DISCUSSION PAPER NO. 67

AN OPERATOR THEORY OF PARAMETRIC PROGRAMMING FOR THE GENERALIZED TRANSPORTATION PROBLEM

I. BASIC THEORY+

by

V. Balachandran *

and

G. L. Thompson **

January 18, 1974

+ Also Released as Management Sciences Research Report No. 293 - October 1972

This report was prepared as part of the activities of the Management Science Research Group, Carnegie-Mellon University, under Contract N00014-67-A-0314-0007 NR 047-048 with the U. S. Office of Naval Research. Reproduction in whole or in part is permitted for any purpose of the U. S. Government.

* Associate Professor, Graduate School of Management, Northwestern University, Evanston, Illinois 60201

** Professor, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213

ABSTRACT

This paper gives a new organization of the theoretical results of the Generalized Transportation Problem with capacity constraints. A graph-theoretic approach is utilized to define the basis as an one-forest consisting of one-trees (a tree with an extra edge). Algorithmic development of the pivot-step is presented by the representation of a two-tree (a tree with two extra edges). Constructive procedures and proofs leading to an efficient computer code are provided. The basic definition of an operator theory which leads to the discussion of various operators is also given. In later papers we will present additional results on the operator theory for the generalized transportation problem based on the results in the present paper.

1. INTRODUCTION

The Generalized Transportation Problem was introduced as an extension of the transportation problem by A. R. Ferguson and G. B. Dantzig [9, 12] in their application to "The Problem of Routing Aircraft." This was further applied by K. Eisemann and J. R. Lourie [11] for "The Machine Loading Problem," and also discussed and applied by A. Charnes and W. W. Cooper [8]. Formal methods of solving the generalized transportation problem and relevant theoretical insights are given in standard texts, for instance, A. Charnes and W. W. Cooper [8], G. B. Dantzig [9], W. W. Garwin [13] and G. Hadley [17]. Extensions of the loop-technique of the stepping-stone algorithm and related theoretical underpinnings are presented by E. Balas and P. L. Ivanescu [6]. Results similar to those by Balas and Ivanescu [6], and consideration of degeneracy and upper bounds are presented by K. Eisemann [10] while the topology of the generalized transportation problem at the end of each iteration of the stepping-stone method of solution was given by J. R. Lourie [19]; and two together developed the original version of the computer code [11] for IBM 704. Post-optimization and inclusion of additional constraints for the generalized transportation problem were treated by E. Balas [5], utilizing the dual-method and the poly-ω technique.

An operator theory of parametric programming for the transportation problem was developed by V. Srinivasan and G. L. Thompson in [21, 22]. In the present and subsequent [1, 2, 3] papers we extend the parametric approach to generalized transportation problems. The present paper develops the basic theoretical framework, including a new approach to representing one-trees (a tree with an extra edge) and two-trees (a tree with two extra edges) in a manner that is efficient for both theory and computation. The basic definitions of operators are also included. The detailed development of the various operators based on the results of the present paper is carried out in [1, 2, 3].

Throughout this paper, our emphasis is on constructive procedures and proofs since they lead directly to usable and hopefully efficient computer programs. Our development of operator theory for the generalized transportation problem is straight forward and requires no reference to the parametric programming theory for the general linear program [9, 19] for justification. Our aim in this paper is to simultaneously provide new theoretical insights and to give constructive procedure for obtaining the practical benefits of the new results. This also facilitates easy application and interpretation to certain management problems which are left for another paper.

The generalized transportation problem formulation arises in different contexts [9, 11, 12], but the most familiar application is the machine loading problem [11]. In this, m types of machines (rows) are available for the production of n types of products (columns). The production process is concerned with each unit of product being processed by a single machine and not by a specific sequence of machines. Each product may be produced by any one or more machines. The utilization of machine type i for product j requires e_{ij} hours per unit and costs c_{ij} dollars per unit. During a fixed time period, machines of type i have a maximum total capacity of a_i hours and product type j is required by an amount b_j. The machine loading problem is: In what amounts of x_{ij} should products be allotted to machines to attain production of required amounts within the available capacities of minimum total cost? Formulated as a linear programming model, the problem is:

$$\text{Minimize} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}x_{ij}$$

$$(1) \quad \text{Subject to:} \quad \sum_{j=1}^{n} e_{ij}x_{ij} \leq a_i \quad (i=1,\ldots,m)$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j=1,\ldots,n)$$

$$x_{ij} \geq 0$$

Due to similarity of (1) to the ordinary transportation problem and to take advantage of the structure in the constraint matrix, this is treated as a generalized transportation problem. In fact there can be weights similar to e_{ij} for the columns as well. For instance, consider the following problem:

$$\text{Minimize} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} c'_{ij}x'_{ij}$$

$$(2) \quad \text{Subject to:} \quad \sum_{j=1}^{n} a'_{ij}x'_{ij} \leq a_i; \quad i=1,\ldots,m$$

$$\sum b'_{ij}x'_{ij} = b_j; \quad j=1,\ldots,n$$

$$x'_{ij} \geq 0$$

where the a'_{ij}'s are non-negative, a_i, b_j, b'_{ij} are positive and c'_{ij} are arbitrary. This problem (2) can be transformed to (1) (or with all column coefficients to be unity and arbitrary row coefficients or vice-versa). Without any loss of generality we shall transform the problem so that the column coefficients are unity as in (1). To do this we let $x_{ij} = b'_{ij}x'_{ij}$. Thus the coefficients for the row equations become $a'_{ij}/b'_{ij} = e_{ij}$ and the coefficients of the objective function become $c'_{ij}/b'_{ij} = c_{ij}$. Thus (2) and (1) are equivalent since $x'_{ij} \geq 0$ implies $x_{ij} \geq 0$ because b'_{ij} is positive.

Throughout the rest of this paper we shall concentrate on (1).

2. PRELIMINARY DEFINITIONS AND RESULTS.

The inequalities given in (1) for the row sums can be changed to equalities by introducing slack variables $x_{i,n+1}$ for each i, with an associated weight $e_{i,n+1} \equiv 1$ and $c_{i,n+1} = 0$ for all i. However no explicit constraint is imposed on the $(n+1)$st column sum (which can be interpreted as the total idle machine time). To ensure an initial feasible solution and also to ensure sufficient machine capacity, a fictitious row (machine type $m+1$) with very large machine capacity a_{m+1} is introduced. By assigning very high per unit production costs $c_{m+1,j} = M$ ($j \leq n$, and M a large positive number) any allotment to this (high cost) machine is penalized. In other words, if the optimal solution has a positive $x_{m+1,j}$ ($j \neq n+1$), then there is no feasible solution to the original problem. Also set $c_{m+1,n+1} = 0$, $e_{m+1,n+1} = 1$. If upper limits U_{ij} (capacity) for x_{ij} is imposed then the <u>capacitated generalized transportation problem</u> can now be stated.

The following index sets are defined:

(3) $I = \{1,2,\ldots,m, m+1\}$ the set of machine types

(3′) $I' = \{1,2,\ldots,m\} = I - \{(m+1)\}$

(4) $J = \{1,2,\ldots,n, (n+1)\}$: the set of product types

(4′) $J' = \{1,2,\ldots,n\} = J - \{(n+1)\}$

For $i \in I$ and $j \in J$, define the following quantities (each with its machine loading interpretation)

x_{ij}: amount of product type j to be allocated to machine type i.

c_{ij}: cost per unit of such allocation.

e_{ij}: production time of machine type i for product type j.

U_{ij}: maximum amount of product j that can be allocated to machine i. If there is no maximum then set $U_{ij} = N$.

a_i: availability (supply)of total time for machine type i.

b_j: requirement (demand) of product j.

We will refer to c_{ij}, e_{ij}, U_{ij}, a_i and b_j as the <u>data</u> of the problem. In particular the a_i's and b_j's will be referred to as '<u>rim conditions</u>.' Thus the capacitated generalized transportation problem (P) will be:

(5) Minimize $\sum_{i \epsilon I} \sum_{j \epsilon J} c_{ij} x_{ij} = Z$

Subject to the following constraints:

(6) $\sum_{j \epsilon J} e_{ij} x_{ij} = a_i$ for $i \epsilon I$

(7) $\sum_{i \epsilon I} x_{ij} = b_j$ for $j \epsilon J'$

(8) $0 \leq x_{ij} \leq U_{ij}$ for $i \epsilon I'$, $j \epsilon J'$

with the following assumptions

(A1) a_i, b_j, e_{ij}, U_{ij} are positive real numbers for all $i \epsilon I'$, and $j \epsilon J'$ and

(A2) the system (6-8) has a feasible solution. (due to construction).

(A3) non-degenerate solutions exist. If degeneracy occurs (primal or dual) familiar methods [8, 10] could be used to prevent it.

REMARK 1. There is always a feasible solution to P due to construction of (m+1)st row. (A2). Finiteness of x_{ij} is due to constraint (8). Multiple optimal solutions if any are due to (dual) degeneracy [10]. $x_{m+1,j} > 0$ ($j \neq n+1$) at the optimal solution is a sufficient condition that no feasible solution to the original problem (1) exists. We also discuss how to find an initial solution (which need not be optimal) later.

REMARK 2. For problem in which some or all of the x_{ij} are not bounded from above, we take $U_{ij} = N$ where N is a large positive number.

If specific lower bounds are given viz $0 \leq \ell_{ij} \leq x_{ij} \leq U_{ij}$ for $i \epsilon I'$ and $j \epsilon J'$, it is possible to effect the linear transformation $x'_{ij} = x_{ij} - \ell_{ij}$ so that $0 \leq x'_{ij} \leq U'_{ij}$ where $U'_{ij} = (U_{ij} - \ell_{ij})$. This will change the rim conditions (6) and (7) as follows:

(6′) $\sum_{i \epsilon I} x'_{ij} = b'_j$ for $j \epsilon J'$ where

$$0 \leq b'_j = b_j - \sum_{i \epsilon I} \ell_{ij} \quad \text{and}$$

(7′) $\sum_{j \epsilon J} e_{ij} x'_{ij} = a'_i$ for $i \epsilon I$ where

$$0 \leq a'_i = a_i - \sum_{j \epsilon J} e_{ij} \ell_{ij}.$$

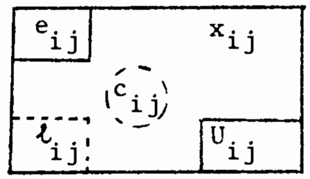
Due to Assumption A2, there will be a feasible solution for x'_{ij} because the mapping is linear and preserves all conditions cited in (A1). The objective (5) will become

(5′) $\sum_{i \epsilon I} \sum_{j \epsilon J} c_{ij} x'_{ij} + k$ where k is a constant and equals

$$\sum_{i \epsilon I} \sum_{j \epsilon J} c_{ij} \ell_{ij}.$$

Since the transformation $X' = X + b$ is linear and (A1) and (A2) hold for the new problem, the optimal solution X^* for the original problem will correspond to optimal solution $X^{*\prime}$ where $X^{*\prime} = X^* - L = \{x^*_{ij} - \ell_{ij}\}$. Henceforth, we will tacitly assume that $\ell_{ij} \equiv 0$ for the generalized transportation problem.

By a cell (i,j) we mean an ordered index pair with row (machine type $i \epsilon I$ and column (product type) $j \epsilon J$. We will use a tableau similar to that of the ordinary transportation problem. A cell of the tableau (matrix) contains the following information.



e_{ij}: in the north-west corner

x_{ij}: in the north-east corner (0 if left blank)

c_{ij}: in the center; if circled it implies cell (i,j) is in the basis

ℓ_{ij}: in the southwest corner (if $\ell_{ij} = 0$ then it is omitted; if $\ell_{ij} > 0$ the transformation $x'_{ij} = x_{ij} - \ell_{ij}$ should be made as an initial step.)

U_{ij}: in the southeast corner = N if no upper bound is specified.

The following two propositions are valid.

Proposition 1. The optimal solution will be unaltered if a constant cost δ_j $(j=1,2,\ldots,n)$ is subtracted from (or added to) all the cost elements of a specified column, say $j = k$. $(j\epsilon J')$.

Proof. Let the new costs be $c'_{ik} = c_{ik} - \delta_k$ (for kth col.)

so that $c'_{ij} = c_{ij} \quad j \neq k; \quad i\epsilon I$

Then $$\sum_{i\epsilon I} \sum_{j\epsilon J} c'_{ij}x_{ij} = \sum_{i\epsilon I} \sum_{j\epsilon J} c_{ij}x_{ij} - \sum_{i\epsilon I} \delta_k x_{ik}$$

$$= \sum_{i\epsilon I} \sum_{j\epsilon J} c_{ij}x_{ij} - \delta_k \sum_{i\epsilon I} x_{ik}$$

$$= \sum_{i\epsilon I} \sum_{j\epsilon J} c_{ij}x_{ij} - b_k\delta_k .$$

Since $b_k\delta_k$ is constant, the optimal solution x_{ij} of problem (P) given by equations (5)-(8) will be optimal if c_{ij} is replaced by c'_{ij} $(i\epsilon I,\ j\epsilon J)$ in equation (5); also the optimal solution found with costs c'_{ij} is also optimal with costs c_{ij}.

PROPOSITION 2. The optimal solution is unaltered if a cost $\delta_i e_{ij}$ is subtracted from (or added to) all the cost elements c_{ij} of a specified row $i = h$, $i \epsilon I'$.

PROOF. Let $c'_{hj} = c_{hj} - e_{hj}\delta_h$ (for hth row) and $c'_{ij} = c_{ij}$ for $i \neq h$ and $j \epsilon J$.

Then $\sum_{i \epsilon I} \sum_{j \epsilon J} c'_{ij} x_{ij}$

$$= \sum_{i \epsilon I} \sum_{j \epsilon J} c_{ij} x_{ij} - \delta_h \sum_{j \epsilon J} e_{hj} x_{hj}$$

$$= \sum_{i \epsilon I} \sum_{j \epsilon J} c_{ij} x_{ij} - \delta_h a_h .$$

Since $\delta_h a_h$ is constant, the optimal solution of problem (P) given by equations (5)-(8) will be the same with costs being either c_{ij} or c'_{ij} ($i \epsilon I$, $j \epsilon J'$) in equation (5).

We now discuss some basic concepts of graph theory. In what follows a <u>line</u> refers to a row or column of the tableau.

DEFINITION 1. Let Ω denote a collection of cells (i,j) of the tableau. <u>Line</u> g is connected to <u>line</u> h in Ω, if and only if there exists a <u>path</u> (sequence) S of distinct cells belonging to Ω

(9) $S = \{(i_1,j_1), (i_2,j_2) \ldots. (i_k,j_k)\}$ such that

(a) (i_1,j_1) is a cell of S in line g, and (i_k,j_k) is a cell of S in line h;

(b) for every $1 < t < k$, either $i_{t-1} = i_t \neq i_{t+1}$ <u>and</u> $j_{t-1} \neq j_t = j_{t+1}$, or $i_{t-1} \neq i_t = i_{t+1}$ and $j_{t-1} = j_t \neq j_{t+1}$

(c) If $j_t = n+1$, then $t=k$ or $t=1$ but not both;

(d) If the number of distinct lines between g and h, excluding g and h in the path S is r, then the length of the path connecting lines g and h is $\ell(g,h) = r+1$. If no path exists then $\ell(g,h) = \infty$.

REMARK 3. The above definition ensures a sequence of cells forming an alternating path, alternating between columns and rows or vice versa along connected lines.

DEFINITION 2. A cycle of order ℓ is a path C containing 2ℓ cells $(\ell \geq 2)$ such that

(i) Every line of the matrix contains either 0 or 2 cells of C;

(ii) $C = S_\alpha \cup S_\beta$ where each set S_α and S_β contains ℓ cells such that

(a) if a line has two cells of C, then one belongs to S_α and the other to S_β and

(b) $\alpha = \prod_{(i,j)\in S_\alpha} e_{ij} > \prod_{(i,j)\in S_\beta} e_{ij} = \beta$.

DEFINITION 3. The circulation factor of a cycle C is $\rho = \alpha/(\alpha-\beta)$.

REMARK 4. Sometimes, while pivoting, a closed path is created which satisfies all the requirements of Definition 2, except (ii) (b); i.e., we have $\alpha = \beta$ for the closed path. In this case it is possible to alter x_{ij} around the path and remove a cell thus breaking the path. (See the discussion of symmetric loops in Balas and Ivanescu [6], or Eisemann [10].) Moreover, no slack cell, i.e. a cell in column (n+1), can be a member of a cycle by condition (c) of Definition 1.

DEFINITION 4. A loop is defined to be a slack cell (i,n+1). We define the circulation factor of a loop to be infinity.

DEFINITION 5. Let Ω be a set of cells. The span $S(\Omega)$ is the set of all lines excluding column (n+1) that contain at least one cell in Ω.

We say that Ω is a connected set if every pair of lines in the span $S(\Omega)$ is connected.

DEFINITION 6. A one-tree is a maximal connected set of cells such that either

(i) it is acyclic (does not contain a cycle) and there is exactly one cell from column n+1; i.e., it contains a loop, or

(ii) it contains no cell from column n+1, and has a unique cycle.

DEFINITION 7. A basis B for problem P is a one-forest; a set of (m+n+1) cells (called basis cells) which is the union of one-trees T_j, $j=1,\dots,k$ $(2 \leq k < m+n+1)$ such that

(i) $S(T_i) \cap S(T_j) = \emptyset$ if $i \neq j$ and

(ii) $\bigcup_{i=1}^{k} S(T_i) = I \cup J'$.

REMARK 5. The m+n+1 cells forming a basis are identified in the tableau (Fig. 3) by those cells where c_{ij}'s are circled.

DEFINITION 8. A basic solution $X = \{x_{ij}\}$ corresponding to a basis B satisfies (6) and (7) and is such that $(i,j) \notin B$ implies that either $x_{ij} = 0$ or $x_{ij} = U_{ij}$. If in addition (8) is satisfied for $(i,j) \in B$, the basic solution is called primal feasible. We define LB and UB as the sets of nonbasic variables that are at their lower or upper bounds respectively. We define (B, LB, UB) as a basis structure. It is known [6, 8] that given a problem and its basis structure, the associated primal solution is unique.

DEFINITION 9. A solution is said to be primal non-degenerate if $0 < x_{ij} < U_{ij}$ for $(i,j) \in B$.

DEFINITION 10. The cell (m+1, n+1) is called the absorbing cell.

REMARK 6. It is shown by Eisemann [10] that the absorbing cell is always in the basis.

EXAMPLE. Consider a machine loading problem with the data given in a tableau form of Fig. 1. In this, there are 3 (m) machine types and 4 (n) product types. The rows correspond to machine types with available time units of 300, 225 and 140 respectively. Product types correspond to columns and they are required in an amount of 170, 60, 35 and 60 respectively. The utilization of machine type i for product j requires e_{ij} time units per unit of product and is given in the northwest corner of the cell (i,j). The cost c_{ij}, the lower bound ℓ_{ij} and upper bound U_{ij} are given in the center, southwest corner and southeast corner respectively by the cell (i,j). These are considered as the <u>data</u> of the problem. Figure 2 gives the same problem of Fig. 1 but with a slack column 5 with $e_{i5} = 1$, $c_{i5} = 0 = \ell_{i5}$ and $U_{i4} = N$ a large positive number for $i=1,2,3$ and an extra machine type, (row 4) introduced where $e_{4j} = 1$; $\ell_{4j} = 0$, $U_{4j} = N$ and $c_{4j} = 100$, a relatively large positive cost for $j = 1$ to 4. A large positive time unit, 1000, is chosen for a_4. No value for b_5 is assigned since there is no constraint corresponding to the slack column.

Solution techniques for solving problem P are discussed in Balas and Ivanescu [6], Eisemann [10] and in other text books [8, 9, 13]. However the following algorithm for finding an initial primal feasible solution has been found to be efficient.

<u>Algorithm A1</u>. For finding an initial feasible solution $X = \{x_{ij}\}$ and an initial basis structure (B, LB, UB).

(0) Start with all $x_{ij} = 0$ and all cells $(i,j) \in LB$.
Let $SR = I$ and $SC = J'$. Let $j = 1$.

(1) If $SC = \emptyset$ go to 7. Else go to 2.

(2) If $j = n+1$ set $j = 1$; if $j \notin SC$ set $j = j+1$ and go to 2.
Else go to 3.

tableau form of Fig. 1. In this, there are 3 (m) machine types and 4 (n) product types. The rows correspond to machine types with available time units of 300, 225 and 140 respectively. Product types correspond to columns and they are required in an amount of 170, 60, 35 and 60 respectively. The utilization of machine type i for product j requires e_{ij} time units per unit of product and is given in the northwest corner of the cell (i,j). The cost c_{ij}, the lower bound ℓ_{ij} and upper bound U_{ij} are given in the center, southwest corner and southeast corner respectively by the cell (i,j). These are considered as the data of the problem. Figure 2 gives the same problem of Fig. 1 but with a slack column 5 with $e_{i5} = 1$, $c_{i5} = 0 = \ell_{i5}$ and $U_{i4} = N$ a large positive number for i=1,2,3 and an extra machine type, (row 4) introduced where $e_{4j} = 1$; $\ell_{4j} = 0$, $U_{4j} = N$ and $c_{4j} = 100$, a relatively large positive cost for j = 1 to 4. A large positive time unit, 1000, is chosen for a_4. No value for b_5 is assigned since there is no constraint corresponding to the slack column.

Solution techniques for solving problem P are discussed in Balas and Ivanescu [6], Eisemann [10] and in other text books [8, 9, 13]. However the following algorithm for finding an initial primal feasible solution has been found to be efficient.

Algorithm A1. For finding an initial feasible solution $X = \{x_{ij}\}$ and an initial basis structure (B, LB, UB).

(0) Start with all $x_{ij} = 0$ and all cells $(i,j) \in LB$.
Let SR = I and SC = J'. Let j = 1.

(1) If SC = $\emptyset$ go to 7. Else go to 2.

(2) If j = n+1 set j = 1; if $j \notin SC$ set j = j+1 and go to 2.
Else go to 3.

(3) Find h such that $c_{hj} = \min_{i \in SR} c_{ij}$ and $x_{ij} < U_{ij}$. Let

$$x_{hj} = \min \left\{ \frac{a_h}{e_{hj}}, b_j, U_{hj} \right\}.$$

(4) Let $a_h = a_h - e_{hj}x_{hj}$ and $b_j = b_j - x_{hj}$. If $a_h = b_j = 0$ go to 6. Else go to 5.

(5) (a) If $b_j = 0$, let $SC = SC - \{j\}$; $(h,j) \in B$, $j = j+1$. Go to 1.

(b) If $a_h = 0$; let $SR = SR - \{h\}$; $(h,j) \in B$ and $j = j+1$. Go to 1.

(c) If $x_{hj} = U_{hj}$; let $(h,j) \in UB, j = j+1$ and go to 1.

(6) Here $a_h = b_j = 0$. If $|SR| < 2$ (number of elements in set SR) let $SC = SC - \{j\}$, $(h,j) \in B$, $j = j+1$, and go to 1. Else (i.e. $|SR| \geq 2$), find $|SC|$. If $|SC| \geq 2$, let $SC = SC - \{j\}$, $(h,j) \in B$, $j = j+1$, and go to 1. Else (i.e. $|SC| < 2$) let $SR = SR - \{h\}$, $(h,j) \in B$, $j = j+1$, and go to 1.

(7) For each $i \in SR$, set $x_{i,n+1} = a_i$ and $(i,n+1) \in B$. STOP.

Figure 3 gives a starting primal feasible solution for the problem given in Figure 2. (Ignore, for the present the numbers given as row and column headings.) This solution is obtained by applying algorithm A1. The following give the steps of A1 and the results due to application of these steps sequentially.

<u>Step No.</u> Result due to this step.

(0) $x_{ij} = 0$ and $(i,j) \in LB$ for all (i,j).

$SR = \{1,2,3,4\}$; $SC = \{1,2,3,4\}$. $j = 1$.

Step No.	Result due to this step.
(2)	$j \in SC$. Go to 3.
(3)	c_{31} is minimum. $x_{31} = 35 = a_3/e_{31}$.
(4)	$a_3 = 0$; $b_1 = 135$.
(5)	$SR = SR - \{3\}$; $j = 2$; $(3,1) \in B$. Go to 1.
(3)	c_{22} is minimum and $x_{22} = 60 = b_2$.
(4)	$a_2 = 45$; $b_2 = 0$.
(5)	$SC = SC - \{2\}$. $j = 3$; $(2,2) \in B$, go to 1
(3)	c_{13} is minimum. $x_{13} = 35 = b_3$.
(4)	$a_1 = 230$; $b_3 = 0$.
(6)	$SC = SC - \{3\}$; $j = 4$; $(1,3) \in B$, go to 1
(3)	c_{14} is minimum. $x_{14} = 60 = b_4$.
(4)	$a_1 = 50$; $b_4 = 0$.
(5)	$SC = SC - \{4\}$; $j = 5$; $(1,4) \in B$; go to 1.
(2)	$j = 5$ so $j = 1$; go to 3.
(3)	c_{11} is minimum. $x_{11} = U_{11} = 15$.
(4)	$a_1 = 5$; $b_1 = 120$.
(5)	$(1,1) \in UB$; $j = 2$, go to 1.
(2)	$j = 2$; $j = 3$; $j = 4$; $j = 5$; so $j = 1$. Go to 3.
(3)	$x_{21} = 45 = a_2/e_{21}$.
(4)	$a_2 = 0$; $b_1 = 75$.
(5)	$SR = SR - \{2\}$, $j = 2$; $(2,1) \in B$; go to 1.
(2)	$j = 2$; $j = 3$; $j = 4$; $j = 5$ so $j = 1$. Go to 3.
(3)	c_{41} is minimum. $x_{41} = 75 = b_1$.
(4)	$a_4 = 925$; $b_1 = 0$.
(5)	$SC = SC - \{1\} = \emptyset$; $j = 2$; $(4,1) \in B$; go to 1.
(1)	Go to 7.
(7)	$x_{1,5} = 5$; $x_{4,5} = 925$; $(1,5)$, $(4,5) \in B$. STOP.

The circled c_{ij} in Fig. 3 indicate that cell $(i,j) \in B$. x_{ij} is written in the northeast corner of cell (i,j). It can be verified that this basis structure is primal feasible.

The following theorem has been proved. [6, 9, 10].

Theorem 1. The simplex algorithm for a nondegenerate generalized transportation problem involves at each pivot step a primal feasible solution $X = \{x_{ij}\}$ and a basis structure (B, LB, UB) such that

(a) $(i,j) \in B$ then $0 < x_{ij} < U_{ij}$.

(b) $(i,j) \notin B$ then either

(i) $x_{ij} = 0$ and $(i,j) \in LB$ or

(ii) $x_{ij} = U_{ij}$ and $(i,j) \in UB$.

Define a maximal connected subset of a basis as a component. It has been proved that the basis of any basic solution to a generalized transportation problem consists of mutually disconnected components where each component is a one-tree [6, 10]. Thus the basis is a one-forest.

From the algorithm for starting solution, it is easy to see that the starting solution is a one-forest consisting of one-trees with loops. Since choice of a_{m+1} is arbitrary, $x_{m+1,n+1}$ will 'absorb' any possible imbalances either in the rows or columns.

Before proceeding further, let us present the graph associated with a one-forest corresponding to a basis B.

For a general reference to graph theory see C. Berge [7] or F. Harary [18]. For a discussion of the tree-index labelling method, definitions, terminologies and algorithms for the ordinary transportation problem, see V. Srinivasan and G. L. Thompson [23] and S. Glicksman, L. Johnson and L. Eselson [14].

The graph $G = (V,E)$ of a basic solution consists of the nodes $V = I \cup J'$ and the edges $E = B$ where B is the set of basis cells. An edge $(i,n+1)$ is a loop connecting to itself. It was seen from definition 7, that the graph $G = (V,E) = (I \cup J',B) = \bigcup_{h=1}^{k} G(T_h)$, is a 'one-forest', where the sub-graph $G(T_h) = (V_h,E_h) = \{I_h \cup J_h', B_h\}$. The graph $G(T_h)$ has certain properties. Each sub-graph $G(T_h)$ has either a cycle C_h or a loop (slack cell), but not both. It has two classes of nodes I_h and J_h' and every path in the sub-graph T_h makes alternate use of one, then the other kind of node.

DEFINITION 11. A <u>rooted-one-tree</u> $G(T_h) = \{I_h \cup J_h', B_h\}$ is a one-tree with a <u>distance function</u> $d_h(v)$ for $v \in V_h$ satisfying conditions (a) to (d) given below:

Choose an edge (r,s) as follows:

(i) if there is a loop $C_h = \{(r,n+1)\}$ then let $(r,s) = (r,n+1)$ or

(ii) if there is a cycle C_h, let (r,s) be any edge (basic cell) in C_h such that $(r,s) \in S_{\alpha h}$ (see Definition 2); then

(a) $G_h = (I_h \cup J_h', B_h - \{r,s\})$ is a <u>tree</u>.

(b) r is the root of G_h.

(c) $d_h(v) = 0$ for all $v \in S(C_h)$.

(d) if u is any node in $G_h - C_h$ then $d_h(u)$ is defined as

$$d_h(u) = \min_{v \in S(C_h)} \ell(v,u);$$

In $G_h \in G(T_h)$ there is a unique path from the root r to any other distinct node $v \in G_h$, since G_h is a tree. Associated with a <u>rooted-one-tree</u> $G(T_h)$, the following <u>predecessor function</u> $p_h(u)$ for any node $u \in I_h \cup J_h'$ is defined.

DEFINITION 12. The predecessor function $p_h(u)$ for every node $u \epsilon I_h \cup J_h'$ satisfies the following:

(a) $p_h(r) = \emptyset$ where r is the root of G_h

(b) $p_h(v) = u$ if $(u,v) \epsilon E_h$ and r connected to u in $B_h - \{(r,s)\} - \{(u,v)\}$.

The following algorithm determines the distance and predecessor function in the tree $G_h = (I_h \cup J_h', B_h - \{(r,s)\})$ for the starting* solution.

ALGORITHM A2. Algorithm for finding root, distance and predecessor functions for the initial* basis produced by algorithm A1.

(0) Set $r = 1$; $i = 0$.

(1) If $r > m+1$, STOP. Else go to (2).

(2) Check to see if $(r,n+1) \epsilon B$. If yes go to (3). If no, $r = r+1$. Go to (1).

(3) Set $i = i+1$. Let r be the root of T_i; $p_i(r) = \emptyset$; $d_i(r) = 0$; $M_r = \{r\}$; $M_c = \emptyset$. $S_r = S_c = \emptyset$.

(4) For each $u \epsilon M_r$

  (a) find the set $D_c(u) = \{v | v \epsilon J'-S_c \text{ and } (u,v) \epsilon B_h\}$

  (b) for each $v \epsilon D_c(u)$ let $d_i(v) = d_i(u) + 1$ and $p_i(v) = u$.

(5) Replace S_r by $S_r \cup M_r$ and M_c by $\bigcup_{u \epsilon M_r} D_c(u)$.

(6) If $M_c = \emptyset$, set $r = r+1$ and go to (1). Else go to (7).

(7) For each $v \epsilon M_c$

  (a) find the set $D_r(v) = \{u | u \epsilon I-S_r \text{ and } (u,v) \epsilon B_h\}$

* This algorithm A2 is only for the initial basis. For any general basis an algorithm similar to A2 (but long) is given in [4]. Further, later on [1] we will define $p_h(r) = s$ and $\neq \emptyset$ where (r,s) is given in Definition 11.

(b) for each $u \in D_r(u)$ let $d_i(u) = d_i(v) + 1$ and $p_i(u) = v$.

(8) Replace S_c by $S_c \cup M_c$ and M_r by $\bigcup_{v \in M_c} D_r(v)$.

(9) If $M_r = \emptyset$, set $r = r+1$ and go to (1). Else go to (4).

EXAMPLE (continued): Let us now consider how this algorithm A2 is used for the starting solution given as Figure 3 to obtain the root, distance and predecessor function for the initial basis:

Steps of A2	Result due to this step of A2
(0)	$r = 1$; $i = 0$; $n+1 = 5$
(2)	$(1,5) \in B$
(3)	$i = 1$; row 1 is the root of one tree T_1
	$p_1(\text{row } 1) = \emptyset$; $d_1(\text{row } 1) = 0$; $M_r = \{1\}$; $M_c = \emptyset$.
	$S_r = S_c = \emptyset$.
(4)	(a) $D_c(1) = $ (columns 3 and 4)
	(b) $d_1(\text{col. } 3) = 1$; $d_1(\text{col. } 4) = 1$;
	$p_1(\text{col.} 3) = p_1(\text{col. } 4) = \text{row } 1$
(5)	$S_r = \{\text{row } 1\}$; $M_c = $ (col. 3, col. 4)
(7)	$D_r(\text{col. } 3) = D_r(\text{col. } 4) = \emptyset$
(8)	$S_c = $ (col. 3, col. 4); $M_r = \emptyset$
(9)	$r = 2$.
(1)	2, 3, 4 are not greater than 4
(2)	$(4,5) \in B$
(3)	$i = 2$; Row 4 is the root of one tree T_2
	$p_2(\text{row } 4) = \emptyset$; $d_2(\text{row } 4) = 0$; $M_r = \{r\}$; $M_c = \emptyset$.
	$S_r = S_c = \emptyset$.
(4)	(a) $D_c(\text{row } 4) = $ (col. 1)
	(b) $d_2(\text{col. } 1) = 1$; $p_2(\text{col. } 1) = \text{row } 4$

Steps of A2	Result due to this step of A2
(5)	$S_r = \{\text{row } 4\}$; $M_c = (\text{col. } 1)$
(7)	(a) $D_r(\text{col. } 1) = (\text{row } 2, \text{row } 3)$
	(b) $d_2(\text{row } 2) = d_2(\text{row } 3) = 2$; $p_2(\text{row } 2)$ $= p_2(\text{row } 3) = \text{col. } 1$
(8)	$S_c = (\text{col. } 1)$ and $M_r = (\text{row } 2, \text{row } 3)$
(4)	(a) $D_c(\text{row } 2) = \text{col. } 2$ $D_c(\text{row } 3) = \emptyset$
	(b) $d_2(\text{col. } 2) = 3$ and $p_2(\text{col. } 2) = \text{row } 2$
(5)	$S_r = (\text{rows } 2, 3, 4)$ and $M_c = \text{col. } 2$
(7)	$D_r(\text{col. } 2) = \emptyset$
(8)	$S_c = (\text{col. } 2)$; $M_r = \emptyset$
(9)	$M_r = \emptyset$; $r = 5$
(1)	$5 > 4$: STOP.

The starting solution consists of a basis whose graph is a one-forest consisting of one-trees each having a loop. Since a loop has just one cell $(r, n+1)$, the specification of the root of the 1-tree is obvious, viz r. However it will be seen that when a new cell enters the basis a <u>two-tree</u> is created which may contain a cycle C_h of 2ℓ ($\ell \geq 2$) cells. It is necessary to identify the two sets $S_{\alpha h}$, $S_{\beta h}$ and determine its circulation factor for finding the new basis. The following algorithm provides this.

ALGORITHM A3. Algorithm for determining the sets $S_{\alpha h}$, $S_{\beta h}$ associated with a cycle C_h and calculation of ρ_h.

(0) Choose any edge $(r,s) \in C_h$. Store s; we now construct a tree with r being the <u>root</u>. Set $A_\alpha = e_{rs}$; $A_\beta = 1$; $S_{\alpha h} = \{(r,s)\}$; $S_{\beta h} = \emptyset$ and $v = s$.

(1) Let $u = p(v)$; if $u \neq r$ go to 2 and if $u = r$ go to 4.

(2) $S_{\beta h} = S_{\beta h} \cup \{(u,v)\}$; $A_\beta = A_\beta * e_{uv}$

(3) Let $v = p(u)$; $S_{\alpha h} = S_{\alpha h} \cup \{(u,v)\}$;

$A_\alpha = A_\alpha * e_{uv}$ and go to 1.

(4) $A_\beta = A_\beta * e_{rv}$. $S_{\beta h} = S_{\beta h} \cup \{(r,v)\}$. If $A_\alpha = A_\beta$ go to 6. If $A_\alpha < A_\beta$ interchange $S_{\alpha h}$ and $S_{\beta h}$; also A_α and A_β. Set $s = v$ in the stored edge (r,s). Go to 5.

(5) $\rho_h = A_\alpha/(A_\alpha - A_\beta)$. STOP.

(6) This is a symmetric closed loop which can be eliminated by an ordinary pivot step, thus changing the two-tree into a one-tree, see Remark 4.

EXAMPLE (continued): Let us now apply algorithm (A3) for the cycle $\{(2,1), (2,2), (3,1), (3,2)\}$ given in Figure 4 to find S_α, S_β and ρ. (Subscript h is dropped for convenience here.)

Let (2,1) be the edge chosen. Column 1 is stored; Row 2 is the root; $A_\alpha = e_{21} = 1$; $A_\beta \equiv 1$. $S_\alpha = \{(2,1)\}$; $S_\beta = \emptyset$; col. v = col. 1.

<u>Step</u>	<u>Result due to this step of A3</u>
(1)	row 3 = p(col. 1); row 3 $\neq$ row 2
(2)	$S_\beta = (3,1)$; $A_\beta = 1 \times e_{31} = 4$
(3)	col. 2 = p(row 3); $S_\alpha = \{(2,1), (3,2)\}$ $A_\alpha = A_\alpha \times e_{32} = 2$.
(1)	row 2 = p(col. 2); row 2 = row 2
(4)	$A_\beta = A_\beta \times e_{22} = 4 \times 3 = 12$; $S_\beta = \{(3,1), (2,2)\}$ Since $A_\alpha = 2 < A_\beta = 12$ interchange S_α and S_β and s = col. 2 and thus (2,2) is the edge removed. Thus $S_\alpha = \{(3,1), (2,2)\}$ $S_\beta = \{(2,1), (3,2)\}$ and $\rho = A_\alpha/(A_\alpha - A_\beta) = 12/10 = 6/5$. STOP.

REMARK 7. Let $T_1, T_2, \ldots, T_h, . T_k$ be the one-trees of a basis B and let $C_1, C_2, \ldots, C_h, \ldots, C_k$ be their respective cycles or loops such that $C_h \subsetneq T_h$. Consider a specific one-tree T_h, and the rooted one-tree $G(T_h) = (I_h \cup J_h', B_h)$ with $(r,s) \in S_{\alpha h}$ being the edge removed from B_h to form the tree G_h (Definition 11). The distance function $d_h(v)$ and predecessor function $p_h(v)$ for $v \in I_h \cup J_h'$ are known from algorithm A2. Let $|I_h|$ and $|J_h'|$ represent the number of elements in I_h and J_h' respectively. It follows from Definition 7 that $I = \bigcup_{h=1}^{k} I_h$ and $J' = \bigcup_{h=1}^{k} J_h'$ with $|I| = m+1$ and $|J'| = n$. Moreover $I_h \cap I_t = J_h' \cap J_t' = \emptyset$ for $h \neq t$.

Let (e,f) be a cell not in B. Then (e,f) is either in LB or in UB. The basic pivot step of the simplex method, or of the operator theory to be developed here, involves inserting (e,f) into B. When (e,f) is added to B two cases arise:

(a) $e \in I_h$ and $f \in J_t'$ where $h \neq t$;

(b) $e \in I_h$ and $f \in J_t'$ with $h = t$.

Since $e \in I_h$ in either case $p_h(e) = j$ is a column which is connected in T_h to C_h. Let D_h consist of the cell (e,f) together with the unique path in T_h that connects j to u where $u \in S(C_h)$ and $\ell(u,j)$ is minimum. (If $j = u$ then $D_h = \{(e,f)\}$). We call D_h the row path.

Similarly, since $f \in J_t$ in either case, $p_t(f) = i$ is a row which is connected in T_t to C_t. Let D_t consist of the cell (e,f) together with the unique path in T_t that connects i to v where $v \in S(C_t)$ and $\ell(i,v)$ is minimum. (If $i = v$ then $D_t = \{(e,f)\}$). We call D_t the column path.

Hence, due to the introduction of the cell $(e,f) \notin B$, into the basis, a set of cells Γ can be determined, where

(10) $\Gamma = (C_h \cup D_h) \cup (D_t \cup C_t)$.

In this, $(C_h \cup D_h)$ is called the <u>row segment</u> of Γ while $(D_t \cup C_t)$ is called the <u>column segment</u> of Γ. The graph associated with the edges given by Γ and the nodes in the span of the cells of $\Gamma(S(\Gamma))$ is called a <u>two-tree</u>.

It has been proved that this graph contains exactly two cycles (lemma 9 of [6]). It is possible that both the row segment and the column segment are identical.

Let us consider changes in x_{ef}. It can be shown [6, 10] that such changes can be compensated for by changing only x_{ij} for $(i,j) \in \Gamma$. In the next algorithm (A4), we shall assume that $(e,f) \in LB$, (the alternate case of $(e,f) \in UB$ is covered in Remark), so that initially $x_{ef} = 0$ and we want to make it positive.

DEFINITION 13. If x_{ef} is changed, then Γ can be partitioned into two sets Γ_1 and Γ_2 such that x_{ij} must be increased by an amount μm_{ij} for $(i,j) \in \Gamma_1$ and x_{ij} must be decreased by an amount μm_{ij} for $(i,j) \in \Gamma_2$.

The following algorithm (A4) determines Γ_1 and Γ_2, the cell $(e,f) \in \Gamma$ to leave the basis, and also finds the coefficients m_{ij} by which the x_{ij}'s must be revised for $(i,j) \in \Gamma$, for the case when $(e,f) \in LB$.

ALGORITHM A4. For finding m_{uv}, Γ_1, Γ_2, μ, x_{uv} for $(u,v) \in \Gamma$ and the <u>outgoing cell</u> (e_1,f_1) when cell $(e,f) \in LB$ is added to B. (If $(e,f) \in UB$ make the changes indicated in Remark 8.)

(0) Let $m_{uv} = 0$ for $(u,v) \in \Gamma$. Let $(e,f) \in \Gamma_1$, $x = 1$, and $m_{ef} = +1$. Let $(u,v) = (e,f)$. Set $Fl = 0$ indicating we are evaluating a row segment. Let the span of the cycle $S(C) = SR \cup SC$.

(1) If $u \in SR$, i.e., $d(u) = 0$ go to (5), else go to (2).

(2) Let $j = p(u)$. Let $x = -x \dfrac{e_{uv}}{e_{uj}}$, $m_{uj} = m_{uj} + x$. If $m_{uh} \geq 0$, let $(u,j) \in \Gamma_1$, else $(u,j) \in \Gamma_2$. Let $v = j$.

(3) If $v \in SC$, i.e., $d(v) = 0$, go to (8); else go to (4).

(4) Let $i = p(v)$. Let $x = -x$, $m_{iv} = m_{iv} + x$; if $m_{iv} \geq 0$ let $(i,v) \in \Gamma_1$; else $(i,v) \in \Gamma_2$. Let $u = i$. Go to (1).

(5) Let $j = p(u)$. If $j = n+1$, let $m_{u,n+1} = m_{u,n+1} - e_{uv}m_{uv}$. If $F1 = 1$ go to (11). If $F1 = 0$ let $F1 = 1$, $(u,v) = (e,f)$, $x = 1$ and go to (3). If $j \neq n+1$, (if $j = \emptyset$, $j = s$) let $x = -x\rho \dfrac{e_{uv}}{e_{uj}}$, $m_{uj} = m_{uj} + x$. If $m_{uj} \geq 0$, let $(u,j) \in \Gamma_1$; else $(u,v) \in \Gamma_2$. Let $SC = SC - \{j\}$ and $v = j$.

(6) If $SR = \emptyset$ and $F1 = 0$, let $(u,v) = (e,f)$, $x = 1$, $F1 = 1$, go to (3); if $SR = \emptyset$ and $F1 = 1$ go to (11). Else let $i = p(v)$; let $x = -x$, $m_{iv} = m_{iv} + x$. If $m_{iv} \geq 0$ let $(i,v) \in \Gamma_1$, else $(i,v) \in \Gamma_2$. Let $SR = SR - \{i\}$ and $u = i$.

(7) If $SC = \emptyset$ and $F1 = 0$, let $F1 = 1$, let $(u,v) = (e,f)$, $x = 1$, go to (3); if $SC = \emptyset$ and $F1 = 1$ go to (11). Else let $j = p(u)$; if $j = \emptyset$, let $j = s$; let $x = -x\dfrac{e_{uv}}{e_{uj}}$, $m_{uj} = m_{uj} + x$. Let $SC = SC - \{v\}$ and $v = j$. Go to (6).

(8) Let $i = p(v)$. Let $x = -x\rho$ and $m_{iv} = m_{iv} + x$. If $m_{iv} \geq 0$ let $(i,v) \in \Gamma_1$; else $(i,v) \in \Gamma_2$. Let $SR = SR - \{i\}$. Let $u = i$.

(9) If $SR = \emptyset$ and $F1 = 0$, let $(u,v) = (e,f)$; $x = 1$, and $F1 = 1$, go to (3); if $SR = \emptyset$ and $F1 = 1$ go to (11). Else let $j = p(u)$; if $j = \emptyset$, let $(u,j) \in \Gamma_1$; else let $(u,j) \in \Gamma_2$. Let $SC = SC - \{j\}$ and $v = j$.

(10) Let $SR = \emptyset$ and $F1 = 0$, let $(u,c) = (e,f)$, $x = 1$ and $F1 = 1$, go to (3); if $SR = \emptyset$ and $F1 = 1$, go to (11). Else let $i = p(v)$; let $x = -x$ and $m_{iv} = m_{iv} + x$. If $m_{iv} \geq 0$, let $(i,v) \in \Gamma_1$; else let $(i,v) \in \Gamma_2$. Let $SR = SR - \{i\}$. Let $u = i$. Go to (9).

(11) Find μ from

$$\mu = \min \begin{cases} (U_{ij} - x_{ij})/m_{ij} & \text{for } (i,j) \in \Gamma_1 \\ x_{ij}/(-m_{ij}) & \text{for } (i,j) \in \Gamma_2 \end{cases}$$

The cell on which this minimum is taken is the <u>leaving cell</u> (e_1,f_1). If $(e_1,f_1) \in \Gamma_1$, $(e_1,f_1) \in UB$, and if $(e_1,f_1) \in \Gamma_2$, $(e_1,f_1) \in LB$. For all $(i,j) \in \Gamma$, let

(12) $x_{ij} = x_{ij} + \mu\, m_{ij}$.

REMARK 8. The algorithm A4, given above provides the basic pivot step of the generalized transportation problem under the assumption that (e,f) was in LB. If (e,f) was in LB then, in the starting step (0) of algorithm A4 make the following changes: "Let $(e,f) \in \Gamma_2$ and $x = -1$" and replace "$x = 1$" by "$x = -1$" in steps (5) - (7), (9), (10), and continue with the rest of the algorithm.

Due to this basic pivot step, the basis structure is changed, since $(e,f) \notin B$ is now in B and $(e_1,f_1) \in B$ is now not contained in B. However, it was shown [6, 10] that the resultant basis is also another one-forest. This fact is given as a theorem, the proof of which is given in [6].

THEOREM 2. The <u>pivot step</u> (Algorithm A4) of the generalized transportation problem preserves the one-forest property of the basis B.

EXAMPLE. (continued) Consider Fig. 3 for which when Algorithm A2 was applied, the distance and predecessor function were obtained. Assume that the cell (3,2) is to enter the basis. Since both row 3, and column 2 are contained in the one-tree consisting of cells $\{(4,5), (4,1), (3,1), (2,1), (2,2)\}$ we will apply algorithm A4 to the set Γ, which is a two-tree, obtained by the introduction of cell (3,2) into the basis. Figure 5 gives the graph of Γ and the distance function.

The following gives the steps of algorithm A4 and the results due to execution of the corresponding step.

Step No.	Result due to this step of A4
(0)	$(u,v) = (3,2)$; $m_{32} = 1$; $F1 = 0$; $(3,2) \in \Gamma_1$; $SR = \{4\}$; $SC = \{5\}$
(1)	$d(R_3) = 2 \neq 0$. Go to (2).
(2)	$C_1 = p(R_3)$; $m_{31} = 0 - 1 \times (2/4) = -1/2$; $(3,1) \in \Gamma_2$
(3)	$d(C_1) = 1 \neq 0$. Go to (4).
(4)	$R_4 = p(C_1)$; $m_{41} = 0 - (-1/2) = 1/2$; $(4,1) \in \Gamma_1$. Go to (1).
(1)	$R_4 \in SR$. Go to (5).
(5)	$C_5 = p(R_4)$; (4,1) is the terminal edge and (4,5) the entry edge of row segment $m_{45} = 0 - 1/2 = -1/2$ and $(4,5) \in \Gamma_2$. $F1 = 1$ $(u,v) = (3,2)$. Go to (3).
(3)	$d(C_2) = 3 \neq 0$. Go to (4).
(4)	$R_2 = p(C_2)$; $m_{22} = 0 - 1 = -1$; $(2,2) \in \Gamma_2$. Go to (1).
(1)	$d(R_2) = 2 \neq 0$. Go to (2).
(2)	$C_1 = p(R_2)$; $m_{21} = 0 - \frac{(-1)-3}{1} = 3$; $(2,1) \in \Gamma_1$.
(3)	$d(C_1) = 1 \neq 0$. Go to (4).
(4)	$R_4 = p(C_1)$; $m_{41} = 1/2 - 3 = -5/2$; $(4,1) \in \Gamma_2$ (note (4,1) is updated from Γ_1 to Γ_2). Go to (1).
(1)	$R_4 \in S((4,5))$. Go to (5).
(5)	$C_5 = p(R_4)$; (4,1) is terminal edge and (4,5) entry edge for column segment also $m_{45} = -1/2 - (-5/2) = 2$, $(4,5) \in \Gamma_1$. Go to (11). (note that (4,5) is updated from Γ_2 to Γ_1.)

$$(11) \qquad \mu = \min \begin{cases} \infty, & \dfrac{150-45}{3}, & \infty \\ & & \\ \dfrac{60}{1}, & \dfrac{35}{(1/2)}, & \dfrac{75}{(5/2)} \end{cases}$$

$(e,f) = (4,1)$ and $\mu = 30$.

After making indicated changes we have:

$$\left.\begin{aligned} x_{32} &= 30 \\ x_{21} &= 135 \\ x_{45} &= 1000 \end{aligned}\right\} \varepsilon\ \Gamma_1 \qquad \left.\begin{aligned} x_{22} &= 30 \\ x_{31} &= 20 \\ x_{41} &= 0 \rightarrow \text{UB} \end{aligned}\right\} \varepsilon\ \Gamma_2$$

Figure 4 gives the new tableau when these new x_{ij}'s given above for $(i,j) \varepsilon \Gamma$ replace the old x_{ij}'s given in Figure 3. The current basis B, in Fig. 4 consists of cells with c_{ij} circled, viz., {(1,3), (1,4), (1,5), (2,1), (2,2), (3,1), (3,2), (4,5)}. The cell {(1,1)} is in its upper bound and hence is contained in UB. The rest of the cells {(1,2), (2,3), (2,4), (2,5), (3,3), (3,4), (3,5), (4,1), (4,2), (4,3), (4,4)} are at their lower bound. This set of cells at their lower bound constitute LB. There are $m+n+1 = 8$ cells in the basis which always includes the absorbing cell (4,5). This basis form an <u>one-forest</u> consisting of 3 mutually disconnected <u>one-trees</u>. The first one-tree T_1 consists of cells {(1,3), (1,4), (1,5)} with the slack element {(1,5)} = C_1 (loop). The cells (1,3), (1,4) are the <u>branches</u> of T_1. Row 1 is the root of T_1. The second <u>one-tree</u> T_2 consists of cells {(2,1), (2,2), (3,1), (3,2)} which coincides with its cycle C_2; while the third one-tree T_3 consists of only the absorbing cell (4,5). It is easy to see that every cell $(i,j) \varepsilon B$ is an element in only one of the one-trees and there is at least one path between any two lines contained in the span of a specified one-tree.

3. THE DUAL PROBLEM

Let us now define the dual problem to P. Let u_i for $i \epsilon I$, v_j for $j \epsilon J'$ and w_{ij} for $i \epsilon I$, $j \epsilon J'$ be dual variables associated with the row constraints (6), column constraints (7) and upper bound constraints (8) respectively. Then the dual problem (D) is:

(13) Maximize $\sum_{i \epsilon I} a_i u_i + \sum_{j \epsilon J'} b_j v_j - \sum_{i \epsilon I} \sum_{j \epsilon J'} U_{ij} w_{ij} = F.$

Subject to the following constaints:

(14) $e_{ij} u_i + v_j - w_{ij} \leq c_{ij} \qquad i \epsilon I;\ j \epsilon J'$

(15) $w_{ij} \geq 0 \qquad i \epsilon I;\ j \epsilon J'$

(16) $u_i \leq 0 \qquad i \epsilon I.$

Equation (16) can be derived as follows: since $U_{i,n+1} = M$ for $j = n+1$, (14) becomes $e_{i,n+1} u_i \leq c_{i,n+1}$; but $c_{i,n+1} = 0$, $e_{i,n+1} = 1$, so (16) follows.

Given a basis B, one can determine a unique set of solutions for u_i and v_j to the equations

(17) $e_{ij} u_i + v_j = c_{ij}$ for $(i,j) \epsilon B$

so that $d_{ij} = e_{ij} u_i + v_j$ is unique for all (i,j); $i \epsilon I$; $j \epsilon J'$.

Because of (16) $d_{i,n+1} = u_i \leq 0$.

REMARK 9. Unlike the ordinary transportation problem u_i and v_j are unique, because the dual variables are uniquely determined by the one-tree in whose span they lie. Moreover in the optimal solution, if line i contains a basis cell in the slack column then $u_i = 0$. Since the absorbing cell is always in the basis [10], $u_{m+1} = 0$. In addition all other u_i must be nonpositive at the optimal solution, due to (16). Further, if c_{ij} and e_{ij} are all positive, then because of (17) the optimal v_j's will be non-negative.

REMARK 10. Since the one-trees are mutually disconnected in the basis graph $G(B)$, u_i and v_j associated with the span of $S(T_h)$ of T_h depend only on the basis cells contained in T_h. Thus they are independent of the dual variables associated with any other one-tree, T_g where $g \neq h$. Hence, in the basic pivot step, if some one-trees of the basis of preceeding iteration are not changed, the associated u_i and v_j are also not changed.

The following algorithm provides a method of evaluation u_i, $i \in I$ and v_j, $j \in J'$ for the initial solution given by A2.

ALGORITHM A5. For finding the dual variable u_i, v_j associated with initial* basic solution.

(0) Let initial basis B consist of k trees T_h with predecessor and distance functions p_h, d_h for $h = 1,\ldots,k$. Let $h = 1$.

(1) Let r be the root of T_h, i.e., $d_h(r) = 0$. Let $u_r = 0$; $t = 1$.

(2) Find $SC_t = \{j \mid j \in J_h' \text{ and } d_h(j) = t\}$.
If $SC_t = \emptyset$, go to 6.

(3) For each $j \in SC_t$ let $i = p_h(j)$ and let $v_j = c_{ij} - e_{ij}u_i$.
Let $t = t+1$.

(4) Find $SR_t = \{i \mid i \in I_h \text{ and } d_h(i) = t\}$. If $SR_t = \emptyset$, go to 6.

(5) For each $i \in SR_t$, let $j = p_h(i)$ and let $u_i = (c_{ij} - v_j)/e_{ij}$.
Let $t = t+1$. Go to 2.

(6) If $h = k$, STOP. Else let $h = h+1$ and go to 1.

EXAMPLE. (continued): Dual variables u_i and v_j for the initial basis given as Fig. 3 are obtained utilizing the distance and predecessor given by A2. These are shown in Figure 3 as row and column headings.

---

* This algorithm is only for the initial basis. However a similar algorithm (though long) for any general basis is given in [4].

Steps of A5	Result due to this step of A5
(0)	Basis B consists of 2 trees T_1, T_2 with p_h and d_h; $h = 1$; $k = 2$.
(1)	Row 1 is the root of T_1 and $d_1(r_1) = 0$; $u_1 = 0$; $t = 1$.
(2)	$SC_1 = \{C_3, C_4\}$ as $d_1(C_3) = d_1(C_4) = 1$.
(3)	$p_1(C_3) = p_1(C_4) = r_1$; $v_3 = 3 = 0$; $v_4 = 4 = 0$; $t = 2$.
(4)	$SR_t = \emptyset$. Go to (6).
(6)	$h = 2$. Go to (1).
(1)	Row 4 is the root of T_2 as $d_2(r_4) = 0$; $u_1 = 0$; $t = 1$.
(2)	$SC_1 = \{$col. 1$\}$ since $d_2(C_1) = 1$.
(3)	$p_1(C_1) = r_4$; $v_1 = 100$; $t = 2$.
(4)	$SR_2 = \{$rows 2, 3$\}$ since $d_2(r_2) = d_2(r_3) = 2$.
(5)	$j =$ col. 1. $u_2 = (6 - 100)1 = -94$; $u_3 = (1-100)/4 = -99/4$. $t = 3$. Go to (2).
(2)	$SC_3 = \{C_2\}$ since $d_2(C_3) = 3$.
(3)	row 2 $= p_2(C_2)$, $v_2 = 5 - 3x(-94) = 287$; $t = 4$.
(4)	$SC_3 = \emptyset$. Go to (6).
(6)	$h = 2 = k$. STOP.

Thus $[u_1, u_2, u_3, u_4] = [9, -94, -99/4, 0]$ and $[v_1, v_2, v_3, v_4] = [100, 287, 3, 4]$ for the initial distribution. These are given as row and column headings of Figure 3 respectively.

Both the algorithms A2 and A5 discuss the determination of the primal and dual solutions. In [4] we will discuss algorithms for changing the basis from a given one-forest to the new one required at a pivot step. These are based on similar algorithms for the ordinary transportation problem [23].

The matrix $D = \{d_{ij}\}$ is called the dual matrix. It is known that [9, 20] a primal basic feasible solution is optimal if its dual solutions satisfy

(18) $\quad d_{ij} = e_{ij}u_i + v_j \leq c_{ij}$ for $(i,j) \in LB$ and

(19) $\quad d_{ij} = e_{ij}u_i + v_j \geq c_{ij}$ for $(i,j) \in UB$.

If equality occurs in either of these two equations then we say the solution is dual degenerate. The usual techniques [20] can be used to prevent dual degeneracy.

A basis B, more precisely a basis structure (B, LB, UB) is said to be dual feasible if u_i and v_j determined from (17) satisfy (18) and (19). If we define

(20) $\quad w_{ij} = \max(0,\ e_{ij}u_i + v_j - c_{ij})$

(which means $w_{ij} = 0$ for $(i,j) \notin UB$) for $i \in I$ and $j \in J'$ it may be verified that (17) - (19) imply (14) and (15).

From the duality theorem of linear programming [9, 20], a basic solution is optimal if it is both primal and dual feasible. Furthermore for such a solution,

(21) $$Z = \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} = \sum_{i\in I} a_i u_i + \sum_{j\in J} b_j v_j - \sum_{i\in I'}\sum_{j\in J'} U_{ij}w_{ij} = F.$$

EXAMPLE.(continued): Consider Figure 4 where the cells with c_{ij} circled give the basis. In the northeast corner of a cell (i,j) is given x_{ij} and it could be verified that this solution is basic and primal feasible. The dual variables u_i and v_j appearing at the left and top rims respectively satisfy (17). Unlike the ordinary transportation problem u_i and v_j are unique. The dual matrix $D = \{d_{ij} = e_{ij}u_i + v_j\}$ is given as Fig. 6. The reader may verify that the solution is dual feasible as well, i.e., satisfies (18) and (19) with $LB = \{(1,2), (2,3), (2,4), (2,5), (3,3), (3,4), (3,5), (4,1), (4,3), (4,4)\}$ and $UB = \{(1,1)\}$. Since this tableau given as Fig. 4 is both primal and dual feasible, it is optimal. Setting w_{ij} as given by (20) ($w_{ij} = 0$ for $(i,j) \neq UB$), it could be verified the $Z = F = \$1460$ as given in (21). In this optimal tableau (Fig. 4) every u_i with a cell, $(i,n+1) \in B$ is zero, (e.g., the absorbing cell). Further, all other $u_i < 0$ as $c_{i,n+1} = 0$. As we have both costs, c_{ij}, and the weights e_{ij} are non-negative, due to (17) all v_j's are also non-negative.

4. OPERATORS AND ASSOCIATED SOLUTIONS.

Let us now consider operators that transform the optimum solutions when the data of the problem are changed as a (linear) function of single parameter.

DEFINITION 14. An operator $\delta \underset{\sim}{T}(P)$ transforms the optimum solution of a problem P into that for problem $P^{\underset{\sim}{T}}$ with data:

$$(22)\quad \begin{cases} a_i^{\underset{\sim}{T}} = a_i + \delta\alpha_i & \text{for } i \in I' \\ b_j^{\underset{\sim}{T}} = b_j + \delta\beta_j & \text{for } j \in J' \\ c_{ij}^{\underset{\sim}{T}} = c_{ij} + \delta\gamma_{ij} & \text{for } i \in I';\ j \in J' \\ e_{ij}^{\underset{\sim}{T}} = e_{ij} + \delta\epsilon_{ij} & \text{for } i \in I';\ j \in J' \\ U_{ij}^{\underset{\sim}{T}} = U_{ij} + \delta\nu_{ij} & \text{for } i \in I';\ j \in J' \end{cases}$$

where a_i, b_j, c_{ij}, e_{ij}, and U_{ij} are the data of the problem P, δ is non-negative and α_i, β_j, γ_{ij}, ϵ_{ij} and ν_{ij} are given numbers, such that the transformed data satisfy assumptions (A1) and (A2). We denote this transformation as

$$(23)\quad \delta^{\underset{\sim}{T}}(P,B,LB,UB,X,D,Z) = (P^{\underset{\sim}{T}}, B^{\underset{\sim}{T}}, LB^{\underset{\sim}{T}}, UB^{\underset{\sim}{T}}, X^{\underset{\sim}{T}}, D^{\underset{\sim}{T}}, Z^{\underset{\sim}{T}})$$

where B, LB, UB, X, D, Z correspond to optimum solution of problem P and $B_{\sim}^{T}$, $LB_{\sim}^{T}$, $UB_{\sim}^{T}$, $X_{\sim}^{T}$, $D_{\sim}^{T}$, $Z_{\sim}^{T}$ correspond to those for problem $P_{\sim}^{T}$. In stating (22) we assume problem $P_{\sim}^{T}$ is also of type P satisfying assumptions A1 and A2. (It is of course possible that $P_{\sim}^{T}$ has no primal feasible solution.

REMARK 11. In the definition of the operator we have assumed that $\delta \geq 0$. This involves no loss of generality, since to study the effects of δ being negative, we can define another operator $\delta' \underset{\sim}{T}(P)$ with $\alpha_i' = -\alpha_i$; $\beta_j' = -\beta_j$; $\gamma_{ij}' = \gamma_{ij}$; $\epsilon_{ij}' = -\epsilon_{ij}$ and $\nu_{ij}' = -\nu_{ij}$ and still have $\delta' \geq 0$.

For most practical applications an operator as general as that of definition 13 is unnecessary. In [1] we consider rim operators ($\delta \underset{\sim}{R}(P)$) which arise when only the rim conditions are changed, i.e., $\nu_{ij} = \epsilon_{ij} = \nu_{ij} = 0$ for all $i \epsilon I'$; $j \epsilon J'$. Cost operators ($\delta(C(P))$ are also taken up in [1] in Section 3 , wherein the data correspond to cost entries alone i.e., $\alpha_i = \beta_j = \epsilon_{ij} = \nu_{ij} = 0$ for $i \epsilon I'$; $j \epsilon J'$. Weight operators $\delta \underset{\sim}{E}(P)$ that arise when only e_{ij}'s are changed (i.e., $\alpha_i = \beta_j = \gamma_{ij} = \nu_{ij} = 0$ for $i \epsilon I'$, $j \epsilon J'$ are discussed in [2].

The bound operators $\delta L(P)$ which result when only the U_{ij}'s change (i.e., $\alpha_i = \beta_j = \gamma_{ij} = \epsilon_{ij} = 0$ for $i \epsilon I'$, $j \epsilon J'$) are discussed in [1]. It is shown that these are equivalent to rim operators. Finally simultaneous application of combinations of operators are also given.

We call the above operators area operators to distinguish from cell operators. For a cell operator either a single cost entry c_{pq} or a single weight entry e_{pq} is changed. i.e., γ_{ij} or $\epsilon_{ij} = 0$ for all $(i,j) \neq (p,q)$, (cell cost operator or cell weight operator). Similarly when a single a_p and b_q alone are changed. i.e., $\alpha_i = 0$ for $i \neq p$ and $\beta_j = 0$ for $j \neq q$ (cell rim operator).

Surprisingly enough many of the applications of operator theory require only cell operators for which reason this is considered separately here in developing efficient algorithms. The cell operators are further classified into positive and negative operators depending on whether the data are increased or decreased.

DEFINITION 15. An operator $\delta \underset{\sim}{T}(P)$ is said to be __basis preserving__, and denoted by light face letters $\delta T(P)$, if the transformed problem P^T has an optimum solution with $B^T = B$, $LB^T = LB$ and $UB^T = UB$ (i.e., the basis structure is preserved).

We denote by μ^T the maximum value for δ (i.e., $0 \leq \delta \leq \mu^T$) so that the operator is basis preserving. In papers [1] and [2] our discussion is limited to such operators. In [3] we show that any operator can be expressed as a product of basis preserving operators.

Table 1 about here

Table 1 summarizes the operator classification discussed above.

Figure 1

i ↓ Mach. \ Prod → j	1	2	3	4	a_i
1	$\boxed{3}$ 3 $\boxed{0}$ $\boxed{15}$	$\boxed{2}$ 6 $\boxed{0}$ $\boxed{N}$	$\boxed{2}$ 3 $\boxed{0}$ $\boxed{40}$	$\boxed{3}$ 4 $\boxed{0}$ $\boxed{80}$	≤ 300
2	$\boxed{1}$ 6 $\boxed{150}$	$\boxed{3}$ 5 $\boxed{0}$ $\boxed{75}$	$\boxed{5}$ 8 $\boxed{0}$ $\boxed{25}$	$\boxed{10}$ 15 $\boxed{0}$ $\boxed{N}$	≤ 225
3	$\boxed{4}$ 1 $\boxed{0}$ $\boxed{60}$	$\boxed{2}$ 3 $\boxed{0}$ $\boxed{N}$	$\boxed{2}$ 10 $\boxed{0}$ $\boxed{30}$	$\boxed{1}$ 6 $\boxed{0}$ $\boxed{50}$	≤ 140
b_j	= 170	= 60	= 35	= 60	

Figure 2

i ↓ \ j →	1	2	3	4	5	a_i
1	$\boxed{3}$ 3 $\boxed{15}$	$\boxed{2}$ 6 $\boxed{N}$	$\boxed{2}$ 3 $\boxed{40}$	$\boxed{3}$ 4 $\boxed{80}$	$\boxed{1}$ 0 $\boxed{N}$	300
2	$\boxed{1}$ 6 $\boxed{150}$	$\boxed{3}$ 5 $\boxed{75}$	$\boxed{5}$ 8 $\boxed{25}$	$\boxed{10}$ 15 $\boxed{N}$	$\boxed{1}$ 0 $\boxed{N}$	225
3	$\boxed{4}$ 1 $\boxed{60}$	$\boxed{2}$ 3 $\boxed{N}$	$\boxed{2}$ 10 $\boxed{30}$	$\boxed{1}$ 6 $\boxed{50}$	$\boxed{1}$ 0 $\boxed{N}$	140
4	$\boxed{1}$ 100 $\boxed{N}$	$\boxed{1}$ 100 $\boxed{N}$	$\boxed{1}$ 100 $\boxed{N}$	$\boxed{1}$ 100 $\boxed{N}$	$\boxed{1}$ 0 $\boxed{N}$	1000
b_j	170	60	35	60	-	

Figure 3

$v_j \rightarrow$ / $u_i \downarrow$	100	287	3	4		a_i
0	$\boxed{3}$ 15 3 $\boxed{15}$	$\boxed{2}$ 6 $\boxed{N}$	$\boxed{2}$ 35 3 $\boxed{40}$	$\boxed{3}$ 60 4 $\boxed{80}$	$\boxed{1}$ 5 0 $\boxed{N}$	300
-94	$\boxed{1}$ 45 6 $\boxed{150}$	$\boxed{3}$ 60 5 $\boxed{75}$	$\boxed{5}$ 8 $\boxed{25}$	$\boxed{10}$ 15 $\boxed{N}$	$\boxed{1}$ 0 $\boxed{N}$	225
-99/4	$\boxed{4}$ 35 1 $\boxed{60}$	$\boxed{2}$ 3 $\boxed{N}$	$\boxed{2}$ 10 $\boxed{30}$	$\boxed{1}$ 6 $\boxed{50}$	$\boxed{1}$ 0 $\boxed{N}$	140
0	$\boxed{1}$ 75 100 $\boxed{N}$	$\boxed{1}$ 100 $\boxed{N}$	$\boxed{1}$ 100 $\boxed{N}$	$\boxed{1}$ 100 $\boxed{N}$	$\boxed{1}$ 925 0 $\boxed{N}$	1000
b_j	170	60	35	60	-	

Figure 4

$v_j \rightarrow$ / $u_i \downarrow$	31/5	28/5	3	4		a_i
0	$\boxed{3}$ 15 3 $\boxed{15}$	$\boxed{2}$ 6 $\boxed{N}$	$\boxed{2}$ 35 3 $\boxed{40}$	$\boxed{5}$ 60 4 $\boxed{80}$	$\boxed{1}$ 5 0 $\boxed{N}$	300
-1/5	$\boxed{1}$ 135 6 $\boxed{150}$	$\boxed{3}$ 30 5 $\boxed{75}$	$\boxed{5}$ 8 $\boxed{25}$	$\boxed{10}$ 15 $\boxed{N}$	$\boxed{1}$ 0 $\boxed{N}$	225
-13/10	$\boxed{4}$ 20 1 $\boxed{60}$	$\boxed{2}$ 30 3 $\boxed{N}$	$\boxed{2}$ 10 $\boxed{30}$	$\boxed{1}$ 6 $\boxed{50}$	$\boxed{1}$ 0 $\boxed{N}$	140
0	$\boxed{1}$ 100 $\boxed{N}$	$\boxed{1}$ 100 $\boxed{N}$	$\boxed{1}$ 100 $\boxed{N}$	$\boxed{1}$ 100 $\boxed{N}$	$\boxed{1}$ 1000 0 $\boxed{N}$	1000
b_j	170	60	35	60	-	

Figure 5

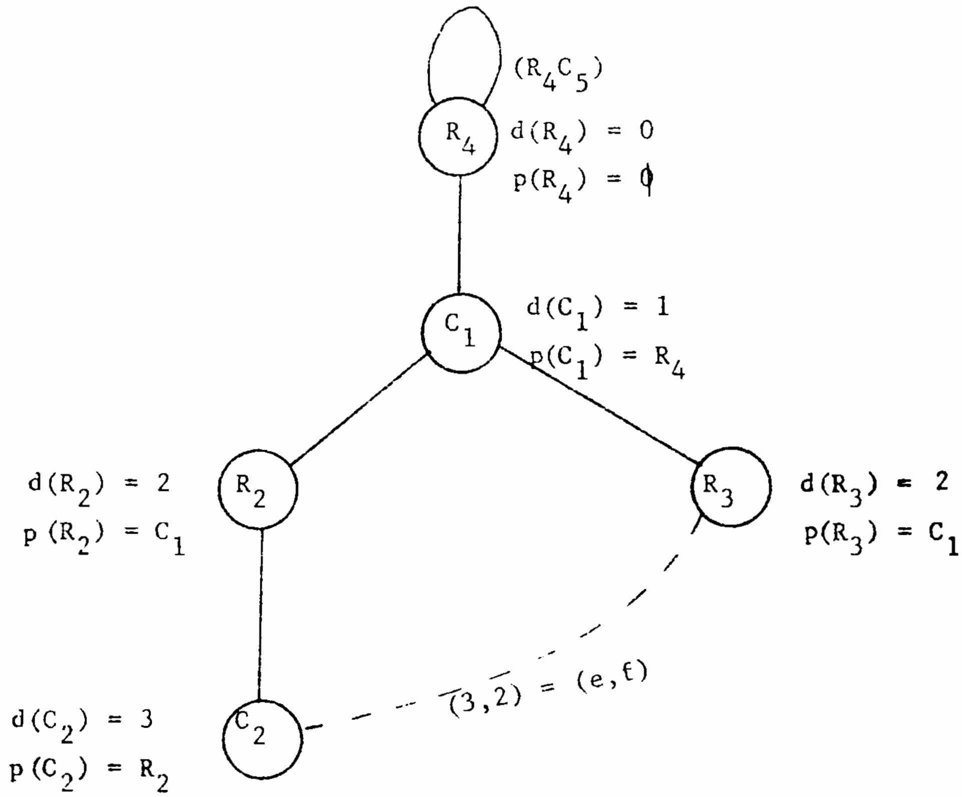


Figure 6

$$D: \quad \{d_{ij} = e_{ij}u_i + v_j\}$$

d_{ij} given in the center and e_{ij} given in northwest corner of a cell (i,j) $i \epsilon I$; $j \epsilon J'$:

$u_i \downarrow$ \ $v_j \rightarrow$	31/5	28/5	3	4
0	$\boxed{3}$ 31/5	$\boxed{2}$ 28/5	$\boxed{2}$ 3	$\boxed{3}$ 4
-1/5	$\boxed{1}$ 6	$\boxed{3}$ 5	$\boxed{5}$ 2	$\boxed{10}$ 2
-13/10	$\boxed{4}$ 1	$\boxed{2}$ 3	$\boxed{2}$ 2/5	$\boxed{1}$ 27/10
0	$\boxed{1}$ 31/5	$\boxed{1}$ 28/5	$\boxed{1}$ 3	$\boxed{1}$ 4

TABLE I. OPERATOR CLASSIFICATION

Symbol for Basis-Preserving operator*	Name	All the data for the transformed problem are the same as the original problem <u>except</u>	Notation for transformed problem**	Constraints on δ for basis preserving operator
$\delta T(P)$	operator	See eqn. (22)	P^T, B^T, LB^T, UB^T, X^T, D^T, Z^T	$0 \leq \delta \leq \mu^T$
$\delta R_A(P)$	area rim operator	$a_i^A = a_i + \delta\alpha_i$ for $i \epsilon I$ $b_j^A = b_j + \delta\beta_j$ for $j \epsilon J'$	P^A, B^A, LB^A, UB^A, X^A, D^A, Z^A	$0 \leq \delta \leq \mu^A$
$\delta R_{pq}^+(P)$	(plus) cell rim operator	$a_p^+ = a_p + e_{pq}\delta$ $b_q^+ = b^q + \delta$	P^+, B^+, LB^+, UB^+, X^+, D^+, Z^+	$0 \leq \delta \leq \mu^+$
$\delta R_{pq}^-(P)$	(minus) cell rim operator	$a_p^- = a_p - e_{pq}\delta$ $b_q^- = b_q - \delta$	P^-, B^-, LB^-, UB^-, X^-, D^-, Z^-	$0 \leq \delta \leq \mu^-$
$\delta C_A(P)$	area cost operator	$c_{ij}^A = c_{ij} + \delta\gamma_{ij}$ for $i \epsilon I$ and $j \epsilon J$	P^A, B^A, LB^A, UB^A, X^A, D^A, Z^A	$0 \leq \delta \leq \mu^A$
$\delta C_{pq}^+(P)$	(plus) cell cost operator	$c_{pq}^+ = c_{pq} + \delta$	P^+, B^+, LB^+, UB^+ X^+, D^+, Z^+	$0 \leq \delta \leq \mu^+$
$\delta C_{pq}^-(P)$	(minus) cell cost operator	$c_{pq}^- = c_{pq} - \delta$	P^-, B^-, LB^-, UB^- X^-, D^-, Z^-	$0 \leq \delta \leq \mu^-$
$\delta E_A(P)$	area weight operator	$e_{ij}^A = c_{ij} + \delta\epsilon_{ij}$ for $i \epsilon I'$, $j \epsilon J'$	P^A, B^A, LB^A, UB^A, X^A, D^A, Z^A	$0 \leq \delta \leq \mu^A$
$\delta E_{pq}^+(P)$	(plus) cell weight operator	$e_{pq}^+ = e_{pq} + \delta$	P^+, B^+, LB^+, UB^+, X^+, D^+, Z^+	$0 \leq \delta \leq \mu^+$
$\delta E_{pq}^-(P)$	(minus) cell weight operator	$e_{pq}^- = e_{pq} - \delta$	P^-, B^-, LB^-, UB^-, X^-, D^-, Z^-	$0 \leq \delta \leq \mu^-$
$\delta L_A(P)$	area bound operator	$U_{ij}^A = U_{ij} + \delta\nu_{ij}$	P^A, B^A, LB^A, UB^A, X^A, D^A, Z^A	$0 \leq \delta \leq \mu^A$
$\delta L_{pq}^+(P)$	(plus) cell bound operator	$U_{pq}^+ = U_{pq} + \delta$	P^+, B^+, LB^+, UB^+, $X^+, D^+, Z-$	$0 \leq \delta \leq \mu^+$
$\delta L_{pq}^-(P)$	(minus) cell	$U_{pq}^- = U_{pq} - \delta$	P^-, B^-, LB^-, Ub^-, X^-, D^-, Z^-	$0 \leq \delta \leq \mu^-$

* We use bold face letters $\underset{\sim}{T}, \underset{\sim}{R}, \underset{\sim}{C}$ to denote operators (not necessarily basis-preserving

** The notation (P,V,LB,UB,X,D,Z) is used for the original problem.

BIBLIOGRAPHY

[1] Balachandran, V. and G. L. Thompson, "An Operator Theory of Parametric Programming for the Generalized Transportation Problem - II; Cost, Rim and Bound Operators." GSIA Working paper No. , Carnegie-Mellon University, Pittsburgh, Pennsylvania.

[2] ________________ and ______________, "An Operator Theory of Parametric Programming for the Generalized Transportation Problem - III; Weight Operators," GSIA Working Paper No. , Carnegie-Mellon University, Pittsburgh, Pennsylvania.

[3] ________________ and ______________, "An Operator Theory of Parametric Programming for the Generalized Transportation Problem - IV; Global Operators," GSIA Working Paper No. , Carnegie-Mellon University, Pittsburgh, Pennsylvania.

[4] ________________ and _______________, "An Efficient Algorithm for the Generalized Transportation Problem," GSIA Working Paper No. , Carnegie-Mellon University, Pittsburgh, Pennsylvania

[5] Balas, E., "The Dual Method for the Generalized Transportation Problem," Management Science, Vol. 12, No. 7, March, 1966.

[6] Balas, E. and P. L. Ivanescu, "On the Generalized Transportation Problem," Management Science, Vol. 11, No. 1, September 1964.

[7] Berge, C., The Theory of Graphs and Its Applications, John Wiley and Sons, Inc., New York, New York, 1962.

[8] Charnes, A. and W. W. Cooper, Management Models and Industrial Applications of Linear Programming, Vol. I and II, John Wiley and Sons, New York, New York, 1961.

[9] Dantzig, G. B., "Linear Programming and Extensions," Princeton University Press, 1963.

[10] Eisemann, K., "The Generalized Stepping Stone Method for the Machine Loading Model," Management Science, Vol. 11, No. 1, September 1964.

[11] Eisemann, K. and J. R. Lourie, "The Machine Loading Problem," IBM 704 Program, IBML-1, IBM Applications Library, New York, 1959.

[12] Ferguson, A. R. and G. B. Dantzig, "The Allocation of Aircraft to Routes -- An Example of Linear Programming Under Uncertain Demand," Management Science, Vol. 3, No. 1, 1956.

[13] Garwin, W. W., Introduction to Linear Programming, McGraw-Hill Book Company, New York, New York., 1960.

[14] Glicksman, S., L. Johnson and L. Eselson, "Coding the Transportation Problem," Naval Research Logistics Quarterly, 7, 1960.