THE THEORY OF NETWORK APPLIED TO TRANSPORTATION, TRANSSHIPMENT AND MAXIMAL FLOW PROBLEMS

by

RAMESH NARAYAN HICHKAD

D. M. E., Sir B. Polytechnic Institute
Bhavnagar, Gujarat, India, 1963

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1967

Approved by:

Frank A. Tillman
Major Professor

TABLE OF CONTENTS

1. INTRODUCTION

In recent years, the theory of networks has received increased interest in the field of operations research, in connection with the solution of allocation problems, shortest route problems, and other flow problems. T. C. Koopmans (9), in his work on transportation problems, was the first to interpret properties of optimal and nonoptimal solutions with respect to the linear graph associated with a network of routes. This work was then followed by the pioneering work on flows in networks by Ford and Fulkerson (6, 7).

The purpose of this report is to present the application of the theory of networks to transportation, transshipment, and maximal flow problems. In Section 2 some definitions and theorems concerning the theory of network and maximal flow problems are presented.

In Section 3 a simple transportation problem is solved by four methods. The type of transportation problem considered consists of a number of source points where the commodities are available in limited quantities, and a number of demand points where there is a requirement for the commodity. The cost of shipping one unit of commodity from a source point to a demand point is known. The problem is to determine the minimum cost route of transporting the commodity and to satisfy all the requirements. The first method used to solve the example is the simplex algorithm, developed by Dantzig (2). The second is the one commonly called "the transportation algorithm" (2, 8, 10), which is a modification of the simplex method. The third method is the discrete maximum principle which has been shown by Fan, Wang, and others (1, 5), that it is applicable to solving transportation problems. The fourth and final method used in section 3 is network theory as applied to the transportation problem.

In section 4 a transshipment problem is solved to point out the ease of solving problems with networks as compared to the simplex method and the transshipment algorithm (2). The transshipment problem is a special type of transportation problem where several possible routes for shipping exist between a source point and a demand point. The main difference and the added difficulty of transshipment problems is that these intermediate points, where neither supply nor any demand exists, have to be considered. An example problem, consisting of eight nodes, two source points, three demand points, and three intermediate points, is set up as a linear programming problem, and is solved by the transshipment algorithm and the method of networks. All the methods used in this report are applicable only to the problems with linear cost function, except discrete maximum principle.

The application of the theory of maximal flows in networks is presented in section 5. The maximal flow problem is one of determining the maximum flow from a source to a sink when each arc of the network has some fixed flow capacity. The sink is the ultimate destination of flow originating at the source. The commodity which flows in networks may be such items as fluids, electricity, funds, and automobiles, where the quality is measured as rate. An example is solved using the method of chains of positive arc capacities, the labeling method, and the matrix method. These methods are presented in Ford and Fulkerson (6, 7) and basically depend on the max-flow min-cut theorem (3, 4), which apply to steady-state flows, that is, arc flows which do not change with time. It is noted that dynamic programming and discrete maximum principle can be applied to all the problems considered in this report but not much work has been done at this time.

2. SOME DEFINITIONS AND THEOREMS

Network. A network or linear graph consists of a number of nodes, junction points, or points, each joined to some or all of the others by arcs, links, branches, or edges.

The numbers with circles in Fig. 1 are the nodes. The arcs or branches are shown by straight or curved line segments, each of which links two nodes, e.g., (1, 2), (1, 3), (3, 2), (5, 4), etc.

If i and j are two nodes, then (i → j) is used to denote a directed arc and represents an allowable precedence between i and j. In Fig. 1, if it is allowed to proceed, starting at 1, to 3 or 4; or starting at 2, to 1 or 3; or starting at 3, to 2 or 5; or starting at 4, to 5, then the directed arcs are (1 → 3), (1 → 4), (2 → 1), (2 → 3), (3 → 2), (3 → 5), and (4 → 5), which form a directed or oriented network as shown in Fig. 2. Arcs in the case of a directed network are also known as oriented arcs or oriented branches.

Chain. A chain is a sequence of arcs (i, i_1), (i_1, i_2), . . ., (i_k, j), connecting the nodes, i and j, regardless of the ways in which these arcs may be directed (Fig. 3).

Path. When every arc of a chain has the same direction, the chain so formed is called a path (Fig. 4).

Loop. A chain of arcs connecting node i to itself is called a loop (Fig. 5).

Tree. A network having no loops in which every node is connected to every other node through a chain of arcs is

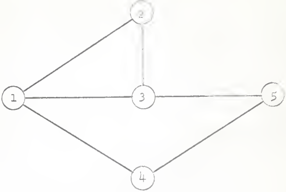


Fig. 1. Example of a network (undirected).

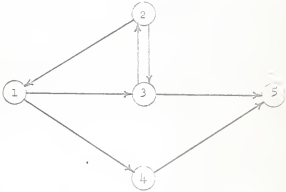


Fig. 2. Example of a directed network.

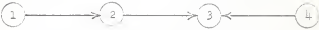


Fig. 3. Example of a chain.

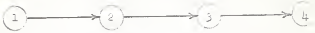


Fig. 4. Example of a path.

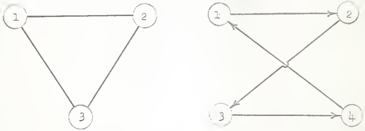


Fig. 5. Examples of loops.

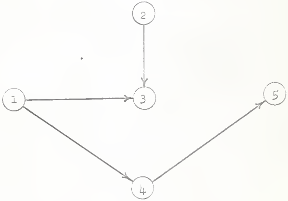


Fig. 6. Example of a tree (directed).

called a tree (Fig. 6).

Theorem 1. A network having n nodes is a tree if it has (n - 1) arcs and no loops.

Capacity. Capacity of the arc (i → j) is the maximal amount of some commodity that can arrive at node j from node i per unit time. Note that it is possible to have infinite arc capacities.

Flow. Flow in the arc (i → j) is the amount of some commodity allowed to arrive at node j from node i per unit time and is always less than or equal to the capacity of that arc. Directed or oriented arcs represent the direction of flow in that arc.

Source, Sink, and Intermediate Nodes or Points. A node i is called a source if every arc which has this node as an end point is oriented in such a way that the flow in the arc moves away from i to another node. A node j is called a sink if every arc which has this node as an end point is oriented in such a way that the flow is from other nodes to j. And all other nodes are known as intermediate nodes.

3. TRANSPORTATION PROBLEM

A classical transportation problem can be described as follows. A given quantity of a commodity is available at each of several origins or sources. There is a requirement for the commodity at each of the destinations or sinks. The cost of shipping one unit from an origin to a destination is known. Also it is possible to ship from any origin to any destination. The costs of shipping between the sources and the destinations are linear and proportional to the amount shipped. The problem is to determine the minimum cost routing from the origins to the deatinations and to satisfy all the requirements.

The transportation problem can be put in equation form as follows:

Let there be m origins (supply points) and n destinations (demand points).

Let x_{ij} be the number of units shipped from origin i to destination j; let O_i be the number of units of the commodity available at origin i, and D_j the number of units required at destination j. Let c_{ij} be the cost of shipping one unit from origin i to destination j. The standard transportation algorithm is based on the assumption that the total demand equals the total supply. Of course, it is always possible to satisfy this condition by inserting dummy demand or supply. Thus the problem is to find $x_{ij} \geqslant 0$ which minimizes the objective function,

$$z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \, x_{ij} \qquad (1)$$

subject to the supply constraints,

$$\sum_{j=1}^{n} x_{ij} = O_i, \qquad i = 1, 2, \ldots, m \qquad (2)$$

and the demand constraints,

$$\sum_{i=1}^{m} x_{ij} = D_j, \qquad j = 1, 2, \ldots, n \qquad (3)$$

with the condition that the total demand equals the total supply; that is,

$$\sum_{j=1}^{n} D_j = \sum_{i=1}^{m} O_i \qquad (4)$$

In this section a transportation problem is solved by the simplex algorithm, the transportation algorithm, the discrete maximum principle, and the method of networks.

An Example. Cities 1 and 2 are the supply points with 7 and 3 units of the product available, respectively. Cities 3, 4, and 5 are demand points with the demand for 4, 4, and 2 units of the product, respectively. The shipping costs per unit are given as follows:

To demand city:		3	4	5
From supplying city:	1	5	3	6
	2	1	4	7

The problem is to find the optimal solution which will minimize the total shipping costs.

3.1. Method I, Simplex Algorithm

The linear progremming problem formulation of the above problem is obtained by using the equetions (1), (2), end (3). The problem becomes one of finding $x_{ij} \geqslant 0$, the amount shipped from source i to destinetion j, i = 1, 2 end j = 3, 4, 5 which minimizes the objective function,

$$z = 5x_{13} + 3x_{14} + 6x_{15} + x_{23} + 4x_{24} + 7x_{25} \quad (5)$$

subject to the supply constreints,

$$\begin{aligned} x_{13} + x_{14} + x_{15} &= 7 \\ x_{23} + x_{24} + x_{25} &= 3 \end{aligned} \quad (6)$$

end the demand constraints,

$$\begin{aligned} x_{13} + x_{23} &= 4 \\ x_{14} + x_{24} &= 4 \\ x_{15} + x_{25} &= 2 \end{aligned} \quad (7)$$

This linear progremming problem can be solved using the simplex algorithm as given in eny standard book on operetions reseerch (Dentzig (2) or Hedley (8)). Refer to Teble 1 for complete solution of the problem. Note thet each equation in (6) and (7) cen be represented as a lineer combination of other equations, owing to the fact that supply equals demand. This reduces the independent constraint equations to four; the fifth (any one) equetion is redundant end cen be discerded. Thus the number of besic variebles in this type of problem is always

Table 1. Simplex Algorithm for the Transportation Problam.

	$c_j=c_{ij}$:		5	3	6	1	4	7	M	M	M	M	
C_{Bi}	X_B	$\overline{b}$	x_{13}	x_{14}	x_{15}	x_{23}	x_{24}	x_{25}	A_1	A_2	A_3	A_4	b/a
M	A_1	7	1	1	1	0	0	0	1	0	0	0	
M	A_2	3	0	0	0	$\textcircled{1}$	1	1	0	1	0	0	3/1 → min
M	A_3	4	1	0	0	1	0	0	0	0	1	0	
M	A_4	4	0	1	0	0	1	0	0	0	0	1	4/1
z_j-	min		-5	-3	-6	-1	-4	-7	0	0	0	0	
c_j	z =	18M	+2M	+2M	+M	+2M	+2M	+M					
						↑							most+ve
M	A_1	7	1	1	1	0	0	0	1	0	0	0	7
1	x_{23}	3	0	0	0	1	1	1	0	1	0	0	
M	A_3	1	1	0	0	0	-1	-1	0	-1	1	0	4 →
M	A_4	4	0	$\textcircled{1}$	0	0	1	0	0	0	0	1	↗
z_j-	min	3	-5	-3	-6	0	-3	-6	0	1	0	0	
c_j	z =	12M	+2M	+2M	+M			-M		-2M			
				↑									
M	A_1	3	1	0	1	0	-1	0	1	0	0	-1	3/1
1	x_{23}	3	0	0	0	1	1	1	0	1	0	0	
M	A_3	1	$\textcircled{1}$	0	0	0	-1	-1	0	-1	1	0	1/1 →
3	x_{14}	4	0	1	0	0	1	0	0	0	0	1	
z_j-	min	15	-5	0	-6	0	0	-6	0	1	0	3	
c_j	z =	4M	+2M		+M		-2M	-M		-2M		-2M	
			↑										
M	A_1	2	0	0	$\textcircled{1}$	0	0	1	1	1	-1	-1	2 →
1	x_{23}	3	0	0	0	1	1	1	0	1	0	0	
5	x_{13}	1	1	0	0	0	-1	-1	0	-1	1	0	
3	x_{14}	4	0	1	0	0	1	0	0	0	0	1	
z_j-	min	20	0	0	-6	0	-5	-11	0	-4	5	3	
c_j	z =	2M			+M			+ M			-2M	-2M	
					↑								
6	x_{15}	2	0	0	1	0	0	1	1	1	-1	-1	
1	x_{23}	3	0	0	0	1	1	1	0	1	0	0	
5	x_{13}	1	1	0	0	0	-1	-1	0	-1	1	0	
3	x_{14}	4	0	1	0	0	1	0	0	0	0	1	
z_j-	min	32	0	0	0	0	-5	-5	6	2	-1	-3	
c_j	z								-M	-M	-M	-M	

Nota: At the fourth iteration the artificial variables are out of the basis and the optimality critarion for minimum z is satisfied. Therefore tha solution at tha fourth itaration is the optimal solution with $x_{13} = 1$, $x_{14} = 4$, $x_{15} = 2$, and $x_{23} = 3$, and tha minimum cost is aqual to 32.

equal to (m + n - 1), where m is the number of origins and n is the number of destinations. The optimal solution obtained using the simplex algorithm is the optimal solution for the transportation problem where the minimum cost is given by z, as shown in Table 1.

It is observed that the constraint equations in any transportation problem have a simple form. All the nonzero coefficients of the active variables x_{ij} are ones; furthermore, any given x_{ij} appears in two and only two of the constraints. (Refer to equations (6) and (7).) Because of these special properties of the constraints, it is possible to solve a transportation problem using a much simpler and faster method known as the "transportation algorithm", which will be discussed in the next section.

3.2. Method II, The Transportation Algorithm

The example solved in section 3.1, can be written in a compact matrix form as shown in Table 2, where c_{ij} is the cost of shipping one unit from origin i to destination j, the O_i is the quantity available at origin i and D_j is the requirement at destination j. To simplify the discussion, the algorithm is explained first and then illustrated by solving the example.

Steps of the Algorithm.

Step 1. Assign initial allocations by one of the three methods given below.

1. Northwest corner rule: Starting with the upper left corner of Table 2, make the appropriate assignments

on a diagonal path to tha lower right corner depleting the sources and satisfying the demand and having no more than $(m + n - 1)$ assignments.

2. Method of allocating to the cell with minimum shipping cost: Determine the smallest cost for a cell in the tableau. Suppose this occurs for cell (i, j). Make the maximum possible assignment to this cell. Then neglect row i or column j depending on whether source i is depleted or demand j is satisfied. Repeat the process until the total supply is exhausted and all the demands ara satisfied and where no more than $(m + n - 1)$ assignments have been made.
3. Vogel's method: For aach row, find the lowest c_{ij} and the next lowest cost c_{it} in that row. Compute $(c_{it} - c_{ij})$. In this way, m numbers are obtained. Do the same for each column obtaining n additional numbers. Choose the largest of these $(m + n)$ numbers and make the maximum possible assignment to the cell with the lowest cost c_{ij} in that row or column depending upon whether this number resulted from a row or column computation. Cross off either row i or column j, depending upon whether the source i is depleted or requirement demand j is satisfied, and repeat the process for the resulting tableau. When the maximum difference is not unique, an arbitrary choice can ba made, and if a row and column constraint are satisfied simultanaously, cross off only the row or the column, not

> both. The process is terminated when the supply is exhausted and the demands are satisfied and when no more than (m + n - 1) assignments have been made.

In each of the three methods the (m + n - 1) allocations constitute the initial basic feasible solution.

Step 2. Determine a set of (m + n) numbers u_i, i = 1, . . ., m, and v_j, j = 1, . . ., n such that for each occupied cell (i, j), $c_{ij} = u_i + v_j$. Note that u_i and v_j are some arbitrarily chosen values and are known as the implicit prices.

Step 3. Calculate $\theta_{ij} = c_{ij} - (u_i + v_j)$ for each unoccupied cell (i, j). θ_{ij} is known as the relative cost factor.

Step 4. Examine the values of θ_{ij} of each unoccupied cell (i, j); if all are positive an optimal solution has been found. If one or more are negative, the solution is not optimal and a change in the existing solution is possible which will reduce the cost.

Step 5. Find a path to a negative θ_{ij} (generally most negative) cell by horizontal and vertical movements along the shortest path of assigned cells.

Step 6. In the path determined in Step 5, find the maximum allocation for each horizontal move, then find the maximum allocation for each vertical move, select the minimum of these values and assign it to the cell with the negative θ_{ij} in Step 5. Adjust the assignments on this path so that all demands and supplies are compatible, then go to Step 2 and repeat the process until an optimal assignment is found according to the criterion in Step 4. In solving the example problem by this method, the initial solution is found using the second method of Step 1, and the allocations which result are: $x_{23} = 3$, $x_{14} = 4$, $x_{13} = 1$, and $x_{16} = 2$, which are shown in Table 3. Note that an occupied or assigned cell (i, j) is also known as the basic variable $x_{ij} = 1$ and the other cells are known as nonbasic variables equal to zero. Thus cell (2,4) and cell (2,5) in the above problem correspond to the nonbasic variables $x_{24} = 0$ and $x_{25} = 0$. According to Step 2, the implicit prices u_i and v_j for each of the basic variables are found from the relation,

$$c_{ij} = u_i + v_j \qquad (8)$$

for the occupied cells.

The procedure is usually initiated by setting the u_i or v_j with the most assignments in the row or column equal to zero and evaluating the remaining values by equation (8). The values of u_i and v_j for the example are shown in Table 4. The relative cost factor θ_{ij} for each of the unoccupied cells is evaluated from equation (9),

$$\theta_{ij} = c_{ij} - (u_i + v_j) \qquad (9)$$

for the unoccupied cells. Thus

$$\theta_{24} = 4 - (-4 + 3) = 5$$

and

$$\theta_{25} = 7 - (-4 + 6) = 5.$$

Since both of these relative cost factors are nonnegative, this solution is the optimal basic feasible solution to the problem as noted in Table 4.

Summarizing, then, the optimal solution is,

Table 2. Transportation example.

		Destination			
	From \ To	3	4	5	Supply
Origin	1	$c_{13}=5$	$c_{14}=3$	$c_{15}=6$	$7 = O_1$
	2	$c_{23}=1$	$c_{24}=4$	$c_{25}=7$	$3 = O_2$
Demand		$4=D_3$	$4=D_4$	$2=D_5$	$10 \quad \sum_{i=1,2} O_i = \sum_{j=3,4,5} D_j$

Table 3. Initial basic feasible solution.

		Destination			
	From \ To	3	4	5	Supply
Origin	1	$1=x_{13}$ 5	$4=x_{14}$ 3	$2=x_{15}$ 6	7
	2	$3=x_{23}$ 1	4	7	3
Demand		4	4	2	10

Table 4. The optimal solution.

		Destination				
	From \ To	3	4	5	Supply	u_i
Origin	1	1 5	4 3	2 6	7	$u_1=0$
	2	3 1	$\theta_{24}=5$ 4	$\theta_{25}=5$ 7	3	$u_2=-4$
Demand		4	4	2	10	
v_j		$v_3=5$	$v_4=3$	$v_5=6$		

Basic variable	From city	To city	No. of units transported
x_{13}	1	3	1
x_{14}	1	4	4
x_{15}	1	5	2
x_{23}	2	3	3

and the total minimum cost, z

$$= 5x_{13} + 3x_{14} + 6x_{15} + x_{23} + 4x_{24} + 7x_{25}$$

$$= 5 + 12 + 12 + 3$$

$$= 32 \text{ units of cost.}$$

3.3. Method III, The Discrete Maximum Principle

The general algorithm of the Discrete Maximum Principle for systems without information feedback is given by Fan and Wang (12) and is summarized below.

The process consists of N stages connected in series. The state of the process stream, denoted by s-dimensional vector, $x = [x_1, x_2, \ldots, x_s]$ is transformed at each stage according to an r-dimensional decision vector $\theta = [\theta_1, \theta_2, \ldots, \theta_r]$, which represents the decisions made at that stage. The transformation of the process stream at the n^{th} stage is described by a set of performance equations.

$$x_i^n = T_i^n (x_1^{n-1}, x_2^{n-1}, \ldots, x_s^{n-1}; \theta_1^n, \theta_2^n, \ldots, \theta_r^n) \quad (10)$$

$$x_i^0 = \alpha_i$$

or in vector form

$$x^n = T^n\ (x^{n-1};\ \theta^n) \text{ and } x^0 = \alpha$$

where $i = 1, 2, \ldots, s$; $n = 1, 2, \ldots, N$; and the superscript n indicates the stage number.

A typical optimization problem is to determine the sequence of θ^n, subject to the constraints,

$$\psi_i^n\ (\theta_1^n, \theta_2^n, \ldots, \theta_r^n) \leq 0,\ \begin{matrix} n = 1, 2, \ldots, N \\ i = 1, 2, \ldots, s \end{matrix} \qquad (11)$$

which optimizes the objective function of the process,

$$S = \sum_{i=1}^{s} c_i\ x_i^n, \quad c_i = \text{constant} \qquad (12)$$

when the initial condition $x_i^0 = \alpha_i$ is given.

The procedure for finding the optimal sequence of θ^n is to introduce an s-dimensional adjoint vector z^n and a Hamiltonian function H^n which satisfy the following relations:

$$H^n = \sum_{i=1}^{s} z_i^n\ T_i^n\ (x^{n-1};\ \theta^n),\ n = 1, 2, \ldots, N \qquad (13)$$

$$z_i^{n-1} = \frac{\partial H^n}{\partial x_i^{n-1}}\ ,\ n = 1, 2, \ldots, N,\ i = 1, 2, \ldots, s\ (14)$$

$$z_i^N = c_i,\ \ i = 1, 2, \ldots, s$$

and to determine the optimal sequence of control action, $\overline{\theta}^n$, from the conditions

$$H^n = \text{extremum or } \frac{\partial H^n}{\partial \theta^n} = 0.$$

Formulation of the Transportation Problem by the Discrete Maximum Principle. To formulate the transportation problem as a discrete maximum principle problem, several quantities have to be defined. Let θ_i^n = the quantity of the commodity sent from i^{th} source or origin to the n^{th} destination or demand point where $C_i^n(\theta_i^n)$ = the cost incurred by sending θ_i^n. For a system with s origins and N destinations, the problem becomes one of determining the values of θ_i^n, $i = 1, 2, \ldots, s$; $n = 1, 2, \ldots, N$, so as to minimize the total cost of transporting the commodity,

$$C_{sN} = \sum_{n=1}^{N} \sum_{i=1}^{s} C_i^n (\theta_i^n)$$

subject to the constraints

(i) $\theta_i^n \geqslant 0$.

(ii) $\sum_{n=1}^{N} \theta_i^n = O_i$, the number of units of the commodity available at i^{th} origin, $i = 1, 2, \ldots, s$.

(iii) $\sum_{i=1}^{s} \theta_i^n = D^n$, the number of units of the commodity required by the n^{th} demand point, $n = 1, 2, \ldots, N$.

The demand points are defined as stages and the quantity shipped from the i^{th} origin to the first n stages or demand points are defined as the state variables x_i^n, $i = 1, 2, \ldots, s - 1$. Thus

$$x_i^n = x_i^{n-1} + \theta_i^n \ , \quad x_i^0 = 0 \ , \quad x_i^N = O_i$$

$$i = 1, 2, \ldots, s-1, \ n = 1, 2, \ldots, N.$$

Note here that the number of units supplied from the sth source to nth destination (stage) can be obtained by subtracting the sum of the units supplied to the nth stage from the first through (s - 1)th sources from the total number of units required by the nth stage; that is

$$\theta_s^n = D^n - \sum_{i=1}^{s-1} \theta_i^n \qquad (18)$$

A new state variable, x_s^n is defined as

$$x_s^n = x_s^{n-1} + \sum_{i=1}^{s} C_i\ (\theta_i^n)\ , \quad x_s^0 = 0$$

$$i = 1,\ 2,\ .\ .\ .,\ N$$

where x_s^N is equal to the total cost of transportation. The problem is to minimize x_s^N with proper choice of the sequence of θ_i^n, i = 1, 2, . . .,(s - 1); n = 1, 2, . . ., N. The Hamiltonian function can be written as

$$H^n = \sum_{i=1}^{s-1} z_i^n\ (x_i^{n-1} + \theta_i^n) + z_s^n \left[x_s^{n-1} + \sum_{i=1}^{s} C_i^n\ (\theta_i^n)\right] \qquad (20)$$

$$n = 1,\ 2,\ .\ .\ .,\ N,$$

and components of the adjoint vector are, in general,

$$z_i^{n-1} = \frac{\partial H^n}{\partial x_i^{n-1}} = z_i^n\ , \quad i = 1,\ 2,\ .\ .\ .,\ s \qquad (21)$$

with $z_s^n = 1$, n = 1, 2, . . ., N.

Since z_i^n and x_i^{n-1} are considered as constants at each step in the minimization of the Hamiltonian function given by equation (20), it is convenient to define the variable part of the

Hamiltonian function as

$$H_v^n = \sum_{i=1}^{s-1} z_i^n \theta_i^n + \sum_{i=1}^{s} C_i^n (\theta_i^n) . \qquad (22)$$

The transportation problem previously solved in sections 3.1 and 3.2 is presented in Table 5 and will be solved again as a discrete maximum principle problem. Since in this example there are two origins and three destinations, then s = 2 and N = 3. C_i^n is the cost incurred in shipping one unit from the i^{th} origin to the n^{th} destination. The total number of units required by N destination is equal to the total number of units supplied from s origins; that is,

$$\sum_{n=1}^{N} D^n = \sum_{i=1}^{s} O_i .$$

The variable part of the Hamiltonian function using equation (22) for the above example is,

$$H_v^n = z_1^n \theta_1^n + C_1^n \theta_1^n + C_2^n \theta_2^n, \quad n = 1, 2, 3.$$

Since $\theta_2^n = D^n - \theta_1^n$, the following is obtained.

$$H_v^n = (z_1^n + C_1^n - C_2^n) \theta_1^n + C_2^n D^n, \quad n = 1, 2, 3.$$

Stage 1. For the first destination, n = 1, the variable part of the Hamiltonian equation becomes,

$$H_v^1 = (z_1^1 + C_1^1 - C_2^1) \theta_1^1 + C_2^1 D^1$$

From Table 5, this becomes

Table 5. Transportation costs, supplies, and demands.

		Destination			
	n	1	2	3	Supply
	i				
Origin	1	$c_1^1 = 5$	$c_1^2 = 3$	$c_1^3 = 6$	$O_1 = 7$
	2	$c_2^1 = 1$	$c_2^2 = 4$	$c_2^3 = 7$	$O_1 = 3$
Demand		$D^1 = 4$	$D^2 = 4$	$D^3 = 2$	10

Table 6. Conditions necessary for H_v^n to be minimum.

n	Minima of H_v^n occurring at	
	θ_1^n	z_1^n
1	0 $0 \leqslant \theta_1^1 \leqslant 4$ 4	> -4 $= -4$ < -4
2	0 $0 \leqslant \theta_1^2 \leqslant 4$ 4	> 1 $= 1$ < 1
3	0 $0 \leqslant \theta_1^3 \leqslant 2$ 2	> 1 $= 1$ < 1

$$H_v^1 = (z_1^1 + 4)\ \theta_1^1 + 4\ .$$

Thus $z_1^1 = -4 = C_2^1 - C_1^1$ is the boundary point such that H_v^1 is minimum for $0 \leq \theta_1^1 \leq 4$.

There are three conditions at which H_v^1 may be minimum:

(a) H_v^1 = minimum at $\theta_1^1 = 0$ when $z_1^1 > -4$

(b) H_v^1 = minimum at $0 \leq \theta_1^1 \leq 4$ when $z_1^1 = -4$

(c) H_v^1 = minimum at $\theta_1^1 = 4$ when $z_1^1 < -4$.

Using this same approach, the conditions on θ_1^n for minimum H_v^n are illustrated in Table 6. As given by equation (21), the value of z_1^n, n = 1, 2, 3, is identical. From Table 6, the boundary values of z_1^n are determined and illustrated in Fig. 7.

The value of z_1^n which gives all the solutions satisfying the constraints from conditions (i), (ii), and (iii) can be obtained. The solution which minimizes the cost is then selected. For example, the solutions θ_1^n and the corresponding values of x_1^n in the region of $-4 < z_1^n < 1$, n = 1, 2, 3, are obtained using Table 6. Knowing the values of θ_1^n, θ_2^n are obtained from equation (18), they are as shown in Table 7. This solution does not satisfy the end point conditions: $O_1 = 7$ and $O_2 = 3$. In order to satisfy the end point conditions, the corresponding solution of $z_1^n = -4$ is found where $0 \leq \theta_1^1 \leq 4$, and thus θ_1^1 has to be equal to 1. By this process other solutions are checked for feasibility and then minimum cost. It turns out $\theta_1^1 = 1$

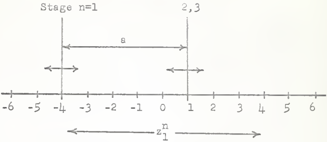


Fig. 7. Boundary values of adjoint vector z_1^n.

Table 7. O_i^n corresponding to value of $-4 < z_1^n < 1$.

		Destination			
	n / i	1	2	3	Supply, O_i
Origin	1	0	4	2	6 (7)
	2	4	0	0	4 (3)
Demand, D^n		4	4	2	10

Table 8. Optimal feasible solution.

		Destination			
	n / i	1	2	3	Supply, O_i
Origin	1	1	4	2	7
	2	3	0	0	3
Demand, D^n		4	4	2	10

gives the feasible optimal solution as shown in Table 8.

The total minimum cost = $\sum_{n=1}^{3} \sum_{i=1}^{2} c_i^n \theta_i^n = 5 \times 1 + 3 \times 4 + 6 \times 2 + 1 \times 3 = 32$ units, which is the same as the one obtained by other methods.

3.4. Method IV, Method of Network

In this method a network representation of the problem is constructed and the calculations are made directly on the network. The transportation problem considered in the previous sections can be represented in the form of a network as shown in Fig. 8. The cost, c_{ij}, of shipping a unit of commodity from origin i to a destination j, is shown on the respective arc. The amount of supply at each origin, O_i, and the amount of demand at each destination, D_j, are noted at each respective node. This type of problem can be solved readily by the method of networks, the steps of which are given below.

Steps of the Method.

Step 1. To find an initial basic feasible solution by forming a tree: To do this, each origin i is joined to the other nodes of the network using the arcs pointing away from the origin. Similarly, each destination is joined to the other nodes using the arcs pointing into the destination. This is continued until a chain of arcs from an origin joins with a chain of arcs to a destination. At this point start joining other nodes to this chain so that no loops are formed. When all nodes are joined in this manner a tree will have been formed as defined in section 2. Then assign values to the basic variables corresponding to the arcs of this tree such that all the supply and demand conditions are met.

Step 2. Compute the implicit price π_i such that equation (23)

$$\pi_j - \pi_i = c_{ij} \tag{23}$$

is true for arcs ($i \rightarrow j$) of the tree in step 1. Note that π_i and π_j of this method correspond to u_i and v_j of the transportation algorithm in equation (8).

Step 3. Calculate the relative cost factor, δ_{ij}, using equation (24), for each arc ($i \rightarrow j$) which is not part of the tree but exists in the original network.

$$\delta_{ij} = c_{ij} - (\pi_j - \pi_i) \tag{24}$$

Here again δ_{ij} corresponds to θ_{ij} of the transportation algorithm in equation (9).

Step 4. Examine the values of δ_{ij}. If all are nonnegative, then this solution is the optimal basic feasible solution. If one or more δ_{ij} are negative, then the solution is not optimal; then go to step 5.

Step 5. Draw the arc with a negative (generally most negative) value of δ_{ij} and include that arc in the tree. The remaining values of the basic variables around the loop thus formed are adjusted with a value assigned to the new arc. The appropriate arc is dropped to obtain a new tree. The procedure is again repeated from step 2 using the new basic feasible

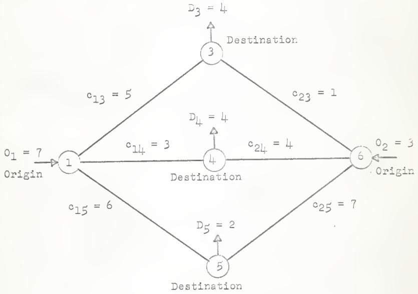


Fig. 8. Network of transportation problem.

solution until the optimal solution is obtained. The previous example as illustrated in Fig. 8 is again solved using the method of networks. In accordance with step 1, an initial basic feasible solution is determined and is shown in Fig. 9. Note that the values of the variables assigned on the arcs of the tree in Fig. 9 should be such that the algebraic sum at each node is zero where values are considered to be positive when the arrow points into the node and are considered to be negative when the arrow points away from the node.

According to step 2, the implicit prices are obtained and are shown in Fig. 10. In determining those implicit prices, the value $\pi_i = 0$ is assigned to the node i which has a large number of arcs pointing away from it. In Fig. 9, this node is 1, so π_1 is given a value of zero and other values are obtained from the equation (23) which is $\pi_j - \pi_i = c_{ij}$, and are shown in Fig. 10.

The arcs which are not in the tree but exist in the original network, Fig. 8, are (2 → 4) and (2 → 5). From step 3, the relative cost factor, δ_{ij}, for each of these arcs is obtained by using equation (24), which is $\delta_{ij} = c_{ij} - (\pi_j - \pi_i)$. Therefore

$$\delta_{24} = c_{24} - (\pi_4 - \pi_2) = 4 - (3 - 4) = 5$$

and

$$\delta_{25} = c_{25} - (\pi_5 - \pi_2) = 7 - (6 - 4) = 5.$$

According to step 4, since all δ_{ij}'s are nonnegative, this

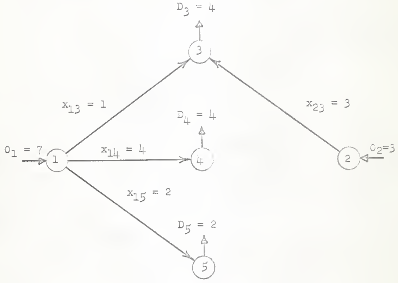


Fig. 9. Initial basic feasible solution.

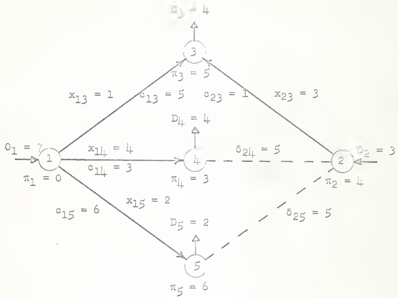


Fig. 10. Feasible solution with implicit prices.

solution is the optimal basic feasible solution.

Note: From steps 3 and 4 it is confirmed that the solution in Fig. 10 is the optimal basic feasible solution. Thus the optimal solution is:

Basic variable	From city	To city	No. of units transported
x_{13}	1	3	1
x_{14}	1	4	4
x_{15}	1	5	2
x_{23}	2	3	3

and the total minimum cost

$$= 5 \times 1 + 3 \times 4 + 6 \times 2 + 1 \times 3$$

$$= 32 \text{ units}$$

which is the same as the one obtained by the other methods.

3.5. Discussion of Methods of Section 3

The example transportation problem used to illustrate the various methods is small in size. Thus the advantages of the various methods may not be obvious. In general, to solve a transportation problem, the simplex algorithm takes more iterations than transportation algorithm. The method which depends on the discrete maximum principle is inefficient, particularly since the problem can be solved with the simple transportation algorithm. The problem is visualized easily using the method of networks. The fastest and simplest method seems to be the transportation algorithm, but there is one-to-one correspondence between this method and the method of networks. In some class of the transportation problems such as the transshipment problems, the method of networks decreases the number of quantities to be calculated and thus may actually be faster than the transportation algorithm. This comparison provides the basis of the discussion in section 4.

4. TRANSSHIPMENT PROBLEM

A transshipment problem is a special type of a transportation problem where there are many possible routes for shipping between an origin and a destination. Since there are several possible routes from an origin i to a destination j, it is possible that there may be no direct route and the commodity will have to be transported through intermediate points where neither supply nor demand exists.

The various quantities and terms of the problem are defined in the following manner. Let n be the total number of nodes, which includes the origins, the destinations, and the intermediate nodes. Let

O_i = net supply at node i, i = 1, 2, . . ., n

D_j = net demand at node j, j = 1, 2, . . ., n

$c_{ij} = c_{ji}$ = cost of shipping a unit from node i directly to node j

$c_{ii} = c_{jj} = 0$

x_{ij} = quantity shipped from node i to node j

x_{jj} = quantity transshipped through j to some other node.

If $c_{ij} = \infty$, this implies that direct shipment between nodes i and j is impossible.

A condition of the problem is that the total supply equals the total demand; that ia,

$$\sum_{i=1}^{n} O_i = \sum_{j=1}^{n} D_j \quad (25)$$

Thus the problem is one of determining the quantity of the commodity supplied from origin i to destination j, $x_{ij} \geqslant 0$, which minimizes the objective function,

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij}, \quad \text{where } i \neq j \tag{26}$$

subject to the supply constraints,

$$\sum_{\substack{j=1 \\ i \neq j}}^{n} x_{ij} - x_{ii} = O_i, \quad i = 1, 2, \ldots, n \tag{27}$$

which states that the total amount shipped out of node i less amount through node i equals net supply from node i and the demand constraints,

$$\sum_{\substack{i=1 \\ i \neq j}}^{n} x_{ij} - x_{jj} = D_j, \quad j = 1, 2, \ldots, n \tag{28}$$

which states that the total amount shipped in node j less amount through node j equals net demand at node j.

It is noted that there are (2n - 1) basic variables x_{ij}. An example problem consisting of eight nodes, two source points, three demand points, and three intermediate points, is given below. It will be set up as a linear programming problem, and will be solved by both the transshipment algorithm and the method of networks.

An Example. There are 45 lots of canned tuna in Seattle and 30 in Los Angeles. Of these, 25 are to be sent to Boston, 35 to New York, and 15 to Miami. The shipping costs per lot between the corresponding cities are illustrated below:

	Denver	Kansas City	Chicago	Boston	New York	Miami
Seattle	11	*	20	*	*	*
Los Angeles	10	15	20	*	*	*
Denver	0	5	10	25	*	*
Kansas City	5	0	10	*	*	15
Chicago	10	10	0	13	12	11
New York	*	*	12	5	0	*

*No direct route available.

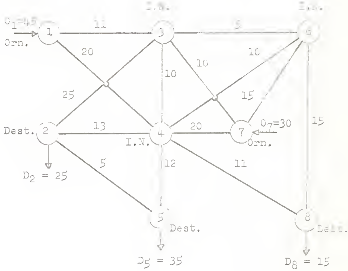
The problem is one of finding the optimal shipping routes which will minimize the total shipping costs.

Before discussing the different methods for solving this problem it is helpful to visualize the problem as the network problem shown in Fig. 11.

4.1. Method I, Simplex Algorithm

As in the case of the transportation problem, using equations (26), (27), and (28), the transshipment problem can be formulated as a linear programming problem and solved by the simplex algorithm. The problem as formally stated becomes one of determining the $x_{ij} \geqslant 0$ which minimize the objective function,

$$z = \sum_{\substack{i=1 \\ i \neq j}}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij}, \text{ where } i = 1, 2, \ldots, n \quad (29)$$
$$j = 1, 2, \ldots, n$$



Orn. : Origin

Dest.: Destination

I.N. : Intermediate node

1. Seattle	5. New York
2. Boston	6. Kansas City
3. Denver	7. Los Angeles
4. Chicago	8. Miami

Fig. 11. Network of transshipment problem with c_{ij}, O_i, and D_j.

subject to the supply constraints,

$$\begin{aligned}
&-x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{18} = 45 \\
&x_{21} - x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} = 0 \\
&\cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \qquad (30) \\
&x_{71} + x_{72} + x_{73} + x_{74} + x_{75} + x_{76} - x_{77} + x_{78} = 30 \\
&x_{81} + x_{82} + x_{83} + x_{84} + x_{85} + x_{86} + x_{87} - x_{88} = 0
\end{aligned}$$

end the demend constraints,

$$\begin{aligned}
&-x_{11} + x_{21} + x_{31} + x_{41} + x_{51} + x_{61} + x_{71} + x_{81} = 0 \\
&x_{12} - x_{22} + x_{32} + x_{42} + x_{52} + x_{62} + x_{72} + x_{82} = 25 \\
&\cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \qquad (31) \\
&x_{15} + x_{25} + x_{35} + x_{45} - x_{55} + x_{65} + x_{75} + x_{85} = 35 \\
&\cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\
&x_{18} + x_{28} + x_{38} + x_{48} + x_{58} + x_{68} + x_{78} - x_{99} = 15
\end{aligned}$$

The total number of dependent equations in (30) and (31) ere 16, and eliminating any one equation leads to a set of 15 independent constraint equetions. Thus the linear programming problem reduces to a problem with 15 equations and 79 variables (64 ective variables + 15 ertificial variebles).

The problem given in this form (equetions (29), (30), and (31)) is easily solved using simplex algorithm as was shown for the transportation problem in section 3.1 (Dantzig (2) or Hedley (8)). This problem wes solved on an IBM 1620 computer and the optimal solution was obtained efter 19 iterations.

4.2. Method II, The Transshipment Algorithm

This method is precisely the transportation algorithm explained in section 3.2, but is known as the transshipment algorithm when used to solve transshipment problems. The only difference is that in the transshipment algorithm an initial basic feasible solution is found by forming a network tree, as noted in step 1 of section 3.4, in lieu of the method suggested in step 1 of section 3.2.

The steps of the algorithm can be summarized as follows:

Step 1. Find an initial basic feasible solution forming a tree (refer to step 1, section 3.4), and form a tableau.

Step 2. Determine a set of 2n arbitrary numbers, u_i and v_j, such that for each occupied cell (i, j), $c_{ij} = u_i + v_j$. Note that these arbitrary numbers u_i, v_j are the same implicit prices as before.

Step 3. Calculate θ_{ij}, which is known as the relative cost factor, for each of the unoccupied cells (i, j) such that $\theta_{ij} = c_{ij} - (u_i + v_j)$.

Step 4. Examine the values of θ_{ij}; if all are nonnegative, then this solution is optimal. If one or more θ_{ij} are negative, then this solution is not optimal and a change in the existing solution is possible which will reduce the cost.

Step 5. Find a path to the cell which has a negative θ_{ij} (generally most negative) by horizontal and vertical movements along the shortest path of occupied cells.

Step 6. In the path determined at step 5 find the maximum assignment for each horizontal move, then the maximum assignment for each vertical move and select the minimum of these values, and assign it to the cell with the negative θ_{ij} in step 5.

Step 7. Adjust the assignments on this path so that all demands and supplies are compatible.

Step 8. Go to step 2 and repeat the procedure until an optimal assignment is found according to the criterion in step 4.

In solving the example problem by this method, the initial basic feasible solution is found according to step 1, as shown in Fig. 12. Note that the tree evaluates (n - 1) basic variables out of a possible (2n - 1) basic variables, where n is the total number of nodes present in the problem. Thus in the above example where n equals 8, seven basic variables are obtained from the tree in Fig. 12 and the other eight basic variables are the transshipment variables themselves, that is, x_{11}, x_{22}, . . ., x_{88}. From this feasible solution the elements of the transshipment algorithm are examined and are shown in Table 9.

The implicit prices u_i, v_j and the relative cost factors θ_{ij}, for the unoccupied cells, are determined and are illustrated in their respective positions in each cell of Table 9. In the transshipment problems note that the transshipment basic variables, which are x_{11}, x_{22}, . . ., x_{88}, can be negative since they do not contribute to the objective function because of their zero cost coefficients, $c_{11} = c_{22}$, . . ., $= c_{88} = 0$. Thus it is seen from Table 9 that x_{33}, x_{44}, and x_{66} are negative

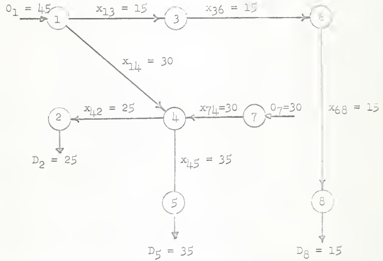


Fig. 12. Initial basic feasible solution (step 1).

Table 9. Transshipment tableau, initial basic feasible solution.

DESTINATION

ORIGIN

	1	2	3	4	5	6	7	8	O_i Sup. u_i
1	0 0	☒	15 - 11	30 + 20	☒	☒	☒	☒	45 0
2	☒	0 0	25 25	46 13	3 5	☒	☒	☒	0 -33
3	0 11	3 25	-15 + 0	1 10	☒	15 - Drop 5	21 10	☒	0 -11
4	20 20	25 13	19 10	-60 0	35 12	14 10	40 20	0 11	0 -20
5	☒	4 5	☒	24 12	0 0	☒	☒	☒	0 -32
6	☒	☒	10 5	6 10	☒	-15 0	21 15	15 15	0 -16
7	☒	☒	-1 10	30 - 20	☒	+ -1 Ent.15	0 0	☒	30 0
8	☒	☒	☒	22 11	☒	30 15	☒	0 0	0 -31
Dem. D_j	0	25	0	0	35	0	0	15	75
v_j	0	33	11	20	32	16	0	31	

quantities. It is observed that $u_i = -v_j$ when $c_{ij} = 0$, where $i = j$ for each of the transshipment variables.

Observing in the above example that two of the relative cost factors are negative, they are θ_{73} and θ_{76}, thus according to step 4 this solution is not the optimal solution. Since both θ_{73} and θ_{76} are equal to -1, either variable may be selected to enter the basis; in this instance x_{76} is selected. Continuing at step 5, a path to the cell (7, 6) is drawn by horizontal and vertical movements along the shortest path of occupied cells, as shown in Table 9.

According to step 6, the value of x_{76} to be assigned is determined to be 15, and the variable x_{36} is dropped to maintain feasibility of supply and demand. This, then, is a new basic feasible solution which is checked for optimality by repeating the procedure from step 2. The second iteration for the example is illustrated in Table 10. The basic solution is not yet optimal as it is seen that the relative cost factor for x_{73} is negative. After drawing the stepping stone path it is observed that the only change possible is the interchanging of the arc $(1 \rightarrow 3)$ and $(7 \rightarrow 3)$. Thus the new basic variable to enter is x_{73} which will have a value of zero since the variable to leave the basis is $x_{13} = 0$, as shown in Table 10. The only change in implicit prices is for u_3 and v_3 which are -10 and 10, respectively. The resulting basis thus formed is the optimal solution since all of the θ_{ij} values are nonnegative. Summarizing, then, the optimal solution is,

Table 10. Transshipment tableau, second iteration.

DESTINATION

		1	2	3	4	5	6	7	8	Sup. O_i u_i
O	1	0 0	X	0 Drop − 11	45 + 20	X	X	X	X	45 0
R	2	X	0 0	47 25	26 13	6 5	X	X	X	0 -33
I	3	22 11	3 25	0 0	1 10	X	1 5	21 10	X	0 -11
G	4	40 20	25 13	19 10	-60 0	35 12	15 10	40 20	1 11	0 -20
I	5	X	4 5	X	24 12	0 0	X	X	X	0 -32
N	6	X	X	9 5	5 10	X	-15 0	30 15	15 15	0 -15
	7	X	X	+ -1 Enter 10	15 - 20	X	15 15	0 0	X	30 0
	8	X	X	X	21 11	X	30 15	X	0 0	0 -30
Dem.	D_j	0	25	0	0	35	0	0	15	75
	v_j	0	33	11	20	32	15	0	30	

Basic variable (those contribute to the objective function)	From city	To city	No. of lots transported
x_{14}	Seattle	Chicago	45
x_{73}	Los Angeles	Denver	0
x_{74}	Los Angeles	Chicago	15
x_{76}	Los Angeles	Kansas City	15
x_{42}	Chicago	Boston	25
x_{45}	Chicago	New York	35
x_{68}	Kansas City	Miami	15

and the minimum shipping cost = z

$$= 45 \times 20 + 15 \times 20 + 15 \times 15 + 25 \times 13 + 35 \times 12$$
$$+ 15 \times 15 = 2395 \text{ units.}$$

4.3. Method III, Method of Networks

In this section the transshipment problem is solved employing the method of networks explained in section 3.4. This method, of course, has a one-to-one correspondence with the transshipment algorithm but in lieu of using a matrix format the quantities are directly calculated on the network graph itself.

The steps of the method can be summarized as follows:

Step 1. Find an initial basic feasible solution forming a tree (refer to step 1, section 3.4).

Step 2. Compute the implicit price π_i such that $\pi_j - \pi_i = c_{ij}$ is true for each arc $(i \rightarrow j)$ of the tree in step 1.

Step 3. Calcul[illegible] the rel[illegible] [illegible]tor, δ_{ij}, such that $\delta_{ij} = c_{ij} - (\pi_j$ [illegible] $_i)$ is t[illegible] for ea[illegible] [illegible]c (i → j) which is not part of the t[illegible] but exi[illegible] in [illegible]he [illegible] nal network.

Step 4. Exam[illegible] the values [illegible] δ_{ij}, if [illegible]y are all nonnegative, this so[illegible]n is optim[illegible] If at [illegible]t one δ_{ij} is negative, then t[illegible] [illegible]imal [illegible] to step 5.

Step 5. Draw the arc [illegible] [illegible]ue of δ_{ij} and include that arc in the [illegible] [illegible]rally t[illegible] negative δ_{ij} is selected). The values of the basic varia[illegible] around the loop are adjusted with a value assigned [illegible] the [illegible] arc entering the basis. The appropriate arc is dropped and a new tree is formed.

Step 6. Using the new basic feasible solution, the procedure is repeated from step 2 until the optimal solution is obtained. The transshipment problem which was treated in sections 4.1 and 4.2 is again illustrated below using the method of networks. The example consists of eight nodes which includes two origins, three destinations, and three intermediate points. The cost of shipping the demand requirements and the quantities available are as shown in Fig. 11.

An initial basic feasible solution including the implicit prices and the relative cost factors from steps 1, 2, and 3 is shown in Fig. 13. It is noted in Fig. 13 that the optimality criterion is not met with the initial basic feasible solution because $\delta_{73} = -1$. This indicates that it is profitable to transport through arc (7 → 3). Nevertheless, when arc (7 → 3) is introduced, the subnetwork does not construct a tree since it contains a loop shown in Fig. 14. Since the values of

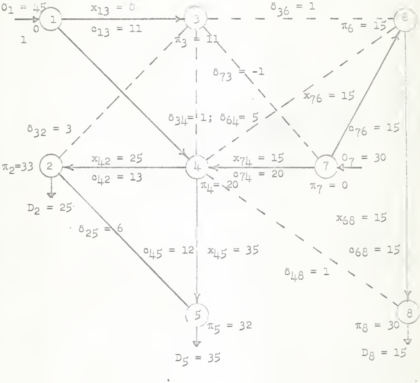


Fig. 13. Initial basic feasible solution with implicit prices, π_i, and relative cost factors, δ_{ij}.

variables constituting the loop are affected by shipping through arc $(7 \rightarrow 3)$, we need only consider these. From the loop it is seen that the maximum allowable value of new variable x_{73} is zero, since the minimum value of the arc with negative sign is zero. To avoid the problem of the loop and to allow flow through $(7 \rightarrow 3)$, the arc $(1 \rightarrow 3)$ is to be dropped. This then forms a new basic feasible solution, as is shown in Fig. 15. The optimality criterion is met with this solution since all the π_i's remain the same and all the other δ_{ij}'s are nonnegative. Therefore this solution to example in Fig. 11 is the optimal basic feasible solution, and is the same as the one obtained in section 4.2.

4.4. Discussion of Methods of Section 4

In solving the transshipment problem by the various methods, it is seen from this example and which is generally true that the simplex algorithm requires more variables and more iterations than the transshipment algorithm. To solve a problem manually the method of network has several advantages over the simplex algorithm and the transshipment algorithm. First the number of quantities to be calculated are always less using the method of network. Secondly, it is easy to visualize the problem through the network. Specifically, comparing the methods applied to the example considered in this section, it is seen that the linear programming problem, where the simplex algorithm can be applied, consisted of 15 equations and 79

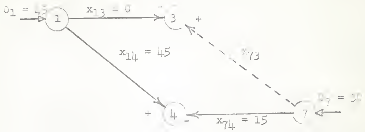


Fig. 14. To find the value of new basic variable, x_{73}.

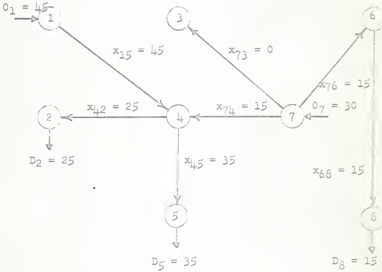


Fig. 15. The optimal basic feasible solution.

veriebles (64 ective variebles plus 15 ertificiel veriebles) when the totel number of nodes is 8. It took 19 iteretions to obtein the optimal solution to this problem, working on an IBM 1620 computer. Employing the trensshipment elgorithm, the number of quentities to be evelueted et eech iteretion is 36, end the optimal solution required three iteretions, whereas working the seme exemple with the method of networks there are only 14 quantities to be celculated at eech iteration and the optimel solution required one iteretion. Thus it eppears as the size of problem increases, the method of network significently reduces the number of calculetions in obtaining the optimal solution.

5. MAXIMAL FLOWS IN NETWORKS

The network problems previously considered had infinite arc capacities where the objective was to minimize cost or maximize profit. The latter is accomplished by considering the negative values of the profit coefficient at each arc. The problems considered in this section have finite arc capacities and consist of a single source and sink connected by several intermediate nodes, as is illustrated in Fig. 16. In problems of this type the source is designated node 1 and the sink as node n. Thus in Fig. 16, node 1 is the source and node 5 is the sink. The flow may represent such things as fluids, electricity, and automobiles with physical units per unit time. The flow capacities, d_{ij}, in each direction are designated along each arc near the node at which the flow originates. The flow is assumed to be steady state, that is, the flow does not change with time.

It is convenient at this point to define some necessary terms and quantities.

Let $x_{ij} \geqslant 0$ denote the quantity of flow from node i to node j where the following capacity constraints hold:

$$\begin{aligned} 0 \leqslant x_{ij} \leqslant d_{ij} \\ 0 \leqslant x_{ji} \leqslant d_{ji} \end{aligned} \qquad (32)$$

and the conservation of flow equations hold at each node k:

$$\sum_{i} x_{ik} - \sum_{j} x_{kj} = 0 \qquad (33)$$

with $k = 2, 3, \ldots, n - 1$.

Equation (33) states that the sum of flows into node k is balanced by the sum of flows from k and where node k is not the source or the sink.

From Fig. 16 and equation (32), the following relations are true:

$$0 \leq x_{12} \leq 5, \qquad 0 \leq x_{21} \leq 0,$$

$$0 \leq x_{23} \leq \infty, \qquad 0 \leq x_{43} \leq 2,$$

and if there is a flow of 2 through arc (2 → 4) and flow of 3 through arc (3 → 4), then the flow through arc (4 → 5) must be 2 + 3 = 5. Similarly, if f is the flow into the source from outside the network, then by definition the net flow at the source is,

$$f + \sum_{i} x_{i1} - \sum_{j} x_{1j} = 0 \qquad (34)$$

and consequently at the sink

$$\sum_{i} x_{in} - \sum_{j} x_{nj} - f = 0 \qquad (35)$$

The equations (33), (34), and (35) correspond to the flow in the arcs at the nodes in the network.

Maximal Flow Problem. There is a class of flow problems which are governed by the above relationships where the objective is to obtain the maximal flow through the network. In particular, the maximal flow problem is one of determining $x_{ij} \geq 0$, the flow through arc (i → j), so that the input f at the source is a maximum and where equations (32), (33), (34), and (35) govern the system. A feasible flow or flow path in a network

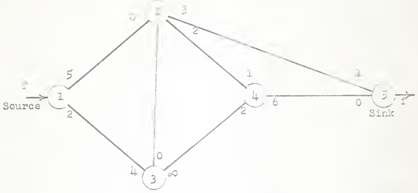


Fig. 16. A network with directed arc capacities.

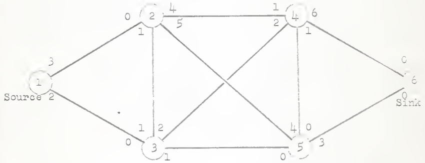


Fig. 17. A maximal flow example.

is said to be maximal if the flow from source to sink is finite, and where no other feasible flow or flow path yields a larger flow from source to sink. There are several methods for solving the maximal flow problem and three are discussed in the following sections.

5.1. Method I, Chains of Positive Arc Capacities Joining Source to Sink

The first method of solving these problems is one which utilizes the chains of positive arc capacities. The method consists of finding the chains connecting the source to the sink which have positive arc capacities. This method is direct application of the following theorem:

Theorem 2. If a maximal flow exists and the capacities are integers, the method will construct only a finite number of positive chain flows, whose algebraic sum is the maximal flow (2). The steps of this method are summarized as follows.

Step 1. Join the arcs of positive capacity until a chain of arcs connects the source to sink.

Step 2. Determine the smallest capacity f of an arc on the chain found in step 1, then f is the flow in this chain, contributing to the final flow from source to sink.

Step 3. Decrease the capacity of each directed arc on the chain in step 1 by amount f found in step 2, and increase the same amount in the opposite direction.

Step 4. The process is repeated for the network formed in step 3 by starting again at step 1.

Step 5. This procedure is repeated until no more chains of positive arc capacity joining source to sink are possible. The sum of the flows in each chain obtained in step 2 is the maximal flow in the network. This method is illustrated by the following example.

An Example. The problem considered is illustrated as a network in Fig. 17. The capacities of the arcs in each direction are indicated on the network. The problem is one of finding the maximal flow through the network.

The method of solution is begun in accordance with step 1 by obtaining a chain with positive arc capacity joining the source to the sink. This is facilitated by forming a tree of arcs with positive capacities and selecting the appropriate chain. One possible tree is shown in Fig. 18, which leads to the chain $(1 \rightarrow 2)$, $(2 \rightarrow 4)$, $(4 \rightarrow 6)$, with capacities (3, 4, 6). As directed by step 2, the minimum capacity among the arcs of this chain is determined to be 3 for arc $(1 \rightarrow 2)$. Thus the flow initiated along this chain is $f_1 = 3$, which is shown at the middle of the arcs; the arrows indicate the direction of flow along the chain (refer to Fig. 18). In step 3, the capacity of each arc is adjusted by subtracting $f_1 = 3$ from the capacity at the base of the arrow and adding it to the capacity at the point of the arrow. This result is expressed in Fig. 19. The procedure is repeated again for the new network starting at step 1. The chain of arcs with positive capacity in the network of Fig. 19 is $(1 \rightarrow 3)$, $(3 \rightarrow 4)$, $(4 \rightarrow 6)$, with capacities (2, 2, 3). Thus the flow $f_2 = 2$ is permissible along this chain. Continuing

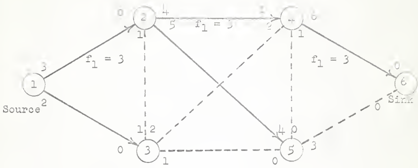


Fig. 18. Initial tree with positive chain flow.

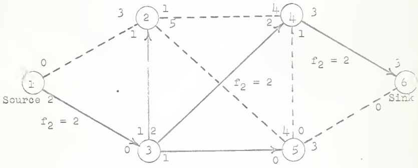


Fig. 19. Adjusted arc capacities and second chain.

the capacities are adjusted egain which results in the final network as shown in Fig. 20. The procedure is stopped because no additional cheins of positive arc capecity connecting 1 end 6 exist. Finally, the total maximel flow in the network is the algebraic sum of the chain flows as obtained from Figs. 18 and 19, and illustrated in Fig. 21. Thus the maximal flow

$$f = f_1 + f_2 = 3 + 2 = 5.$$

The above method and the methods which follow in this section are based on the max-flow min-cut theorem of Ford and Fulkerson (3, 7, 8). All of these methods lead us to the value of minimum cut which is equal to the maximel flow. Before stating the theorem the following concepts are defined.

Definition of e Cut. A cut is eny set of directed arcs containing at least one arc from every chain of positive capacity joining the source to the sink.

Definition of the Cut Value. The cut value is the sum of the capacities of the ercs of the cut.

With these definitions the theorem is stated as follows.

Theorem 3, Max-flow Min-cut Theorem. For eny network the maximal flow value from the source to the sink is equal to the minimal cut capacity of all cuts separeting the source from the sink; thet is, mex-flow value equals the min-cut velue.

In the above exemple, see Fig. 21, the set of directed arcs, $(1 \rightarrow 2)$, $(1 \rightarrow 3)$, with cepacities (3, 2), constitute a cut. The velue in this case is $3 + 2 = 5$. This is the only cut

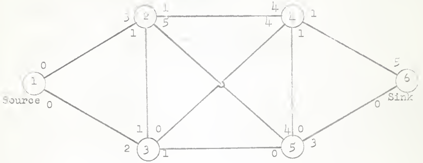


Fig. 20. Final network, no chain of positive capacity.

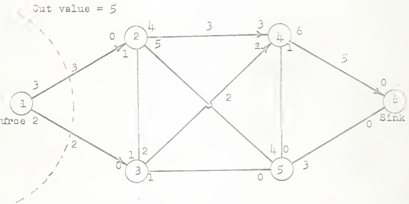


Fig. 21. The maximal flow.

in this particular case; therefore the minimum cut value is 5 which is also maximal flow value as seen above.

5.2. Method II, The Labeling Technique

The labeling technique which is used with the maximal flow problems is a more systematic way of determining whether a flow in the network exists or not. Let the excess capacity through arc $(i \rightarrow j)$ be defined by

$$g_{ij} = d_{ij} - x_{ij} \geqslant 0 \qquad (36)$$

where d_{ij} is the capacity of arc $(i \rightarrow j)$ and x_{ij} is the net flow through arc $(i \rightarrow j)$ from node i to node j. Similarly,

$$g_{ji} = d_{ji} + x_{ij}, \text{ since } x_{ji} = -x_{ij} \qquad (37)$$

where x_{ij} may have a value at the subsequent steps of the procedure but it is assumed to be zero initially. Having defined these quantities, the discussion of the method will now proceed.

The steps of the method are as listed below:

Step 1. Start by considering all the nodes which are joined to the source by arcs of positive excess capacity. On the network, label a node j with two numbers (δ_j, r_j), where $\delta_j = g_{1j}$, and $r_j = 1$. (Note that the value of δ_j is the excess capacity from the source to node j, and the value of r_j indicates the node from where one started and proceeded to label j; in the above case this node is the source, therefore $r_j = 1$.) If the sink n is labeled, go to step 4; if not, go to step 2.

Step 2. When the sink is not labeled in step 1, let j be the smellest index of the node from the set of nodes lebeled at step 1. Determine if there are eny unlabeled nodes which are joined to j by ercs of positive excess cepacity. If there are no such nodes, move to the next lowest index j end repeat the process. If some unlebeled nodes cen be reached, using the index k for these unlebeled nodes, label eech as follows:

$$\delta_k = \min(g_{jk}, \delta_j), \quad r_k = j \qquad (38)$$

The label δ_k on node k gives the minimum excess capecity of the two ercs which form the peth from the source to j and from j to k. The label r_k indicetes the node from which one proceeded to label k. After heving lebeled all possible nodes k with the lowest index j, if sink is labeled go to step 4; if not, go to step 3.

Step 3. Select the node with lowest index from the set of nodes denoted by k in step 2; look for the unlabeled nodes that are joined to this node by ercs of positive excess capecity. Let these nodes be denoted by the index q. Lebel the nodes q in the similar wey as before, that is,

$$\delta_q = \min(g_{kq}, \delta_k), \quad r_q = k \qquad (39)$$

This process is repeated until, in e finite number of steps, one reaches one of the two following states:

(1) No edditional nodes cen be labeled, end the sink is not lebeled.

(2) The sink is labeled.

If state (1) is reached, the existing flow is maximal. If state (2) is reached, the existing flow can be increased; go to step 4.

<u>Step 4</u>. If δ_n is the label on the sink, then determine the chain with this positive flow which has been followed in moving from the source to the sink. Work backwards and trace the path, noting the second label on the nodes which indicates the preceding node.

<u>Step 5</u>. Let g_{ij} be the excess capacities of the arcs in the chain which lead to the labeling of the sink, in the direction in which one moved in going from source to sink. Then consider the new network whose excess capacities are,

$$\begin{aligned} g'_{i'j'} &= g_{ij} - \delta_n \\ g'_{j'i'} &= g_{ji} + \delta_n \end{aligned} \tag{40}$$

for each arc ($i' \rightarrow j'$) in the chain in step 4

$$g'_{i'j'} = g_{ij}, \qquad g''_{ji} = g_{ji} \tag{41}$$

for arcs not in the above chain.

<u>Step 6</u>. Repeat the entire labeling process considering the new network obtained in step 5.

<u>Step 7</u>. Add all δ_n obtained in step 4 to compute the maximal flow.

After a finite number of steps, one must reach state (1) of step 3, since δ_n is always strictly positive, which assumes that there does exist a maximal flow.

The example network problem considered in section 5.1 will now be solved by the labeling technique.

Assuming initial flow $x_{ij} = x_{ji} = 0$, from equations (36) and (37), the excess capacities for the arcs of the network are illustrated in Fig. 23. Since $x_{ij} = x_{ji} = 0$, then $g_{ij} = d_{ij}$. According to step 1, the nodes joined to the source by arcs of positive capacity are 2 and 3, as $g_{12} = 3$ and $g_{13} = 2$. These nodes are labeled with (3, 1) and (2, 1), respectively; the first number indicates δ_j and the second number r_j. In this case $r_2 = r_3 = 1$ since they are reached from the source. According to step 2, the lowest index node from the nodes labeled above, that is, between 2 and 3, is node 2. The nodes which are not labeled and are joined to node 2 by arcs of positive excess capacity are 4 and 5. Node 3 is ignored as it has been already labeled. Nodes 4 and 5 are labeled; for example, for the node 4, $\delta_4 = \min(\delta_2, g_{2,4}) = \min(3, 4) = 3$ and $r_4 = 2$ since node 4 is reached from node 2 (refer to Fig. 23). The sink is still unlabeled; hence the second lowest index node among those labeled at step 1 is considered. This is node 3, but there are no unlabeled nodes which are joined to node 3 by arcs of positive excess capacity. Therefore according to step 3, the lowest index node from those labeled at second step is considered, that is, the node 4. The only unlabeled node connected to the node 4 by an arc of positive capacity is node 6 which has a label (3, 4). Thus the sink is labeled; therefore the flow in the network can be increased from zero by $\delta_{n=6} = 3$ (refer to Fig. 23). The chain is traced working backwards as instructed by step 4: the sink, node 6, is reached from node 4, node 4 is reached from node 2, and node 2 is

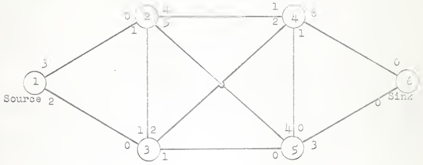


Fig. 22. Maximal flow problem with arc capacities.

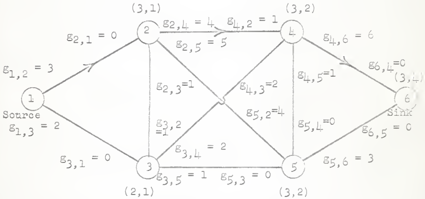


Fig. 23. Maximal flow problem with excess capacities and nodes labeled.

reachad from noda 1, which is the source. Therefore the chain is $(1 \rightarrow 2)$, $(2 \rightarrow 4)$, $(4 \rightarrow 6)$. Now utilizing aquations (40) and (41), capacitias of tha arcs ara adjustad and tha naw network is formad as shown in Fig. 24. The whole procadura from step 1 is rapeated again. Labels on the nodas are also shown in Fig. 24. The flow in tha network is again increased by $\delta_{n=6} = 2$ through the chain $(1 \rightarrow 3)$, $(3 \rightarrow 4)$, $(4 \rightarrow 6)$, as illustratad in Fig. 24. At this point the total flow is equal to $3 + 2 = 5$. Again the nacassary adjustments in tha capacities ara made which rasults in new natwork as shown in Fig. 25. It is observed that no node can ba labaled from tha source, node 1, and thus the sink, node 6, is unlabaled. This indicates that stata (1) of step 3 is raached, and consaquently the existing flow is maximal. Thus tha maximal flow for this example is 5, which is the sama as that obtainad in saction 5.1.

5.3. Mathod III, The Matrix Solution

Tha matrix method which is usad with the max-flow problems, is based on the fact that any natwork with n nodas can be reprasentad by an equivalent matrix of order n. The basic princ̣ipla of this mathod is pracisaly the same as that of labeling technique. The steps of tha method can ba summarized as follows:

Step 1. Using the values of excess capacities as the alamants, form an n x n matrix whera n is tha total number of nodes of the natwork and allow two additional columns for δ_j and r_j.

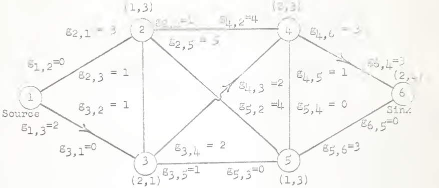


Fig. 24. Adjusted excess capacities with nodes labeled.

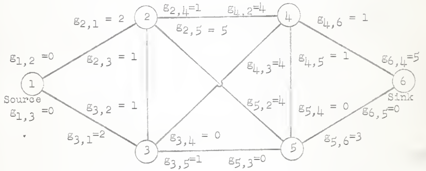


Fig. 25. Final network, no labeling is possible.

<u>Step</u> 2. Inspect row 1 for $g_{1j} > 0$, write in row j under column δ these velues of g_{1j}, respectively; end under column r write a 1 which indicetes that row j is reached from the source. If the sink is not one of the j's, go to step 3; end if it is, go to step 5.

<u>Step</u> 3. Consider the row of lowest index from the set of rows lebeled in step 2. If this is row i, look for $g_{ij} > 0$ for which row j has not been labeled. Then set $\delta_j = \min(g_{ij}, \delta_i)$ end $r_j = i$. Repeat the process until all the rows lebeled at step 2 are exhausted. If the n^{th} row is labeled, go to step 5; if not, go to step 4.

<u>Step</u> 4. Repeat step 3 for the set of rows labeled in step 3. In a finite number of steps one of the following stages is reached:

(1) No additional rows can be labeled, and the n^{th} row is unlabeled.

(2) The n^{th} row is labeled.

If state (1) ia reached, the existing flow ia maximel. If state (2) is reeched, the existing flow can be increased; go to step 5.

<u>Step</u> 5. If δ_n is the label for the n^{th} row, then determine the peth by trecing beckwards on the metrix from the velue of r for the connected nodes. The increase in flow is δ_n.

<u>Step</u> 6. Circle the elements on the path in step 5; note diagonally opposite elements and drew a square eround them. Subtract the increased flow, δ_n, from each circled element and edd δ_n to each element with a square.

Step 7. Repeat the process for the new matrix formed at step 6.

To illustrate the above method, the excess capacities from Fig. 23 are noted in a matrix as shown in Table 11. Then according to step 2, the positive elements in row 4 correspond to columns 2 and 3 with $g_{12} = 3$ and $g_{13} = 2$, respectively. These values are noted in the δ column by rows 2 and 3 and r in both cases is equal to 1 since the source is row 1. This terminates step 2. Since δ_n has not been assigned in step 2, then according to step 3, the row with the lowest index from the rows labeled in step 2 is selected and is the second row. The columns 4 and 5 can be reached since the elements are $g_{24} = 4$ and $g_{25} = 5$, and where the rows 4 and 5 have not been labeled. Accordingly, the rows 4 and 5 are labeled with $\delta_4 = \min(g_{24}, \delta_2) = \min(4, 3) = 3$, $r_4 = 2$, and $\delta_5 = \min(g_{25}, \delta_2) = \min(5, 3) = 3$, $r_5 = 2$, respectively. Continuing, the next lowest row index labeled in step 2 is row 3. But there is no column which has a row which is unlabeled. Thus step 3 is terminated. It is noted that the n^{th} row is not labeled; therefore, according to step 4, the first row which has been labeled at step 3 and is to be considered is row 4. The column for which the row is unlabeled is column 6. Now row 6 is labeled by $\delta_6 = \min(g_{46}, \delta_4) = \min(6, 3) = 3$, $r_6 = 4$. Finally, $\delta_{n=6} = 3$; therefore flow can be increased by 3. (Refer to Table 11.) Then according to step 5, the path is traced as follows: $r_6 = 4$, go to row 4 and circle element $g_{46} = 6$. Then note that $r_4 = 2$ and go to row 2 and circle the element $g_{24} = 4$. Note that $r_2 = 1$ and go

Table 11. Matrix of excess capacities (first iteration).

Nodes	1	2	3	4	5	6	Labels		Step
1	-	③	2	0	0	0	δ	r	
2	$\boxed{0}$	-	1	④	5	0	3	1	2
3	0	1	-	2	1	0	2	1	2
4	0	$\boxed{1}$	2	-	1	⑥	3	2	3
5	0	4	0	0	-	3	3	2	3
6	0	0	0	$\boxed{0}$	0	-	3	4	4

Table 12. Adjusted new matrix (second iteration).

Nodes	1	2	3	4	5	6	Labels		Step
1	-	0	②	0	0	0	δ	r	
2	3	-	1	1	5	0	1	3	3
3	$\boxed{0}$	1	-	②	1	0	2	1	2
4	0	4	$\boxed{2}$	-	1	③	2	3	3
5	0	4	0	0	-	3	1	3	3
6	0	0	0	$\boxed{3}$	0	-	2	4	4

to row 1 and circle the element $g_{12} = 3$. Thus the path is $1 \rightarrow 2$, $2 \rightarrow 4$, $4 \rightarrow 6$, and is given by the circled elements (refer to Table 11). From step 6, the diagonally opposite elements $g_{2,1}$, $g_{4,2}$, $g_{6,4}$ are marked by squares. Continuing, δ_6 is subtracted from each circled element and is added to the elements in squeres, thus forming the new matrix which is illustrated in Teble 12. The process is repeated to determine if the flow cen be increased, end continues until no more rows cen be labeled and the nth row remains unlebeled. This then indicetes that the flow hes reeched its meximal value. The second iteretion of the example is illustreted in Teble 12 end the finel is illustreted in Table 13. The sum of the elements in the lest row of the matrix gives the value of maximal flow. From Teble 13 it is noted that the meximel flow for the above exemple is 5 end is the same es the one obtained for this exemple by other methods.

Table 13. Finel metrix (meximal flow = 5).

Nodes	1	2	3	4	5	6	Labels		Step
1	-	0	0	0	0	0	δ	r	
2	3	-	1	1	5	0			
3	2	1	-	0	1	0			
4	0	4	4	-	1	1			
5	0	4	0	0	-	3			
6	0	0	0	5	0	-			

5.4. Discussion of Methods of Section 5

All the above methods used to solve the maximal flow problem are the application of max-flow min-cut theorem, which apply to steady-state flow. When the network is large the matrix solution eliminates the need of drawing the network for each iteration which saves a lot of time. The example problem considered consists of only one source and one sink but it is possible to have many sources and sinks. However, such problem can be reduced to the case of one source and one sink (Ford, Fulkerson (6), Hadley (8)).

SUMMARY AND CONCLUSIONS

The purpose of this report is to review the technique for solving transportation, transshipment, and maximal flow problems. Some definitions and theorems concerning the theory of network and maximal flow problems are presented. In section 3, several methods are discussed for solving a simple transportation problem. These are:

1. Simplex algorithm
2. Transportation algorithm
3. Discrete maximum principle
4. Method of networks.

The methods discussed in this report are only applicable to problems with linear cost functions except the discrete maximum principle. It is noted that transportation problems involve shipping commodities from supply points to demand points such that the shipping costs are minimized. The transshipment problem contrasts with this in that intermediate points may be utilized to minimize the shipping costs; consequently the transshipment problem is a special case of the transportation problem.

The method of chains of positive arc capacities, the labeling technique, and the matrix method, which are all refinements of the maximal flow theory of networks, are reviewed in section 5, as techniques for solving maximal flow problems. These problems are characterized by maximizing the flow from a source to a destination through intermediate arcs which may have limited capacities.

In section 3, an example of a transportation problem with two sources and three destinations was solved by the various methods. The number of iterations required by each method is:

Method	Iterations
Simplex algorithm	4
Transportation algorithm	1
Discrete maximum principle	1
Method of networks	1

From these results and the ease of programming the algorithms for the computers, the transportation algorithm appears to be better.

In section 4, an example of a transshipment problem with two sources, three destinations, and three intermediate points was formulated and solved as a linear programming problem, and was solved using the transshipment algorithm and the method of networks. It is noted from the results that the method of networks is fastest when the problem is solved manually. But to work out the problem on computers the transshipment algorithm is the best.

In section 5, an example of maximal flow problem was solved using the method of chains of positive arc capacities, the labeling technique, and the matrix method, which are basically the application of the max-flow min-cut theorem. It seems that the matrix method in this case is the fastest one as it eliminates the need to draw the network at each iteration.

ACKNOWLEDGMENTS

I am taking this opportunity to extend my sincere thanks to Dr. F. A. Tillman, major professor and Head of the Department of Industrial Engineering, and to Dr. G. F. Schrader, professor and former Head of the Department of Industrial Engineering, for their cooperation and encouragement to make this report a success.

REFERENCES

1. Chen, S. K., L. T. Fan, C. L. Hwang, J. M. Panchal, G. F. Schrader. "The Application of the Discrete Maximum Principle to Transportation Problems with Linear and Nonlinear Cost Functions," submitted for publication (1966).

2. Dantzig, G. B. "Linear Programming and Extensions," Princeton University Press, Princeton, New Jersey, 1963.

3. Dantzig, G. B. and D. R. Fulkerson. "On the Max-flow Min-cut Theorem of Networks," Rand Research Memorandum Rm-1418-1, January, 1955.

4. ____________. "Computation of Maximal Flows in Networks," Rand Research Memorandum Rm-1489, April, 1955.

5. Fan, L. T. and C. S. Wang. "The Application of the Discrete Maximum Principle to Transportation Problems," J. Math. and Physics, 4-3, 255 (1964).

6. Ford, L. R. and D. R. Fulkerson. "Maximal Flow Through a Network," Rand Research Memorandum Rm-1400.

7. ____________. "Flows in Networks," Princeton University Press, Princeton, New Jersey, 1962.

8. Hadley, G. "Linear Programming," Addison-Wesley Publishing Company, Inc., 1963.

9. Koopmans, T. C., ed. "Cowles Commission for Research in Economics," John Wiley & Sons, Inc., New York, 1951.

10. Sasieni, M., A. Yaspan, and L. Friedman. "Operations Research--Methods and Problems," John Wiley & Sons, Inc., New York, 1964.

11. Fan, L. T. and C. S. Wang. "The Discrete Maximum Principle--A Study of Multistage Systems Optimization," John Wiley & Sons, Inc., New York, 1964.

THE THEORY OF NETWORK APPLIED TO TRANSPORTATION TRANSSHIPMENT AND MAXIMAL FLOW PROBLEMS

by

RAMESH NARAYAN HICHKAD

D. M. E., Sir B. Polytechnic Institute
Bhavnagar, Gujarat, India, 1963

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1967

The purpose of this report is to review the techniques for solving transportation, transshipment, and maximal flow problems. The methods which are discussed for solving the transportation and transshipment problems are:

1. Simplex algorithm
2. Transportation algorithm
3. Discrete maximum principle
4. Method of networks.

It is noted that a transportation problem involves shipping commodities from supply points to demand points such that the overall cost is minimized. This contrasts with the transshipment problem, in that intermediate points may be utilized to minimize the shipping cost.

The method of chains of positive arc capacities, the labeling technique and the matrix method, which are all versions of the method of networks, are reviewed as techniques for solving maximal flow problems. The objective of this type of problems is to maximize the flow from a source to a destination through intermediate arcs which may have limited capacities. Examples of transportation, transshipment, and maximal flow problems are solved using the various techniques.