ANNALS OF COMMUNICATIONS IN MATHEMATICS
Volume 1, Number 1 (2018), 48-64
ISSN: 2582-0818

FUZZY SHORTEST PATH IN AN INTERVAL-VALUED FUZZY HYPERGRAPH USING SIMILARITY MEASURES

TARASANKAR PRAMANIK* AND MADHUMANGAL PAL

ABSTRACT. In this paper, a new approach is introduced to find the shortest path between two given vertices on interval-valued fuzzy hypergraphs. When only crisp numbers are not sufficient to measure a real world parameter, fuzzy numbers are considered. But, there are many types of fuzzy numbers available in literature. Most useful fuzzy number is trapezoidal fuzzy numbers. Throughout this paper the interval-valued trapezoidal fuzzy number is used as the arc length of the interval-valued fuzzy hypergraph. We have measured similarity between two interval-valued fuzzy numbers to find the shortest path. An algorithm is also designed to find all possible hyperpaths in a hypergraph and calculated its time complexity.

1. INTRODUCTION

The shortest path problem mainly traces on finding minimum distance between two specified vertices in a graph. The fuzzy shortest path problem was first analysed by Duboids and Prade [15] using edge weight as fuzzy number instead of a real number. Later, the concept of degree of possibility in which an arc is a shortest path is introduced by Okada [24]. Takahashi and Yamakami [43] discussed the shortest path problem from a specified node to all other nodes on a network. Chuang and Kung introduced several methods to solve this kind of problem. In [14], they proposed a procedure that can find a fuzzy shortest path among all possible paths in a network. There is another approach to find fuzzy shortest path, namely fuzzy linear programming approach, has been found by Lin and Chern [22]. Nayeem and Pal [23] have designed algorithms to solve fuzzy shortest path problem on a network with imprecise edge weights. Chuang and Kung [13] have introduced a new algorithm for discrete fuzzy shortest path problem in a network. Recently, Kumar et. al. [21] have worked on algorithms to find the shortest path in fuzzy graphs with interval-valued intuitionistic fuzzy edge weights.

Akram and Davvaz [1] discussed strong intuitionistic fuzzy graphs. A novel application of intuitionistic fuzzy digraphs in decision support systems is given in [2]. Akram and Dudek [3] discussed intuitionistic fuzzy hypergraphs and provided an application. Also, Akram and Al-Shehrie defined intuitionistic fuzzy cycles and intuitionistic fuzzy tree [4],

2010 *Mathematics Subject Classification.* 05C85, 05C90, 68R05.
Key words and phrases. fuzzy number; fuzzy hyperpaths; interval-valued fuzzy hypergraph; fuzzy shortest hyperpath; algorithm.
*Corresponding author.

bipolar fuzzy competition graphs [5] and intuitionistic fuzzy planar graphs [6]. Sahoo and Pal [31] discussed the concept of intuitionistic fuzzy competition graph. They also discuss different types of products on intuitionistic fuzzy graphs [30], product of intuitionistic fuzzy graphs and degree [32], etc.

Many researchers have focused on fuzzy shortest path problem in a network due to its importance to many applications such as communications, routing, transportation, etc. In traditional shortest path problems, the arc lengths of the network are taken as precise numbers, but in the real-world problem, the arc length may represent transportation time or cost which can be known only approximately due to vagueness of information, and hence it can be considered as a fuzzy number. Here, we considered a special type of fuzzy numbers namely, trapezoidal fuzzy number.

A hypergraph is a generalization of a graph in which an edge can connect any number of vertices. Formally, a hypergraph H is a pair $H = (X, E)$ where X is a set of elements called nodes or vertices and E is a set of non-empty subsets of X called hyperedges or edges. Therefore, E is a subset of $P(X)\backslash\{\phi\}$, where $P(X)$ is the power set (collection of all subsets) of X. While edges of a graph are pair of nodes or vertices, hyperedges are arbitrary sets of nodes and therefore contain arbitrary number of nodes. A directed hypergraph is a generalization of the concept of directed graph. It was first introduced in [7] to represent functional dependencies in relational data base. A *directed hypergraph* is given by a set of nodes V and a set of pairs (T, h) (*hyperedges*) where T is a subset of V and h is a single node in V. The most obvious interpretation of a hyperedge (T, h) is that the information associated to h functionally depends on the information associated to nodes in T.

A hypergraph is useful in various combinatorial structures that generalize graphs. Directed hypergraph is an extension of directed graphs, and have often used in several areas such as a modelling and algorithmic tool. A brief introduction to directed hypergraphs is given by Gallo et al. [17]. Geotschel [18] introduced the concept of fuzzy hypergraphs and Hebbian structures. Goetschel [19] also explained the coloring of fuzzy hypergraphs. Intersecting fuzzy hypergraphs are defined by Goetschel [20]. Samanta and Pal have also worked on bipolar fuzzy hypergraphs and related fuzzy graphs in [26–28, 33–42]. Readers can found many recent works on [25, 29].

Interval valued fuzzy set is a generalization of traditional fuzzy set. So it is more adequate to describe the uncertainty than the traditional fuzzy sets. It is therefore important to use interval valued fuzzy sets in applications such as fuzzy control, network topology, transportation, etc. When in a fuzzy graph the arc (edge) weights and/or vertex weights are considered as the interval-valued fuzzy sets, the resultant graph becomes an interval-valued fuzzy graph.

There are many real life problems such as communications, routing, transportation, etc., finding shortest path is very essential in research purposes. But finding shortest paths in hypergraphs are too more demanding in recent research areas. Hypergraphs can consider more complex networks such as protein-protein interaction network, social networks, information theory, publication data, collaborations, chemical processes, etc. Hypergraphs are learned to segment or classify the datas. In learning of distances between two nodes in a hypergraph, first create a normal graph by connecting nodes with weighted edges. Weight is the sum of the weights of hyperedges traversed in the shortest path. An example is given here.

Suppose there are seven cities, say, A, B, C, D, E, F, G in a country. The hypergraph relation of these cities are given in Figure 1. We can construct the normal graph

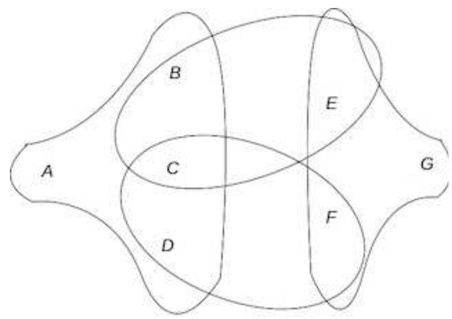

FIGURE 1. Hypergraph of seven cities

by specifying connections to each cities which are connected by roads such as $A \longrightarrow B$, $A \longrightarrow C$, $A \longrightarrow D$, $B \longrightarrow E$, $C \longrightarrow E$, $C \longrightarrow F$, $D \longrightarrow F$, $E \longrightarrow G$, $F \longrightarrow G$. The corresponding graph is shown in Figure 2.

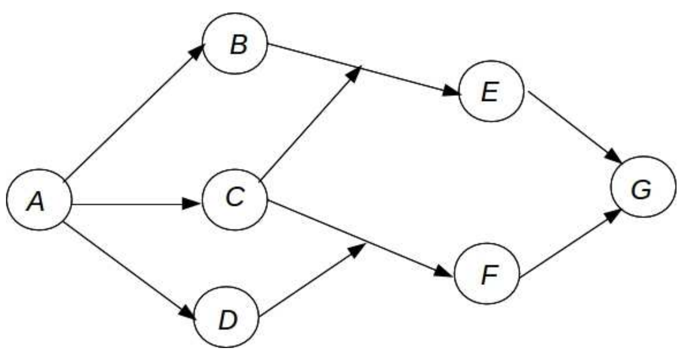

FIGURE 2. Graph of Figure 1

The population of G has demand of some products. The city A has a supplier to supply the products but wants the minimum cost of transportation to supply. Each road has cost of transportaion. Depending on many parameters (such as toll taxes, levies, security taxes, etc.) costs of transportation varies road by road and also time by time. So the costs can be treated as interval-valued trapezoidal fuzzy numbers where left end trapezoidal fuzzy number is the minimum cost and right end trapezoidal fuzzy number is the maximum cost to transport by that road. So this problem can be modelled as the interval-valued fuzzy hypernetwork with edge weights as trapezoidal fuzzy numbers. Being motivated from this essence, an algorithm is designed to find shortest path of interval-valued fuzzy hypernetwork. Though there are many research articles to find shortest paths, here we have generalized the fuzziness by introducing interval-valued fuzzy numbers and trapezoidal fuzzy numbers together. The interval-valued trapezoidal fuzzy numbers are the generalization of interval-valued fuzzy numbers, trapezoidal fuzzy numbers and also fuzzy numbers. All these numbers can be obtained by considering particular cases.

In this paper, we have designed an algorithm to find the fuzzy shortest path on a fuzzy hypernetwork. The membership values of the edges are taken as interval-valued trapezoidal fuzzy number. The remaining part of the paper is organized as follows:

In Section 2, preliminaries of the main work is introduced. In Section 3, an algorithm based on BFS (Breadth-First Search) technique is described to find all the paths between two nodes. Section 4 computes the fuzzy shortest hyperpaths of a network. At the end of the paper, conclusion has been drawn.

2. PRELIMINARIES

In this section, the definition of interval-valued fuzzy hypergraph, directed interval-valued fuzzy hypergraph, interval-valued trapezoidal fuzzy number, similarity measures of two interval-valued trapezoidal fuzzy numbers are given. These are the basic concepts required to design the algorithm to find the fuzzy shortest hyperpath.

A fuzzy set $\mathcal{F}$ on a universal set X is defined by a mapping $m : X \to [0,1]$, which is called the membership function. A fuzzy set is denoted by $\mathcal{F} = (X, m)$. A fuzzy graph $\mathcal{G} = (V, \sigma, \mu)$ is a non-empty set V together with a pair of functions $\sigma : V \to [0,1]$ and $\mu : V \times V \to [0,1]$ such that for all $x, y \in V, \mu(x,y) \leq \min\{\sigma(x), \sigma(y)\}$, where $\sigma(x)$ represents the membership values of the vertex x and $\mu(x,y)$ represents membership values of the edge (x,y) in $\mathcal{G}$. A loop at a vertex x in a fuzzy graph is represented by $\mu(x,x) \neq 0$. An edge is non-trivial if $\mu(x,y) \neq 0$. A fuzzy graph $\mathcal{G} = (V, \sigma, \mu)$ is complete if $\mu(x,y) = \min\{\sigma(x), \sigma(y)\}$ for all $x, y \in V$, where (x,y) denotes the edge between the vertices x and y.

An interval-valued fuzzy set A on a set X is defined by $A = \{(x, [\sigma^L(x), \sigma^U(x)] : x \in X\}$ where the membership functions $\sigma^L(x), \sigma^U(x)$ are such that $\sigma^L(x) \leq \sigma^U(x)$ and $\sigma^L(x), \sigma^U(x) \in [0,1]$ for all $x \in X$.

An interval-valued fuzzy graph ξ is denoted by $\xi = (V, A, B)$, where $A = (V, [\sigma^L, \sigma^U])$ is an interval-valued fuzzy set on V and $B = (V \times V, [\mu^L, \mu^U])$ is an interval-valued fuzzy set on $V \times V$, such that $\mu^L(x,y) \leq \min\{\sigma^L(x), \sigma^L(y)\}$ and $\mu^U(x,y) \leq \min\{\sigma^U(x), \sigma^U(y)\}$ for all $x, y \in V$. We call A as the interval-valued fuzzy vertex set of ξ and B as the interval-valued fuzzy edge set of ξ respectively.

A *hypergraph* is a generalization of a graph in which an edge can connect any number of vertices. Formal definition is as follows:

Definition 2.1 (Hypergraph)**.** Let $X = \{x_1, x_2, \ldots, x_n\}$ be a finite set. A hypergraph on X is a family $H = \{E_1, E_2, \ldots, E_m\}$ of subsets of X such that

(i) $E_i \neq \phi$, $(i = 1, 2, \ldots, m)$ and

(ii) $\displaystyle\bigcup_{i=1}^{m} E_i = X$.

The elements $x_1, x_2, \ldots, x_n$ are called the *vertices* and the sets $E_1, E_2, \ldots, E_m$ are called the *hyperedges* (or, simply *edges*) of the hypergraph.

A *simple hypergraph* is a hypergraph $H = \{E_1, E_2, \ldots, E_m\}$ such that $E_i \subset E_j \Rightarrow i = j$. A graph is a simple hypergraph each of whose edges has cardinality 2.

Let X be a finite set and let E be a family of non-empty fuzzy subsets of X such that $X = \bigcup\{\text{supp}\, A | A \in E\}$. Then the pair $H = (X, E)$ is called a *fuzzy hypergraph* on X.

Let X be a finite set and let E be a family of non-empty interval-valued fuzzy subsets of X such that $X = \bigcup\{\text{supp}\, A | A \in E\}$. Then the pair $H = (X, E)$ is called an *interval-valued fuzzy hypergraph* on X.

In classical graph theory, parameters are measured by a single number e.g., 5 km (for distances), 30 kg (for weights), etc. But in practical situation, a single number could not give enough information of the parameters. It may come in an imprecise way like 'about 5 km', 'between 10-15 yards', etc. In this paper, the arc lengths of a hypergraph are taken as interval number.

In general, an interval number is defined as $I = [a_L, a_R] = \{a : a_L \leq a \leq a_R\}$ where a_L and a_R are the real numbers called the left end point and the right end point of the interval I respectively. Besides interval number there are so many fuzzy numbers viz. triangular fuzzy number, trapezoidal fuzzy number, etc.

A trapezoidal fuzzy number $\widetilde{A} = [a_1, a_2, a_3, a_4; w_{\widetilde{A}}]$, $a_1 \leq a_2 \leq a_3 \leq a_4$ and $0 \leq w_{\widetilde{A}} \leq 1$ is represented by the membership function (shown in Figure 3) as

$$\mu_{\widetilde{A}}(x) = \begin{cases} 0, & x \leq a_1 \\ \frac{x-a_1}{a_2-a_1} w_{\widetilde{A}}, & a_1 < x \leq a_2 \\ w_{\widetilde{A}}, & a_2 < x \leq a_3 \\ \frac{a_4-x}{a_4-a_3} w_{\widetilde{A}}, & a_3 < x \leq a_4 \\ 0, & x \geq a_4. \end{cases}$$

Chen [8, 9] defines the generalized trapezoidal fuzzy number $\widetilde{A} = [a_1, a_2, a_3, a_4; w_{\widetilde{A}}]$, $a_1 \leq a_2 \leq a_3 \leq a_4$ and $0 \leq w_{\widetilde{A}} \leq 1$ where, $w_{\widetilde{A}}$ represents the degree of confidence of the liguistic opinion. Chen and Chen [10] introduces the Simple Center of Gravity Method (SCGM) to calculate the Center of Gravity (COG) point $(x^*_{\widetilde{A}}, y^*_{\widetilde{A}})$ of a generalized trapezoidal fuzzy number as

$$y^*_{\widetilde{A}} = \begin{cases} \frac{w_{\widetilde{A}} \times \left(\frac{a_3-a_2}{a_4-a_1}+2\right)}{6}, & \text{if } a_1 \neq a_4 \\ \frac{w_{\widetilde{A}}}{2}, & \text{if } a_1 = a_4. \end{cases}$$

$$x^*_{\widetilde{A}} = \frac{y^*_{\widetilde{A}}(a_3+a_2) + (a_4+a_1)(w_{\widetilde{A}} - y^*_{\widetilde{A}})}{2w_{\widetilde{A}}}.$$

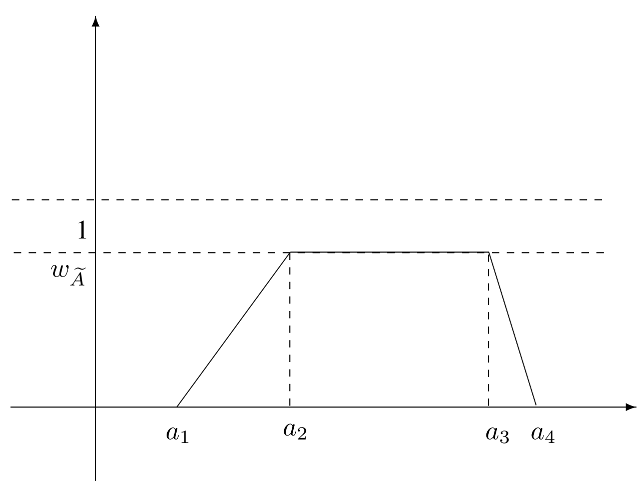

FIGURE 3. Trapezoidal fuzzy number

Yao and Lin [45] studied the interval-valued trapezoidal fuzzy number $\widetilde{\widetilde{A}} = [\widetilde{A}^L, \widetilde{A}^U]$ as shown in Figure 4, where each $\widetilde{A}^L = (a^L_1, a^L_2, a^L_3, a^L_4; w_{\widetilde{A}^L})$ and $\widetilde{A}^U = (a^U_1, a^U_2, a^U_3, a^U_4; w_{\widetilde{A}^U})$ are trapezoidal fuzzy numbers and $\widetilde{A}^L \subseteq \widetilde{A}^U$.

Definition 2.2 (Interval-valued trapezoidal fuzzy number)**.** An interval-valued trapezoidal fuzzy number (IVTFN) $\widetilde{\widetilde{A}}$ is denoted by $\widetilde{\widetilde{A}} = \{(\mu^L_{\widetilde{A}}(x), \mu^U_{\widetilde{A}}(x)) | \mu^L_{\widetilde{A}}(x) \le \mu^U_{\widetilde{A}}(x), x \in X\}$, where

$$\mu^L_{\widetilde{A}}(x) = \begin{cases} 0, & x \le a^L_1 \\ \frac{x - a^L_1}{a^L_2 - a^L_1} w_{\widetilde{A}^L}, & a^L_1 < x \le a^L_2 \\ w_{\widetilde{A}^L}, & a^L_2 < x \le a^L_3 \\ \frac{a^L_4 - x}{a^L_4 - a^L_3} w_{\widetilde{A}^L}, & a^L_3 < x \le a^L_4 \\ 0, & x \ge a^L_4, \end{cases} \quad \text{and } \mu^U_{\widetilde{A}}(x) = \begin{cases} 0, & x \le a^U_1 \\ \frac{x - a^U_1}{a^U_2 - a^U_1} w_{\widetilde{A}^U}, & a^U_1 < x \le a^U_2 \\ w_{\widetilde{A}^U}, & a^U_2 < x \le a^U_3 \\ \frac{a^U_4 - x}{a^U_4 - a^U_3} w_{\widetilde{A}^U}, & a^U_3 < x \le a^U_4 \\ 0, & x \ge a^U_4. \end{cases}$$

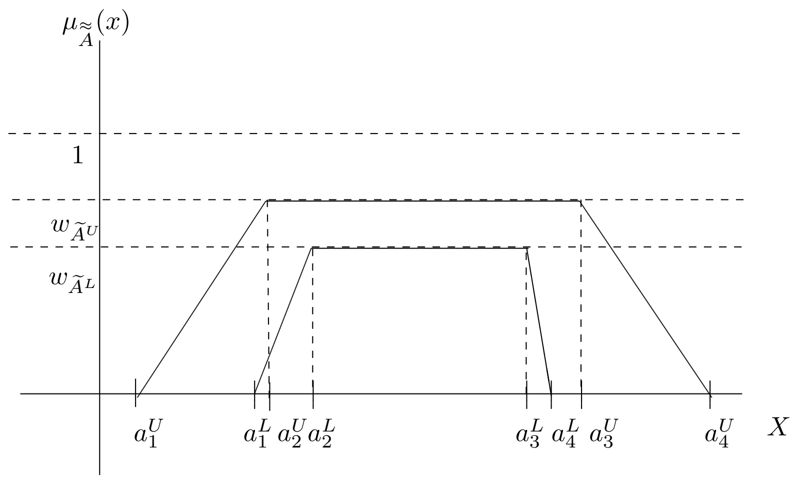

FIGURE 4. Interval-valued trapezoidal fuzzy number

Since $\mu^L_A(x) \le \mu^U_A(x)$, therefore $a^L_1 \ge a^U_1, a^L_2 \ge a^U_2$ and $a^L_3 \le a^U_3, a^L_4 \le a^U_4$ must hold.

Interval-valued fuzzy numbers is used by Lin [16] to represent vague processing time in job-shop scheduling problems. Wang and Li [44] present the correlation coefficient of interval valued fuzzy numbers and some of their properties. Yao and Lin used interval-valued fuzzy numbers to represent unknown job processing time for constructing a fuzzy flow-shop sequencing model. Some methods have been proposed in [10,11] for measuring the degree of similarity between interval-valued fuzzy numbers. In [12], Chen et al. represents a method to measure the similarity between two IVTFNs. In this paper, we used the method to compare two IVTFNs.

Consider two IVTFNs $\widetilde{\widetilde{A}}$ and $\widetilde{\widetilde{B}}$, where $\widetilde{\widetilde{A}} = [\widetilde{A}^L, \widetilde{A}^U] = [(a^L_1, a^L_2, a^L_3, a^L_4; w_{\widetilde{A}^L}), (a^U_1, a^U_2, a^U_3, a^U_4;\ w_{\widetilde{A}^U})]$ and $\widetilde{\widetilde{B}} = [\widetilde{B}^L, \widetilde{B}^U] = [(b^L_1, b^L_2, b^L_3, b^L_4;\ w_{\widetilde{B}^L}), (b^U_1, b^U_2,\ b^U_3, b^U_4;\ w_{\widetilde{B}^U})]$. The degree of similarity [12] between IVTFNs is denoted by $S(\widetilde{\widetilde{A}}, \widetilde{\widetilde{B}})$ and is defined by

$$S(\widetilde{\widetilde{A}}, \widetilde{\widetilde{B}}) = \frac{S(\widetilde{A}^U, \widetilde{B}^U) \times (1 + S(\widetilde{A}^\Delta, \widetilde{B}^\Delta))}{2}$$

where, $S(\widetilde{A}^{\Delta}, \widetilde{B}^{\Delta}) =$ degree of similarity between the distance values of the two IVTFNs

$$\Delta a_i = |a_i^U - a_i^L| \text{ and } \Delta b_i = |b_i^U - b_i^L|$$

$$= \left[1 - \frac{\sqrt{\sum_{i=1}^{4}(\Delta a_i - \Delta b_i)^2}}{2}\right] \times \left[1 - \sqrt{\frac{|\Delta S_a - \Delta S_b|}{2}}\right]$$

$$\times \left|1 - \frac{|w_{\widetilde{A}^L} - w_{\widetilde{B}^L}|}{|w_{\widetilde{A}^U} - w_{\widetilde{B}^U}|}\right| \times T^{\Delta}.$$

Here $\Delta S_a = |S_{\widetilde{A}^U} - S_{\widetilde{A}^L}|$ and $\Delta S_b = |S_{\widetilde{B}^U} - S_{\widetilde{B}^L}|$. T^{Δ} denotes map distance between the lower and upper trapezoidal fuzzy numbers $\widetilde{A}^L$ and $\widetilde{A}^U$ of IVTFN $\approx{A}$. The parameters $S_{\widetilde{A}^L}, S_{\widetilde{A}^U}, S_{\widetilde{B}^L}, S_{\widetilde{B}^U}$ and T^{Δ} can be calculated as follows:

$$S_{\widetilde{A}^L} = \sqrt{\frac{\sum_{i=1}^{4}(a_i^L - \bar{a}^L)^2}{n-1}}, \qquad S_{\widetilde{B}^L} = \sqrt{\frac{\sum_{i=1}^{4}(b_i^L - \bar{b}^L)^2}{n-1}}$$

$$S_{\widetilde{A}^U} = \sqrt{\frac{\sum_{i=1}^{4}(a_i^U - \bar{a}^U)^2}{n-1}}, \qquad S_{\widetilde{B}^U} = \sqrt{\frac{\sum_{i=1}^{4}(b_i^U - \bar{b}^U)^2}{n-1}}$$

and $T^{\Delta} = \dfrac{\left[\left(2 - \frac{1+\max\{|\Delta a_2 - \Delta a_1|, |\Delta b_2 - \Delta b_1|\}}{1+\min\{|\Delta a_2 - \Delta a_1|, |\Delta b_2 - \Delta b_1|\}}\right) + \left(2 - \frac{1+\max\{|\Delta a_4 - \Delta a_3|, |\Delta b_4 - \Delta b_3|\}}{1+\min\{|\Delta a_4 - \Delta a_3|, |\Delta b_4 - \Delta b_3|\}}\right)\right]}{2}$
where $\bar{a}^U$ denotes the average of the four values a_1^U, a_2^U, a_3^U, a_4^U at the upper trapezoidal fuzzy number $\widetilde{A}^U$ and the similar concept is used for the notations $\bar{a}^L$, $\bar{b}^U$ and $\bar{b}^L$ and

$$S(\widetilde{A}^U, \widetilde{B}^U) = \left[1 - \frac{\sqrt{\sum_{i=1}^{4}(a_i^U - b_i^U)^2}}{2}\right] \times \left[1 - \sqrt{\frac{|S_{\widetilde{A}^U} - S_{\widetilde{B}^U}|}{2}}\right] \times \frac{\min\{w_{\widetilde{A}^U}, w_{\widetilde{B}^U}\}}{\max\{w_{\widetilde{A}^U}, w_{\widetilde{B}^U}\}} \times T^U$$

where $T^U = \dfrac{\left[\left(2 - \frac{1+\max\{|a_2^U - a_1^U|, |b_2^U - b_1^U|\}}{1+\min\{|a_2^U - a_1^U|, |b_2^U - b_1^U|\}}\right) + \left(2 - \frac{1+\max\{|a_4^U - a_3^U|, |b_4^U - b_3^U|\}}{1+\min\{|a_4^U - a_3^U|, |b_4^U - b_3^U|\}}\right)\right]}{2}$.

The larger the value of $S(\approx{A}, \approx{B})$, the greater the similarity between the IVTFNs $\approx{A}$ and $\approx{B}$.

Next, we consider the addition of two IVTFNs.

Definition 2.3 (Addition of two IVTFNs)**.** Let $\approx{A} = [\widetilde{A}^L, \widetilde{A}^U] = [(a_1^L, a_2^L, a_3^L, a_4^L; w_{\widetilde{A}^L}),$ $(a_1^U, a_2^U, a_3^U, a_4^U; w_{\widetilde{A}^U})]$ and $\approx{B} = [\widetilde{B}^L, \widetilde{B}^U] = [(b_1^L, b_2^L, b_3^L, b_4^L; w_{\widetilde{B}^L}), (b_1^U, b_2^U, b_3^U, b_4^U;$ $w_{\widetilde{B}^U})]$ be two IVTFNs then the addition of these IVTFNs is denoted by $\approx{A} + \approx{B}$ and is defined by $\approx{A} + \approx{B} = [(a_1^L + b_1^L, a_2^L + b_2^L, a_3^L + b_3^L, a_4^L + b_4^L; \max\{w_{\widetilde{A}^L}, w_{\widetilde{B}^L}\}), (a_1^U + b_1^U,$ $a_2^U + b_2^U, a_3^U + b_3^U, a_4^U + b_4^U; \max\{w_{\widetilde{A}^U}, w_{\widetilde{B}^U}\})]$.

In general, networks are directed graphs. So directed interval-valued fuzzy hypergraphs are considered. In a directed network, two nodes may connect with two different edges each with opposite direction but in a simple undirected network, two nodes are connected by only one edge. Now, we define following terms:

Definition 2.4 (Directed interval-valued fuzzy hypergraph)**.** A directed interval-valued fuzzy hypergraph $\overrightarrow{G}$ is a pair $(V, \overrightarrow{E})$ where V is a non-empty set of vertices (called nodes) and $\overrightarrow{E}$ is the set of interval-valued fuzzy hyperarcs; an interval-valued fuzzy hyperarc

$e \in \overrightarrow{E}$ is defined as a pair $(T(e), h(e))$, where $T(e)$ is a subset of V, with $T(e) \neq \phi$ is its tail and $h(e)$ is a vertex in $V - T(e)$ is its head. A node s is a source node in $\overrightarrow{G}$ if $h(e) \neq s$ for every $e \in \overrightarrow{E}$ and a node d is said to be a destination node if $d \notin T(e)$ for every $e \in \overrightarrow{E}$.

Definition 2.5 (Valid ordering in a fuzzy hypergraph)**.** Let $G = (V, \overrightarrow{E})$ be a directed fuzzy hypergraph. A valid ordering in G is a lexicographic ordering of nodes $V = \{v_1, v_2, \cdots, v_n\}$, such that for any $e \in E$

$$(v_i \in T(e)) \text{ and } (h(e) = v_j) \Rightarrow i < j.$$

For example, the hypergraph shown in Figure 5 has a valid ordering of nodes namely, $\{1, 2, 3, 4, 5, 6, 7, 8\}$.

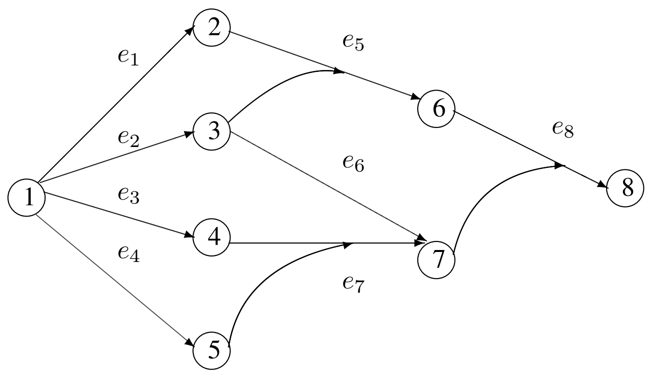

FIGURE 5. An example of hypernetwork

Definition 2.6 (Minimal fuzzy hypergraph)**.** A minimal fuzzy hypergraph is a fuzzy hypergraph in which after deletion of a fuzzy vertex (node) or fuzzy hyperarc, the resultant graph is not a fuzzy hypergraph.

Definition 2.7 (Fuzzy hyperpath)**.** Consider a directed fuzzy hypergraph $G = (V, \overrightarrow{E})$. A fuzzy hyperpath Π_{st} of source s and destination t is a minimal fuzzy hypergraph $G_\pi = (V_\pi, E_\pi)$ with valid ordering of nodes satisfying the following conditions:

(1) $E_\pi \subseteq E$ and $V_\pi = \bigcup_{e \in E_\pi} (T(e) \cup \{h(e)\})$,
(2) $s, t \in V_\pi$,
(3) $u \in V_\pi \backslash \{s\} \Rightarrow u$ is connected to s in G_π.

For example, in Figure 5, $1 \to 3 \to 7 \to 8$ is a fuzzy hyperpath.

3. ENUMERATION OF ALL HYPERPATHS IN A HYPERGRAPH

In this section, we describe an algorithm based on BFS technique to find the paths between two nodes.

For the sake of algorithm we use some notations. Let P_q be a queue of edges forming the path and V_q is a queue of vertices. $enqueue(v)$ adds the vertex v to the queue V_q and $dequeue$ removes a vertex from the queue V_q.

In this algorithm, we use the following functions:

i) $Array(v)$: array of vertices associated to a vertex v.

ii) $Arrays(v)$: array of elements associated to a vertex v; an element of $\mathrm{Arrays}(v)$ is either an array or a single vertex.
iii) *CreateArray(arrays, vertex_to_add)*: creates a new array by adding a vertex *vertex_to_add* to the existing array *arrays*.
iv) *AddArray(arrays, vertex_to _add)*: adds a vertex *vertex_to_add* to the associated array *arrays*.

Initially, every vertex is labeled as 'not visited'. After each iteration when a vertex is picked up and assigned to an array the vertex is immediately being labeled as 'visited'. In this algorithm $vertex.visited$ denotes the label of a vertex as either 'visited' or 'not visited'.

Algorithm PF

Input:: Source node s and the destination node t of the directed fuzzy hypergraph $G = (V, \overrightarrow{E})$.

Output:: All hyperpaths.

Step 1.: Create an array with one element s and set it to the vertex s. Each time we call this array by $Array(s)$, the associated array for the vertex s. If a vertex has two or more arrays then all the arrays are stored in a single array called $Arrays(s)$.

Step 2.: For each $e \in \overrightarrow{E}$
    if $s \in T(e)$ then
        $V_q \leftarrow enqueue(h(e))$
        if $h(e).visited =$'visited' then
            $CreateArray(Arrays(s), h(e))$
            $\backslash\backslash$ Create a new array for the vertex $s\backslash\backslash$.
        else
            $AddArray(Arrays(s), h(e))$
            $\backslash\backslash$ Add the vertex $h(e)$ to $\mathrm{Arrays}(s)\backslash\backslash$.
        end if;
        Set $h(e).visited =$'visited'.
    end if;
end for;

Step 3.: $s \leftarrow dequeue(V_q)$.

Step 4.: If V_q is not empty, then go to Step 2 otherwise Stop the process.

end PF.

At the end of the algorithm an array is created for each vertex. We find the arrays of the destination vertex t from $Arrays(t)$. All the arrays $Array(t)$ of $Arrays(t)$ are all the possible hyperpaths.

3.1. **Proof of correctness of the algorithm.** In Algorithm PF, every vertex v is associated with an array $\mathrm{Arrays}(v)$, where $\mathrm{Arrays}(v)$ contain some array of vertices $\mathrm{Array}(v_1), \mathrm{Array}(v_2), \ldots, \mathrm{Array}(v_k)$. Each of these arrays determines path from source node to the vertex v. We claim that Algorithm PF determines all the hyperpaths between two given nodes.

First, Algorithm PF sets an array $\mathrm{Arrays}(v)$ with only one element s which is the source node. In a hypergraph all the paths are of the form

$$s \in T(e_1) \to h(e_1) \to T(e_2) \to h(e_2) \to \cdots \to T(e_m) \to h(e_m) = t,$$

where t is the destination node. Step 2 determines all these paths by checking each e if $s \in T(e)$. $s \in T(e)$ implies that there exist a path from s to $h(e)$ along the edge e. This $h(e)$ is the next starter of the edge connected to s. This algorithm stores the path traversed from s to $h(e)$. In this way, all the paths are traversed and found the path. Since each path from the source vertex to a vertex is stored along with that vertex then $\mathrm{Arrays}(t)$ of the destination vertex t has all the paths from source node to destination node.

Theorem 3.1. *The Algorithm PF runs in $O(mn)$ time, where m is the number of edges and n is the number of vertices.*

Proof. Let the processor takes unit time to perform a single instruction. Step 1 of the Algorithm PF takes $O(1)$ time. The algorithm consists of a loop from Step 2 to Step 4. This loop carry over $O(n)$ times as V_q contains only the vertices of the graph. Within this loop we see that a loop occurs in Step 2 which is terminated after m times. Hence the overall time complexity of the Algorithm PF is of $O(mn)$. ■

Illustration of algorithm PF. Consider a directed hypergraph (hypernetwork) shown in Figure 5.
Step 1: Consider the source vertex as 1 and the destination vertex as 8. Assign an array (1) with only one entry 1 to the vertex 1.
Step 2: $1 \in T(e_1)$, $1 \in T(e_2)$, $1 \in T(e_3)$ and $1 \in T(e_4)$.
Step 3: $h(e_1) = 2$. So the vertex 2 is enqueued to the queue V_q.
Step 4: The vertex 2 is marked as visited and we create an array $(1, 2)$.
Step 5: Similarly, the vertices 3 and 4 are marked as visited and queued to V_q and we set the arrays $(1, 3)$ and $(1, 4)$ associated to 3 and 4 respectively.
Step 6: Dequeue the vertex 2 from the queue V_q and see that $2 \in T(e_5)$. Then $h(e_5) = 6$.
Step 7: The vertex 6 is marked as visited and queued to V_q. Assign an array $(1, 2, 6)$ to the vertex 6.
Step 8: For the vertex 3 of V_q, we see that $h(e_5) = 6$ is visited then we create a new array $(1, 3, 6)$ and assigned to 6.
Step 9: Proceeding in the similar way we assign arrays to each vertices. And observe that the destination vertex 8 is assigned with the arrays $(1, 2, 6, 8)$, $(1, 3, 6, 8)$, $(1, 3, 7, 8)$, $(1, 4, 7, 8)$, $(1, 5, 7, 8)$.

Therefore all the possible hyperpaths in the given hypernetwork are
P-1: $1 \to 2 \to 6 \to 8$,
P-2: $1 \to 3 \to 6 \to 8$,
P-3: $1 \to 3 \to 7 \to 8$,
P-4: $1 \to 4 \to 7 \to 8$,
P-5: $1 \to 5 \to 7 \to 8$.

The length of a hyperpath of a network is the sum of the lengths of all hyperarcs on the hyperpath.

Here, we describe an algorithm to find the minimum length of all fuzzy hyperpaths of the interval-valued fuzzy hypergraph from a given source node to a destination node.

Algorithm MLIVFH

: \\This algorithm determines the minimum weighted (trapezoidal fuzzy) hyperpaths of an interval-valued fuzzy hypergraph between two given nodes.\\

Input: Weights $(\overset{\approx}{A}_i)$ $(i = 1, 2, 3, \ldots)$ of all fuzzy hyperedges of the fuzzy hypergraph.

Output: Minimum weighted fuzzy hyperpath $(\overset{\approx}{W}_{\min})$ among all fuzzy hyperpaths from source to destination.

Step 1.: Find all fuzzy hyperpaths and compute weights $(\overset{\approx}{W}_i)$ $(i = 1, 2, 3, \ldots)$ of each fuzzy hyperpaths by the rule of addition of two IVTFNs.

Let each $\widetilde{\widetilde{W}}_i$ is of the form

$$\widetilde{\widetilde{W}}_i = [({a_1}_i^L, {a_2}_i^L, {a_3}_i^L, {a_4}_i^L; w_{\widetilde{W_i}^L}), ({a_1}_i^U, {a_2}_i^U, {a_3}_i^U, {a_4}_i^U; w_{\widetilde{W_i}^U})],\ i = 1, 2, 3, \ldots.$$

Step 2.: Initialize

$$\begin{aligned}\widetilde{\widetilde{W}}_{\min} &= [({a_1}^L, {a_2}^L, {a_3}^L, {a_4}^L; w_{\widetilde{W}^L}), ({a_1}^U, {a_2}^U, {a_3}^U, {a_4}^U; w_{\widetilde{W}^U})] \\ &= \widetilde{\widetilde{W}}_1 = [({a_1}_1^L, {a_2}_1^L, {a_3}_1^L, {a_4}_1^L; w_{\widetilde{W_1}^L}), ({a_1}_1^U, {a_2}_1^U, {a_3}_1^U, {a_4}_1^U; w_{\widetilde{W_1}^U})].\end{aligned}$$

Step 3.: Set $i = 2$.

Step 4.: Compute $[({a_1}^L, {a_2}^L, {a_3}^L, {a_4}^L; w_{\widetilde{W}^L}), ({a_1}^U, {a_2}^U, {a_3}^U, {a_4}^U; w_{\widetilde{W}^U})]$ of each fuzzy hyperpath as

$${a_1}^L = \min({a_1}^L, {a_1}_i^L)$$

$${a_2}^L = \begin{cases} {a_2}^L, & \text{if } {a_2}^L \leq {a_1}_i^L \\ \frac{{a_2}^L {a_2}_i^L - {a_1}^L {a_1}_i^L}{({a_2}^L + {a_2}_i^L) - ({a_1}^L + {a_1}_i^L)}, & \text{if } {a_2}^L > {a_1}_i^L, \end{cases}$$

$${a_3}^L = \begin{cases} {a_3}^L, & \text{if } {a_3}^L \leq {a_1}_i^L \\ \frac{{a_3}^L {a_3}_i^L - {a_1}^L {a_1}_i^L}{({a_3}^L + {a_3}_i^L) - ({a_1}^L + {a_1}_i^L)}, & \text{if } {a_3}^L > {a_1}_i^L, \end{cases}$$

$${a_4}^L = \min({a_4}^L, {a_3}_i^L)$$

$$w_{\widetilde{W}^L} = \min\{w_{\widetilde{W}^L}, w_{\widetilde{W_i}^L}\}$$

and

$${a_1}_1^U = \min({a_1}^U, {a_1}_i^U)$$

$${a_2}^U = \begin{cases} {a_2}^U, & \text{if } {a_2}^U \leq {a_1}_i^U \\ \frac{{a_2}^U {a_2}_i^U - {a_1}^L {a_1}_i^U}{({a_2}^U + {a_2}_i^U) - ({a_1}^U + {a_1}_i^U)}, & \text{if } {a_2}^U > {a_1}_i^U \end{cases}$$

$${a_3}^U = \begin{cases} {a_3}^U, & \text{if } {a_3}^U \leq {a_1}_i^U \\ \frac{{a_3}^U {a_3}_i^U - {a_1}^U {a_1}_i^U}{({a_3}^U + {a_3}_i^U) - ({a_1}^U + {a_1}_i^U)}, & \text{if } {a_3}^U > {a_1}_i^U \end{cases}$$

$${a_4}^U = \min({a_4}^U, {a_3}_i^U)$$

$$w_{\widetilde{W}^U} = \min\{w_{\widetilde{W}^U}, w_{\widetilde{W_i}^U}\}$$

Step 5.: Set $\widetilde{\widetilde{W}}_{\min} = [({a_1}^L, {a_2}^L, {a_3}^L, {a_4}^L; w_{\widetilde{W}^L}), ({a_1}^U, {a_2}^U, {a_3}^U, {a_4}^U; w_{\widetilde{W}^U})]$ as calculated in Step 4.

Step 6.: Increase the value of i by 1.

Step 7.: If $i < n + 1$ go to Step 4, otherwise stop the procedure.

end MLIVFH.

Illustrative example for the Algorithm MLIVFH:. Consider the hypernetwork shown in Figure 5. Now the weights of all hyperarcs of this hypernetwork are taken as follows:

$e_1 : [(0.4, 0.6, 0.7, 0.8; 0.7), (0.3, 0.5, 0.7, 0.9; 0.8)]$,

$e_2 : [(0.4, 0.8, 0.10, 0.12; 0.6), (0.2, 0.8, 0.13, 0.15; 0.8)]$,

$e_3 : [(0.9, 0.15, 0.16, 0.22; 0.6), (0.7, 0.13, 0.17, 0.23; 0.7)]$,

$e_4 : [(0.13, 0.16, 0.18, 0.27; 0.8), (0.10, 0.12, 0.19, 0.28; 0.87)]$,

$e_5 : [(0.11, 0.11, 0.21, 25; 0.82), (0.1, 0.1, 21, 0.26; 0.9)]$,

$e_6 : [(0.9, 0.17, 0.19, 0.23; 0.87), (0.6, 0.12, 0.21, 0.25; 0.92)]$,

$e_7 : [(0.6, 0.8, 0.11, 0.15; 0.5), (0.3, 0.5, 0.12, 0.17; 0.7)]$,

$e_8 : [(0.3, 0.5, 0.7, 0.11; 0.65), (0.1, 0.4, 0.11, 0.18; 0.78)]$.

In Section 3 we see that all the hyperpaths are
P-1: $1 \to 2 \to 6 \to 8$,
P-2: $1 \to 3 \to 6 \to 8$,
P-3: $1 \to 3 \to 7 \to 8$,
P-4: $1 \to 4 \to 7 \to 8$,
P-5: $1 \to 5 \to 7 \to 8$.

The weights of these hyperpaths are respectively
P-1: $\overset{\approx}{W}_1 = [(0.21, 0.25, 0.37, 0.48; 0.87), (0.13, 0.18, 0.40, 0.52; 0.92)]$,
P-2: $\overset{\approx}{W}_2 = [(0.21, 0.27, 0.40, 0.52; 0.87), (0.12, 0.21, 0.42, 0.57; 0.92)]$,
P-3: $\overset{\approx}{W}_3 = [(0.16, 0.30, 0.36, 0.45; 0.7), (0.9, 0.22, 0.40, 0.58; 0.8)]$,
P-4: $\overset{\approx}{W}_4 = [(0.18, 0.28, 0.33, 0.45; 0.8), (0.11, 0.22, 0.39, 0.56; 0.87)]$,
P-5: $\overset{\approx}{W}_5 = [(0.22, 0.29, 0.35, 0.52; 0.82), (0.14, 0.21, 0.41, 0.63; 0.9)]$.

Now, the value of $\overset{\approx}{W}_{\min}$ is given by $\overset{\approx}{W}_{\min} = [(0.16, 0.21, 0.22, 0.33; 0.7), (0.9, 0.13, 0.17, 0.39; 0.8)]$.

4. Computation of fuzzy shortest hyperpath

In this section, we compute similarity measures [12] $S(\overset{\approx}{A}, \overset{\approx}{B})$ to compare two IVTFNs $\overset{\approx}{A}$ and $\overset{\approx}{B}$. An algorithm is designed to find the fuzzy shortest hyperpath using similarity measurements.

Algorithm FSHP

Input:: An interval-valued fuzzy hypergraph with a source node and a destination node.
Output:: Fuzzy shortest hyperpath of the given interval-valued fuzzy hypergraph.
Step 1.: Find out all possible fuzzy hyperpaths from source node to destination node by the Algorithm PF.
Step 2.: Compute $\overset{\approx}{W}_{\min}$ by using Algorithm MLIVFH.
Step 3.: Find the similarity measures $S(\overset{\approx}{W}_i, \overset{\approx}{W}_{\min})$ for $i = 1, 2, \ldots, p$ (p being the number of all possible fuzzy hyperpaths) of $\overset{\approx}{W}_i$ and $\overset{\approx}{W}_{\min}$.
Step 5.: Decide k-th hyperpath as the fuzzy shortest hyperpath having the highest similarity measure $S(\overset{\approx}{W}_k, \overset{\approx}{W}_{\min})$ among all i's.

end FSHP.

Correctness and time complexity of the Algorithm FSHP:. Step 1 of Algorithm FSHP determines all the hyperpaths of a given hypergraph. Now, $\overset{\approx}{W}_{\min}$ is minimum among all IVTFNs associated to hyperpaths. By calculating similarity measures between two IVTFNs we have decided that which IVTFN is nearly equal to the minimum most IVTFN $\overset{\approx}{W}_{\min}$ and hence it is claimed that the hyperpath associating that IVTFN is the best shortest path.

Theorem 4.1. *The Algorithm FSHP runs in $O(mn)$ time, where m is the number of edges and n is the number of vertices.*

Proof. Step 1 takes $O(mn)$ time. Algorithm MLIVFH takes $O(m)$ time as number of paths cannot be greater than the number of edges in a hypergraph. By the same reason Step 3, Step 4 and Step 5 takes $O(m)$ time. So, overall worse case time complexity of the Algorithm FSHP is $O(mn)$. ■

5. Application to find fuzzy shortest hyperpath in railways network

Here, we have considered the railways network to find the shortest time require to traverse from a source station to a destination station. The railways networks are connected through more than 7000 stations in India although, we consider a simple structure to understand the work presented in this paper. Assume there are 11 stations A, B, C, D, E, F, G, H, I, J, K. In hypergraph model we take these stations as vertices of the hypergraph and each train as a hyperedge. Since, a train can traverse more than two stations, so this is a hypergraph. The hypergraph of this proposed model is shown in Figure 6. In Figure 6, it is seen that, there are five trains and these trains traverse the stations A, B, C, D, E; B, C, F, G; D, E, H, I, J; G, H, K; J, K. Depending on the real phenomenon, the time required to traverse a train can be considered as fuzzy number. To generalize the proposed problem, here we have considered the time required to traverse the stations is interval-valued trapezoidal fuzzy number. Now, to find the shortest time require to traverse the destination from a source station, hypergraph model is drawn in normal graph model as shown in Figure 7.

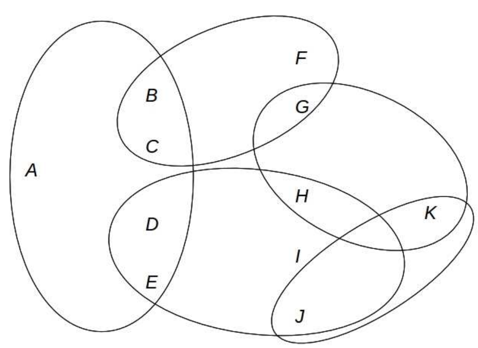

FIGURE 6. Hypergraph of railways network

The time (in hrs.) required to traverse the train between the stations are given in Table 1. For computations, time is converted in terms of 100.

Edge	Edge weight
(A, B)	$[(0.15, 0.16, 0.18, 0.22; 0.8), (0.13, 0.14, 0.19, 0.23; 0.9)]$
(A, C)	$[(0.18, 0.19, 0.20, 0.21; 0.7), (0.14, 0.17, 0.22, 0.25; 0.9)]$
(A, D)	$[(0.16, 0.17, 0.18, 0.19; 0.6), (0.15, 0.16, 0.19, 0.22; 0.9)]$
(A, E)	$[(0.15, 0.17, 0.19, 0.20; 0.7), (0.12, 0.13, 0.19, 0.22; 0.8)]$
(B, C)	$[(0.17, 0.18, 0.20, 0.22; 0.8), (0.16, 0.17, 0.22, 0.23; 0.9)]$
(C, D)	$[(0.16, 0.18, 0.19, 0.20; 0.8), (0.15, 0.16, 0.19, 0.21; 0.9)]$
(D, E)	$[(0.15, 0.16, 0.17, 0.20; 0.5), (0.14, 0.16, 0.19, 0.20; 0.7)]$
(B, D)	$[(0.33, 0.36, 0.39, 0.42; 0.8), (0.31, 0.33, 0.41, 0.44; 0.9)]$
(B, E)	$[(0.48, 0.52, 0.56, 0.62; 0.8), (0.45, 0.49, 0.60, 0.63; 0.9)]$
(B, F)	$[(0.18, 0.19, 0.21, 0.22; 0.6), (0.14, 0.16, 0.20, 0.23; 0.9)]$
(B, G)	$[(0.19, 0.20, 0.21, 0.22; 0.7), (0.16, 0.18, 0.22, 0.23; 0.8)]$
(C, F)	$[(0.17, 0.18, 0.19, 0.21; 0.5), (0.12, 0.14, 0.19, 0.23; 0.7)]$
(C, G)	$[(0.14, 0.15, 0.18, 0.20; 0.7), (0.13, 0.14, 0.19, 0.22; 0.8)]$
(D, H)	$[(0.15, 0.16, 0.19, 0.23; 0.8), (0.12, 0.15, 0.20, 0.25; 0.9)]$
(D, I)	$[(0.17, 0.18, 0.19, 0.25; 0.5), (0.14, 0.16, 0.22, 0.27; 0.8)]$
(D, J)	$[(0.16, 0.17, 0.18, 0.20; 0.6), (0.13, 0.14, 0.19, 0.21; 0.7)]$
(E, H)	$[(0.16, 0.17, 0.18, 0.20; 0.6), (0.13, 0.14, 0.19, 0.21; 0.7)]$
(E, I)	$[(0.18, 0.19, 0.20, 0.22; 0.7), (0.16, 0.17, 0.22, 0.23; 0.8)]$
(E, J)	$[(0.16, 0.17, 0.19, 0.20; 0.8), (0.13, 0.14, 0.19, 0.21; 0.9)]$
(F, G)	$[(0.15, 0.17, 0.18, 0.21; 0.6), (0.13, 0.14, 0.19, 0.21; 0.8)]$
(H, I)	$[(0.16, 0.18, 0.18, 0.20; 0.5), (0.13, 0.16, 0.19, 0.21; 0.7)]$
(I, J)	$[(0.14, 0.16, 0.18, 0.20; 0.7), (0.13, 0.14, 0.19, 0.21; 0.8)]$
(H, J)	$[(0.30, 0.34, 0.36, 0.40; 0.7), (0.26, 0.30, 0.38, 0.42; 0.8)]$
(G, K)	$[(0.16, 0.17, 0.18, 0.21; 0.6), (0.13, 0.14, 0.19, 0.21; 0.7)]$
(H, K)	$[(0.16, 0.18, 0.19, 0.22; 0.6), (0.13, 0.14, 0.19, 0.23; 0.8)]$
(J, K)	$[(0.16, 0.17, 0.18, 0.20; 0.6), (0.13, 0.15, 0.19, 0.21; 0.7)]$

TABLE 1. Edge weights of the graph shown in Figure 7

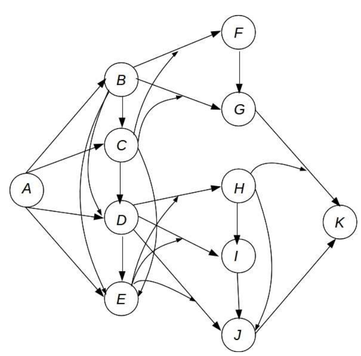

FIGURE 7. Graph of railways network to find the shortest time to traverse the station K from the station A

Now, among all the hyperpaths the shortest paths and their weights between the stations A and K are as follows:

P-1: $A \longrightarrow B \longrightarrow G \longrightarrow K; \stackrel{\approx}{W}_1 = [(0.50, 0.53, 0.57, 0.65; 0.8), (0.42, 0.46, 0.60, 0.67; 0.9)]$

P-2: $A \longrightarrow C \longrightarrow G \longrightarrow K; \stackrel{\approx}{W}_2 = [(0.48, 0.51, 0.56, 0.62; 0.7), (0.40, 0.45, 0.60, 0.68; 0.9)]$

P-3: $A \longrightarrow D \longrightarrow H \longrightarrow K; \stackrel{\approx}{W}_3 = [(0.47, 0.51, 0.56, 0.64; 0.8), (0.40, 0.45, 0.58, 0.70; 0.9)]$

P-4: $A \longrightarrow D \longrightarrow J \longrightarrow K; \stackrel{\approx}{W}_4 = [(0.48, 0.51, 0.54, 0.59; 0.6), (0.41, 0.45, 0.57, 0.64; 0.9)]$

P-5: $A \longrightarrow E \longrightarrow H \longrightarrow K; \stackrel{\approx}{W}_5 = [(0.47, 0.52, 0.56, 0.62; 0.7), (0.38, 0.41, 0.57, 0.66; 0.8)]$

P-6: $A \longrightarrow E \longrightarrow J \longrightarrow K; \stackrel{\approx}{W}_6 = [(0.47, 0.51, 0.56, 0.60; 0.8), (0.38, 0.42, 0.57, 0.64; 0.9)]$

After usual computations, one can find $\stackrel{\approx}{W}_{\min} = [(0.47, 0.48, 0.49, 0.54; 0.6), (0.38, 0.38, 0.42, 0.57; 0.8)]$.

Routine computations can be done and the results are $S(\stackrel{\approx}{W}_1, \stackrel{\approx}{W}_{\min}) = 0.87$, $S(\stackrel{\approx}{W}_2, \stackrel{\approx}{W}_{\min}) = 0.46$, $S(\stackrel{\approx}{W}_3, \stackrel{\approx}{W}_{\min}) = 0.88$, $S(\stackrel{\approx}{W}_4, \stackrel{\approx}{W}_{\min}) = 0.90$, $S(\stackrel{\approx}{W}_5, \stackrel{\approx}{W}_{\min}) = 0.47$, $S(\stackrel{\approx}{W}_6, \stackrel{\approx}{W}_{\min}) = 0.88$.

So, the shortest path from A to K is $A \longrightarrow D \longrightarrow J \longrightarrow K$.

6. CONCLUSION

Several methods have been found in literature to find the fuzzy shortest path in a hypernetwork. Here we proposed a method using similarity measure to find the fuzzy shortest path in a network with imprecise arc lengths which are IVTFNs. Here BFS technique is used to find all hyperpaths in a hypernetwork. Our proposed algorithm takes $O(mn)$ time, where m and n represent the number of edges and vertices of a fuzzy hypernetwork. Since the IVTFN is more general fuzzy number, our algorithm can be used to solve more generalized shortest path problem on a fuzzy hypernetwork.

7. ACKNOWLEDGMENT

We thank reviewers and editors of the journal entitled “Annals of Communications in Mathematics” for their comments that greatly improved the manuscript.

REFERENCES

[1] M. Akram and B. Davvaz. Strong intuitionistic fuzzy graphs, Filomat, 26(1)(2012) 177-196.
[2] M. Akram, A. Ashraf, and S. Swrwar. Novel application of intuitionistic fuzzy digraphs in decision support systems, The Scientific World Journal, 2015, Article ID 904606, 11 pages.
[3] M. Akram and W. Dudek. Intuitionistic fuzzy hypergraphs with applications, Information Sciences, 218 (2013) 182-193.
[4] M. Akram and N. Al-Shehrie. Intuitionistic fuzzy cycles and intuitionistic fuzzy trees, The Scientific World Journal, 2014 : 11 pages.
[5] N. Al-Shehrie and M. Akram. Bipolar fuzzy competition graphs, Ars Combinatoria, 121(2015) 385-402.
[6] N. Al-Shehrie and M. Akram. Intuitionistic fuzzy planar graphs, Discrete Dynamics in Nature and Society, (2014) : 9 pages.
[7] G. Ausiello, A. D'Atri and D. Saccà. Graph algorithms for functional dependency manipulation, Journal of the ACM, 30 (1983) 752-766.
[8] S. H. Chen. Operations on fuzzy numbers with function principal, Tamkang Journal of Management Sciences, 6(1) (1985) 13-25.
[9] S. H. Chen. Ranking generalized fuzzy number with graded mean integration, Proceedings of the Eighth International Fuzzy Systems Association World Congress, Taipei, Taiwan, Republic of China, 2 (1999) 899-902.
[10] S. J. Chen and S. M. Chen. A new fuzzy query processing method for handling fuzzy number information retrieval problems, Proceedings of the 14th International Conference on Information Management, Chiayi, Taiwan, Republic of China, 2003.

[11] S. J. Chen. Measure of similarity between interval-valued fuzzy numbers for fuzzy risk analysis based on Geometric-Mean operator, The 2009 international conference on e-Technology (e-Tech 2009), Singapore, Jan 2009.
[12] S. J. Chen, Z. Y. Wang, and W. R. Li. Calculating degree of similarity between interval-valued fuzzy numbers based on map distance, Proceedings of the international multi-conference of engineers and computer scientists 2013, Hong-Kong, vol. 1, 2013.
[13] T. N. Chuang and J. Y. Kung. A new algorithm for the discrete fuzzy shortest path problem in a network, Applied Mathematics and Computation, 174 (2006) 660-668.
[14] T. N. Chuang and J. Y. Kung. The fuzzy shortest path length and the corresponding shortest path in a network, Computers and Operations Research, 32 (2005) 1409-1428.
[15] D. Dubois and H. Prade. Fuzzy sets and systems, Academic Press, New York, 1980.
[16] F. T. Lin. Fuzzy job shop scheduling based on ranking level $(\lambda, 1)$ interval-valued fuzzy number, IEEE transactions on fuzzy systems, 10 (4) (2002) 510-522.
[17] G. Gallo, G. Longo, S. Nguyen and S. Pallottino. Directed hypergraphs and applications, Discrete Applied Mathematics, 40 (1993) 177-201.
[18] R. H. Goetschel. Introduction to fuzzy hypergraphs and Hebbian structures, Fuzzy Sets and Systems, 76 (1995) 113-130.
[19] R. H. Goetschel. Fuzzy colorings of fuzzy hypergraphs, Fuzzy Sets and Systems, 94 (1998) 185-204.
[20] R. H. Goetschel and W. Voxman. Intersecting fuzzy hypergraphs, Fuzzy Sets and Systems, 99 (1998) 81-96.
[21] G. Kumar, R. K. Bajaj and N. Gandotra. Algorithm for Shortest Path Problem in a Network with Interval-valued Intuitionistic Trapezoidal Fuzzy Number, Procedia Computer Science, 70 (2015) 123-129.
[22] K. C. Lin and M. S. Chern. The fuzzy shortest path problem and its most vital arcs, Fuzzy Sets and Systems, 58 (1993) 343-353.
[23] A. Nayeem and M. Pal. Shortest path problem on a network with imprecise edge weights, Fuzzy Optimization and Decision Making, 4 (2005) 293-312.
[24] S. Okada and T. Soper. A shortest path problem on a network with fuzzy arc lengths, Fuzzy sets and systems, 109 (2000) 129-140.
[25] T. Pramanik, M. Pal and S. Mondal. Interval-valued fuzzy threshold graph, Pacific Science Review A: Natural Science and Engineering, 18(1) (2016) 66-71.
[26] T. Pramanik, S. Samanta and M. Pal. Interval-valued fuzzy planar graphs, International Journal of Machine Learning and Cybernetics, 7 (4) (2016) 653-664.
[27] T. Pramanik, M. Pal, S. Mondal and S. Samanta. Bipolar fuzzy planar graph, Journal of Intelligent & Fuzzy Systems, 34(3) (2018) 1863-1874.
[28] T. Pramanik, S. Samanta, B. Sarkar and M. Pal. Fuzzy ϕ-tolerance competition graphs, Soft Computing, 21(13) (2017) 3723-3734.
[29] T. Pramanik, S. Samanta, M. Pal, S. Mondal and B. Sarkar. Interval-valued fuzzy ϕ-tolerance competition graph, SpringerPlus, 5 (2016) 1981.
[30] S. Sahoo and M. Pal. Different types of products on intuitionistic fuzzy graphs, Pacific Science Review A: Natural Science and Engineering, 17(3) (2015) 87-96.
[31] S. Sahoo and M. Pal. Intuitionistic fuzzy competition graph. Journal of Applied Mathematics and Computing, 52 (2016) 37-57.
[32] S. Sahoo and M. Pal. Product of intuitionistic fuzzy graphs and degree, Journal of Intelligent and Fuzzy Systems, 32(1) (2017) 1059-1067.
[33] S. Samanta, T. Pramanik and M. Pal. Fuzzy colouring of fuzzy graphs, Afrika Matematika, 27(1-2) (2016) 37-50.
[34] S. Samanta and M. Pal. Fuzzy k-competition graphs and p-competition fuzzy graphs, Fuzzy Engineering and Information, 5(2) (2013) 191-204.
[35] S. Samanta and M. Pal. Fuzzy tolerance graphs, International Journal of Latest Trends in Mathematics, 1(2) (2011) 57-67.
[36] S. Samanta and M. Pal. Fuzzy threshold graphs, CIIT International Journal of Fuzzy Systems, 3(12) (2011) 360-364.
[37] S. Samanta and M. Pal. Irregular bipolar fuzzy graphs, International Journal of Applications of Fuzzy Sets, 2 (2012) 91-102.
[38] S. Samanta and M. Pal. Bipolar fuzzy hypergraphs, International Journal of Fuzzy Logic Systems, 2 (1) (2012) 17-28.
[39] S. Samanta, M. Pal and A. Pal. Some more results on bipolar fuzzy sets and bipolar fuzzy intersection graphs, The Journal of fuzzy mathematics, 22 (2) (2014) 253-262.

[40] S. Samanta, M. Pal and A. Pal. New concepts of fuzzy planar graph, International Journal of Advanced Research in Artificial Intelligence, 3 (1) (2014) 52-59.
[41] S. Samanta, M. Akram and M. Pal. m-step fuzzy competition graphs, Journal of Applied Mathematics and Computing, 47(1-2) (2015) 461-472. doi: 10.1007/s12190-014-0785-2, 2014.
[42] S. Samanta and M. Pal. Telecommunication system based on fuzzy graphs, Telecommunications System & Management, 3 (1) (2013) 1-6.
[43] M. T. Takahashi and A. Yaamakani. On fuzzy shortest path problem with fuzzy parameter an algorithm American approach, Proceedings of the Annual Meeting of the North Fuzzy Information Processing Society, 654-657, 2005.
[44] G. Wang and X. Li. Correlation and information energy of interval-valued fuzzy number, Fuzzy Sets and Systems, 103 (1) (2001) 169-175 .
[45] J. S. Yao and F. T. Lin. Constructing a fuzzy flow-shop sequencing model based on statistical data, International Journal of Approximate Reasoning, 29 (3) (2002) 215-234.

TARASANKAR PRAMANIK
DEPARTMENT OF MATHEMATICS, KHANPUR GANGCHE HIGH SCHOOL, PASCHIM MEDINIPUR, 721201, INDIA
Email address: tarasankar.math07@gmail.com

MADHUMANGAL PAL
DEPARTMENT OF APPLIED MATHEMATICS WITH OCEANOLOGY AND COMPUTER PROGRAMMING, VIDYASAGAR UNIVERSITY, MIDNAPORE, 721102, INDIA
Email address: mmpalvu@gmail.com