A STRONGLY POLYNOMIAL ALGORITHM FOR A SPECIAL CLASS OF LINEAR PROGRAMS

I. ADLER
University of California, Berkeley, California

S. COSARES
Bell Communications Research, Piscataway, New Jersey

(Received July 1989; revision received February 1990; accepted July 1990)

We extend the list of linear programming problems that are known to be solvable in strongly polynomial time to include a class of LPs which contains special cases of the generalized transshipment problem. The result is facilitated by exploiting some special properties associated with Leontief substitution systems and observing that a feasible solution to the system, $Ax = b$, $x \geq 0$, in which no variable appears in more than two equations, can be found in strongly polynomial time for $b$ belonging to some set $\Omega$.

Consider the pair of linear programming problems (LPs):

**Problem P**
Minimize $c^T x$
subject to $Ax = b$
$x \geq 0$.

**Problem D**
Maximize $b^T y$
subject to $A^T y \leq c$
where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times n}$ has full row rank, so $m \leq n$.

When some problem parameter, $a_{ij}$, $b_i$, or $c_j$, is a rational number, say $p/q$, we say that it has a "length," which is equal to the number of digits in its binary encoding, $\lceil \log_2(\lceil p \rceil + 1) \rceil + \lceil \log_2(\lceil q \rceil + 1) \rceil$. (Irrational numbers are assumed to have infinite length.)

The sum of the lengths of the parameters representing an instance of P or D is usually denoted by $L$.

An algorithm that solves a class of LPs is said to run in strongly polynomial time if the number of elementary operations can be bounded by a function that is polynomial in $n$ and $m$, and is independent of $L$.

At present, there is no algorithm that can solve a general instance of the linear programming problem in strongly polynomial time. However, by using any of the polynomial algorithms for the problem (e.g., the method of Khachiyan 1979 or Karmarkar 1984, or their variants) as a subroutine, Tardos (1986) provided an algorithm for linear programming whose running time bound is a polynomial function of $m$, $n$ and the length of the elements in $A$, and is independent of the length of the elements in $b$ and $c$. Consequently, the class of LPs having a constraint matrix with elements whose length is boundable by a polynomial in $n$ and $m$, can be solved in strongly polynomial time.

In order to find additional classes of the LP problem that are solvable in strongly polynomial time, we have to identify conditions for $A$, $b$ or $c$, which define LP classes with properties that lend themselves to efficient solution techniques. For LPs with an $a_{ij}$ very large in length, we expect to find encouraging results when $A$ has a peculiar distribution of nonzero elements or element signs. For such systems, feasible bases may be structured specially, so it would be easier to characterize and hence find feasible or optimal solutions. In the next section, we describe Leontief substitution systems and systems of inequalities in which each no constraint has more than two participating variables. These systems have properties which allow for more efficient solution techniques than do more general ones.
systems. We also describe the (generalized) transshipment problem, which is related to these systems.

Megiddo (1983) presented a strongly polynomial algorithm which finds a feasible solution to inequality systems in which each no constraint has more than two participating variables. In Section 2, we show how the algorithm can be used to find a feasible solution to a related system, $Ax = b$, $x \geq 0$, in which no variable appears in more than two equations in strongly polynomial time for $b$ belonging to some set $\Omega$. In Section 3, we show that if the system is also a pre-Leontief substitution system, then an optimal solution to any LP having the system as a constraint set (and its dual) can be found in strongly polynomial time. Thus, we extend the list of LP classes that are known to be solvable by a strongly polynomial algorithm.

1. PRELIMINARIES

1.1. Notation

Let $A_i$ denote the $i$th column of $A$ and $A_i^r$ denote the $i$th row of $A$. For a set $S \subseteq \{1, \ldots, n\}$, $A_S$ is defined to be the submatrix of $A$ consisting only of those columns with an index in $S$. Similarly, $A_S^r$ is defined to be the submatrix consisting only of those rows with indices in $R \subseteq \{1, \ldots, m\}$. For a (column) vector, $v$, we let $v_i$ denote the vector consisting of the elements with indices in $R$.

A set of $m$ linearly independent columns of $A$ is called a basis. If $A_S^r$ is a basis, we say that it is feasible for the system in $P$ if the unique (basic) solution to the system, $A_S^r x_S = b$, is such that $x_S \geq 0$ (i.e., if $A_S^r b$ is nonnegative). The basis $A_S^r$ is feasible for the system in $D$ if the unique solution, $x_S \in \mathbb{R}^m$, to $y^T A_S = c_S$ is such that $y^T A_j \leq c_j$ for all $j \in S$.

We say that a system of linear constraints has a maximal solution if there exists a feasible solution, $x^*$, such that $x^* \geq y$ for any feasible solution $y$. Similarly, the system has a minimal solution if there exists a feasible, $x^*$, such that $x^* \leq y$ for any feasible $y$. Obviously, a maximal (minimal) solution is optimal for a maximization (minimization) LP problem with a nonnegative objective function, if such a solution exists.

1.2. Leontief Substitution Systems

If a matrix $A$ has no more that one positive element in any column it is called a pre-Leontief matrix. The system $Ax = b$, $x \geq 0$ is called a pre-Leontief substitution system (PLSS) if $A$ is pre-Leontief and $b$ is a nonnegative vector. $A$ is called a Leontief matrix if there exists at least one solution to the PLSS, whenever $b$ is strictly positive. The system is called a Leontief substitution system (LSS) if $A$ is Leontief and $b$ is strictly positive.

The relationship between pre-Leontief and Leontief substitution systems is described by the following result due to Veinott (1968).

**Proposition 1.** If $Ax = b$, $x \geq 0$ is a PLSS, with at least one feasible solution, then $A$ and $b$ can be partitioned after suitably permuting rows and columns to take the form:

\[
\begin{pmatrix}
A^1 & A^2 \\
0 & A^3
\end{pmatrix}
\begin{pmatrix}
x^1 \\
x^2
\end{pmatrix}
= \begin{pmatrix}
 b^1 \\
0
\end{pmatrix}
\]

where

i. $A^1$ is a Leontief matrix, and
ii. $x$ is basic feasible solution to $A x = b$, $x \geq 0$ if and only if $(\tilde{x}^1, 0)$ is a basic feasible solution to $Ax = b$, $x \geq 0$.

Leontief substitution systems and their feasible bases are discussed in detail by Veinott (1968) and Koehler, Whinston and Wright (1975). LSS have many properties not found in more general systems, so they are worthy of special consideration. For instance, every feasible basis for an LSS has a nonnegative inverse (Dantzig 1955), so the system has a feasible solution for any nonnegative right-hand side. The following property of Leontief matrices also serves to be particularly useful.

**Proposition 2.** If $A$ is a Leontief matrix, then the system $A^T y \leq c$, has a finite maximal solution for all $c$ for which the system is feasible. Furthermore, the system $A^T y \geq c$, has a finite minimal solution for all $c$ for which the system is feasible.

A proof of this result is presented by Cottle and Veinott (1972), who characterize polyhedral sets defined by inequality systems with maximal (minimal) solutions.

1.3. Constraint Systems With Two Variables

Per Inequality

Consider cases of $P$ and $D$ in which every column in matrix $A$ contains no more than two nonzero elements. We may chose to represent the structure of the constraint matrix by way of an undirected graph with $n$ edges, corresponding to the columns of $A$.

If the $k$th column has nonzero elements in rows $i$ and $j$, then edge $e_k$ in the graph links nodes $i$ and $j$ (i.e., $e_k = \{i, j\}$). If the column has only one nonzero element located in row $j$, then $e_k$ links node $j$ to a root
A path of length $K$ from node $i$ to node $j$ is a sequence of adjacent edges of the form $\{i, h_1, \ldots, h_{K-1}, j\}$.

Shostak (1981) describes a methodology by which the graph structure may be used to obtain a feasible solution to $D$ (i.e., where the inequalities are of the form, $ay_j + by_i \leq c$). Suppose that the system contains the pair of constraints, $ay_j + ay_i \leq c_1$ and $by_j - by_i \leq c_2$, where both $a_i$ and $b_j$ are strictly positive. A nonnegative linear combination of these constraints implies the (redundant) constraint, $(b_ia_j) y_j + (a_ib_j) y_i \leq (b_ic_1 + a_ic_2)$, which also involves no more than two variables. (Either of $a_i$ or $b_j$ may be 0.)

Generating a new constraint by combining a pair of constraints in this way is called performing an admissible aggregation. Notice that the admissible aggregation above involves a pair of constraints which corresponds to adjacent edges, $\{i, j\}$ and $\{j, k\}$, in the graph representing $A$. We define an admissible path to be a sequence of admissible aggregations of a set of constraints which corresponds to a path in the graph. (It should be made clear that not every adjacent pair of edges in the graph corresponds to an admissible aggregation because the signs of the coefficients involved must be appropriate.)

Combining constraints along an admissible path from node $i$ to node $j$ determines a constraint involving only variables $y_i$ and $y_j$. Thus, an admissible path from node $i$ to node 0 or from node $i$ to itself determines a constraint involving only $y_i$, revealing either an implicit upper bound or an implicit lower bound to the variable $y_i$.

We define $y_i^{\text{max}}$ and $y_i^{\text{min}}$ to be, respectively, the tightest upper bound and lower bound to $y_i$ obtainable from admissible paths in the graph. If no such upper (lower) bound exists, then $y_i^{\text{max}}$ ($y_i^{\text{min}}$) is set to infinity (negative infinity).

From an adaptation of a theorem by Shostak (1981) we find the following proposition.

**Proposition 3.** The system $A^T y \leq c$ is feasible if and only if $y_{i}^{\text{max}} \geq y_{i}^{\text{min}}$ for all $i$. Furthermore, if the system is feasible, then $y_{i}^{\text{max}}$ and $y_{i}^{\text{min}}$ are, respectively, the largest and smallest values $y_i$ can take in any feasible solution to the system.

Aspvall and Shiloach (1980) present an algorithm which finds $y_{i}^{\text{max}}$ and $y_{i}^{\text{min}}$ for all $k$ in $O(m^3n \log n)$ time. The authors also describe a routine which finds a feasible solution to the system within the same time bound. Megiddo (1983) uses a technique, which we call “parameterized searching,” to improve the running time bound on the algorithm to $O(m^3n \log n)$ time (i.e., strongly polynomial time).

### 1.4. The (Generalized) Transshipment Problem

Let $G = [N, E]$ be a directed graph with node set, $N = \{0, 1, \ldots, m\}$, and edge set,

$E = \{e_k = (i, j) \mid k = 1 \ldots n\}$.

The “flow” into edge $e_k = (i, j)$ is defined to be a number assigned to the edge, which represents the amount of some commodity sent from node $i$ to node $j$. Associated with each edge is a positive “flow multiplier,” $d_k$. If the flow assigned to $e_k$ is $x_k$, then $d_k x_k$ units are received at node $j$.

The vector $b \in \mathbb{R}^m$ represents the total demand for the commodity at each of nodes $1 \ldots m$ (the negative demand at node $i$ corresponds to a “supply” that must be depleted). The quantity $c_k$ represents the cost incurred for each unit of flow assigned to edge $e_k$. The generalized transshipment problem, (GTP) is defined to be the task of finding an assignment of flow to the edges which satisfies the demand at nodes $1 \ldots m$ at minimum total cost. The ordinary transshipment problem (TP) is the special case of GTP where each of the flow multipliers is unity.

A commonly used LP formulation for the GTP is as follows.

**Problem GTP**

Minimize $\sum_{k=1}^{n} c_k x_k$

subject to $\sum_{k: e_k = (i, j)} d_k x_k - \sum_{k: e_k = (j, i)} x_k = b_j$

$0 \leq x_k \leq b_j$

which has, as its dual:

**Problem DGTP**

Maximize $\sum_{i=1}^{m} b_i y_i$

subject to $y_j - y_i \leq c_{ij}$ (i, j) = $e_k$, $k = 1 \ldots n$

$y_0 = 0$.

A “flow conservation” constraint for node 0 is not included in the primal LP representation. Accordingly, the variable in the dual problem corresponding
to this node is set to 0.) This allows our discussion to be applicable to the ordinary TP case, where the constraint would be redundant. Notice that GTP and DTP can be represented as cases of P and D, respectively, in which the constraint matrix A has no more than two nonzero elements in any of its columns and both A and \(-A\) are pre-Leontief.

Since it is possible for an edge multiplier to have a length, which is not boundable by a polynomial in \(n\) and \(m\), an arbitrary instance of GTP is not guaranteed to be solvable in strongly polynomial time by any of the algorithms described earlier.

It turns out, though, that efficient algorithms exist for some special cases of GTP. Goldberg, Plotkin and Tardos (1988) provide the first polynomial algorithms for the capacitated case which are combinatorial in nature (i.e., not based on an interior point or ellipsoid method). Charnes and Raike (1966) present a one-pass method which solves GTP when \(b \geq 0\) and \(c \geq 0\) and the network either has no directed cycles or no edges with a flow multiplier greater than one. In such cases, the feasible bases for P, if any exist, have a structure which makes the problem solvable in \(O(m^3n \log n)\) time by a variant of Dijkstra’s (1959) method. The authors also show that if \(A\) without the columns corresponding to the edges incident to node 0, is row rank deficient, then there exist positive diagonal scalars \(R\) and \(C\) (of appropriate dimension), such that \(RAC\) is the node-arc incidence matrix of an ordinary network. Glover and Klingman (1973) give an efficient method for finding \(R\) and \(C\), if they exist. This case is then solvable in strongly polynomial time, for any values of \(b\) or \(c\) by the method of Tardos (1985).

**2. STRONGLY POLYNOMIAL ALGORITHMS**

### 2.1. A Feasibility Problem

Consider instances of P and D, where matrix \(A\) has no more than two nonzero elements in any column. Recall that \(y_i^{\text{max}}\) and \(y_i^{\text{min}}\) are, respectively, the largest and smallest values \(y_i\) can take in any feasible solution to the system in D. Let

\[
I^+ = \{i \in [1 \ldots m] \mid y_i^{\text{max}} < \infty\}
\]

\[
I^- = \{i \in [1 \ldots m] \mid y_i^{\text{min}} > -\infty\}
\]

\[
\Omega = \{w \in \mathbb{R}^m \mid w_i > 0 \text{ implies } i \in I^+, \quad w_i < 0 \text{ implies } i \in I^-.\}
\]

**Lemma 1.** A feasible solution to the system in P, where \(b \in \Omega\), can be found in \(O(m^3n \log n)\) time.

**Proof.** \(y_i^{\text{max}}\) can be interpreted as the optimal objective value for D when \(b = e_i\), the \(i\)th column of an \(m \times m\) identity matrix. Similarly, \(y_i^{\text{min}}\) can be interpreted as the optimal objective value for D when \(b = -e_i\). From duality theory, we know that if \(y_i^{\text{max}}\) is finite, then the system, \(Ax = e_i, x \geq 0\), has a feasible solution and if \(y_i^{\text{min}}\) is finite, then the system, \(Ax = -e_i, x \geq 0\), has a feasible solution.

For \(i \in I^+\), the constraint \(y_i \leq y_i^{\text{max}}\) is obtained by taking a nonnegative linear combination of a set of constraints in the system in D that corresponds to an admissible path in the graph representing \(A\). Let \(x_+ \in \mathbb{R}^n\) be the vector whose components are these nonnegative multipliers. Similarly, for \(i \in I^-\), let \(x_- \in \mathbb{R}^n\) the vector of nonnegative multipliers used to generate the bound, \(y_i \geq y_i^{\text{min}}\).

It is easy to verify that, \(Ax_+ = e_i, x_+ \geq 0\) and \(Ax_- = -e_i, x_- \geq 0\). For any \(b \in \Omega\), a feasible solution to the system in P, \(x\), is obtained as

\[
\hat{x} = \sum_{i \in I^+} \max(b_i, 0) \times x_+^i + \sum_{i \in I^-} \max(-b_i, 0) \times x_-^i.
\]

By proper “bookkeeping” in the algorithm of Megiddo (1983), it is possible to find the set of constraints that determine each of \(y_i^{\text{max}}\) and \(y_i^{\text{min}}\) and the associated nonnegative multipliers which determine these bounds. All of this can be accomplished in \(O(m^3n \log n)\) time.

### 2.2. A Special Class of LPs

Consider instances of P and D, where \(A\) is pre-Leontief and has no more than two nonzero elements in any column, and \(b\) is nonnegative. (Note: certain cases of GTP and DGTP can be cast as problems in this class.)

**Lemma 2.** Suppose that the system in D is feasible. If \(y_i^{\text{max}}\) is finite for \(i \in M_1\), and infinite for \(i \in M_2\), then P has a feasible solution if and only if \(b_i = 0\) for all \(i \in M_2\).

**Proof.** If \(b_i = 0\) for all \(i \in M_2\), then \(\sum_{i \in M_1} b_i y_i^{\text{max}}\) is an upper bound to the optimal objective function value of D. Since D is feasible and bounded, P must also be feasible.

If, on the other hand, P is feasible, with \(b_i > 0\) for some \(i \in M_2\), then \(A\) must be one of the rows in the Leontief submatrix of \(A\), described in Proposition 1. Because the feasible bases of LSS have nonnegative inverses, it follows that the system, \(Ax = e_i, x \geq 0\), has a feasible solution, say \(\hat{x}\). By weak duality, \(c^T\hat{x}\) provides a finite upper bound to the optimal objective of:
Maximize $y_i$
subject to $A^Ty \leq c$
which contradicts that $y_i^{\text{max}}$ is infinite.

Permute the rows and columns of $A$ so that

\[
A = \begin{pmatrix} A_{M_1N_1} & A_{M_1N_2} \\ 0 & A_{M_2N_1} \end{pmatrix} = \begin{pmatrix} A_1 & A_2 \\ 0 & A_3 \end{pmatrix}.
\]

From the proof of the lemma, if $P$ is feasible, then $A_1$ is a Leontief submatrix of $A$. So the system, $A_1^Ty_{M_1} \leq c_{N_1}$, has a finite maximal solution, $y_{M_1}^{\text{max}}$. Since $b_{M_1}$ is nonnegative and $b_{M_2} = 0$, we can conclude that the vector $y^* = (y_{M_1}^{\text{max}}, y_{M_2})$ is an optimal solution to $D$, where $y_{M_2}$ satisfies

\[
A_2^Ty_{M_2} \leq c_{N_2} - A_3^Ty_{M_1}^{\text{max}}.
\]

(Note that the existence of a feasible solution to this system is assured because for each $i \in M_2$, $y_i^{\text{max}}$ is infinite, hence, larger than $y_i$. It could be found in $O(m^2n)$ time (Aspvall and Shiloach).

From duality theory and Proposition 1, if $y^*$ is an optimal solution to $D$, then an optimal solution to $P$ is $x^* = (x_S, 0)$, which can be obtained by finding a feasible solution to the system

\[
A_{M_S}x_S = b_{M_1}, \quad x_S \geq 0
\]
where $S = \{j \in N_1 | (A_1^Ty_{M_1}^*) = c_j\}$. Since $y_{M_1}^{\text{max}}$ is finite for all $i \in M_1$, this can be accomplished by the methodology described in the previous section. Thus, we have proved the following theorem.

**Theorem 1.** If $A$ is a pre-Leontief matrix with no more than two nonzero elements in any column, then optimal solutions to $P$ and $D$, where $b \geq 0$, if they exist, can be found in $O(m^2n \log n)$ time.

**Corollary.** Any instance of GTP with either $b \leq 0$ or $b \geq 0$ can be solved in strongly polynomial time.

(Note that for GTP, the statements made above apply to the case where $b \leq 0$, because for GTP $-Ax = -b$, $x \geq 0$ is also a pre-Leontief substitution system.)

### 3. REMARKS

After having found efficient solution techniques for LPs with a constraint set which comprises an LSS whose variables appear in no more than two constraints, we are curious as to whether there are other classes of LPs with Leontief constraint matrices for which an efficient algorithm can be found. It seems that the special properties associated with these problems can be exploited to allow for a strongly polynomial algorithm for a more general class of problems, particularly since there is a characterization of feasible bases which is not dependent on the particular values of the numbers in the constraint matrix.

Jeroslow et al. (1989) describe properties associated with a class of LSS, which are called "gainfree." By exploiting some hypergraph representation of the system, they have shown that the class of LPs with a gainfree Leontief constraint matrix can be solved in strongly polynomial time. They also show that since primal feasible bases for these LPs are triangular, instances with an integral constraint matrix and rational right-hand side are totally dual integral.

The notion of parameterized searching, which was originally introduced by Megiddo (1979), has proven to be a very useful tool for algorithmic development. Performing a parameterized search, in place of, say, a series of binary searches not only helps in improving the running time of existing algorithms, but has led the way for the development of new efficient algorithms. Adler and Cosares (1989) describe a "nested" version of the technique which has led the way for the solution of additional classes of the LP problem in strongly polynomial time.

### ACKNOWLEDGMENT

This research was partially funded by the United States Office of Naval Research under contract N00014-87-K-0202. Their support is gratefully acknowledged. The authors also wish to thank the anonymous referees for their helpful comments.

### REFERENCES


