5		
	A^0	**A**^{1}	**A**^{2}	**A**^{3}	**A**^{4}	**A**^{5}	**A**^{6}	**A**^{7}	**A**^{8}	**A**^{9}	**b**
	1	0	2	0	3	4	5	1	-1	-12	15
	1	1	-1	0	2	1	4	2	-3	6	7
	0	0	0	1	0	0	0	0	0	0	0
		1	1	1							1
					1						1
						1					1
							1	1			1
									1	1	1
$\bar{b}^T =$	(3	$\frac{1}{2}$	$\frac{1}{2}$	0	1	1	1		1	)	
		x			x	x	x		x		

[3] We are indebted to the referee for this last reference.

The initial basis is $A^0A^1A^2A^3A^4A^5A^6A^8$ and $A^1A^4A^5A^6A^8$ are key. The working basis is $B = \{A^0, A^2 - A^1, A^3 - A^1\}$ is

$$B = \begin{bmatrix} 1 & 2 & 0 \\ 1 & -2 & -1 \\ 0 & 0 & 1 \end{bmatrix},$$

which has an invese

$$B^{-1} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ 0 & 0 & 1 \end{bmatrix}.$$

With the aid of (5) we find the prices $[\pi, \mu] = [\frac{1}{2}, \frac{1}{2}, \frac{1}{2}; -\frac{1}{2}, -\frac{5}{2}, -\frac{5}{2}, -\frac{9}{2}, 2]$.

We then price out and find $A^7 \in S_4$ wins.

$$\bar{A}^7 - \bar{A}^6 = B^{-1}[A^7 - A^6] = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -4 \\ -2 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} -3 \\ -\frac{1}{2} \\ 0 \end{bmatrix};$$

i.e., $A^7 - A^6 = -3A^0 - \frac{1}{2}(A^2 - A^1)$ or $\mathbf{A}^7 = -3\mathbf{A}^0 + \frac{1}{2}\mathbf{A}^1 - \frac{1}{2}\mathbf{A}^2 + \mathbf{A}^6$. giving a representation of $\mathbf{A}^7$ in terms of the full basis;

$$\bar{\mathbf{A}}^7 = (-3, \tfrac{1}{2}, -\tfrac{1}{2}, 0, 0, 0, 1, 0)^T.$$

We obtain the values of the variables by considering

$$B^{-1}\left[b - \sum A^{k_\ell}\right] = B^{-1}\begin{bmatrix} 4 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ \frac{1}{2} \\ 0 \end{bmatrix}.$$

This means

$$\mathbf{b} - \sum \mathbf{A}^{k_\ell} = 3\mathbf{A}^0 - \tfrac{1}{2}\mathbf{A}^1 + \tfrac{1}{2}\mathbf{A}^2$$

or

$$\mathbf{b} = 3\mathbf{A}^0 + \tfrac{1}{2}\mathbf{A}^1 + \tfrac{1}{2}\mathbf{A}^2 + \mathbf{A}^4 + \mathbf{A}^5 + \mathbf{A}^6 + \mathbf{A}^8;$$

hence

$$\mathbf{b} = [3, \tfrac{1}{2}, \tfrac{1}{2}, 0, 1, 1, 1, 1]^T.$$

We now determine the variable going out of the basis by

$$\theta = \min_{\bar{A}_i^7 > 0} \frac{\bar{b}_i}{\bar{A}_i^7} \triangleq \frac{\bar{b}_r}{\bar{A}_r^7} = 1$$

and r could be 2 or 7. Taking it to be 7, we see that since the set S_4 is inessential and $\theta = 1$ we just replace A^6 by A^7 as a key variable and B remains unchanged. The new multipliers are

$$[\bar{\pi}, \mu] = [\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{1}{2}; -\tfrac{1}{2}, -\tfrac{5}{2}, -\tfrac{5}{2}, -\tfrac{3}{2}, 2]$$

and this time $\mathbf{A}^9 \in S_5$ prices out optimally.

$$B^{-1}[A^9 - A^8] = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -11 \\ 9 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -5 \\ 0 \end{bmatrix},$$

that is, $\mathbf{A}^9 = -\mathbf{A}^0 + 5\mathbf{A}^1 - 5\mathbf{A}^2 + \mathbf{A}^8$ and $\bar{\mathbf{A}}^9 = [-1, 5, -5, 0, 0, 0, 0, 1]$.

$$B^{-1}\left[\mathbf{b} - \sum \mathbf{A}^{k_\ell}\right] = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{4} & -\frac{1}{4} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 8 \\ 4 \\ 0 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \\ 0 \end{bmatrix},$$

$$\begin{aligned} b &= 6A^0 + A^2 - A^1 + A^1 + A^4 + A^5 + A^7 + A^8 \\ &= 6A^0 + A^2 + A^4 + A^5 + A^7 + A^8, \\ \mathbf{b} &= [6, 0, 1, 0, 1, 1, 1, 1], \\ \theta &= \min_{\bar{\bar{A}}_i^9 > 0} \frac{\bar{\bar{b}}_i}{\bar{\bar{A}}_i^9} = \frac{\bar{\bar{b}}_2}{\bar{\bar{A}}_2^9} = \frac{0}{5}; \quad r = 2, j_r = 1 \end{aligned}$$

therefore, we want to drop A^1, which, however, is key. So first we must replace A^1 by A^2 as a key variable. To do this we take our current working basis

$$B = [A^0 - 0, A^2 - A^1, A^3 - A^1]$$

and postmultiply it by

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & 0 & 1 \end{bmatrix},$$

which has the effect of subtracting the second column from the third and reversing the sign of the second column.

$$B' = BT = [A^0 - 0,\ A^1 - A^2,\ A^3 - A^2],$$
$$(B')^{-1} = T^{-1}B^{-1},$$

where

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Hence

$$(B')^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{4} & \frac{1}{4} & -\frac{3}{4} \\ 0 & 0 & 1 \end{bmatrix}$$

We then pivot in the vector column

$$(B')^{-1}(A^9 - A^8) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{4} & \frac{1}{4} & -\frac{3}{4} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -11 \\ 9 \\ 0 \end{bmatrix}$$
$$= \begin{bmatrix} -1 \\ 5 \\ 0 \end{bmatrix}$$

on the second component.

This gives us a new inverse basis

$$B^{-1} = PB'^{-1}$$
$$= \begin{bmatrix} 1 & \frac{1}{5} & 0 \\ 0 & \frac{1}{5} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{4} & \frac{1}{4} & -\frac{3}{4} \\ 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} \frac{9}{20} & \frac{11}{20} & \frac{7}{20} \\ \frac{1}{20} & \frac{1}{20} & -\frac{3}{20} \\ 0 & 0 & 1 \end{bmatrix}.$$

The new prices are

$$\left(\frac{9}{20}, \frac{11}{20}, \frac{7}{20}; -\frac{7}{20}, -\frac{49}{20}, -\frac{47}{20}, -\frac{31}{20}, \frac{42}{20}\right),$$

and upon pricing out we find that all columns price out nonnegatively and the optimal solution is given by

$$\bar{B}^{-1}\left[b - \sum A^{k_\ell}\right] = \begin{bmatrix} \frac{9}{20} & \frac{11}{20} & \frac{7}{20} \\ \frac{1}{20} & \frac{1}{20} & -\frac{3}{20} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 6 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix}$$

and

$$b - A^2 - A^4 - A^5 - A^7 - A^8 = 6A^0$$

or

$$\bar{\bar{\mathbf{b}}} = [6, 1, 0, 1, 1, 1, 1, 0],$$

the values of the basic variables. Another way to compute $\bar{\bar{b}}$ is, of course, by

$$\bar{\bar{b}} = \bar{b} - \theta \bar{\bar{A}}^9 ,$$

the usual formula for updating the values of the basic variables in the simplex method.

References

1. G. B. Dantzig and R. M. Van Slyke, A generalized upper-bounded technique for linear programming, *in* "Proceedings of the IBM Scientific Computing Symposium on Combinatorial Problems." IBM, White Plains, New York, 1966.
2. R. N. Kaul. "An Extension of Generalized Upper Bounded Techniques for Linear Programming." [(ORC 65-27), Operations Research Center, University of California, Berkeley, 1965]
3. Bennett. "An Approach to Some Structured Linear Programming Problems", Bassar Computing Department, School of Physics, University of Sydney, Australia, 1963.
4. J. B. Rosen. Primal partition programming for block diagonal matrices. *Numerische Mathematik* **6**, 250-260 (1964).
5. A. Charnes and C. Lemke. "Multi-copy Generalized Networks and Multi-page Programs" (R.P.I. Math Rep. No. 41, Rensselaer Polytechnic Institute; Troy, New York, 1960.)