ALGORITHMS FOR BIOBJECTIVE SHORTEST PATH PROBLEMS IN FUZZY NETWORKS

I. MAHDAVI, N. MAHDAVI-AMIRI AND S. NEJATI

Abstract. We consider biobjective shortest path problems in networks with fuzzy arc lengths. Considering the available studies for single objective shortest path problems in fuzzy networks, using a distance function for comparison of fuzzy numbers, we propose three approaches for solving the biobjective problems. The first and second approaches are extensions of the labeling method to solve the single objective problem and the third approach is based on dynamic programming. The labeling methods usually producing several nondominated paths, we propose a fuzzy number ranking method to determine a fuzzy shortest path. Illustrative examples are worked out to show the effectiveness of our algorithms.

1. **Introduction**

The problem of finding a shortest path from a specified source node to a specified destination node is a classical and fundamental problem appearing in various applications. A solution generates essential information in transportation, scheduling routing, communication and computer network applications. For these problems, the aim is to find a path with a minimal distance from a source node to a destination node in a network. Several efficient algorithms have been developed by Bellman [3], Dijkstra [12] and Dreyfus [13], often referred to as the standard shortest path algorithms. In spite of efficiency of these algorithms, they may not be useful when instead of minimizing distance, optimization of time or cost is of interest. Time or cost may fluctuate depending on traffic conditions, payloads and so on, and thus be often uncertain. In this case, it is proper to consider fuzzy numbers as the value of arcs and consider the fuzzy shortest path in a fuzzy network. The fuzzy shortest path problem was first analyzed by Dubis and Prade [27] in 1980. Since then, significant research work has been carried out on the problem [16, 28, 14, 7, 20, 43, 11, 44, 22, 29, 21]. As examples, Klein [20] considered finding a path or paths corresponding to the threshold of membership degree set by a decision maker, Yager [43] studied a path problem in terms of possibilistic production systems, Lin and Chern [20] derived the membership function of the shortest length and proposed an algorithm for finding the most vital arc in a network, and Okada and Soper [28] introduced the concept of nondominated paths and gave an algorithm based on multiple labeling method for a multicriteria shortest path.

Received: August 2010; Revised: November 2010; Accepted: December 2010
Key words and phrases: Biobjective shortest path, Fuzzy network, Labeling method, Dynamic programming, Fuzzy ranking methods.

However, it is often not sufficient to be restricted to one objective. Applications often prompt the necessity for taking two or more objectives into account, resulting in biobjective or multiple objective shortest path problems. We consider the biobjective shortest path (BSP) problem as a natural extension of the single-objective case. BSP problem belongs to the class of multiple objective combinatorial optimization (MOCO) problems [17, 34]. In a BSP problem, the aim is to find efficient solutions. There are two main approaches to solve BSP problems: the first one is enumerative in nature such as label correcting [35, 5] and label setting [38, 39, 25] or ranking [27] methods, and the second one is a two phase method [36, 19].

To the best of our knowledge, not much study exists on biobjective shortest path problems in fuzzy networks. Here, we first extend the labeling approach of Okada et al. [28] proposed for the single objective fuzzy shortest path problem to the biobjective case. In doing this, using the labeling approach we present two algorithms by proposing a ranking method to rank the efficient paths. We also present a third algorithm by an extension of a dynamic programming approach for the single objective case. The remainder of our work is organized as follows. In the next section, we review basic definitions on fuzzy numbers, followed by a discussion on ranking fuzzy numbers in Section 3. There, we also propose an algorithm for ranking fuzzy vectors. In Section 4, we explain the BSP problems briefly. Section 5 gives the labeling algorithms. In Section 6, a dynamic programming approach is presented. In Section 7, we work out an example to give a comparison of the results obtained by the proposed algorithms. Finally, we conclude in Section 8.

2. **Fuzzy Numbers**

Fuzzy set theory and its attendant fuzzy logic were developed by Lotfi Zadeh in 1965 to handle semantic and subjective ambiguity [45]. Here, we give some basic definitions and results pertinent to our work following the material in [15].

Definition 2.1. Let X be a set of elements x. A fuzzy set A is a collection of ordered pairs $(x, \mu_A(x))$, for $x \in X$. X is called the universe of discourse and $\mu_A(x) : X \to [0, 1]$ is the membership function.

Definition 2.2. The support of a fuzzy set A is defined to be $Supp(A) = \{x \in X | \mu_A(x) > 0.\}$.

Definition 2.3. A fuzzy set A is said to be convex if the membership function is quasiconcave; that is, $\forall x_1, x_2 \in X$ and $\lambda \in [0, 1]$, the relationship $\mu_A[\lambda x_1 + (1-\lambda)x_2] \geq \min[\mu_A(x_1), \mu_A(x_2)]$ holds.

Definition 2.4. The height of a fuzzy set A is defined to be $h(A) = \max_{x \in X} \mu_A(x)$.

Definition 2.5. If *h(A)*=1, then A is called normal; otherwise, it is called subnormal.

Definition 2.6. A fuzzy number A is a fuzzy set of real numbers $\mathcal{R}$ for which the followings are true: A is normal ($\exists x$: $\mu_A(x) = 1$), convex, upper semi-continuous, and has a bounded support.

Definition 2.7. A flat fuzzy number $\tilde{a}$ is a fuzzy number with the property that there exists $\underline{m}, \overline{m} \in \mathcal{R}$, with $\underline{m} < \overline{m}$, such that $\mu_{\tilde{a}}(x) = 1,\ \forall x \in [\underline{m}, \overline{m}]$. An LR type flat fuzzy number is denoted by $(\underline{m}, \overline{m},\ \alpha,\ \beta)_{LR}$, where $\alpha,\ \beta$ are called the left-hand and right-hand spreads, and defined as follows:

$$\mu_{\tilde{a}}(x) = \begin{cases} L\left[(\underline{m} - x)/\alpha\right] & \text{if } x < \underline{m},\ \alpha \in \mathcal{R}^+, \\ 1 & \text{if } x \in [\underline{m}, \overline{m}], \\ R\left[(x - \overline{m})/\beta\right] & \text{if } x > \overline{m},\ \beta \in \mathcal{R}^+, \end{cases} \tag{1}$$

where L and R are reference functions such that $L(x) = L(-x), R(x) = R(-x), L(0) = R(0) = 1, L(1) = R(1) = 0$, and L and R are strictly decreasing on $[0,\ +\infty)$.

Definition 2.8. A fuzzy number $\tilde{a}$ is said to be positive if its membership function is such that $\mu_{\tilde{a}}(x) = 0,\ \forall x < 0$.

Definition 2.9. Let $\tilde{a}$ and $\tilde{b}$ be LR fuzzy numbers. An addition of fuzzy numbers $\tilde{a}$ and $\tilde{b}$, $\tilde{c} = \tilde{a} \bigoplus \tilde{b}$, is defined by the membership function, $\mu_{\tilde{c}}(t) = \sup_{t=u+v}$ $min\{\mu_{\tilde{a}}(x), \mu_{\tilde{b}}(x)\}$.

Note that we can simply represent the above addition of $\tilde{a}$ and $\tilde{b}$ as follows:

$$(\underline{a}, \overline{a},\ \alpha,\ \beta)_{LR} \bigoplus (\underline{b}, \overline{b},\ \gamma,\ \delta)_{LR} = (\underline{a+b}, \overline{a+b},\ \alpha+\gamma,\ \beta+\delta)_{LR}. \tag{2}$$

As special cases of LR fuzzy numbers, triangular and trapezoidal fuzzy numbers are often used. A trapezoidal membership function is a piecewise linear and continuous function characterized by four parameters $a,\ b,\ c$ and d:

$$\mu_{trapezoid}(x; a,\ b,\ c,\ d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & d \leq x. \end{cases} \tag{3}$$

The parameters $a \leq b \leq c \leq d$ define four breakpoints, here designated as left footpoint (a), left shoulderpoint (b), right shoulderpoint (c), and right footpoint (d).

A triangular membership function is a special case of the trapezoidal membership function obtained by merging the two shoulderpoints into one; that is, by setting b=c.

3. Ranking Fuzzy Numbers

The order relation "$\tilde{a} \leq \tilde{b}$" between fuzzy numbers, being essential for "fuzzy max" and "fuzzy min", renders some definitions that are very difficult to implement as computer programs [28]. But, if we restrict ourselves to a class of fuzzy numbers, we can simplify the definition of order relation "$\leq$" by using the assertion presented by Ramik and Rimanek [31] as follows.

Definition 3.1. Let $\tilde{a} = (\underline{a}, \overline{a},\ \alpha,\ \beta)_{LR}$ and $\tilde{b} = \left(\underline{b}, \overline{b},\ \gamma,\ \delta\right)_{LR}$ be LR fuzzy numbers. We say $\tilde{a} \leq \tilde{b}$ if the following four inequalities hold:

$$\underline{a} \leq \underline{b},\ \overline{a} \leq \overline{b},\ \underline{a} - \alpha \leq \underline{b} - \gamma,\ \overline{a} + \beta \leq \overline{b} + \delta. \tag{4}$$

In spite of the simplicity of this definition, it is not so useful since one can rarely expect that all the four inequalities hold. So, we should consider other ways of ordering fuzzy quantities.

The problem of ordering fuzzy quantities has been addressed by several researchers. Examples include [1, 2, 4, 6, 8, 9, 10, 23, 32, 40, 41, 42] In [41, 42], thorough reviews are found. However, there is no universal approach for ordering fuzzy quantities.

Unlike real numbers, fuzzy quantities have no natural order. A straightforward idea for ordering fuzzy quantities is to convert them into real numbers and base the comparison on the obtained crisp values. Any conversion scheme, however, pays attention to a special aspect of a fuzzy quantity. As a consequence, a given approach may not be effective if it merely associates a real number with each fuzzy quantity.

Here, we utilize an ordering approach based on a distance function proposed by Sadeghpour-Gildeh and Gien [33]. This method also converts fuzzy numbers to real numbers, but it has an advantage that the obtained distances can effectively be used for finding minimum of the numbers.

Definition 3.2. The α-cut, $\tilde{A}_\alpha$, of a fuzzy set $\tilde{A}$ in the universe of discourse X is defined to be:

$$\tilde{A}_\alpha = \{ x |\, \mu_{\tilde{A}}(x) \geq \alpha,\ x \in X \}, \qquad \text{where} \ \ \alpha \in [0,1]. \tag{5}$$

The lower and upper points of any α-cut, $\tilde{A}_\alpha$, are represented by $\inf\tilde{A}_\alpha$ and $\sup\tilde{A}_\alpha$, respectively, and we suppose that both are finite. For convenience, we denote $\inf\tilde{A}_\alpha$ by A_α^- and $\sup\tilde{A}_\alpha$ by A_α^+.

Definition 3.3. Let $\tilde{a}$ and $\tilde{b}$ be two fuzzy numbers. The distance between $\tilde{a}$ and $\tilde{b}$ is defined to be:

$$D_{p,q}\left(\tilde{a},\tilde{b}\right) = \begin{cases} \left[(1-q)\int_0^1 \left|a_\alpha^- - b_\alpha^-\right|^p d\alpha + q\int_0^1 \left|a_\alpha^+ - b_\alpha^+\right|^p d\alpha\right]^{\frac{1}{p}}, & p < \infty \\ (1-q)\sup_{0<\alpha\leq 1}\left(\left|a_\alpha^- - b_\alpha^-\right|\right) + q\inf_{0<\alpha\leq 1}\left(\left|a_\alpha^+ - b_\alpha^+\right|\right), & p = \infty. \end{cases} \tag{6}$$

For two triangular fuzzy numbers $\tilde{a} = (a_1, a_2, a_3)$ and $\tilde{b} = (b_1, b_2, b_3)$, assuming $p = 2$ and $q = \frac{1}{2}$, we have:

$$D_{2,\frac{1}{2}}\left(\tilde{a},\tilde{b}\right) = \sqrt{\frac{1}{6}\left[\sum_{i=1}^{3}(b_i - a_i)^2 + (b_2 - a_2)^2 + \sum_{i\in\{1,2\}}(b_i - a_i)(b_{i+1} - a_{i+1})\right]}, \tag{7}$$

and for two trapezoidal fuzzy numbers $\tilde{a} = (a_1, a_2, a_3, a_4)$ and $\tilde{b} = (b_1, b_2, b_3, b_4)$, we have:

$$D_{2,\frac{1}{2}}\left(\tilde{a}, \tilde{b}\right) = \sqrt{\frac{1}{6}\left[\sum_{i=1}^{4}(b_i - a_i)^2 + \sum_{i\in\{1,3\}}(b_i - a_i)(b_{i+1} - a_{i+1})\right]}. \tag{8}$$

Using this distance function, Mahdavi et al. [24] presented an approach for comparison of two trapezoidal fuzzy numbers and proposed the following algorithm for computing the smaller number.

Algorithm 3.4. Comparison minimum of two trapezoidal fuzzy numbers.
Input: Two trapezoidal fuzzy numbers $\left(\tilde{a}, \tilde{b}\right)$.
Output: Minimum of two trapezoidal fuzzy numbers.

(1) Calculate the minimum value (MV) as:

$$\text{MV} = \text{Min}\left(\tilde{a}, \tilde{b}\right) = \left(\min(a_1, b_1), \min(a_2, b_2), \min(a_3, b_3), \min(a_4, b_4)\right).$$

(2) Compute the distance of each number from MV using the distance formula (8).
(3) Designate the smaller fuzzy number to be the minimum.

Now, we use this approach to present an algorithm for comparison of a set of fuzzy vectors. We first give the following definition.

Definition 3.5. Suppose that $\tilde{X} = \{\tilde{x}_1,\ \tilde{x}_2\ ,\ \ldots,\ \tilde{x}_n\}$ is a set of fuzzy vectors such that $\tilde{x}_j = (\tilde{x}_{1j}, \tilde{x}_{2j}, \ldots \tilde{x}_{mj})$. So, we have,

$$\tilde{X} = \{(\tilde{x}_{11}, \tilde{x}_{21}, \ldots \tilde{x}_{m1}),\ (\tilde{x}_{12}, \tilde{x}_{22}, \ldots \tilde{x}_{m2})\ ,\ \ldots,\ (\tilde{x}_{1n}, \tilde{x}_{2n}, \ldots \tilde{x}_{mn})\},$$

with

$$\tilde{x}_{ij} = (a_{ij}, b_{ij}, c_{ij}, d_{ij}),\ i = 1, 2, \ldots, m,\quad j = 1, 2, \ldots, n,$$

being trapezoidal fuzzy numbers.

Here, using Definition 3.5 and Algorithm 3.4, we propose the following algorithm for ranking a set of fuzzy vectors. This algorithm will be used in the labeling algorithms (algorithms 5.5 and 5.7 to be seen later) for ranking nondominated paths based on the first, second or both objective values which are fuzzy vectors.

Algorithm 3.6. Ranking a set of fuzzy vectors by distance method.

(1) Suppose that we have a set $\tilde{X}$ of fuzzy vectors as specified by Definition 3.5. For every i, compute a corresponding "minimal" fuzzy number and call it $p_i min$ as follows:

$$p_i min = \left(\min_{j=1:n}(a_{ij})\ , \min_{j=1:n}(b_{ij})\ , \min_{j=1:n}(c_{ij})\ , \min_{j=1:n}(d_{ij})\ \right),\ i = 1, 2, \ldots, m.$$

(2) For $i = 1, 2, \dots, m,\ j = 1, 2, \dots, n$, compute d_{ij}, the distance between $\tilde{x}_{ij}$ and $p_i min$, using the distance formula (8) and form the d set as follows:

$$d = \{(d_{11}, d_{21}, \dots, d_{m1}),\ (d_{12}, d_{22}, \dots, d_{m2}),\ \dots,\ (d_{1n}, d_{2n}, \dots, d_{mn})\}.$$

For every i, compute the minimum value as follows:

$$d_i^* = \min_{j=1:n} (d_{ij})\ ,\ i = 1, 2, \dots, m.$$

(3) For $i = 1, 2, \dots, m,\ j = 1, 2, \dots, n$, divide d_{ij} into d_i^* and form the set S in turn corresponding to $\tilde{X}$ as below:

$$s_{ij} = d_{ij}/d_i^*,$$
$$S = \{(s_{11}, s_{21}, \dots, s_{m1}),\ (s_{12}, s_{22}, \dots, s_{m2})\ ,\ \dots,\ (s_{1n}, s_{2n}, \dots, s_{mn})\}\ .$$

{Notice that the s_{ij} have no units now and they can be added together.}

(4) For every element j of S, compute r_j as the rank and form the set of ranks R as follows:

$$r_j = s_{1j} + s_{2j} + \cdots + s_{mj},\ 1 \le j \le n,$$
$$R = \{r_1, r_2, \dots, r_n\}.$$

(5) Sort the set R in ascending order. Note that if $r_i < r_j$ then decide that $\tilde{x}_i < \tilde{x}_j$. So, by using the set R, we obtain a ranking of the set $\tilde{X}$.

4. **BSP Problems**

Here, the terminology and basic elements of BSP problems are described following the notation of Przybylski et al. [30].

Definition 4.1. Let $G=(N,\ A)$ be a directed network with a set of nodes N $=\{1,\dots,n\}$ and a set of arcs $A = \{(i_1, j_1), \dots, (i_m, j_m)\} \subseteq N \times N$. Two positive costs $c_{ij} = \left(c_{ij}^1, c_{ij}^2\right) \in N \times N$ are associated with each arc $(i, j) \in A$. In a road network, for example, the costs c_{ij}^1 and c_{ij}^2 could represent the time and distance for traversing arc $(i,\ j)$, respectively. A path in G from node $i_0 \in N$ to node $i_l \in N$ is a sequence of arcs $(i_0, i_1), (i_1, i_2), \dots, (i_{l-1}, i_l)$ in A. The biobjective shortest path (BSP) problem with source node $s \in N$ and target node $t \in N$ can be formulated as the following network flow problem,

$$\min z(x) = \begin{cases} z_1(x) = \sum_{(i,j)\in A} c_{ij}^1 x_{ij}\ , \\ z_2(x) = \sum_{(i,j)\in A} c_{ij}^2 x_{ij}\ , \end{cases} \tag{9}$$

$$\text{s.t.} \qquad \sum_{(i,j)\in A} x_{ij} \ - - \sum_{(j,i)\in A} x_{ji} \ = \begin{cases} 1 & \text{if } i = s \\ 0 & \text{if } i \neq s \\ -1 & \text{if } i = t \end{cases} \tag{10}$$

$$x_{ij} \in \{0, 1\}, \qquad \text{for all } (i, j) \in A, \tag{11}$$

where, x is a vector of flows on the arcs. Two objectives z_1 and z_2 are specified in (9), and the constraints (10) represent flow balance at different nodes. Value of 1, -1 or 0 respectively indicates that there exists a supply of one unit of flow, a demand of one unit of flow, or neither of the two. The model ensures that one unit of flow is transported through the network from s to t through the arcs with flow value 1 in a path from s to t. The feasible set X is described by constraints (10) and (11), and its image under the objective function is *Z := z(X)*.

We are seeking those feasible solutions x which do not allow improvement of one component of the objective vector *z(x)* without deteriorating the other one.

Definition 4.2. A feasible solution $\hat{x} \in X$ is called efficient if there exist no element $\acute{x} \in X$ with $(z_1(\acute{x}), z_2(\acute{x})) \leq (z_1(\hat{x}), z_2(\hat{x}))$; the image of $\hat{x}$, $z(\hat{x}) = (z_1(\hat{x}), z_2(\hat{x}))$, is called non-dominated.

Let X_E denote the set of all efficient solutions and let Z_N denote the set of all non-dominated points.

Definition 4.3. Two feasible solutions x and $\acute{x}$ are called equivalent if $z(x) = z(\acute{x})$.

Definition 4.4. A complete set X_E is a set of efficient solutions such that all $x \in X \backslash X_E$ are either dominated or equivalent to at least one $x \in X_E$.

Another notion of optimality that is used in the context of biobjective optimization is lexicographic minimization. Here, we choose among all optimal feasible solutions for the preferred *component* k of the objective vector one that is optimal for the other *component* l.

Definition 4.5. Let $k \in \{1,2\}$ and $l \in \{1,2\} \setminus \{k\}$. Then, $z(\hat{x}) \leq_{lex(k,l)} z(\acute{x})$ if either $z_k(\hat{x}) < z_k(\acute{x})$ or both $z_k(\hat{x}) = z_k(\acute{x})$ and $z_l(\hat{x}) \leq z_l(\acute{x})$. We call $\hat{x}$ a *lex(k,l)*-best solution if $z(\hat{x}) \leq_{lex(k,l)} z(x)$ for all $x \in X$. Let $x_{lex(k,l)}$ denote a *lex(k,l)*-best solution.

When solving a single-objective version of the BSP problem (parametric approach) with the network simplex algorithm (e.g., see [18]), the formulation (9)–(11) is not favorable. Difficulties arise as the network simplex method performs many basis changes without an actual flow change, because the flow on all basic arcs that are not part of the actual path from s to t is zero; if a basis exchange involves only those arcs, there is no flow change at all. To avoid this situation, we use another formulation, the biobjective shortest path tree (BSPT) problem. This formulation was proposed by Mote et al. [26]:

$$\min z(x) = \begin{cases} z_1(x) = \sum_{(i,j)\in A} c^1_{ij} x_{ij} \ , \\ z_2(x) = \sum_{(i,j)\in A} c^2_{ij} x_{ij} \ , \end{cases} \tag{12}$$

$$\text{s.t.} \qquad \sum_{(i,j)\in A} x_{ij} \ -- \sum_{(j,i)\in A} x_{ji} \ = \begin{cases} n-1 & \text{if } \ i = s \\ \\ -1 & \text{if } \ i \neq s \end{cases} \tag{13}$$

$$x_{ij} \geq 0 \text{ and integer, for all } \ (i,j) \in A. \tag{14}$$

By modifying the constraint set of the BSP problem, we now state the problem of finding the shortest path from a source node s to all the other nodes, resulting in non-zero flow on all basic arcs. Although not every basis exchange leads to a change of the shortest s_t path, it does lead to some change in the shortest path tree rooted at s. This approach ensures a flow change whenever the basis is changed.

BSP problem in a fuzzy network

Consider the BSP problem with the costs $c_{ij} = \left(\tilde{c}_{ij}^1, \tilde{c}_{ij}^2\right) \in N \times N$ associated with the arcs $(i,j) \in A$ being fuzzy numbers. A BSP problem in a fuzzy network can be formulated as:

$$\min z(x) = \begin{cases} z_1(x) = \sum_{(i,j)\in A} \tilde{c}_{ij}^1 x_{ij} \ , \\ z_2(x) = \sum_{(i,j)\in A} \tilde{c}_{ij}^2 x_{ij} \ , \end{cases} \tag{15}$$

$$\text{s.t.} \qquad \sum_{(i,j)\in A} x_{ij} \ -- \sum_{(j,i)\in A} x_{ji} \ = \begin{cases} 1 & \text{if } i = s \\ 0 & \text{if } i \neq s \\ -1 & \text{if } i = t \end{cases} \tag{16}$$

$$x_{ij} \in \{0,1\}, \qquad \text{for all } (i,j) \in A. \tag{17}$$

5. **Labeling Type Algorithms**

We begin by a definition for dominance.

Definition 5.1. Let $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_m)$ and $\tilde{y} = (\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_m)$ be two fuzzy vectors. We say $\tilde{x}$ dominates $\tilde{y}$ if $\tilde{x} < \tilde{y}$, where,

$$\tilde{x} < \tilde{y} \Leftrightarrow \tilde{x}_1 < \tilde{y}_1 \ , \ \tilde{x}_2 < \tilde{y}_2 \ , \ \dots, \ \tilde{x}_m < \tilde{y}_m. \tag{18}$$

The algorithm we consider here is the extension of Okada et al.'s algorithm [28]. They proposed their algorithm based on the multiple labeling method, but for a single objective. Here, we adjust the labeling algorithm for the biobjective model of shortest path problem in fuzzy networks. However, there are two difficulties associated with the definition of nondominated paths. With this definition, firstly a large number of comparisons may be required to find the nondominated paths, and secondly a path may rarely dominate all the other ones and thus almost all paths turn out to be nondominated. So, at first we propose two definitions for dominance check to help us reduce the number of comparisons. By using these definitions, we may not need to compare every path with all the other paths. To rectify the second difficulty, we suggest ranking the obtained nondominated paths using Algorithm 3.6. To give the definitions for reducing dominance comparisons, we first give a definition for ordering trapezoidal fuzzy numbers.

Definition 5.2. Let $\tilde{A} = (a_1, b_1, c_1, d_1)$ and $\tilde{B} = (a_2, b_2, c_2, d_2)$ be two trapezoidal fuzzy numbers. We say $\tilde{A} <_s \tilde{B}$ if one of the following 4 cases holds:

$$a_1 < a_2$$

$$a_1 = a_2,\ b_1 < b_2$$

$$a_1 = a_2,\ b_1 = b_2,\ c_1 < c_2$$

$$a_1 = a_2,\ b_1 = b_2,\ c_1 = c_2,\ d_1 < d_2$$

We now use Definition 5.2 to set up an ordering of the elements of the set $\tilde{X}$ that will be used in subsequent definitions.

Suppose that $\tilde{X} = \{\tilde{x}_1,\ \tilde{x}_2\ ,\ \ldots,\ \tilde{x}_n\}$ is a set of fuzzy vectors as in Definition 3.5. First, sort the set $\tilde{X}$ based on values of the first component; i.e., obtain $\tilde{X} = \left\{\ \tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^n\right\}$, where,
$\tilde{x}^1 = (\tilde{x}_{1k_1}, \tilde{x}_{2k_1}, \ldots, \tilde{x}_{mk_1}), \tilde{x}^2 = (\tilde{x}_{1k_2}, \tilde{x}_{2k_2}, \ldots, \tilde{x}_{mk_2}),\ ..,\ \tilde{x}^n = (\tilde{x}_{1k_n}, \tilde{x}_{2k_n}, \ldots, \tilde{x}_{mk_n})$, and $\tilde{x}_{1k_1} <_s \tilde{x}_{1k_2} <_s \cdots <_s \tilde{x}_{1k_n}$, according to Definition 5.2. For $i < j$, if $\tilde{x}_{1k_i} = \tilde{x}_{1k_j}$, then sort $\tilde{x}_{k_i}$ and $\tilde{x}_{k_j}$ based on the second components $\tilde{x}_{2k_i}$ and $\tilde{x}_{2k_j}$; if the second components are also equal as described for the first component, then sort $\tilde{X}$ based on the third components, and so on.

Now, for a given $\tilde{Y} = (\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_m)$, $y_i = \left(\acute{a}_i, \acute{b}_i, \acute{c}_i, \acute{d}_i\right)$, $i = 1, 2,\ \ldots, m$, we explore the dominance between $\tilde{Y}$ and members of the set $\tilde{X}$. Three cases can occur: (1) at least one of the members of $\tilde{X}$ dominates $\tilde{Y}$, (2) $\tilde{Y}$ dominates some members of $\tilde{X}$, and (3) neither $\tilde{X}$ nor $\tilde{Y}$ has dominating members. In the definitions below, we consider (1) and (2). Clearly, if neither (1) nor (2) occurs, then (3) takes place. The following definitions will be useful in reducing the number of comparisons in finding nondominated paths in labeling algorithms (algorithms 5.5 and 5.7 to be seen later).

Definition 5.3. Suppose that $\tilde{X} = \left\{\ \tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^n\right\}$ is an ordered set of fuzzy vectors (as above) and $\tilde{y} = (\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_m)$ is a fuzzy vector. We say $\tilde{X}$ dominates $\tilde{y}$, if there exists a $j \in \{1, \ldots, n\}$ such that $\tilde{x}^j < \tilde{y}$ based on Definition 5.1.

Definition 5.4. Suppose that $\tilde{X} = \left\{\ \tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^n\right\}$ is an ordered set of fuzzy vectors (as above) and $\tilde{y} = (\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_m)$ is a fuzzy vector. For every $j = n, \ldots, 1$, we say $\tilde{y}$ dominates $\tilde{x}^j$, if $\tilde{x}^j > \tilde{y}$ based on Definition 5.1.

Note that the above definitions can also be used for crisp sets.

In the first labeling approach to be given as Algorithm 5.5, each label is composed of four parts: the first and second objective values which are fuzzy numbers, the pointers and the distance parameters.

Let $j \in N$ be a node of $G(N,\ A)$. Since a number of labels may be assigned to the node j, the kth label associated with j, one of the labels, is denoted by $\left[\tilde{T}_k\left(p_{sj}\right),\ \tilde{C}_k\left(p_{sj}\right), (i, k_1), (d_{1k}, d_{2k})\right]_k$, where $i\ (i \neq j)$ is a predecessor of node j, $\tilde{T}_k\left(p_{sj}\right)$ and $\tilde{C}_k\left(p_{sj}\right)$ are time and cost along the path p_{sj} of the kth label of j, and k_1 indicates some label of i, for which $\tilde{T}_k\left(p_{sj}\right) = \tilde{T}_{k_1}\left(p_{si}\right) \bigoplus \tilde{T}_{ij}$ and $\tilde{C}_k\left(p_{sj}\right) = \tilde{C}_{k_1}\left(p_{si}\right) \bigoplus \tilde{C}_{ij}$, respectively.

Let P and T be sets of pointers to permanent and temporary labels, respectively. An element (i,k) in the set P or T means a pointer to the kth label of the node

i, and (d_{1k}, d_{2k}) is the comparing parameter that is computed using the distance formula (8). The procedure is as follows:

(1) Divide the problem into two single objective problems and solve them separately using dynamic programming (Algorithm 6.1 to be given later), and find the $\tilde{C}min$ and $\tilde{T}min$ values.
(2) Use the distance formula (8) and find the distances of $\tilde{T}_k\,(p_{sj})$ from $\tilde{T}min$ and $\tilde{C}_k\,(p_{sj})$ from $\tilde{C}min$, calling them d_{1k} and d_{2k}, respectively.

While a temporary label may be deleted or changed to a permanent label, a permanent label remains unchanged once a temporary label changes to a permanent one. The method for determining a permanent label uses the lexicographic order as specified below.

For two fuzzy vectors $\tilde{X} = (\tilde{x}_1,\ \tilde{x}_2)$ and $\tilde{Y} = (\tilde{y}_1,\ \tilde{y}_2)$ with $\tilde{x}_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4})$ and $\tilde{y}_i = (y_{i1}, y_{i2}, y_{i3}, y_{i4})$, for $i = 1, 2$, we say $\tilde{X}$ is lexicographically smaller than $\tilde{Y}$ if one of the following 8 cases holds:

$$x_{11} < y_{11}$$
$$x_{11} = y_{11} \text{ and } x_{12} < y_{12}$$
$$x_{11} = y_{11}\ ,\ x_{12} = y_{12} \text{ and } x_{13} < y_{13}$$
$$x_{11} = y_{11}\ ,\ x_{12} = y_{12}\ ,\ x_{13} = y_{13} \text{ and } x_{14} < y_{14}$$
$$x_{11} = y_{11}\ ,\ x_{12} = y_{12}\ ,\ x_{13} = y_{13}\ , x_{14} = y_{14} \text{ and } x_{21} < y_{21}$$
$$x_{11} = y_{11}\ ,\ x_{12} = y_{12}\ ,\ x_{13} = y_{13}\ , x_{14} = y_{14}\ , x_{21} = y_{21} \text{ and } x_{22} < y_{22}$$
$$x_{11} = y_{11}\ ,\ x_{12} = y_{12}\ ,\ x_{13} = y_{13}\ , x_{14} = y_{14}\ , x_{21} = y_{21},\ x_{22} = y_{22} \text{ and } x_{23} < y_{23}$$
$$x_{11} = y_{11}\ , x_{12} = y_{12}\ , x_{13} = y_{13}\ , x_{14} = y_{14}\ , x_{21} = y_{21},\ x_{22} = y_{22},\ x_{23} = y_{23} \text{ and } x_{24} < y_{24}$$

The lexicographic order can be changed according to a decision maker's preference. However, the order does not play an important role in the steps of our algorithm.

We are now ready to give our first algorithm for the BSP problem as Algorithm 5.5.

Algorithm 5.5. The first labeling approach for finding the shortest path in biobjective networks.

```
(0) [Initialization]
    (i) Assign the label $\left[\tilde{0}, \tilde{0}, (-,-), (-,-)\right]_1$ to node $s$, where $\tilde{0} = (0,0,0,0)$.
    (ii) Set the label to a temporary one and initialize the set $P$ to be empty
         as follows:
              $P = \emptyset\ ,\ T = \{(1,1)\}$ .
(1) [Label selection]
    (i) If, $T$=$\emptyset$ then go to (4), else among all the temporary labels determine
        lexicographically smallest one. Let it be the $k$th label associated with
        node $i$.
    (ii) Remove the pointer $(i,k)$ to this label from $T$, and add it to $P$.
```

(2) [Label scan and dominance check]

For each node $j \in N$ such that $(i, j) \in A$, do the following 3 steps:

(i) Determine a fuzzy path cost and time as follows:

$$\tilde{T}_l\left(p_{sj}\right) = \tilde{T}_k\left(p_{si}\right) \bigoplus \tilde{T}_{ij}$$
$$\tilde{C}_l\left(p_{sj}\right) = \tilde{C}_k\left(p_{si}\right) \bigoplus \tilde{C}_{ij}.$$

(ii) Let $\left[\tilde{T}_l\left(p_{sj}\right),\ \tilde{C}_l\left(p_{sj}\right), (i, k), (d_{1l}, d_{2l})\right]_l$ be a new temporary label of the node j, and update the set T by adding this label to T.

(iii) Perform the dominance check among all temporary labels of the node j as follows:

First, check if the new label is dominated: find all temporary and permanent labels of node j in T and P, that is, all u such that $(j, u) \in T \cup P$. Let d_1 and d_2 be the comparing parameters. Contrast the new label with the existing labels of node j.

If the new label is dominated based on Definition 5.3, then

Remove the new label from the set T and stop

Else

Check if the new label dominates some existing temporary labels of node j: find all temporary labels of node j in T, that is, all u such that $(j, u) \in T$. Compare the new label with the existing temporary labels of node j. If the new label dominates some previous labels based on Definition 5.4, then remove these previous labels from the set T.

End if

(3) Go to (1).

(4) [Nondominated path composition]

Find the nondominated paths from s to t. For that, the two pointers of each label are used to recompose backwards the list of nodes of the path until the node s is reached.

(5) [Ranking the nondominated paths]

Using Algorithm 3.6, first rank nondominated paths according to the first objective, and then based on the second objective, and finally according to both objectives. By this way, we let the decision maker decide based on his/her preference.

(6) [Termination]

Stop.

We now consider an example to show how Algorithm 5.5 works.

Example 5.6. Suppose that we have a network with 6 nodes and 8 arcs as shown in Figure 1. For every arc, two fuzzy numbers are considered, the first as the cost coefficient for the first objective and the second as the time coefficient for the second objective. Find the shortest path from 1 to 6 using Algorithm 5.5.

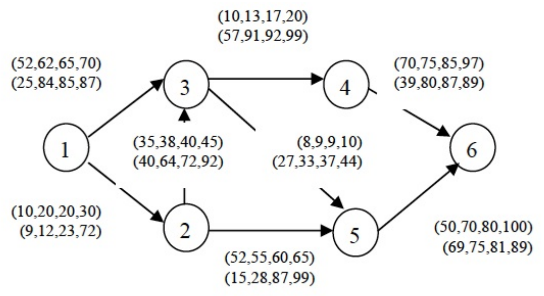

FIGURE 1. A Fuzzy Network with Two Fuzzy Numbers for Each Arc

Iteration 1

Step (0)

Assign label $\left[\tilde{0}, \tilde{0}, (-,-), (-,-)\right]_1$ to the starting node, here node 1. Let temporary and permanent label sets be $P = \emptyset$ and $T = \{(1,1)\}$, respectively.

Step (1)

Since $T \neq \emptyset$, we select the label of node 1 as the smallest label and have:

$$T = \emptyset \ , \ P = \{(1,1)\} \, .$$

Step (2)

For outgoing arcs from node 1, here nodes 2 and 3, we have:

Label of node 2: $[(10,20,20,30) \, , \ (9,12,23,72) \, , (1,1) \, , (12.2, 11.4)]_1$

Label of node 3: $[(52,62,65,70) \, , \ (25,84,85,87) \, , (1,1) \, , (53.5, 28.4)]_1$

$$T = \{(2,1) \, , (3,1)\} \, .$$

Since there is just one label for nodes 2 and 3, we do not have any dominance check.

Step (3)

Return to Step (1) and continue with the next iteration.

Continuing this procedure, after 13 iterations the algorithm terminates. The results being labels of the nodes are shown in Tables 1 and 2.

We realize that only the second and fifth labels of node 6 are dominated. In iteration 13, we have $T = \emptyset$, and so we perform Step (4) and compose the nondominated paths. According to the nondominated labels of node 6 and the pointers, we find 3 nondominated paths as follows:

Path 1: $\left\{1-2-3-5-6 \ : \ [(103,137,149,185) \, , \ (145,184,213,297) \, , (136.8, 122.8)]_1 \ \right\}$

Path 2: $\left\{1-3-5-6 \ : \ [(110,141,154,180) \, , \ (121,192,203,220) \, , (139, 104.2)]_3 \ \right\}$

Path 3: $\left\{1-2-5-6 \ : \ [(112,145,160,195) \, , \ (93,115,191,260) \, , (146.3, 93.8)]_4 \ \right\}$.

Node 1	$[\tilde{0}, \tilde{0}, (-,-), (-,-)]_1$
Node 2	$[(10,20,20,30),\ (9,12,23,72), (1,1), (12.2,11.4)]_1$
Node 3	$[(52,62,65,70),\ (25,84,85,87), (1,1), (53.5,28.4)]_1$ $[(45,58,60,75),\ (49,76,95,164), (2,1), (51.2,47.6)]_2$
Node 4	$[(55,71,77,95),\ (106,167,187,263), (3,2), (66.6,100.9)]_1$ $[(62,75,82,90),\ (82,175,177,186), (3,1), (68.8,81.5)]_2$
Node 5	$[(62,75,80,95),\ (24,40,110,171), (2,1), (69.7,45.6)]_1$ $[(53,67,69,85),\ (76,109,132,208), (3,2), (60.2,70.3)]_2$ $[(60,71,74,80),\ (52,117,122,131), (3,1), (62.5,51.2)]_3$
Node 6	$[(103,137,149,185),\ (145,184,213,297), (5,2), (136.8,123)]_1$ $[(125,146,162,192),\ (145,247,274,352), (4,1), (148.8,148.4)]_2$ $[(110,141,154,180),\ (121,192,203,220), (5,3), (139,104.2)]_3$ $[(112,145,160,195),\ (93,115,191,260), (5,1), (146.3,93.8)]_4$ $[(132,150,167,187),\ (121,255,264,275), (4,2), (151.1,126.6)]_5$

Table 1. Resulting Labels for Each Node

Iteration 1	$P=\{(1,1)\}$, $T=\{(2,1),(3,1)\}$
Iteration 2	$P=\{(1,1),(2,1)\}$, $T=\{(3,1),(3,2),(5,1)\}$
Iteration 3	$P=\{(1,1),(2,1),(3,2)\}$, $T=\{(3,1),(5,1),(4,1),(5,2)\}$
Iteration 4	$P=\{(1,1),(2,1),(3,2),(3,1)\}$, $T=\{(5,1),(4,1),(5,2),(4,2),(5,3)\}$
Iteration 5	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2)\}$, $T=\{(5,1),(4,1),(4,2),(5,3),(6,1)\}$
Iteration 6	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2),(4,1)\}$, $T=\{(5,1),(4,2),(5,3),(6,1),(6,2)\}$
Iteration 7	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2),(4,1),(5,3)\}$, $T=\{(5,1),(4,2),(6,1),(6,3)\}$
Iteration 8	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2),(4,1),(5,3),(5,1)\}$, $T=\{(5,1),(4,2),(5,3),(6,1),(6,3),(6,4)\}$
Iteration 9	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2),(4,1),(5,3),(5,1),(4,2)\}$, $T=\{(6,1),(6,3),(6,4),(6,5)\}$
Iteration 10	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2),(4,1),(5,3),(5,1),(4,2),(6,1)\}$, $T=\{(6,3),(6,4)\}$
Iteration 11	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2),(4,1),(5,3),(5,1),(4,2),(6,1),$ $(6,3)\}$, $T=\{(6,4)\}$
Iteration 12	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2),(4,1),(5,3),(5,1),(4,2),(6,1),(6,3),$ $(6,4)\}$, $T=\emptyset$

Table 2. Pointer Sets in Iterations of Algorithm 5.5

Step (5)

To rank the paths based on the first and second objectives, using Algorithm 3.6 we form the sets S_1 and S_2 as follows:

$$S_1=\{1,1.01,1.07\}\ ,\ S_2=\{1.3,1.1,1\}.$$

So, based on the first objective, the ranking is paths 1, 2 and 3, while based on the second objective, the ranking is paths 3, 2 and 1.

Ranking based on both objectives yields $R = \{2.3, 2.11, 2.07\}$, and so paths are ranked as paths 3, 2 and 1.

The second labeling approach is quite similar to Algorithm 5.5, except that we do not calculate the distance in iterations of the algorithm but they are computed in the final iteration for comparing the paths. So, here the fourth element of the labels is discarded. Algorithm 5.5, using the distance method for comparing fuzzy numbers may not compute all the nondominated paths, but in the following algorithm we find all the nondominated paths.

We now give Algorithm 5.7.

Algorithm 5.7. The second labeling approach for finding the shortest path in biobjective networks.

```
(0) [Initialization]
    (i) Assign the label $\left[\tilde{0}, \tilde{0}, (-,-)\right]_1$ to node $s$, where $\tilde{0} = (0,0,0,0)$.
    (ii) Form the sets $P$ and $T$ exactly the same as the step (0) (ii) of Algorithm 5.5.
(1) [Label selection]
    Do this step exactly the same as step (1) of Algorithm 5.5.
(2) [Label scan and dominance check]
    Do parts (i) and (ii) of this step exactly the same as the ones in step (2) of Algorithm 5.5, and for part (iii), performing the dominance check, in comparing labels do not use $d_1$ and $d_2$ but compare the labels directly.
(3) Go to (1).
(4) [ Nondominated path composition]
    Do this step exactly the same as step (4) of Algorithm 5.5.
(5) [Ranking the nondominated paths]
    Using Algorithm 3.6, form the set $d$ and rank nondominated paths quite similar to this step in Algorithm 5.5.
(6) [Termination]
    Stop.
```

Now, to illustrate Algorithm 5.7, we solve an example.

Example 5.8. Solve Example 5.6 using Algorithm 5.7.

Following the same procedure as demonstrated for application of Algorithm 5.5, the results obtained are shown in Tables 3 and 4.

In iterations 1 to 12, only the second and fifth labels of node 6 are dominated. In iteration 13, we have $T = \emptyset$, and so we perform Step (4) and compose the nondominated paths. According to the nondominated labels of node 6 and the pointers, we find 3 nondominated paths as follows:

Path 1: $\{1-2-3-5-6 \ : \ [(103,137,149,185)\,,\ (145,184,213,297)]_1\ \}$
Path 2: $\{1-3-5-6 \ : \ [(110,141,154,180)\,,\ (121,192,203,220)]_3\ \}$
Path 3: $\{1-2-5-6 \ : \ [(112,145,160,195)\,,\ (93,115,191,260)]_4\ \}$.

Node 1	$[\tilde{0}, \tilde{0}, (-,-)]_1$
Node 2	$[(10,20,20,30)\,,\ (9,12,23,72)\,,(1,1)]_1$
Node 3	$[(52,62,65,70)\,,\ (25,84,85,87)\,,(1,1)]_1\ ,$ $[(45,58,60,75)\,,\ (49,76,95,164)\,,(2,1)]_2$
Node 4	$[(55,71,77,95)\,,\ (106,167,187,263)\,,(3,2)]_1\ ,$ $[(62,75,82,90)\,,\ (82,175,177,186)\,,(3,1)]_2$
Node 5	$[(62,75,80,95)\,,\ (24,40,110,171)\,,(2,1)]_1\ ,$ $[(53,67,69,85)\,,\ (76,109,132,208)\,,(3,2)]_2\ ,$ $[(60,71,74,80)\,,\ (52,117,122,131)\,,(3,1)]_3$
Node 6	$[(103,137,149,185)\,,\ (145,184,213,297)\,,(5,2)]_1\ ,$ $[(125,146,162,192)\,,\ (145,247,274,352)\,,(4,1)]_2\ ,$ $[(110,141,154,180)\,,\ (121,192,203,220)\,,(5,3)]_3\ ,$ $[(112,145,160,195)\,,\ (93,115,191,260)\,,(5,1)]_4\ ,$ $[(132,150,167,187)\,,\ (121,255,264,275)\,,(4,2)]_5$

TABLE 3. Resulting Labels for Each Node

Iteration 1	$\mathbf{P}=\{(\mathbf{1},\mathbf{1})\}\ ,\ \mathbf{T}=\{(\mathbf{2},\mathbf{1}),(\mathbf{3},\mathbf{1})\}$
Iteration 2	$P=\{(1,1),(2,1)\}\ ,\ T=\{(3,1),(3,2),(5,1)\}$
Iteration 3	$P=\{(1,1),(2,1),(3,2)\}\ ,$ $T=\{(3,1),(5,1),(4,1),(5,2)\}$
Iteration 4	$P=\{(1,1),(2,1),(3,2),(3,1)\}\ ,$ $T=\{(5,1),(4,1),(5,2),(4,2),(5,3)\}$
Iteration 5	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2)\}\ ,$ $T=\{(5,1),(4,1),(4,2),(5,3),(6,1)\}$
Iteration 6	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2),(4,1)\},$ $T=\{(5,1),(4,2),(5,3),(6,1),(6,2)\}$
Iteration 7	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2),(4,1),(5,3)\},$ $T=\{(5,1),(4,2),(6,1),(6,3)\}$
Iteration 8	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2),(4,1),(5,3),$ $(5,1)\},\ T=\{(4,2),(6,1),(6,3),(6,4)\}$
Iteration 9	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2),(4,1),(5,3),(5,1),$ $(4,2)\},\ T=\{(6,1),(6,3),(6,4),(6,5)\}$
Iteration 10	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2),(4,1),(5,3),(5,1),$ $(4,2),(6,1)\},\ T=\{(6,3),(6,4)\}$
Iteration 11	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2),(4,1),(5,3),(5,1),$ $(4,2),(6,1),(6,3)\},\ T=\{(6,4)\}$
Iteration 12	$P=\{(1,1),(2,1),(3,2),(3,1),(5,2),(4,1),(5,3),(5,1),$ $(4,2),(6,1),(6,3),(6,4)\},\ T=\emptyset$

TABLE 4. Pointer Sets in Iterations of Algorithm 5.7

Step (5)

To rank the paths based on the first and second objectives, we first compute (d_1, d_2) for every path and then form the sets S_1 and S_2 using Algorithm 3.6:

$$S_1 = \{1, 2.2, 5.5\}\ ,\ S_2 = \{3.5, 2.4, 1\}\,.$$

So, based on the first objective, the ranking is paths 1, 2 and 3, while based on the second objective, the ranking is paths 3, 2 and 1.

Ranking based on both objectives is $R = \{4.5, 4.6, 6.5\}$, and so paths are ranked as paths 1, 2 and 3.

Remark 5.9. Okada and Soper point out in [28] that the multicriteria shortest path problems belong to the class of NP-hard discrete optimization problems. Let V_j be the permanent labels of node j, and let V_{max} be $\max\{V_1, V_2, \ldots, V_N\}$. The computing complexity of labeling algorithm is bounded by $\mathrm{O}\left(N^3 V_{max}^2\right)$ and the space requirement is bounded by $\mathrm{O}\left(N V_{max}\right)$ excluding the problem data. V_{max}, however, may exponentially grow due to conflicting objectives in a multicritria shortest path problem, but our use of the distance formula (8) for dominance check helps in reducing the size of V_{max}, and hence improves the efficiency of the labeling algorithm.

6. A Dynamic Programming Approach

Here, we propose a procedure for converting two fuzzy numbers into a single number for comparison purposes. Using this procedure, we can then adapt the dynamic programming approach for a single objective fuzzy network to the biobjective model.

A dynamic programming approach for finding a shortest path is composed of three main parts as specified below:

- Objective function:
  Let $f_i(j,k)$ be the objective function giving the length of the shortest path from node j to node k with some nodes in $\{1,\ 2,\ \ldots, i\}$ as intermediate nodes and $P_i(j,k)$ be the path pointer that is the last intermediate nodes on the shortest path from node j to node k using $\{1,\ 2,\ \ldots, i\}$ as intermediate nodes. Clearly, for $i=0$, there can be no intermediate node between node i and node j, and so there exists a directed arc from node j to node k.
- Recursive relation:

$$f_i(j,k) = \min\left\{\begin{array}{ll} f_{i-1}(j,k)\,, & \\ f_{i-1}(j,i) + f_{i-1}(i,k) & \qquad (i = 1,\ 2,\ \ldots,\ n)\end{array}\right\},$$

$$P_i(j,k) = \begin{cases} P_{i-1}(j,k) & \text{if } i \text{ is not on shortest path from } j \text{ to } k \text{ using } \{1,\ 2,\ \ldots, i\} \\ P_{i-1}(j,i) & \text{otherwise.}\end{cases}$$

- Starting conditions:

$$f_0(j,k) = \begin{cases} d_{jk} & (j,k) \in A \\ \infty & (j,k) \notin A,\end{cases}$$

$$P_0\left(j,k\right)=\begin{cases} k & (j,k)\in A \\ 0 & (j,k)\notin A. \end{cases}$$

Dynamic programming is used to find the shortest path between every pair of nodes in a fuzzy network. However, for fuzzy networks, since the two values to be compared in determining the minimum value are fuzzy numbers, we should find a way for comparing the fuzzy numbers. In the following algorithm based on the approach in [24], the distance method given by eq. (8) is used. This approach has also recently been effectively used for computing a shortest path in a fuzzy network by Tajdin et al. [37].

Algorithm 6.1. A dynamic program to find a shortest path between every pair of nodes in a fuzzy network.

(0) Let $i=0$. If there is an arc between node j to k, then let $f_i\left(j,k\right)=d_{jk}$ and $P_i\left(j,k\right)=k$, else let $f_i\left(j,k\right)=\infty$ and $P_i\left(j,k\right)=0$.

(1) Let $i=i+1$.
Do the following 3 steps to find $\tilde{f}_i\left(j,k\right)$, for $k=1,2,\ldots,N$, $j=1,2,\ldots,N$.
  (i) Use Algorithm 3.4 and determine:
$$\text{MV=min}\Big[\tilde{f}_{i-1}\left(j,k\right)\ \ ,(\tilde{f}_{i-1}\left(j,i\right)+\tilde{f}_{i-1}\left(i,k\right))\Big].$$
  (ii) Use the distance formula (8), compute the distances of $\tilde{f}_{i-1}\left(j,k\right)$ and $\tilde{f}_{i-1}\left(j,i\right)+\tilde{f}_{i-1}\left(i,k\right)$ from MV, and decide the one with the least distance to be the smaller one.
  (iii) Based on the computed distances find:
$$\tilde{f}_i\left(j,k\right)=\min\Big[\tilde{f}_{i-1}\left(j,k\right)\ \ ,(\tilde{f}_{i-1}\left(j,i\right)+\tilde{f}_{i-1}\left(i,k\right))\Big].$$

(2) If node i is not located in the shortest path obtained in Step (1), then let $P_i\left(j,k\right)=P_{i-1}\left(j,k\right)$ else let $P_i\left(j,k\right)=P_{i-1}\left(j,i\right)$.

(3) If $i<N$ then Go To (1) else stop.

Before deploying a dynamic programming approach for BSP problems, we should find a way to compare the values of two fuzzy objectives. By comparing two fuzzy objective values directly, we cannot expect to find the minimum value and hence be able to compute the shortest path. For this reason, we propose an algorithm as a function to convert the values of two fuzzy numbers into one crisp number to be compared. This algorithm will be used in the recursive function of a dynamic programming algorithm (Algorithm 6.3 to be seen later) to find the minimum fuzzy values of objectives and the shortest path.

Algorithm 6.2. Convert the two fuzzy objective values into one comparable number.

(0) For the BSP problem in a fuzzy network, suppose that for every arc $(i,j)\in A$ we have $\tilde{c}^1_{ij}=\tilde{C}_{ij}$ and $\tilde{c}^2_{ij}=\tilde{T}_{ij}$, with $\tilde{C}_{ij}=(a_{ij},b_{ij},c_{ij},d_{ij})$ and $\tilde{T}_{ij}=\left(\acute{a}_{ij},\acute{b}_{ij},\acute{c}_{ij},\acute{d}_{ij}\right)$. Find $\widetilde{MC}$ and $\widetilde{MT}$ as follows:

$$\widetilde{MC} = (min\,(a_{ij})\ ,min\,(b_{ij})\ ,min\,(c_{ij})\ ,min\,(d_{ij})\ )$$

$$\widetilde{MT} = \left(min\,(\acute{a}_{ij})\ ,min\left(\acute{b}_{ij}\right)\ ,min\,(\acute{c}_{ij})\ ,min\left(\acute{d}_{ij}\right)\ \right).$$

(1) Divide the problem into two single objective problems according to the first and second objectives. Then, solve these two problems by Algorithm 6.1 and call the answers $\tilde{C}^*$ and $\tilde{T}^*$, respectively. Compute the distances between $\tilde{C}^*$ and $\widetilde{MC}$, and $\tilde{T}^*$ and $\widetilde{MT}$, using the distance formula (8) and call them d_1^* and d_2^*, respectively.

(2) For path from j to k find the two components of $\tilde{f}_i\,(j,k)$ as follows:

$$\begin{cases} \tilde{f}_{1i}\,(j,k) = \tilde{C}_{jk} \\ \tilde{f}_{2i}\,(j,k) = \tilde{T}_{jk}. \end{cases}$$

Compute the distance between $\tilde{f}_{1i}\,(j,k)$ and $\widetilde{MC}$, and $\tilde{f}_{2i}\,(j,k)$ and $\widetilde{MT}$, using the distance formula (8), and call them d_{1i} *and* d_{2i}, respectively.

(3) Compute the ratios and function R as follows:

$$R_{1i}=d_{1i}/d_1^*\ ,\ R_{2i}=d_{2i}/d_2^*\ ,\ R_i=R_{1i}+R_{2i},$$

$$R\left(\tilde{f}_i\,(j,k)\right)=R_i.$$

Now, the value of R, obtained by Algorithm 6.2, can serve as a single objective value and be used for the comparison corresponding to the two objectives. We can now give our dynamic program for finding the biobjective shortest path between every pair of nodes in a fuzzy network, as specified by Algorithm 6.3 below.

Algorithm 6.3. A dynamic program for finding the biobjective shortest path between every pair of nodes in a fuzzy network.

(0) Use steps (0) and (1) of Algorithm 6.2 to compute d_1^* and d_2^*. Let $i = 0$. Compute the components of $\tilde{f}_i\,(j,k)$ as:

$$\begin{cases} \tilde{f}_{1i}\,(j,k) = \tilde{C}_{jk} \\ \tilde{f}_{2i}\,(j,k) = \tilde{T}_{jk}. \end{cases}$$

If there is an arc between nodes j and k, then Compute $R\left(\tilde{f}_i\,(j,k)\right)$ using Algorithm 6.2 and let $P_i\,(j,k)=k$, else let $R\left(\tilde{f}_i\,(j,k)\right) = \infty$ and $P_i\,(j,k)=0$.

(1) Let $i = i+1$. Use the distance formula (8) and Algorithm 6.2, and compute $R\left(\tilde{f}_i\,(j,k)\right)$ and $\tilde{f}_i\,(j,k)$, for $k = 1,2,\ldots,N$, $j = 1,2,\ldots,N$, as follows:

$$R(\tilde{f}_i\,(j,k)) = \min\left[R\left(\tilde{f}_{i-1}\,(j,k)\right), R(\tilde{f}_{i-1}\,(j,i) + \tilde{f}_{i-1}\,(i,k))\right].$$

If $R(\tilde{f}_i\,(j,k)) = R\left(\tilde{f}_{i-1}\,(j,k)\right)$, then let $\tilde{f}_i\,(j,k) = \tilde{f}_{i-1}\,(j,k)$, else let $\tilde{f}_i\,(j,k) = \tilde{f}_{i-1}\,(j,i) + \tilde{f}_{i-1}\,(i,k)$.

(2) If node i is not located in the shortest path obtained in Step (1), then let $P_i(j,k) = P_{i-1}(j,k)$, else let $P_i(j,k) = P_{i-1}(j,i)$.

(3) If $i < N$, then Go To (1) else stop.

Remark 6.4. As indicated in [37], Algorithm 6.3 terminates after N outer iterations corresponding to each i. A total of $N(N-1)^2$ additions and comparisons are needed for every i, and thus the total number of additions and comparisons turns out to be $N^2(N-1)^2$. For each comparison, two fuzzy additions and five simple operations are performed.

In the following example, we illustrate Algorithm 6.3.

Example 6.5. Solve Example 5.6 using Algorithm 6.3.

Iteration 1

Step (0)

For the network, we have: $\widetilde{MC} = (8,9,9,10)$, $\widetilde{MT} = (9,12,23,44)$. Based on the first objective, the optimal path is 1-2-3-5-6 with $C^* = (103,137,149,185)$, and based on the second objective, the optimal path is 1-2-5-6 with $T^* = (93,115,191,260)$. So, we have $d_1^* = 136.8$ and $d_2^* = 93.8$ as the distances between C^* and $\widetilde{MC}$, and T^* and $\widetilde{MT}$, respectively.

i= 0 and the computed values for $\tilde{f}_0(j,k)$, $R\left(\tilde{f}_0(j,k)\right)$ and $P_0(j,k)$ are shown in Tables 5-7.

	1	2	3	4	5	6
1	—	(10,20,20,30), (9,12,23,72)	(52,62,65,70), (25,84,85,87)	—	—	—
2	—	—	(35,38,40,45), (40,64,72,92)	—	(52,55,60,65), (15,28,87,99)	—
3	—	—	—	(10,13,17,20), (57,91,92,99)	(8,9,9,10), (27,33,37,44)	—
4	—	—	—	—	—	(70,75,85,97), (39,80,87,89)
5	—	—	—	—	—	(50,70,80,100), (69,75,81,89)
6	—	—	—	—	—	—

Table 5. $\tilde{f}_0(j,k)$ Values

Steps (1) and (2)

i= 1 and the computed values for $\tilde{f}_1(j,k)$, $R\left(\tilde{f}_1(j,k)\right)$ and $P_1(j,k)$ are shown in tables 8-10.

Step (3)

Since $1 < 6$, the next iteration starts with Step (1).

	1	2	3	4	5	6
1	—	0.21	0.69	—	—	—
2	—	—	0.55	—	0.65	—
3	—	—	—	0.5	0.17	—
4	—	—	—	—	—	0.89
5	—	—	—	—	—	0.95
6	—	—	—	—	—	—

Table 6. $R\left(\tilde{f}_0\left(j,k\right)\right)$ Values

	1	2	3	4	5	6
1	—	2	3	—	—	—
2	—	—	3	—	5	—
3	—	—	—	4	5	—
4	—	—	—	—	—	6
5	—	—	—	—	—	6
6	—	—	—	—	—	—

Table 7. $P_0\left(j,k\right)$ Values

	1	2	3	4	5	6
1	—	(10,20,20,30), (9,12,23,72)	(52,62,65,70), (25,84,85,87)	—	—	—
2	—	—	(35,38,40,45), (40,64,72,92)	—	(52,55,60,65), (15,28,87,99)	—
3	—	—	—	(10,13,17,20), (57,91,92,99)	(8,9,9,10), (27,33,37,44)	—
4	—	—	—	—	—	(70,75,85,97), (39,80,87,89)
5	—	—	—	—	—	(50,70,80,100), (69,75,81,89)
6	—	—	—	—	—	—

Table 8. $\tilde{f}_1\left(j,k\right)$ Values

	1	2	3	4	5	6
1	—	0.21	0.69	—	—	—
2	—	—	0.55	—	0.65	—
3	—	—	—	0.5	0.17	—
4	—	—	—	—	—	0.89
5	—	—	—	—	—	0.95
6	—	—	—	—	—	—

Table 9. $R\left(\tilde{f}_1\left(j,k\right)\right)$ Values

	1	2	3	4	5	6
1	—	2	3	—	—	—
2	—	—	3	—	5	—
3	—	—	—	4	5	—
4	—	—	—	—	—	6
5	—	—	—	—	—	6
6	—	—	—	—	—	—

TABLE 10. $P_1\left(j,k\right)$ Values

Continuing this procedure, at iteration 6 we have:
Steps (1) and(2)
$i=6$ and the computed values for $\tilde{f}_6\left(j,k\right)$, $R\left(\tilde{f}_6\left(j,k\right)\right)$ and $P_6\left(j,k\right)$ are given in tables 11-13.

	1	2	3	4	5	6
1	—	0.21	0.69	1.37	0.99	2.06
2	—	—	0.55	1.22	0.65	1.72
3	—	—	—	0.5	0.17	1.25
4	—	—	—	—	—	0.89
5	—	—	—	—	—	0.95
6	—	—	—	—	—	—

TABLE 11. $R\left(\tilde{f}_6\left(j,k\right)\right)$ Values

	1	2	3	4	5	6
1	—	(10,20,20,30), (9,12,23,72)	(52,62,65,70), (25,84,85,87)	(62,75,82,90), (82,175,177,186)	(62,75,80,95), (24,40,110,171)	(112,145,160,195), (93,115,191,260)
2	—	—	(35,38,40,45), (40,64,72,92)	(45,51,57,65), (97,155,164,191)	(52,55,60,65), (15,28,87,99)	(102,125,140,165), (84,103,168,188)
3	—	—	—	(10,13,17,20), (57,91,92,99)	(8,9,9,10), (27,33,37,44)	(58,79,89,110), (96,108,118,133)
4	—	—	—	—	—	(70,75,85,97), (39,80,87,89)
5	—	—	—	—	—	(50,70,80,100), (69,75,81,89)
6	—	—	—	—	—	—

TABLE 12. $\tilde{f}_6\left(j,k\right)$ Values

Step (3)
Since $6 \not< 6$, the algorithm terminates. So, the path 1-2-5-6 is the shortest path and the values of the objectives are:

$$\tilde{C}^{**}=(112,145,160,195)\ ,\ \tilde{T}^{**}=(93,115,191,260).$$

	1	2	3	4	5	6
1	—	2	3	3	2	5
2	—	—	3	3	5	5
3	—	—	—	4	5	5
4	—	—	—	—	—	6
5	—	—	—	—	—	6
6	—	—	—	—	—	—

Table 13. $P_6\,(j,k)$ Values

7. A Comparison of the Proposed Algorithms

Here, we solve an example with 23 nodes, taken from Mahdavi et al. [24], as shown in Figure 2. We assume the cost and time associated with the arcs to be fuzzy trapezoidal numbers as shown in Table 17. We used the MATLAB 7.3 software environment to implement our algorithms and report the results. Each algorithm solves the problem three times, the first time based on the first objective preference, the second time based on the second objective preference and the third time based on both objectives with natural preference. The results obtained are given in Tables 18-24.

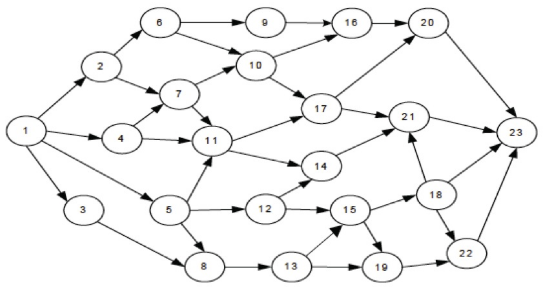

Figure 2. A Fuzzy Network with Two Fuzzy Trapezoidal Numbers as Cost and Time for Each Arc

As pointed out before, using the labeling type Algorithms 5.5 and 5.7, we obtained some nondominated paths and ranked them based on first, second or both objectives. But in the dynamic programming algorithm, Algorithm 6.3, we obtained only one path as the shortest path. In Tables 25 and 26, we have a comparison of the performance of these algorithms. As we see, the first and second ranked solutions are the same as for both Algorithms 5.5 and 5.7. This is reasonable, because both are labeling algorithms; however, we see that using the distance method for comparison of fuzzy numbers does not have much effect on the ranked solutions.

Arc	Fuzzy Number	Arc	Fuzzy Number	Arc	Fuzzy Number
(1,2)	(12,13,15,17) (14,17,21,24)	(7,10)	(9,10,12,13) (12,17,22,23)	(15,18)	(8,9,11,13) (12,17,21,26)
(1,3)	(9,11,13,15) (16,17,19,21)	(7,11)	(6,7,8,9) (13,17,20,21)	(15,19)	(5,7,10,12) (8,9,13,18)
(1,4)	(8,10,12,13) (17,19,21,23)	(8,12)	(5,8,9,10) (13,14,16,18)	(16,20)	(9,12,14,16) (9,11,13,18)
(1,5)	(7,8,9,10) (15,16,20,25)	(8,13)	(3,5,8,10) (5,6,10,12)	(17,20)	(7,10,11,12) (7,9,11,16)
(2,6)	(5,10,15,16) (3,8,11,16)	(9,16)	(6,7,9,10) (3,7,10,15)	(17,21)	(6,7,8,10) (16,21,23,25)
(2,7)	(6,11,11,13) (14,17,18,19)	(10,16)	(12,13,16,17) (16,17,21,26)	(18,21)	(15,17,18,19) (8,9,11,13)
(3,8)	(10,11,16,17) (11,14,19,22)	(10,17)	(15,19,20,21) (15,18,23,25)	(18,22)	(3,5,7,9) (3,5,9,12)
(4,7)	(17,20,22,24) (5,6,8,9)	(11,14)	(8,9,11,13) (16,18,22,26)	(18,23)	(5,7,9,11) (16,19,21,24)
(4,11)	(6,10,13,14) (14,16,21,23)	(11,17)	(6,9,11,13) (5,10,12,15)	(19,22)	(15,16,17,19) (2,3,6,8)
(5,8)	(6,9,11,13) (10,14,19,23)	(12,14)	(13,14,16,18) (11,13,14,16)	(20,23)	(13,14,16,17) (10,14,17,18)
(5,11)	(7,10,13,14) (5,6,8,11)	(12,15)	(12,14,15,16) (6,8,10,11)	(21,23)	(12,15,17,18) (16,21,24,26)
(5,12)	(10,13,15,17) (1,4,8,13)	(13,15)	(10,12,14,15) (3,7,10,15)	(22,23)	(4,5,6,8) (11,13,17,19)
(6,9)	(6,8,10,11) (12,14,19,23)	(13,19)	(17,18,19,20) (10,15,16,20)		
(6,10)	(10,11,14,15) (14,15,20,24)	(14,21)	(11,12,13,14) (3,6,8,10)		

Table 14. Fuzzy Numbers for Arcs

Path	Value of the first objective	Value of the second objective
1-5-11-17-21-23	(38 49 58 65)	(57 74 87 102)
1-5-11-17-20-23	(40 51 60 66)	(42 55 68 85)
1-5-12-15-18-23	(42 51 59 67)	(50 64 80 99)
1-5-12-15-18-22-23	(44 54 63 73)	(48 63 85 106)
1-5-12-14-21-23	(53 62 70 77)	(46 60 74 90)
1-5-8-13-19-22-23	(52 61 70 80)	(53 67 88 107)

Table 15. Results of Algorithm 5.5 Based on the First Objective Preference

Path	Value of the first objective	Value of the second objective
1-5-11-17-20-23	(40 51 60 66)	(42 55 68 85)
1-5-12-14-21-23	(53 62 70 77)	(46 60 74 90)
1-5-12-15-18-23	(42 51 59 67)	(50 64 80 99)
1-5-12-15-18-22-23	(44 54 63 73)	(48 63 85 106)
1-5-8-13-19-22-23	(52 61 70 80)	(53 67 88 107)
1-5-11-17-21-23	(38 49 58 65)	(57 74 87 102)

Table 16. Results of Algorithm 5.5 Based on the Second Objective Preference

Path	Value of the first objective	Value of the second objective
1-5-11-17-20-23	(40 51 60 66)	(42 55 68 85)
1-5-12-15-18-23	(42 51 59 67)	(50 64 80 99)
1-5-11-17-21-23	(38 49 58 65)	(57 74 87 102)
1-5-12-14-21-23	(53 62 70 77)	(46 60 74 90)
1-5-12-15-18-22-23	(44 54 63 73)	(48 63 85 106)
1-5-8-13-19-22-23	(52 61 70 80)	(53 67 88 107)

Table 17. Results of Algorithm 5.5 Based on both Objectives with Natural Preference

Path	Value of the first objective	Value of the second objective
1-5-11-17-21-23	(38 49 58 65)	(57 74 87 102)
1-5-11-17-20-23	(40 51 60 66)	(42 55 68 85)
1-4-11-17-21-23	(38 51 61 68)	(68 87 101 112)
1-5-12-15-18-23	(42 51 59 67)	(50 64 80 99)
1-4-11-17-20-23	(40 53 63 69)	(53 68 82 95)
1-5-11-14-21-23	(45 54 63 69)	(55 67 82 98)
1-5-12-15-18-22-23	(44 54 63 73)	(48 63 85 106)
1-5-12-14-21-23	(53 62 70 77)	(46 60 74 90)
1-5-8-13-15-19-22-23	(50 62 75 87)	(54 68 95 120)
1-3-8-13-15-19-22-23	(56 67 84 96)	(56 69 94 115)

Table 18. Results of Algorithm 5.7 Based on the First Objective Preference

Path	Value of the first objective	Value of the second objective
1-5-11-17-20-23	(40 51 60 66)	(42 55 68 85)
1-5-12-14-21-23	(53 62 70 77)	(46 60 74 90)
1-5-12-15-18-23	(42 51 59 67)	(50 64 80 99)
1-4-11-17-20-23	(40 53 63 69)	(53 68 82 95)
1-5-11-14-21-23	(45 54 63 69)	(55 67 82 98)
1-5-12-15-18-22-23	(44 54 63 73)	(48 63 85 106)
1-5-11-17-21-23	(38 49 58 65)	(57 74 87 102)
1-3-8-13-15-19-22-23	(56 67 84 96)	(56 69 94 115)
1-5-8-13-15-19-22-23	(50 62 75 87)	(54 68 95 120)
1-4-11-17-21-23	(40 53 63 69)	(53 68 82 95)

Table 19. Results of Algorithm 5.7 Based on the Second Objective Preference

Path	Value of the first objective	Value of the second objective
1-5-11-17-20-23	(40 51 60 66)	(42 55 68 85)
1-5-12-15-18-23	(42 51 59 67)	(50 64 80 99)
1-4-11-17-20-23	(40 53 63 69)	(53 68 82 95)
1-5-11-17-21-23	(38 49 58 65)	(57 74 87 102)
1-5-12-14-21-23	(53 62 70 77)	(46 60 74 90)
1-5-11-14-21-23	(45 54 63 69)	(55 67 82 98)
1-5-12-15-18-22-23	(44 54 63 73)	(48 63 85 106)
1-4-11-17-21-23	(40 53 63 69)	(53 68 82 95)
1-5-8-13-15-19-22-23	(50 62 75 87)	(54 68 95 120)
1-3-8-13-15-19-22-23	(50 62 75 87)	(54 68 95 120)

Table 20. Results of Algorithm 5.7 Based on Both Objectives with Natural Preference

Objective Preference	Path	Value of the first objective	Value of the second objective
First	1-5-11-17-21-23	(65 58 49 38)	(57 74 87 102)
Second	1-5-11-17-20-23	(66 60 51 40)	(85 68 55 42)
Both (natural)	1-5-11-17-20-23	(66 60 51 40)	(85 68 55 42)

Table 21. Results of Algorithm 6.3

Nevertheless, using the distance method may produce more dominated and less nondominated paths.

We observe that the first ranked solutions of Algorithms 5.5 and 5.7 are the same as the solution obtained by Algorithm 6.3.

	Algorithm 5.5	Algorithm 5.7
Shortest path based on the first objective preference	1-5-11-17-21-23	1-5-11-17-21-23
	1-5-11-17-20-23	1-5-11-17-20-23
Shortest path based on the second objective preference	1-5-11-17-20-23	1-5-11-17-20-23
	1-5-12-14-21-23	1-5-12-14-21-23
Shortest path based on both objectives with natural preference	1-5-11-17-20-23	1-5-11-17-20-23
	1-5-12-15-18-23	1-5-12-15-18-23

Table 22. Results Obtained by Algorithms 5.5 and 5.7

	Algorithm 5.5	Algorithm 5.7	Algorithm 6.3
Shortest path based on the first objective preference	1-5-11-17-21-23	1-5-11-17-21-23	1-5-11-17-21-23
Shortest path based on the second objective preference	1-5-11-17-20-23	1-5-11-17-20-23	1-5-11-17-20-23
Shortest path based on both objectives with natural preference	1-5-11-17-20-23	1-5-11-17-20-23	1-5-11-17-20-23

Table 23. Results Obtained by Algorithms 5.5, 5.7 and 6.3

8. **Conclusions**

We considered the biobjective shortest path problem (BSP) in fuzzy networks and developed three algorithms for solving this problem. We presented two labeling type algorithms for finding nondominated paths from a specified node to every other node. Since the number of nondominated paths may be large, choosing a preferred path may be required. Thus, we utilized a distance function for comparison of fuzzy numbers in a fuzzy ranking method to rank the obtained nondominated paths according to the first objective, the second objective or both. We then presented a procedure for computing the nondominated paths requiring less computing time. The third algorithm was based on a dynamic programming approach. For this algorithm, we presented an approach for converting the two fuzzy numbers corresponding to the two objectives into a single number for comparison purposes. The dynamic programming algorithm for the single objective case and the distance method for ranking fuzzy numbers were adaptively used in the proposed algorithm. The algorithm computes one shortest path, as an alternative solution to several nondominated paths. Examples were worked out to show the effectiveness of the proposed algorithms.

References

[1] S. M. Baas and H. Kwakernaak, *Rating and ranking of multiple-aspect alternatives using fuzzy sets*, Automatica, **13(1)** (1977), 47-58.

[2] J. F. Baldwin and N. C. F Guild, *Comparison of fuzzy sets on the same decision space*, Fuzzy Sets and Systems, **2(3)** (1979), 213-231.

[3] R. Bellman, *On a routing problem, Quart*, J. Appl. Math, **16(1)** (1958), 87-90.

[4] G. Bortolan and R. Degani, *A review of some methods for ranking fuzzy subsets*, Fuzzy Sets and Systems, **15(1)** (1985), 1-19.

[5] D. Brumbaugh-Smith and D. Shier, *An empirical investigation of some bicriterion shortest path algorithms*, European Journal of Operational Research, **43(2)** (1989), 216-224.

[6] L. Campos and A. Munoz, *A subjective approach for ranking fuzzy numbers*, Fuzzy Sets and Systems, **29(12)** (1989), 145-153.

[7] S. Chanas, M. Delgado, J. L. Verdegay, and M. A. Vila, *Fuzzy optimal flow on imprecise structures*, European Journal of Operational Research, **83(3)** (1995), 568-580.

[8] W. Chang, *Ranking of fuzzy utilities with triangular membership functions*, In Proc. Int. Conf. of Policy Anal. and Inf. Systems, (1981), 263-272.

[9] S. H. Chen, *Ranking fuzzy numbers with maximizing set and minimizing set*, Fuzzy Sets and Systems, **17(2)** (1985), 113-129.

[10] F. Choobineh and H. Li, *An index for ordering fuzzy numbers*, Fuzzy Sets and Systems, **54(3)** (1993), 287-294.

[11] M. Delgado, J. L. Verdegay and M. A. Vila, *On valuation and optimization problems in fuzzy graphs: a general approach and some particular cases*, INFORMS Journal on Computing, **2(1)** (1990), 74.

[12] E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numerische mathematik, **1(1)** (1959), 269-271.

[13] S. E. Dreyfus, *An appraisal of some shortest-path algorithms*, Operations Research, **17(3)** (1969), 395-412.

[14] D. Dubois and H. Prade, *Algorithmes de plus courts chemins pour traiter des donnees floues. RAIRO-Recherche Opérationnelle*, Operations Research, **12**(1978), 212–227.

[15] D. Dubois and H. Prade, *Fuzzy sets and systems: theory and applications*, Academic Pr, 1980.

[16] N. Furukawa, *A parametric total order on fuzzy numbers and a fuzzy shortest route problem*, Optimization, **30(4)** (1994), 367-377.

[17] P. Hansen, *Bicriterion path problems*, In Multiple criteria decision making: theory and application: proceedings of the third conference, Hagen/Kèonigswinter, West Germany, August 20-24, (1979), 109, Springer, 1980.

[18] R.V. Helgason and J. L. Kennington, *Primal simplex algorithms for minimum cost network flows*, Handbooks in Operations Research and Management Science, **7** (1995), 85-133.

[19] F. Huarng, P. Pulat, and L. S. Shih, *A computational comparison of some bicriterion shortest path algorithms*, Journal of the Chinese Institute of Industrial Engineers, **13(2)** (1996), 121-125.

[20] C. M. Klein, *Fuzzy shortest paths*, Fuzzy Sets and Systems, **39(1)** (1991), 27-41.

[21] L. T. Kóczy, *Fuzzy graphs in the evaluation and optimization of networks*, Fuzzy Sets and Systems, **46(3)** (1992), 307-319.

[22] K. C. Lin and M. S. Chern, *The fuzzy shortest path problem and its most vital arcs*, Fuzzy Sets and Systems, **58(3)** (1993), 343-353.
[23] T. S. Liou and M. J. J. Wang, *Ranking fuzzy numbers with integral value*, Fuzzy Sets and Systems, **50(3)** (1992), 247-255.
[24] I. Mahdavi, R. Nourifar, A. Heidarzade and N. M. Amiri, *A dynamic programming approach for finding shortest chains in a fuzzy network*, Applied Soft Computing, **9(2)** (2009), 503-511.
[25] E. Q. V. Martins, *On a multicriteria shortest path problem*, European Journal of Operational Research, **16(2)** (1984), 236-245.
[26] J. Mote, I.Murthy and D. L. Olson *A parametric approach to solving bicriterion shortest path problems*, European Journal of Operational Research, **53(1)** (1991), 81-92.
[27] J. C. Namorado Climaco and E. Queiros Vieira Martins, *A bicriterion shortest path algorithm*, European Journal of Operational Research, **11(4)** (1982), 399-404.
[28] S. Okada and T. Soper, *A shortest path problem on a network with fuzzy arc lengths*, Fuzzy Sets and Systems, **109(1)** (2000), 129-140.
[29] H. Prade, *Using fuzzy set theory in a scheduling problem: a case study*, Fuzzy Sets and Systems, **2(2)** (1979), 153-165.
[30] A. Przybylski, X. Gandibleux and M. Ehrgott, *Two phase algorithms for the bi-objective assignment problem*, European Journal of Operational Research, **185(2)** (2008), 509-533.
[31] J. Ramík and J. Rimanek, *Inequality relation between fuzzy numbers and its use in fuzzy optimization*, Fuzzy Sets and Systems, **16(2)** (1985), 123-138.
[32] J. J. Saade and H. Schwarzlander, *Ordering fuzzy sets over the real line: an approach based on decision making under uncertainty*, Fuzzy Sets and Systems, **50(3)** (1992), 237-246.
[33] B. Sadeghpour Gildeh and D. Gien, *La distance-Dp, q et le cofficient de corrélation entre deux variables aléatoires floues*, Actes de LFA, (2001), 97-102.
[34] P. Serafini, *Some considerations about computational complexity for multi objective combinatorial problems*, In Recent Advances and Historical Development of Vector Optimization: Proceedings of an International Conference of Vector Optimization Held at the Technical University of Darmstadt, FRG, August 4-7, (1986), 222, Springer, 1987.
[35] A. J. V. Skriver and K. A. Andersen, *A label correcting approach for solving bicriterion shortest-path problems*, Computers & Operations Research, **27(6)** (2000), 507-524.
[36] A. J. V. Skriver, *A classification of bicriterion shortest path (BSP) algorithms*, Asia Pacific Journal of Operational Research, **17(2)** (2000), 199-212.
[37] A. Tajdin, I. Mahdavi, N. Mahdavi-Amiri and B. Sadeghpour-Gildeh, *Computing a fuzzy shortest path in a network with mixed fuzzy arc lengths using α -cuts*, Computer and Mathematics with Applications, 2010.
[38] C. T. Tung and K. L. Chew, *A bicriterion Pareto-optimal path algorithm*, ASIA-PACIFIC J. OPER. RES., **5(2)** (1988), 166-172.
[39] C. Tung and K. Lin Chew, *A multicriteria Pareto-optimal path algorithm*, European Journal of Operational Research, **62(2)** (1992), 203-209.
[40] X. Wang, *A comparative study of the ranking methods for fuzzy quantities*, Ghent University, Ghent, 1997.
[41] X. Wang and E. E. Kerre, *Reasonable properties for the ordering of fuzzy quantities (I)*, Fuzzy Sets and Systems, **118(3)** (2001), 375-385.
[42] X. Wang and E. E. Kerre, *Reasonable properties for the ordering of fuzzy quantities (II)*, Fuzzy Sets and Systems, **118(3)** (2001), 387-405.
[43] R. A. Yager, *On a general class of fuzzy connectives*, Fuzzy sets and Systems, **4(3)** (1980), 235-242.
[44] R. A. Yager, *Paths of least resistance in possibilistic production systems*, Fuzzy Sets and Systems, **19(2)** (1986), 121-132.
[45] L. A. Zadeh, *Fuzzy sets*, Information and control, **8(3)** (1965), 338-353.

Iraj Mahdavi, Department of Industrial Engineering, Mazandaran University of Science & Technology, Babol, Iran
E-mail address: `irajarash@rediffmail.com`

Nezam Mahdavi-Amiri, Faculty of Mathematical Sciences, Sharif University of Technology, Tehran, Iran
E-mail address: `nezamm@sina.sharif.edu`

Shahrbanoo Nejati*, Department of Industrial Engineering, Mazandaran University of Science & Technology, Babol, Iran
E-mail address: `nejati_sh@yahoo.com`

*Corresponding author