RAIRO-Oper. Res. 55 (2021) S1767–S1787
https://doi.org/10.1051/ro/2020033

RAIRO Operations Research
www.rairo-ro.org

AN ACCEPTABILITY INDEX BASED APPROACH FOR SOLVING SHORTEST PATH PROBLEM ON A NETWORK WITH INTERVAL WEIGHTS

Ali Ebrahimnejad*

Abstract. Based on the acceptability index for comparison of any two imprecise values, efficient algorithms have been proposed in the literature for solving shortest path (SP) problem when the weights of connected arcs in a transportation network are represented as interval numbers. In this study, a generalized Dijkstra algorithm is proposed to handle the SP problem with interval weights. Here it is shown that once the acceptability index is chosen, the interval SP problem is converted into crisp one, which is easily solved by the standard SP algorithms. The main contribution here is the reduction of the computational complexity of the existing algorithm for solving interval SP problem. To show the advantages of the proposed algorithm over existing algorithm the numerical example presented in literature is solved using the proposed algorithm and the obtained results are discussed. Moreover, an small sized telecommunication network is provided to illustrate the potential application of the proposed method. Finally, the practical relevance of the proposed algorithm is evaluated by means of a large scale pilot case where a pharmaceutical shipment between the cities in Iran should be transported.

Mathematics Subject Classification. 90Bxx, 90B18, 90C70.

Received May 7, 2019. Accepted March 26, 2020.

1. Introduction

Finding shortest path is among one of the most fundamental optimization problems in network theory. Although this problem is one of the simple network problems, it has a wide range of applications. The aim of shortest path (SP) problem is to find a path with minimum weight (cost, time or length) between two specified nodes. Even though SP problems are relatively easy to solve, the design and analysis of most algorithms for solving them require considerable ingenuity [4]. The deterministic SP problem has been widely studied in the literature as (1) it can be used for a wide variety of situations such as transportation, routing, communications, supply chain management and models involving agents [8,28]; and (2) it arises frequently as sub-problems when solving many combinatorial and network optimization problems.

In traditional networks, the weights are deterministic. However, because of failure, maintenance, or other reasons, the weights are non-deterministic in many situations. Transshipment time and transportation cost are examples of data that are uncertain. In fact, since the time and cost fluctuate depending on traffic conditions, payload and so on, imprecise or uncertain data of some kind are to be applied as weights of arcs. Some researchers

Keywords. Shortest path problem, interval numbers, acceptability index.

Department of Mathematics, Qaemshahr Branch, Islamic Azad University, Qaemshahr, Iran.
*Corresponding author: a.ebrahimnejad@qaemiau.ac.ir, aemarzoun@gmail.com

Article published by EDP Sciences

believed that these non-deterministic phenomena conform to interval numbers, so they introduced interval theory into network optimization problems and used interval variables to describe the non-deterministic weights.

The problem of finding the shortest path in a network with interval weights is known as the interval SP problem which was analyzed by Nayeem and Pal [46]. They solved this problem by considering the acceptance index introduced by Sengupta and Pal [55] for comparing the interval weights of paths. In this paper, it is motivated to develop a simplified algorithm for solving the interval SP problems. Hereafter the existing algorithm and the proposed algorithm are named Nayeem–Pal algorithm and SP-IW (shortest path with interval weights) algorithm, respectively.

Since the SP problem is essentially a minimum cost network flow problem, using existing interval minimum cost network flow problem techniques [24, 25, 29, 54] is a straightforward approach to the interval one. However, its frequent occurrence in practice and the specialized, more efficient procedures that can be developed for handling this problem provide a strong case for considering them separately.

Generally speaking, one main issue that needs to be solved when applying the interval SP algorithms is the ranking and comparison of interval numbers. To solve this problem, in this paper a simplified new algorithm is presented when the weights in a transportation network are represented as interval numbers. It is shown that the interval SP problem can be solved simply by solving an equivalent crisp one.

This rest of the paper is organized as follows. The relevant stochastic, fuzzy and interval SP problems literature are reviewed in Section 2. In Section 3, some necessary concepts and backgrounds on interval arithmetic are given. In Section 4, first the Nayeem–Pal algorithm for solving interval SP problem is stated and then SP-IW algorithm is presented to find the interval path with lowest weights. In Section 5, two examples are given to demonstrate the SP-IW algorithm. In Section 6, the practical relevance of the SP-IW algorithm is evaluated by means of a case study for transportation of a pharmaceutical shipment in Iran. Finally, our conclusions and future research directions are presented in Section 7.

2. Literature review

Shortest path problem with uncertainty weights is one of the important research topics from both theoretical and practical aspects. Several researchers have contributed significantly in this area of research. In this section a comprehensive literature review on solving SP problem with uncertain data is presented.

Fuzzy and stochastic approaches are frequently used to describe and treat imprecise and uncertain weights in SP problem. Such approaches have received little attention although the deterministic version of the problem has been studied extensively.

The probabilistic analysis is the most widely used method for characterizing uncertainty arc weights in SP problem. As we know, Croucher [13] is the first to develop an algorithm for solving a "shortest route" network problem in order to find the path which yields the shortest expected distance through the network. Sigal *et al.* [58] introduces the concept of a path optimality index as a performance measure for selecting a path in stochastic networks. They define the path optimality index as the probability a given path is shorter than all other paths. Kamburowski [36] proposes an optimality index for choosing the shortest route in an acyclic stochastic network. He associates a random variable given by a known probability distribution with each arc and assumes all random variable are independent. Polychronopoulos and Tsitliklis [53] uses dynamic algorithms for solving SP problem defined on a network with random costs. They assume that information on cost values is accumulated as network is being traversed. Murthy and Sarkar [44] consider a form of the stochastic SP problem where the optimal path is given by maximizing the concave and quadratic expected utility. The principal contribution of that paper is the development of a relaxation based pruning technique which is incorporated into a label setting procedure. Also, Murthy and Sarkar [45] consider a stochastic SP problem where the lengths are independent random variables following a normal distribution. Alexopoulos [5] gives methods based on an iterative partition of the network state space for computing measures related to shortest paths in networks with discrete random lengths. These measures are included the probability that there exists a path with length not exceeding a specified value and the probability that a given path is shortest. Cheung [10] considers a routing policy that forms a dynamic shortest path in a network with independent, positive and discrete random costs. He proposes an approach, which mimics the classical label-correcting approach, to compute the expected path cost. Bander and White [6] present a best-first heuristic search approach for determining an optimal policy for a stochastic SP problem. Ji [34] solves the SP problem with stochastic length. According to different decision criteria, he originally proposes the concepts of expected shortest path, α-shortest path and the most shortest path, and presents three new types of models: expected value model, chance-constrained programming and dependent-chance programming. Then he develops a hybrid intelligent algorithm integrating stochastic simulation and genetic algorithm in order to solve these models. Nikolova *et al.* [47] considers the problem of finding SP in a network with independent randomly distributed lengths. Beigy and Meybodi [7] first introduce a network of learning automata called as distributed learning automata and then propose some iterative algorithms for solving stochastic SP problem. Their algorithms use distributed learning automata to find a policy that determines a path from a source node to a destination node with minimal expected cost (length). Peer and Sharma [52] present a new methodology for computing shortest expected lengths in incomplete stochastic networks by utilizing linear programming. Ohtsubo [48] considers a stochastic SP problem with associative criteria in which for each node of a graph a probability distribution is chosen over the set of successor nodes so as to reach a given target node optimally. Hutson and Shier [33] studies the problem of finding a path in the context of stochastic networks that combines the mean and variance of its length. Horoba [32] consider ant colony optimization for stochastic SP problems where weights are subject to noise that reflects delays and uncertainty. Yu and Bertsekas [60] consider a totally asynchronous stochastic approximation algorithm, Q-learning, for solving finite space stochastic SP problems, which are undiscounted, total cost Markov decision processes with an absorbing and cost-free state.

Guo *et al.* [27] propose hierarchical structured learning automata to determine the SP from source to destination when a stochastic network is given, whose weights are dynamically changed and the probability distribution of the constantly changing lengths is unknown beforehand. Guillot and Stauffer [26] extend Dijkstra's algorithm for the stochastic SP problem in finite state and action spaces. Zhang *et al.* [62] study the issue of border security in terms of resource allocation, and formed the problem as stochastic SP network interdiction models. Ahmadi *et al.* [3] propose genetic algorithm to deal with the stochastic SP problem under probable weather conditions. Sever *et al.* [57] study a dynamic SP problem with travel time-dependent stochastic disruptions and proposed a hybrid approximate dynamic programming with a deterministic look ahead policy and value function approximation in order to deal with the complexity of the problem.

Similarly, fuzzy theory can be used to uncertainty analysis with imprecise weights in SP problem. As we know, Dubois and Prade [17] are the first to analyze the fuzzy SP (FSP) problem and to consider extensions of the classic Floyd and Ford–Moore–Bellman (FMB) algorithms that return distances without an associated path. Klein [37] introduces new models based on fuzzy shortest paths and also proposes a general algorithm based on dynamic programming to solve the new models. In addition the SFP algorithms are analyzed in terms of sub-modular functions in that paper. Lin and Chern [40] consider the case that the weights are fuzzy numbers and propose an algorithm for finding the single most vital arc in a network as being that whose removal from the path results in an increase in cost. Li *et al.* [39] propose the neural networks for solving SP problems. Okada and Super [51] concentrate on the SP problem in a network with fuzzy weights. Then they propose an algorithm to obtain all Pareto Optimal paths from the specified node to every other node by introducing a concept of Pareto Optimal path based on an order relation between fuzzy numbers. Following the idea of finding a fuzzy set solution, Okada [49] presents an algorithm to determine the degree of possibility for each arc on the shortest path. Chuang and Kung [11] propose a heuristic procedure to find the FSP length among all possible paths in a network. Chuang and Kung [12] propose a new algorithm that gives the FSP length and the corresponding shortest path in a discrete FSP problem. Hernandes *et al.* [31] consider a generic algorithm for solving FSP problem that can be implemented using any fuzzy numbers ranking index chosen by the decision-maker. Ji *et al.* [35] introduce three types of models for FSP problem based on the concepts of expected shortest path, α-shortest path and the most shortest path in fuzzy environment. They also propose a hybrid intelligent algorithm integrating simulation and genetic algorithm in order to solve these models. Gao [23] proves that there exists an equivalence relation between the α-shortest path of an uncertain network and the shortest path of the corresponding deterministic network. Kumar and Kaur [38] present a new algorithm for solving SP problem on a network with imprecise weights. Deng *et al.* [14] propose a method to find the shortest path under fuzzy lengths based on the graded mean integration representation of fuzzy numbers. Dou *et al.* [16] apply an approach to select the SP in multi-constrained network using multi-criteria decision method based on vague similarity measure. Mahdavi *et al.* [41] propose a dynamic programming approach to solve the fuzzy shortest chain problem using a suitable ranking method. Tajdin *et al.* [59] design an algorithm for computing the SP in a network having various types of fuzzy lengths. Moreover, Hassanzadeh *et al.* [30] present a genetic algorithm for finding the SP in the network due to the complexity of the addition of various fuzzy numbers for larger problems. Ebrahimnejad *et al.* [18, 19] use particle swarm optimization algorithm and artificial bee colony algorithm to approximate the SP on a network with different types of fuzzy numbers. Dey *et al.* [15] propose an algorithmic approach based on genetic algorithm for solving SP problem in a fuzzy network involving type-2 fuzzy weights. Eshaghnezhad *et al.* [22] uses the KKT optimality conditions to propose an artificial neural network model as a high-performance tool to provide the solution of FSP problem. Motameni and Ebrahimnejad [43] formulate a constrained SP problem in a directed network where the weights represented by cost and time are intuitionistic trapezoidal fuzzy numbers and then develop an approach for solving the resulting problem. Mani *et al.* [42] propose an algorithm for finding shortest path and shortest weight on a hesitancy fuzzy network. Zero *et al.* [61] propose two approaches for the bi-objective SP problem by introducing fuzzy logic to model the risk exposure in solving hazmat routing selection problem. Enayattabar *et al.* [20] formulate the SP problem in an interval-valued Pythagorean fuzzy environment and developed the traditional Dijkstra algorithm to find the cost of interval-valued Pythagorean fuzzy SP. Enayattabar *et al.* [21] investigate a generalized kind of the SP problem under interval-valued fuzzy environment namely all pairs shortest path (APSP) problem. Broumi *et al.* [9] consider SP problem through Bellman's algorithm for a network using interval-valued neutrosophic numbers and propose a novel algorithm to obtain the neutrosophic shortest path between each pair of nodes. Abbaszadeh Sori *et al.* [1] propose elite artificial bees colony algorithm to solve robot fuzzy constrained shortest route problem. Moreover, Abbaszadeh Sori *et al.* [2] propose a fuzzy inference approach for solving multi objective constrained SP.

However, in literature in stochastic optimization, parameters are assumed to be with known probability distributions and in the works of fuzzy optimization, fuzzy parameters are assumed to have known membership functions. But, it is not always possible for a decision maker to determine the probability distribution or membership function in an inexact environment. At least in some of the cases, the use of interval coefficients may serve the purpose better. Okada and Gen [50] discuss the problem of finding the shortest paths from a fixed origin to a specified node in a network with interval weights. Sengupta and Pal [56] define a solution procedure for a SP problem in which the network arcs are represented as interval numbers. They generalize the Dijkstra's algorithm to incorporate interval parameters as data and considered fuzzy preference ordering of intervals from pessimistic and optimistic decision maker's point of view. Nayeem and Pal [46] propose an algorithm based on the acceptance index introduced by Sengupta and Pal [55]. Their algorithm gives a single interval shortest path or a guideline for choosing the best interval shortest path according to the decision maker's viewpoint. However one key issue need to be addressed in that algorithm is how to compare the distance between two different paths with interval weights. Here it is shown that once the acceptability index is chosen, the interval SP problem is converted into crisp one, which is easily solved by the standard SP algorithms.

3. Preliminaries

In this section, some necessary definitions, additions of interval numbers and comparison methods of such numbers are recalled which will be used in the rest of paper [46].

3.1. Interval numbers

Definition 3.1. An interval number is defined as follows

$$A = [a_L, a_R] = \{t \,|a_L \leq t \leq a_R\} \tag{3.1}$$

where a_L and a_R are the real numbers called the left end point and the right end point of the interval number A.

In alternative notation, an interval number can be defined in terms of midpoint and width as follows

$$A = \langle m(A), w(A) \rangle = \{t \,|m(A) - w(A) \leq t \leq m(A) + w(A)\,\} \tag{3.2}$$

where $m(A) = \frac{a_L + a_R}{2}$ and $w(A) = \frac{a_R - a_L}{2}$ are the mid-point and half-width or, width (henceforth referred to as width) of the interval number A, respectively.

Example 3.2. Let $A = [a_L, a_R] = [20, 24]$ and $B = [b_L, b_R] = [30, 36]$. Then these interval numbers in terms of midpoint and width are represented as $A = \langle m(A), w(A)\rangle = \langle 22, 2\rangle$ and $B = \langle m(B), w(B)\rangle = \langle 33, 3\rangle$, respectively.

Definition 3.3. Two interval numbers $A = \langle m(A), w(A)\rangle$ and $B = \langle m(B), w(B)\rangle$ are said to be non-dominating if $m(A) = m(B)$ and $w(A) \neq w(B)$.

Example 3.4. Let $A = \langle m(A), w(A)\rangle = \langle 22, 2\rangle$ and $B = \langle m(B), w(B)\rangle = \langle 22, 3\rangle$. Then these two interval numbers are non-dominated since $m(A) = m(B) = 22$ and $w(A) = 2 \neq w(B) = 3$.

Definition 3.5. The addition of two interval numbers $A = [a_L, a_R]$ and $B = [b_L, b_R]$ is given by

$$A \oplus B = [a_L + b_L, a_R + b_R]\,. \tag{3.3}$$

Example 3.6. Let $A = [a_L, a_R] = [20, 24]$ and $B = [b_L, b_R] = [30, 36]$. Then$A \oplus B = [50, 60]$.

Alternately, in mean-width notations, if $A = \langle m(A), w(A)\rangle$ and $B = \langle m(B), w(B)\rangle$ then,

$$A \oplus B = \langle m(A) + m(B), w(A) + w(B)\rangle. \tag{3.4}$$

Example 3.7. Let $A = \langle m(A), w(A)\rangle = \langle 22, 2\rangle$ and $B = \langle m(B), w(B)\rangle = \langle 33, 3\rangle$. Then $A \oplus B = \langle 55, 5\rangle$.

Comparison between two interval numbers is very important in interval arithmetic. Sengupta and Pal [55] define an acceptability index ($\mathcal{A}$-index) for comparing the values of any two interval numbers. The $\mathcal{A}$-index defines how much higher one interval is than another in terms of the interpreter's level of satisfaction. Nayeem and Pal [46] use this index for solving shortest path problem.

Definition 3.8. The $\mathcal{A}$-index to proposition "A is inferior to B" is defined as follows:

$$\mathcal{A}(A \prec B) = \frac{m(B) - m(A)}{w(A) + w(B)}\cdot \tag{3.5}$$

In connection with this acceptability index, Nayeem and Pal [46] define the total dominance and partial dominance of two interval numbers $A = \langle m(A), w(A)\rangle$ and $B = \langle m(B), w(B)\rangle$ one over another as follow:

Definition 3.9. If $\mathcal{A}(A \prec B) \geq 1$ then, A is said to be totally dominating over B in the sense of minimization and B is said to be totally dominating over A in the sense of maximization. This concept is denoted by $A \prec B$, *i.e.*, $\min\{A, B\} = B$.

Definition 3.10. If $0 < \mathcal{A}(A \prec B) < 1$ then, A is said to be partially dominating over B in the sense of minimization and B is said to be partially dominating over A in the sense of maximization. This concept is denoted by $A \prec_P B$, *i.e.*, $\min\{A, B\} = A$.

But when, $\mathcal{A}(A \prec B) = 0$, *i.e.*, $m(B) - m(A) = 0$ then it is not possible to get an order relation from the above cases. Then it may emphasize on the widths of the interval numbers A and B.

If $w(A) < w(B)$ then the left end point of A is less than that of B and on finding a minimum distance, there is a chance that the distance may lie on A. But at the same time, since the right end point of A is greater than that of B, if one prefers A to B in minimization then in worst case, he may be looser than one who prefers B to A. Thus in such a situation an optimistic decision-maker would prefer A to B whereas a pessimistic decision-maker would do the converse.

Example 3.11. Let $A = \langle m(A), w(A)\rangle = \langle 22, 2\rangle$ and $B = \langle m(B), w(B)\rangle = \langle 33, 3\rangle$. Then $\mathcal{A}(A \prec B) = \frac{33-22}{5} = \frac{11}{5} = 2.2 > 1$. So regarding Definition 3.9 in minimization, A is totally dominating over B.

Example 3.12. Let $A = \langle m(A), w(A)\rangle = \langle 22, 2\rangle$ and $B = \langle m(B), w(B)\rangle = \langle 26, 3\rangle$. Then $\mathcal{A}(A \prec B) = \frac{26-22}{5} = \frac{4}{5} = 0.8 < 1$. So regarding Definition 3.10 in minimization, A is partially dominating over B.

Example 3.13. Let $A = \langle m(A), w(A)\rangle = \langle 22, 3\rangle$ and $B = \langle m(B), w(B)\rangle = \langle 22, 3\rangle$. Then $m(A) = m(B) = 22$ and $w(A) = w(B) = 3$. So an optimistic decision-maker would prefer A to B whereas a pessimistic decision-maker would do the converse.

4. Interval shortest path problem

In this section, first the Nayeem–Pal algorithm is described for finding the imprecise length of shortest path between different nodes and then SP-IW algorithm is explored for the same problem.

4.1. The Nayeem–Pal algorithm

Consider a connected network $G = (N, E)$, where N is the set of nodes ($|N| = r$) and E is the set of arcs. Each arc is denoted by an ordered pair (i, j), where $i, j \in E$. The network has two distinguished nodes 1 and t, called the source node and the destination node, respectively. It is supposed that there is only one arc (i, j) from node i to node j. A path p_{ij} from node i to node j is a sequence of arcs $p_{ij} = \{(i, i_1), (i, i_2), \ldots, (i_k, j)\}$ in which the initial node of each arc is same as the terminal node of preceding arc in the sequence. The length (weight) of a path is defined as the sum of the lengths (weights) of arcs in the path. A shortest path problem is to determine for every node $i \in N - \{1\}$, a shortest length path from source node 1 to node i.

The problem of finding the shortest path form source node to every other node in a network with interval weights is called interval SP problem. Let $C_{ij} = [c_{ij,L}, c_{ij,R}]$ denotes an interval number associated with the arc (i, j) corresponding to the length necessary to traverse (i, j) from node i to node j. In this case, the mathematical model of the interval shortest path problem is formulated as follows:

$$\begin{aligned}
&\min \sum_{i=1}^{m}\sum_{j=1}^{m} [c_{ij,L}, c_{ij,R}]x_{ij} = \left[\sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij,L}x_{ij}, \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij,R}x_{ij}\right] \\
&\text{s.t.} \\
&\qquad \sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = \begin{cases} 1, & i = 1 \\ 0, & i \neq 1, m \\ -1, & i = m \end{cases} \\
&\qquad x_{ij} \geq 0, i, j = 1, 2, \ldots, m.
\end{aligned}$$

The variable x_{ij} is the decision variable of the arc (i, j). Its takes is one if the arc (i, j) is a part of the shortest path in the solution of the problem. Otherwise, its value is zero.

Define

$$D_{ij} = \begin{cases} [0, 0], & i = j, \\ [\infty, \infty], & (i, j) \notin V, \\ [c_{ij,L}, c_{ij,R}], & (i, j) \in V. \end{cases} \tag{4.1}$$

Nayeem and Pal [46] develop an algorithm based on Dijkstra's (1959) algorithm with the concept of multi-labelling process. At every step, the temporary label of a node is obtained as a single number or more than one with same mean. Then the minimum among the temporary labels is determined and it is called as permanent label. In the case of multiple labels of a node as permanent label, then the method is proceeded for other remaining nodes by taking each of the non-dominating multiple labels. Finally, the permanent label of the node is the length of the shortest paths from the source node to that node. This algorithm is named Nayeem–Pal algorithm.

The general steps of the Nayeem–Pal algorithm to find the interval shortest distances from the source node 1 to all other nodes are as follows:

Step 1: Set the permanent label of node 1 as $\langle 0,0\rangle$ and temporary label of every other node as $\langle\infty,0\rangle$. Denote qth Permanent label of node i, kth Temporary Label of the node i and the last node got permanent label by $\mathrm{PL}_q(i)$, $\mathrm{TL}_k(i)$ and LN, respectively. Set $q := 0$, $\mathrm{PL}_q(1) := \langle 0,0\rangle$, $\mathrm{LN} = 1$, $k := 0$ and $N^* := \{1\}$, $\mathrm{TL}_k(j) = \langle\infty,0\rangle$ for all $j \in N - N^*$.

Step 2: while $N \neq N^*$ do

Step 2.1: for $j \in N - N^*$ do

$$\langle m(L_j), w(L_j)\rangle := \langle m(D_{LNj}), w(D_{LNj})\rangle \oplus \mathrm{PL}_q(\mathrm{LN})$$

If $\langle m(L_j), w(L_j)\rangle$ and $\mathrm{TL}_k(j)$ are non-dominating then $\mathrm{TL}_{k+1}(j) = \langle m(L_j), w(L_j)\rangle$;
Else, based on the acceptability index find the minimum between $\mathrm{TL}_k(i)$ and $\langle m(D_{LNj}), w(D_{LNj})\rangle \oplus \mathrm{PL}_q(\mathrm{LN})$. Let $\mathrm{TL}_k(j) = \min\{\mathrm{TL}_k(j), \langle m(L_j), w(L_j)\rangle\}$.

Step 2.2: If $\mathrm{TL}_q(i)$ totally dominating or partially dominating over all other temporary labels of all other nodes of $N - N^*$ then, $\mathrm{LN} := i$, $\mathrm{PL}_q(\mathrm{LN}) := \mathrm{TL}_q(i)$, $N^* := N^* \cup \{i\}$ and $q := q + 1$.

Step 2.3: But if there are more than one temporary label (say, r non-dominating temporary labels) of the node i then, we split-up the process into r branches and do the Step 2.2 for each of the r branches and then continue.

Step 3: The permanent labels $\mathrm{PL}_k(j)$ of the node j are the final shortest distances from the node 1 to the node j for all $j \in N$, where k represents the number of possible interval shortest distances from source node 1 to j with same mean but different widths.

In Nayyem–Pal algorithm all arithmetic operations are performed on interval numbers. In addition interval numbers are compared in all iterations of this algorithm. In the following section, it is shown that the same solution of the interval shortest path problem can be found with the help of crisp shortest path problem. Hence, all arithmetic operations are done on real numbers instead of interval numbers.

4.2. The SP-IW algorithm

In this subsection, a simplified approach is presented for solving the interval SP problem. It is demonstrated that once the acceptability index is chosen, the interval SP problem is converted into crisp one, which is easily solved by the standard SP algorithms.

Applying Definitions 3.9 and 3.10 in the Sengupta and Pal acceptability index, in order to compassion of two interval numbers $A = \langle m(A), w(A)\rangle$ and $B = \langle m(B), w(B)\rangle$ the following result is obtained.

Lemma 4.1. *For two not non-dominating interval numbers* $A = \langle m(A), w(A)\rangle$ *and* $B = \langle m(B), w(B)\rangle$, $A \prec B$ *or* $A \prec_p B$ *if and only if* $m(A) < m(B)$.

Proof. Applying the acceptability index in equation (3.5) and regarding to Definitions 3.9 and 3.10, it follows that $A \prec B$ or $A \prec_p B$ if and only if $\mathcal{A}(A \prec B) > \mathcal{A}(B \prec A)$. This means that

$$\begin{aligned} \frac{m(B)-m(A)}{w(A)+w(B)} > \frac{m(A)-m(B)}{w(A)+w(B)} &\Leftrightarrow \frac{m(B)-m(A)}{w(A)+w(B)} - \frac{m(A)-m(B)}{w(A)+w(B)} > 0 \\ &\Leftrightarrow \frac{2m(B)-2m(A)}{w(A)+w(B)} > 0 \\ &\Leftrightarrow m(A) < m(B). \end{aligned}$$

Hence, the corollary follows. □

Lemma 4.1 helps us to convert the interval shortest path problem into an equivalent crisp one. To this aim, the module of each interval number is substituted instead of the corresponding interval number in the interval shortest path problem under consideration. This leads to an equivalent crisp one which can be solved by the standard shortest path algorithms. Then, all arithmetic operations are done on the crisp numbers. As a result, the computational effort is decreased significantly in our proposed approach.

Now the optimality conditions in interval networks are stated as follows to design of solution algorithm. To do this, the method given in Lemma 4.1 is used to compare the interval costs between different interval paths.

Theorem 4.2 (Optimality conditions)**.** *For every node $j \in N$, let $S_j = \langle m(S_j), w(S_j)\rangle$ denotes the interval cost of some directed path from the node 1 to node j. Then, interval numbers S_j represent interval SP costs if and only if they satisfy the following optimality conditions:*

$$\mathcal{A}(S_j \prec S_i \oplus C_{ij}) \geq \mathcal{A}(S_i \oplus C_{ij} \prec S_j) \quad \text{for all} \quad (i,j) \in E. \tag{4.2}$$

Proof. If S_j is the interval cost of a shortest path from the source node 1 to node j then it must satisfy the conditions (4.2). Suppose not, *i.e.*, $\mathcal{A}(S_j \prec S_i \oplus C_{ij}) < \mathcal{A}(S_i \oplus C_{ij} \prec S_j)$ for some arc $(i,j) \in E$. By Lemma 4.1, this means that $m(S_j) > m(S_i \oplus C_{ij})$. By assumption S_i is the interval cost of a directed path like p_{1i} from the source node 1 to node i. This path plus the arc (i,j) constructs a new path from the source node 1 to node j with the interval cost $S_i \oplus C_{ij}$. This contradicts the optimality of interval cost S_j.

Now, it is shown in the case that the interval costs S_j satisfy the conditions in (4.2), they represent interval SP costs. To do this, consider any interval cost S_j satisfying (4.2). Let $1 = i_1 - i_2 - \cdots - i_k = j$ be any path p_{1j} from the source node 1 to node j. The conditions (4.2) imply that

$$\begin{aligned} m(S_j) = m(S_{i_k}) &\leq m\left(S_{i_{k-1}} \oplus C_{i_{k-1}i_k}\right), \\ m(S_{i_{k-1}}) &\leq m\left(S_{i_{k-2}} \oplus C_{i_{k-2}i_{k-1}}\right), \\ &\vdots \\ m(S_{i_2}) &\leq m\left(S_{i_1} \oplus C_{i_1 i_2}\right) = m\left(C_{i_1 i_2}\right). \end{aligned}$$

The last equality follows from the fact that $S_{i_1} = S_1 = [0,0]$. Adding the equalities follows

$$m(S_j) = m(S_{i_k}) \leq m(C_{i_{k-1}i_k}) + m(C_{i_{k-2}i_{k-1}}) + \cdots + m(C_{i_1 i_2}) = \sum_{(i,j)\in p_{1j}} m(C_{ij})$$

Thus S_j is a lower bound on the interval cost of any path from the node 1 to node j. On the other hand, since S_j is the interval cost of some path from the source node 1 to node j, it also is an upper bound on the interval SP cost. Therefore, S_j is the interval SP cost. □

For avoiding the shortcoming of Nayeem–Pal algorithm, let the following steps of the SP-IW algorithm for finding the interval SP from node 1 to every node in the directed network with interval costs.

Step 1: Represent each interval weight $C_{ij} = [c_{ij,L}, c_{ij,R}]$ in terms of midpoint and width $\langle m(C_{ij}), w(C_{ij})\rangle$. In this case, the interval shortest path is formulated as follows:

$$
\begin{aligned}
&\min \sum_{i=1}^{m}\sum_{j=1}^{m} \langle m(C_{ij}), w(C_{ij})\rangle\, x_{ij}\\
&\text{s.t.}\\
&\qquad \sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = \begin{cases} 1, & i=1\\ 0, & i\neq 1, m\\ -1, & i=m \end{cases}\\
&\qquad x_{ij} \geq 0, i, j = 1, 2, \cdots, m.
\end{aligned}
$$

Step 2: Substitute the module of each interval number instead of the corresponding interval number in the interval shortest path problem under consideration and define:

$$
d_{ij} = \begin{cases} 0, & i = j,\\ \infty, & (i,j) \notin E,\\ m(C_{ij}), & (i,j) \in E. \end{cases} \tag{4.3}
$$

This means that the interval shortest path problem formulated in Step 1 is reformulated as the following crisp SP problem:

$$
\begin{aligned}
&\min \sum_{i=1}^{m}\sum_{j=1}^{m} m(C_{ij}) x_{ij}\\
&\text{s.t.}\\
&\qquad \sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = \begin{cases} 1, & i=1\\ 0, & i\neq 1, m\\ -1, & i=m \end{cases}\\
&\qquad x_{ij} \geq 0, i, j = 1, 2, \ldots, m.
\end{aligned}
$$

Step 3: Solve the obtained crisp SP problem in Step 2 by the generalized Dijkstra's algorithm given in Appendix A.

Step 4: Sum the interval weights of the arc on the shortest path obtained in Step 3 in order to find the interval shortest path weight.

Step 5: Let $P_1, P_2, \ldots, P_t$ are all shortest paths obtained in Step 3 and $\langle m(P_1), w(P_1)\rangle$,$\langle m(P_2), w(P_2)\rangle$, $\ldots, \langle m(P_t), w(P_t)\rangle$ are their corresponding interval weights. Let $w(P_g) = \min\{w(P_1), w(P_2), \ldots, w(P_t)\}$ and $w(P_h) = \max\{w(P_1), w(P_2), \ldots, w(P_t)\}$. In this case, an optimistic decision-maker selects path P_g as the interval shortest path and a pessimistic decision-makers selects path P_h as the interval shortest path.

The overall procedure of the SP-IW algorithm to find the shortest path of an interval shortest path problem and its associated interval fuzz is summarized in the flowchart of Figure 1.

It is worth noting that according to Nayeem–Pal algorithm, in order to carry out the Step 2.1 a lot of interval additions and comparison on interval numbers over the all iterations are required. In addition, to carry out the Step 2.2 it is required to compare more than two interval numbers simultaneously. But it is very difficult to compare a large number of interval numbers simultaneously using the acceptability index. While based on SP-IW algorithm all additions and comparisons are done on real numbers. These results confirm that SP-IW algorithm is simpler and computationally more efficient than Nayeem–Pal algorithm.

We also consider the following additional methodology for assessing the validity of the SP-IW algorithm as well as the validity of the Nayeem–Pal algorithm. Since the arc weights are represented with interval related lower and upper bounds, let us assume that the interval for arc, say, (i,j) is divided in a given set of bundles, so that let $W_{ij}^1, W_{ij}^2, \ldots, W_{ij}^n$ denote the weights of arc (i,j), being (W_{ij}^1, W_{ij}^n) the weight interval, W_{ij}^1 is the lower bound and W_{ij}^n is the upper bound. It seems acceptable to consider that the occurrence of the weights has equal probability. The new scheme should be based on the simulation where a realization of the weight of

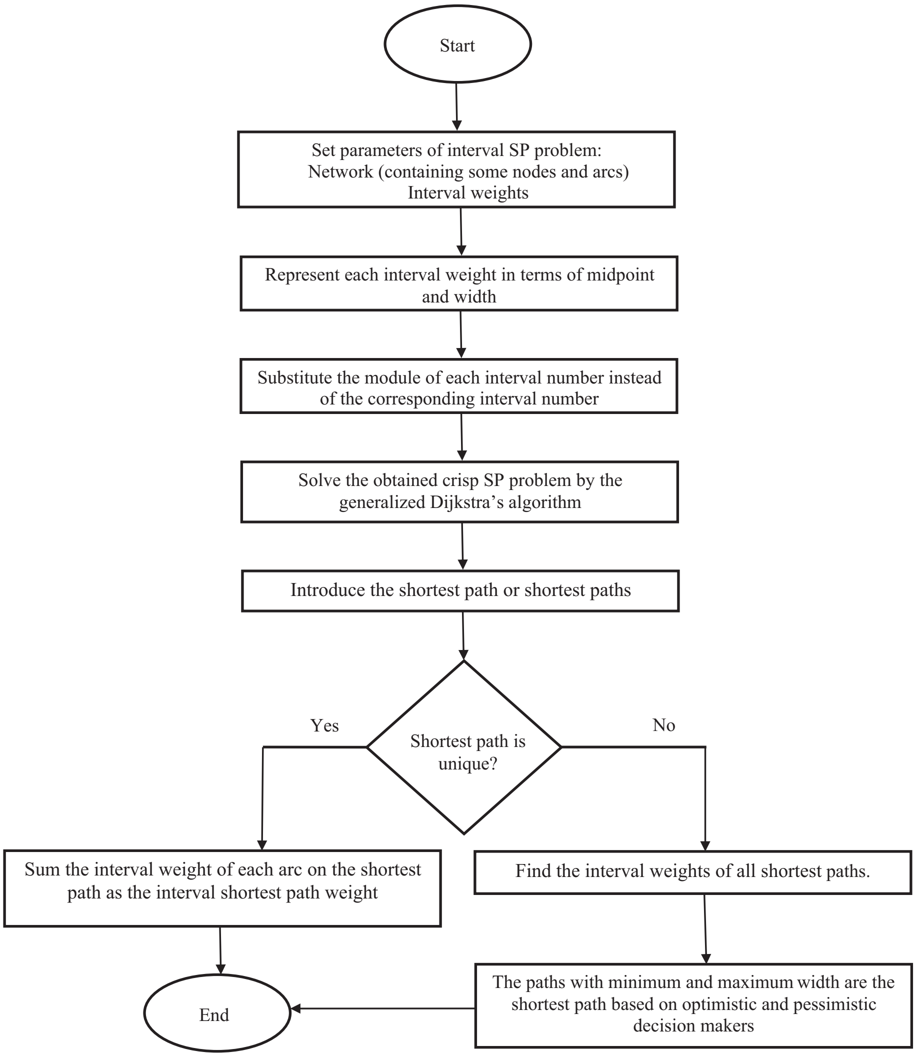
Start
Set parameters of interval SP problem:
Network (containing some nodes and arcs)
Interval weights
Represent each interval weight in terms of midpoint and width
Substitute the module of each interval number instead of the corresponding interval number
Solve the obtained crisp SP problem by the generalized Dijkstra's algorithm
Introduce the shortest path or shortest paths
Shortest path is unique?
Yes
No
Sum the interval weight of each arc on the shortest path as the interval shortest path weight
Find the interval weights of all shortest paths.
The paths with minimum and maximum width are the shortest path based on optimistic and pessimistic decision makers
End

FIGURE 1. The flowchart of the SP-IW algorithm.

each arc (i, j) is taken from the interval (W_{ij}^1, W_{ij}^n). For each simulation, the Dijkstra algorithm is executed. Once a given number of simulations are run, the interval of the minimum path is obtained, having minimum, mean and maximum. Results can be compared with the ones obtained from the Neyeen–Pal algorithm and the proposed one to assess the validity of the new proposal as well as the validity of the other algorithm.

5. Numerical examples

In this section, the advantages of SP-IW algorithm over Nayeem–Pal algorithm are illustrated by the numerical example given in Nayeem and Pal [46]. Also, one application of SP-IW algorithm on a small sized telecommunication network is presented.

Example 5.1. Let us consider the network in Figure 2 with interval weights as given in Table 1.

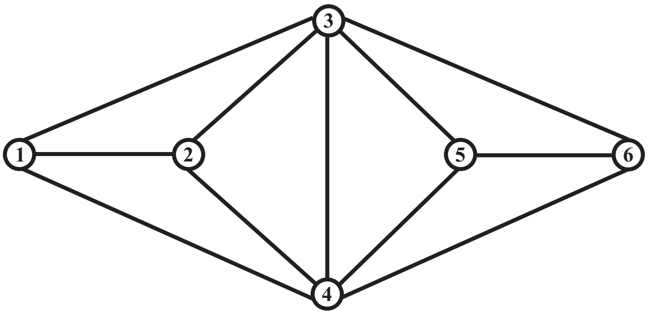

Figure 2. A network having interval weights.

Table 1. Arc information of Example 5.1 in terms of interval numbers.

Arc	Interval weight
(1, 2)	[10, 12]
(1,3)	[25, 28]
(1, 4)	[19, 20]
(2, 3)	[20, 21]
(2, 4)	[30, 35]
(3, 4)	[6.5, 7.5]
(3, 5)	[38, 40]
(3, 6)	[43, 44]
(4, 5)	[35, 40]
(4, 6)	[49, 51]
(5, 6)	[12, 13]

Table 2. Arc information of Example 5.1 in terms of mean-width notation.

Arc	Mean-width representation	Midpoint of interval weight
(1, 2)	$\langle 11, 1\rangle$	11
(1, 3)	$\langle 26.5, 1.5\rangle$	26.5
(1, 4)	$\langle 19.5, 0.5\rangle$	19.5
(2, 3)	$\langle 20.5, 0.5\rangle$	20.5
(2, 4)	$\langle 32.5, 2.5\rangle$	32.5
(3, 4)	$\langle 7, 0.5\rangle$	7
(3, 5)	$\langle 39, 1\rangle$	39
(3, 6)	$\langle 43.5, 0.5\rangle$	43.5
(4, 5)	$\langle 37.5, 2.5\rangle$	37.5
(4, 6)	$\langle 50, 1\rangle$	50
(5, 6)	$\langle 12.5, 0.5\rangle$	12.5

Based on Step 1 of SP-IW algorithm, the mean-width representation of each interval weight is given in Table 2.

Regarding the Step 2 of SP-IW algorithm, the module of each interval weight (given in Tab. 1) is put instead of corresponding interval numbers to obtain the classical shortest path problem. The results are given in Table 2.

The crisp shortest path problem with weights given in the third column of Table 2 can be solved by the standard Dijkstra's (1959) algorithm.

Now the Dijkstra's (1959) algorithm given in Appendix A is used to obtain the shortest path between node 1 and node 6. Initialization Step of the algorithm follows:

$$P = \{1\},\ T = \{2,3,4,5,6\},\ \mathrm{PL}(1) = 0,\ \mathrm{TL}(2) = 11,\ \mathrm{TL}(3) = 26.5,\ \mathrm{TL}(4) = 19.5,\ \mathrm{TL}(5) = \mathrm{TL}(6) = \infty.$$

Iteration 1

Step 1: By Step 1 TL(2) = 11 is the minimum temporary label among all other temporary labels and it is set as permanent label of node 2. Thus, $P = \{1,2\},\ T = \{3,4,5,6\},\ \mathrm{PL}(1) = 0,\ \mathrm{PL}(2) = 11.$

Step 2: Since $P \neq N$, the process is repeated. The new temporary labels of nodes belonging to T are calculated as follows:

$$\begin{aligned}
\mathrm{TL}(3) &= \min\{\mathrm{TL}(3),\ \mathrm{TL}(2) + m(d_{23})\} = \min\{26.5, 11 + 20.5\} = 26.5,\\
\mathrm{TL}(4) &= \min\{\mathrm{TL}(4),\ \mathrm{TL}(2) + m(d_{24})\} = \min\{19.5, 11 + 32.5\} = 19.5,\\
\mathrm{TL}(5) &= \min\{\mathrm{TL}(5),\ \mathrm{TL}(2) + m(d_{25})\} = \min\{\infty, 11 + \infty\} = \infty,\\
\mathrm{TL}(6) &= \min\{\mathrm{TL}(6),\ \mathrm{TL}(2) + m(d_{26})\} = \min\{\infty, 11 + \infty\} = \infty.
\end{aligned}$$

Iteration 2

Step 1: Node 4 has the lowest temporary label among all other temporary labels and TL(4) = 19.5 is set as permanent label of node 4. Thus, $P = \{1,2,4\},\ T = \{3,5,6\},\ \mathrm{PL}(1) = 0,\ \mathrm{PL}(2) = 11,\ \mathrm{PL}(4) = 19.5.$

Step 2: Since $P \neq N$, the process is repeated. The new temporary labels of nodes belonging to T are calculated as follows:

$$\begin{aligned}
\mathrm{TL}(3) &= \min\{\mathrm{TL}(3),\ \mathrm{TL}(4) + m(d_{43})\} = \min\{26.5, 19.5 + 7\} = 26.5,\\
\mathrm{TL}(5) &= \min\{\mathrm{TL}(5),\ \mathrm{TL}(4) + m(d_{45})\} = \min\{\infty, 19.5 + 37.5\} = 57,\\
\mathrm{TL}(6) &= \min\{\mathrm{TL}(6),\ \mathrm{TL}(4) + m(d_{46})\} = \min\{\infty, 19.5 + 50\} = 69.5.
\end{aligned}$$

Iteration 3

Step 1: Node 3 has the lowest temporary label among all other temporary labels of nodes and TL(3) = 26.5 is set as permanent label of node 3. Thus,

$$P = \{1,2,4,3\},\ T = \{5,6\},\ \mathrm{PL}(1) = 0,\ \mathrm{PL}(2) = 11,\ \mathrm{PL}(4) = 19.5,\ \mathrm{PL}(3) = 26.5.$$

It is worth noting that, in this step the remaining steps can be splatted up into two steps, setting $\mathrm{TL}(4) + m(d_{43})$ in one and TL(3) derived in Iteration 2 in the other as the permanent label of node 3.

Step 2: Since $P \neq N$, the process is repeated. The new temporary labels of nodes belonging to T are calculated as follows:

$$\begin{aligned}
\mathrm{TL}(5) &= \min\{\mathrm{TL}(5),\ \mathrm{TL}(3) + m(d_{35})\} = \min\{57, 26.5 + 39\} = 57,\\
\mathrm{TL}(6) &= \min\{\mathrm{TL}(6),\ \mathrm{TL}(3) + m(d_{36})\} = \min\{69.5, 26.5 + 43.5\} = 69.5.
\end{aligned}$$

Iteration 4

Step 1: Node 5 has the lowest temporary label among all other temporary labels of nodes and TL(5) = 57 is set as permanent label of node 5. Thus,

$$P = \{1,2,4,3,5\},\ T = \{6\},\ \mathrm{PL}(1) = 0,\ \mathrm{PL}(2) = 11,\ \mathrm{PL}(4) = 19.5,\ \mathrm{PL}(3) = 26.5,\ \mathrm{PL}(5) = 57.$$

Step 2: Since $P \neq N$, the process is repeated. The new temporary labels of nodes belonging to T are calculated as follows:

$$\mathrm{TL}(6) = \min\{\mathrm{TL}(6),\ \mathrm{TL}(5) + m(d_{56})\} = \min\{69.5, 57 + 12.5\} = 69.5.$$

Table 3. Simulation results.

Path	Number of being shortest path
$P_1 : 1 \to 4 \to 5 \to 6$	838
$P_2 : 1 \to 4 \to 6$	162
Min	66
Max	71
Average	67.54
Count	1000

Iteration 5

Step 1: In this case $\mathrm{TL}(6) = 69.5$ is set as permanent label of node 6. Thus,

$$P = \{1, 2, 4, 3, 5, 6\},\ T = \{\},\ \mathrm{PL}(1) = 0,\ \mathrm{PL}(2) = 11,\ \mathrm{PL}(4) = 19.5,\ \mathrm{PL}(3) = 26.5, \mathrm{PL}(5) = 57,\ \mathrm{PL}(6) = 62.5.$$

Again, in this step the remaining steps can be splatted up into two steps, setting $\mathrm{TL}(5) + m(d_{56})$ in one and $\mathrm{TL}(6)$ derived in Iteration 3 in the other as the permanent label of node 6.

Step 2: Since $P = N$, the process stops.

Now based on Step 4 of SP-IW algorithm, the interval shortest path between node 1 and node 6 can be obtained by the following procedure:

Case 1: If $\mathrm{TL}(5) + m(d_{56})$ is considered as the permanent label of node 6, then it comes from node 5. The permanent label of node 5 is $\mathrm{TL}(4) + m(d_{45})$ which represents that it comes from node 4. The permanent label of node 4 is $\mathrm{TL}(1) + m(d_{14})$ which represents that it comes from node 1. Hence the interval shortest path is $P_1 : 1 \to 4 \to 5 \to 6$. Also, the interval weight of this path is obtained as follows:

$$\langle m(P_1), w(P_1)\rangle = \langle m(C_{14}), w(C_{14})\rangle \oplus \langle m(C_{45}), w(C_{45})\rangle \oplus \langle m(C_{56}), w(C_{56})\rangle = \langle 69.5, 3.5\rangle\,.$$

Case 2: If $\mathrm{TL}(6) = \mathrm{TL}(4) + m(d_{46})$ derived in Iteration 2 or 3 is considered as the permanent label of node 6, then it comes from node 4. The permanent label of node 4 is $\mathrm{TL}(1) + m(d_{14})$ which represents that it comes from node 1. Hence the interval shortest path is $P_2 : 1 \to 4 \to 6$. Also, the interval weight of this path is obtained as follows:

$$\langle m(P_2), w(P_2)\rangle = \langle m(C_{14}), w(C_{14})\rangle \oplus \langle m(C_{46}), w(C_{46})\rangle = \langle 69.5, 1.5\rangle\,.$$

It can be seen that that Nayeem–Pal algorithm and SP-IW algorithm produce the same paths, while Nayeem–Pal algorithm is very confusing to understand and to apply for obtaining the interval shortest path compare to SP-IW algorithm. Also, in Nayeem–Pal algorithm all arithmetic operations are performed on the interval numbers, while in SP-IW algorithm all arithmetic operations are done on real numbers. Thus, it is concluded that SP-IW algorithm is by far simpler and computationally more efficient than Nayeem–Pal algorithm.

Finally, we apply the mentioned methodology at the end of the previous section on the network given in Figure 2 for assessing the validity of the proposed SP-IW algorithm and Nayeem–Pal algorithm. To do this, we have divided each interval into $n = 10$ subinterval and for each simulation the Dijkstra algorithm is executed 100 times independently. The results are given in Table 3. As can be seen from Table 3, the path $P_1 : 1 \to 4 \to 5 \to 6$ is identified as the shortest path 838 times and the path $P_2 : 1 \to 4 \to 6$ is identified as the shortest path 162 times. It should be mentioned that both the proposed SP-IW algorithm and Nayeem–Pal algorithm identified the paths $P_1 : 1 \to 4 \to 5 \to 6$ and $P_2 : 1 \to 4 \to 6$ as the shortest path. This confirms the validity of the proposed SP-IW algorithm as well as the validity of the Nayeem–Pal algorithm.

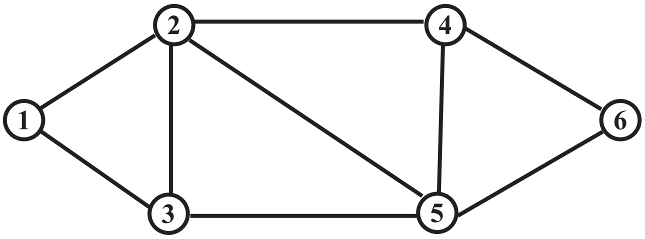

Figure 3. A telecommunication network.

Table 4. Arc information of Example 5.2 in terms of interval numbers.

Arc	Interval weight
(1, 2)	[1.5, 3.5]
(1, 3)	[2.5, 4.5]
(2, 3)	[4, 6]
(2, 4)	[1.5, 3.5]
(2, 5)	[3.5, 5.5]
(3, 4)	[2, 4]
(4, 5)	[2.5, 4.5]
(4, 6)	[1.5, 3.5]
(5, 6)	[4.5, 6.5]

Table 5. Arc information of Example 5.2 in terms of mean-width notation.

Arc	Mean-width representation	Midpoint of interval weight
(1, 2)	$\langle 2.5, 1\rangle$	2.5
(1, 3)	$\langle 3.5, 1.5\rangle$	3.5
(2, 3)	$\langle 5, 1\rangle$	5
(2, 4)	$\langle 2.5, 1\rangle$	2.5
(2, 5)	$\langle 4.5, 1\rangle$	4.5
(3, 5)	$\langle 3, 1\rangle$	3
(4, 5)	$\langle 3.5, 1\rangle$	3.5
(4, 6)	$\langle 2.5, 1\rangle$	2.5
(5, 6)	$\langle 5.5, 10\rangle$	5.5

Example 5.2. Consider a mobile service company which handles six geographical centers. A configuration of a telecommunication network is presented in Figure 3. Assume that the distance between any two centers is an interval number (the arc lengths are given in Tab. 4). The company wants to find a shortest path for an effective message flow amongst the centers.

Based on Step 1 of SP-IW algorithm, the mean-width representation of each interval weight is given in Table 5.

Regarding Step 2, the module of each interval weight (given in Tab. 4) is put instead of corresponding interval numbers to get the crisp shortest path problem. The results are given in Table 5.

The crisp shortest path problem with weights given in the third column of Table 5 can be solved by the Dijkstra's (1959) algorithm given in the Appendix A. Now this algorithm is utilized to find the shortest path between node 1 and node 6. Initialization Step of the algorithm follows:

$$P = \{1\},\ T = \{2,3,4,5,6\},\ \mathrm{PL}(1) = 0,\ \mathrm{TL}(2) = 2.5,\ \mathrm{TL}(3) = 3.5,\ \mathrm{TL}(4) = \mathrm{TL}(5) = \mathrm{TL}(6) = \infty.$$

Iteration 1

Step 1: By Step 1 TL(2) $= 2.5$ is the minimum temporary label among all other temporary labels and it is set as permanent label of node 2. Thus, $P = \{1,2\},\ T = \{3,4,5,6\},\ \mathrm{PL}(1) = 0,\ \mathrm{PL}(2) = 2.5$.

Step 2: Since $P \neq N$, the process is repeated. The new temporary labels of nodes belonging to T are calculated as follows:

$$\begin{aligned}
\mathrm{TL}(3) &= \min\{\mathrm{TL}(3),\ \mathrm{TL}(2) + m(d_{23})\} = \min\{3.5, 2.5 + 5\} = 3.5,\\
\mathrm{TL}(4) &= \min\{\mathrm{TL}(4),\ \mathrm{TL}(2) + m(d_{24})\} = \min\{\infty, 2.5 + 2.5\} = 5,\\
\mathrm{TL}(5) &= \min\{\mathrm{TL}(5),\ \mathrm{TL}(2) + m(d_{25})\} = \min\{\infty, 2.5 + 4.5\} = 7,\\
\mathrm{TL}(6) &= \min\{\mathrm{TL}(6),\ \mathrm{TL}(2) + m(d_{26})\} = \min\{\infty, 2.5 + \infty\} = \infty.
\end{aligned}$$

Iteration 2

Step 1: Node 3 has the lowest temporary label among all other temporary labels and TL(3) $= 3.5$ is set as permanent label of node 3. Thus, $P = \{1,2,3\},\ T = \{4,5,6\},\ \mathrm{PL}(1) = 0,\ \mathrm{PL}(2) = 2.5,\ \mathrm{PL}(3) = 3.5$.

Step 2: Since $P \neq N$, the process is repeated. The new temporary labels of nodes belonging to T are calculated as follows:

$$\begin{aligned}
\mathrm{TL}(4) &= \min\{\mathrm{TL}(4),\ \mathrm{TL}(3) + m(d_{34})\} = \min\{5, 3.5 + \infty\} = 5,\\
\mathrm{TL}(5) &= \min\{\mathrm{TL}(5),\ \mathrm{TL}(3) + m(d_{35})\} = \min\{7, 3.5 + 3\} = 6.5,\\
\mathrm{TL}(6) &= \min\{\mathrm{TL}(6),\ \mathrm{TL}(3) + m(d_{36})\} = \min\{\infty, 3.5 + \infty\} = \infty.
\end{aligned}$$

Iteration 3

Step 1: Node 4 has the lowest temporary label among all other temporary labels of nodes and TL(4) $= 5$ is set as permanent label of node 5. Thus,

$$P = \{1,2,3,4\},\ T = \{5,6\},\ \mathrm{PL}(1) = 0,\ \mathrm{PL}(2) = 2.5,\ \mathrm{PL}(3) = 3.5,\ \mathrm{PL}(4) = 5.$$

Step 2: Since $P \neq N$, the process is repeated. The new temporary labels of nodes belonging to T are calculated as follows:

$$\begin{aligned}
\mathrm{TL}(5) &= \min\{\mathrm{TL}(5),\ \mathrm{TL}(4) + m(d_{45})\} = \min\{6.5, 5 + 3.5\} = 6.5,\\
\mathrm{TL}(6) &= \min\{\mathrm{TL}(6),\ \mathrm{TL}(4) + m(d_{46})\} = \min\{\infty, 5 + 2.5\} = 7.5.
\end{aligned}$$

Iteration 4

Step 1: Node 5 has the lowest temporary label among all other temporary labels of nodes and TL(5) $= 6.5$ is set as permanent label of node 5. Thus, we have

$$P = \{1,2,3,4,5\},\ T = \{6\},\ \mathrm{PL}(1) = 0,\ \mathrm{PL}(2) = 2.5,\ \mathrm{PL}(3) = 3.5,\ \mathrm{PL}(4) = 5,\ \mathrm{PL}(5) = 6.5.$$

Step 2: Since $P \neq N$, the process is repeated. The new temporary labels of nodes belonging to T are calculated as follows:

$$\mathrm{TL}(6) = \min\{\mathrm{TL}(6),\ \mathrm{TL}(5) + m(d_{56})\} = \min\{7.5, 6.5 + 5.5\} = 7.5.$$

Table 6. Shortest path from source node 1 to every other node.

Destination node	Shortest path	Shortest path weight
2	$1 \to 2$	$\langle 2.5, 1\rangle$
3	$1 \to 3$	$\langle 3.5, 1.5\rangle$
4	$1 \to 2 \to 4$	$\langle 5, 2\rangle$
5	$1 \to 3 \to 5$	$\langle 6.5, 2.5\rangle$

Iteration 5

Step 1: In this case TL(6) = 7.5 is set as permanent label of node 6. Thus,

$$P = \{1, 2, 3, 4, 5, 6\},\ T = \{\},\ \mathrm{PL}(1) = 0,\ \mathrm{PL}(2) = 2.5,\ \mathrm{PL}(3) = 3.5,\ \mathrm{PL}(4) = 5, \mathrm{PL}(5) = 6.5,\ \mathrm{PL}(6) = 7.5.$$

Step 2: Since $P = N$, the process stops.

Now, the interval shortest path between node 1 and every other node can be obtained. For instance, the interval shortest path between node 1 and node 6 is obtained by the following procedure:

The permanent label of node 6 is $\mathrm{TL}(4) + m(d_{46})$ which represents it comes from node 4. The permanent label of node 4 is $\mathrm{TL}(2) + m(d_{25})$ which represents that it comes from node 2. The permanent label of node 2 is $\mathrm{TL}(1) + m(d_{12})$ which represents that it comes from node 1. Hence the interval shortest path is $P_1 : 1 \to 2 \to 4 \to 6$. Also, the interval weight of this path is obtained as follows:

$$\langle m(P_1), w(P_1)\rangle = \langle m(C_{12}), w(C_{12})\rangle \oplus \langle m(C_{24}), w(C_{24})\rangle \oplus \langle m(C_{46}), w(C_{46})\rangle = \langle 7.5, 3\rangle\,.$$

In a similar way, the interval shortest path from source node 1 to every other node can be determined. The results are given in Table 6.

Moreover, the shortest path between source node $i (i \in N, i \neq 1)$ and every other node can be determined based on SP-IW algorithm.

Finally, it can be said that it is better to use SP-IW algorithm compared to Nayeem–Pal algorithm in order to solve the interval SP problem from a computational point of view.

6. Large scale pilot case

In this section, the practical relevance of SP-IW algorithm is evaluated by means of a large-scale pilot case. A pharmaceutical shipment must be shipped from Tabriz in Azarbayjan province to Bushehr in Bushehr province in Iran. There are several different paths from Tabriz to Zahedan, shown in Figure 4. The problem is to obtain a path to travel from Tabriz to Bushehr in the shortest time by car according to the map given in Figure 4. The road network consists of 17 cities (each city is considered as a node) and 29 connecting links between cities and their associated driving times. The shipping costs between cities connected by lines have been expressed as intervals with a width equal to about 20% of the average shipping cost (unit: Riyal).

To determine the non-dominated shortest paths from Tabriz to Bushehr, the shortest path problem indicated by the data of Table 7 is solved.

The SP-IW algorithm is used to solve the corresponding shortest path problem with the data in the last column of Table 6. Three non-dominated paths are obtained as follows:

Path 1 : Tabriz → Zanjan → Hamedan → Sharekord → Yasouj → Bushehr
Path 2 : Tabriz → Zanjan → Ghazvin → Isfahan → Shiraz → Yasouj → Bushehr
Path 3 : Tabriz → Zanjan → Hamedan → Sharekord → Isfahan → Shiraz → Yasouj → Bushehr

The interval weights corresponding to Path 1, Path 2 and Path 3 are $\langle m(Path1), w(Path1)\rangle = \langle 394, 21\rangle$, $\langle m(Path2), w(Path2)\rangle = \langle 394, 14\rangle$ and $\langle m(Path3), w(Path3)\rangle = \langle 394, 15\rangle$, respectively.

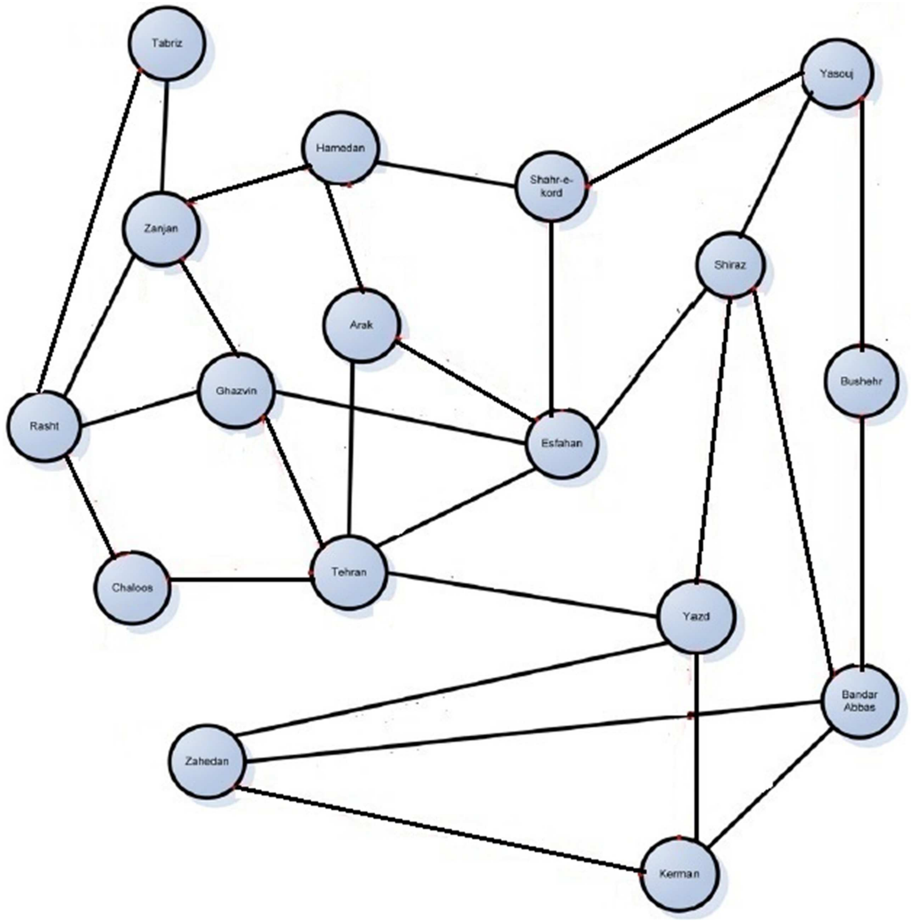

FIGURE 4. The routes network.

Since there are three non-dominated paths from Tabriz to Bushehr, the SP-IW algorithm gives a guideline to the decision maker to choose the best of these paths according to his/her view. Since $w(Path2) < w(Path1), w(Path3)$ and $w(Path1) > w(Path2), w(Path3)$, an optimistic decision maker selects Path 3 as the interval shortest path and a pessimistic decision makers selects Path 1 as the interval shortest path.

7. CONCLUSIONS AND DISCUSSIONS

In this study, we proposed a solution methodology for solving SP problems where imprecise weights are represented in terms of interval numbers. An equivalent crisp SP problem was proposed to derive the interval optimal path of the interval SP problem. In the proposed approach, the interval SP problem was transformed into a crisp equivalent SP problem. The obtained results confirmed that SP-IW algorithm requires less interval computations as opposed to Nayeem–Pal algorithm. In summary, to solve the interval SP problem by using Nayeem–Pal algorithm, interval arithmetic operations and interval comparisons are required. On the other hand, only arithmetic operations and comparison of real numbers are required to solve the same problem with SP-IW algorithm. Let us explore the main advantages of the proposed method briefly.

(1) The proposed technique doesn't use the genetic and interval solution approaches which are difficult to apply in real life situations.

Table 7. Interval shipping costs.

Arc number	Route	Interval weight	Mean-width representation	Midpoint
1	(Tabriz–Rasht)	[69, 71]	$\langle 70, 1\rangle$	70
2	(Tabriz–Zanjan)	[63, 67]	$\langle 65, 2\rangle$	65
3	(Zanjan–Rasht)	[48, 56]	$\langle 52, 4\rangle$	52
4	(Zanjan–Hamedan)	[56, 62]	$\langle 59, 3\rangle$	59
5	(Zanjan–Ghazvin)	[42, 44]	$\langle 43, 1\rangle$	43
6	(Gazvin–Rasht)	[48, 54]	$\langle 51, 3\rangle$	51
7	(Ghazvin–Isfahan)	[91, 99]	$\langle 95, 4\rangle$	95
8	(Ghazvin–Tehran)	[46, 50]	$\langle 48, 2\rangle$	48
9	(Yazd–Tehran)	[103, 111]	$\langle 107, 4\rangle$	107
10	(Rasht–Chalous)	[34, 38]	$\langle 36, 2\rangle$	36
11	(Chalous–Tehran)	[62, 64]	$\langle 63, 1\rangle$	63
12	(Tehran–Arak)	[49, 53]	$\langle 51, 3\rangle$	51
13	(Tehran–Isfahan)	[65, 73]	$\langle 69, 4\rangle$	69
14	(Hamedan–Arak)	[38, 44]	$\langle 41, 3\rangle$	41
15	(Hamedan–Shahrekord)	[59, 63]	$\langle 61, 2\rangle$	61
16	(Arak–Isfahan)	[73, 83]	$\langle 78, 5\rangle$	78
17	(Isfahan–Shiraz)	[75, 83]	$\langle 79, 4\rangle$	79
18	(Shiraz–Yasouj)	[50, 52]	$\langle 51, 1\rangle$	51
19	(Shiraz–Bandar Abbas)	[98, 110	$\langle 104, 6\rangle$	104
20	(Shiraz–Yazd)	[77, 81]	$\langle 79, 2\rangle$	79
21	(Yazd–Zahedan)	[147, 167]	$\langle 157, 10\rangle$	157
22	(Yazd–Kerman)	[88, 98]	$\langle 93, 5\rangle$	93
23	(Kerman–Zahedan)	[133, 149]	$\langle 141, 8\rangle$	141
24	(Kerman–Bandar Abbas)	[117, 131]	$\langle 124, 7\rangle$	124
25	(Bandar Abbas–Bushehr)	[127, 147]	$\langle 137, 10\rangle$	137
26	(Bushehr–Yasouj)	[59, 63]	$\langle 61, 2\rangle$	61
27	(Yasouj–Shahrekord)	[136, 160]	$\langle 148, 12\rangle$	148
28	(Isfahan–Shahrekord)	[17, 19]	$\langle 18, 1\rangle$	18
29	(Bandar Abbas–Zahedan)	[180, 210]	$\langle 195, 15\rangle$	195

(2) Since, for applying SP-IW algorithm there is a need to solve crisp SP problem. So, the existing and easily available software can be used for the same. However, for applying Nayeem–Pal algorithm there is a need to solve interval SP problem so the existing and easily available software cannot be used and need to develop new software.

(3) In contrast to Nayeem–Pal algorithm, which uses interval arithmetic operations for solving interval SP problem, the SP-IWA needs arithmetic operations of real numbers. Moreover, in contrast to Nayeem–Pal algorithm that compares a large number of interval numbers simultaneously using the acceptability index, the SP-IWA determines the interval SP without any interval method. Hence, from a computation point of view the SP-IWA is preferable to the Nayeem–Pal algorithm for solving the interval SP problem.

Many other areas remain to be researched. Some of these are discussed below.

- In many real-world applications several conflicting objectives must be considered. The proposed approach can be extended for solving multi objective SP problem with interval numbers.
- The SP-IW algorithm cannot be applied in networks with negative interval numbers and it cannot detect whether there are negative circuits. The generalization of this algorithm to overcome these shortcomings is an interesting topic for future research.
- The development of SP-IW algorithm for deriving the interval SP between all pairs of nodes is left to the next study.

Appendix A.

Here, the steps of Dijkstra's (1995) algorithm are presented for solving shortest path problem with the midpoint of each interval weight as the length of each arc:

Initialization step

Assume P be the set of nodes with permanent labels and $T = N - P$. The permanent and temporary labels of node i are represented by PL(i) and TL(i), respectively. Let,

$$P = \{1\},\ \mathrm{PL}(1) = 0,\ \mathrm{TL}(j) = d_{1j}, j \in N.$$

Main step

Step 1: If TL(k) is the minimum temporary label among all other temporary labels then it is set as permanent label of node k and let $P := P \cup \{k\},\ \mathrm{PL}(k) := \mathrm{TL}(k),\ T := T - \{k\}$. If $P = N$, then stop.

Step 2: Modify the temporary label of node as follows and return to Step1:

$$\mathrm{TL}(j) = \min\left\{\mathrm{TL}(j),\ \mathrm{TL}(k) + d_{kj}\right\}.$$

Remark A.1. If the minimum test of Step 1 accrues at more than one index (say, r temporary labels) then we split-up the process into r branches and do the Step 1 for each of the r branches and then continue.

Acknowledgements. The author would like to thank the anonymous reviewers and the editors for their insightful comments and suggestions.

References

[1] A. Abbaszadeh Sori, A. Ebrahimnejad and H. Motameni, Elite artificial bees' colony algorithm to solve robot's fuzzy constrained routing problem. *Comput. Intell.* **36** (2019) 659–681.

[2] A. Abbaszadeh Sori, A. Ebrahimnejad and H. Motameni, The fuzzy inference approach to solve multi-objective constrained shortest path problem. *J. Intell. Fuzzy Syst.* **38** (**4**) (2019) 1–10.

[3] E. Ahmadi, G.A. Süer and F. Al-Ogaili, Solving stochastic shortest distance path problem by using genetic algorithms. *Proc. Comput. Sci.* **140** (2018) 79–86.

[4] R.K. Ahuja, T.L. Magnanti and J.B. Orlin, Network Flows: Theory, Algorithms and Applications. Prentice-Hall, Englewood Cliffs, NJ (1993).

[5] C. Alexopoulos, State space partitioning methods for stochastic shortest path problems. *Networks* **30** (1997) 9–21.

[6] J.L. Bander and C.C. White, A heuristic search approach for a nonstationary stochastic shortest path problem with terminal cost. *Trans. Sci.* **36** (2002) 218–230.

[7] H. Beigy and M.R. Meybodi, Utilizing distributed learning automata to solve stochastic shortest path problems. *Int. J. Uncertainty Fuzziness Knowledge Based Syst.* **14** (2006) 591–616.

[8] B. Bosek, D. Leniowski, P. Sankowski and A. Zych-Pawlewicz, Shortest augmenting paths for online matchings on trees. *Theory Comput. Syst.* **62** (2018) 337–348.

[9] S. Broumi, A. Dey, M. Talea, A. Bakali, F. Smarandache, D. Nagarajan, M. Lathamaheswari and R. Kumar, Shortest path problem using Bellman algorithm under neutrosophic environment. *Complex Intell. Syst.* **5** (2019) 409–416.

[10] R.K. Cheung, Iterative methods for dynamic stochastic shortest path problems. *Nav. Res. Logist.* **45** (1998) 769–789.

[11] T.N. Chuang and J.Y. Kung, The fuzzy shortest path length and the corresponding shortest path in a network, *Comput. Oper. Res.* **32** (2005) 1409–1428.

[12] T.N. Chuang and J.Y. Kung, A new algorithm for the discrete fuzzy shortest path problem in a network. *Appl. Math. Comput.* **174** (2006) 660–668.

[13] J.S. Croucher, A note on the stochastic shortest-route problem. *Nav. Res. Logist.* **25** (1978) 729–732.

[14] Y. Deng, Y. Chen, Y. Zhang and S. Mahadevan, Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Appl. Soft Comput.* **12** (2012) 1231–1237.

[15] A. Dey, R. Pradhan, A. Pal and T. Pal, A genetic algorithm for solving fuzzy shortest path problems with interval type-2 fuzzy arc lengths. *Malaysian J. Comput. Sci.* **31** (2018) 255–270.

[16] Y. Dou, L. Zhu and H.S. Wang, Solving the fuzzy shortest path problem using multi-criteria decision method based on vague similarity measure. *Appl. Soft Comput.* **12** (2012) 1621–1631.

[17] D. Dubois and H. Prade, Fuzzy Sets and Systems: Theory and Applications. Academic Press, New York, NY (1980).

[18] A. Ebrahimnejad, Z. Karimnejad and H. Alrezaamiri, Particle swarm optimization algorithm for solving shortest path problems with mixed fuzzy arc weights. *Int. J. Appl. Decis. Sci.* **8** (2015) 203–222.

[19] A. Ebrahimnejad, M. Tavana and H. Alrezaamiri, A novel artificial bee colony algorithm for shortest path problems with fuzzy arc weights. *Measurement* **93** (2016) 48–56.

[20] M. Enayattabar, A. Ebrahimnejad and H. Motameni, Dijkstra algorithm for shortest path problem under interval-valued Pythagorean fuzzy environment. *Complex Intell. Syst.* **5** (2019) 93–10.

[21] M. Enayattabar, A. Ebrahimnejad, H. Motameni and H. Garg, A novel approach for solving all-pairs shortest path problem in an interval-valued fuzzy network. *J. Intell. Fuzzy Syst.* **37** (2019) 6865–6877.

[22] M. Eshaghnezhad, F. Rahbarnia, S. Effati and A. Mansoori, An artificial neural network model to solve the fuzzy shortest path problem. *Neural Proc. Lett.* **50** (2019) 1527–1548.

[23] Y. Gao, Shortest path problem with uncertain arc lengths. *Comput. Math. App.* **62** (2011) 2591–2600.

[24] M. Ghiyasvand, A new approach for solving the minimum cost flow problem with interval and fuzzy data. *Int. J. Uncertainty Fuzziness Knowledge Based Syst.* **19** (2011) 71–88.

[25] M. Ghiyasvand, Solving the minimum flow problem with interval bounds and flows. *Sadhana* **37** (2012) 665–674.

[26] M. Guillot and G. Stauffer, The stochastic shortest path problem: a polyhedral combinatorics perspective. *Eur. J. Oper. Res.* **285** (**1**) (2018) 148–158.

[27] Y. Guo, S. Li, W. Jiang, B. Zhang and Y. Ma, Learning automata-based algorithms for solving the stochastic shortest path routing problems in 5G wireless communication. *Phys. Commun.* **25** (2017) 376–385.

[28] T. Hagerup, Simpler computation of single-source shortest paths in linear average time. *Theory Comput. Syst.* **39** (2006) 113–120.

[29] S.M. Hashemi, M. Ghatee and E. Nasrabadi, Combinatorial algorithms for the minimum interval cost flow problem. *Appl. Math. Comput.* **175** (2006) 1200–1216.

[30] R. Hassanzadeh, I. Mahdavi, N. Mahdavi-Amiri and A. Tajdin, A genetic algorithm for solving fuzzy shortest path problems with mixed fuzzy arc lengths. *Math. Comput. Model.* **57** (2013) 84–99.

[31] F. Hernandes, M.T. Lamata, J.L. Verdegay and A. Yamakami, The shortest path problem on networks with fuzzy parameters. *Fuzzy Sets Syst.* **158** (2007) 1561–1570.

[32] C. Horoba, Ant colony optimization for stochastic shortest path problems. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation. Association for Computing Machinery, New York, NY (2010) 1465–1472.

[33] K.R. Hutson and D.R. Shier, Extended dominance and a stochastic shortest path problem. *Comput. Oper. Res.* **36** (2009) 584–596.

[34] X. Ji, Models and algorithm for shortest path problem. *Appl. Math. Comput.* **170** (2005) 503–514.

[35] X. Ji, K. Iwamura and Z. Shao, New models for shortest path problem with problem with fuzzy arc lengths. *Appl. Math. Model.* **31** (2007) 259–269.

[36] J. Kamburowski, A note on the stochastic shortest route problem. *Oper. Res.* **33** (1985) 696–698.

[37] C.M. Klein, Fuzzy shortest paths. *Fuzzy Sets Syst.* **39** (1991) 27–41.

[38] A. Kumar and M. Kaur, A new algorithm, for solving shortest path problem on a network with imprecise edge weight. *App. Appl. Math. Int. J.* **6** (2011) 602–619.

[39] Y. Li, M. Gen and K. Ida, Solving fuzzy shortest path problems by neural networks. *Comput. Ind. Eng.* **31** (1996) 861–865.

[40] K.C. Lin and M.S. Chern, The fuzzy shortest path problem and its most vital arcs. *Fuzzy Sets Syst.* **58** (1993) 343–353.

[41] I. Mahdavi, R. Nourifar, A. Heidarzade and N. Mahdavi-Amiri, A dynamic programming approach for finding shortest chains in fuzzy network. *Appl. Soft Comput.* **9** (2009) 503–511.

[42] P. Mani, S. Broumi and K. Muthusamy, A network shortest path algorithm via hesitancy fuzzy digraph. *J. New Theory* **27** (2019) 52–62.

[43] H. Motameni and A. Ebrahimnejad, Constraint shortest path problem in a network with intuitionistic fuzzy arc weights, edited by J. Medina, M. Ojeda-Aciego, J. Verdegay, I. Perfilieva, B. Bouchon-Meunier and R. Yager. In: Vol. 855 of *Communications in Computer and Information Science. Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications. IPMU 2018*. Springer, New York, NY (2018).

[44] I. Murthy and S. Sarkar, A relaxation-based pruning technique for a class of stochastic shortest path problems. *Trans. Sci.* **30** (1996) 220–236.

[45] I. Murthy and S. Sarkar, Stochastic shortest path problems with piecewise-linear concave utility functions. *Manage. Sci.* **44** (1998) 125–136.

[46] S.M.A. Nayeem and M. Pal, Shortest path problem on a network with imprecise edge weight. *Fuzzy Optim. Decis. Making* **4** (2005) 293–312.

[47] E. Nikolova, J.A. Kelner, M. Brand and M. Mitzenmacher, Stochastic shortest paths via quasi-convex maximization. In: Vol. 4168 of *Lecture Notes in Computer Science. Algorithms–ESA 2006*. Springer, New York, NY (2006) 552–563.

[48] Y. Ohtsubo, Stochastic shortest path problems with associative accumulative criteria. *Appl. Math. Comput.* **198** (2008) 198–208.

[49] S. Okada, Fuzzy shortest path problems incorporating interactivity among paths. *Fuzzy Sets Syst.* **142** (2004) 335–357.

[50] S. Okada and M. Gen, Order relation between intervals and its application to shortest path problem. *Comput. Ind. Eng.* **25** (1993) 147–150.

[51] S. Okada and T. Soper, A shortest path problem on a network with fuzzy arc lengths. *Fuzzy Sets Syst.* **109** (2000) 129–140.

[52] S.K. Peer and D.K. Sharma, Finding the shortest path in stochastic networks. *Comput. Math. App.* **53** (2007) 729–740.

[53] G.H. Polychronopoulos and J.N. Tsitliklis, Stochastic shortest path problem with recourse. *Networks* **27** (1996) 133–143.

[54] C.M. Ramos and F. Sagols, The minimum cost flow problem with interval and fuzzy arc costs. *Morfismos* **5** (2011) 57–68.

[55] A. Sengupta and T.K. Pal, Theory and methodology on comparing interval numbers. *Eur. J. Oper. Res.* **127** (2000) 28–43.

[56] A. Sengupta and T.K. Pal, Solving the shortest path problem with interval arcs. *Fuzzy Optim. Decis. Making* **5** (2006) 71-89.

[57] D. Sever, L. Zhao, N. Dellaert, E. Demir, T.V. Woensel and T.D. Kok, The dynamic shortest path problem with time-dependent stochastic disruptions. *Trans. Res. Part C: Emerg. Technol.* **92** (2018) 42–57.

[58] C.E. Sigal, A.A.B. Pritsker and J.J. Solberg, The stochastic shortest route problem. *Oper. Res.* **28** (1980) 1122–1129.

[59] A. Tajdin, I. Mahdavi, N. Mahdavi-Amiri and B. Sadeghpour-Gildeh, Computing a fuzzy shortest path in a network with mixed fuzzy lengths using α-cut. *Comput. Math. App.* **60** (2010) 989–1002.

[60] H. Yu and D. Bertsekas, On boundedness of Q-learning iterates for stochastic shortest path problems. *Math. Oper. Res.* **38** (2013) 209–227.

[61] Z. Zero, C. Bersani, M. Paolucci and R. Sacile, Two new approaches for the bi-objective shortest path with a fuzzy objective applied to HAZMAT transportation. *J. Hazard. Mater.* **375** (2019) 96–106.

[62] J. Zhang, J. Zhuang and B. Behlendorf, Stochastic shortest path network interdiction with a case study of Arizona–Mexico border. *Reliab. Eng. Syst. Saf.* **179** (2018) 62–73.