DISCUSSION PAPER NO. 68

THE SOLUTION OF LEONTIEF SUBSTITUTION SYSTEMS USING MATRIX ITERATIVE TECHNIQUES

Gary J. Koehler,* Andrew B. Whinston**
and Gordon P. Wright**

January 20, 1974

* Northwestern University
Evanston, Illinois 60201

** Purdue University
West Lafayette, Indiana 47907

It is well known that most large linear programs are solved by the Revised Simplex with a compact inversion method. Although considerable research has been devoted to methods for storing the basis inverse with minimum storage requirements and to finding reinversion procedures that produce sparse but accurate and easy to use inverses,the major demand on storage, computational effort, and arithmetic rounding errors has been the maintenance of the basis inverse.

Now, it is also well known that systems of equations may be solved by iterative or indirect methods [13]. In such methods, an inverse is never explicitly calculated. Furthermore, by the appropriate choice of iterative methods, structural data may be used directly in computations. Hence, the only storage required beyond structural data is the solution vector. Other advantages of iterative methods are: (1) the prior knowledge of a close solution vector can be used to accelerate convergence speed; (2) rounding and arithmetic errors are readily controlled; (3) bounds on the solution vector can be developed and used to detect solution termination and, in some cases, to speed up convergence.

In this paper, a general framework is developed for examining a class of linear programs that can be solved by iterative methods. The advantages of solving systems of equations by iterative methods carry over to solving linear programs. In addition, several new useful properties arise such as: (1) a linear program can be reduced in size as the computational procedure progresses and (2) certain iterative methods which converge asymptotically faster than other iterative methods when used to solve systems of equations can be shown to converge uniformly faster when used to solve linear programs.

I. Background

A few well known results are summarized for reference throughout the following. From Fiedler and Pták [5], we have:

Definition 1 (Class Z)

The set of all real square matrices having non-positive off-diagonal elements forms the set referred to as Class Z.

Definition 2 (Class K)

The subset of Z satisfying any property of the following theorem comprises Class K.

Theorem 1

Let $A \in Z$. Then the following are equivalent to each other:

(i) There exists a vector $x \geqq 0$ such that $Ax > 0$;

(ii) there exists a vector $x > 0$ such that $Ax > 0$;

(iii) there exists a diagonal matrix D with positive diagonal elements such that $ADe > 0$ where e is a vector of ones;

(iv) there exists a diagonal matrix D with positive diagonal elements such that the matrix $W = AD$ is a matrix with a dominant positive principal diagonal;

(v) for each diagonal matrix R such that $R \geqq A$, the inverse of R exists and $p\left(R^{-1}(P-A)\right) < 1$ where $p(\cdot)$ is the spectral radius and P is the diagonal of A;

(vi) if $B \in Z$ and $B \geqq A$, then B is non-singular;

(vii) each real eigenvalue of A is positive;

(viii) all principal minors of A are positive;

(ix) there exists a permutation matrix P such that PAP^{-1} may be written in the form RS where R(S) is lower (upper) triangular with positive diagonal elements and $R(S) \in Z$;

(x) A is nonsingular and $A^{-1} \geqq 0$;

(xi) the real portion of every eigenvalue of A is positive;

(xii) for each vector $x \neq 0$ there exists an index K such that $x_K y_K > 0$ for $y = Ax$.

Corollary 1

Let $A \in K$, $B \in Z$, and $B \geq A$. Then $B \in K$ and B satisfies:

(i) $A^{-1} \geqq B^{-1} \geqq 0$

(ii) $\det(B) \geq \det(A) > 0$

(iii) $A^{-1}B \geqq I$ and $BA^{-1} \geqq I$

(iv) $B^{-1}A \in K$ and $AB^{-1} \in K$

(v) $B^{-1}A \leqq I$ and $AB^{-1} \leqq I$

(vi) $\rho(I-B^{-1}A) < 1$ and $\rho(I-AB^{-1}) < 1$

From the above, the following results are immediately evident (the Perron-Frobenius Theorem for non-negative matrices is required in places [13]).

Corollary 2

For $A \in K$ and $A = R_1 - S_1 = R_2 - S_2$ with $S_1 \geq S_2 \geqq 0$, then

(i) $R_1 \in K$ and $R_2 \in K$

(ii) $R_2^{-1}R_1 \geq I$ and $R_1R_2^{-1} \geqq I$

(iii) $R_1^{-1} R_2 \in K$ and $R_2R_1^{-1} \in K$

(iv) $1 > \rho(R_2^{-1} S_2) \geqq \rho(R_1^{-1}S_1) \geqq 0$

(v) $R_2^{-1}S_1 \geqq R_1^{-1}S_1 \geq R_1^{-1}S_2 \geq 0$

(vi) $R_2^{-1}S_1 \geqq R_2^{-1}S_2 \geq R_1^{-1}S_2 \geqq 0$

(vii) $A^{-1} = (I-R_1^{-1}S_1)^{-1}R_1^{-1} = (I-R_2^{-1}S_2)^{-1} R_2^{-1}$

(viii) Ax=b if and only if x is a fixed point of

$$x = R_1^{-1}S_1 x + R_1^{-1}b$$

and

$$x = R_2^{-1}S_2x + R_2^{-1}b$$

Now, consider a linear program of the form

(1) $$\text{Max} \quad c'x$$

subject to

$$Ax=b$$
$$x \geqq 0$$
$$b \geqq 0$$

where A is m by n, c and x are n by 1, b is m by 1, and c' is the transpose of c. Let A_J denote the square submatrix of A formed by the m column identifications listed in J. The program listed in (1) will be referred to as a class K optimization if the transpose of every feasible basis A_J is a member of class K. The dual to program (1) is

$$\text{Min} \quad b'v$$

subject to

$$A'v \geqq c$$

Since b is non-negative, and as will be shown, the optimal v satisfies $v_{opt} \geqq v$, the dual problem is simply

(2)

$$\underset{J}{\text{Max}} \quad v = (A'_J)^{-1} c_J$$

Hence, for v an extreme point, v must satisfy

(3) $$A'_J v = c_J$$

for some J. Let N be the set of all J such that $A'_J \in K$ for $J \in N$. Problem (2) and Equation (3) immediately suggest the use of matrix iterative methods in finding v*, the optimal dual values. That is, let

$$A'_J = R_J - S_J$$

Then by Equation (3), we have that

$$R_J v = S_J v + c_J$$

or

(4) $$v = R_J^{-1} S_J v + R_J^{-1} c_J$$

provided R and S are chosen so that R is non-singular. Matrix iterative analysis suggests solving Equation (4) by the following recursion

(5) $$v^{n+1} = R_J^{-1} S_J v^n + R_J^{-1} c_J$$

Equation (5) converges if and only if $p(R_J^{-1} S_J) < 1$ [9].

Noting Equation (5) and Problem (2), the following procedure suggests itself for finding v*:

(6) $$v^{n+1} = \max_{J \in N} \quad R_J^{-1} S_J v^n + R_J^{-1} c_J$$

Now, providing $p(R_J^{-1} S_J) < 1$ and $R_J^{-1} S_J \geqq 0$ Veinott [14] has generalized results of Denardo [3] and proven the following:

Theorem 2

For a class K optimization problem where each A'_J is split so that

$$p(R_J^{-1} S_J) < 1$$

and

$$R_J^{-1} S_J \geqq 0$$

then Equation (6) provides a sequence which converges to the (unique) optimal dual values and the optimal basis A'_{J*} satisfies

$$v* = R_{J*}^{-1} S_{J*} v* + R_{J*}^{-1} c_{J*}$$

Splits of A'_j satisfying the requirements of Theorem 2 are referred to as asymptotic monotonic contractions or simply monotonic contractions.

II. Splits, Convergence Speed, and Computational Aspects

A class of splits which satisfy the requirements of Theorem 2 are now characterized.

Theorem 3

For $A'_J \in K$ and $B \in K$ where $B \geqq A'_J$, $J \in N$, then the set of all $\hat{R}$ and $\hat{S}$ satisfying

$$B^{-1}A'_J = \hat{R}_J - \hat{S}_J$$

with $\hat{S}_J \geqq 0$ provide monotonic contractions.

Proof

(i) Monotonicity

$$A'_J = B\hat{R}_J - B\hat{S}_J = R_J - S_J$$

then $\quad R_J^{-1}S_J = \hat{R}_J^{-1}B^{-1}B\hat{S}_J = \hat{R}_J^{-1}\hat{S}_J$ providing $\hat{R}_J$ is nonsingular.

Now, by Corollary 1, $B^{-1}A'_J \in K$ and, hence, $\hat{R}_J \in K$.

By Theorem 1, $\hat{R}_J^{-1}$ exists and $\hat{R}_J^{-1} \geqq 0$. Thus

$$R_J^{-1} S_J = \hat{R}_J^{-1} \hat{S}_J \geqq 0.$$

(ii) Contraction

Since $\hat{R}_J \geqq B^{-1}A'_J$, by Corollary 1

$$1 > p(I - \hat{R}_J^{-1}B^{-1}A'_J) = p\left(\hat{R}_J^{-1}(\hat{R}_J - B^{-1}A'_J)\right) = p(\hat{R}_J^{-1}\hat{S}_J) = p(R_J^{-1} S_J)$$

Q.E.D.

The convergence properties for two different splits of $B^{-1}A'_J$, labeled by

$$B^{-1}A'_J = \hat{R}_{J_1} - \hat{S}_{J_1} = \hat{R}_{J_2} - \hat{S}_{J_2} \qquad J \in N$$

where the subscript on J denotes a split type, are now characterized. For simplicity let $\hat{R}_{J_i}$ and $\hat{S}_{J_i}$ be written R_{J_i} and S_{J_i} respectively.

Theorem 4

In using Equation (6) with splits satisfying Theorem 2 where

$$R_{J_1} - S_{J_1} = R_{J_2} - S_{J_2} \text{ with } S_{J_1} \geqq S_{J_2} \geqq 0$$

then

$$v_2^{n+1} = \underset{J \in N}{\text{Max}} \; R_{J_2}^{-1} S_{J_2} v_2^n + R_{J_2}^{-1} c_{J_2}$$

converges to v* faster than

$$v_1^{n+1} = \underset{J \in N}{\text{Max}} \; R_{J_1}^{-1} S_{J_1} 1^n + R_{J_1}^{-1} c_{J_1}$$

Proof

Assume $v^i \rightarrow v^*$ from below (the proof for above follows in a similar fashion).

(i) let n = 1 and $v^0 \leq v^*$ be the initial guess. Then we have

$$v_1^1 = \underset{J \in N}{\text{Max}} \; R_{J_1}^{-1} S_{J_1} v^0 + R_{J_1}^{-1} c_{J_1} = R_{J_1^*}^{-1} S_{J_1^*} v^0 + R_{J_1^*}^{-1} c_{J_1^*}$$

$$v_2^1 = \underset{J \in N}{\text{Max}} \; R_{J_2}^{-1} S_{J_2} v^0 + R_{J_2}^{-1} c_{J_2} = R_{J_2^*}^{-1} S_{J_2^*} v^0 + R_{J_2^*}^{-1} c_{J_2^*}$$

Now, since $R_{J_1^*} = R_{J_2} - S_{J_2} + S_{J_1^*}$ for $J_2 = J_1^*$

$$v_1^1 = [I - R_{J_2}^{-1} S_{J_2} + R_{J_2}^{-1} S_{J_1^*}]^{-1} R_{J_2}^{-1} [S_{J_1^*} v^0 + c_{J_2}]$$

or

$$[I - R_{J_2}^{-1} S_{J_2} + R_{J_2}^{-1} S_{J_1^*}] v_1^1 = R_{J_2}^{-1} S_{J_1^*} v^0 + R_{J_2}^{-1} c_{J_2}$$

But,

$$v_2^1 \geqq R_{J_2}^{-1} S_{J_2} v^0 + R_{J_2}^{-1} c_{J_2}$$

Thus,

$$v_2^1 + (R_{J_2}^{-1} S_{J_1^*} - R_{J_2}^{-1} S_{J_2}) v^0 \geqq (I-R_{J_2}^{-1} S_{J_2} + R_{J_2}^{-1} S_{J_1^*}) v_1^1$$

Since

$v_1^1 \geqq v^0$ and by Corollary 2, part (vi) $v_2^1 \geqq v_1^1$

(ii) Assume $v_2^K \geqq v_1^K$ and consider

$$v_1^{K+1} = \underset{J \in N}{\text{Max}} R_{J_1}^{-1} S_{J_1} v_1^K + R_{J_1}^{-1} c_{J_1}$$

$$v_2^{K+1} = \underset{J \in N}{\text{Max}} R_{J_2}^{-1} S_{J_2} v_2^K + R_{J_2}^{-1} c_{J_2}$$

As in part (i)

$$(I-R_{J_2}^{-1} S_{J_2} + R_{J_2}^{-1} S_{J_1^*}) v_1^{K+1} = (R_{J_2}^{-1} S_{J_1^*} - R_{J_2}^{-1} S_{J_2}) v_1^K + R_{J_2}^{-1} S_{J_2} v_1^K + R_{J_2}^{-1} c_{J_2}$$

Thus,

$$v_2^{K+1} + (R_{J_2}^{-1} S_{J_1^*} - R_{J_2}^{-1} S_{J_2}) v_1^K \geqq R_{J_2}^{-1} S_{J_2} v_2^K + R_{J_2}^{-1} c_{J_2} + (R_{J_2}^{-1} S_{J_1^*} - R_{J_2}^{-1} S_{J_2}) v_1^K$$

But $v_2^K \geqq v_1^K$, thus

$$v_2^{K+1} + (R_{J_2}^{-1} S_{J_1^*} - R_{J_2}^{-1} S_{J_2}) v_1^K \geqq (I-R_{J_2}^{-1} S_{J_2} + R_{J_2}^{-1} S_{J_1^*}) v_1^{K+1}$$

Since $v_1^{K+1} \geqq v_1^K$ and, by Corollary 2, part (vi), $v_2^{K+1} \geqq v_1^{K+1}$

Q.E.D.

Another class of splits satisfying the requirements of Theorem 2 are given in the next theorem.

Theorem 5

For $A'_J \in K$ and $B^{-1}\hat{R}_J \in K$ with $B \in K$, $J \in N$, then the set of all $\hat{R}$ and $\hat{S}$ satisfying $BA'_J = \hat{R}_J - \hat{S}_J$ with $B^{-1}\hat{S}_J \geqq 0$ provide monotonic contractions.

Proof

(i) Monotonicity

$$A_J' = B^{-1}\hat{R}_J - B^{-1}\hat{S}_J = R_J - S_J$$

$R_J^{-1} S_J = \hat{R}_J^{-1} \hat{S}_J$ providing $\hat{R}_J$ is nonsingular.

But since $B^{-1}\hat{R}_J \in K$, $\hat{R}_J$ is nonsingular. Furthermore,

by Theorem 1, $\hat{R}_J^{-1} B \geqq 0$, thus $\hat{R}_J^{-1} \hat{S}_J \geqq 0$

(ii) Contraction

Now $B^{-1}\hat{R}_J \geqq A'_J$. Thus, by Corollary 1,

$$1 > \rho(I-\hat{R}_J^{-1} BA'_J) = \rho(\hat{R}_J^{-1}(\hat{R}_J - BA'_J)) = \rho(\hat{R}_J^{-1} \hat{S}_J)$$

Q.E.D.

Results similar to Theorem 4 can now be derived for the split category of Theorem 5. It is to be noted that replacing $B^{-1}A'_J$ is Theorem 3 by $A'_J B^{-1}$ also gives monotonic contractions as well as replacing BA'_J by $A'_J B$ in Theorem 5 with $\hat{R}_J B^{-1} \in K$ and $\hat{S}_J B^{-1} \geqq 0$.

Several splits ($\hat{R}$ and $\hat{S}$) of A'_J are now summarized.

(a) Regular splits

$B = sI$ where $s > 0$ (usually $s = 1$)

(i) Neumann Split

$$\hat{R}_J = I$$

$$S_J = I - \frac{1}{s} A'_J$$

(ii) Jacobi Split

$$\hat{R}_J = D_J \quad \text{(the diagonal of } \frac{1}{s} A'_J\text{)}$$

$$\hat{S}_J = D_J - \frac{1}{s} A'_J$$

(iii) Gauss-Seidel Split

$$\hat{R}_J = L_J \quad \text{(the lower triangular portion of } \frac{1}{s} A'_J\text{)}$$

$$\hat{S}_J = L_J - \frac{1}{s} A'_J$$

(iv) Totten Split [12]

$$R_J = \hat{R}_J (I-Q)$$

$$S_J = \hat{S}_J - \hat{R}_J Q$$

where $\hat{R}_J$ and $\hat{S}_J$ may be a Neumann, Jacobi, or Gauss-Seidel split of $\frac{1}{s} A'_J$ and $\hat{R}_J^{-1} \hat{S}_J \geqq Q \geqq 0 \qquad J \in N$.

(b) Similarity Transformation [14]

$$B \in K \text{ and } B^{-1} \in K \text{ and } A'_J B^{-1} \text{ is split.}$$

Again, the Neumann, Jacobi, Gauss-Seidel, and Totten variations may be considered. Note: this split does not fit Theorem 3 but can also be shown to yield monotonic contractions.

By Theorem 4, it is readily noted that, for a given B, the Gauss-Seidel split converges faster than the Jacobi split which converges faster than the Neumann split. This is desirable since the Gauss-Seidel split requires only one vector of storage as opposed to two for the Jacobi and Neumann split. It is readily shown that the Totten split converges faster than the corresponding companion split.

The fact that the Gauss-Seidel split converges faster than the Jacobi and Neumann splits has been known for iteratively solving systems of equations. But, in such settings, the Gauss-Seidel split converges asymptotically faster and not necessarily uniformly faster. Hence, Theorem 4 is somewhat surprising.

Now, the split yielding the fastest convergent sequence is not necessarily the best since computer storage and computational effort are not necessarily equal for different splits. It should be noted at this time that in computing v^n, R_J^{-1} should be carried out implicitly (if R_J^{-1} is diagonal on lower triangular) which permits the use of the original data for certain B matrices. Thus, the original problem sparsity is maintained.

A further complexing item in choosing a split is the following useful result of Hastings and Mello [6]

<u>Theorem 6</u> (Basis Elimination)

The basis $A_J^!$, $J \in N$ is sub-optimal if

$$R_J^{-1} S_J v^n + R_J^{-1} c_J \lneq v^n + b_2 - R_J^{-1} S_J b_1$$

where $$v^n + b_1 \geqq v^* \geqq v^n + b_2.$$

Thus, one split may yield a slower convergent split than another but be such as to eliminate potential basis faster, which, of course, is desirable. Even the computation of bounds [10,11] for one split might impose a far greater effort than for other splits. For example, it is well known that

$$\underset{i}{\text{Max}}\ R_J^{-1}\ S_J e \geqq p(R_J^{-1}\ S_J)$$

Thus, the maximum row sum of $R_J^{-1}\ S_J$ may not be strictly less than 1 as currently required in bound computations. To circumvent such cases, an integer M_J must be determined where

$$\|(R_J^{-1} S_J)^{M_J}\| < 1$$

and bounds computed only after M_J iterations of the algorithm.

A composite algorithm for solving class K problems is now given. Let parameters M, δ, and v^0 be specified where δ is a solution tolerance and M is a refinement parameter. A procedure, extending Equations (6), for solving class K problems is:

Step 1 (Pre-Processing)

Perform any possible pre-processing of data such as determining B or Q. Set n = 0 and go to Step 4.

Step 2 (Value Refinement)

$$v^{n+M+1} = (R_J^{-1}\ S_J)^{M+1}\ v^n + \sum_{\ell=0}^{M} (R_J^{-1}\ S_J)^{\ell}\ R_J^{-1}\ c_J$$

Step 3 (Basis Elimination)

Using Theorem 6, detect and eliminate $J \in N$ which are sub-optimal. If only one J is left in N, go to Step 6.

Step 4 (Basis Selection)

Choose another basis $J \in N$, labelled J_2, such that J_2 satisfies

$$\underset{J\in N}{\text{Max}}\ R_J^{-1}\ S_J\ v^n + R_J^{-1}\ c_J$$

for the current value v^n.

Step 5 (Termination)

If

$$|R_{J_2}^{-1} S_{J_2} v^n + R_J^{-1} c_J - v^n| \leqq \delta e$$ go to Step 6. Otherwise

set $v^{n+1} = R_{J_2}^{-1} S_{J_2} v^n + R_{J_2}^{-1} c_{J_2}$ and relabel J_2 by J and return to Step 2.

Step 6 (Post-Processing)

Perform any post-optimal computations such as finding v* and the optimal primal values given the optimal basis J*. Also, various types of splits require a post-optimal transformation.

It is to be noted that Step 3 should only be used when tight enough bounds are present. Furthermore, Steps 3 and 4 should be combined so as to avoid redundant computations.

III. Applications

It is well know that discrete-time discounted (semi-) Markov decisions may be considered as Class K problems. In such problems, the total discounted returns can be expressed as

$$v = (I - Q_J)^{-1} c_J$$

where J denotes a policy and Q_J a transient transition matrix. In the terminology of this paper

$$A'_J = I - Q_J \qquad J \in N$$

where N is the policy space. Totten [12] was the first to exploit the properties of the underlying functional equations to speed up convergence by using various types of splits of A'_J other than the split

$$R_J = I$$

$$S_J = Q_J$$

A much broader class of problems that can be solved as class K optimizations is the Leontief substitution system [15]. Consider the linear program given in (1) where the constraint matrix A contains, at most, one positive element per column. A is termed pre-Leontief. The i^{th} row of A is called trivial if for every $x \geqq 0$ for which $Ax \geqq 0$, the i^{th} component of Ax is zero, otherwise the i^{th} row is called non-trivial.

<u>Definition 3</u> (Leontief Matrix)

A matrix with exactly one positive element per column and non-trivial rows is termed a Leontief Matrix.

A pre-Leontief matrix with all its rows trivial is called a sub-Leontief matrix. Veinott [15] has shown that a pre-Leontief constraint set may first be permuted and partitioned to the form

$$Ax = \begin{pmatrix} A_1 & A_2 \\ 0 & A_3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

where A_1 is Leontief, A_3 is sub-Leontief, and A_2 has its positive elements above trivial columns of A_2. Then, the constraint set is equivalent to:

$$x_2 = 0$$

$$A_1 x_1 = b_1$$

$$x_1 \geqq 0$$

That is, the original pre-Leontief constraint set is equivalent to a smaller Leontief constraint set.

Now by the definition of a Leontief matrix, it is readily apparent that a square Leontief matrix, suitably permuted, belongs to class K. Using duality theory, it is also clear that the transpose of a square Leontief matrix belongs to class K. Consider a square pre-Leontief matrix with no vanishing rows. Veinott has shown that if such a matrix has linearly independent columns then it is equivalent to a square Leontief matrix. Since a basis must have linearly independent columns, it is clear that every feasible basis of a Leontief constraint set is a square Leontief matrix (hence, $A_J' \in K$). Thus, the iterative procedure of Section II can be used to solve pre-Leontief constrained linear programs

The extension of class K optimization techniques to Leontief Substitution Systems is significant since many typically large problems have pre-Leontief constraint sets. Such problems include input-output problems with substitution [4], deterministic inventory problems and certain network problems [16].

Another well known feature of Leontief Substitutions Systems is that if the final Leontief constraint set is reducible, then the problem may be solved as a sequence of smaller Leontief problems [1]. This observation offers considerable computational advantages.

The Leontief Substitution System is useful in relating other optimization problems to class K optimization problems. If an optimization problem can be shown to be equivalent to a Leontief Substitution System by an appropriate non-singular transformation, then that problem can be solved by class K optimization methods. For example, Nanda [8] has given algorithms for converting integer programs to equivalent Leontief Substitutions Systems (with possible bounded variables). Koehler, Whinston, and Wright [7] have presented a problem category which can be converted to Leontief Substitution Systems.

The last observation to be made concerning the Leontief Substitution System is that many problems have a partial Leontief Substitution System with other side constraints (such as bounds on variables). For such cases, the Dantzig-Wolfe decomposition method [2], using the Leontief Substitution constraint set as the sub-problem, is particularly efficient because of the ease of solution of the sub-problem.

IV. Computational Example

Consider the following linear program:

$$\text{Max} \quad -x_1 - 100x_2 - x_3 - 200x_4$$

subject to

$$x_1 + x_2 - x_3 - 0.8x_4 = 0.5$$

$$-x_1 - 0.8x_2 + x_3 + x_4 = 0.5$$

$$x_1, x_2, x_3, x_4 \geqq 0$$

The constraint set is pre-Leontief. Feasible basis are

$$N = \{(1,4),\ (2,3),\ (2,4)\}$$

Notice that A_J for $J = (1,3)$ is not Leontief due to the linear dependence of its columns. (Other criteria for detecting square Leontief - class K - matrices are given by Fiedler and Pták [5]).

The above problem illustrates a property observed by Totten [12] that if $c \leqq 0$ and the optimal basis A_J is Leontief, then the iterative procedure of Section II still converges to v^* even if some A_J' are pre-Leontief but not Leontief. Hence, letting

$$N = \{(1,3),\ (1,4),\ (2,3),\ (2,4)\}$$

will not disturb the computational procedure for this problem.

To illustrate the computational procedure of Section II, consider the following linear program:

$$\text{Max} \quad 2x_1 + 3x_2 + 1.6x_3 + 1.7x_4$$

subject to

$$0.8x_1 + 1.0x_2 - 1.0x_3 - 0.8x_4 = 2.0$$

$$-0.4x_1 - 0.2x_2 + 0.7x_3 + 0.5x_4 = 3.0$$

$$x_1, x_2, x_3, x_4, \geqq 0$$

let

$$N = \{(1,3),\ (1,4),\ (2,3),\ (2,4)\}$$

$$\delta = 0.00001$$

$$M = 1$$

$$v^0 = \binom{0}{0}$$

Furthermore, a Gauss-Seidel split with B = I will be used. The steps of the algorithm are:

Step 4:

$$v_1^1 = \text{Max} \begin{cases} \frac{2}{0.8} = 2.50 \\ \frac{3.0}{1.0} = 3.00 \end{cases}$$

$$v_2^1 = \text{Max} \begin{cases} \frac{1}{0.7}\{1.6 + 3.00\} = 6.571 \\ \frac{1}{0.5}\{1.7 + 0.8(3.00)\} = 8.200 \end{cases}$$

$$v^1 = \begin{pmatrix} 3.00 \\ 8.20 \end{pmatrix} \qquad J = (2,4)$$

Step 5:

$$|v^1 - v^0| > \delta e$$

Step 2:

$$v^3 = (R_J^{-1} S_J)^2 \; v^1 + \sum_{\ell=0}^{1} (R_J^{-1} S_J)^{\ell} \; R_J^{-1} c_J$$

v^3 is computed recursively as

$$v^2 = R_J^{-1} S_J \; v^1 + R_J^{-1} \; c_J$$

$$v^3 = R_J^{-1} \; S_J \; v^2 + R_J^{-1} \; c_J$$

$$v_1^2 = \frac{1}{1.0} \quad (3.00 + 0.2 \; (8.20)) = 4.640$$

$$v_2^2 = \frac{1}{0.5} \; (1.70 + 0.8 \; (4.640)) = 10.824$$

$$v_1^3 = \frac{1}{1.0} \; (3.0 + 0.2 \; (10.824)) = 5.165$$

$$v_2^3 = \frac{1}{0.5} \; (1.70 + 0.8 \; (5.165)) = 11.664$$

Step 3:

In the initial stages of computation, bounds on v* are not usually tight enough to permit their useage in a worthwhile manner.

Step 4:

$$v_1^4 = \text{Max} \begin{cases} \frac{1}{0.8} \; (2.00 + 0.4 \; (11.664)) = 8.332 \\ \frac{1}{1.0} \; (3.00 + 0.2 \; (11.664)) = 5.333 \end{cases}$$

$$v_2^4 = \text{Max} \begin{cases} \frac{1}{0.7} \; (1.60 + 8.332) = 14.189 \\ \frac{1}{0.5} \; (1.70 + 0.8 \; (8.332)) = 16.731 \end{cases}$$

or $$v^4 = \begin{pmatrix} 8.332 \\ 14.189 \end{pmatrix} \qquad J = (1,4)$$

REFERENCES

1. Dantzig, G. B., "Optimal Solution of a Dynamic Leontief Model with Substitution," Econometrica 23, 1955, pp. 295-302.

2. Dantzig, G. B. and P. Wolfe, "Decomposition Principle for Linear Programs," Operations Research, 8, 1960, pp. 101-111.

3. Denardo, E. V., "Contraction Mapping in the Theory Underlying Dynamic Programming," SIAM Review, 9, No. 2, 1967, pp. 165-177.

4. Dorfman, R., P. A. Samuelson, and R. M. Solow, Linear Programming and Economic Analysis, Rand Corporation, 1958, McGraw-Hill Book Company, N.Y.

5. Fiedler, M. and V. Pták, "On Matrices with Non-Positive Off-Diagonal Elements and Positive Principle Minor," Czech. Mathematics Journal, 12, 1962, pp. 382-400.

6. Hastings, N.A.J., and J.M.C. Mello, "Tests for Suboptimal Actions in Discounted Markov Programming," Management Science, 19, No. 9, 1973, pp. 1019-1022.

7. Koehler, G. J., A. B. Whinston, and G.P. Wright, "An Iterative Procedure for Non-Discounted Discrete-Time Markov Decisions," submitted for publication.

8. Nanda, P. S., "The Equivalence of Integer Programs to Constrained Recursive Systems," Management Science, 19, No. 7, 1973, pp. 809-824.

9. Oldenburger, R., "Infinite Powers of Matrices and Characteristic Roots," Duke Mathematical Journal, 6, 1940, pp. 357-361.

10. Porteus, E. L., "Bounds and Transformations in the Computational Theory for Finite State and Action Markov Decision Processes," Research Paper No. 97, Graduate School of Business, Stanford University, August, 1973.

11. Porteus, E. L., "Some Bounds for Discounted Sequential Decision Processes," Management Science, 18, No. 1, 1971, pp. 7-11.

12. Totten, J. C., "Computational Methods for Finite State Finite Valued Markovian Decision Problems," Operations Research Center, University of California, Berkeley, 1971.

13. Varga, R. S., Matrix Iterative Analysis, 1962, Prentice-Hall, Inc., Englewood Cliffs, N.J.

14. Veinott, A. F., Jr., "Discrete Dynamic Programming with Sensitive Optimality Criteria," Annals of Mathematical Statistics, Vol. 40, 1969, pp. 1635-1660.