Journal of the Operations Research
Society of Japan
Vol. 26, No. 3, September 1983

THE LEXICO-SHORTEST ROUTE ALGORITHM FOR SOLVING THE MINIMUM COST FLOW PROBLEM WITH AN ADDITIONAL LINEAR CONSTRAINT

Takashi Kobayashi
Saitama University

(Received July 12, 1982; Final May 6, 1983)

Abstract This paper proposes a primal-dual method for solving the minimum cost flow problem with an additional linear constraint. Each branch of a given network has two dimensional distance associated with it. The first element is related to the cost, and the second to the coefficient of the additional constraint. The lexico-shortest route is defined as the route with the minimum distance in the lexicographical ordering. A loop with negative distance is called a lexico-negative loop. A lexico-negative loop such that the first element of its distance is negative, never appears. When there exists a lexico-negative loop such that the first element is zero, and that the second is negative, the current primal solution (flow) is improved by changing the flow around the loop. Otherwise, the lexico-shortest route exists and the current dual solution is improved. Our algorithm is a pure network algorithm in the meaning that we need not know what is the basis for the current solution. It is very suitable to the cases when degeneracies often occur.

1. Introduction

We often meet the network programming problems with additional linear constraints like bundle capacities or divisible activities [1,4]. When they have two or more additional constraints, we need some simplex method techniques [2,3,5,8]. But we can solve the maximal flow problem with a bundle, or the critical path problem with a divisible activity by the pure network algorithms with the help of multipliers [6,7]. For the minimum cost flow problem with an additional constraint, Masch [9] proposed the cyclic method based on the simplex procedure. Here, we show a pure network algorithm called the lexico-shortest route algorithm for solving it.

2. Problem Formulation

Let us consider a directed network G=(N,B) where N is the set of n nodes, numbered 1,2,..., n, node 1 is the source, node n is the sink, and B is the set of directed branches (pairs of nodes). Then, the minimum cost flow problem with an additional linear constraint is formulated as follows:

$P_0(\Theta)$: Minimize

$$z_0(x) = \sum_{(i,j)} c_{ij}x_{ij}$$

subject to

$$(2.1) \qquad \sum_i x_{ij} - \sum_i x_{ji} = \begin{cases} q_0 (i=1), \\ 0 \ (i=2,3,\dots,n-1), \\ -q_0 (i=n), \end{cases}$$

$$(2.2) \qquad 0 \leq x_{ij} \leq a_{ij} \ ((i,j)\varepsilon B),$$

$$(2.3) \qquad \sum_{(i,j)} b_{ij}x_{ij} \leq \Theta,$$

where q_0(total flow value) is a given positive number, and Θ is a parameter. Now, let

$$y(x) = \sum_{(i,j)} b_{ij}x_{ij},$$

and we shall consider the following parametric programming problem with a parameter ϕ instead of Θ.

$P(\phi)$: Minimize

$$z(x,\phi) = z_0(x) + \phi y(x)$$

subject to (2.1) and (2.2).

Let

$$c^*_{ij}(\phi) = c_{ij} + \phi b_{ij} \quad \text{for } (i,j)\varepsilon B.$$

Then

$$z(x,\phi) = \sum_{(i,j)} c^*_{ij}(\phi)x_{ij}.$$

Hence, $P(\phi)$ is a usual minimum cost flow problem when ϕ is given. The relations between the solutions of two problems are stated by the following theorems.

Theorem 1. Let $\bar{x}$ be an optimal solution of P(0). Then it is also optimal to $P_0(\Theta)$ for any Θ such that $\Theta \geqq y(\bar{x})$.

It is obvious from that $\bar{x}$ is feasible to $P_0(\Theta)$ if $\Theta \geqq y(\bar{x})$.

Theorem 2. Let $\bar{x}$ be an optimal solution of $P(\phi)$ for some positive ϕ. Then it is also optimal to $P_0(\eta)$, where $\eta = y(\bar{x})$.

Proof: $\bar{x}$ is feasible to $P_0(\eta)$. Let x' be any feasible solution of $P_0(\eta)$. x' is feasible to $P(\phi)$. Therefore,

$$z(\bar{x},\phi) - z(x',\phi) = (z_0(\bar{x}) - z_0(x')) + \phi(y(\bar{x}) - y(x')) \leqq 0.$$

Since $\phi > 0$ and $y(\bar{x}) = \eta \geqq y(x')$, $z_0(\bar{x}) - z_0(x') \leqq 0$. Q.E.D.

3. Primal-dual Method for $P(\phi)$

Consider the dual problem to $P(\phi)$.

$D(\phi)$: Maximize

$$w = q_0(v_n - v_1) - \sum_{(i,j)} a_{ij} u_{ij}$$

subject to

$$v_j - v_i - u_{ij} \leqq c^*_{ij}(\phi),\ u_{ij} \geqq 0, \quad \text{for every } (i,j) \varepsilon B.$$

Note that in any optimal solution of $D(\phi)$

$$u_{ij} = \max(v_j - v_i - c^*_{ij}(\phi),\ 0) \tag{3.1}$$

is satisfied for any $(i,j) \varepsilon B$. The complementary slackness conditions for $P(\phi)$ and $D(\phi)$ are as follows:

$$\begin{aligned} &x_{ij}(v_j - v_i - u_{ij} - c^*_{ij}(\phi)) = 0, \\ &(x_{ij} - a_{ij}) u_{ij} = 0, \end{aligned} \tag{3.2}$$

for any $(i,j) \varepsilon B$.

From (3.1), (3.2) can be replaced by

$$\begin{cases} \text{if } v_j - v_i < c^*_{ij}(\phi),\ x_{ij} = 0, \\ \text{if } v_j - v_i > c^*_{ij}(\phi),\ x_{ij} = a_{ij}, \end{cases} \tag{3.3}$$

or

$$\begin{cases} \text{if } x_{ij} = 0,\ v_j - v_i \leqq c^*_{ij}(\phi), \\ \text{if } 0 < x_{ij} < a_{ij},\ v_j - v_i = c^*_{ij}(\phi), \\ \text{if } x_{ij} = a_{ij},\ v_j - v_i \geqq c^*_{ij}(\phi). \end{cases} \tag{3.4}$$

Now, we shall consider the primal problem restricted by an optimal solution of $D(\bar{\phi})$, $\bar{v}$.

$RP(\bar{\phi},\bar{v})$: Minimize

$$\eta = \Sigma b_{ij}x_{ij}$$

subject to (2.1) and

$$x_{ij}=0 \qquad \text{for } (i,j)\in B_{d1},$$

$$0\leqq x_{ij}\leqq a_{ij} \qquad \text{for } (i,j)\in B_{d2},$$

$$x_{ij}=a_{ij} \qquad \text{for } (i,j)\in B_{d3},$$

where

$$B_{d1} = \{(i,j)\,|\,(i,j)\in B,\ \bar{v}_j-\bar{v}_i<c^*_{ij}(\bar{\phi})\},$$

$$B_{d2} = \{(i,j)\,|\,(i,j)\in B,\ \bar{v}_j-\bar{v}_i=c^*_{ij}(\bar{\phi})\},$$

$$B_{d3} = \{(i,j)\,|\,(i,j)\in B,\ \bar{v}_j-\bar{v}_i>c^*_{ij}(\bar{\phi})\}.$$

Any feasible solution of $RP(\bar{\phi},\bar{v})$ is a feasible solution of $P(\bar{\phi})$, and satisfies the complementary slackness conditions for $\bar{v}$.
Hence, we get the following theorem.

Theorem 3. Any feasible solution of $RP(\bar{\phi},\bar{v})$ is an optimal solution of $P(\bar{\phi})$.

Next, we shall consider the dual problem restricted by an optimal solution of $P(\bar{\phi})$, $\bar{x}$.

$RD(\bar{x})$: Maximize ϕ

subject to

$$v_j-v_i-\phi b_{ij}\leqq c_{ij} \qquad \text{for } (i,j)\in B_{p1},$$

$$v_j-v_i-\phi b_{ij}=c_{ij} \qquad \text{for } (i,j)\in B_{p2},$$

$$v_j-v_i-\phi b_{ij}\geqq c_{ij} \qquad \text{for } (i,j)\in B_{p3},$$

where

$$B_{p1} = \{(i,j)\,|\,(i,j)\in B,\ \bar{x}_{ij}=0\},$$

$$B_{p2} = \{(i,j)\,|\,(i,j)\in B,\ 0<\bar{x}_{ij}<a_{ij}\},$$

$$B_{p3} = \{(i,j)\,|\,(i,j)\in B,\ \bar{x}_{ij}=a_{ij}\}.$$

Theorem 4. Let $(\bar{\phi},\bar{v})$ be a feasible solution of $RD(\bar{x})$. Then, $\bar{v}$ is an optimal solution of $D(\bar{\phi})$.

Now, we show a primal-dual method for $P(\phi)$ (and $D(\phi)$).

Primal-dual method

Phase 1. Solve $P(0)$. Let $\bar{x}$ be its optimal solution. (If it does not exist, stop.)

Phase 2. (a) Solve $RD(\bar{x})$. If an optimal solution $(\bar{\phi},\bar{v})$ is obtained, go to (b). Otherwise (If ϕ is infinite), stop.

(b) Solve $RP(\bar{\phi},\bar{v})$. Let $\bar{x}$ be its optimal solution and go to (a).

4. Lexico-shortest Route Algorithm

We shall show the lexico-shortest route algorithm for solving $P(\phi)$. First, we introduce some definitions. Let

$$B^* = \{(i,j)\,|\,(i,j)\varepsilon B \text{ or } (i,j)\varepsilon B\}.$$

A sequence of nodes $(i_0, i_1, \ldots, i_r)$ such that $(i_{k-1},i_k)\varepsilon B^*$ $(k=1, 2, \ldots, r)$ is called a route from node i_0 to node i_r. A route with $i_o=i_r$ is called a loop. Suppose that for any $(i,j)\varepsilon B^*$, two dimensional vector d_{ij} is given as its distance. The distance of a route $R=(i_0, i_1, \ldots, i_r)$ is defined as

$$d(R) = d_{i_0 i_1} + d_{i_1 i_2} + \ldots + d_{i_{r-1} i_r}.$$

Definition 1. Route R is called *lexico-shorter* than route R' if $d(R)$ is lexicographically less than $d(R')$.

Definition 2. A route from node s to node t is called the *lexico-shortest route* (*LS route*) from node s to node t if any route from node s to node t is not lexico-shorter than it. The distance of the LS route from node s to node t is called the *lexico-shortest distance* (*LS distance*) from node s to node t.

Definition 3. A loop is called a *lexico-negative loop* (*LN loop*) if its distance is lexicographically less than (0,0).

Suppose that we know a pair of optimal solutions of $P(\bar{\phi})$ and $D(\bar{\phi})$, $\bar{x}$ and $\bar{v}$. For each branch (i,j), let

$$(4.1)\qquad d_{ij} = \begin{cases} (c^*_{ij}(\bar{\phi}), b_{ij}) & \text{if } \bar{x}_{ij}<a_{ij}, \\ (\infty,\infty) & \text{if } \bar{x}_{ij}=a_{ij}, \end{cases}$$

and

$$(4.2)\qquad d_{ji} = \begin{cases} (-c^*_{ij}(\bar\phi), -b_{ij}) & \text{if } \bar x_{ij}>0, \\ (\infty,\infty) & \text{if } \bar x_{ij}=0. \end{cases}$$

Since $P(\bar\phi)$ is a usual minimum cost flow problem with costs $(c^*_{ij}(\bar\phi))$, we get the following theorem.

Theorem 5. There exists no LN loop such that the first element of its distance is negative.

Proof: Suppose that there exists an LN loop, say L, such that the first element of its distance is negative. We divide the set of branches of the loop into two sets L^+ and L^- such that L^+ is the set of branches whose directions are identical with that of L, and that L^- is the complement of L^+. Then,

$$\bar x_{ij}<a_{ij} \qquad \text{for } (i,j)\varepsilon L^+,$$

$$\bar x_{ij}>0 \qquad \text{for } (i,j)\varepsilon L^-$$

and

$$\sum_{(i,j)\varepsilon L^+} c^*_{ij}(\bar\phi) - \sum_{(i,j)\varepsilon L^-} c^*_{ij}(\bar\phi) < 0.$$

Determine Δx by

$$(4.3)\qquad \Delta x = \min\{\min_{(i,j)\varepsilon L^+}(a_{ij}-\bar x_{ij}),\ \min_{(i,j)\varepsilon L^-}(\bar x_{ij})\}.$$

If we set

$$(4.4)\qquad \bar x'_{ij} = \begin{cases} \bar x_{ij}+\varepsilon & \text{if } (i,j)\varepsilon L^+, \\ \bar x_{ij}-\varepsilon & \text{if } (i,j)\varepsilon L^-, \\ \bar x_{ij} & \text{otherwise,} \end{cases}$$

for ε such that $0<\varepsilon\leqq\Delta x$, $(\bar x'_{ij})$ is a feasible solution of $P(\bar\phi)$, and

$$z(\bar x',\bar\phi)<z(\bar x,\bar\phi),$$

which contradicts that $\bar x$ is an optimal solution of $P(\bar\phi)$. Q.E.D.

Assume that there exists an LN loop, say L, such that the first element is 0, and that the second is negative. Let us define $(\bar x'_{ij})$ as shown in the above proof, and then $(\bar x'_{ij})$ is a feasible solution of $RP(\bar\phi,\bar v)$, and η is decreased by $\varepsilon|d_2(L)|$, where $d_2(L)$ is the second element of $d(L)$.

Theorem 6. If there exists an LN loop (such that the second element of its distance is negative), $\bar x$ is not optimal to $RP(\bar\phi,\bar v)$.

Next, suppose that there exists no LN loop. Then, there exist LS routes from node 1 to all nodes. Let t_j be the LS distance from node 1 to node j (j=1, 2, ..., n). $(v_j)=(t_j^{(1)})$ is optimal to D($\bar{\phi}$). For each branch (i,j), t_j is not greater than t_i+d_{ij}, and t_i is not greater than t_j+d_{ji} in the lexicographical ordering. Define $s_{ij}=(s_{ij}^{(1)}, s_{ij}^{(2)})$ by

$$(4.5) \qquad s_{ij} = t_i + (c_{ij}^*(\bar{\phi}),\ b_{ij}) - t_j \qquad \text{for } (i,j)\varepsilon\text{B}.$$

If $0<\bar{x}_{ij}<a_{ij}$,

$$d_{ij} = -d_{ji} = (c_{ij}^*(\bar{\phi}),\ b_{ij}).$$

Hence, $s_{ij}=(0,0)$. If $\bar{x}_{ij}=0$, s_{ij} is not exicographically less than (0,0).

Therefore, $s_{ij}^{(1)}=0$ implies $s_{ij}^{(2)}\geq 0$. If $\bar{x}_{ij}=a_{ij}$, and $s_{ij}^{(1)}=0$, $s_{ij}^{(2)}\leq 0$.

Let $\quad \text{B}_0 = \{(i,j)\,|\,(i,j)\varepsilon\text{B},\ s_{ij}^{(1)}\cdot s_{ij}^{(2)}<0\}$,

and determine $\Delta\phi$ by

$$(4.6) \qquad \Delta\phi = \min_{(i,j)\varepsilon \text{B}_0} (-s_{ij}^{(1)}/s_{ij}^{(2)}).$$

If B_0 is empty, let $\Delta\phi=\infty$. Then, for λ such that $0\leq\lambda\leq\Delta\phi$,

$$(4.7) \qquad \phi = \bar{\phi}+\lambda,\ (v_j) = (t_j^{(1)}+\lambda t_j^{(2)})$$

is a feasible solution of RD($\bar{x}$).

Theorem 7. If there exists no LN loop, $(\bar{\phi},\bar{v})$ is not optimal to RD($\bar{x}$).

Now, we show our algorithm. If has two phases corresponding to those of the primal-dual method.

LS route algorithm

Phase 1.

Let x=0 and q=0, and iterate step 1 and step 2 alternately.

Step 1. For each branch (i,j), determine d_{ij} and d_{ji} by (4.1) and (4.2), where $\bar{\phi}$=0 and $\bar{x}_{ij}$ is the current value of $\bar{x}_{ij}$. Find the LS route from node 1 to node n. (If it doesn't exist, stop.)

Step 2. If it's possible to increase the flow along the route by q_0-q, do so and go to phase 2. Otherwise, increase it as much as possible, add the increment to q and go back to step 1.

Phase 2.

Let $(\bar{v}_j)=(t_j^{(1)})$, where $t_j^{(1)}$ is the first element of the LS distance from node 1 to node j at termination of phase 1. (It is an optimal solution of D(0).)

Step 1. For each branch (i,j), determine d_{ij} and d_{ji} by (4.1) and (4.2). Obtain the LS distances from node 1 to all node. If impossible (there exists an LN loop), go to step 2. If possible, go to step 3.

Step 2. (Improving the solution of RP($\bar{\phi},\bar{v}$).)
Let L be the LN loop. Determine Δx by (4.3). Get an improved solution by (4.4) for $\varepsilon=\Delta x$. Go back to step 1.

Step 3. (Improving the solution of RD($\bar{x}$).)
Let t_j be the LS distance from node 1 to node j.
Determine s_{ij} by (4.5) for each branch, and $\Delta\phi$ by (4.6). If $\Delta\phi=\infty$, stop. Otherwise, get an improved solution by (4.7) for $\lambda=\Delta\phi$. Go back to step 1.

Remark 1. For finding the LS route, we can use the usual shortest route algorithms by introducing the lexicographical order.

Remark 2. In step 1 of the first iteration of phase 2, we can get the LS distances. Hence, we always go to step 3. Except in the first iteration, we do not know which we solve, RP($\bar{\phi},\bar{v}$) or RD($\bar{x}$), at the start of step 1.

Remark 3. In step 3 of phase 2, for a branch (i,j) such that it is on the LS route from node 1 to some node, $s_{ij}=0$.

Remark 4. We can determine d's by only the value of ϕ. Hence, we need not have the values of v's explicitly.

Remark 5. If q_0 is the maximal flow value, the distances from node 1 to some nodes are not finite. Then, obtain the LS distances from node n in step 1 of phase 2.

5. Illustrative Example

To illustrate our algorithm, consider the network of Fig. 1.

Phase 1.

We start with $x=0$ and $q=0$.

Iteration 1.

Step 1. For each branch (i,j), $d_{ij}=(c_{ij},b_{ij})$ and $d_{ji}=(\infty,\infty)$. LS distances from node 1 are indicated by bracketed numbers and LS routes by broken lines in Fig. 2(a). The LS route from node 1 to node 7 is (1,2,3,4,5,7).

Step 2. Δx=10, and the new flow (x_{ij}) is shown in Fig. 2(b).

Iteration 2.

Step 1. For the current flow, d's are defined by (4.1) and (4.2).
In Fig. 3(a), for each branch (i,j), (c_{ij},b_{ij}) and its condition (if $x_{ij}=0$, $0<x_{ij}<a_{ij}$ or $x_{ij}=a_{ij}$) are shown. We can know d_{ij} and d_{ji} from them. For example, from $0<x_{12}<a_{12}$, $d_{12}=(5,0)$ and $d_{21}=(-5,0)$. Since $x_{23}=a_{23}$, $d_{23}=(\infty,\infty)$ and $d_{32}=(-3,0)$. The LS route is (1,2,6,7).

Step 2. Δx=40, and $q=50=q_0$. We get the flow shown in Fig. 3(b), which is an optimal solution of P(0), and terminate phase 1.

Phase 2.

Iteration 1.

Step 1. We get the LS route (1,3,4,5,7) as shown in Fig. 4(a). Therefore, we can increase the value of ϕ(improve the current solution of RD($\bar{x}$)).

Step 2. (s_{ij}) are determined as in Fig. 4(b) except on LS routes. The elements (branches) of B_0 are indicated by symbol o. $\Delta\phi$ is determined by

$$\Delta\phi=\min(6/4,\ 6/3)=1.5.$$

The new value of ϕ is 1.5.

Iteration 2.

Step 1. We replace c_{ij} by $c^*_{ij}(1.5)=c_{ij}+1.5b_{ij}$. Then, we find the LS route (1,3,4,5,7) (Fig. 5(a)). (Note that the route is the same that we got in the last iteration, but that the LS distance is greater.)

Step 3. $\Delta\phi$=0.5, and ϕ is increased to 2(=1.5+0.5) (Fig. 5(b)).

Iteration 3.

Step 1. When we replace $c^*_{ij}(1.5)$ by $c^*_{ij}(2)$, an LN loop L=(2,4,6,2) exists as shown in Fig. 6(a). d(L)=(0,-3). We can decrease the value of η (improve of the current solution of RP($\bar{\phi},\bar{v}$) of Fig. 3(b)).

Step 2. Δx is determined by

$$\Delta x=\min(a_{24}-x_{24},a_{46}-x_{46},x_{26})=\min(30,10,40)=10.$$

The new flow is shown in Fig. 6(b). Then, η=290-3x10=260.

Iteration 4.

Step 1. See Fig. 7(a). The LS route is (1,3,4,5,7).

Step 2. See Fig. 7(b). $\Delta\phi=2/2=1$, and the new value of ϕ is 3.

Iteration 5.

Step 1. See Fig. 8(a). There exists an LN loop L=(2,4,5,6,2), and d(L)=(0,-2).

Step 2. $x=\min(a_{24}-x_{24},a_{45}-x_{45},a_{56}-x_{56},x_{26})=\min(20,30,30,40)=20$
Fig. 8(b) shows the new flow with η=220(=260-20x20).

Iteration 6.

Step 1. See Fig. 9(a). There exists an LN loop L=(1,3,4,5,6,2,1).

Step 2. $\Delta x=10$, and the new flow is shown in Fig. 9(b).

Iteration 7.

Step 1. See Fig. 10(a). We can find the LS route (1,2,6,5,7).

Step 2. (s_{ij}) are shown in Fig. 10(b). Since B_0 is empty, $\Delta\phi=\infty$, and we terminate our algorithm.

Fig. 11 shows the locus of (ϕ,η). For (ϕ,η) on the locus, there exists an optimal solution of P(ϕ) with $y(x)=\eta$, which is optimal to $P_0(\eta)$. On a horizontal segment, the solution of P(ϕ) does not vary. For a point on a vertical segment, we can obtain a solution of $P_0(\eta)$ by the linear interpolation of two solutions of P(ϕ) corresponding to terminal points of the segment.

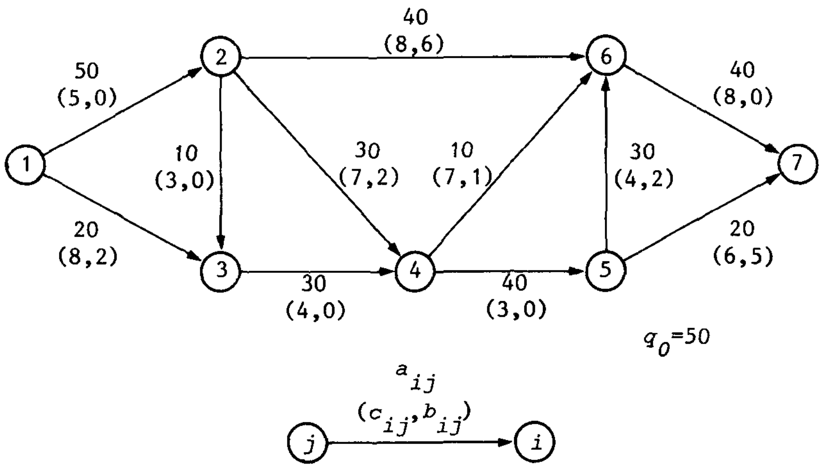
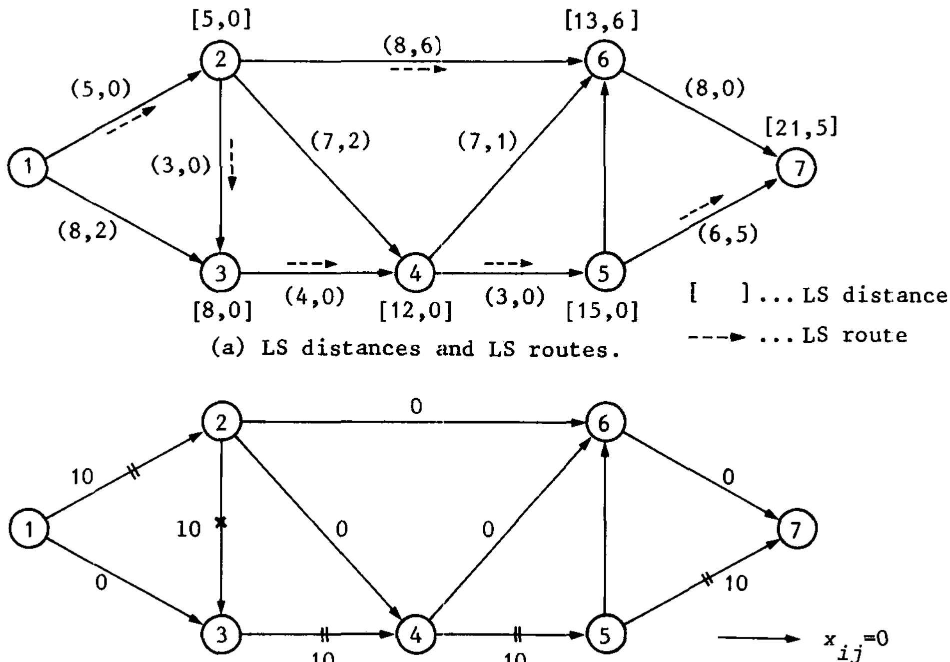


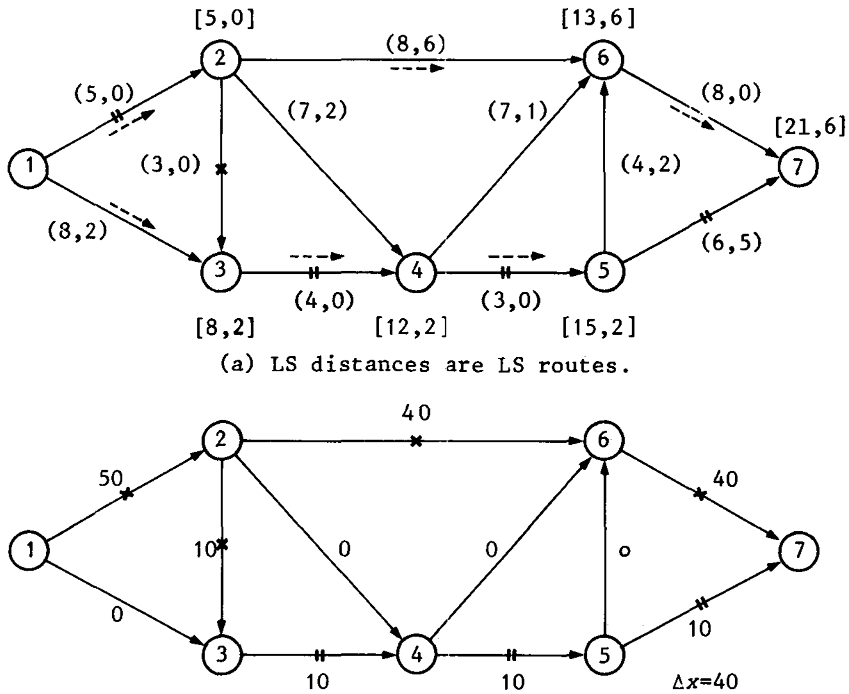
Fig. 1. Example of a network.



(a) LS distances and LS routes.

(b) Flow (q=10, η=50).

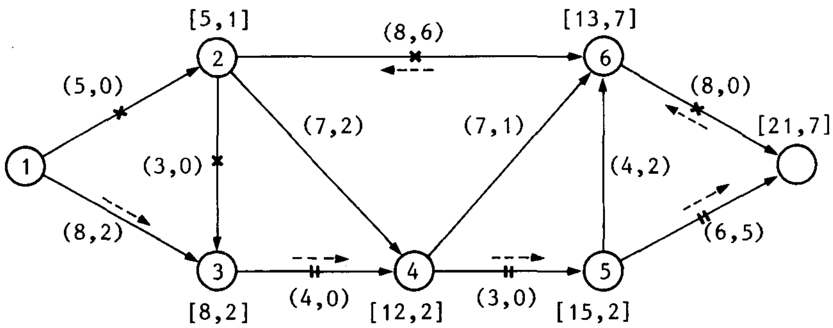
Fig. 2. Iteration 1 of Phase 1.



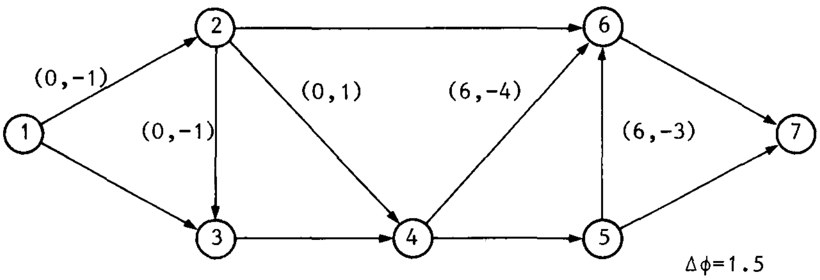
(a) LS distances are LS routes.

(b) Flow (q = 50, η=290).

Fig. 3. Iteration 2 of phase 1.

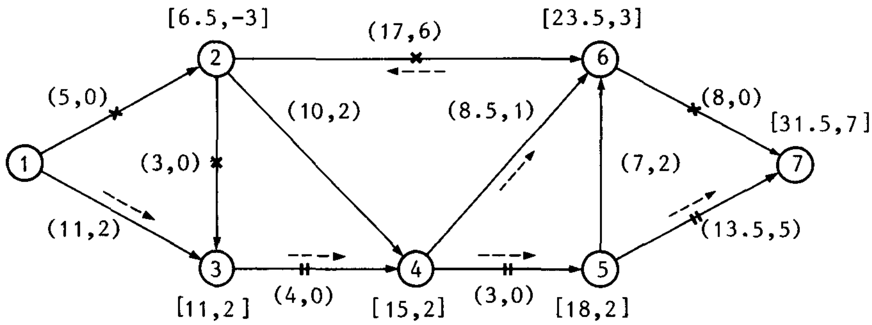


(a) LS distances and LS routes.

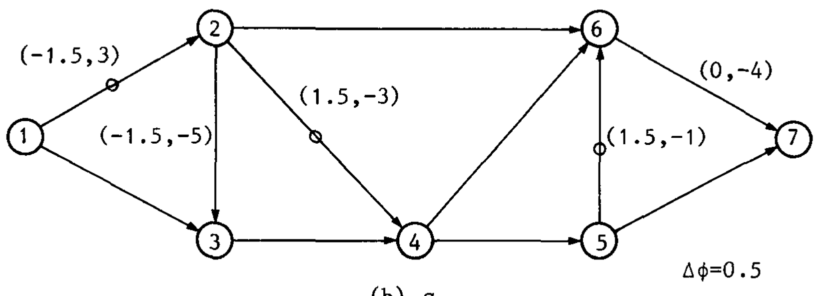


(b) s_{ij}

Fig. 4. Iteration 1 (ϕ=0).

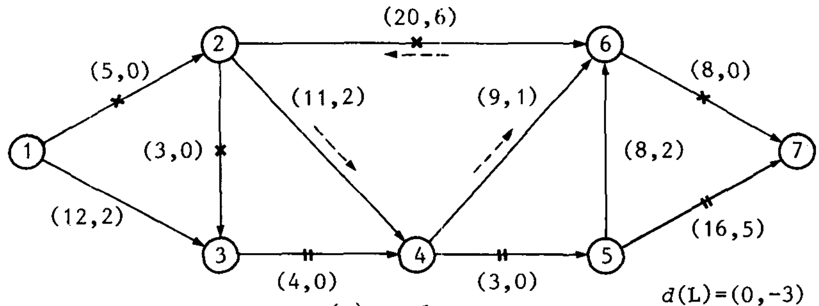


(a) LS distances and LS routes.

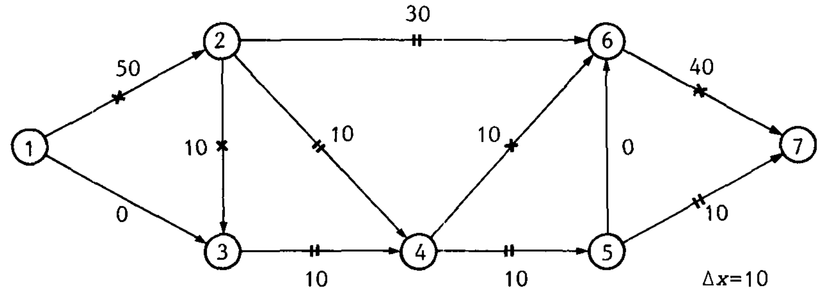


(b) s_{ij}

Fig. 5. Iteration 2 (ϕ=1.5).

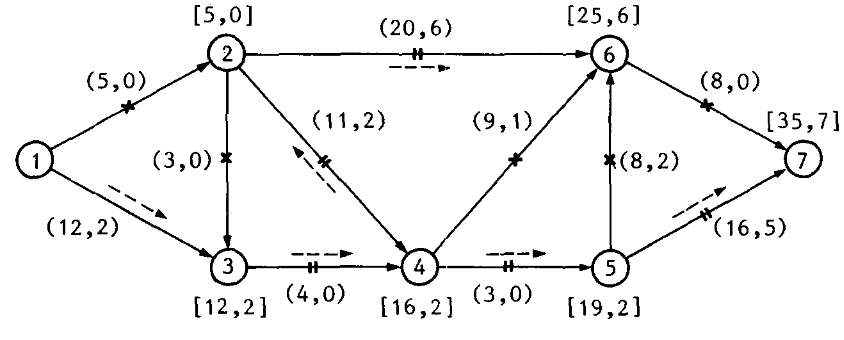


(a) LN loop.

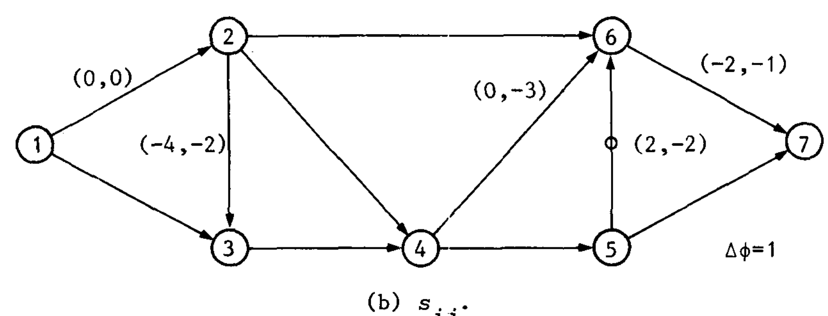


(b) Flow (η=260).

Fig. 6. Iteration 3 (ϕ=2).

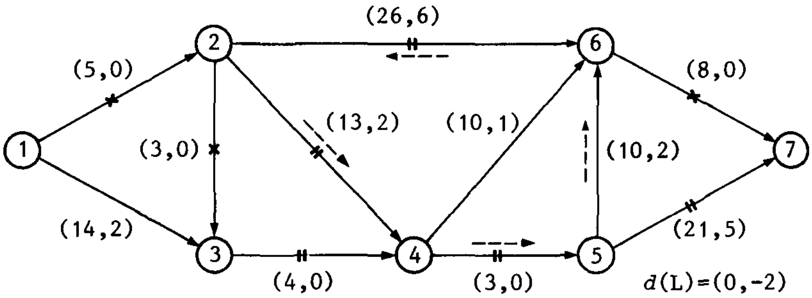


(a) LS distances and LS routes.

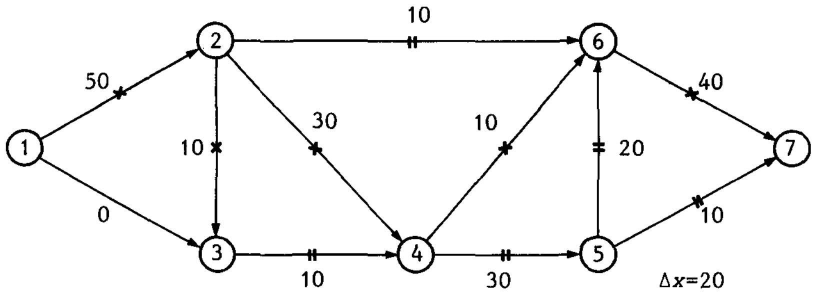


(b) s_{ij}.

Fig. 7. Iteration 4 (ϕ=2).

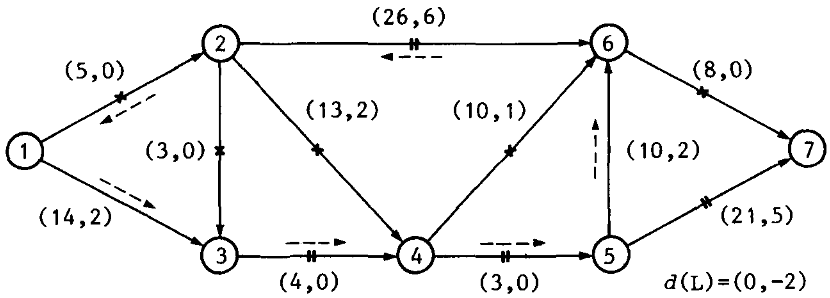


(a) LN loop.

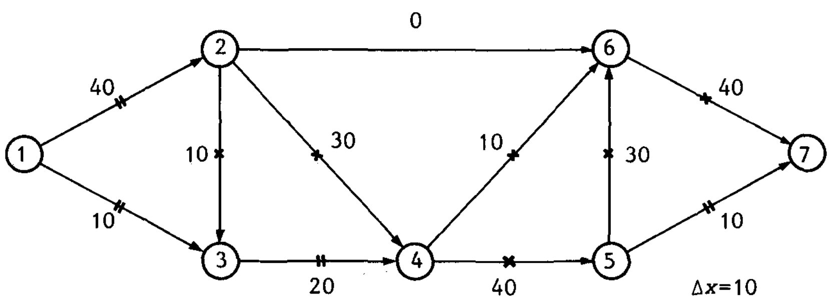


(b) Flow (η=220).

Fig. 8. Iteration 5 (ϕ=3).

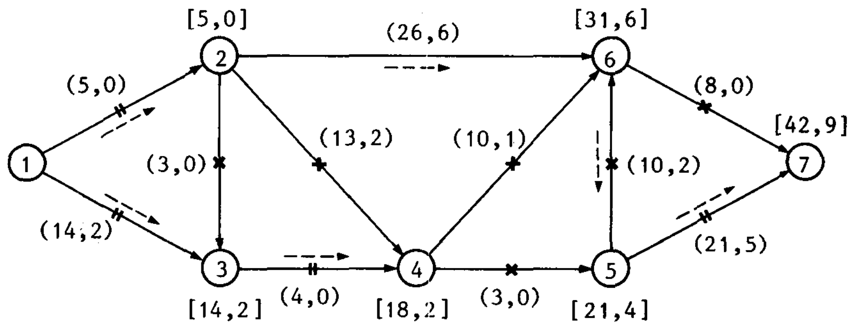


(a) LN loop.

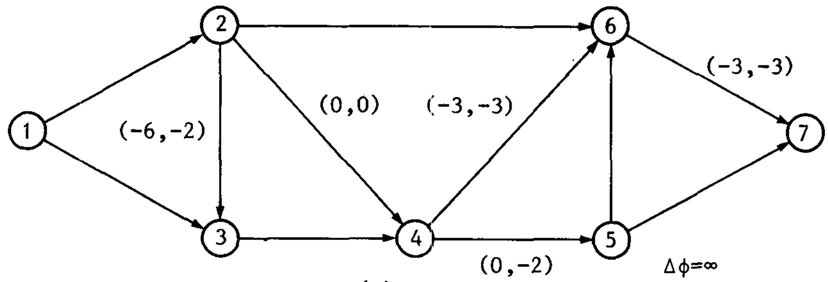


(b) Flow (η=200).

Fig. 9. Iteration 6 (ϕ=3).



(a) LS distances and LS routes.



(b) s_{ij}.

Fig. 10. Iteration 7 (ϕ=3).

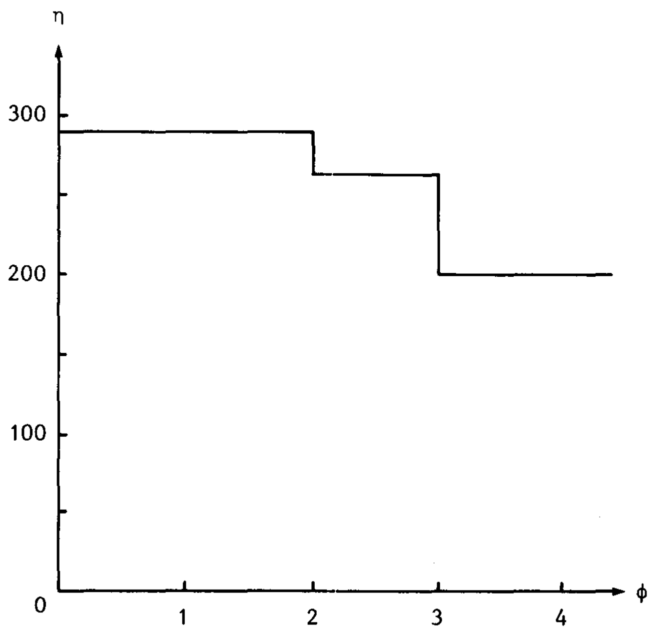


Fig. 11. Locus of (ϕ,η).

6. Concluding remarks

The differences between our algorithm and the cyclic method modified for our problem are as follows.

(1) The cyclic method is based on the simplex procedure. Therefore, it always keeps the basis which is represented by a tree. The "node potential" variables which correspond to $(t_j^{(1)}, t_j^{(2)})$ in our algorithm are computed by the equations:

$$t_j^{(1)}-t_i^{(1)}=c_{ij}^*(\phi),$$

$$t_j^{(2)}-t_i^{(2)}=b_{ij}$$

for every (i,j) in the tree. For example, let us consider the basis for the solution shown in Fig. 3(b). Since it is degenerate (three basic variables are zero), there are many trees to be the basis. If we choose

$$T = \{(1,2),(2,3),(2,6),(3,4),(4,5),(5,7)\}$$

among them, node potentials are computed as shown in Fig. 12.

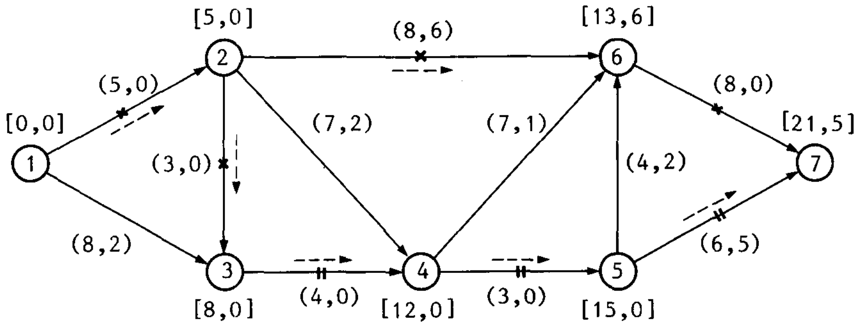


Fig. 12. A basic tree and node potentials.

$$x_{67}=40=a_{67},$$

$$s_{67}^{(1)}=13+8-21=0,$$

and $$s_{67}^{(2)}=6+0-5=1.$$

Hence, (6,7) enters T instead of (2,3), but neither ϕ increases, nor η decreases. In general, it takes some iterations before getting out a degenerate solution because only one basic branch is exchanged by a non-basic branch at each iteration, and some extra technique is necessary for avoiding the circling as Masch mentioned. On the other hand, we do not keep any tree, but we make a tree such that ϕ strictly increases whenever no LN loop exists by the shortest route algorithm using lexicographical ordering, and it may have two or more branches which do not belong to the former tree. From the above, our algorithm is very suitable for solving problems such that degeneracies often occur in applying the simplex procedure.

(2) We can get the best solution of P(0) by using the lexicographical ordering. Here, the best means that the solution which minimizes η among the optimal solutions of P(0). In the cyclic method, the initial solution is not always the best because coefficients b's are not used for obtaining it.

Our algorithm is also applied to a network flow problem with two objectives like cost and time. Suppose that we wish to minimize

$$z_c(x) = \Sigma c_{ij}x_{ij},$$

and

$$z_b(x) = \Sigma b_{ij}x_{ij}$$

subject to (2.1) and (2.2). In general, it is impossible to minimize two objective functions in the same time. So, let us combine them as

$$z(x) = z_c(x) + \phi\cdot z_b(x).$$

Then, the problem is P(ϕ). Therefore, we can get optimal solutions for every ϕ by our algorithm. Here, we note that the network flow problem with two objectives may be an assignment problem or a shortest route problem.

Acknowledgements

The author is very grateful to the refrees for their valuable comments on this paper.

References

[1] Berge, C. and Ghouila-Houri, A.: *Programming, Games and Transportation Networks*. Methuen, London, 1965.

[2] Chen, S. and Saigal, R.: A Primal Algorithm for Solving a Capacitated Network Flow Problem with Additional Linear Constraints. *Networks* Vol.7, No.1 (1977), 59-79.

[3] Iri, M. and Kobayashi, T.: *Network Theory* (in Japanese). Nikkagiren-shuppansha, Tokyo, 1976.

[4] Jewell, W. S.: Divisible Activities in Critical Path Analysis. *Operations Research* Vol.13 (1965), 747-760.

[5] Klingman, D. and Russell, R.: Solving Constrained Transportation Problems. *Operations Research* Vol.23, No.1 (1975), 91-106.

[6] Kobayashi, T.: Note on the Critical Path Analysis for a Project with a Divisible Activity. *Journal of the Operation Research Society of Japan* Vol.9, No.3 & 4 (1967), 137-140.

[7] Kobayashi, T.: On Maximal Flow Problem in a Transportation Network with a Bundle. *Journal of the Operation Research Society of Japan* Vol.10, No.3 & 4 (1968), 69-75.

[8] Kobayashi, T.: Critical Path Analysis for a Project with Divisible Activities. *Journal of the Operation Research Society of Japan* Vol.11, No.2 (1968), 55-65.

[9] Masch, V. V.: The Cyclic Method of Solving the Transshimpment Problem with an Additional Linear Constraint. *Networks* Vol.10, No.1 (1980), 17-31.

Takashi KOBAYASHI: Graduate School for Policy Science, Saitama University, Shimo-okubo, Urawa, Saitama, 338, Japan.