THE TREE ALGORITHM FOR SOLVING NETWORK TRANSPORTATION PROBLEMS*

IWARŌ TAKAHASHI

Waseda University

(*Presented at the 17th meeting, May. 13, 1965.*
Received June, 1, 1965)

PREFACE

Methods for solving network transportation problems were already proposed by L. R. Ford & D. R. Fulkerson [1], M. Iri [2] and T. Fujisawa [6]. The basic idea of these methods, were primal dual method [3].

A new algorithm for solving network transportation problem is proposed. It is basically the dual method [4] from the view point of linear programming theory, but from the view point of network topology the idea of tree is the essential feature of our algorithm.

Applying our algorithm to the Hitchcock transportation problem, we get a new simple algorithm for solving this problem.

§ 1. THE NETWORK TRANSPORTATION PROBLEM

Formulation

Fig. 1 shows an example of the network transportation problem we shall treat. Node 1 and 2 are *sources*, and their supplies are 15 and 12 respectively. Node 6 and 7 are *sinks* and their demands are 6 and 18 respectively. The number with parentheses on each *arc* shows the capacity and the number without parentheses the cost per unit flow. The problem is to find the transportation program minimizing the total cost.

We may formulate the problem in general as follows. Let a_i be the supply of source i, b_j the demand of sink j, c_{ij} the unit cost on arc (ij), and M_{ij} the capacity of arc (ij).

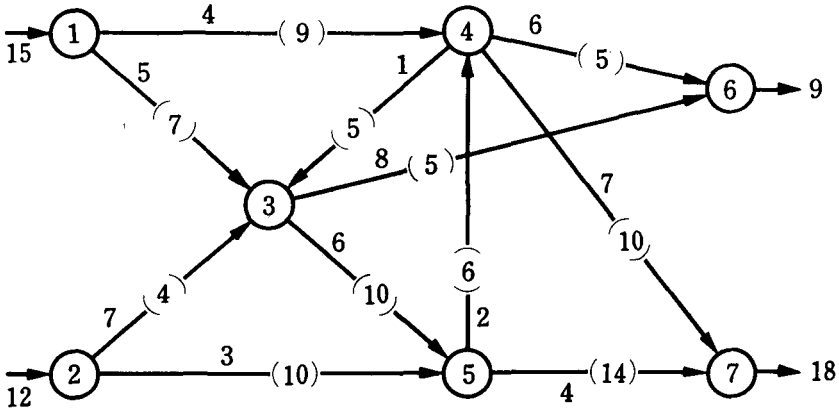


Fig. 1. A Network Transportation Problem.

We wish to minimize

$$\sum_{(ij)\in A} c_{ij}x_{ij} \tag{1.1}$$

subject to

conservation constraints:

$$\begin{cases} a_i=\sum_l x_{il} & \text{for } i\in S, \\ \sum_l x_{lj}=b_j & \text{for } j\in T, \\ \sum_l x_{li}=\sum_k x_{ik} \end{cases} \tag{1.2}$$

capacity constraints:

$$x_{ij}\leqq M_{ij} \quad \text{for } (ij)\in A, \tag{1.3}$$

nonnegativityconstraints:

$$x_{ij}\geqq 0 \quad \text{for } (ij)\in A, \tag{1.4}$$

and

$$\sum_{i\in S} a_i=\sum_{j\in T} b_j, \tag{1.5}$$

where S is the set of sources, T the set of sinks, I the set of intermediate nodes, and A the set of arcs on the given network.

We insert slack variables x_{ji} such that

$$x_{ij}+v_{ji}=M_{ij} \quad \text{for} \quad (ij)\in A, \tag{1.3'}$$

$$x_{ji}\geqq 0 \quad \text{for} \quad (ij)\in A. \tag{1.4'}$$

The problem now becomes to minimize (1.1) subject to (1.2), (1.3'), (1.4') and (1.4).

Dual Problem

The dual problem of (1.1), (1.2), (1.3'), (1.4), (1.4') is to maximize

$$\left(\sum_{j\in T} b_j u_j-\sum_{i\in S} a_i u_i\right)+\sum_{(ij)\in A} M_{ij}v_{ij} \tag{1.6}$$

subject to

$$c_{ij}-(u_j-u_i)\geqq v_{ij}, \tag{1.7}$$

$$v_{ij}\leqq 0. \tag{1.8}$$

Since M_{ij}'s are $\geqq 0$, the optimal values of v_{ij} (under the given value of u_i) are determined such that

$$v_{ij}=\begin{cases} 0 & \text{if} \quad c_{ij}-(u_j-u_i)\geqq 0, \\ c_{ij}-(u_j-u_i) & \text{if} \quad c_{ij}-(u_j-u_i)<0. \end{cases} \tag{1.9}$$

So we consider hereafter only v_{ij}'s which satisfy condition (1.9). Further we call u_i *node potentials*, and

$$\zeta_{ij}=c_{ij}-(u_j-u_i) \tag{1.10}$$

the *imputed cost* of arc (ij).

§ 2. TREE ALGORITHM

The Concept of a Tree

A *tree* is a minimum set of arcs which connects (without considering direction) all of nodes. In other words, a tree is a set of arcs which

connects all of nodes and has no loops. We call a *cotree* a set of arcs which are not on a tree.

Two important properties of trees and cotrees are

(i) One arc on a tree divides the nodes on the network into two groups,

(ii) One arc on a cotree determines one loop which consists of the arc and some arcs on the tree.

For example in Fig. 4 (i) solid arcs constitute a tree and dotted arcs a cotree. Arc (3,5) on the tree divides all nodes into labelled and unlabelled nodes. And arc (4,6) on the cotree determines a loop (4, 6, 3, 5, 7, 4) which consists of arcs (6,3), (3,5), (5,7) (7,4) on the tree and (4,6) itself.

Tree Algorithm

Our algorithm consists of three main parts.

I. Determination of the Initial Tree

We select a tree on the given network such that the imputed costs of the arcs on the tree are all $=0$ and the imputed costs on the cotree are all $\geqq 0$. Detailed procedure is as follows:

First set the potentials of all the nodes equal to zero, draw all the arcs with dotted line (see Fig. 2 (i)) and label an arbitrary node, and then repeat the following iterative process.

Let X be the set of nodes which have been labelled, and X be the set of nodes which have been unlabelled. Raise up the potentials of all the nodes in X by

$$\theta = Min\ (\zeta_{ij} : i \in X,\ \ j \in X) \tag{2.1}$$

where ζ_{ij} is the imputed cost of arc (ij) (see (1.10)).

When arc (ij) $(i \in X,\ j \in X)$ does not exist (see Fig. 2 (v)), decrease the potentials of all the nodes in X by

$$\theta' = Min\ (\zeta_{ij} : i \in X,\ \ j \in X)\,. \tag{2.2}$$

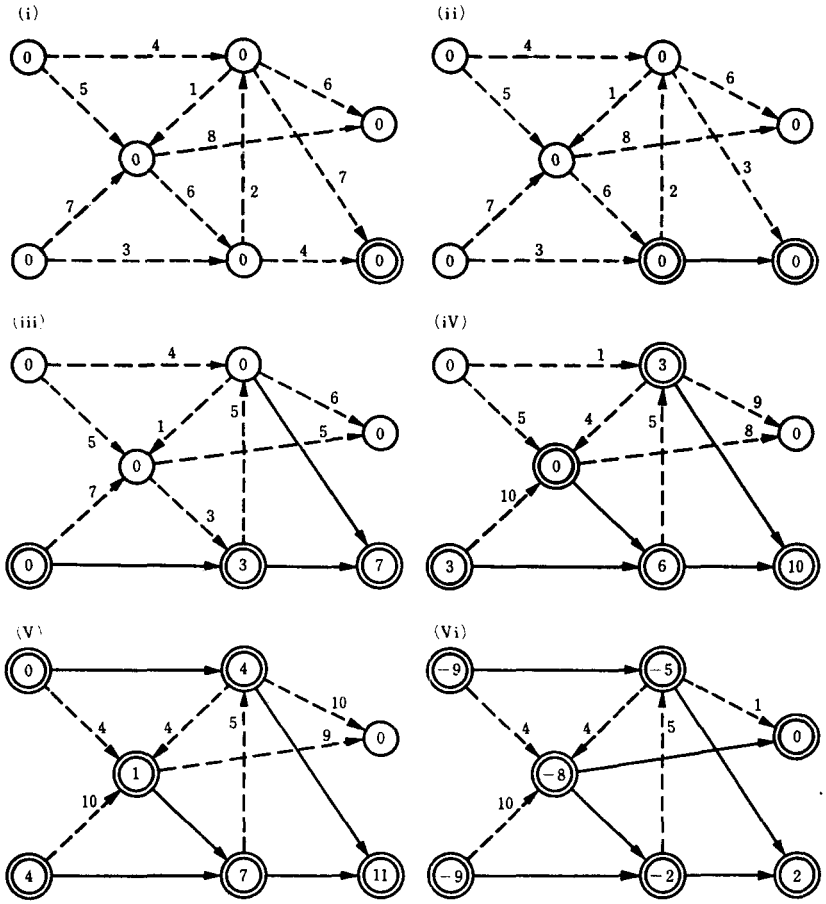


Fig. 2. Determination of Initial Tree

Draw the arcs, whose newly determined imputed costs are zero, with solid lines, and label the nodes having been connected with the solid lines (see Fig. 2 (ii)).

When all nodes have been labelled, the set of arcs with solid is the desired tree (see Fig. 2 (vi)). We will show the treatment of the degenerate case later in this §.

II. Determination of Flows on the Initial Tree

Let $x_{ij}=0$ for (ij) on the initial cotree. (see Fig. 3 (i)).

Next determine flows x_{ij} on the initial tree so that x_{ij} satisfy the conservation constraints (1.2), neglecting the capacity constraints (1.3) and the nonnegativity constraints (1.4). The actual procedures are as follows: Let T denote the set of arcs on the initial tree. Select an arbitrary *end node** of T, and determine the flow value of the arc incident to the end node to satisfy the conservation constraints on the mode. Then delete the arc from T. Repeat above process until the values of

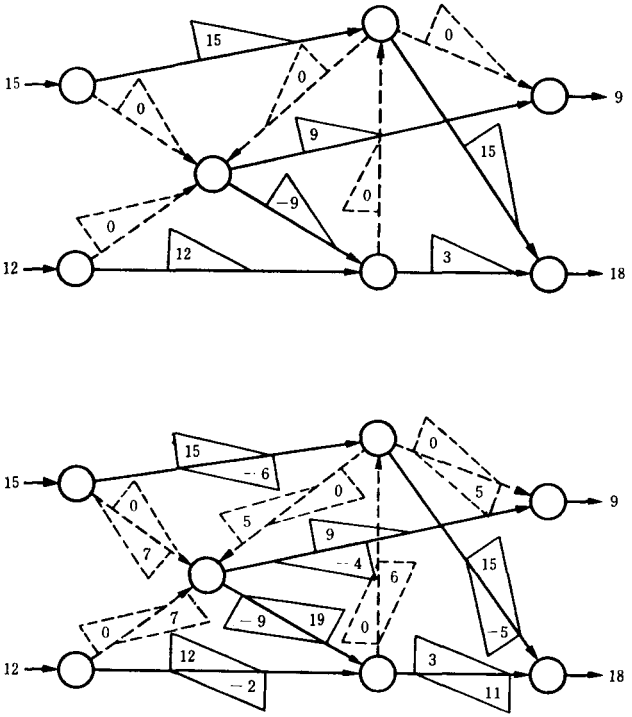


Fig. 3. Determination of Flows on Initial Tree

* An end node of a tree is a node incident to only one arc of the tree

the arcs on the tree are all determined (see Fig. 3 (i)).

Next determine the slack variables x_{ji} by (1.3′), neglecting (1.4′) (see Fig. 3 (ii)).

Hereafter let x_{ij} denote both flow and slack variables. For example $x_{41}(=-6)$ in Fig. 4 (ii) shows the slack variable in the arc (1,4).

III. Iterative Process

We have determined all x_{ij} in process II. On the initial cotree x_{ij} are all $=0$. If x_{ij} on the initial tree are all $\geqq 0$, then these constitute an optimum solution. If not, we go into the following iteration. In iterative process let x_{ij} denote the given flow and slack variables, and let x'_{ij} denote the newly determined flow and slack variables. We use ζ_{ij} and ζ'_{ij} by the same rule.

Select a $x_{kl}<0$ on the tree (where x_{kl} may be a flow or a slack), the node pair (k,l) devides all nodes on the network into two groups, K and L, where K is the set of all nodes connected with node k on the tree arcs, and L is the set of all nodes connected with node l on the tree arcs. (In Fig. 4 (iv), $(k,l)=(6,3)$, $K=(6,4,7,5)$, $L=(3,1,2)$).

Raise up potentials of all nodes in K, until for the first time the newly determined imputed cost for an arc of the cotree becomes zero. That is, let

$$\begin{aligned}(2.3)\qquad \theta = Min\{&(\zeta_{ij}:\zeta_{ij}\geqq 0, \quad (ij)\in A_{LK}),\\ &(|\zeta_{in}|:\zeta_{ij}\leqq 0, \quad (ij)\in A_{KL})\},\end{aligned}$$

where

$$(2.4)\qquad \begin{cases} A_{LK}=\text{the set of arcs } (ij)\neq(lk),\ i\in L,\ j\in K,\ (ij)\in A,\\ A_{KL}=\text{the set of arcs } (ij)\neq(kl),\ j\in L,\ i\in k,\ (ij)\in A.\end{cases}$$

When the arc which meets the conditions of the right hand side of (2.3) does not exist, the given primal problem is infeasible (the dual optimum solution is infinite). (In Fig. 4 (iv), $A_{LK}=\{(14),(25),(35)\}$, $A_{KK}=\{(43)\}$, $\theta=1$).

Let (m,n) denote an arc which attains *Min.* in (2.3), then (m,n) which is an arc on the cotree determines a loop $(m\,n\,i_1\,i_2\cdots kl\cdots i_r m)$, where (n,i_1) $(i_1,i_2)\cdots(k,l)\cdots(i_r,m)$ are on the tree. (In Fig. 4 (iv), $(m,n)=(35)$, $(m\,n\,i_1\,i_2\cdots kl\cdots i_r m)=(3\ 5\ 7\ 4\ 6\ 3)$.

Let

$$
\left\{\begin{array}{ll}
x'_{mn}=x_{mn}+x\,, & x'_{nm}=x_{nm}-x \\
x'_{mi_1}=x_{ni_1}+x\,, & x'_{i_1n}=x_{i_1n}-x \\
\cdots & \\
x'_{kl}=x_{kl}+x\,, & x'_{lk}=x_{lk}-x \\
\cdots & \\
x'_{i_rm}=x_{i_rm}+x\,, & x'_{mi_r}=x_{mi_r}-x\,,
\end{array}\right. \tag{2.5}
$$

where

$$
x=|x_{kl}|\,, \tag{2.6}
$$

then these and other $x'_{ij}=x_{ij}$ are new improved flows and slacks.

Raise up potentials of all nodes in K by θ, that is

$$
u'_i=u_i+\theta \qquad \text{for} \quad i\in K \tag{2.7}
$$

and compute ζ'_{ij} from (1.10).

Taking (m,n) into and deleting (k,l) from the tree, we get the new improved tree (In Fig. 4, (ii) is the newly improved tree from (i)).

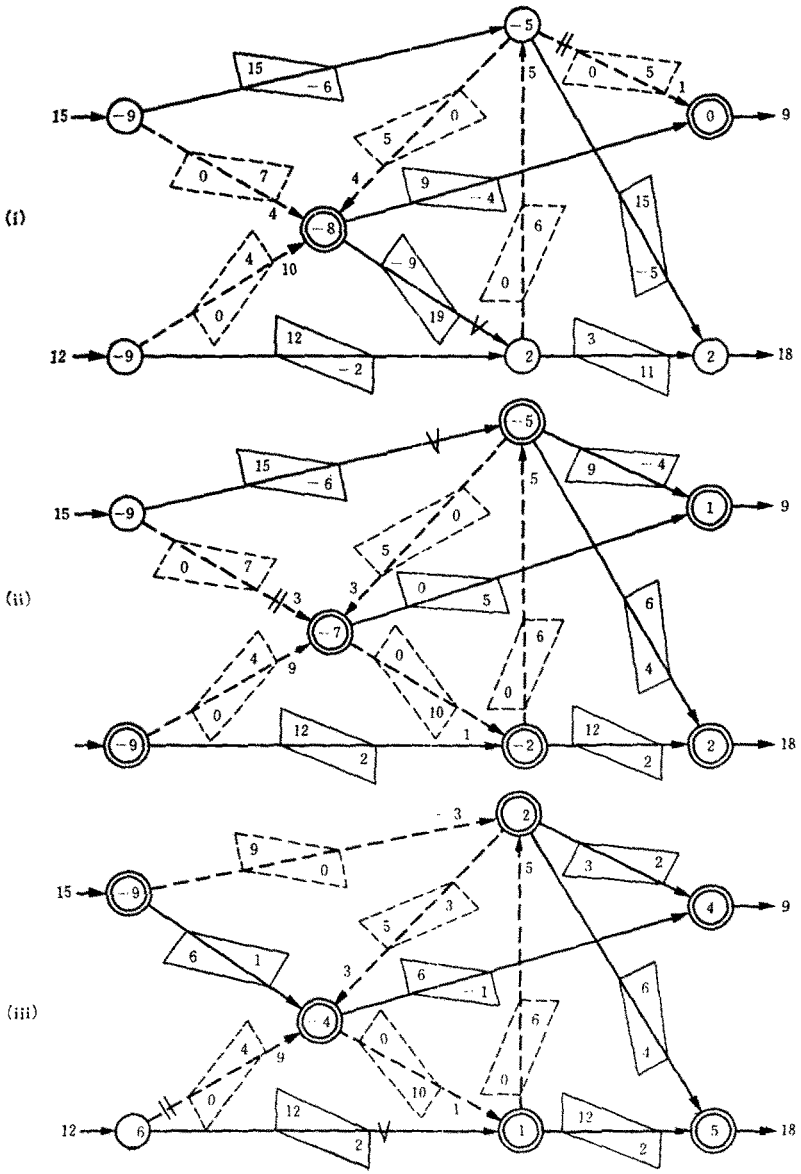
Continue above iteration process until all x_{ij} (flows and slacks) become non-negative (Fig. 4 (v)).

Optimum solutions are x_{ij} on $(ij)\in A$ (Fig. 5).

Treatment of the Degenerate Case [5]

When more than one arcs minimize (2.1) (2.2) or (2.3), which arc of these should be taken into the tree?

Number the arcs in an arbitrary order, for example like Fig. 2 (a). Consider arbitrary small $\varepsilon>0$,



(iV)

(V)

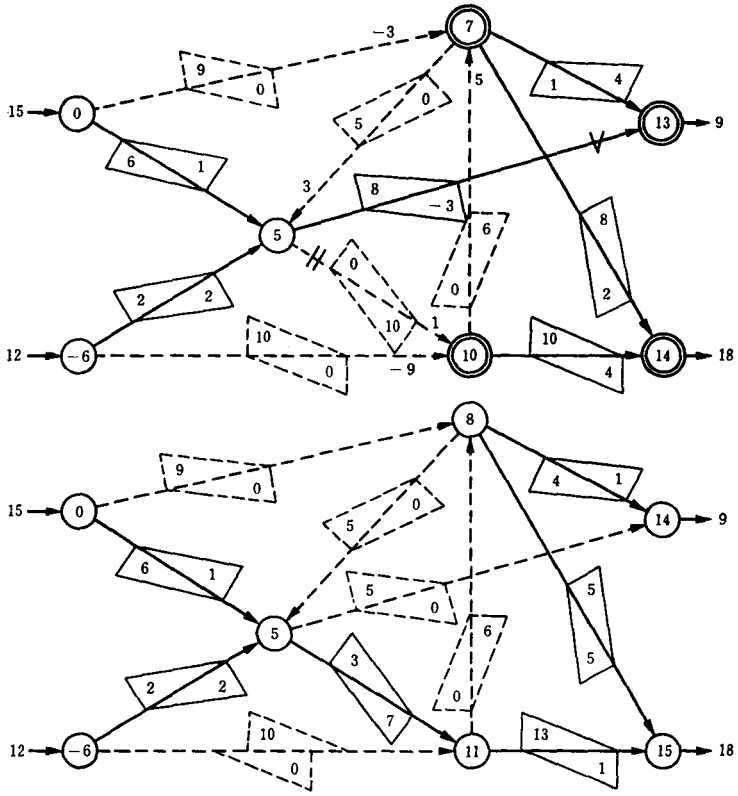


Fig. 4. Iterative Process

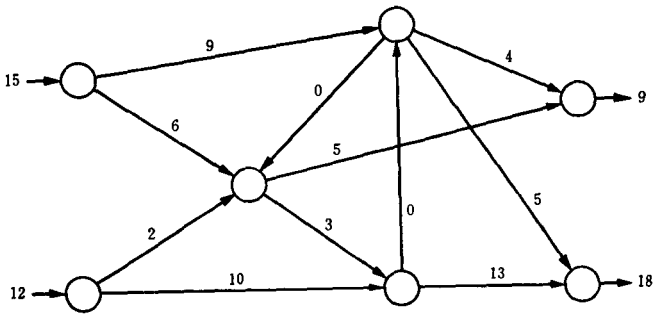


Fig. 5. Optimum Solution

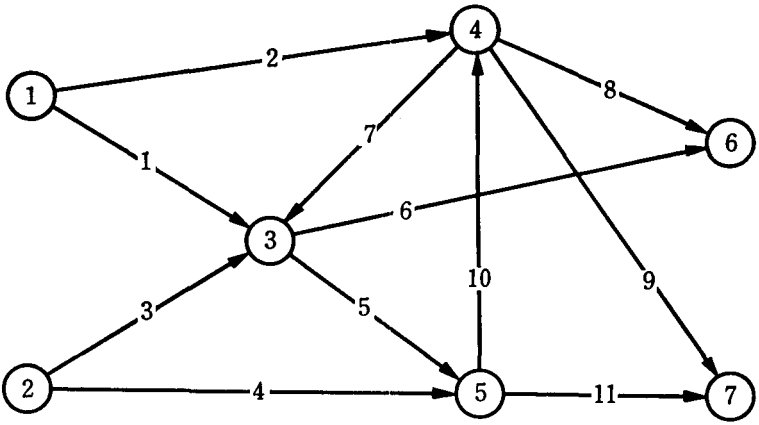


Fig. 2(a). Numbering of Arcs

and let

$$c_{ij}+\varepsilon^{\nu} \tag{2.8}$$

be the imaginary cost coefficient of arc $(ij)=\nu$. Then in every iterative step imaginary imputed costs on any two arcs do not take the same value.

Actual procedures are as follows:

(i) Case (2.1) or (2.2)

If for example $\zeta_{ij}=\zeta_{kl}$ minimize (2.1) or (2.2), the numbers of arc (ij) and (kl) are μ and ν respectively and $\mu>\nu$, then take arc μ into the tree.

(ii) Case (2.3)

Prefer $(r,s)\in A_{LK}$ to $(p,q)\in A_{KL}$, higher arc-number within A_{LK}, and lower arc-number within A_{KL}. For example; if $\zeta_{rs}=|\zeta_{pq}|$ minimize (2.3) and $(r,s)\in A_{LK}$, $(p,q)\in A_{KL}$, then take (r,s); if $\zeta_{rs}=\zeta_{pq}$ minimize (2.3), (r,s), $(p,q)\in A_{LK}$, and the arc number of $(r,s)>$ the arc number of (p,q) then take (r,s); if $\zeta rs=\zeta_{pq}$ minimize (2.3), (r,s), $(p,q)\in A_{KL}$, and the arc number of $(r,s)>$ the arc number of (p,q), then take (p,q).

§ 3. VALIDITY OF THE TREE ALGORITHM

Here we show that tree algorithm terminates in a finite number of iterations and that we get the optimum solution in its final stage.

We consider u_i and v_{ij} satisfying (1.9) as dual variables, and set conditions

$$(3.1)\qquad \begin{cases} x_{ij}=0 & \text{if } \zeta_{ij}=c_{ij}-(u_j-u_i)>0, \\ x_{ji}=0 & \text{if } \zeta_{ij}<0, \end{cases}$$

then we call flows and slacks x_{ij} which satisfy (3.1) *contact to* u_i and v_{ij}.

Theorem 1.

Primal feasible x_{ij} (flows and slacks), which are contact to dual feasible u_i, v_{ij}, are optimum.

Proof.

Construct the following formula from (1.1), (1.2) and (1.3′).

$$\begin{aligned}(3.2)\qquad &\sum_{(ij)\in A} c_{ij}x_{ij}-\sum_{i\in S} u_i(a_i-\textstyle\sum_\nu x_{i\nu}) \\ &-\sum_{i\in I} u_i(\textstyle\sum_\mu x_{\mu i}-\sum_\nu x_{i\nu}) \\ &-\sum_{i\in T} u_i(\textstyle\sum_\mu x_{\mu i}-b_i) \\ &+\sum_{(ij)\in A} v_{ij}(M_{ij}-x_{ij}-x_{ji}) \\ &=\sum_{(ij)\in A}(c_{ij}-(u_j-u_i)-v_{ij})x_{ij}+\sum_{(ij)\in A}(-v_{ij})x_{ji} \\ &+\sum_{i\in T} b_iu_i-\sum_{i\in S} a_iu_i+\sum_{(ij)\in A} M_{ij}v_{ij}.\end{aligned}$$

From (1.9) and (1.10), (3.1) is equivalent to

$$(3.3)\qquad \begin{cases} x_{ij}=0 & \text{if } c_{ij}-(u_j-u_i)-v_{ij}>0, \\ x_{ji}=0 & \qquad\qquad -v_{ij}>0. \end{cases}$$

So for x_{ij} contact to u_i, v_{ij} the right hand side of (3.2)

$$=\sum_{i\in T} b_iu_i-\sum_{i\in S} a_iu_i+\sum_{(ij)\in A} M_{ij}v_{ij}$$

which is the objective function of the dual problem. Further for feasible

x_{ij} the left hand side of (3.2)

$$= \sum_{(ij)\in A} c_{ij}x_{ij}$$

which is the objective function of the primal problem.

Therefore from the basic dual theorem the primal feasible x_{ij} contact to u_i, v_{ij} are optimum. q.e.d.

Theorem 2.

(i) In every iteration of the tree algorithm (§2, III) flows and slacks x_{ij} satisfy the conservation constraints (1.2) and the modified capacity constraints (1.3').

(ii) In every iteration of the tree algorithm (§2, III) flows and slacks x_{ij} are contact to u_i and v_{ij} satisfying (1.9). Hence from (i)

$$(3.3) \qquad \begin{aligned} x_{ij}&=0 \quad \text{if} \quad \zeta_{ij}>0, \\ x_{ij}&=M_{ij} \quad \text{if} \quad \zeta_{ij}<0. \end{aligned}$$

Proof.

(i) is clear from procedures in §2 II and formula (2.5).

(ii) In the initial step all ζ_{ij} are $\geqq 0$, if $\zeta_{ij}>0$ then arc (ij) is on the cotree, so $x_{ij}=0$ from §2 II. Therefore the initial flows and slacks are contact to dual variables.

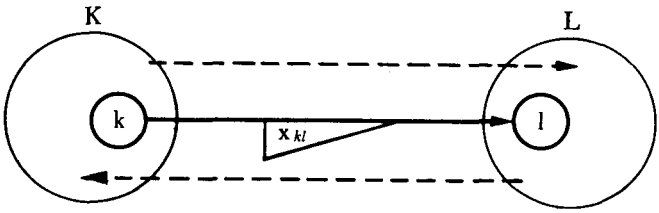
In the iterative process, the potential-up procedures (2.7) and the flow-computation (2.5) assert that the newly determined flows and slacks x_{ij} are contact to the newly determined dual variables u_i and v_{ij}. q.e.d.

Theorem 3.

Every iteration of the algorithm (§2, III) increases the value of the objective function of the dual problem by θx, where θ is to be determined by (2.3) and x by (2.6).

Proof.

Case (i): x_{kl} is a flow (In Fig. 4 (i), $(kl)=(35)$).



From Theorem 2 (i), the total flows into K is equal to the total flow out of K. That is

$$\sum_{(ij)\in ALK,\ \zeta_{ij}<0} x_{ij}+\sum_{i\in S\cap K} a_i=x_{kl}+\sum_{(ij)\in AKL,\ \zeta_{ij}<0} x_{ij}+\sum_{j\in T\cap K} b_j$$

which we can rewrite by (3.3) as

$$(3.4)\qquad \sum_{(ij)\in ALK,\ \zeta_{ij}<0} M_{ij}+\sum_{i\in S\cap K} a_i=x_{kl}+\sum_{(ij)\in AKL,\ \zeta_{ij}<0} M_{ij}+\sum_{j\in T\cup K} b_j .$$

So we have

$$(3.5)\qquad \theta\{\sum_{j\in T\cap K} b_j-\sum_{i\in S\cup L} a_i+\sum_{(ij)\in ALK,\ \zeta_{ij}<0} M_{ij}-\sum_{(ij)\in ALK,\ \zeta_{ij}<0} M_{ij}\}=-\theta x_{kl}.$$

The left hand side of (3.5) is equal the increase of the objective function of the dual problem due to (2.7). q.e.d.

Case (ii): x_{kl} is a slack (in Fig. 4 (ii), $(kl)=(14)$)

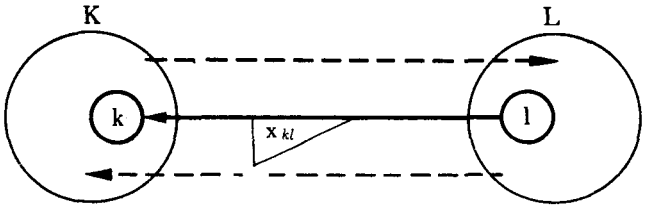


Fig. 5. b

By the same reasoning as (3.4), we have

$$(3.6)\qquad \sum_{(ij)\in ALK,\ \zeta_{ij}<0} M_{ij}+\sum_{i\in S\cap K} a_i+M_{kl}-x_{kl}=\sum_{(ij)\in AKL,\ \zeta_{ij}<0} M_{ij}+\sum_{j\in T\cup K} b_j .$$

So we have

$$(3.7)\qquad \theta\{\sum_{j\in T\cap K} b_j - \sum_{i\in S\cup K} a_i + \sum_{(ij)\in AKL,\ \zeta_{ij}<0} M_{ij} - \sum_{(ij)\in ALK,\ \zeta_{ij}<0} M_{ij} - M_{lk}\} = -\theta x_{kl}\,.$$

The left hand side of (3.7) is equal to the increase of the objective function of the dual problem due to (2.7) q.e.d.

To summarize Theorems 1 and 2 assert that, if in our tree algorithm x_{ij} (flows and slacks) are all $\geqq 0$, then we have attained the optimal solution, and Theorem 3 asserts that our tree algorithm (with treatment on degenerate case) terminates in a finite number of iterations; {The number of different trees on the given network is finite. Each tree in our algorithm determines a value of the dual objective function. The values strictly increase (in imaginary imputed cost) because of Theorem 3, so that the trees which appear in our algorithm are all different. Therefore iteration terminates in a finite number}.

§ 4. NOTES ON COMPUTERIZING THE TREE ALGORITHM

To determine the loop $(m\,n\,i_1\,i_2\cdots kl\cdots i\,m)$ in our algorithm in § 2 III would be rather difficult to program for digital computers. It has connection with determining the sets of nodes K and L. So here we shall make a remark on these procedures.

Determination of K and L

In Fig. 6 let $k=3$, $l=5$. Label with number $k=3$ the nodes connected with node $k=3$ on just one arc of the tree (except node $l=5$). Select a node, say node 4, of labelled nodes, and label with number 4 the unlabelled nodes connected with node 4 on just one arc of the tree. Continue above process until no nodes remain to be labelled. Then the set of all the labelled nodes is K.

Starting with $l=5$, the same procedures as above determines the set L.

Determination of the Loop $(m\,n\,i_1\,i_2\cdots kl\cdots i_r m)$

Let the labelled number on node n be i_1, and let labelled number on node i_1 be $i_2, \cdots$, then $i_s=k$ for some number s, and we get the path $(n\,i_1\,i_2\cdots k)$. Similaly starting with node m, we get the path $(m\,i_r\,i_{r-1}\cdots l)$. Then the loop $(m\,n\,i_1\,i_2\cdots k_1\cdots i_r m)$ is the desired one. In Fig. 6, $(m\,n\,i_1\,i_2\cdots k_1\cdots i_r m)=(10,9,4,3,5,11,10)$.

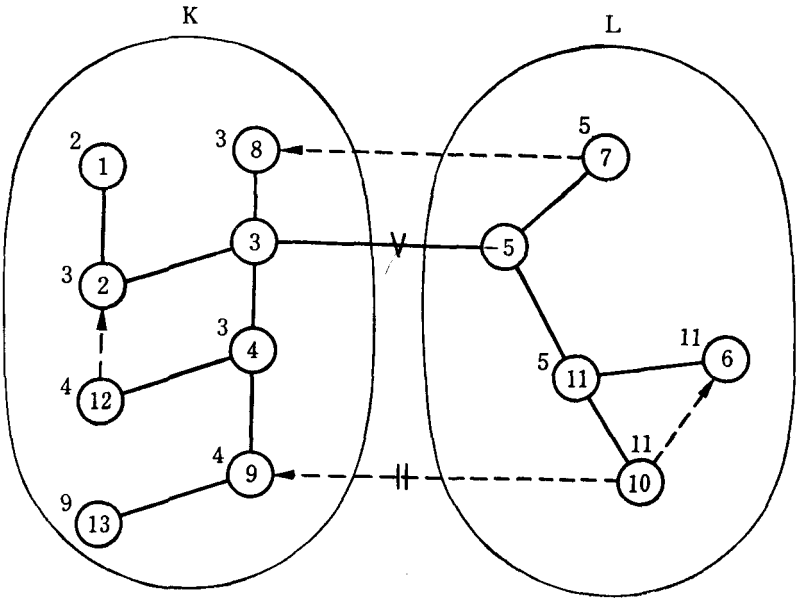


Fig. 6. Determination of the loop

§5. APPLICATION TO THE HITCHCOCK TRANSPORTATION PROBLEM

Table 1 shows the well-known Hitchcock Transportation Problem. This is a special case of the network transportation problem in §1—no intermediate nodes, no capacity constraint, no arcs among sources and no arcs among sinks. So we can apply our tree algorithm to the Hitchcock problem, and get a new simple algorithm.

We use the table-forms such as Table 2, 3, $\cdots$ to represent the algorithm, since these are more convenient than network figures. In a table-form a row number coresponds to a source, a column number to a sink, and an entry to an arc.

Table 1 c_{ij}

souce \ snik	5	6	7	8	9	supplies a_i
1	4	9	8	10	12	24
2	6	10	3	2	3	18
3	3	2	7	10	3	20
4	3	5	5	4	8	16
demands b_j	10	20	10	18	20	78

I. Determination of Initial Tree

(i) Select a sink (sink 9 in Table 2 (i)) and set its potential to be zero, and set the potentials of all sources so that the imputed costs of all entries in the selected column are zero. Set the potentials of all sinks to be zero, and compute the imputed costs of all entries (Table 2 (i)).

(ii) Set a potential of each sink to be the minimum of the imputed costs of the column coresponding to the sink. Enclose by squares the entries with zero-imputed cost (Table 2 (ii)).

When there are more than one zero in a column (except the selected column in tep (i)), select one with lower row number of these zeros (see treatment on degenerate case).

Then squared entries constitute an initial tree.

II. Determination of Flows on Initial Tree

Set the flows on the entries enclosed with the squares so that each column sum cquals to its demand and each row sum cquals to its supply. (Table 3 (i)).

Table 2

(i)

	u_i	5	6	7	8	9
		0	0	0	0	0
1	−12	− 8	− 3	− 4	− 2	0
2	− 3	3	7	0	− 1	0
3	− 3	0	− 1	4	7	0
4	− 8	− 5	− 3	− 3	− 4	0

(ii)

	u_i	5 − 8	6 − 3	7 − 4	8 −4	9 0
1	−12				2	
2	− 3	11	10	4	3	
3	− 3	8	2	8	11	
4	− 8	3	0	1		

III. Iterative Process

When no negative flows exist on entries enclosed with the squares the optimum solution is attained.

(i) Determination of the Sets K and L.

Select an entry with negative flow out of the entries enclosed with the squares and mark it with ν. Let its row number be k and column number l (In Table 3 (i), $k=1$, $l=9$).

Label sinks, connected with the source k on the tree arcs (the entries enclosed with the squares), with ◎$_k$. Next label unlabelled sources, connected with each labelled sinks on the tree arcs, with ◎ and its sink number (Table 3 (i)). Continue above process until no nodes to

be labelled exist. The set of labelled exist. The set of labelled nodes with ◎ is K.

Starting with sink l, repeat the same process as above, using ○ instead of ◎. The set of labelled nodes with ○ is L.

(ii) Determination of the Loop $(n\, i_1\, i_2 \cdots kl \cdots i_r mn)$

Let

$$\theta = Min\{\zeta_{ij} : i \in S \cap L,\quad j \in T \cap K\} \tag{5.1}$$

(S is the set of sources and T the set of sinks).

And let (m, n) denote the arc minimizing (5.1) (In Table 3 (iii), $m=2$, $n=8$), and $L(i)$ denote the labelled number of node i. Let us set

$$\begin{cases} L(n)=i_1, & L(i_1)=i_2, \cdots, L(i_s)=k \\ L(m)=i_r, & L(i_r)=i_{r-1}, \cdots, L(i_t)=l \end{cases} \tag{5.2}$$

Then $(n\, i_1\, i_2 \cdots kl \cdots i_r mn)$ is the desired loop. (In Table 3 (iii), $(n\, i_1\, i_2 \cdots kl \cdots i_r mn)=(8, 4, 6, 3, 9, 2)$)

(iii) Potential Up and Modifing Flow Values

Up the potentials of nodes $\in K$, and calculate imputed costs. Enclose the entry (mn) with a square, and delete the square of (kl). Further calculate new flow values x'_{ij} from old flow values x_{ij} by

$$\begin{cases} x'_{mn} = x_{mn} + |\, x_{kl}\,|, \\ x'_{i_1 n} = x_{i_1 n} - |\, x_{kl}\,|, \\ \vdots \\ x'_{kl} = x_{kl} + |\, x_{kl}\,|, \\ \vdots \\ x'_{mi_r} = x_{mi_r} - |\, x_{kl}\,|. \end{cases} \tag{5.3}$$

In Table 3 (iii),

$$\begin{aligned} x'_{28} &= x_{28} + |\, x_{36}\,| = 0+2=2 \\ x'_{48} &= x_{48} - |\, x_{36}\,| = 18-2=16 \\ x'_{46} &= x_{46} + |\, x_{36}\,| = -2+2=0 \\ x'_{36} &= x_{36} - |\, x_{36}\,| = 18-2=16 \end{aligned}$$

$$x'_{39}=x_{39}+|x_{36}|=2+2=4$$
$$x'_{29}=x_{29}-|x_{36}|=18-2=16.$$

Other $x'_{ij}=x_{ij}$.

Table 3

i)

		(5)$_1$	(6)$_1$	(7)$_1$	(8)$_4$	(9)
		−8	−3	−4	−4	0
(1)	−12	10	20 //	10	2	−16 V
(2)$_9$	−3	11	10	4	3	18
(3)$_9$	−3	8	2	8	11	20
(4)$_9$	−8	3	0 //	1	18	−2 //

		(5)$_1$	(6)$_4$	(7)$_1$	(8)$_4$	(9)
		−8	−3	−4	−4	0
(1)$_6$	−12	10	4	10	2	0
(2)$_9$	−3	11	10	4	3	18
(3)$_9$	−3	8	2 //	8	11	20 //
(4)	−8	3	16 //	1	18	−18 V

		(5)$_1$	(6)	(7)$_1$	(8)$_4$	(9)$_3$
		−6	−1	−2	−2	0
(1)$_6$	−10	10	4	10	2	2
(2)$_9$	−3	9	8	2	1 //	18 //
(3)$_6$	−3	6	18 //	6	9	2 //
(4)	−6	3	−2 V	1	18 //	2

(iV)

		5	6	7	8	9
		−6	−1	−2	−1	0
1	−10	10	4	10	1	2
2	− 3	9	8	2	2	16
3	− 3	6	16	6	8	4
4	− 5	4	1	2	16	3

Treatment of the Degenerate Case

In the Hitchock transportation problem, treatment of the degenerate case is very simple because of non-negative imputed costs. We have only to obey the rule that the entry with lowest row or column number out of entries minimizing (5. 1) should be taken into the new tree. For example if $\zeta_{23}=\zeta_{25}$ minimize (5. 1), then let $(mn)=(23)$; if $\zeta_{23}=\zeta_{43}$ minimize (5. 1), then let $(mn)=(23)$; if $\zeta_{23}=\zeta_{57}$ minize (5. 1), then let $(mn)=(23)$.

§ 6. TWO DIRECTIONAL CASE AND NO DIRECTIONAL CASE

So far we have considered one directional case, where every node pair had at most on arc. Here we note about two directional case and no-directional case.

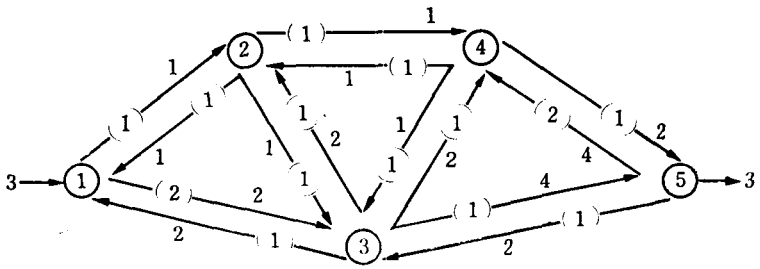


Fig. 7. Two-Directional Network Transportation

(i) Two-directional Case

Fig. 7 shows a two-directional case, to which we can apply our tree algorithm directly. Fig. 8 shows determination of the initial tree, and Fig. 9 the iterative process.

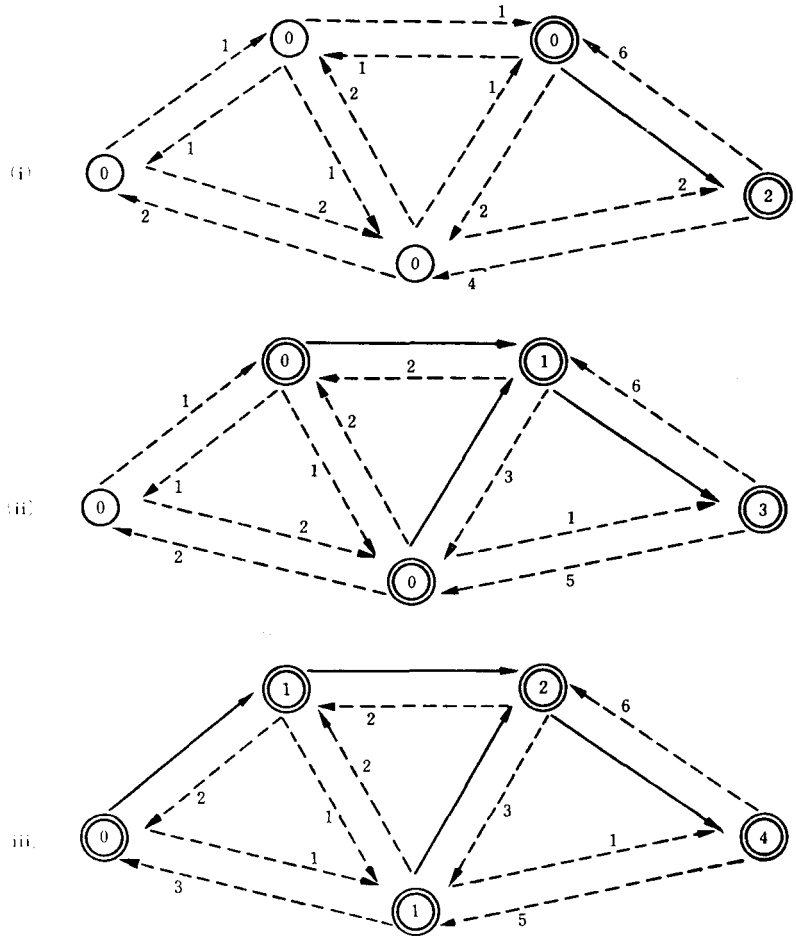


Fig 8. Determination of Initial Tree

(ii) No-directional Case

For a node pair (ij) to have a no-directional arc with capacity M means that; we can transport either from node i to node j or from i

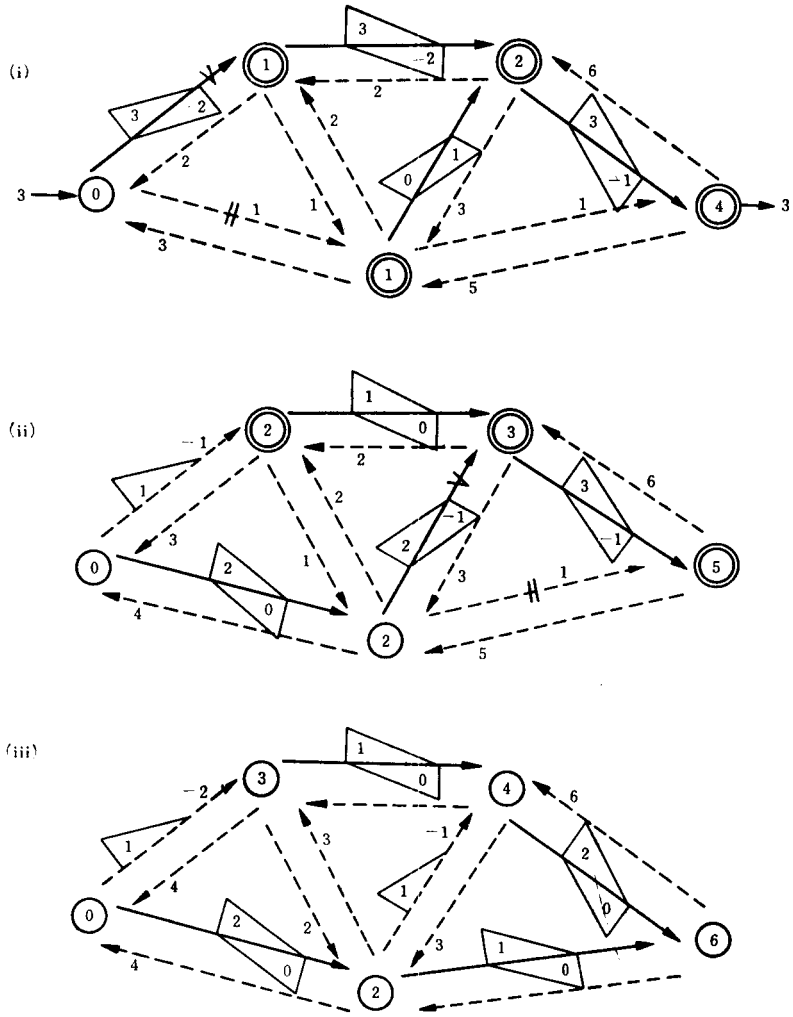


Fig. 9. Iteratine Process

to j, but $x_{ij} \cdot x_{ji}=0$, $0 \leqq x_{ij} \leqq M$, $0 \leqq x_{ji} \leqq M$.

Let denote $\textcircled{i}\overset{C}{\text{———}}(M)\text{———}\textcircled{j}$ no-directional arc with capacity M and cost coefficient C, then $\textcircled{i}\overset{C}{\text{———}}(M)\text{———}\textcircled{j}$ is equivalent to two-directional arc $\textcircled{i}\overset{C}{\text{———}}(M)\text{———}\rightarrow$ $\textcircled{j}$ in our algorithm. $\leftarrow\text{———}(M)\underset{C}{\text{———}}$

REFERENCES

[1] L.R. Ford and D.R. Fulkerson -Flows in Networks, Princeton University Press, Princeton (1962).

[2] M. Iri—A New Method of Solving Transportation Network Problems, Journal of the Operations Research Society of Japan, Vol. 3 (1960) pp. 27-87.

[3] G.B. Dantzig, L.R. Ford and D.R. Fulkerson—A Primal Dual Algorithm for Linear Programs, in "Linear Inequalities and Related Systems" H.W. Kuhn and W. Tucker ed. Princeton Univ. Press (1956).

[4] C.E. Lemke and A. Charnes—Extremal Problems in Linear Inequalities, Carnegie Inst. of Tech. Dept. of Math. Tech. Rep., No. 36 (1953) 78.

[5] A. Charnes, W. Cooper and A. Henderson—An Introduction to Linear Programming (Wiley, 1953) 74.

[6] Toshio Fujisawa—A Computational Method for the Transportation Problem on a Network, Journal of the Operations Research Society of Japan, Vol. 1 No. 4 (1958) pp. 157-174.