



Solving the shortest path problem in an imprecise and random environment

V P SINGH^{1,*}, KIRTI SHARMA¹ and DEBJANI CHAKRABORTY²

¹Department of Mathematics, VNIT Nagpur, Nagpur, India

²Department of Mathematics, IIT Kharagpur, Kharagpur, India

e-mail: vpsingh@mth.vnit.ac.in

MS received 2 August 2019; revised 25 February 2020; accepted 17 March 2020; published online 4 July 2020

Abstract. This paper considers a shortest path problem in an imprecise and random environment. The edges in the network represent the approximate time required to cover the distance from one vertex to another vertex while the traffic conditions change randomly for each edge. The approximate time has been defined by using trapezoidal fuzzy number whereas the traffic conditions has been defined in linguistic term. Such type of network problem can be called as Fuzzy Stochastic Shortest Path Problem (FSSPP) in imprecise and random environment. In order to solve the model, a method has been proposed based on the Dijkstra's algorithm and some numerous example have been solved to present its effectiveness.

Keywords. Shortest path problem; Fuzzy random arc; Dijkstra's algorithm; Network; Trapezoidal fuzzy number.

1. Introduction

In a network, the problem of obtaining the shortest path from a designated source vertex to the other vertices is a basic problem that arises in various real-life applications. It produces necessary learning in shipping, routing, and communications purposes. So the shortest path problem has been examined widely in the areas of transportation, operation research and logistics and so on. The solution of deterministic variant of this problem is simply achievable. Several effective algorithms have been provided by Bellman, Dijkstra's [1, 9, 10] to solve the standard shortest problem.

In this work, we consider a network where the arc lengths denote the time required to cover the arc length while the traffic condition on the particular arc is changing randomly. As the time required to cover the distance with the random change in traffic condition can not be defined precisely. A trapezoidal fuzzy number has been taken to define the time required to cover each arc length in the network. On the other hand, the traffic condition for each arc has been defined linguistically as low, medium and high. To solve the proposed problem, a matrix storing traffic condition for each arc has been defined. In the

literature, the shortest path problem in the imprecise environment has been studied by many of the researchers. They used the fuzzy number to define the imprecise arc length and proposed various methods to solve fuzzy shortest path problem using ranking methods. Some of the references [2, 17, 20, 23, 25] can be found related to the fuzzy shortest path problem. Whereas the shortest path problem in a random environment has been termed as stochastic shortest path problem. In this case, the probability theory has been used to tackle the randomness. The following references [11, 22, 27, 32] can be found in the literature which solves the stochastic shortest path problem. However, it may not be possible always that the impreciseness and randomness [31] occur separately in the shortest path problem. To the best of author's knowledge shortest path problem in the mixed environment has not been found in the literature. In real life situation, the problem of ambulance finding the shortest path to the hospital while covering the arcs with random traffic condition in a network is a problem of mixed environment where the impreciseness and randomness occur simultaneously. Inspired by [28–30] such type of problem can be formulated as fuzzy stochastic shortest path problem. To solve the fuzzy stochastic shortest path problem, Dijkstras' algorithm has been proposed for which the system inputs are taken as the time taken to cover any arc in a given traffic condition which is in the form of trapezoidal fuzzy number and the network connecting the source node and destination node. Whereas the user inputs are: the source node, the destination node, the traffic conditions on every arc of the given network. The output of the system is

*For correspondence

the shortest path from the source vertex to destination vertex.

This paper is organized as follows. In section 2, some basic definitions regarding fuzzy sets and fuzzy numbers are reviewed. The method to compare two trapezoidal fuzzy numbers has also been discussed in this section and definitions of fuzzy random variables and their expectation with an example has also been illustrated. In section 3, we have introduced a mathematical model of shortest path problem in an imprecise and random environment. In section 4, a descriptive algorithm of the proposed method is given. This section also comprises of the flow chart of the given algorithm. In section 5, we have presented two different example where the adjacency matrices remains same as well as the source node and the destination node remains same but a slight change in the traffic conditions of the network results in the change of the shortest path. The last section comprises of the conclusion.

2. Preliminaries

2.1 Fuzzy Set [17]

Definition 1 If X is a universe of discourse and x is a particular element of X , then a fuzzy set A defined on X can be written as collection of ordered pairs,

$$A = \{(x, \mu_A(x)), x \in X\} \tag{1}$$

where

$$\mu_A(x) : X \rightarrow [0, 1].$$

2.2 Fuzzy Number [35]

Definition 2 A fuzzy set \tilde{A} on \mathbb{R} is said to be a fuzzy number [35], if the following three properties are satisfied:

1. Fuzzy set \tilde{A} must be normal, i.e. $\exists x$ such that $\sup \mu_{\tilde{A}}(x) = 1$, where \sup stands for supremum.
2. The support of fuzzy set, i.e. set of all the elements with non zero degree of membership, must be bounded.
3. α level set i.e. set of all the elements with membership degree greater than α , must be a closed interval for $\alpha \in [0,1]$.

2.3 Trapezoidal Fuzzy Number

Definition 3 A fuzzy number is said to be a trapezoidal fuzzy number[16] if the graph of its membership function is trapezium. The membership function of a trapezoidal fuzzy number is defined by Eq. (2) and the membership function of such a trapezoidal fuzzy number is represented by Figure 1.

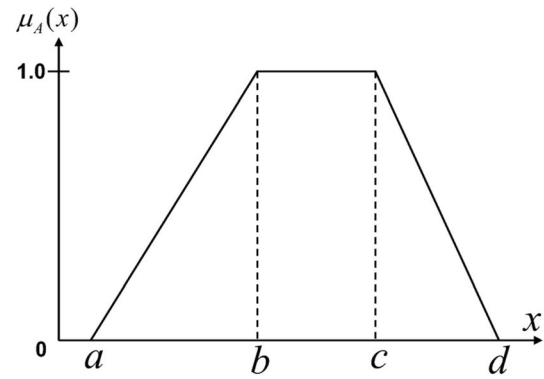


Figure 1. A Trapezoidal Fuzzy Number.

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & \text{if } x < a \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ 1 & \text{if } b \leq x \leq c \\ \frac{d-x}{d-c} & \text{if } c \leq x \leq d \\ 0 & \text{if } x > d \end{cases} \tag{2}$$

This trapezoidal fuzzy number is denoted as (a, b, c, d)

2.4 Operations on Trapezoidal Fuzzy numbers[16]

Let $\tilde{A}=(a,b,c,d)$ and $\tilde{B}=(p,q,r,s)$ are two trapezoidal fuzzy numbers, then

• **Addition:**

$$(a, b, c, d) \oplus (p, q, r, s) = (a + p, b + q, c + r, d + s). \tag{3}$$

• **Subtraction:**

$$(a, b, c, d) \ominus (p, q, r, s) = (a - s, b - r, c - q, d - p). \tag{4}$$

• **Scalar Multiplication:**

$$k(p, q, r, s) = (kp, kq, kr, ks). \tag{5}$$

2.5 Defuzzy Value of A Trapezoidal Fuzzy numbers [3]

Definition 4 The defuzzy value of a trapezoidal fuzzy number[4] $\tilde{A} = (a,b,c,d)$ is given by Eq. (6)

$$\begin{aligned}
 Val(\tilde{A}) &= \int_0^1 \alpha(A_d(x) + A_a(x))dx \\
 &= \int_0^1 \alpha((d - (d - c)\alpha) + (a + \alpha(b - a)))dx \\
 &= \int_0^1 \alpha(d + a) + \alpha^2(b - a - d + c)dx \\
 &= \frac{d + a}{2} + \frac{b - a - d - c}{3} \\
 &= \frac{a}{6} + \frac{b}{3} + \frac{c}{3} + \frac{d}{6}
 \end{aligned}$$

Thus,

$$Val(\tilde{A}) = \frac{a}{6} + \frac{b}{3} + \frac{c}{3} + \frac{d}{6} \tag{6}$$

2.6 Comparison of Trapezoidal Fuzzy numbers

Definition 5 Based on the defuzzy values of two trapezoidal fuzzy numbers, \tilde{A} and \tilde{B} , they can be compared in the following manner:

$$\begin{aligned}
 \text{If } Val(\tilde{A}) > Val(\tilde{B}) & \text{ then } \tilde{A} > \tilde{B} \\
 \text{If } Val(\tilde{A}) < Val(\tilde{B}) & \text{ then } \tilde{A} < \tilde{B} \\
 \text{If } Val(\tilde{A}) = Val(\tilde{B}) & \text{ then } \tilde{A} = \tilde{B}
 \end{aligned} \tag{7}$$

2.7 Fuzzy Linguistic variables [35]

Definition 6 There are linguistic variables whose states are fuzzy numbers. Fuzzy variables representing linguistic concepts such as hot, cold, pleasant etc. with respect to temperature are known as fuzzy linguistic variables.

The Figure 2 represents the fuzzy linguistic variable cool, pleasant and hot with respect to the environment temperature.

2.8 Fuzzy Random Variable [18]

Definition 7 A fuzzy random variable[26] is a random variable whose value is not real, but a fuzzy number.

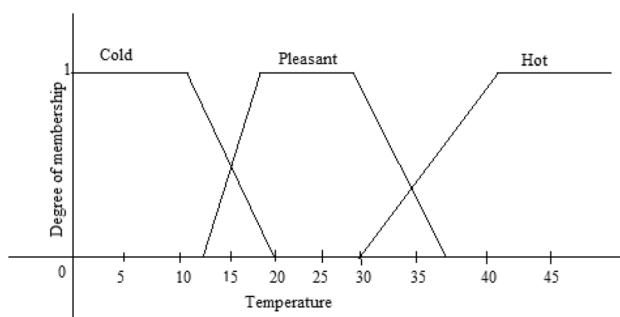


Figure 2. Fuzzy linguistic variable.

Example 1 Consider a survey where individuals are questioned about the customer services of a particular company. The responses of customers are classified into three categories:

1. Disappointing
2. Satisfactory
3. Amazing

In the above case, randomness occurs because the opinion of customers are not known. Once the opinion of customer is available, there is still some uncertainty about the accurate meaning of the response. This lack of preciseness denotes the fuzziness of random variable *customer services*.

2.9 Expectation of a Fuzzy Random Variable

Definition 8 Let \tilde{X} be a fuzzy random variable. Then the expectation of \tilde{X} is defined as fuzzy number $E\tilde{X}$, which can be formally defined as

$$E[\tilde{X}] = \sum_{\tilde{x} \in \tilde{X}} \tilde{x} \odot p(\tilde{x}) \tag{8}$$

where $p(\tilde{x})$ denotes the probability of fuzzy random variable \tilde{X} taking the value \tilde{x}

3. Impreciseness and Randomness in shortest path problems

In the case of shortest path problem[9, 15], the motive is to find a shortest path from the source node to the destination node in such a way that the sum of weights of constituent edges is minimized. In classical shortest path problem, the weight of the edge represents the distance of that edge or time taken to cover that edge, which is always given in crisp form. But in day to day life, situations are not always defined in the crisp form. Time taken to cover a path is not always crisp. One may take more time to cover a path depending upon various conditions. In the case of impreciseness and randomness in shortest path problem, if we take the weight of the edge to be the approximate time taken to cover that edge, this approximate time will never be a real number (crisp). It will always be something imprecise given in the form of fuzzy number. In real life situations, to go from one node to another node we follow different paths and those different paths may have different traffic conditions. These different traffic conditions are random in nature. In the mathematical modeling of the shortest path problem in an imprecise and random environment, our objective is to find the path with minimum weight from the source vertex to destination vertex with the assumption that formation of such a path is possible.

A function \tilde{f}_{ij} denotes the fuzzy random variable for each edge e_{ij} joining the vertex i and j . This sample space for this

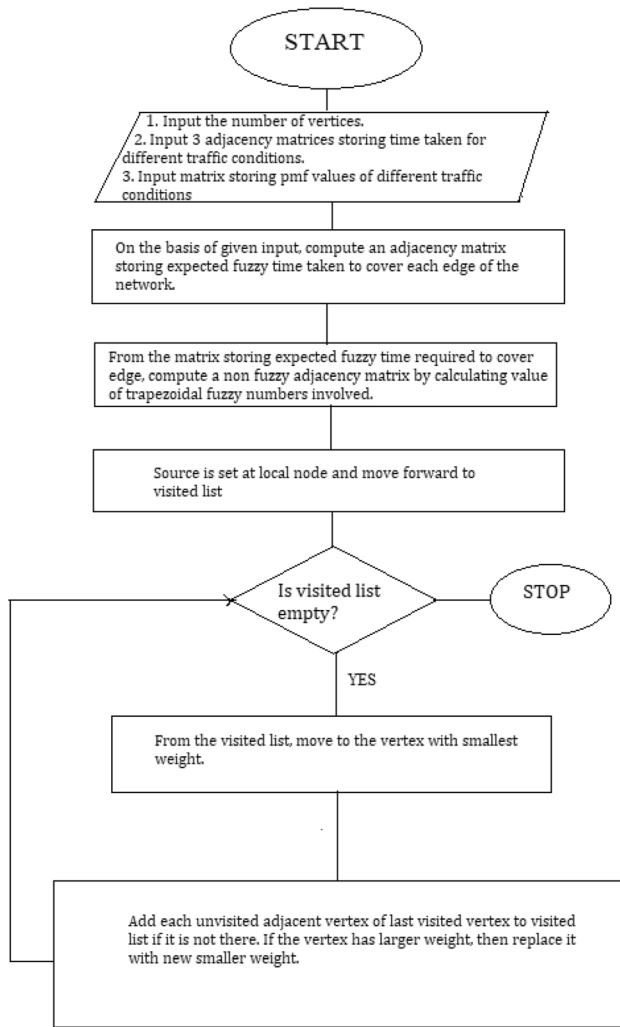


Figure 3. Flowchart of the proposed method.

Table 1. Symbols used in the algorithm with their descriptions.

| Symbols | Description |
|-----------------|--|
| t-l [] [] [] | Adjacency matrix with low traffic conditions |
| t-u [] [] [] | Adjacency matrix with usual traffic conditions |
| t-h [] [] [] | Adjacency matrix with high traffic conditions |
| p [] [] [] | Matrix storing pmf values for different traffic conditions |
| n | Number of Vertices |
| f-t [] [] [] | Fuzzy Adjacency matrix |
| G [] [] [] | Crisp Adjacency matrix |
| Q | List of all unvisited vertices |
| source | source node |

fuzzy random variable include the different types of traffic conditions and this fuzzy random variable maps each traffic condition with the time taken (here, it is expressed in form of trapezoidal fuzzy number) to cover a certain edge under those traffic conditions. This introduces randomness as well as impreciseness in the model.

$$\tilde{f}_{ij}(D) \rightarrow \tilde{F}_K(R)$$

The domain is the set of traffic conditions. $D = \{\text{High, Low, Usual}\}$. The range set is the set of fuzzy number which informs about approximate time taken to cover that edge in a given traffic condition.

3.1 Mathematical model

Objective: Minimize: $\sum E[\tilde{f}_{ij}] \cdot a_{ij}$
 where, a_{ij} is an indicator function.

$$a_{ij} = \begin{cases} 1 & \text{if edge } ij \text{ is included in path} \\ 0 & \text{otherwise} \end{cases}$$

subject to:

$$\sum_j a_{ij} - \sum_j a_{ji} = \begin{cases} 1 & \text{if } i \text{ is source node} \\ -1 & \text{if } j \text{ is destination node} \\ 0 & \text{otherwise} \end{cases}$$

The function \tilde{f}_{ij} satisfies a specific distribution or the distribution can be already provided. The major intuition behind the indicator function, a_{ij} is to find out whether the edge (i, j) is a part of shortest path or not. We wish to select the set of edges with minimal weight, subject to the constraint that this set forms a path from the source node to destination node.

4. Methodology

Suppose we are given a network that consists of a path from the source node to the destination node. Three adjacency matrices storing the information about the time taken to cover every edge in three different traffic conditions, namely, high, low and usual are also given and corresponding to every edge, a probability mass function stores the probability values of different traffic conditions of that edge, e.g. let $p_{ij} = (a, b, c)$ be the probability mass function storing the information that a, b and c corresponds to the probabilities of having low, medium and high traffic densities respectively on edge ij . A matrix \tilde{D} storing the pmf values for different edges is also provided and the objective is to find a shortest path from source node to a destination node. Since there is no unique explanation about finding a shortest path under uncertainty. One possible way to do this

could be to find a path with minimum expected time i.e. the objective is to find a path with minimum expected time.

4.1 Procedure

Step 1: With the help of given three adjacency matrices and probability mass function corresponding to every edge, assign a fuzzy number to every edge of the network, where the fuzzy number represents the expected time taken to cover that edge.

Step 2: Now, we have obtained a network whose weights are fuzzy and the aim is to find the shortest path for this network. Now, these fuzzy weights can be defuzzified by replacing the fuzzy number with the defuzzy value of the given fuzzy number as stated in Eq. (6).

Step 3: Dijkstra method [6] can now be used to solve the fuzzy stochastic shortest path model.

Step 4: Assign the source vertex a permanent label of 0 and all other vertices are assigned a temporary label of ∞ .

Step 5: Starting from the source vertex, search for all the adjacent vertices and the vertex with minimum weight is given a permanent label.

Step 6: Explore the vertices which are adjacent to the vertices with permanent label and repeat Step 5 until we reach the destination vertex.

System Inputs System Inputs are the inputs which are given to the system to solve a particular problem. Here, the system inputs are:

1. The time taken to cover every arc in any given traffic conditions which is in the form of a trapezoidal fuzzy number.
2. The network connecting the source node and destination node.

User Inputs The User Inputs are the inputs which are given by the user and on the basis of these inputs only, the system gives the best suitable path. In our example, the user inputs are:

1. The source node.
2. The destination node.
3. A probability mass function indicating the probabilities of various traffic conditions on every edge.

Output The output of the system is a shortest path from source node to destination node i.e. a path such that the sum of the constituent edges is minimum. The method also gives us vague information about the time taken to reach destination node from the source node.

4.2 Flowchart

The flowchart of the model is represented by Figure 3.

4.3 Algorithm

In the given algorithm, the symbols along with their descriptions has been provided in Table 1.

Algorithm 1 An algorithm for finding a weighted graph corresponding to given information

Input: The inputs for this algorithm are:

- Adjacency matrices corresponding to different traffic conditions
- Matrix storing pmf values of different traffic conditions for a given network

Output: Adjacency matrix storing expected time taken (crisp) to cover every edge

```

1: Start.
2: Declare matrix t-l[n][n][k], t-u[n][n][k], t-h[n][n][k],
   p[n][n][2], number of vertices=n;
3: Declare matrix f-t[n][n][k], G[n][n];
4: Declare variables i, j, k
5: i ← 0
6: j ← 0
7: k ← 0
8: for all i < n do
9:   for all j < n do
10:    for all k < 4 do
11:      f-t ← p[i][j][0].t-l+p[i][j][1].t-u +p[i][j][2].t-h
12:      k ← k + 1
13:    end for
14:    j ← j + 1
15:  end for
16:  i ← i + 1
17: end for
18: i ← 0
19: j ← 0
20: for all i < n do
21:   for all j < n do
22:     G[i][j] :=  $\frac{f-t[i][j][0]+2*f-t[i][j][1]+2*f-t[i][j][2]+f-t[i][j][3]}{6}$ 
                j ← j + 1
23:   end for
                i ← i + 1
24: end for
25: return G[i][j]

```

Algorithm 2 An algorithm for calculating shortest distance from source vertex to the destination vertex

Input: Matrix Graph[i][j] from Eq. 25 of Algorithm 1, Source node and Destination node.

Output: Shortest distance from source vertex to destination vertex

```

1: Declare Graph[i][j], source node, Q.
2: dist[source] ← 0
3: for all i < n do
4:   if i ≠ source then
5:     dist[i] ← infinity
6:   end if
7:   add v to Q
8:   i ← i + 1
9: end for
10: while Q is not empty do
11:   u ← vertex in Q with min dist[i]
12:   remove u from Q
13:   for all v = u + 1 do
14:     alt ← dist[v] + length(v, u)
15:     if alt < dist[u] then
16:       dist[u] ← alt
17:     end if
18:   end for
19: end while
20: dist[destination] is the required distance.
21: return dist[destination]

```

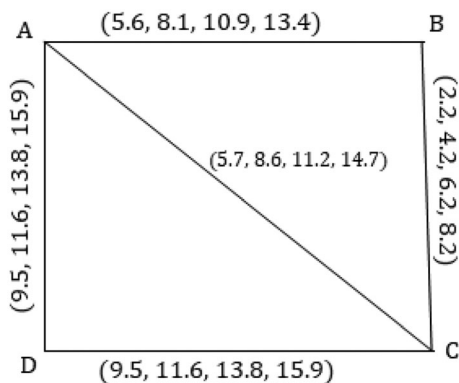


Figure 4. A network with fuzzy edge weights.

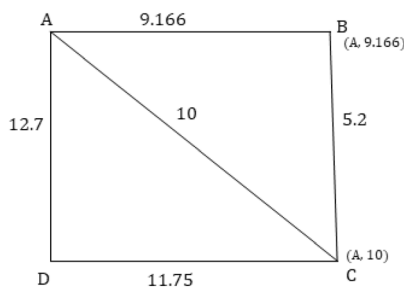


Figure 5. A network with crisp edge weights.

5. Numerical example

5.1 Example 1

User Input The inputs given by user will be in the form of a source node and a destination node and in this case user will also be entering about the traffic conditions on the different arcs. For current example, the user inputs are:

- The source node i.e. node A.
- The destination node i.e. node C.
- The matrix representing the probability mass function of every arc of the network.

Let \tilde{A} , \tilde{B} and \tilde{C} are three fuzzy matrices storing the fuzzy time taken to cover each vertices when the traffic conditions are low, usual and high respectively and a matrix storing the probability mass function values of different traffic condition values on different edges of the network is also given. Our aim is to find the shortest path from the source vertex, i.e. A, to the destination vertex, i.e. C.

$$\tilde{A} = \begin{bmatrix} [0,0,0,0] & [2,4,6,8] & [4,7,9,13] & [5,7,9,11] \\ [2,4,6,8] & [0,0,0,0] & [1,3,5,7] & [0,0,0,0] \\ [4,7,9,13] & [1,3,5,7] & [0,0,0,0] & [1,4,6,9] \\ [5,7,9,11] & [0,0,0,0] & [1,4,6,9] & [0,0,0,0] \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} [0,0,0,0] & [4,6,9,11] & [7,10,14,17] & [12,14,16,18] \\ [4,6,9,11] & [0,0,0,0] & [2,4,6,8] & [0,0,0,0] \\ [7,10,14,17] & [2,4,6,8] & [0,0,0,0] & [8,10,14,16] \\ [12,14,16,18] & [0,0,0,0] & [8,10,14,16] & [0,0,0,0] \end{bmatrix}$$

$$\tilde{C} = \begin{bmatrix} [0,0,0,0] & [8,11,14,17] & [12,14,16,18] & [15,18,22,25] \\ [8,11,14,17] & [0,0,0,0] & [3,5,7,9] & [0,0,0,0] \\ [12,14,16,18] & [3,5,7,9] & [0,0,0,0] & [12,17,19,23] \\ [15,18,22,25] & [0,0,0,0] & [12,17,19,23] & [0,0,0,0] \end{bmatrix}$$

If the matrix storing pmf values, i.e. probability values of different traffic conditions on the network is given by \tilde{D}_1 , then the shortest path is A-C, i.e. via the diagonal.

$$\tilde{D}_1 = \begin{bmatrix} (0,0,0) & (0.2,0.3,0.5) & (0.6,0.3,0.1) & (0.4,0.5,0.1) \\ (0.2,0.3,0.5) & (0,0,0) & (0.1,0.6,0.3) & (0,0,0) \\ (0.6,0.3,0.1) & (0.1,0.6,0.3) & (0,0,0) & (0.3,0.4,0.3) \\ (0.4,0.5,0.1) & (0,0,0) & (0.3,0.4,0.3) & (0,0,0) \end{bmatrix}$$

A detailed solution is given below:

Step 1: First we construct a network, where each edge has a fuzzy weight representing the expected fuzzy time taken to cover that edge. The Figure 4 corresponds to this information.

Step 2: Now we construct a network where the edge represents the defuzzy value of the trapezoidal fuzzy numbers given in the above network. The Figure 5 corresponds to this information.

Step 3: Now we have obtained a weighted graph, same as we get in the classical shortest path problem. So,

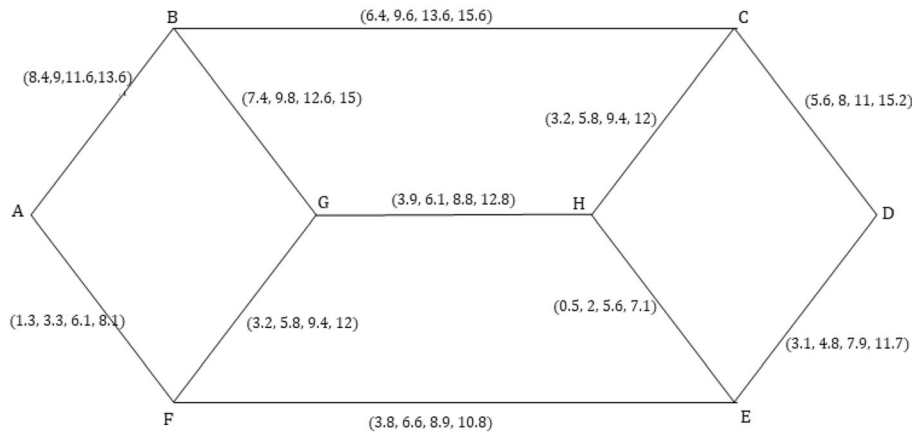


Figure 6. A network of 8 vertices with fuzzy edges weights corresponding to pmf values given by \tilde{D}_2 .

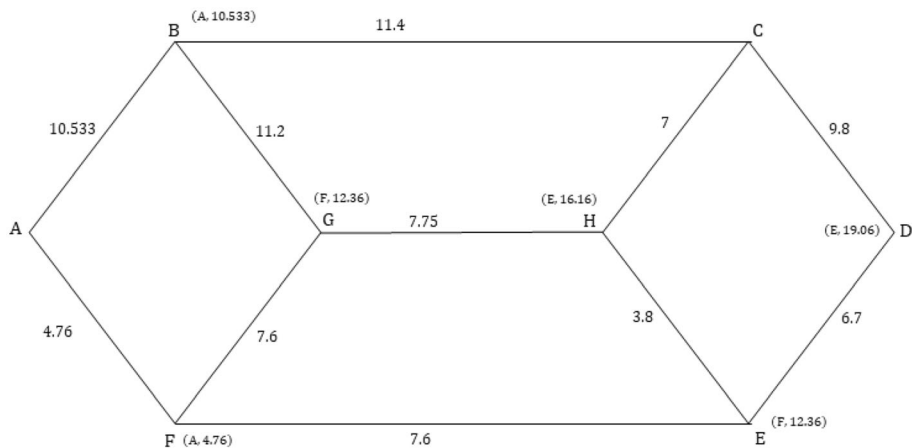


Figure 7. A network of 8 vertices with crisp edges weights.

Dijkstra’s method can be easily applied to solve this shortest path problem.

Step 4: We assign a permanent label of 0 to the source vertex and every other edge is assigned a temporary label of ∞ .

Step 5: Out of all the adjacent vertices, the vertex with minimum weight is B. We make the temporary label of B

permanent and now we look at only those vertices which are adjacent to either A or B.

Step 6: Now, the vertex C adjacent to A via the edge A-C gives the minimum weight and hence this vertex gets a permanent label.

Since we have reached the destination vertex. So, the process stops here. Hence, the shortest path from source vertex to destination vertex is the path A-C, i.e. via the diagonal and the weight of that path is calculated as 10.

Table 2. Comparison of various methods.

| Method | Length of the shortest path |
|-----------------------------|------------------------------|
| Kung’s method [5] | (8.2, 14.7, 22.9, 30.6) |
| Lin’s method [21] | 19.06 |
| Mukherjee’s Method [23] | (20.400, 26.66, 36.2, 44.44) |
| Niroomand’s method [24] | (8.2, 14.7, 22.9, 30.6) |
| Fuzzy Dijkstra’s method [7] | 19.06 |
| Proposed method | 19.06 (expected value) |

5.2 Example 2

Let \tilde{A} , \tilde{B} and \tilde{C} are three matrices storing the time taken to cover each vertices when the traffic conditions are low, usual and high respectively and a matrix storing the probability mass function values of different traffic conditions on different edges of the network is also given. The aim is to find the shortest path from the source vertex i.e. A to the destination vertex i.e. D.

$$\tilde{A} = \begin{bmatrix} [0, 0, 0, 0] & [2, 4, 8, 10] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [1, 3, 7, 9] & [0, 0, 0, 0] & [0, 0, 0, 0] \\ [2, 4, 8, 10] & [0, 0, 0, 0] & [3, 5, 7, 9] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [4, 6, 8, 10] & [0, 0, 0, 0] \\ [0, 0, 0, 0] & [3, 5, 7, 9] & [0, 0, 0, 0] & [2, 6, 10, 14] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 1, 5, 6] \\ [0, 0, 0, 0] & [0, 0, 0, 0] & [2, 6, 10, 14] & [0, 0, 0, 0] & [3, 4, 7, 11] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] \\ [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [3, 4, 7, 11] & [0, 0, 0, 0] & [1, 5, 8, 9] & [0, 0, 0, 0] & [1, 3, 5, 7] \\ [1, 3, 7, 9] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [1, 5, 8, 9] & [0, 0, 0, 0] & [2, 4, 8, 10] & [0, 0, 0, 0] \\ [0, 0, 0, 0] & [4, 6, 8, 10] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [2, 4, 8, 10] & [0, 0, 0, 0] & [3, 4, 7, 11] \\ [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 1, 5, 6] & [0, 0, 0, 0] & [1, 3, 5, 7] & [0, 0, 0, 0] & [3, 4, 7, 11] & [0, 0, 0, 0] \end{bmatrix}$$

$$\tilde{B} = \begin{bmatrix} [0, 0, 0, 0] & [8, 10, 12, 14] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [1, 3, 5, 7] & [0, 0, 0, 0] & [0, 0, 0, 0] \\ [8, 10, 12, 14] & [0, 0, 0, 0] & [4, 9, 16, 18] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [6, 8, 10, 12] & [0, 0, 0, 0] \\ [0, 0, 0, 0] & [4, 9, 16, 18] & [0, 0, 0, 0] & [6, 8, 11, 16] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [4, 7, 9, 12] \\ [0, 0, 0, 0] & [0, 0, 0, 0] & [6, 8, 11, 16] & [0, 0, 0, 0] & [2, 6, 10, 14] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] \\ [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [2, 6, 10, 14] & [0, 0, 0, 0] & [5, 7, 9, 11] & [0, 0, 0, 0] & [0, 1, 5, 6] \\ [1, 3, 5, 7] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [5, 7, 9, 11] & [0, 0, 0, 0] & [2, 6, 10, 14] & [0, 0, 0, 0] \\ [0, 0, 0, 0] & [6, 8, 10, 12] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [2, 6, 10, 14] & [0, 0, 0, 0] & [4, 7, 9, 12] \\ [0, 0, 0, 0] & [0, 0, 0, 0] & [4, 7, 9, 12] & [0, 0, 0, 0] & [0, 1, 5, 6] & [0, 0, 0, 0] & [4, 7, 9, 12] & [0, 0, 0, 0] \end{bmatrix}$$

$$\tilde{C} = \begin{bmatrix} [0, 0, 0, 0] & [10, 12, 16, 18] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [4, 6, 8, 10] & [0, 0, 0, 0] & [0, 0, 0, 0] \\ [10, 12, 16, 18] & [0, 0, 0, 0] & [13, 15, 17, 19] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [12, 14, 18, 20] & [0, 0, 0, 0] \\ [0, 0, 0, 0] & [13, 15, 17, 19] & [0, 0, 0, 0] & [8, 10, 12, 14] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [4, 7, 9, 12] \\ [0, 0, 0, 0] & [0, 0, 0, 0] & [8, 10, 12, 14] & [0, 0, 0, 0] & [6, 8, 10, 12] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] \\ [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [6, 8, 10, 12] & [0, 0, 0, 0] & [5, 8, 10, 13] & [0, 0, 0, 0] & [1, 3, 7, 9] \\ [4, 6, 8, 10] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [5, 8, 10, 13] & [0, 0, 0, 0] & [8, 11, 13, 16] & [0, 0, 0, 0] \\ [0, 0, 0, 0] & [12, 14, 18, 20] & [0, 0, 0, 0] & [0, 0, 0, 0] & [0, 0, 0, 0] & [8, 11, 13, 16] & [0, 0, 0, 0] & [5, 8, 11, 16] \\ [0, 0, 0, 0] & [0, 0, 0, 0] & [4, 7, 9, 12] & [0, 0, 0, 0] & [1, 3, 7, 9] & [0, 0, 0, 0] & [6, 8, 11, 16] & [0, 0, 0, 0] \end{bmatrix}$$

If the matrix storing pmf values, i.e. probability values of different traffic conditions on the network is given by D_2 , then the shortest path is via the lower outer boundary i.e. via the path A-F-E-D. **Step 1:** First we construct the net-

the above network. Figure 7 corresponds to this information.

Step 3: Now we have obtained a weighted graph same as we get in the classical shortest path problem. Now we

$$\tilde{D} = \begin{bmatrix} (0, 0, 0) & (0.2, 0.7, 0.1) & (0, 0, 0) & (0, 0, 0) & (0, 0, 0) & (0.4, 0.5, 0.1) & (0, 0, 0) & (0, 0, 0) \\ (0.2, 0.7, 0.1) & (0, 0, 0) & (0.3, 0.4, 0.3) & (0, 0, 0) & (0, 0, 0) & (0, 0, 0) & (0.5, 0.3, 0.2) & (0, 0, 0) \\ (0, 0, 0) & (0.3, 0.4, 0.3) & (0, 0, 0) & (0.2, 0.6, 0.2) & (0, 0, 0) & (0, 0, 0) & (0, 0, 0) & (0.2, 0.3, 0.5) \\ (0, 0, 0) & (0, 0, 0) & (0.2, 0.6, 0.2) & (0, 0, 0) & (0.7, 0.2, 0.1) & (0, 0, 0) & (0, 0, 0) & (0, 0, 0) \\ (0, 0, 0) & (0, 0, 0) & (0, 0, 0) & (0.7, 0.2, 0.1) & (0, 0, 0) & (0.3, 0.5, 0.2) & (0, 0, 0) & (0.2, 0.5, 0.3) \\ (0.4, 0.5, 0.1) & (0, 0, 0) & (0, 0, 0) & (0, 0, 0) & (0.3, 0.5, 0.2) & (0, 0, 0) & (0.6, 0.2, 0.2) & (0, 0, 0) \\ (0, 0, 0) & (0.5, 0.3, 0.2) & (0, 0, 0) & (0, 0, 0) & (0, 0, 0) & (0.6, 0.2, 0.2) & (0, 0, 0) & (0.4, 0.3, 0.3) \\ (0, 0, 0) & (0, 0, 0) & (0.2, 0.3, 0.5) & (0, 0, 0) & (0.2, 0.5, 0.3) & (0, 0, 0) & (0.4, 0.3, 0.3) & (0, 0, 0) \end{bmatrix}$$

work where each edge has a fuzzy weight representing the expected fuzzy time taken to cover that edge. Figure 6 corresponds to this information.

Step 2: Now we obtain a network where the edge represents the value of the trapezoidal fuzzy numbers given in

can apply Dijkstra method to solve this shortest path problem.

Step 4: We assign a permanent label of 0 to the source vertex and every other edge is assigned a temporary label of ∞ .

Step 5: Out of all the adjacent vertex, the vertex having minimum weight is F. We make the temporary label of F permanent and now we look at only those vertices which are adjacent to A or F.

Step 6: The vertex B adjacent to A via the edge A-B gives the minimum weight and hence this vertex gets a permanent label. Now, we look for the vertices which are adjacent to A, B or F. The vertex G(or E) adjacent to F via the edge F-G gives the minimum weight and hence G gets the permanent label. Now, we search for the vertices adjacent to B, G or F. This time the vertex E via the edge F-E gives the minimum weight, same as that of G and hence gets the permanent label. On further solving, vertex H gets permanent label via edge E-H and after that finally our destination vertex D gets permanent label via the edge E-D.

Hence, the shortest path from source vertex to destination vertex is via the lower outer boundary i.e. from A-F-E-D and weight of this path is calculated as 19.06 units.

5.3 Discussion and analysis

The presented method to find the path of Shortest path Problem in imprecise and random environment is based on Dijkstra's algorithm, which is a greedy approach, i.e. in this method, the traveler keeps on choosing the path with lowest expected weight without having a visionary approach. The method always guarantees an optimal solution, when all the edge weights are positive which is the case considered in this work. In other classical methods used for solving fuzzy shortest path problem, issues like random nature of the network because of various conditions on the network have not been discussed. In this work, randomness has also been dealt by fuzzy random variable. Expectation of fuzzy random variable has been used to convert the given imprecise and random network into the corresponding fuzzy network, where edge weights are given by fuzzy numbers. There are several methods in the literature for solving fuzzy shortest path problem which uses the linear programming approach like an extension principle based approach by Niroomand et. al [24] and [23]. The time complexity of the methods based on linear programming approach is exponential whereas the time complexity of the proposed method is polynomial. The method presented earlier either have time complexity or optimality issues whereas the proposed method overcome both the issues with consideration of impreciseness and randomness of SPP, which has not been discussed earlier. Comparison of the method with other works can only be done once a fuzzy network has been obtained from the corresponding imprecise and random network. Table 2 comprises the comparison of the method presented in this paper with other works. The comparison has been done on the network presented in Example 2.

6. Conclusion

In this work, the mathematical model of a fuzzy stochastic shortest path problem and the algorithm to solve this problem has been introduced. In real life situations, to go from one node to another node, we follow different paths and those different paths have different traffic conditions. These different traffic conditions are random in the shortest path model. In the mathematical modeling of the shortest path problem in an imprecise and random environment, our objective is to find the path with minimum weight from the source vertex to destination vertex with the assumption that formation of such a path is possible. In this work, different traffic conditions occur on different edges with certain probabilities and in those traffic conditions, time taken to cover every vertex is given by trapezoidal fuzzy numbers. Randomness and impreciseness has been dealt in this work by using expectations of fuzzy random variables and ranking of trapezoidal fuzzy numbers respectively. Two examples have also been solved using the proposed approach. This result may be useful to find the shortest path for ambulance problem in a route, when the traffic conditions are changing randomly.

References

- [1] Bellman R 1958 On a routing problem. *Quarterly of applied mathematics*.16(1):87-90.
- [2] Broumi S, Talea M, Bakali A, Smarandache F, Nagarajan D, Lathamaheswari M, Parimala M 2019 Shortest path problem in fuzzy, intuitionistic fuzzy and neutrosophic environment: an overview. *Complex & Intelligent Systems*.5(4):371-8.
- [3] Chiao KP 2016 Jul Ranking interval type 2 fuzzy sets using parametric graded mean integration representation. *In 2016 International Conference on Machine Learning and Cybernetics (ICMLC)* 10 (Vol. 2, pp. 606-611). IEEE.
- [4] Chiao KP 2012 Jun Trapezoidal interval type-2 fuzzy set extension of analytic hierarchy process. *In 2012 IEEE International Conference on Fuzzy Systems* 10 (pp. 1-8). IEEE.
- [5] Chuang TN, Kung JY 2005 The fuzzy shortest path length and the corresponding shortest path in a network. *Computers & Operations Research*.32(6):1409-28.
- [6] Cormen TH, Leiserson CE, Rivest RL, Stein C 2009 *Introduction to algorithms*. MIT press.
- [7] Deng Y, Chen Y, Zhang Y, Mahadevan S 2012 Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Applied Soft Computing*.12(3):1231-7.
- [8] Dey A, Pal A, Pal T 2016 Dec Interval type 2 fuzzy set in fuzzy shortest path problem. *Mathematics*.4(4):62.
- [9] Dijkstra EW 1959 Dec 1 A note on two problems in connexion with graphs. *Numerische mathematik*.1(1):269-71.
- [10] Dreyfus SE 1969 Jun An appraisal of some shortest-path algorithms. *Operations research*.17(3):395-412.
- [11] Frank H 1969 Aug Shortest paths in probabilistic graphs. *Operations Research*.17(4):583-99.

- [12] Gil MA, López-Díaz M, Ralescu DA 2006 Jan 1 Overview on the development of fuzzy random variables. *Fuzzy sets and systems*.157(19):2546-57.
- [13] Goldberg D E 1989 Genetic Algorithm in Search, Optimization and Machine Learning. *Reading: Addison Wesley*.
- [14] Hassanzadeh R, Mahdavi I, Mahdavi-Amiri N, Tajdin A 2013 A genetic algorithm for solving fuzzy shortest path problems with mixed fuzzy arc lengths. *Mathematical and Computer Modelling*. 57(1-2):84-99.
- [15] Horowitz E, Sahni S, Anderson-Freed S 1976 Fundamentals of data structures. Potomac, MD: *Computer science press*.
- [16] Klir G, Yuan B 1995 Fuzzy sets and fuzzy logic. *New Jersey: Prentice hall*.
- [17] Klein CM 1991 Jan 10 Fuzzy shortest paths. *Fuzzy sets and systems*.39(1):27-41.
- [18] Kwakernaak H 1978 Jul 1 Fuzzy random variables—I. Definitions and theorems. *Information sciences*.15(1):1-29.
- [19] Kwakernaak H 1979 Apr 1 Fuzzy random variables—II. Algorithms and examples for the discrete case. *Information Sciences*.17(3):253-78.
- [20] Lin KC, Chern MS 1993 The fuzzy shortest path problem and its most vital arcs. *Fuzzy Sets and Systems*.58(3):343-53.
- [21] Lin FT 2001 Dec 2 A shortest-path network problem in a fuzzy environment. In *10th IEEE International Conference on Fuzzy Systems*. (Cat. No. 01CH37297) (Vol. 3, pp. 1096-1100). IEEE.
- [22] Loui RP 1983 Optimal paths in graphs with stochastic or multidimensional weights. *Communications of the ACM*.26(9):670-6.
- [23] Mukherjee S 2015 Fuzzy programming technique for solving the shortest path problem on networks under triangular and trapezoidal fuzzy environment. *International Journal of Mathematics in Operational Research*.7(5):576-94.
- [24] Niroomand S, Mahmoodirad A, Heydari A, Kardani F, Hadi-Vencheh A 2017 An extension principle based solution approach for shortest path problem with fuzzy arc lengths. *Operational Research*.17(2):395-411.
- [25] Okada S, Soper T 2000 Jan 1 A shortest path problem on a network with fuzzy arc lengths. *Fuzzy sets and systems*.109(1):129-40.
- [26] Puri ML, Ralescu DA, Zadeh L 1993 Jan 1 Fuzzy random variables. In *Readings in fuzzy sets for intelligent systems* (pp. 265-271). Morgan Kaufmann.
- [27] Randour M, Raskin JF, Sankur O 2015 Jan 12 Variations on the stochastic shortest path problem. In *International Workshop on Verification, Model Checking, and Abstract Interpretation* (pp. 1-18). Springer, Berlin, Heidelberg.
- [28] Singh VP, Chakraborty D 2019 Bi-level optimization based on fuzzy if-then rule. *Croatian Operational Research Review*. 13:315-28.
- [29] Singh VP, Chakraborty D 2017 Solving bi-level programming problem with fuzzy random variable coefficients. *Journal of Intelligent & Fuzzy Systems*.32(1):521-8.
- [30] Singh VP 2020 An Approach to Solve Fuzzy Knapsack Problem in Investment and Business Model. In *Networked Business Models in the Circular Economy* (pp. 154-173). IGI Global.
- [31] Singh VP Multi-stage decision making in imprecise and uncertain environment (Doctoral dissertation, IIT, Kharagpur).
- [32] Wei X, Xu K, Jiao P, Yin Q, Zha Y 2019 A Decomposition Approach for Stochastic Shortest-Path Network Interdiction with Goal Threshold. *Symmetry*.11(2):237.
- [33] Yadav K, Biswas R 2009 On searching fuzzy shortest path in a network. *International Journal of Recent Trends in Engineering*. 2(3):16.
- [34] Zhang Y, Zhang Z, Deng Y, Mahadevan S 2013 A biologically inspired solution for fuzzy shortest path problems. *Applied Soft Computing*. 13(5):2356-63.
- [35] Zimmermann HJ 2011 Jun 27 Fuzzy set theory—and its applications. *Springer Science & Business Media*.