

ASSOCIATIVE SHORTEST AND LONGEST PATH PROBLEMS

Maruyama, Yukihiro

Department of General Education, Faculty of Economics, Nagasaki University

<https://doi.org/10.5109/13487>

出版情報 : Bulletin of informatics and cybernetics. 31 (2), pp.147-163, 1999-12. Research
Association of Statistical Sciences

バージョン :

権利関係 :

ASSOCIATIVE SHORTEST AND LONGEST PATH PROBLEMS

By

Yukihiro MARUYAMA *

Abstract

In the paper we consider a wide class of shortest path problems where the length of a path is defined through various associative binary operations. Solving a system of two interrelated recursive equations, we simultaneously find both shortest and longest path lengths. We show the existence and uniqueness of the solution of the system. Further, we propose an algorithm which solves the class of shortest path problems.

1. Introduction

In the so-called shortest path problem, the objective is to find a path of minimum length from an origin node 1 to a destination node N in a network $G(V, A)$; V and A are finite sets of nodes numbered $1, 2, \dots, N$ and of directed arcs, respectively. With each arc $(i, j) \in A$ an arc length (or cost, ...) t_{ij} is associated. Many authors have studied the problem in which the length of a path is the sum of its arc lengths (Dreyfus (1969), Bellman, Esogbue and Nabeshima (1982)): *additive problem*. They solved the additive problem by using the recursive equation

$$f_i = \min_{j \in D(i)} [t_{ij} + f_j], \quad i \neq N, \quad f_N = 0, \quad (1.1)$$

where

$$f_i = \min_p [t_{ij} + t_{jk} + \dots + t_{mN}], \quad p = (i, j, k, \dots, m, N), \quad D(i) = \{j \in V | (i, j) \in A\}.$$

Since the addition $+$ satisfies the monotonicity condition (nondecreasingness with respect to the second variable) on R^1 , the recursive equation (1.1) holds even if t_{ij} may be negative.

Some authors considered the problem where the length of a path is the multiplication of its arc lengths (Iwamoto (1987), Smith (1991), Sniedovich (1992)): *multiplicative problem*. They have solved the multiplicative problem under the restriction that each arc length takes the nonnegative value ($t_{ij} \geq 0$). Since the restriction implies the monotonicity in the operation $\circ = \times$, the multiplicative problem can be also solved by the

* Department of General Economics, Faculty of Economics, Nagasaki University, 4-2-1, Katafuchi, Nagasaki 850-8506, Japan

single recursive equation

$$f_i = \min_{j \in D(i)} [t_{ij} \times f_j], \quad i \neq N, \quad f_N = 1. \quad (1.2)$$

where

$$f_i = \min_p [t_{ij} \times t_{jk} \times \cdots \times t_{mN}], \quad p = (i, j, k, \dots, m, N).$$

But for the case some $t_{ij} < 0$, the equation (1.2) does not necessarily hold (see Remark in Section 2). Recently, Iwamoto (1993) has proposed dynamic programming which contains the multiplicative programming and the multiplicatively additive one. So, through the dynamic programming, Maruyama (1996) has solved the multiplicatively additive shortest path problem, where a pair of arc length and discount rate is associated with each arc. The shortest path problem involves the multiplicative shortest path problem with a negative arc length as a special case. However, these papers (Iwamoto (1993), Maruyama (1996)) treated only the optimization problems for objective functions with the multiplicatively additive value and the multiplicative one.

In this paper, we consider a wide class of shortest path problems (called *associative shortest path problem* (ASP)) with the length

$$t_{ij} \circ t_{jk} \circ \cdots \circ t_{mN}, \quad (1.3)$$

for a path $p = (i, j, k, \dots, m, N)$ where $\circ : R \times R \rightarrow R$ is an associative binary operation: $(x \circ y) \circ z = x \circ (y \circ z)$. The problem (ASP) includes not only the multiplicative problem with a negative arc length but also many other problems, for example, maximum problem, multiplicative-additive problem, fractional problem and so on, which were not studied in Iwamoto (1993) or Maruyama (1996). On the other hand, Frieze (1977) considered a wider class of shortest path problems where path length is defined as a real valued function defined on paths; the class contains the additive problem, the maximum problem and problem with time dependent arc lengths. Under certain monotonicity condition, the class of problems was solved by using some single recursive equation. Further, in Maruyama (1997), we derived a necessary and sufficient condition for the associative shortest path problem (ASP) to admit the single recursive equation

$$f_i = \min_{j \in D(i)} [t_{ij} \circ f_j], \quad i \neq N, \quad f_N = R(\circ), \quad (1.4)$$

where

$$f_i = \min_p [t_{ij} \circ t_{jk} \circ \cdots \circ t_{mN}]$$

and $R(\circ)$ is a right identity: $a \circ R(\circ) = a \quad \forall a \in S$ and solved the problem (ASP) by the Ford's procedure under the necessary and sufficient condition; of course, (1.4) does not hold for (ASP) which does not satisfy the condition, (see Remark and Examples in Section 2).

In the present paper, we will solve the problem (ASP) without the monotonicity conditions but with a more generalized monotonicity (*bitonicity*) on the associative operation \circ . We derive a system of two interrelated recursive equations in place of (1.4);

the system holds for the problem which does not admit the single recursive equation (1.4), that is, the problem excluded in Maruyama (1997). Solving the system, we find both shortest and longest path lengths. Though this approach is also based on dynamic programming, it is different from one in Maruyama (1996). Using an algebraic structure (semigroup) with bitonicity with respect to the binary operations, we give a dynamic programming formulaion for the wide class of problems; however, in Maruyama (1996), the formulation was obtained only for the specific case (multiplicatively additive problem).

In Section 2, we discuss the uniqueness of a system of two recursive equations in the problem (ASP). Furthermore, we give several examples of the problem (ASP): maximum problem, multiplicative problem, multiplicative-additive problem and two types of fractional problems. It is shown that the maximum problem can be solved through the single recursive equation (1.4). On the other hand, the multiplicative problem with negative arc lengths, the multiplicative-additive problem and the fractional problems admit a system of two recursive equations but, in general, not the single one.

In Section 3, we present an algorithm (bidecision algorithm) which solves the problem (ASP) with the bitonicity conditions; the algorithm is applicable to the problem which can not be solved by the Ford's procedure. Moreover, we investigate the complexity of the algorithm for each type of problem.

2. Existence and Uniqueness

In this section we derive a system of interrelated recursive equations through dynamic programming (Iwamoto (1993)). Moreover we show the existence and uniqueness of the solution to the system, which was proved by Bellman, Esogbue and Nabeshima (1982) for the additive problem (for the multiplicative problem, see Maruyama (1996)).

Throughout this paper we assume the following:

- (H0) network $G(V, A)$ contains no cycles;
- (H1) each t_{ij} belongs to a nonempty set $S \subset R$, where (S, \circ) is a semigroup:
 $\circ : S \times S \rightarrow S$ is an associative operation;
- (H2) there exists an element $R(\circ) \in S$, called a right identity:
 $a \circ R(\circ) = a \quad \forall a \in S$;
- (H3) the associative operation $\circ : S \times S \rightarrow S$ satisfies *bitonicity*:
 - (a) S is the disjoint union of A^+ , $A^- : S = A^+ + A^-$,
 - (b) $a \in A^+, a_1, a_2 \in S, a_1 < a_2 \implies a \circ a_1 \leq a \circ a_2$,
 - (c) $a \in A^-, a_1, a_2 \in S, a_1 < a_2 \implies a \circ a_1 \geq a \circ a_2$.

To solve the original problem (ASP):

$$\min_p [t_{1i} \circ t_{ij} \circ \dots \circ t_{mN}], \quad p = (1, i, j, \dots, m, N),$$

we imbed it into the following family of problems of finding both the shortest and longest

path from $i (\neq N)$ to N :

$$f_i = \min_p [t_{ij} \circ t_{jk} \circ \cdots \circ t_{mN}], \quad (2.1)$$

$$F_i = \text{Max}_p [t_{ij} \circ t_{jk} \circ \cdots \circ t_{mN}], \quad i \neq N, \quad (2.2)$$

$$f_N = F_N = R(o), \quad (2.3)$$

where p is a path from i to N :

$$p = (i, j, k, \dots, m, N).$$

Then we have a system of two interrelated recursive equations.

THEOREM 2.1. *Under the assumptions (H0) \sim (H3), it holds that*

$$f_i = \min_{j \in D^+(i)} [t_{ij} \circ f_j] \wedge \min_{j \in D^-(i)} [t_{ij} \circ F_j], \quad (2.4)$$

$$F_i = \text{Max}_{j \in D^+(i)} [t_{ij} \circ F_j] \vee \text{Max}_{j \in D^-(i)} [t_{ij} \circ f_j], \quad i \neq N, \quad (2.5)$$

$$f_N = F_N = R(o), \quad (2.6)$$

where

$$\begin{aligned} D^+(i) &= \{j \in D(i) \mid t_{ij} \in A^+\}, \\ D^-(i) &= \{j \in D(i) \mid t_{ij} \in A^-\}. \end{aligned}$$

PROOF. It suffices to show only (2.4). Similarly, (2.5) is proved.

Let $i (\neq N)$ be given but arbitrary. Put

$$\min_{j \in D^+(i)} [t_{ij} \circ f_j] \wedge \min_{j \in D^-(i)} [t_{ij} \circ F_j] = g_i.$$

Moreover, suppose that there exists a path (i, j, k, \dots, n, N) satisfying that

$$g_i > t_{ij} \circ t_{jk} \circ \cdots \circ t_{nN}. \quad (2.7)$$

If $t_{ij} \in A^+$, then

$$t_{ij} \circ f_j \geq \min_{j \in D^+(i)} [t_{ij} \circ f_j] \geq g_i. \quad (2.8)$$

From this and (2.7), it follows that

$$t_{ij} \circ f_j > t_{ij} \circ t_{jk} \circ \cdots \circ t_{nN}.$$

From (H3), we have

$$f_j > t_{jk} \circ \cdots \circ t_{nN},$$

which contradicts the definition of f_j . Similarly, in case $t_{ij} \in A^-$, we obtain

$$F_j < t_{jk} \circ \cdots \circ t_{nN},$$

which contradicts the definition of F_j . Consequently, we see that

$$g_i \leq \min_p [t_{ij} \circ t_{jk} \circ \dots \circ t_{nN}] = f_i. \tag{2.9}$$

On the other hand, the reverse inequality to (2.9) follows from the definitions of g_i , f_i and F_i .

Solving two recursive equations (2.4), (2.5) with (2.6), simultaneously we find both shortest and longest path lengths from $i (\neq N)$ to N . In this sense we can also refer to the problem (ASP) solved by using (2.4), (2.5), (2.6) as the *associative shortest and longest path problem*.

From Theorem 2.1, we see that $\{f_i, F_i | i = 1, 2, \dots, N\}$ defined by (2.1), (2.2), (2.3) is one solution of the system of eqs. (2.4), (2.5) with (2.6). Hence the existence of the solution of the system has already been proved. Next we will show the uniqueness of the solution of the system.

Since the following lemma is proved in Maruyama (1996), we omit the proof.

LEMMA 2.2. Let $\{x_j\}_{j=1}^N, \{y_j\}_{j=1}^N, \{u_j\}_{j=1}^N, \{v_j\}_{j=1}^N \subset R$, and put $a = \wedge_{j=1}^N x_j, b = \wedge_{j=1}^N y_j, c = \vee_{j=1}^N u_j, d = \vee_{j=1}^N v_j$. Then there exists an index $j \in \{1, 2, \dots, N\}$ such that

$$|a - b| \vee |c - d| \leq |z_j - w_j|, \tag{2.10}$$

where

$$z_j - w_j = x_j - y_j \text{ or } u_j - v_j.$$

THEOREM 2.3 (UNIQUENESS). The system of eqs. (2.4), (2.5) with (2.6) possesses a unique solution.

PROOF. We suppose that $\{f_i, F_i | i = 1, 2, \dots, N\}, \{g_i, G_i | i = 1, 2, \dots, N\}$ are two solutions of the system of eqs. (2.4), (2.5) with (2.6). Let i be an arbitrary but fixed node of V and put

$$\begin{aligned} x_j &= \begin{cases} t_{ij} \circ f_j & \text{if } j \in D^+(i), \\ t_{ij} \circ F_j & \text{if } j \in D^-(i), \end{cases} & y_j &= \begin{cases} t_{ij} \circ g_j & \text{if } j \in D^+(i), \\ t_{ij} \circ G_j & \text{if } j \in D^-(i), \end{cases} \\ u_j &= \begin{cases} t_{ij} \circ F_j & \text{if } j \in D^+(i), \\ t_{ij} \circ f_j & \text{if } j \in D^-(i), \end{cases} & v_j &= \begin{cases} t_{ij} \circ G_j & \text{if } j \in D^+(i), \\ t_{ij} \circ g_j & \text{if } j \in D^-(i). \end{cases} \end{aligned}$$

Then from Lemma 2.2, it follows that there exists $j \in D(i)$ such that $(i, j) \in A$ and

$$|f_i - g_i| \vee |F_i - G_i| \leq |z_j - w_j|, \tag{2.11}$$

where $z_j - w_j = t_{ij} \circ f_j - t_{ij} \circ g_j$ or $t_{ij} \circ F_j - t_{ij} \circ G_j$.

Since

$$|z_j - w_j| \leq |t_{ij} \circ f_j - t_{ij} \circ g_j| \vee |t_{ij} \circ F_j - t_{ij} \circ G_j|,$$

it follows from (2.11) that

$$|f_i - g_i| \vee |F_i - G_i| \leq |t_{ij} \circ f_j - t_{ij} \circ g_j| \vee |t_{ij} \circ F_j - t_{ij} \circ G_j|. \tag{2.12}$$

Let $i_0 \in V$ be an arbitrary node. Then we consider the following algorithm:

Initialize Set $i = i_0 (\neq N)$. Go to step 1.

Step1. Find $j \in D(i)$ s.t.

$$|f_i - g_i| \vee |F_i - G_i| \leq |t_{ij} \circ f_j - t_{ij} \circ g_j| \vee |t_{ij} \circ F_j - t_{ij} \circ G_j|.$$

Go to step 2.

Step2. If $j = N$, then **stop**; we obtain $f_i = g_i, F_i = G_i$. Else go to step 3.

Step3. Set $i = j$. Go to step 1.

Since $G(V, A)$ contains no cycles, $j = N$ for a finite number of repetition. Hence we conclude that

$$f_{i_0} = g_{i_0}, \quad F_{i_0} = G_{i_0}.$$

Moreover

$$f_N = g_N = F_N = G_N = R(\circ).$$

Consequently, the two solutions are in fact identical.

Through the equations (2.4), (2.5) with (2.6), we can define the minimum decision function π and the maximum decision function σ as follows:

- $\pi(i)$ = the node $j \in V$ which attains the minimum of r.h.s. of (2.4),
- $\sigma(i)$ = the node $j \in V$ which attains the maximum of r.h.s. of (2.5).

Hence both optimal decision functions $\pi(\cdot)$ and $\sigma(\cdot)$ generate both a shortest path $(i, \hat{j}, \hat{k}, \hat{l}, \dots, \hat{m}, \hat{n}, N)$; $\hat{o} = N$ and a longest path $(i, j^*, k^*, l^*, \dots, m^*, n^*, N)$; $o^* = N$ as follows:

$$\begin{aligned} \hat{j} &= \pi(i), \quad \hat{k} = \begin{cases} \pi(\hat{j}), & \text{if } \hat{j} \in D^+(i), \\ \sigma(\hat{j}), & \text{if } \hat{j} \in D^-(i), \end{cases} \\ \hat{l} &= \begin{cases} \pi(\hat{k}), & \text{if } \hat{k} \in D^+(\hat{j}), \hat{k} = \pi(\hat{j}), \\ \sigma(\hat{k}), & \text{if } \hat{k} \in D^+(\hat{j}), \hat{k} = \sigma(\hat{j}), \\ \sigma(\hat{k}), & \text{if } \hat{k} \in D^-(\hat{j}), \hat{k} = \pi(\hat{j}), \dots \\ \pi(\hat{k}), & \text{if } \hat{k} \in D^-(\hat{j}), \hat{k} = \sigma(\hat{j}), \end{cases} \\ N &= \begin{cases} \pi(\hat{n}), & \text{if } \hat{n} \in D^+(\hat{m}), \hat{n} = \pi(\hat{m}), \\ \sigma(\hat{n}), & \text{if } \hat{n} \in D^+(\hat{m}), \hat{n} = \sigma(\hat{m}), \\ \sigma(\hat{n}), & \text{if } \hat{n} \in D^-(\hat{m}), \hat{n} = \pi(\hat{m}), \\ \pi(\hat{n}), & \text{if } \hat{n} \in D^-(\hat{m}), \hat{n} = \sigma(\hat{m}), \end{cases} \end{aligned}$$

and

$$\begin{aligned} j^* &= \sigma(i), \quad k^* = \begin{cases} \sigma(j^*), & \text{if } j^* \in D^+(i), \\ \pi(j^*), & \text{if } j^* \in D^-(i), \end{cases} \\ l^* &= \begin{cases} \sigma(k^*), & \text{if } k^* \in D^+(j^*), k^* = \sigma(j^*), \\ \pi(k^*), & \text{if } k^* \in D^+(j^*), k^* = \pi(j^*), \\ \pi(k^*), & \text{if } k^* \in D^-(j^*), k^* = \sigma(j^*), \dots \\ \sigma(k^*), & \text{if } k^* \in D^-(j^*), k^* = \pi(j^*), \end{cases} \end{aligned}$$

$$N = \begin{cases} \sigma(n^*), & \text{if } n^* \in D^+(m^*), n^* = \sigma(m^*), \\ \pi(n^*), & \text{if } n^* \in D^+(m^*), n^* = \pi(m^*), \\ \pi(n^*), & \text{if } n^* \in D^-(m^*), n^* = \sigma(m^*), \\ \sigma(n^*), & \text{if } n^* \in D^-(m^*), n^* = \pi(m^*). \end{cases}$$

Now we derive the system of two recursive equations for each type of problem.

EXAMPLE 2.4 (MAXIMUM PROBLEM). Let us consider the case $\circ = \vee$. Take $S = [a, b]$. Then, since $D^-(i) = \emptyset$, $D^+(i) = D(i)$, the system of eqs. (2.4), (2.5) with (2.6) is

$$f_i = \min_{j \in D(i)} [t_{ij} \vee f_j], \tag{2.13}$$

$$F_i = \text{Max}_{j \in D(i)} [t_{ij} \vee F_j], \quad i \neq N, \tag{2.14}$$

$$f_N = F_N = a = R(\circ). \tag{2.15}$$

In this case, two optimal value functions f_i , F_i are quite separate. Therefore, in this problem it suffices to solve the single recursive equation (2.13) or (2.14) with (2.15) as well as in the additive problem and in the multiplicative problem with nonnegative arc lengths.

The maximum problem has the following practical meaning: The traveller whose heart is weak wants to go to the destination. Unfortunately, he must walk up an uphill road. So, in order to lighten a burden on his heart, he will select a gentler slope rather than a shorter road. For example, the traveller wants to go from the starting point 1 to the terminal point N , where roads are uphill all the way. With each arc (i, j) , a real number $t_{ij} = \frac{\alpha_{ij}}{90}$ is associated, where α_{ij} denotes the angle of elevation of j from i . Then, the number t_{ij} is considered to be a degree of steepness of a slope. In this problem, it is absolutely appropriate that the length (measure) of a road p from 1 to N is defined by

$$t_{1i} \vee t_{ij} \vee \dots \vee t_{mN}, \quad p = (1, i, j, \dots, m, N)$$

and the traveller's objective is to find a path from 1 to N which has a minimum length.

REMARK. As can be seen from Example 2.4, if the associative operation \circ is monotone on S :

$$\langle R_0 \rangle \quad a, b, c \in S, \quad b < c \implies a \circ b \leq a \circ c,$$

then, the single recursive equation (1.4) holds for the problem (ASP) because $S = A^+$, $D^+(i) = D(i)$. However, the monotonicity condition $\langle R_0 \rangle$ is not a necessary and sufficient one for the problem (ASP) to admit the recursive equation; it is a sufficient condition. In fact, we have the following example. Let us consider the multiplicative shortest path problem ($a \circ b = ab$) on a network given in Fig. 1. Since there exist negative arcs ($t_{23} = -2$, $t_{24} = -1$), in this problem the operation $\circ = \times$ is not monotone. But we can verify that the single recursive equation (1.2) holds for the problem. In Maruyama

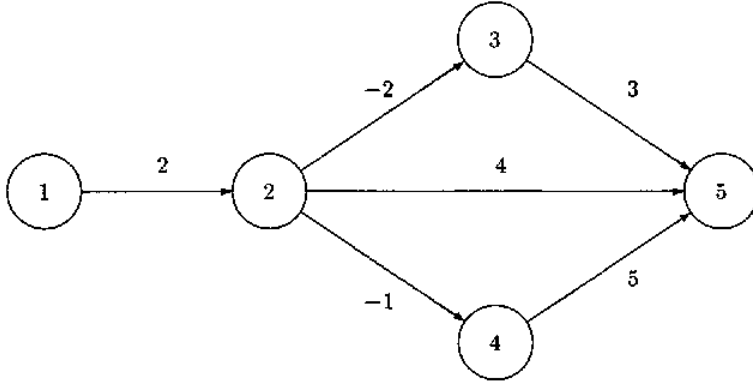


Figure 1: Multiplicative shortest path problem $a \circ b = ab$

(1997) it was shown that the problem (ASP) admits the single recursive equation (1.4) if and only if (ASP) satisfies the condition:

$$\langle R_1 \rangle \quad \min_{j \in D(i)} [t_{ij} \circ f_j] \leq \min\{t_{ij} \circ a \mid a \in A_j\} \quad \forall j \in D(i), \forall i \neq N,$$

where

$$A_j = \{t_{jk} \circ \dots \circ t_{mN} \mid (j, k, \dots, m, N) : \text{path}\}$$

for $j \neq N$ and $A_N = \{R(\circ)\}$. The multiplicative problem stated above is sure to satisfy the condition $\langle R_1 \rangle$. In fact, it is clear that

$$\min_{j \in D(i)} [t_{ij} \times f_j] = \min\{t_{ij} \times a \mid a \in A_j\} \text{ for all } j \in D(i), \quad i = 2, 3, 4.$$

Moreover, since $A_2 = \{-6, -5, 4\}$, $f_2 = -6$, it holds that

$$\min_{j \in D(1)} [t_{1j} \times f_j] = 2 \times (-6) = -12 = \min\{-12, -10, 8\} = \min\{t_{12} \times a \mid a \in A_2\}.$$

On the other hand, replacing only the arc length t_{12} with -2 in Fig.1, we see that the multiplicative problem does not satisfy $\langle R_1 \rangle$; hence the problem does not admit the single recursive equation. For this problem, the system of recursive equations

$$f_i = \min_{t_{ij} \geq 0} [t_{ij} \times f_j] \wedge \min_{t_{ij} < 0} [t_{ij} \times F_j], \tag{2.16}$$

$$F_i = \text{Max}_{t_{ij} \geq 0} [t_{ij} \times F_j] \vee \text{Max}_{t_{ij} < 0} [t_{ij} \times f_j], \quad i \neq N. \tag{2.17}$$

$$f_N = F_N = 1 = R(\times) \tag{2.18}$$

holds. In fact, the case $i \neq 1$ is clear. Further, we have

$$f_1 = -8 = t_{12} \times F_2, \quad F_1 = 12 = t_{12} \times f_2.$$

The above system, which was given in Maruyama (1996), is also derived from (2.4), (2.5) with (2.6) since $t_{ij} \in R = S$ and $A^+ = [0, +\infty)$, $A^- = (-\infty, 0)$.

It is noted that by using the system of two recursive equations (2.4), (2.5) with (2.6), we can solve not only the multiplicative problem with negative arcs but also other associative problems which do not satisfy $\langle R_1 \rangle$ as far as we are concerned with the problems satisfying the bitonicity condition (H3), (see Examples 2.5~2.7).

EXAMPLE 2.5 (MULTIPLICATIVE-ADDITIVE PROBLEM). Let us consider the problem in which $a \circ b = a + b - ab$ and $t_{ij} \in R = S$. For example, we consider the problem on a network given in Fig.2. Since

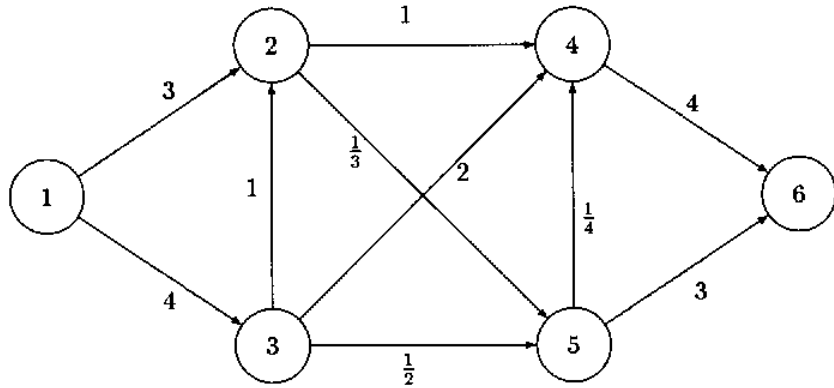


Figure 2: Multiplicative-additive problem $a \circ b = a + b - ab$

$$f_2 = (1 \circ 4) \wedge \left(\frac{1}{3} \circ \frac{1}{4} \circ 4\right) \wedge \left(\frac{1}{3} \circ 3\right) = 1 \wedge \frac{5}{2} \wedge \frac{7}{3} = 1,$$

$$f_3 = 1 \wedge (-2) \wedge \frac{17}{8} \wedge 2 = -2,$$

we have

$$\min_{j \in D(1)} [t_{1j} \circ f_j] = \min\{3 \circ 1, 4 \circ (-2)\} = 1 > -2 = \min\{1, -2, -\frac{5}{3}\} = \min\{t_{12} \circ a | a \in A_2\}.$$

Hence this problem does not satisfy the condition $\langle R_1 \rangle$, which implies that it does not admit the single recursive equation.

On the other hand, since the operation \circ satisfies the bitonicity for S ($A^+ = (-\infty, 1]$, $A^- = (1, +\infty]$) we can solve the problem by using the system of eqs.

$$f_i = \min_{t_{ij} \leq 1} [t_{ij} + f_j - t_{ij}f_j] \wedge \min_{t_{ij} > 1} [t_{ij} + F_j - t_{ij}F_j], \tag{2.19}$$

$$F_i = \text{Max}_{t_{ij} \leq 1} [t_{ij} + F_j - t_{ij}F_j] \vee \text{Max}_{t_{ij} > 1} [t_{ij} + f_j - t_{ij}f_j], \text{ for } i \neq N, \tag{2.20}$$

$$f_N = F_N = 0; \tag{2.21}$$

this system will be solved in Example 3.2.

EXAMPLE 2.6 (FRACTIONAL PROBLEM I). Let us consider the case $a \circ b = (a + b)/(1 + ab)$ and $S = [0, +\infty)$. For example, we consider the problem on a network given in Fig.3. In the same way as in Example 2.5, we have

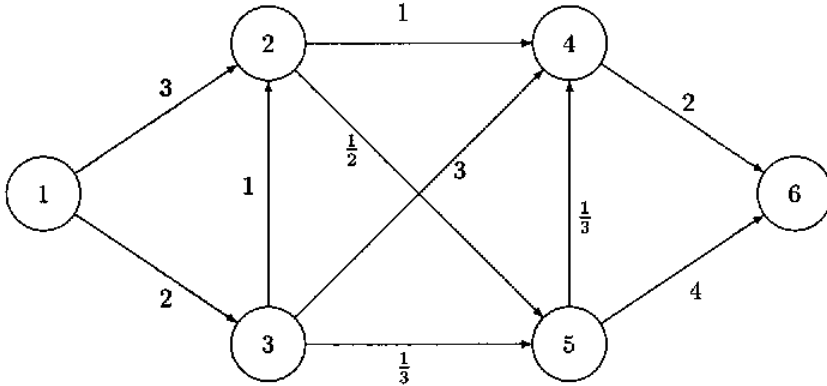


Figure 3: Fractional problem I $a \circ b = \frac{a+b}{1+ab}$

$$\min_{j \in D(1)} [t_{1j} \circ f_j] = 1 > \frac{9}{11} = \min\{t_{12} \circ a | a \in A_2\},$$

which implies that this problem does not admit the single recursive equation.

But, since the operation \circ satisfies the bitonicity for S ($A^+ = [0, 1)$, $A^- = [1, +\infty)$), we can solve the problem by using the system of eqs.

$$f_i = \min_{0 \leq t_{ij} < 1} \left[\frac{t_{ij} + f_j}{1 + t_{ij}f_j} \right] \wedge \min_{t_{ij} \geq 1} \left[\frac{t_{ij} + F_j}{1 + t_{ij}F_j} \right], \tag{2.22}$$

$$F_i = \text{Max}_{0 \leq t_{ij} < 1} \left[\frac{t_{ij} + F_j}{1 + t_{ij}F_j} \right] \vee \text{Max}_{t_{ij} \geq 1} \left[\frac{t_{ij} + f_j}{1 + t_{ij}f_j} \right], \text{ } i \neq N, \tag{2.23}$$

$$f_N = F_N = 0 \in [0, +\infty); \tag{2.24}$$

this system will be solved in Example 3.3.

EXAMPLE 2.7 (FRACTIONAL PROBLEM II). Let us consider the case $a \circ b = ab/(1 + (1 - a)(1 - b))$ and $S = (-\infty, 1]$. For example, we are concerned with the problem on a network given in Fig.4. Since we have

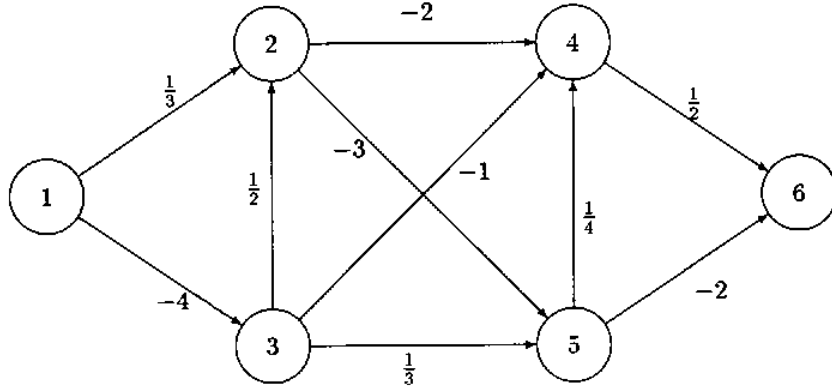


Figure 4: Fractional problem II $a \circ b = \frac{ab}{1+(1-a)(1-b)}$

$$\min_{j \in D(1)} [t_{1j} \circ f_j] = -\frac{2}{29} > -\frac{1}{7} = \min\{t_{13} \circ a \mid a \in A_3\},$$

this problem can not be also solved by the single recursive equation. However, since the operation \circ satisfies the bitonicity for S ($A^+ = (0, 1]$, $A^- = (-\infty, 0]$), the problem admits the system of eqs.

$$f_i = \min_{0 < t_{ij} \leq 1} \left[\frac{t_{ij} f_j}{1 + (1 - t_{ij})(1 - f_j)} \right] \wedge \min_{t_{ij} \leq 0} \left[\frac{t_{ij} F_j}{1 + (1 - t_{ij})(1 - F_j)} \right], \tag{2.25}$$

$$F_i = \text{Max}_{0 < t_{ij} \leq 1} \left[\frac{t_{ij} F_j}{1 + (1 - t_{ij})(1 - F_j)} \right] \vee \text{Max}_{t_{ij} \leq 0} \left[\frac{t_{ij} f_j}{1 + (1 - t_{ij})(1 - f_j)} \right], \quad i \neq N, \tag{2.26}$$

$$f_N = F_N = 1 \in (-\infty, 1]; \tag{2.27}$$

this system will be solved in Example 3.4.

REMARK. Each of the operations $a \circ b = a \times b$, $a \circ b = ab/(1 + (1 - a)(1 - b))$ is called triangular norm (t-norm), when $a, b \in [0, 1]$. On the other hand, each of the operations $a \circ b = a + b - ab$, $a \circ b = (a + b)/(1 + ab)$ is known as triangular conorm (t-conorm), when $a, b \in [0, 1]$ (see Butnariu and Klement (1993)).

3. Bidecision Algorithm

In this section we constitute an algorithm which solves the associative shortest and longest path problem (ASP). We generate a sequence $\{(f_i^{(k)}, F_i^{(k)}) | i \in V, k = 0, 1, 2, \dots\}$ which converges to the solution $\{(f_i, F_i) | i \in V\}$ of the system of eqs. (2.4), (2.5) with (2.6). The desired minimal and maximal path lengths form one solution of the system. Moreover, the uniqueness of the solution of the system was shown in Theorem 2.3. Therefore, using the sequence which converges to the solution, we can find the desired minimal and maximal path lengths.

Bidecision algorithm

Step 1. Initialization. Set

$$f_i^{(0)} = \min_{j \in D^+(i)} t_{ij}, \quad i = 1, 2, \dots, N-1, \quad f_N^{(0)} = R(o), \quad (3.1)$$

$$F_i^{(0)} = \text{Max}_{j \in D^+(i)} t_{ij}, \quad i = 1, 2, \dots, N-1, \quad F_N^{(0)} = R(o). \quad (3.2)$$

Step 2. Iteration. Set

$$f_i^{(k)} = \min_{j \in D^+(i)} [t_{ij} \circ f_j^{(k-1)}] \wedge \min_{j \in D^-(i)} [t_{ij} \circ F_j^{(k-1)}], \quad (3.3)$$

$$F_i^{(k)} = \text{Max}_{j \in D^+(i)} [t_{ij} \circ F_j^{(k-1)}] \vee \text{Max}_{j \in D^-(i)} [t_{ij} \circ f_j^{(k-1)}], \quad i \neq N, \quad (3.4)$$

$$f_N^{(k)} = F_N^{(k)} = R(o). \quad (3.5)$$

Step 3. Stopping rule. If

$$f_i^{(k)} = f_i^{(k-1)}, \quad F_i^{(k)} = F_i^{(k-1)}, \quad i = 1, 2, \dots, N,$$

then **stop**. Otherwise, set $k = k + 1$ and go to step 2.

THEOREM 3.1. *The bidecision algorithm terminates in at most $N - 1$ iterations. Then we obtain the solution of the system of eqs. (2.4), (2.5) with (2.6).*

PROOF. Let $i \neq N$ be a given but arbitrary node. Then from (3.3), (3.4) and Lemma 2.2, it follows that there exists $j \in D(i)$ satisfying that

$$\begin{aligned} |f_i^{(N-1)} - f_i^{(N-2)}| &\vee |F_i^{(N-1)} - F_i^{(N-2)}| \\ &\leq |t_{ij} \circ f_j^{(N-2)} - t_{ij} \circ f_j^{(N-3)}| \vee |t_{ij} \circ F_j^{(N-2)} - t_{ij} \circ F_j^{(N-3)}|. \end{aligned}$$

Hence, in the same way as in the proof of Theorem 2.3, we obtain

$$f_i^{(N-1)} = f_i^{(N-2)}, \quad F_i^{(N-1)} = F_i^{(N-2)}, \quad i = 1, 2, \dots, N-1, \quad (3.6)$$

Furthermore,

$$f_N^{(N-1)} = f_N^{(N-2)} = F_N^{(N-1)} = F_N^{(N-2)}. \quad (3.7)$$

Therefore, the bidecision algorithm terminates in at most $N - 1$ iterations. Moreover, from (3.3), (3.4), (3.6) and (3.7) we have

$$\begin{aligned} f_i^{(N-1)} &= \min_{j \in D^+(i)} [t_{ij} \circ f_j^{(N-2)}] \wedge \min_{j \in D^-(i)} [t_{ij} \circ F_j^{(N-2)}] \\ &= \min_{j \in D^+(i)} [t_{ij} \circ f_j^{(N-1)}] \wedge \min_{j \in D^-(i)} [t_{ij} \circ F_j^{(N-1)}], \\ F_i^{(N-1)} &= \text{Max}_{j \in D^+(i)} [t_{ij} \circ F_j^{(N-2)}] \vee \text{Max}_{j \in D^-(i)} [t_{ij} \circ f_j^{(N-2)}] \\ &= \text{Max}_{j \in D^+(i)} [t_{ij} \circ F_j^{(N-1)}] \vee \text{Max}_{j \in D^-(i)} [t_{ij} \circ f_j^{(N-1)}], \\ f_N^{(N-1)} &= F_N^{(N-1)} = R(\circ). \end{aligned}$$

Consequently, $\{(f_i^{(N-1)}, F_i^{(N-1)}) | i \in V\}$ is the solution of the system of eqs. (2.4), (2.5) with (2.6).

REMARK. Meanings of $f_i^{(k)}, F_i^{(k)}$ generated above are as follows:

$f_i^{(k)}$ = the length (1.3) of the shortest path from node i to reachable node or to N when $k + 1$ or fewer arcs are used, respectively.

$F_i^{(k)}$ = the length (1.3) of the longest path from node i to reachable node or to N when $k + 1$ or fewer arcs are used, respectively,

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

Since these can be proved in the same way in remark 3 of Maruyama (1996), we omit the demonstration.

Let

$$\begin{aligned} \pi^{(k)}(i) &= \text{the node } j \in V \text{ which attains the minimum} \\ &\quad \text{of r.h.s. of (3.3),} \end{aligned} \tag{3.8}$$

$$\begin{aligned} \sigma^{(k)}(i) &= \text{the node } j \in V \text{ which attains the maximum} \\ &\quad \text{of r.h.s. of (3.4).} \end{aligned} \tag{3.9}$$

Then in the same method as in Section 2 both optimal decision functions $\pi^{(k)}(\cdot)$ and $\sigma^{(k)}(\cdot)$ generate both the shortest and the longest path from node i to reachable node or N when $k + 1$ or fewer arcs used, respectively.

REMARK. In order to solve the problem (ASP) which admits the single recursive equation (1.4), we can use the following algorithm (Ford's Algorithm):

Step 1. Initialization. Set

$$f_i^{(0)} = \min_{j \in D(i)} t_{ij}, \quad i = 1, 2, \dots, N-1, \quad f_N^{(0)} = R(o). \quad (3.10)$$

Step 2. Iteration. Set

$$f_i^{(k)} = \min_{j \in D(i)} [t_{ij} \circ f_j^{(k-1)}], \quad i \neq N, \quad f_N^{(k)} = R(o). \quad (3.11)$$

Step 3. Stopping rule. If

$$f_i^{(k)} = f_i^{(k-1)}, \quad i = 1, 2, \dots, N,$$

then stop. Otherwise, set $k = k + 1$ and go to step 2.

REMARK. Let us compute the number of operations required by the bidecision algorithm for each type of problem.

The maximum problem can be solved by the Ford's algorithm. So, each iteration of the algorithm requires $(N-1)^2$ maximization (\vee) and $(N-1)(N-2)$ comparisons. Since $N-1$ iterations are required, for all iterations $2N^3 - 7N^2 + 8N - 3$ operations are required; the number of operations is the same as that in the additive problem.

In the multiplicative problem, each iteration of the bidecision algorithm requires $2(N-1)^2$ multiplications and $2(N-1)(N-2)$ comparisons. Since $N-1$ iterations are required, for all iterations $4N^3 - 14N^2 + 16N - 6$ operations are required; the number is two times that in the additive problem.

In the multiplicative-additive problem, each iteration of the bidecision algorithm requires $6(N-1)^2$ operations ($2(N-1)^2$ additions, $2(N-1)^2$ subtractions and $2(N-1)^2$ multiplications) and $2(N-1)(N-2)$ comparisons. Since $N-1$ iterations are required, for all iterations $8N^3 - 26N^2 + 28N - 10$ operations are required; the number is approximately four times that in the additive problem.

In the fractional problem I, each iteration of the bidecision algorithm requires $8(N-1)^2$ operations ($4(N-1)^2$ additions, $2(N-1)^2$ multiplications and $2(N-1)^2$ fractions) and $2(N-1)(N-2)$ comparisons. Since $N-1$ iterations are required, for all iterations $10N^3 - 32N^2 + 34N - 12$ operations are required; the number is approximately five times that in the additive problem.

In the fractional problem II, each iteration of the bidecision algorithm requires $12(N-1)^2$ operations ($2(N-1)^2$ additions, $4(N-1)^2$ multiplications and $4(N-1)^2$ subtractions and $2(N-1)^2$ fractions) and $2(N-1)(N-2)$ comparisons. Since $N-1$ iterations are required, for all iterations $14N^3 - 44N^2 + 46N - 16$ operations are required; the number is approximately seven times that in the additive problem.

Consequently, in all problems above, we obtain the running time of $O(N^3)$ for all iterations.

EXAMPLE 3.2. We reconsider the multiplicative-additive problem ($a \circ b = a + b - ab$) on a network given in Fig.2. By applying the bidecision algorithm, the sequence $\{(f_i^{(k)}, F_i^{(k)})\}$, $k = 0, 1, 2, \dots$ can be computed successively as shown in Table 1. From this table, we can see that the shortest path length and the longest one from 1 to 6 are $-\frac{19}{8}$ and 10, respectively. The pairs of the node $\pi^{(k)}(i)$ and the node $\sigma^{(k)}(i)$ which are defined by (3.8) and (3.9), respectively are given in Table 2. Using the optimal decision functions $\pi^{(k)}(\cdot)$, $\sigma^{(k)}(\cdot)$, we can find the shortest path (1, 3, 5, 4, 6) and the longest path (1, 3, 4, 6), simultaneously. We remark that since this problem does not admit the single recursive equation, it can not be solved by the Ford's algorithm.

Table 1: Sequence in the multiplicative-additive problem ($a \circ b = a + b - ab$)

Node	$(f_i^{(0)}, F_i^{(0)})$	$(f_i^{(1)}, F_i^{(1)})$	$(f_i^{(2)}, F_i^{(2)})$	$(f_i^{(3)}, F_i^{(3)})$	$(f_i^{(4)}, F_i^{(4)}) = (f_i, F_i)$
1	(3, 4)	$(-2, \frac{5}{2})$	$(-11, 2)$	$(-\frac{19}{8}, 10)$	$(-\frac{19}{8}, 10)$
2	$(\frac{1}{3}, 1)$	$(\frac{1}{2}, \frac{7}{3})$	$(1, \frac{5}{2})$	$(1, \frac{5}{2})$	$(1, \frac{5}{2})$
3	$(\frac{1}{2}, 2)$	(4, 5)	$(-2, \frac{17}{8})$	$(-2, \frac{17}{8})$	$(-2, \frac{17}{8})$
4	(4, 4)	(4, 4)	(4, 4)	(4, 4)	(4, 4)
5	$(\frac{1}{4}, 3)$	$(3, \frac{13}{4})$	$(3, \frac{13}{4})$	$(3, \frac{13}{4})$	$(3, \frac{13}{4})$
6	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)

Table 2: Sequence of optimal decision function in the multiplicative-additive problem

Node	$(\pi^{(1)}(i), \sigma^{(1)}(i))$	$(\pi^{(2)}(i), \sigma^{(2)}(i))$	$(\pi^{(3)}(i), \sigma^{(3)}(i))$	$(\pi^{(4)}(i), \sigma^{(4)}(i))$
1	(3, 3)	(3, 2)	(3, 3)	(3, 3)
2	(5, 5)	(4, 5)	(4, 5)	(4, 5)
3	(4, 5)	(4, 5)	(4, 5)	(4, 5)
4	(6, 6)	(6, 6)	(6, 6)	(6, 6)
5	(6, 4)	(6, 4)	(6, 4)	(6, 4)

EXAMPLE 3.3. Let us reconsider the fractional problem I ($a \circ b = (a + b)/(1 + ab)$) on a network given in Fig.3. Since this problem does not admit the single recursive equation, we must use the bidecision algorithm. The sequence $\{(f_i^{(k)}, F_i^{(k)})\}$ can be computed successively as shown in Table 3. From Table 4, we can find the shortest paths (1, 3, 5, 6), (1, 2, 5, 6) and the longest path (1, 3, 4, 6) whose lengths are $\frac{9}{11}$ and $\frac{19}{17}$, respectively.

Table 3: Sequence in the fractional problem I ($a \circ b = (a + b)/(1 + ab)$)

Node	$(f_i^{(0)}, F_i^{(0)})$	$(f_i^{(1)}, F_i^{(1)})$	$(f_i^{(2)}, F_i^{(2)})$	$(f_i^{(3)}, F_i^{(3)})$	$(f_i^{(4)}, F_i^{(4)}) = (f_i, F_i)$
1	(2, 3)	$(\frac{5}{7}, \frac{7}{5})$	$(\frac{9}{11}, \frac{13}{11})$	$(\frac{9}{11}, \frac{19}{17})$	$(\frac{9}{11}, \frac{19}{17})$
2	$(\frac{1}{2}, 1)$	$(\frac{5}{7}, \frac{3}{2})$	$(1, \frac{3}{2})$	$(1, \frac{13}{8})$	$(1, \frac{13}{8})$
3	$(\frac{1}{3}, 3)$	$(\frac{3}{5}, \frac{13}{7})$	$(\frac{5}{7}, \frac{13}{7})$	$(\frac{5}{7}, \frac{13}{7})$	$(\frac{5}{7}, \frac{13}{7})$
4	(2, 2)	(2, 2)	(2, 2)	(2, 2)	(2, 2)
5	$(\frac{1}{3}, 4)$	$(\frac{7}{5}, 4)$	$(\frac{7}{5}, 4)$	$(\frac{7}{5}, 4)$	$(\frac{7}{5}, 4)$
6	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)

Table 4: Sequence of optimal decision function in the fractional problem I

Node	$(\pi^{(1)}(i), \sigma^{(1)}(i))$	$(\pi^{(2)}(i), \sigma^{(2)}(i))$	$(\pi^{(3)}(i), \sigma^{(3)}(i))$	$(\pi^{(4)}(i), \sigma^{(4)}(i))$
1	(3, 2), (3, 3)	(2, 2), (2, 3), (3, 2), (3, 3)	(2, 3), (3, 3)	(2, 3), (3, 3)
2	(5, 5)	(4, 5)	(4, 5)	(4, 5)
3	(3, 5)	(4, 5)	(4, 5)	(4, 5)
4	(6, 6)	(6, 6)	(6, 6)	(6, 6)
5	(4, 6)	(4, 6)	(4, 6)	(4, 6)

EXAMPLE 3.4. We review the fractional problem II ($a \circ b = ab/(1 + (1 - a)(1 - b))$) on a network given in Fig.4. The single recursive equation does not hold for this problem; hence we can not use the Ford's algorithm. So, by applying the bidecision algorithm, we can compute the sequence $\{(f_i^{(k)}, F_i^{(k)})\}$ successively as shown in Table 5. From Table 6, we can find both the shortest path (1, 3, 2, 5, 6) and the longest path (1, 3, 4, 6) whose lengths are $-\frac{1}{7}$ and $\frac{4}{29}$, respectively.

Table 5: Sequence in the fractional problem II ($a \circ b = ab/(1 + (1 - a)(1 - b))$)

Node	$(f_i^{(0)}, F_i^{(0)})$	$(f_i^{(1)}, F_i^{(1)})$	$(f_i^{(2)}, F_i^{(2)})$	$(f_i^{(3)}, F_i^{(3)})$	$(f_i^{(4)}, F_i^{(4)}) = (f_i, F_i)$
1	$(-4, \frac{1}{3})$	$(-\frac{4}{7}, \frac{4}{11})$	$(-\frac{1}{4}, \frac{1}{8})$	$(-\frac{1}{7}, \frac{4}{29})$	$(-\frac{1}{7}, \frac{4}{29})$
2	(-3, -2)	$(-\frac{2}{5}, \frac{6}{13})$	$(-\frac{2}{5}, \frac{6}{13})$	$(-\frac{2}{5}, \frac{6}{13})$	$(-\frac{2}{5}, \frac{6}{13})$
3	$(-1, \frac{1}{2})$	$(-\frac{1}{4}, \frac{1}{18})$	$(-\frac{1}{4}, \frac{2}{11})$	$(-\frac{1}{4}, \frac{2}{11})$	$(-\frac{1}{4}, \frac{2}{11})$
4	$(\frac{1}{2}, \frac{1}{2})$	$(\frac{1}{2}, \frac{1}{2})$	$(\frac{1}{2}, \frac{1}{2})$	$(\frac{1}{2}, \frac{1}{2})$	$(\frac{1}{2}, \frac{1}{2})$
5	$(-2, \frac{1}{4})$	$(-2, \frac{1}{11})$	$(-2, \frac{1}{11})$	$(-2, \frac{1}{11})$	$(-2, \frac{1}{11})$
6	(1, 1)	(1, 1)	(1, 1)	(1, 1)	(1, 1)

Table 6: Sequence of optimal decision function in the fractional problem II

Node	$(\pi^{(1)}(i), \sigma^{(1)}(i))$	$(\pi^{(2)}(i), \sigma^{(2)}(i))$	$(\pi^{(3)}(i), \sigma^{(3)}(i))$	$(\pi^{(4)}(i), \sigma^{(4)}(i))$
1	(3, 3)	(2, 3)	(3, 3)	(3, 3)
2	(4, 5)	(4, 5)	(4, 5)	(4, 5)
3	(4, 5)	(4, 2)	(4, 2)	(4, 2)
4	(6, 6)	(6, 6)	(6, 6)	(6, 6)
5	(6, 4)	(6, 4)	(6, 4)	(6, 4)

Acknowledgement

The author would like to thank the referee for his several suggestions, which improve the first manuscript.

References

- Bellman, R., Esogbue, A.O. and Nabeshima, I. (1982), *Mathematical Aspects of Scheduling and Applications*, Pergamon Press, New York.
- Butnariu, D. and Klement, E.P. (1993), *Triangular norm-based measures and games with fuzzy coalitions*, Kluwer Academic Publishers.
- Dreyfus, S.E. (1969), An appraisal of some shortest-path algorithms, *Oper. Res.*, **17**, 395-412.
- Frieze, A. (1977), Minimum paths in directed graphs, *Oper. Res. Quart.*, **28**, 339-346.
- Iwamoto, S. (1987), *Theory of Dynamic Programs* (in Japanese), Kyushu University Press.
- Iwamoto, S. (1993), From dynamic programming to bynamic programming, *J. Math. Anal. Appl.*, **177**, 56-74.
- Maruyama, Y. (1996), Shortest and longest path problems, *Optimization*, **38**, 287-299.
- Maruyama, Y. (1997), On associative shortest path problems, *Bull. Inform. Cybernet.*, **29**, 67-81.
- Smith, D.K. (1991) *Dynamic Programming*, Ellis Horwood Limited.
- Sniedovich, M. (1992) *Dynamic Programming*, Marcel Dekker.

Received November 25, 1998

Revised March 1, 1999