On Some Techniques Useful for Solution of Transportation Network Problems

N. Tomizawa
University of Tokyo
Tokyo, Japan

ABSTRACT

This paper presents an efficient algorithm for solving transportation problems. The improvement over the existing algorithms of the "primal-dual" type [3], [5] *consists in modifying the "potential-raising" stages of the solution process in such a way that negative-cost arcs are removed so that the Dijkstra's algorithm may be applied. Especially, the algorithm requires at most* n^3 *additions and comparisons when applied to an n-by-n assignment problem, as compared with the theoretical upper bound proportional to* n^4 *for the number of such operations required by currently available methods. Furthermore, auxiliary techniques of simplifying the original network by means of "reduction" and "induction" are also introduced as useful tools to treat large-scale problems and specially-structured problems with.*

1. FORMULATION OF THE PROBLEM

Let A denote the set of directed arcs (i,j) ordered pairs of nodes, of a transportation network N with n nodes: 1(source), 2, 3, ..., n-1, n(sink). We shall consider the case where every arc $(i,j)\varepsilon A$ is assigned both a capacity $c_{ij}(\geq 0)$, which limits the amount of flow that can pass through it, and a cost $d_{ij}(\geq 0$, if $c_{ij}>0)$ of shipping one unit of flow along it. We are interested in the problem of shipping a given total amount v of flow from the source to the sink with

Networks, 1: 173-194

a least possible cost.

In precise terms, the problem is stated as follows:

To minimize

$$\sum_{i,j} d_{ij} f_{ij} \tag{1}$$

under the constraints

$$\sum_j f_{ij} - \sum_j f_{ji} = \begin{cases} v, & i = 1, \\ 0, & i \neq 1, n, \\ -v, & i = n, \end{cases} \tag{2}$$

$$f_{ij} \cdot f_{ji} = 0, \tag{3}$$

$$0 \leq f_{ij} \leq c_{ij}, \tag{4}$$

where c_{ij}'s(capacities) and d_{ij}'s(costs) are given nonnegative constants and f_{ij}'s(flows) are nonnegative variables whose totality is called a flow configuration in the network. If all $c_{ij} = 1$ and $v = 1$, that is, if there is no essential arc-capacity constraint, then the problem is reduced to one of finding a *shortest route* from the source to the sink with d_{ij} as distances.

2. DEFINITIONS AND NOTATIONS

The *modified network* with respect to a flow configuration in a network N is defined as the network N* of the same topological structure as the original network N but with *modified capacities* c^*_{ij} defined by

$$c^*_{ij} = \begin{cases} c_{ij} - f_{ij}, & \text{if } f_{ji} = 0, \\ f_{ji}, & \text{if } f_{ji} > 0, \end{cases} \tag{5}$$

as well as with the *modified costs* d^*_{ij} defined by

$$d^*_{ij} = \begin{cases} d_{ij}, & \text{if } c_{ij} > f_{ij} \text{ and } f_{ji} = 0, \\ \infty, & \text{if } c^*_{ij} = 0, \text{ i.e. } c_{ij} = f_{ij}, \\ -d_{ji}, & \text{if } f_{ji} > 0. \end{cases} \tag{6}$$

We shall make use also of the *relative costs* $\tilde{d}_{ij}$ defined by

$$\tilde{d}_{ij} = d_{ij} + q_i - q_j \ , \qquad (7)$$

where q_i's are appropriately defined *potentials* at nodes. The *modified relative costs* $\tilde{d}^*_{ij}$ are defined from $\tilde{d}_{ij}$'s in the same way as d^*_{ij}'s are defined from d_{ij}'s. It is easy to verify that

$$\tilde{d}^*_{ij} = d^*_{ij} + q_i - q_j \qquad (8)$$

3. SOLUTION ALGORITHM

The fundamental idea of our algorithm is to apply the shortest-route algorithm of E. W. Dijkstra [1] (which is one of the computationally most efficient algorithms now available in the case of nonnegative arc lengths) by regarding the modified relative costs of the modified network as arc lengths. (The Dijkstra's algorithm solves the shortest-route problem on an n-node network with nonnegative arc lengths through additions and comparisons of a number at most proportional to n^2, while the theoretical upper bound of the number of such operations required by any known method applicable to networks containing arcs of negative length is at least proportional to n^3 [2].) In fact, we shall see later that, unlike modified costs, modified relative costs remain nonnegative during the course of the solution process.

To find the shortest routes from the source to all the other nodes i, we may resort to the method of labelling nodes. Each node i will have a label of the form $[p_i, r_i]$. The first element p_i of the label indicates the potential with regard to modified relative costs whereas the second element r_i indicates the node number of the node preceeding node i on the shortest route from the source to node i.

Our algorithm may briefly be stated as follows:[1)]

1) *The essentially nonexistent arcs, i.e. the arcs with* $d^*_{ij} = \infty$ *may be ignored in the course of calculation (specifically, in (2) of STEP 2), but, for the sake of simplicity of presentation, we state the following algorithm as if all the arcs were essentially existent.*

STEP 0. Start with $f_{ij} = 0$ for all arcs (i,j), putting all $q_i = 0$.

STEP 1. Regarding the modified relative costs $\tilde{d}^*_{ij}$ (with respect to q_i's) as arc lengths, find the shortest routes r_i and the potentials p_i (i.e. the shortest distances) from the source to all the other nodes i according to the Dijkstra algorithm slightly modified as follows:[2]

(0) To begin with, put $K = \{1\}$, $p_1 = 0$, $r_j = 1$ and $p_j = d^*_{1j} - q_j$ $(= d^*_{1j})$ for all $j \notin K$.

(1) Let $p_k = \min \{p_j \; ; \; j \notin K\}$. If $p_k = \infty$, then we assert the nonexistence of a route of finite length from the source to the sink. Otherwise, put $K = K \cup \{k\}$ and $q_k = q_k + p_k$.

(2) For every $j \notin K$, if $p_j > d^*_{kj} + q_k - q_j$ $(= \tilde{d}^*_{kj} + p_k)$, then put $p_j = d^*_{kj} + q_k - q_j$ $(= \tilde{d}^*_{kj} + p_k)$ and $r_j = k$.

(3) Repeat (1) and (2) alternately until all the nodes 1 through n are contained in K.[3]

STEP 2. Assign as much incremental flow as possible along the shortest route chosen in STEP 1 from the source to the sink, i.e. make the corresponding change in the flows f_{ij} in the arcs (i,j) on the route as far as the new total flow value does not exceed v. If the new total flow value attains v, then stop. Otherwise, return to STEP 1.

4. VALIDITY OF THE ALGORITHM

From the general standpoint expounded in §23 of [5], the algorithm in the preceeding section may be regarded as a variant of ordinary primal-dual two terminal network algorithm in which the "associated minimum-route problem" are solved by the Dijkstra algorithm, so that the validity of our algorithm for a restricted class of problems defined in section 1 without resorting to the general discussion.

If, at the beginning of STEP 1 in every iteration in the course of solving a problem according to our algorithm, the

---

2) We compute $\tilde{d}^*_{ij}$'s from d_{ij}'s, q_i's and p_i's as they are needed, instead of computing them in advance.

3) It is more efficient to stop as soon as the sink n is contained in K and then to put $q_j = q_j + p_n$ for all $j \notin K$.

modified relative costs (which will be denoted by $\tilde{d}^*_{ij}$ in the remainder of this section) are nonnegative, then it is obvious from the ordinary optimality argument in linear programming that the algorithm works well until either the optimal flow configuration is obtained or it is seen that no feasible flow configuration exists. Therefore, it is sufficient to prove that $\tilde{d}^*_{ij} \geq 0$ in every iteration, Since, at the outset, we have $f_{ij} = 0$ so that $\tilde{d}^*_{ij} = d_{ij} \geq 0$ for all i, j, the proof can be made by induction on iteration number. Therefore, we assume that we have nonnegative $\tilde{d}^*_{ij}$'s with respect to a flow configuration and a set of potentials, and enter STEP 1. Then we show that at the end of STEP 1, we have nonnegative modified relative costs (which will be denoted by $\tilde{d}^{*'}_{ij}$ in the following) with respect to the new potentials q_i obtained in STEP 1 and that, at the end of STEP 2, we have also nonnegative modified relative costs (which will be denoted by $\tilde{d}^{*''}_{ij}$ in the following) with respect to the new potentials q_i obtained in the preceding STEP 1 and the new flow configuration obtained in STEP 2.

First, we observe that we have

$$\tilde{d}^*_{ij} + p_i - p_j \begin{cases} = 0, & \text{if } (i,j) \text{ is contained in the shortest route,} \\ \geq 0, & \text{otherwise,} \end{cases} \tag{9}$$

according to the Dijkstra algorithm in STEP 1. Since $p_i \leq p_n \leq p_j$ for all $i \in K$ and $j \notin K$, where K is the set of nodes contained in the spanning tree obtained at the end of STEP 1, we have from the assumption and (7), (8) and (9),

$$\tilde{d}^{*'}_{ij} = \begin{cases} \tilde{d}^*_{ij} + p_i - p_j \geq 0, & \text{if } i,j \in K, \\ \tilde{d}^*_{ij} + p_i - p_n \geq \tilde{d}^*_{ij} + p_i - p_j \geq 0, & \text{if } i \in K,\ j \notin K, \\ \tilde{d}^*_{ij} + p_n - p_j \geq \tilde{d}^*_{ij} + p_j - p_j \geq 0, & \text{if } i \notin K,\ j \in K, \\ \tilde{d}^*_{ij} + p_n - p_n = \tilde{d}^*_{ij} \geq 0, & \text{if } i,j \notin K. \end{cases} \tag{10}$$

Thus it has been seen that all the $\tilde{d}^{*'}_{ij}$'s are nonnegative.

Next, from (6), (8) and (10), and from the fact that the incremental flow is assigned along the arcs (direction taken into account) with $\tilde{d}^*_{ij} = 0$, we have

$$\tilde{d}^{*''}_{ij} = \begin{cases} -\tilde{d}^{*'}_{ji} \ (=0) \text{ if } f_{ji} = 0 \text{ and } f'_{ji} > 0 \\ \qquad\qquad \text{or } c_{ji} = f_{ji} \text{ and } c_{ji} > f'_{ji}, \\ \infty (\geq 0) \text{ if } c_{ij} > f_{ij} \text{ and } c_{ij} = f'_{ij}, \\ d_{ij} + d_{ji} \ (\geq 0) \text{ if } f_{ji} > 0 \text{ and } f'_{ji} = 0, \\ \tilde{d}^{*'}_{ij} \ (\geq 0) \text{ otherwise,} \end{cases} \tag{11}$$

where f'_{ij}'s are the new flows and f_{ij}'s are the old ones. Thus, all the $\tilde{d}^{*''}_{ij}$'s have been shown to be nonnegative.

5. ILLUSTRATIVE EXAMPLE

As a numerical example illustrating our algorithm, let us consider the network of Figure 1-1, where the pairs of numbers (d_{ij}, c_{ij}) beside the arrows of the arcs (i,j) indicate the costs d_{ij} and the capacities c_{ij}.

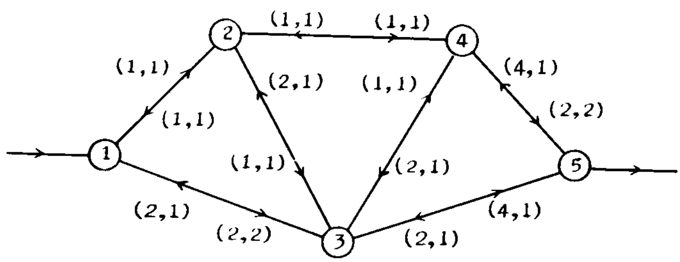

Fig. 1-1 Given Network

We shall obtain a minimum-cost flow configuration with the maximum total flow value v on this network. We put $q_i = 0$ and set out the Dijkstra algorithm with $K = \{1\}$. Since node 2 and node 3 are the only neighboring nodes of the source 1 and $d^*_{12} = d_{12} = 1$, $d^*_{13} = d_{13} = 2$, we give the temporary label [1,1] to node 2 and [2,1] to node 3 and [∞,1] to all the other nodes. We find $p_2 = \min\{1,2\} = 1$, and put $q_2 = q_2 + p_2 = 1$, $r_2 = 1$, and

$K = \{1, 2\}$, so that the label [1,1] becomes the permanent label of node 2, and that arc (1,2) is known to belong to the shortest route from the source.

Since the neighboring nodes of node 2 are nodes 3 and 4, and

$$p_3 \ (= 2) \leq d^*_{23} + q_2 - q_3 \ (= 1+1-0 = 2),$$

$$p_4 \ (= \infty) > d^*_{24} + q_2 - q_4 \ (= 1+1-0 = 2),$$

we put $[p_4, r_4] = [2,2]$ and leave $[p_3, r_3]$ untouched.
Since p_3 is the smallest among the p_i's for nodes not in K and $r_3=1$, $q_3 = q_3 + p_3 = 2$, make the label [2,1] for node 3 permanent and enroll arc (1,3) as an arc in the shortest route.
Since the neighboring nodes of node 3 are nodes 4 and 5, and

$$p_4 \ (= 2) < d^*_{34} + q_3 - q_4 \ (= 1+2-0),$$

$$p_5 \ (= \infty) > d^*_{35} + q_3 - q_5 \ (= 4+2-0),$$

we put $[p_5, r_5] = [6,3]$ leaving $[p_4, r_4]$ untouched. Continuing similarly, we have in succession:

$K = \{1,2,3,4\}$
- $[p_4, r_4] = [2,2]$ made permanent
- (2,4) enrolled in the shortest-route arcs
- $q_4 = q_4 + p_4 = 2$
- $p_5 := \min(p_5, d^*_{45} + q_4 - q_5) = \min(6, 2+2-0) = 4$
- $r_5 = 4$

$K = \{1,2,3,4,5\}$
- $[p_5, r_5] = [4,4]$ made permanent
- (4,5) enrolled in the shortest-route arcs
- $q_5 = q_5 + p_5 = 4.$

Thus the computation of STEP 1 in the first iteration is completed. The shortest route thus obtained is drawn by bold lines, and the permanent labels $[p_i, r_i]$ are indicated by the side of respective nodes in Figure 1-2, where the numbers over the arrows and the numbers beside the nodes indicate the modified costs and the potentials q_i at the beginning of STEP 1.

The maximum amount Δv of incremental flow along the shortest route equals $\min(c^*_{12}, c^*_{24}, c^*_{45}) = \min(1,1,2) = 1$ and the total cost equals $q_5 \times v = 4 \times 1 = 4$. The new f_{ij}'s are indicated under the respective arcs in Figure 1-2.

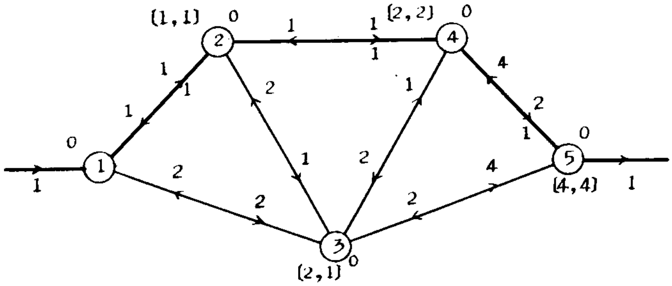

Fig. 1-2 Modified network ($\Delta q \times \Delta v = 4$)

Continuing this process, we have similarly Figure 1-3, Figure 1-4, and Figure 1-5.

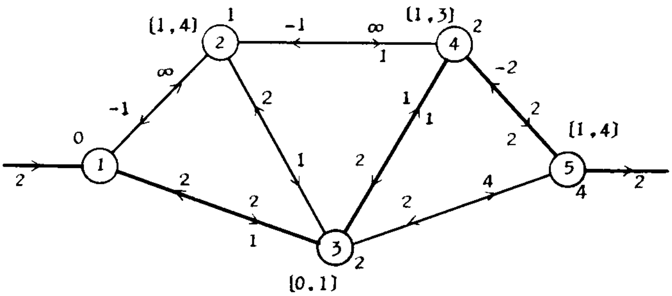

Fig. 1-3 Modified network ($\Delta q \times \Delta v = 5$)

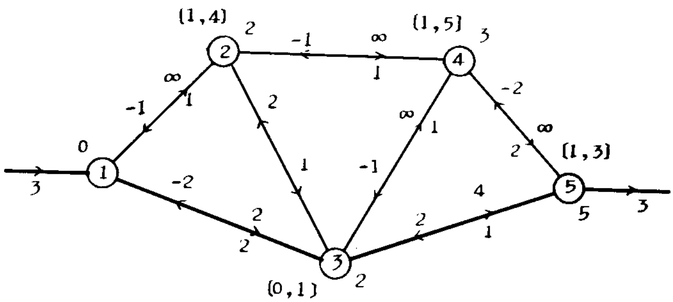

Fig. 1-4 Modified network ($\Delta q \times \Delta v = 6$)

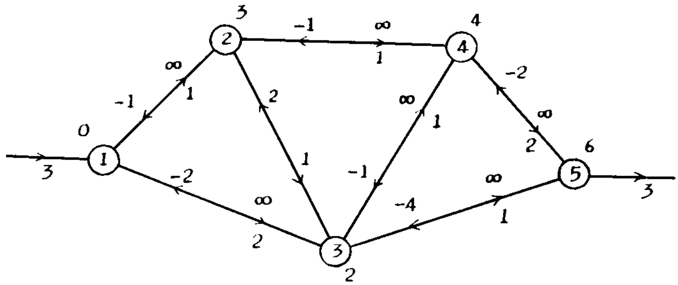

Fig. 1-5 Optimal solution

It should be noted that we always have $\tilde{d}^*_{ij} \geq 0$ during the course of solution. Since we cannot increase the flow in Figure 1-5, the optimal solution is given by it and the total cost equals $4 \cdot 1 + 5 \cdot 1 + 6 \cdot 1 = 15$.

6. REDUCTION ALGORITHM

For a given network N and a given flow configuration, let us consider the modified network (which will be denoted by $\tilde{N}^*$ in the following) with the same graphical structure as N but with the modified relative costs assigned to its arcs. Let us call an arc (i,j) in $\tilde{N}^*$ with $\tilde{d}^*_{ij} = 0$, i.e. an arc which is not saturated and has cost zero, a *free arc*.

A *connected component* $M_1 (M_r)$ is defined as a subnetwork such that there is a route consisting of free arcs alone from the source to all the other nodes in it (to the sink from all the other nodes in it) and a *connected component* M_i $(i \neq 1,r)$ is defined as a subnetwork such that it does not contain the source nor sink and that there is a route consisting of free arcs alone between every pair of nodes in it.

For any network N, a reduced network φN is defined as a network with the set of nodes $\hat{M}_i$ and with the set of arcs $(\hat{M}_i, \hat{M}_j)$, where each $\hat{M}_i$ corresponds to a connected component M_i in N, and the arc $(\hat{M}_i, \hat{M}_j)$ in φN is assigned the capacity $c_{k\ell}$ and the cost $d_{k\ell}$ of the arc (k,ℓ) in N such that $d_{k\ell} = \min \{d_{k'\ell'} \; ; \; k' \in M_i, \ \ell' \in M_j\}$. Hereafter, we shall call k the sink of M_i, ℓ the source of M_j. The source 1 in N is the source of M_1, and the sink n in N is the sink of M_r.

Obviously, the potential p_j (with respect to the modified relative costs) at each node j in the connected component M_i of the original network N equals the potential at the node $\hat{M}_i$ of the reduced network $\varphi \tilde{N}^*$.

In large-scale network problems, the reduction of an original network N to φN such as above will reduce considerably the computational labor in finding the shortest route.

7. INDUCTIVE ALGORITHM FOR MULTISOURCE-MULTISINK NETWORKS

In this section, we shall consider a multisource-multisink network $N(m;n)$ with sources $s_1, s_2, \ldots, s_m$ and sinks $t_1, t_2, \ldots, t_n$, along the lines of thought suggested by Iri in §24.2.3 of [5]. Let $a_i \geq 0$ denote the supply at the i-th source and $b_j \geq 0$ the demand at the j-th sink, where it is assumed that $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$. Let $N(k;\ell)$ denote the subnetwork with sources $s_1, \ldots, s_k$ $(k \leq m)$ and sinks $t_1, \ldots, t_\ell$ $(\ell \leq n)$ as well as with all the nodes and arcs that are contained in some routes of nonnull residual capacity from an s_i $(i \leq k)$ to a t_j $(j \leq \ell)$.

Instead of treating the entire network $N(m;n)$ from the outset, we solve the problems for partial networks $N(k;\ell)$ with $k \leq m$ and $\ell \leq n$. The partial network for which we seek the solution is enlarged step by step by attaching a source or sink, one at a time.

The inductive algorithm for a multisource-multisink network may briefly be stated as follows:

STEP 0. Start with $k = 1$, $\ell = 1$, putting $f_{ij} = 0$ for all arcs (i,j), $q_i = 0$ for all nodes i, and $u = 0$.

STEP 1. In the partial network $N(k;\ell)$, ship a total amount $\Delta v = \min\left(\sum_{i=1}^{k} a_i, \sum_{j=1}^{\ell} b_j\right) - u$ of incremental minimum-cost flow from the source s_k to the sink t_ℓ according to the algorithm stated in section 3 (exactly speaking, we should skip STEP 0 of that algorithm and start from STEP 1). If the new total flow value u attains v, then stop. Otherwise, go to STEP 2.

STEP 2.

(a) If the new total flow value u attains $\sum_{i=1}^{k} a_i$, then increase by 1 and consider the partial network $N(k;\ell)$ thus augmented. Put the potentials q_i at the node i of the new partial network $N(k;\ell)$, which were not contained in the old partial network $N(k-1;\ell)$, equal to one and the same value $q^+ = \max_{i,j} (q_j - d_{ij})$, where q_j's are the potentials at the nodes of the old partial network $N(k-1;\ell)$.[4] Otherwise do nothing. Then go to (b).

(b) If the new total flow value u attains $\sum_{j=1}^{\ell} b_j$, then increase by 1 and consider the partial network $N(k;\ell)$ thus augmented. Put the potentials q_j at the node j of the new partial network $N(k;\ell)$, which were not contained in the old partial network $N(k;\ell-1)$, equal to one and the same value $q^- = \min_{i,j} (q_i + d_{ij})$, where q_i's are the potentials at the nodes of the old partial network $N(k;\ell-1)$.[4)] Otherwise do nothing. Then return to STEP 1.

8. VALIDITY OF THE INDUCTIVE ALGORITHM

The proof of the validity of the inductive algorithm can be done by mathematical induction on iteration number as follows:

Since, at the outset, we have $\tilde{d}^*_{ij} = d_{ij} \geq 0$ for all i, j, we assume that we have nonnegative $\tilde{d}^*_{ij}$'s of the arcs (i,j) in $N(k;\ell)$ with respect to a flow configuration and a set of potentials at the beginning of STEP 1. Then we continue to have nonnegative $\tilde{d}^*_{ij}$'s during the course of the solution process in Step 1 by our algorithm stated in section 3. Therefore, it is sufficient to prove that at the end of STEP 2, we have nonnegative $\tilde{d}^*_{ij}$'s of those arcs (i,j) of the new partial network, which were not contained in the old partial network, with respect to the new potentials obtained in STEP 2.

There is one thing to be noted. The flows are not assigned along the arcs of the new partial network, which were not contained in the old partial network.

Therefore, in the case where the new total flow value u attains $\sum_{i=1}^{k} a_i$, since in the augmented partial network $N(k;\ell)$ there is no arc (i,j) of finite length such that $i \in N(k-1;\ell)$ and $j \notin N(k-1;\ell)$, we have the following modified relative costs of arcs (i,j) in $N(k;\ell)$ but not in $N(k-1;\ell)$:

---

4) Without loss of validity of our algorithm, we can replace q^+ by a number larger than q^+ and can replace q^- by a number smaller than q^-. Specifically, if we replace q^+ by $q^* = \max\{0, q^+\}$ during the course of the solution process, we can always replace q^- by zero.

$$\tilde{d}^*_{ij} = \begin{cases} d_{ij}+q^+ - q_j = d_{ij}-q_j+\max_{i,j}(q_j-d_{ij}) \ (\geq 0), \ \text{if } j \in N(k-1;\ell), \\ d_{ij}+q^+ - q^+ = d_{ij} \ (\geq 0), \ \text{if } j \notin N(k-1;\ell). \end{cases}$$

Similarly, in the case where the new total flow value u attains $\sum_{j=1}^{\ell} b_j$, since in the augmented partial network $N(k;\ell)$ there is no arc (i,j) of finite length such that $i \notin N(k;\ell-1)$ and $j \in N(k;\ell-1)$, we have the following modified relative costs of arcs (i,j) in $N(k;\ell)$ but not in $N(k;\ell-1)$:

$$\tilde{d}^*_{ij} = \begin{cases} d_{ij}+q_i-q^- = d_{ij}+q_i-\min(q_i+d_{ij}) \ (\geq 0), \ \text{if } i \in N(k;\ell-1), \\ d_{ij}+q^- - q^- = d_{ij} \ (\geq 0), \ \text{if } i \notin N(k;\ell-1). \end{cases}$$

Thus, all the $\tilde{d}^*_{ij}$'s of $N(k;\ell)$ at the end of STEP 2 have been shown to be nonnegative.

9. EXAMPLE FOR TECHNIQUES OF REDUCTION AND INDUCTION

As a numerical example of illustrating the usefulness of techniques of reduction and induction stated in the foregoing sections, consider the network in Figure 2-1, where the pairs of numbers (d_{ij}, c_{ij}) beside the arrows indicate the costs and the capacities of the arcs.

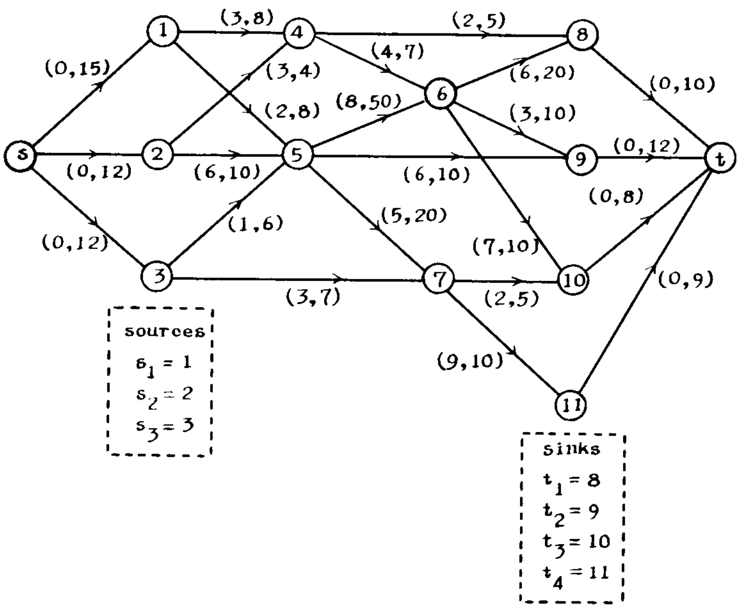

Fig. 2-1 Given network

Now we are interested in finding a maximum flow with a minimum cost in the given network. It is easily seen that this network is equivalent to the multisource-multisink network whose total amount of supplies and total amount of demands equal 39. The initial partial network is shown in Figure 2-2. Hereafter, the numbers over the forward arrows and under the backward arrows of the directed arcs (i,j) indicate the relative costs $\tilde{d}_{ij}$ and the modified capacities c^*_{ji}, respectively, the numbers over the lines incident to the sources or sinks indicate the supplies or demands to be augmented, respectively, the numbers beside the nodes indicate the potentials q_i, the bold line arcs indicate the shortest route at the respective stage, and the cancelled arrows —+— indicate the saturated arcs. Furthermore, a part encircled by a dotted line indicates a connected component in the modified network (i.e. a node of a reduced network) and the numbers attached to connected components indicate the potentials of the corresponding nodes in the reduced network.

Continuing the solution process, we have the following networks shown in Figures 2-2 through 2-15.

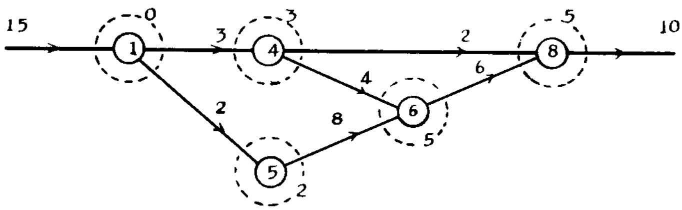

Fig. 2-2 $\tilde{N}^*$ (1;1) ($\Delta q x \Delta v$=5 x 5 = 25)

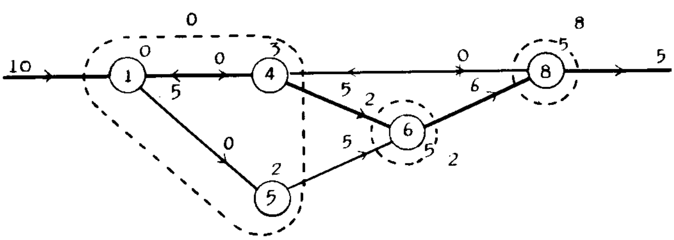

Fig. 2-3 $\tilde{N}^*$ (1;1) ($\Delta q x \Delta v$ = 13 x 3 = 39)

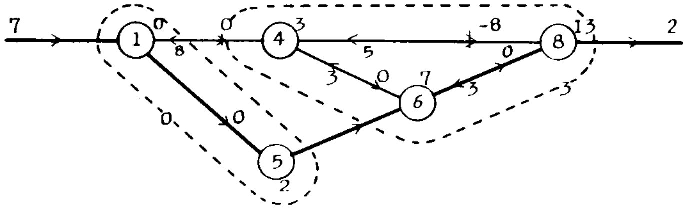

Fig. 2-4 $\tilde{N}^*$ (1 ; 1) ($\Delta q x \Delta v$ = 16 x 2 = 32)

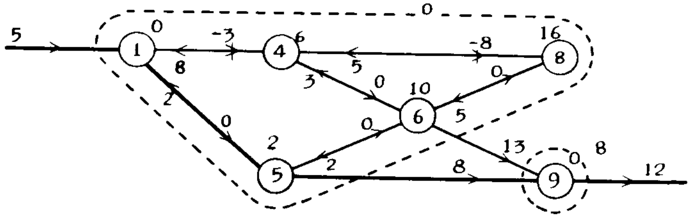

Fig. 2-5 $\tilde{N}^*$ (1 ; 2) ($\Delta q x \Delta v$ = 8 x 5 = 40)

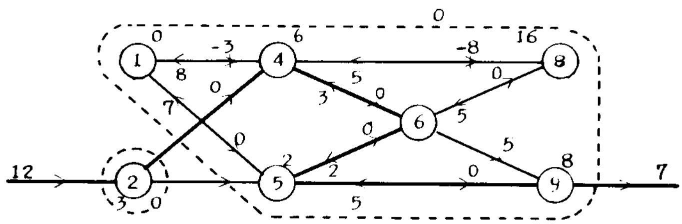

Fig. 2-6 $\tilde{N}^*$ (2 ; 2) ($\Delta q x \Delta v$ = 5 x 2 = 10)

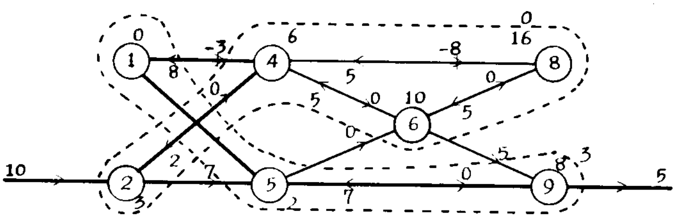

Fig. 2-7 $\tilde{N}^*$ (2 ; 2) ($\Delta q x \Delta v$ = 8 x 1 = 8)

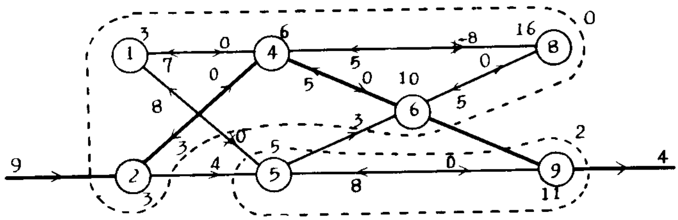

Fig. 2-8 $\tilde{N}^*$ (2 ; 2) ($\Delta q \times \Delta v$ = 10 x 1 = 10)

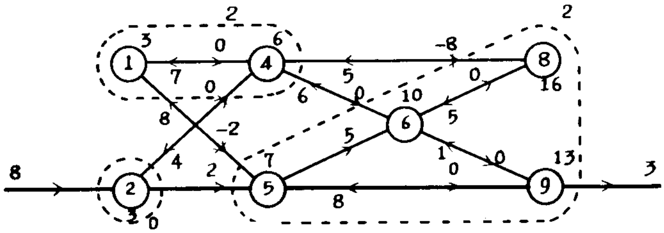

Fig. 2-9 $\tilde{N}^*$ (2 ; 2) ($\Delta q \times \Delta v$ = 12 x 2 = 24)

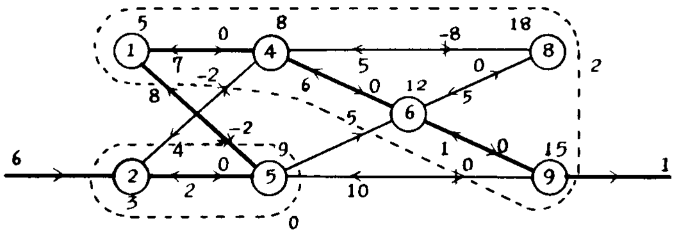

Fig. 2-10 $\tilde{N}^*$ (2 ; 2) ($\Delta q \times \Delta v$ = 14 x 1 = 14)

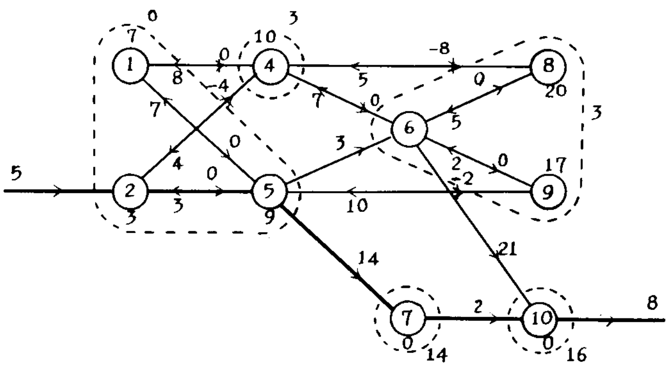

Fig. 2-11 $\tilde{N}^*$ (2 ; 3) ($\Delta q \times \Delta v$ = 13 x 5 = 65)

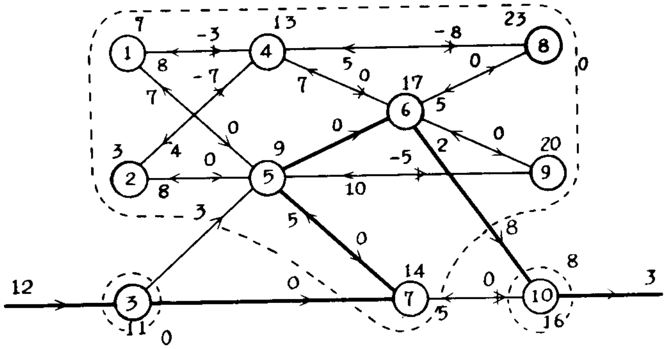

Fig. 2-12 $\widetilde{N}^*$ (3 ; 3) ($\Delta q \times \Delta v$ = 13 x 3 = 39)

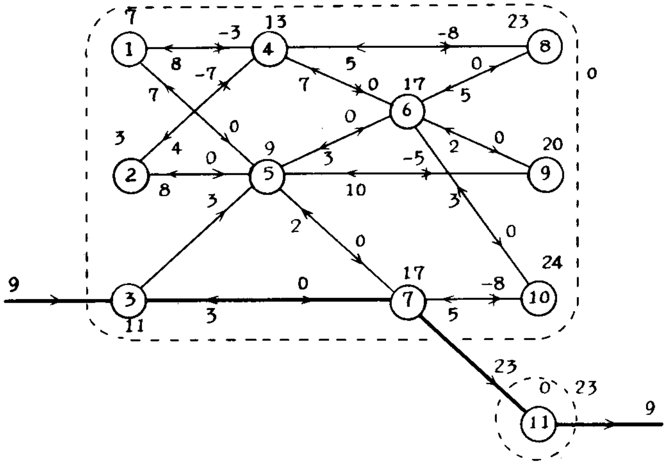

Fig. 2-13 $\widetilde{N}^*$ (3 ; 4) ($\Delta q \times \Delta v$ = 12 x 4 = 48)

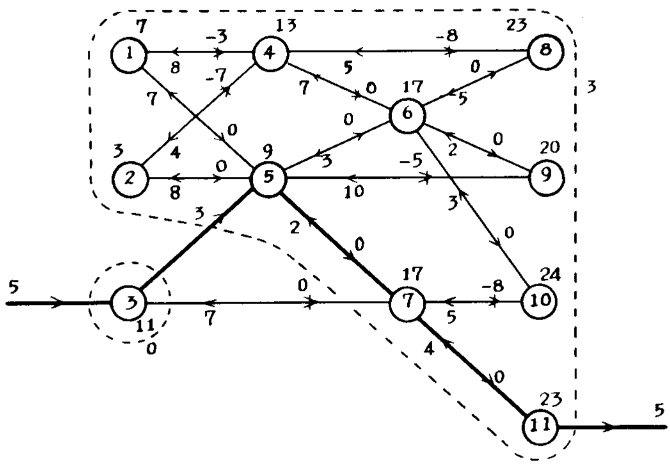

Fig. 2-14 $\widetilde{N}^*$ (3 ; 4) ($\Delta q \times \Delta v$ = 15 x 5 = 75)

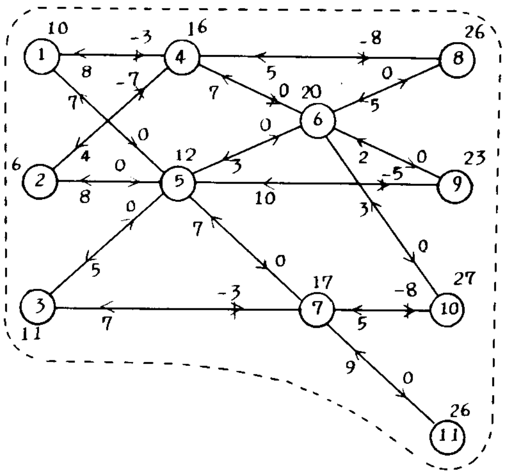

Fig. 2-15 $\tilde{N}^*$ ($\Delta q = \infty$)

The optimal flow configuration is given in Figure 2-16, and the total cost equals $\sum \Delta q \times \Delta v = 429$, where Δq is the potential difference between source and sink, and Δv is the incremental flow value along the shortest route.

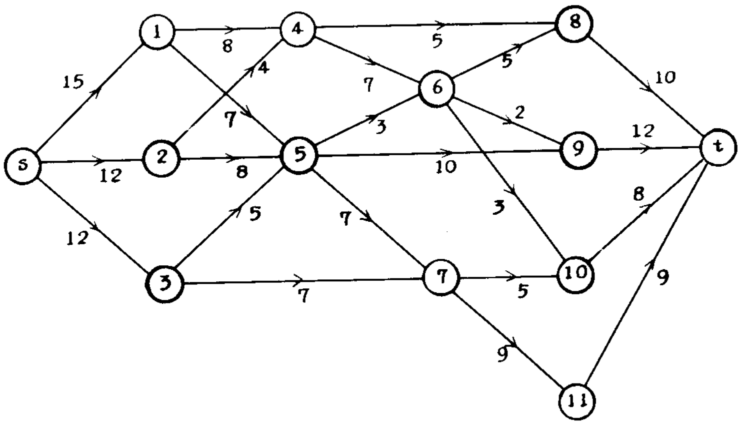

Fig. 2-16 Optimal flow configuration

10. HITCHCOCK-TYPE TRANSPORTATION PROBLEMS

Since the network for a *Hitchcock-type transportation problem* is a bipartite graph and is regarded as a multisource-multisink network, a *transportation problem* of the Hitchcock type, an *assignment problem* and a *matching problem* can be solved by using the tableaux of simple form incorporating the techniques of reduction and induction proposed in the preceding sections.

In particular, in the case of an assignment problem or a matching problem, the shortest-route problem for a partial network with 2r nodes is reduced to that for a network with r nodes by shortcircuiting the arcs contained in the current partial assignment or matching.

The solution of a shortest-route problem with r nodes requires r^2 additions and $2r^2$ comparisons [2]. Accordingly, a total number of about $\frac{1}{3} n^3$ additions and $\frac{2}{3} n^3$ comparisons are sufficient during the whole solution process for an n-by-n assignment or matching problem. Hence, our algorithm requires elementary operations proportional in number to at most n^3. This theoretical upperbound is better than the well-known standard methods according to which the bound is proportional to n^4 [3], [5].

Let us give an example which shows how our algorithm with reduction and induction techniques behaves itself for the Hitchcock-type problems using matrix notation and terminology.

A transportation problem of the Hitchcock type is described as the problem of finding an m-by-n nonnegative matrix f_{ij} that satisfies the row sum constraints

$$\sum_j f_{ij} = a_i (\geq 0), \quad i = 1, 2 \ldots, m \tag{12}$$

and the column sum constraints

$$\sum_i f_{ij} = b_j (\geq 0), \quad j = 1, 2, \ldots, n, \tag{13}$$

and minimizes

$$\sum_{i,j} d_{ij} f_{ij} \qquad (d_{ij} \geq 0), \tag{14}$$

where $\sum_i a_i = \sum_j b_j$.

If, for all i, j, we set $a_i = 1$, $b_j = 1$ and $m = n$, then we have the matrix representation of an assignment problem; furthermore, if, for all i, j, we set $d_{ij} = 0$ or 1, then we have a matching problem.

As a numerical example for the Hitchcock-type problems, for simplicity, let us consider the assignment problem given in Tableau 1-1.

i \ j	6	7	8	9	10
1	3	1	6	1	0
2	5	6	2	6	0
3	0	1	1	3	2
4	3	2	0	5	2
5	2	3	2	0	4

Tableau 1-1

The initial tableau corresponding to the initial partial network is shown in Tableau 1-2. Hereafter, each tableau corresponds to a partial network $N(k;\ell)$, its entries indicate the modified relative costs $\tilde{d}^{*}_{ij}$ $(= \tilde{d}_{ij})$, the entries in the last row and the last column indicate the potentials at the nodes of $\tilde{N}^{*}(k;\ell)$, and the entry at the lower right corner indicates the total cost of the partial assignment. Furthermore, the encircled row number and the encircled column number indicate the source and the sink of $\varphi\tilde{N}^{*}(k;\ell)$, respectively, and the encircled elements indicate the current partial assignment (they may be regarded also as the nodes of $\varphi\tilde{N}^{*}(k;\ell)$ in the shortest-route problem). The numbers beside the circiles indicate the potentials with respect to $\tilde{d}^{*}_{ij}$'s, and the elements marked with ▢ indicate the arcs contained in the shortest route from the source to the sink in $\varphi\tilde{N}^{*}(k;\ell)$. The potential q_i marked with asterisk indicates

$$q^{*} = \max\{0, \max_{i,j}(q_j - d_{ij})\} .$$

Continuing the solution process, we have the following tableaux shown in Tableaux 1-2 through 1-6.

i \ j	⑥ 3	q_i
① 0	[3]	0
q_j	0	0

Tableau 1-2

i \ j	6	(7)^3	q_i
1	(0)^2	[1]	0
(2)^0	[2]	6	0*
q_j	3	0	3

Tableau 1-3

i \ j	6	7	(8)^2	q_i
1	0	(0)^2	8	2
2	(0)^0	3	[2]	0
(3)^0	[0]	2	6	5*
q_j	5	3	0	6

Tableau 1-4

i \ j	6	7	8	(9)^5	q_i
1	2	(0)^0	8	[5]	4
2	0	1	(0)^1	6	0
3	(0)^1	1	4	8	5
(4)^0	1	[0]	1	8	3*
q_j	5	5	2	0	3

Tableau 1-5

i \ j	6	7	8	9	(10)^1	q_i
1	1	[0]	7	(0)^0	4	4
2	0	2	(0)^0	1	[1]	1
3	(0)^0	2	4	3	8	6
4	0	(0)^0	[0]	3	5	3
(5)^0	1	3	4	[0]	9	5*
q_j	6	5	3	5	0	5

Tableau 1-6

Thus we have an optimal solution shown in Tableau 1-7.

i \ j	6	7	8	9	10	q_i
1	1	(0)	7	0	3	4
2	0	2	0	1	(0)	1
3	(0)	2	4	3	7	6
4	0	0	(0)	3	4	3
5	1	3	4	(0)	8	5
q_j	6	5	3	5	1	1

Tableau 1-7

ACKNOWLEDGMENT

The author is indebted to Professor T. Asaka and Professor M. Iri of the University of Tokyo for suggestions, helpful discussions and valuable comments on the manuscript. He expresses his gratitude also to Mr. T. Kobayashi of the University of Tokyo for comments on the manuscript.

REFERENCES

1. Dijkstra, E. W., "A Note on Two Problems in Connection with Graphs," *Numerische Mathematik*, 1, 269-271 (1959).

2. Dreyfus, S. E., "An Appraisal of Some Shortest Path Algorithms," *Operations Research*, 17, 395-412 (1969).

3. Ford, L. R., Jr., and D. R. Fulkerson, *"Flows in Networks,"* Princeton University Press, Princeton, N. J., 1962.

4. Hu, T. C., *"Integer Programming and Network Flows,"* Addison-Wesley, Reeding, Mass., 1969.

5. Iri, M., *"Network Flow, Transportation and Scheduling: Theory and Algorithms,"* Academic Press, New York, 1969.

Paper received October 17, 1970