



## Employing the Bellman-Ford Algorithm with Score Functions to Address the Linear Diophantine Fuzzy Shortest Path Problem in Network Analysis

Vidhya Kannan<sup>1</sup>, Saraswathi Appasamy<sup>2\*</sup>

Department of Mathematics, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu 603203, Chennai, India

Corresponding Author Email: [saraswaa@srmist.edu.in](mailto:saraswaa@srmist.edu.in)

<https://doi.org/10.18280/mmep.100542>

### ABSTRACT

**Received:** 4 April 2023

**Revised:** 20 July 2023

**Accepted:** 11 August 2023

**Available online:** 27 October 2023

#### Keywords:

*shortest path problem, Bellman-Ford algorithm, score functions, Linear Diophantine Fuzzy arc weights, Linear Diophantine Fuzzy shortest path problem*

The realms of Intuitionistic Fuzzy Sets (IFSs), Pythagorean Fuzzy Sets (PFS), and q-rung Orthopair Fuzzy Sets (q-ROFSs) have found extensive applications across various disciplines, notably in resolving real-world problems. However, limitations concerning membership and non-membership grades pose challenges to these theories. Efforts to mitigate these constraints have led to the introduction of a new concept, the Linear Diophantine Fuzzy Set (LDFS), with reference parameters. This study advances the shortest path (SP) problem for Linear Diophantine Fuzzy graphs. An innovative method for constructing direct network graphs within a Linear Diophantine Fuzzy (LDF) context is proposed. Distances or costs between nodes are encapsulated by Linear Diophantine Fuzzy numbers. The principal contribution of this investigation lies in proposing a novel approach to solving the Linear Fuzzy Diophantine Fuzzy shortest path problem using the Bellman-Ford algorithm for optimal solution attainment. Usage of the score function enables the comparison and identification of the minimum arc value between nodes. The proposed algorithm's validity is demonstrated through a numerical example, and a comparison with existing methodologies underscores the benefits of the proposed algorithm.

## 1. INTRODUCTION

The shortest path problem, an intriguing subject in operational research and graph theory, has been the focus of much scholarly attention. Researchers have developed numerous innovative approaches, algorithms, and models, all striving to find the optimal values—be they cost, time, or other considerations—that define the shortest path. Real-world complexities, such as varied weather conditions and traffic congestion, add layers of difficulty to these problems.

In 1965, Lotfi Zadeh introduced the concept of fuzzy set theory [1] to address these complexities. This theory expanded the traditional set theory to encapsulate fuzzy sets, which handle membership functions within a range of [0, 1]. Zadeh later proposed the concept of linguistic variables, whose values are expressed in natural language terms. When these words are articulated through fuzzy sets defined over a universal set, the variable is referred to as a fuzzy linguistic variable [2]. However, fuzzy sets only deal with membership functions.

To counter this limitation, Atanassov introduced an intuitionistic fuzzy set (IFS) [3], an extension of the fuzzy set where the total value of membership and non-membership does not exceed one. Further developments in this field led to the creation of the interval-valued intuitionistic fuzzy set (IVIFS) [4, 5], a generalization of the IFS. Despite their versatility, IFSs show limitations when the combined values of their membership and non-membership exceed one.

To address this, Yager proposed the Pythagorean fuzzy set (PyFS) [6-8], where the sum of the squared membership and non-membership values does not exceed one. Subsequently,

Zhang and Xu [9, 10], Fei and Deng [11] and Peng and Ma [12] introduced ranking techniques for the order preference ideal solution and distance-based score methods for the Pythagorean fuzzy decision-making process. Garg and Rahman and Abdullah [13-17] proposed the interval-valued Pythagorean fuzzy set, a generalized form of PyFS, demonstrating its utility in decision-making processes through aggregation operators, Einstein operations, t-norms, and t-conorms.

Despite these advancements, the theories of IFS, PyFS, and q-ROFS have limitations in their membership and non-membership grades. To overcome this, several researchers [18-21] introduced the Linear Diophantine Fuzzy set (LDFS). Thanks to its unique ability to provide a broad characterization area for permissible doublets, the LDFS theory surpasses the IFS, PyFS, and q-ROFS theories in handling vague and uncertain information via reference parameters.

A comprehensive range of membership and non-membership grades in decision analysis is crucial for analyzing complex situations. The decision-maker has the flexibility to choose these degrees to incorporate reference factors. The LDFS, along with reference parameters, provides a reliable method for addressing modeling uncertainties, which encourages us to find the optimal path under LDFN.

The fuzzy shortest path problem (SPP) on a network was first addressed by Dubois [22]. Although they analyzed the shortest path, the shortest path corresponding to the distance no longer existed. Okada and Soper [23] and colleagues then introduced an algorithm for solving the SPP using the multiple labeling method for the multicriteria shortest path. Chuang and Kung [24, 25] proposed a novel algorithm where each arc length is represented as a triangular fuzzy set to find the fuzzy

shortest path and fuzzy shortest path length. To address the shortcomings of existing algorithms, Deng et al. [26] employed the fuzzy Dijkstra algorithm to handle the SPP under uncertain environments. Numerous studies [27] have extended the Dijkstra algorithm to various fuzzy environments. Mukherjee [28] proposed a Dijkstra algorithm variation for the shortest path problem under intuitionistic fuzzy numbers. Kannan et al. [29] discussed the comparison between the fuzzy Floyd Warshall and fuzzy rectangular algorithms for the FSPP in their study. Kumar et al. [30] reviewed existing methods and proposed a fresh approach for the shortest path problem under a fuzzy environment.

A procedure [31] was developed for the shortest path problem on a network with fuzzy arc length named interval-valued fuzzy number matrices. Also, several approaches [32-40] were developed to evaluate the optimality of a variety of fuzzy theories for the fuzzy, intuitionistic fuzzy, and Pythagorean fuzzy shortest path problem. Broumi et al. [41] and Biswas et al. [42] conducted a literature survey on SPP under different fuzzy environments for various algorithms and identified the best methods for suitable conditions.

From the literature, it's evident that numerous algorithms exist for the study of the shortest path under various fuzzy environments. However, only limited work has incorporated reference parameters for SPP, motivating us to focus on the LDFN using the Bellman-Ford dynamic programming to find optimal solutions.

The Bellman-Ford method can handle graphs better than existing algorithms, particularly when certain edge weights are negative. It computes the shortest path in a graph with negative edges, reports the existence of a negative cycle, and aborts the computation of the undefined shortest path. This algorithm aims to provide the optimal fuzzy shortest path and the least LDF cost/time in a network-directed graph with LDF arc weights. The existing score function is used to compare the LDFNs and choose the minimum among them. A numerical example is provided to illustrate and discuss the efficiency and capabilities of the proposed method.

The contributions of this paper are:

- i. The structure, conditions, arithmetic operations, and ranking functions of the LDFN are discussed.
- ii. The Bellman-Ford algorithm has been extended to LDFN Edge values, and their mathematical formulation has been analyzed.
- iii. A numerical example is solved using the proposed method, and the optimal results are identified.
- iv. A comparison is validated for the fuzzy shortest path problem across different fuzzy theories.
- v. Finally, the paper concludes with a conclusion.

## 2. PRELIMINARIES

In this section, the definitions of IFS, PyFS and LDFN, as well as some of their arithmetic and ranking operations are discussed.

### Definition

An IFS  $\mathfrak{F}$  on the universe  $\mathfrak{Q}$  is defined by:  $\mathfrak{F} = \{x, m(x), n(x), | x \in \mathfrak{Q}\}$  where  $m: \mathfrak{Q} \rightarrow [0, 1]$  and  $n: \mathfrak{Q} \rightarrow [0, 1]$  represents the membership grade and non-membership grade of each  $x \in \mathfrak{Q}$ , respectively. A double set  $(m, n)$  is said to be IFS. Such that  $0 \leq m(x) + n(x) \leq 1$  for all  $x \in \mathfrak{Q}$ . Hesitation or

indeterminacy part can be calculated as  $\pi(x) = 1 - (m(x) + n(x))$ . In some of the real-life scenarios IFS can't work when  $m(x) + n(x) > 1$  [6] named as PFS, which is also known as IFS type 2 by the study [3].

### Definition [6]

A Pythagorean  $\mathfrak{Q}_p$  fuzzy set (PyFS) on  $\mathfrak{Q}$  is an object written as  $\mathfrak{Q}_p = \{ \langle x, m_p(x), n_p(x) \rangle : x \in \mathfrak{Q} \}$  where  $m_p(x) \in [0, 1]$  is the membership grade and  $n_p(x) \in [0, 1]$  is the non-membership grade for each  $x \in \mathfrak{Q}$ , respectively and the inequality  $m_p^2(x) + n_p^2(x) \leq 1$  holds for every  $x \in \mathfrak{Q}$ . A double set  $(m, n)$  is said to be PyFS. The indeterminacy or hesitant part is calculated by  $\pi_p(x) = \sqrt{1 - m_p^2(x) - n_p^2(x)}$ .

### Definition [21]

Let  $\mathfrak{Q}$  be the non-reference set. A LDFS  $L$  on  $\mathfrak{Q}$  is an object of the form  $\mathfrak{L}_{\mathfrak{D}} = \{ \langle x, \langle m_{\mathfrak{D}}(x), n_{\mathfrak{D}}(x) \rangle, \langle \alpha, \beta \rangle \rangle : x \in \mathfrak{Q} \}$  where  $m_{\mathfrak{D}}(x), n_{\mathfrak{D}}(x)$  are the membership grade and non-membership grade respectively. Let  $\alpha(x) \in [0, 1]$  is the satisfaction grade reference parameter and  $\beta(x) \in [0, 1]$  is the dissatisfaction grade reference parameter. These grades satisfy the condition  $0 \leq \alpha m_{\mathfrak{D}}(x) + \beta n_{\mathfrak{D}}(x) \leq 1 \forall x \in \mathfrak{Q}$  with  $0 \leq \alpha + \beta \leq 1$ . These reference parameters can help in defining or classifying a particular system. By moving the physical meaning of these parameters, we can categorize the system. They expand the space used in LDFSs for grades and lift the limitation on them. The hesitation part can be evaluated as  $\xi \pi_{\mathfrak{D}} = 1 - (\alpha m_{\mathfrak{D}}(x) + \beta n_{\mathfrak{D}}(x))$ , where  $\xi$  is the reference parameter related to degree of indeterminacy. The Linear Diophantine Fuzzy Number (LDFN) is defined as  $\mathfrak{L}_{\mathfrak{D}} = (\langle m_{\mathfrak{D}}, n_{\mathfrak{D}} \rangle, \langle \alpha, \beta \rangle)$  with  $0 \leq \alpha + \beta \leq 1$  and  $0 \leq \alpha m_{\mathfrak{D}} + \beta n_{\mathfrak{D}} \leq 1$ . The space of IFS, PyFs, LDFS in shown in Figure 1.

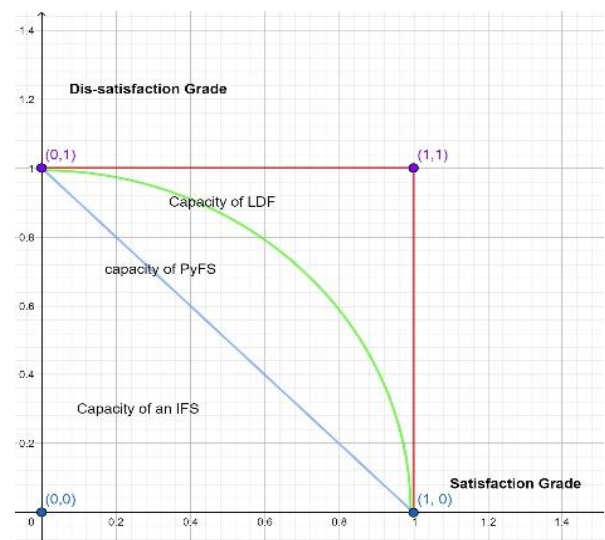


Figure 1. Space of IFS, PyFs, LDFS

### Definition [21]

A Linear Diophantine Fuzzy number (LDFN) can be written in the form of  $\mathfrak{L}_{\mathfrak{D}} = (\langle m_{\mathfrak{D}}, n_{\mathfrak{D}} \rangle, \langle \alpha_{\mathfrak{D}}, \beta_{\mathfrak{D}} \rangle)$  satisfy the following condition.

- $m_{\mathfrak{D}}, n_{\mathfrak{D}}, \alpha_{\mathfrak{D}}, \beta_{\mathfrak{D}} \in [0, 1]$
- $0 \leq \alpha_{\mathfrak{D}} + \beta_{\mathfrak{D}} \leq 1$

$$0 \leq m_{\mathcal{D}} \alpha_{\mathcal{D}} + n_{\mathcal{D}} \beta_{\mathcal{D}} \leq 1$$

**Example:** If  $m_{\mathcal{D}} = 0.89$  and  $n_{\mathcal{D}} = 0.73$ , then  $0.89 + 0.73 = 1.62 \not\leq 1$  as well as  $(0.89)^2 + (0.73)^2 = 1.325 \not\leq 1$ . However, with the arbitrary choice of parameter where  $\alpha, \beta \in [0, 1]$  and  $0 \leq \alpha + \beta \leq 1$  so that  $0 \leq \alpha m_{\mathcal{D}}(\mathbf{x}) + \beta n_{\mathcal{D}}(\mathbf{x}) \leq 1$ . Let  $(\alpha, \beta) = (0.55, 0.49)$ , therefore  $(0.55) + (0.73) = (0.49) = 0.8472 < 1$ . As a result, we were able to develop a more extensive space than the IFS and PyFS, and we now have additional alternatives for defining values to  $m_{\mathcal{D}}$  and  $n_{\mathcal{D}}$ , which is not possible in the IFS and PyFS.

### Definition [21]

A LDFS on  $\mathcal{Q}$  of the form  ${}^1\mathcal{L}_{\mathcal{D}} = \{(\mathbf{x}, \langle 1, 0 \rangle, \langle 1, 0 \rangle) : \mathbf{x} \in \mathcal{Q}\}$  is called absolute LDFS and  ${}^0\mathcal{L}_{\mathcal{D}} = \{(\mathbf{x}, \langle 0, 1 \rangle, \langle 0, 1 \rangle) : \mathbf{x} \in \mathcal{Q}\}$  is called null or empty LDFS.

### Definition [21]

Let  $\mathcal{L}_{\mathcal{D}_i} = (\langle m_{\mathcal{D}_i}, n_{\mathcal{D}_i} \rangle, \langle \alpha_{\mathcal{D}_i}, \beta_{\mathcal{D}_i} \rangle)$  for  $i \in \Delta$  be an assembling of LDFNs on  $\mathcal{Q}$  and  $\mathfrak{X} > 0$ , then  $\mathcal{L}_{\mathcal{D}_1} \oplus \mathcal{L}_{\mathcal{D}_2} = (\langle m_{\mathcal{D}_1} + m_{\mathcal{D}_2} - m_{\mathcal{D}_1} m_{\mathcal{D}_2}, n_{\mathcal{D}_1} n_{\mathcal{D}_2} \rangle, \langle \alpha_{\mathcal{D}_1} + \alpha_{\mathcal{D}_2} - \alpha_{\mathcal{D}_1} \alpha_{\mathcal{D}_2}, \beta_{\mathcal{D}_1} \beta_{\mathcal{D}_2} \rangle)$ .

### Definition [21]

Let  $\mathcal{L}_{\mathcal{D}} = (\langle m_{\mathcal{D}}, n_{\mathcal{D}} \rangle, \langle \alpha, \beta \rangle)$  be an LDFN, then the score function (SF) is denoted by  $\mathfrak{S}(\mathcal{L}_{\mathcal{D}})$  and the accuracy function (AF) by  $\mathfrak{H}(\mathcal{L}_{\mathcal{D}})$  on  $\mathcal{D}$  and can be defined by mapping  $\mathfrak{S} : \text{LDFN}(\mathcal{Q}) \rightarrow [-1, -1]$  and given by:

$$\mathfrak{S}(\mathcal{L}_{\mathcal{D}}) = \frac{1}{2} [(m_{\mathcal{D}} - n_{\mathcal{D}}) + (\alpha - \beta)] \quad (1)$$

$$\mathfrak{H}(\mathcal{L}_{\mathcal{D}}) = \frac{1}{2} \left[ \frac{(m_{\mathcal{D}} + n_{\mathcal{D}})}{2} + (\alpha + \beta) \right] \quad (2)$$

where,  $\mathfrak{H}(\mathcal{L}_{\mathcal{D}})(\mathcal{Q})$  is the assembling of all LDFNs on  $\mathcal{Q}$ .

### Definition [21]

If two LDFNs  $\mathcal{L}_{\mathcal{D}_1}$  and  $\mathcal{L}_{\mathcal{D}_2}$  can be comparable using the SF and the AF. This is defined as follows:

- If  $\mathcal{L}_{\mathcal{D}_1} > \mathcal{L}_{\mathcal{D}_2}$  then  $\mathfrak{S}(\mathcal{L}_{\mathcal{D}_1}) > \mathfrak{S}(\mathcal{L}_{\mathcal{D}_2})$
  - If  $\mathcal{L}_{\mathcal{D}_1} < \mathcal{L}_{\mathcal{D}_2}$  then  $\mathfrak{S}(\mathcal{L}_{\mathcal{D}_1}) < \mathfrak{S}(\mathcal{L}_{\mathcal{D}_2})$
  - If  $\mathcal{L}_{\mathcal{D}_1} = \mathcal{L}_{\mathcal{D}_2}$  then  $\mathfrak{S}(\mathcal{L}_{\mathcal{D}_1}) \sim \mathfrak{S}(\mathcal{L}_{\mathcal{D}_2})$
1. If  $\mathcal{L}_{\mathcal{D}_1} > \mathcal{L}_{\mathcal{D}_2}$  then  $\mathfrak{H}(\mathcal{L}_{\mathcal{D}_1}) > \mathfrak{H}(\mathcal{L}_{\mathcal{D}_2})$
  2. If  $\mathcal{L}_{\mathcal{D}_1} < \mathcal{L}_{\mathcal{D}_2}$  then  $\mathfrak{H}(\mathcal{L}_{\mathcal{D}_1}) < \mathfrak{H}(\mathcal{L}_{\mathcal{D}_2})$
  3. If  $\mathcal{L}_{\mathcal{D}_1} = \mathcal{L}_{\mathcal{D}_2}$  then  $\mathfrak{H}(\mathcal{L}_{\mathcal{D}_1}) \sim \mathfrak{H}(\mathcal{L}_{\mathcal{D}_2})$

## 3. LINEAR DIOPHANTINE FUZZY SHORTEST PATH PROBLEM

In this section, the mathematical formulation for Linear Diophantine Fuzzy shortest path is presented. Consider a graph  $\mathfrak{G} = (\mathfrak{V}, \mathfrak{E})$  is an LDF-directed graph, where  $\mathfrak{V} = \{s=1, 2, \dots, e = m\}$  and  $\mathfrak{V} \times \mathfrak{V} = \mathfrak{E} = \{(i, j) : i, j \in \mathfrak{V}, i \neq j\}$  denotes the vertex and edge set, respectively. The ordered pairs  $(i, j)$  represent the two different vertices that are connected by an edge of the graph  $i, j \in \mathfrak{V}$ . In a connected network, the arc source node  $s$  and destination node  $t$ . There is assumed to be

only one path from node  $i$  to node  $j$ . The path  $p_{ij}$  from node  $i$  to node  $j$  is a sequence of arcs  $p_{ij} = \{(i, i_1), (i_1, i_2), \dots, (i_k, j)\}$  in which each arc's initial node is same as the corresponding destination node. The problem is to identify the optimal path from node  $s$  to  $t$  for each arc-related parameter in terms of cost (or time, or distance, etc). In terms of LDFNs, this parameter is assumed to be  $c_{ij} = (\alpha_{\mathfrak{R}}(\mathbf{ab}), m_{\mathfrak{R}}(\mathbf{ab}), \beta_{\mathfrak{R}}(\mathbf{ab}), n_{\mathfrak{R}}(\mathbf{ab}))$  where  $m_{\mathfrak{R}}(\mathbf{ab})$  is the membership grade and  $n_{\mathfrak{R}}(\mathbf{ab})$  is the non-membership grade and  $\alpha_{\mathfrak{R}}(\mathbf{ab}), \beta_{\mathfrak{R}}(\mathbf{ab})$  are the reference parameters. The arc  $i-j$  represents the distance of the path in the shortest path problem. The main objective of the shortest path is to find the optimal path and minimum cost/distance/time from source node  $s$  to the destination node  $t$ . Due to ambiguous situations, LDFN arc values are considered in the SP problem. Therefore, the resulting problem is called a Linear Diophantine Fuzzy shortest path problem (LDFSPP). The mathematical model of the problem is formulated as:

$$\begin{aligned} \text{Min } \tilde{Z} &= \sum_{i=1}^m \sum_{j=1}^m \tilde{c}^p x_{ij} \\ \text{s.t. } \sum_{j=1}^m x_{ij} - \sum_{k=1}^m x_{ki} &= \begin{cases} 1 & i = 1 \\ 0 & i \neq 1, m \\ -1 & i = m \end{cases} \\ x_{ij} &\geq 0, i, j = 1, 2, \dots, m \end{aligned} \quad (3)$$

If the arc  $(i, j)$  is in the path, then  $x_{ij} = 1$  else  $x_{ij} = 0$ . Let  $T_{st}$  denote the set of all paths from node  $s$  to node  $t$ . Let  $\tilde{\mathfrak{C}}_{ij}^p$  be the Linear Diophantine Fuzzy cost of path from the node  $u$  to node  $v$ .

### 3.1 The extended Bellman-Ford algorithm in Linear Diophantine Fuzzy environment

Let us consider acyclic-directed graph  $\mathfrak{G} = (\mathfrak{V}, \mathfrak{E})$  from source node  $s$  to the destination node  $t$  with Linear Diophantine Fuzzy arc weights. The Bellman dynamic programming is used to determine the shortest path by the forward pass calculation. The Extended Bellman-Ford dynamic programming under Linear Diophantine Fuzzy environment is formulated as:

$$\begin{aligned} \text{Initialization Step: Set the Source node as } g(1) &= \langle 0, 1 \rangle \langle 0, 1 \rangle \\ \text{Main Step: } \min_{\alpha < \beta} [g(\alpha) + \mathfrak{d}_{\alpha\beta}] & \end{aligned} \quad (4)$$

here,  $\mathfrak{d}_{\alpha\beta}$  is the directed LDFN edge weight,  $g(\alpha)$  is the LDFN arc length of the shortest path from the source node to destination node. The diagram illustrating the proposed model is depicted in Figure 2.

**Theorem:** If a LDF directed graph  $\mathfrak{G} = (\mathfrak{V}, \mathfrak{E})$  contains no non-negative weight cycles then after Bellman-Ford completes execution  $\mathfrak{d}[v] = \delta(s, v)$  for all  $v \in \mathfrak{V}$ .

**Proof:** Let  $v \in \mathfrak{V}$ . If there is a path  $p = \langle v_0, v_1, \dots, v_k \rangle$  then  $v_0 = s$  to  $v_k = v$ . Then a path  $p$  is a simple path with minimum LDF number of edges. If there is no negative LDFN edge cycle then it implies  $p$  is simple.

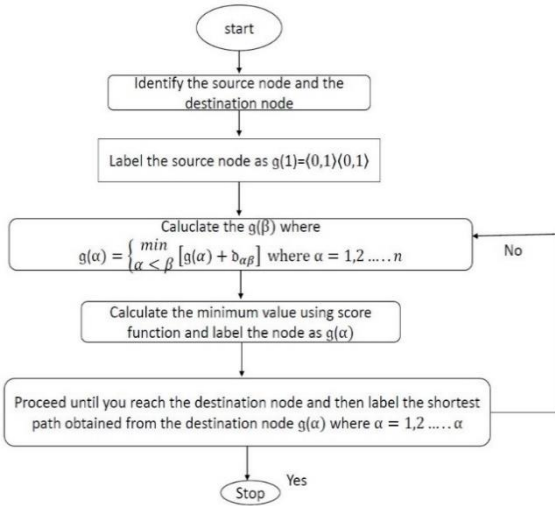
$$\Rightarrow k \leq \mathfrak{d}[v].$$

After one visit through all the LDFN edges  $E$ , we have  $\mathfrak{d}[v_1] = \delta(s, v_1)$  because the LDFN edge  $v_0, v_1$  is relaxed. After the second visit through all the LDFN edges, we have  $\mathfrak{d}[v_2] = \delta(s, v_2)$  because the LDFN edge  $(v_1, v_2)$ . Similarly, after  $k$  visits,  $\mathfrak{d}[v_k] = \delta(s, v_k)$  which implies  $|\mathfrak{V}| - 1 \Rightarrow$  all reachable vertex has

$\delta$  values. Therefore, the Bellman-Ford executes  $d[v]=\delta(s, v)$  for all  $v \in \mathfrak{B}$  when there are no non-negative weight cycles.

**Corollary:** If a value  $\delta[v]$  fails to converge after  $|\mathfrak{B}|-1$  visits there exists a negative LDFN weight cycle reachable from  $s$  because we will be able to relax the LDFN edge and reduce the LDFN weight after added vertex that form a cycle. Therefore, the cycle will be negative.

**Proof:** After  $|\mathfrak{B}|-1$  visits. If there is an LDFN edge that can be relaxed. (i.e.) current shortest path from  $s$  to some vertex  $v$  which is reachable is not simple. Having repeated vertex,  $\Rightarrow$  then there exists a negative weight cycle.



**Figure 2.** Flow chart diagram representing the algorithm

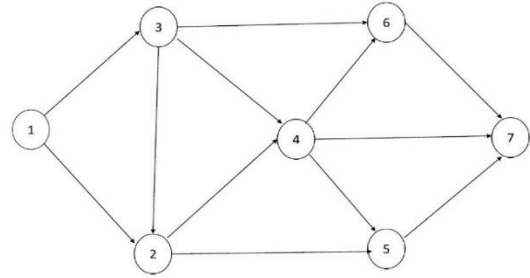
Pseudocode for the proposed Linear Diophantine Fuzzy Bellman-Ford Algorithm (LDFBFA)	
1.	Function Linear Diophantine Fuzzy Bellman-Ford Algorithm ( $\mathfrak{G}, s$ )
2.	For each vertex $\mathfrak{B} \in \mathfrak{G}$ do
3.	$\delta[v] \leftarrow \infty$
4.	end for
5.	$\delta[s] \leftarrow \langle 0,1 \rangle \langle 0,1 \rangle$
6.	for $i \leftarrow 1$ to $ \mathfrak{B} $ do
7.	relaxed $\leftarrow$ false
8.	for each $(\alpha\beta) \in \mathfrak{G}$ do
9.	if $g[\alpha] > g[\beta] + \delta(\alpha, \beta)$ then
10.	$\mathfrak{S}(g[\alpha] > \mathfrak{S}(g[\beta] + \delta(\alpha, \beta)))$ do
11.	$g[\alpha] \leftarrow g[\beta] + \delta(\alpha, \beta)$
12.	relaxed $\leftarrow$ true
13.	end if
14.	end for
15.	if relaxed=false then
16.	exit the loop
17.	end if
18.	end for
19.	for each $(\alpha\beta) \in \mathfrak{G}$ do
20.	if $\mathfrak{S}(g[\alpha] < \mathfrak{S}(g[\beta] + \delta(\alpha, \beta)))$
21.	return false
22.	end if
23.	end for
24.	return true

### 3.2 Numerical illustration

This section is based on adaptable numerical problem to illustrate the possible implementation of the proposed

algorithm.

**Example 1:** Let us consider a disaster road network Figure 3, that occurred in Wenling City with Linear Diophantine Fuzzy distance discussed by the study [18]. The arc values for the network in terms of LDF Distance are shown in the Table 1.



**Figure 3.** The graph with Linear Diophantine Fuzzy distance

**Table 1.** Arc value in term of Linear Diophantine Fuzzy distance for Example 1

Edges	LDFN	Edges	LDFN
(1,2)	$\langle \langle 0.81, 0.37 \rangle, \langle 0.51, 0.18 \rangle \rangle$	(3,6)	$\langle \langle 0.91, 0.73 \rangle, \langle 0.46, 0.18 \rangle \rangle$
(1,3)	$\langle \langle 0.93, 0.68 \rangle, \langle 0.53, 0.12 \rangle \rangle$	(4,5)	$\langle \langle 0.64, 0.29 \rangle, \langle 0.37, 0.28 \rangle \rangle$
(2,4)	$\langle \langle 0.74, 0.47 \rangle, \langle 0.43, 0.32 \rangle \rangle$	(4,6)	$\langle \langle 0.87, 0.39 \rangle, \langle 0.25, 0.22 \rangle \rangle$
(2,5)	$\langle \langle 0.93, 0.63 \rangle, \langle 0.46, 0.29 \rangle \rangle$	(4,7)	$\langle \langle 0.78, 0.57 \rangle, \langle 0.45, 0.21 \rangle \rangle$
(3,2)	$\langle \langle 0.94, 0.58 \rangle, \langle 0.58, 0.13 \rangle \rangle$	(5,7)	$\langle \langle 0.73, 0.68 \rangle, \langle 0.41, 0.37 \rangle \rangle$
(3,4)	$\langle \langle 0.64, 0.21 \rangle, \langle 0.37, 0.28 \rangle \rangle$	(6,7)	$\langle \langle 0.83, 0.43 \rangle, \langle 0.51, 0.15 \rangle \rangle$

**Iteration 1:** Start from the source node. Let the  $g(1) = \langle 0,1 \rangle \langle 0,1 \rangle$  and label  $g(1) = \langle \langle 0, 1 \rangle \langle 0, 1 \rangle, 1 \rangle$ .

**Iteration 2:** Consider the node 1 as  $\alpha$  and node 2 as  $\beta$ . Now relax the edges using Eq. (4). Using the score function [1] to choose the minimum one and label it as temporary node.

$$g(2) = \min_{\alpha < 2} [g(\alpha) + \delta_{\alpha 2}] = g(1) + \delta_{12} = \langle 0, 1 \rangle \langle 0, 1 \rangle + \langle 0.81, 0.37 \rangle \langle 0.51, 0.18 \rangle = \langle 0.81, 0.37 \rangle \langle 0.51, 0.18 \rangle \in \langle \langle 0.81, 0.37 \rangle \langle 0.51, 0.18 \rangle \rangle = 0.385$$

Therefore, Label  $g(2) = \langle \langle 0.81, 0.37 \rangle \langle 0.51, 0.18 \rangle, 1 \rightarrow 2 \rangle$

**Iteration 3:** Repeat the process with the node 3 and relax all edges using the Eq. (4).

$$g(3) = \min_{\alpha < 3} [g(\alpha) + \delta_{\alpha 3}] = g(1) + \delta_{13} = \langle 0, 1 \rangle \langle 0, 1 \rangle + \langle 0.93, 0.68 \rangle \langle 0.53, 0.12 \rangle = \langle 0.93, 0.68 \rangle \langle 0.53, 0.12 \rangle$$

The score function 1 is used to identify the minimum one and label it as temporary node,  $\mathfrak{S}(\langle 0.93, 0.68 \rangle \langle 0.53, 0.12 \rangle) = 0.33$  and Label  $g(3) = \langle \langle 0.93, 0.68 \rangle \langle 0.53, 0.12 \rangle, 1 \rightarrow 3 \rangle$

**Iteration 4:** Repeat the same process for the node 4 and relax all the edges using the equation and using score function pick the minimum value and label it as temporary node.

$$g(4) = \min_{\alpha < 4} [g(\alpha) + \delta_{\alpha 4}] = \min \{ g(2) + \delta_{24}, g(3) + \delta_{34} \} = \min \{ \langle 0.81, 0.37 \rangle \langle 0.51, 0.18 \rangle + \langle 0.74, 0.47 \rangle \langle 0.43, 0.32 \rangle, \langle 0.93, 0.68 \rangle \langle 0.53, 0.12 \rangle + \langle 0.64, 0.21 \rangle \langle 0.37, 0.28 \rangle \} = \min \{ \langle 0.9506, 0.1739 \rangle \langle 0.7207, 0.057 \rangle, \langle 0.9748, 0.1428 \rangle \langle 0.7039, 0.33 \rangle \}$$

Using Score Function 1,  $\mathfrak{S}(\langle 0.9506, 0.1739 \rangle \langle 0.7207, 0.057 \rangle) = 0.7202$   $\mathfrak{S}(\langle 0.9748, 0.1428 \rangle \langle 0.7039, 0.33 \rangle) = 0.7514$

Therefore,  $\min \{ 0.7202, 0.7514 \} = 0.7202$ .

Hence label  $g(4) = \langle (0.9506, 0.1739) \langle 0.7207, 0.057 \rangle, 1 \rightarrow 2 \rightarrow 4 \rangle$

**Iteration 5:** Visit each edge that reaches the vertex 5 from the source node using the Eq. (4). Using the score function 1 pick the minimum one and label it as temporary vertex. Repeat this process for  $|\mathfrak{B}|-1$  times.

$$g(5) = \min_{\alpha < 5} [g(\alpha) + d_{\alpha 5}] = \{g(2) + d_{25}, g(4) + d_{45}\} = \min \{ \langle 0.81, 0.37 \rangle \langle 0.51, 0.18 \rangle + \langle 0.93, 0.63 \rangle \langle 0.46, 0.29 \rangle, \langle 0.9506, 0.1739 \rangle \langle 0.7207, 0.057 \rangle + \langle 0.64, 0.29 \rangle \langle 0.37, 0.28 \rangle \} = \min \{ \langle 0.9867, 0.2233 \rangle \langle 0.7354, 0.0522 \rangle, \langle 0.982, 0.0493 \rangle \langle 0.8236, 0.0168 \rangle \}$$

Hence  $\ominus (\langle 0.9867, 0.2233 \rangle \langle 0.7354, 0.0522 \rangle) = 0.7233$   
 $\ominus (\langle 0.982, 0.0493 \rangle \langle 0.8236, 0.0168 \rangle) = 0.8695$ . Therefore,  $\min \{0.7233, 0.8695\} = 0.7233$

The minimum node is chosen and labeled  $g(5) = \langle \langle 0.9867, 0.2233 \rangle \langle 0.7354, 0.0522 \rangle, 1 \rightarrow 2 \rightarrow 5 \rangle$ .

**Iteration 6:** Relax all edges that are reachable from the vertex 6 using the Eq. (4) and to fix the temporary node use the 1 score function.

$$g(6) = \min_{\alpha < 6} [g(\alpha) + d_{\alpha 6}] = \min \{g(3) + d_{36}, g(4) + d_{46}\} = \min \{ \langle 0.93, 0.68 \rangle \langle 0.53, 0.12 \rangle + \langle 0.91, 0.73 \rangle \langle 0.46, 0.18 \rangle, \langle 0.9506, 0.1739 \rangle \langle 0.7207, 0.057 \rangle + \langle 0.97, 0.39 \rangle \langle 0.25, 0.22 \rangle \} = \min \{ \langle 0.9937, 0.4964 \rangle \langle 0.7462, 0.0216 \rangle, \langle 0.9935, 0.0663 \rangle \langle 0.79, 0.011 \rangle \}$$

Hence  $\ominus (\langle 0.9937, 0.4964 \rangle \langle 0.7462, 0.0216 \rangle) = 0.61095$   
 $\ominus (\langle 0.9935, 0.0663 \rangle \langle 0.79, 0.011 \rangle) = 0.8536$

Therefore,  $\min \{0.61095, 0.8536\} = 0.61095$ .

Hence, Label  $g(6) = \langle \langle 0.9937, 0.4964 \rangle \langle 0.7462, 0.0216 \rangle, 1 \rightarrow 3 \rightarrow 6 \rangle$

**Iteration 7:** Visit all the edges that are reachable to the vertex 7 (destination node) using 4 and use the score function to find the minimum value.

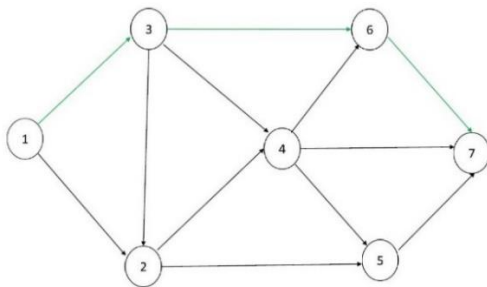
$$g(7) = \min_{\alpha < 7} [g(\alpha) + d_{\alpha 7}] = \min \{g(4) + d_{47}, g(5) + d_{57}, g(6) + d_{67}\} = \min \{ \langle 0.9056, 0.1739 \rangle \langle 0.7207, 0.057 \rangle + \langle 0.78, 0.57 \rangle \langle 0.45, 0.21 \rangle, \langle 0.986, 0.223 \rangle \langle 0.74, 0.052 \rangle + \langle 0.73, 0.68 \rangle \langle 0.41, 0.37 \rangle, \langle 0.99, 0.49 \rangle \langle 0.74, 0.22 \rangle + \langle 0.83, 0.43 \rangle \langle 0.51, 0.15 \rangle \} = \min \{ \langle 0.989, 0.096 \rangle \langle 0.846, 0.012 \rangle, \langle 0.99, 0.149 \rangle \langle 0.846, 0.018 \rangle, \langle 0.998, 0.2107 \rangle \langle 0.8726, 0.0003 \rangle \}$$

Hence,  $\ominus (\langle 0.989, 0.096 \rangle \langle 0.846, 0.012 \rangle) = 0.8635$   
 $\ominus (\langle 0.99, 0.149 \rangle \langle 0.846, 0.018 \rangle) = 0.834$

$\ominus (\langle 0.998, 0.2107 \rangle \langle 0.8726, 0.0003 \rangle) = 0.8301$

Therefore,  $\min \{0.8635, 0.834, 0.8301\} = 0.8301$

Therefore, choose the minimum node and Label  $g(7) = \langle \langle 0.998, 0.2107 \rangle \langle 0.8726, 0.0003 \rangle, 1 \rightarrow 3 \rightarrow 6 \rightarrow 7 \rangle$ . The shortest path and the shortest distance using the proposed algorithm for the (Figure 1) are  $1 \rightarrow 3 \rightarrow 6 \rightarrow 7$  and 0.8301 respectively. Figure 4 represents the optimal path using the proposed method. Table 2 represents the LDFN optimal values at each node using the proposed method.

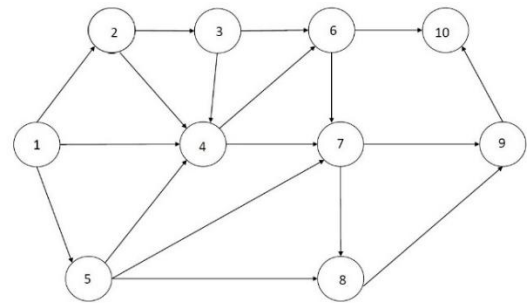


**Figure 4.** The Shortest path of the network with Linear Diophantine Fuzzy distance using the proposed algorithm

**Example 2:** An example problem has been considered a wireless charging approach method for sensor networks as shown in Figure 5 from the study [40] for the proposed method. Table 3 presents the values of arcs expressed in terms of Linear Diophantine Fuzzy numbers for the network.

**Table 2.** The optimal value of LDFN at every node utilizing the proposed method in Example 1

Node $\alpha$	Shortest LDFN Path from Node 1 to $\alpha$	$g(\alpha)$
2	1→2	$\langle 0.81, 0.37 \rangle \langle 0.51, 0.18 \rangle$
3	1→3	$\langle 0.93, 0.68 \rangle \langle 0.53, 0.12 \rangle$
4	1→2→4	$\langle 0.9506, 0.1739 \rangle \langle 0.7207, 0.057 \rangle$
5	1→2→5	$\langle 0.9867, 0.2233 \rangle \langle 0.7354, 0.0522 \rangle$
6	1→3→6	$\langle 0.9937, 0.4964 \rangle \langle 0.7462, 0.0216 \rangle$
7	1→3→6→7	$\langle 0.998, 0.2107 \rangle \langle 0.8726, 0.0003 \rangle$



**Figure 5.** The graph with Linear Diophantine Fuzzy distance

**Table 3.** Arc value in term of Linear Diophantine Fuzzy distance for Example 2

Edges	LDFN
(1,2)	$\langle \langle 0.7, 0.6 \rangle, \langle 0.5, 0.3 \rangle \rangle$
(1,4)	$\langle \langle 0.4, 0.4 \rangle, \langle 0.6, 0.4 \rangle \rangle$
(1,5)	$\langle \langle 0.7, 0.8 \rangle, \langle 0.5, 0.3 \rangle \rangle$
(2,3)	$\langle \langle 0.9, 0.7 \rangle, \langle 0.6, 0.2 \rangle \rangle$
(2,4)	$\langle \langle 0.6, 0.9 \rangle, \langle 0.8, 0.2 \rangle \rangle$
(3,4)	$\langle \langle 0.8, 0.7 \rangle, \langle 0.4, 0.3 \rangle \rangle$
(3,6)	$\langle \langle 0.8, 0.2 \rangle, \langle 0.3, 0.2 \rangle \rangle$
(4,5)	$\langle \langle 0.9, 0.8 \rangle, \langle 0.5, 0.3 \rangle \rangle$
(4,6)	$\langle \langle 0.6, 0.5 \rangle, \langle 0.4, 0.1 \rangle \rangle$
(4,7)	$\langle \langle 0.8, 0.7 \rangle, \langle 0.4, 0.3 \rangle \rangle$
(5,7)	$\langle \langle 0.8, 0.7 \rangle, \langle 0.4, 0.3 \rangle \rangle$
(5,8)	$\langle \langle 0.9, 0.2 \rangle, \langle 0.3, 0.1 \rangle \rangle$
(6,7)	$\langle \langle 0.4, 0.3 \rangle, \langle 0.5, 0.3 \rangle \rangle$
(7,8)	$\langle \langle 0.9, 0.4 \rangle, \langle 0.6, 0.4 \rangle \rangle$
(6,10)	$\langle \langle 0.7, 0.5 \rangle, \langle 0.2, 0.1 \rangle \rangle$
(8,9)	$\langle \langle 0.3, 0.2 \rangle, \langle 0.6, 0.4 \rangle \rangle$

**Iteration 1:** Start from the source node.

Let the  $g(1) = \langle 0, 1 \rangle \langle 0, 1 \rangle$  and label  $g(1) = \langle \langle 0, 1 \rangle \langle 0, 1 \rangle, 1 \rangle$

**Iteration 2:** Consider the node 1 as  $\alpha$  and node 2 as  $\beta$ . Now relax the edges which reaches the node 2 using Eq. (4). Using the score function [1] to choose the minimum one and label it as temporary node.

$$g(2) = \min_{\alpha < 2} [g(\alpha) + d_{\alpha 2}] = g(1) + d_{12} = \langle 0, 1 \rangle \langle 0, 1 \rangle + \langle 0.7, 0.6 \rangle \langle 0.5, 0.3 \rangle = \langle 0.7, 0.6 \rangle \langle 0.5, 0.3 \rangle \ominus (\langle 0.7, 0.6 \rangle \langle 0.5, 0.3 \rangle) = 0.15$$

Therefore, Label  $g(2)=[\langle 0.7, 0.6 \rangle \langle 0.5, 0.3 \rangle, 1 \rightarrow 2]$

**Iteration 3:** Repeat the above process with the node 3 and relax all edges using the Eq. (4).

$$g(3) = \min_{\alpha < 3} [g(\alpha) + d_{\alpha 3}] = g(2) + d_{13} = \langle 0.7, 0.6 \rangle \langle 0.5, 0.3 \rangle + \langle 0.9, 0.7 \rangle \langle 0.6, 0.2 \rangle = \langle 0.97, 0.42 \rangle \langle 0.8, 0.06 \rangle$$

The score function [1] is used to identify the minimum one and label it as temporary node, Therefore,  $\ominus(\langle 0.97, 0.42 \rangle \langle 0.8, 0.06 \rangle) = 0.65$ .

Hence, Label  $g(3)=[\langle 0.97, 0.42 \rangle \langle 0.8, 0.06 \rangle, 1 \rightarrow 3]$

**Iteration 4:** Repeat the same process for the node 4 and relax all the edges using the equation and using score function pick the minimum value and label it as temporary node.

$$g(4) = \min_{\alpha < 4} [g(\alpha) + d_{\alpha 4}] = \min \{g(1) + d_{14}, g(2) + d_{24}, g(3) + d_{34}\} = \min \{ \langle 0, 1 \rangle \langle 0, 1 \rangle + \langle 0.4, 0.2 \rangle \langle 0.6, 0.4 \rangle, \langle 0.7, 0.6 \rangle \langle 0.5, 0.3 \rangle + \langle 0.6, 0.9 \rangle \langle 0.8, 0.2 \rangle, \langle 0.97, 0.42 \rangle, \langle 0.8, 0.06 \rangle + \langle 0.8, 0.7 \rangle \langle 0.4, 0.3 \rangle \} = \min \{ \langle 0.4, 0.2 \rangle \langle 0.6, 0.4 \rangle, \langle 0.88, 0.54 \rangle \langle 0.9, 0.06 \rangle, \langle 0.99, 0.29 \rangle \langle 0.88, 0.02 \rangle \}$$

Hence,  $\ominus(\langle 0.4, 0.2 \rangle \langle 0.6, 0.4 \rangle) = 0.2 \ominus(\langle 0.88, 0.54 \rangle \langle 0.9, 0.06 \rangle) = 0.59 \ominus(\langle 0.99, 0.29 \rangle \langle 0.88, 0.02 \rangle) = 0.78$

Therefore,  $\min \{0.2, 0.59, 0.78\} = 0.2$

Hence label  $g(4)=[\langle 0.4, 0.2 \rangle \langle 0.6, 0.4 \rangle, 1 \rightarrow 4]$

**Iteration 5:** Visit each edge that reaches the vertex 5 from the source node using the Eq. (4). Using the score function I pick the minimum one and label it as temporary vertex. Repeat this process for  $|\mathcal{B}| - 1$  times.

$$g(5) = \min_{\alpha < 5} [g(\alpha) + d_{\alpha 5}] = \min \{g(1) + d_{15}, g(4) + d_{45}\} = \min \{ \langle 0, 1 \rangle \langle 0, 1 \rangle + \langle 0.7, 0.8 \rangle \langle 0.5, 0.3 \rangle, \langle 0.4, 0.2 \rangle \langle 0.6, 0.4 \rangle + \langle 0.9, 0.8 \rangle \langle 0.5, 0.3 \rangle \} = \min \{ \langle 0.7, 0.8 \rangle \langle 0.5, 0.3 \rangle, \langle 0.94, 0.16 \rangle \langle 0.8, 0.24 \rangle \}$$

Hence,  $\ominus(\langle 0.7, 0.8 \rangle \langle 0.5, 0.3 \rangle) = 0.05 \ominus(\langle 0.94, 0.16 \rangle \langle 0.8, 0.24 \rangle) = 0.67$

Therefore,  $\min \{0.05, 0.67\} = 0.05$

Hence, the minimum value is chosen and Labeled  $g(5) = [\langle 0.7, 0.8 \rangle \langle 0.5, 0.3 \rangle, 1 \rightarrow 5]$ .

**Iteration 6:** Relax all edges that are reachable from the vertex 6 using the Eq. (4) and to fix the temporary node use the 1 score function.

$$g(6) = \min_{\alpha < 6} [g(\alpha) + d_{\alpha 6}] = \min \{g(3) + d_{36}, g(4) + d_{46}\} = \min \{ \langle 0.97, 0.42 \rangle \langle 0.8, 0.06 \rangle + \langle 0.8, 0.2 \rangle \langle 0.3, 0.2 \rangle, \langle 0.4, 0.2 \rangle \langle 0.6, 0.4 \rangle + \langle 0.6, 0.5 \rangle \langle 0, 0.1 \rangle \} = \min \{ \langle 0.99, 0.08 \rangle \langle 0.86, 0.01 \rangle, \langle 0.76, 0.1 \rangle \langle 0.76, 0.04 \rangle \}$$

Hence,  $\ominus(\langle 0.99, 0.08 \rangle \langle 0.86, 0.01 \rangle) = 0.88 \ominus(\langle 0.76, 0.1 \rangle \langle 0.76, 0.04 \rangle) = 0.69$

Therefore,  $\min \{0.88, 0.69\} = 0.69$

Hence, Label  $g(6)=[\langle 0.76, 0.1 \rangle \langle 0.76, 0.04 \rangle, 1 \rightarrow 4 \rightarrow 6]$ .

**Iteration 7:** Visit all the edges that are reachable to the vertex 7 (destination node) using 4 and use the score function to find the minimum value.

$$g(7) = \min_{\alpha < 7} [g(\alpha) + d_{\alpha 7}] = \min \{g(4) + d_{47}, g(5) + d_{57}, g(6) + d_{67}\} = \min \{ \langle 0.4, 0.2 \rangle \langle 0.6, 0.4 \rangle + \langle 0.9, 0.5 \rangle \langle 0.6, 0.3 \rangle, \langle 0.7, 0.8 \rangle \langle 0.5, 0.3 \rangle + \langle 0.8, 0.7 \rangle \langle 0.4, 0.3 \rangle, \langle 0.76, 0.01 \rangle \langle 0.76, 0.04 \rangle + \langle 0.4, 0.3 \rangle \langle 0.5, 0.3 \rangle \} = \min \{ \langle 0.94, 0.1 \rangle \langle 0.84, 0.12 \rangle, \langle 0.94, 0.56 \rangle \langle 0.7, 0.09 \rangle, \langle 0.86, 0.0003 \rangle \langle 0.88, 0.01 \rangle \}$$

Hence,  $\ominus(\langle 0.94, 0.1 \rangle \langle 0.84, 0.12 \rangle) = 0.78 \ominus(\langle 0.94, 0.56 \rangle \langle 0.7, 0.09 \rangle) = 0.49 \ominus(\langle 0.86, 0.0003 \rangle \langle 0.88, 0.01 \rangle) = 0.86$

Therefore,  $\min \{0.78, 0.49, 0.86\} = 0.49$

Hence label  $g(7)=[\langle 0.94, 0.56 \rangle \langle 0.7, 0.09 \rangle, 1 \rightarrow 5 \rightarrow 7]$

**Iteration 8:** Repeat the process for the vertex 8 and relax the edges that reaches the vertex 8 using the Eq. (1) and to fix the temporary node use the Eq. (1).

$$g(8) = \min_{\alpha < 8} [g(\alpha) + d_{\alpha 8}] = \min \{g(5) + d_{58}, g(7) + d_{78}\} = \min \{ \langle 0.7, 0.8 \rangle \langle 0.5, 0.3 \rangle + \langle 0.9, 0.2 \rangle \langle 0.3, 0.1 \rangle, \langle 0.94, 0.56 \rangle \langle 0.7, 0.09 \rangle + \langle 0.9, 0.4 \rangle \langle 0.6, 0.4 \rangle \} = \min \{ \langle 0.97, 0.16 \rangle \langle 0.65, 0.03 \rangle, \langle 0.99, 0.22 \rangle \langle 0.88, 0.04 \rangle \}$$

Hence  $\ominus(\langle 0.97, 0.16 \rangle \langle 0.65, 0.03 \rangle) = 0.72$

$\ominus(\langle 0.99, 0.22 \rangle \langle 0.88, 0.04 \rangle) = 0.81$ .

Therefore,  $\min \{0.72, 0.81\} = 0.72$

Hence, the minimum node is chosen and labeled  $g(8)=[\langle 0.97, 0.16 \rangle \langle 0.65, 0.03 \rangle, 1 \rightarrow 5 \rightarrow 8]$ .

**Iteration 9:** Repeat the same process for the vertex 9.

$$g(9) = \min_{\alpha < 9} [g(\alpha) + d_{\alpha 9}] = \min \{g(7) + d_{79}, g(8) + d_{89}\} = \min \{ \langle 0.94, 0.56 \rangle \langle 0.7, 0.09 \rangle + \langle 0.5, 0.2 \rangle \langle 0.4, 0.3 \rangle, \langle 0.97, 0.16 \rangle \langle 0.65, 0.03 \rangle + \langle 0.3, 0.2 \rangle \langle 0.6, 0.4 \rangle \} = \min \{ \langle 0.97, 0.11 \rangle \langle 0.82, 0.02 \rangle, \langle 0.98, 0.3 \rangle \langle 0.86, 0.01 \rangle \}$$

Hence  $\ominus(\langle 0.97, 0.11 \rangle \langle 0.82, 0.02 \rangle) = 0.83$

$\ominus(\langle 0.98, 0.3 \rangle \langle 0.86, 0.01 \rangle) = 0.9$

Therefore,  $\min \{0.83, 0.9\} = 0.83$

Hence, the minimum node is chosen and labeled  $g(9)=[\langle 0.97, 0.11 \rangle \langle 0.82, 0.02 \rangle, 1 \rightarrow 5 \rightarrow 7 \rightarrow 9]$

**Iteration 10:** Repeat the above process for the vertex 10

$$g(10) = \min_{\alpha < 10} [g(\alpha) + d_{\alpha 10}] = \min \{g(6) + d_{610}, g(9) + d_{910}\} = \min \{ \langle 0.76, 0.01 \rangle \langle 0.76, 0.04 \rangle + \langle 0.7, 0.5 \rangle \langle 0.2, 0.1 \rangle, \langle 0.97, 0.11 \rangle \langle 0.82, 0.02 \rangle + \langle 0.7, 0.3 \rangle \langle 0.5, 0.4 \rangle \} = \min \{ \langle 0.93, 0.0005 \rangle \langle 0.81, 0.0004 \rangle, \langle 0.99, 0.03 \rangle \langle 0.91, 0.0008 \rangle \}$$

Hence  $\ominus(\langle 0.93, 0.0005 \rangle \langle 0.81, 0.0004 \rangle) = 0.87$

$\ominus(\langle 0.99, 0.03 \rangle \langle 0.91, 0.0008 \rangle) = 0.93$

Therefore,  $\min \{0.87, 0.93\} = 0.87$

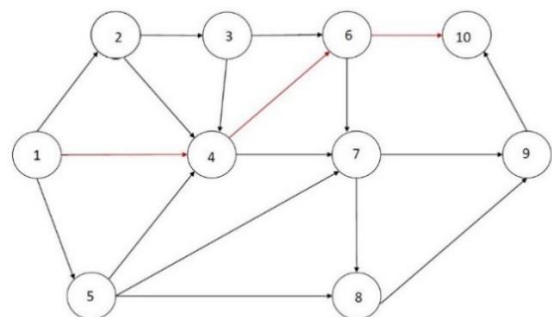
Hence, Label  $g(10)=[\langle 0.93, 0.0005 \rangle \langle 0.81, 0.0004 \rangle, 1 \rightarrow 4 \rightarrow 6 \rightarrow 10]$ .

The LDFN optimal value at each node using the proposed method is shown in Table 4.

Hence the shortest path is  $1 \rightarrow 4 \rightarrow 6 \rightarrow 10$  (is shown in Figure 6) and the shortest distance is  $\langle 0.93, 0.0005 \rangle \langle 0.81, 0.0004 \rangle$

**Table 4.** LDFN optimal value at each node using the proposed algorithm for Example 2

Node $\alpha$	Shortest LDFN Path from Node 1 to $\alpha$	$g(\alpha)$
2	1 → 2	$\langle 0.7, 0.6 \rangle \langle 0.5, 0.3 \rangle$
3	1 → 2 → 3	$\langle 0.97, 0.42 \rangle \langle 0.8, 0.06 \rangle$
4	1 → 4	$\langle 0.4, 0.2 \rangle \langle 0.6, 0.4 \rangle$
5	1 → 5	$\langle 0.7, 0.8 \rangle \langle 0.5, 0.3 \rangle$
6	1 → 4 → 6	$\langle 0.76, 0.01 \rangle \langle 0.76, 0.04 \rangle$
7	1 → 5 → 7	$\langle 0.94, 0.56 \rangle \langle 0.7, 0.09 \rangle$
8	1 → 5 → 8	$\langle 0.97, 0.16 \rangle \langle 0.65, 0.03 \rangle$
9	1 → 5 → 7 → 9	$\langle 0.97, 0.11 \rangle \langle 0.82, 0.03 \rangle$
10	1 → 4 → 6 → 10	$\langle 0.93, 0.0005 \rangle \langle 0.81, 0.0004 \rangle$



**Figure 6.** The shortest path for the given network with Linear Diophantine Fuzzy distance using the proposed algorithm

### 3.3 Result analysis

The shortest path  $1 \rightarrow 3 \rightarrow 6 \rightarrow 7$  and the optimal distance  $\langle 0.998, 0.2107 \rangle \langle 0.8726, 0.0003 \rangle$  obtained from the proposed algorithm give the same shortest path and the minimal distance when compared with the study [18]. Consequently, the LDFSPP employing the proposed approach benefited a rescue crew quickly arriving at their location within the shortest distance. Compared with the study [39], it is observed that the proposed work provides the most convincing results,  $1 \rightarrow 4 \rightarrow 6 \rightarrow 10$  and  $\langle 0.93, 0.0005 \rangle \langle 0.81, 0.0004 \rangle$  which helps to increase the network's lifetime. Hence our proposed technique is better suited for any network with reference parameters and satisfaction and dissatisfaction grades. Figure 7 illustrates the comparison between the Proposed method and the results from previous approaches. Computational development has to be implemented as the suggested method for extensive networks. This solution approach can apply to more extensive networks with various fuzzy environments.



Figure 7. Result analysis

Table 5. Advantages and restrictions for different FSPP

Types of Environments	Advantages	Restrictions
Fuzzy Environment	When the edge weight is uncertain this can be employed. It can be used in vagueness and unclear situations.	only the membership degree with the edge values can be applied. It is impotent for non-membership grades. It is ineffective when the sum of membership and non-membership is more than one. Eg: $0.53 + 0.49 = 1.02 \not\leq 1$
IFS	It can be applied with imprecise edge weight having both membership and non-membership values	When the sum of the square of the membership and non-membership extends beyond one, it is inadequate to apply. Eg: $(0.89)^2 + (0.75)^2 = 1.3546 \not\leq 1$
PyFS	When the total edge weight of the membership and non-membership degrees exceeds one, this environment may cope with approximate edge weight and can evaluate the optimal data. Eg: $(0.53)^2 + (0.49)^2 = 0.521$	
LDFS	This approach can be used when the reference parameter is provided in the problem. Eg: $(0.89) (0.32) + (0.75) (0.32) = 0.5248 < 1$	It cannot be executed if the arc weight has indeterminacy grade

The above Table 5 describes the Advantages and Restrictions of different fuzzy environment for the Fuzzy Shortest path problem.

### 4. CONCLUSIONS

The shortest path problem is a crucial area of study with applications spanning a broad range of real-world scenarios. Classical, fuzzy, IFS, and PyFS theories each offer unique potential in addressing the shortest path problem, but they require further exploration, particularly regarding the integration of reference parameters vital to the situation at hand.

In this paper, we have considered the Linear Diophantine Fuzzy arc cost and formulated its application to the shortest path problem. We have extended the Bellman-Ford algorithm to the Linear Diophantine Fuzzy shortest path and LDF distance/time/cost. Our focus was to establish LDF optimality constraints in directed network graphs and to develop a method to address them effectively. This involves applying the score function to calculate the weights of various paths, with edge weights represented by LDFNs.

One of the significant advantages of our approach is its efficiency in identifying the existence of a negative cycle. We have demonstrated our proposed work through a numerical example, identifying optimal results. We also compared our results with those found using existing methods.

Our proposed approach pays additional attention to reference parameters in unpredictable, critical situations. It enables the determination of the LDFN Shortest path of all vertices from a single source vertex with a time complexity of  $O(VE)$ .

As a future direction, we aim to apply our proposed method to large-scale, pragmatic, and varied fuzzy SPPs.

### REFERENCES

- [1] Zadeh, L.A. (1965). Fuzzy sets. *Information and Control*, 8(3): 338-353. [https://doi.org/10.1016/s0019-9958\(65\)90241-x](https://doi.org/10.1016/s0019-9958(65)90241-x)
- [2] Zadeh L.A. (1975). The concept of a linguistic variable and its application to approximate reasoning-III. *Information Sciences*, 9(1): 43-80. [https://doi.org/10.1016/0020-0255\(75\)90017-1](https://doi.org/10.1016/0020-0255(75)90017-1)
- [3] Atanassov, K.T. (1999). Intuitionistic Fuzzy Sets, *Intuitionistic Fuzzy Sets*. Physica, Heidelberg, 1-137.
- [4] Atanassov, K.T. (1999). Interval valued intuitionistic fuzzy sets. *Intuitionistic Fuzzy Sets*. Physica, Heidelberg, 139-177. <https://doi.org/10.1007/978-3-7908-1870-3-2>
- [5] Atanassov, K.T. (1986). Intuitionistic fuzzy sets. *Fuzzy Sets and Systems*, 20: 87-96. [https://doi.org/10.1016/S0165-0114\(86\)80034-3](https://doi.org/10.1016/S0165-0114(86)80034-3)
- [6] Yager, R.R. (2013). Pythagorean fuzzy subsets. In 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), pp. 57-61. <https://doi.org/10.1109/IFSANAFIPS.2013.6608375>
- [7] Yager, R.R. (2013). Pythagorean membership grades in multicriteria decision making. *IEEE Transactions on Fuzzy Systems*, 22(4): 958-965. <https://doi.org/10.1109/TFUZZ.2013.2278989>
- [8] Yager, R.R. (2016). Generalized orthopair fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 25(5): 1222-1230.

- <https://doi.org/10.1109/TFUZZ.2016.2604005>
- [9] Zhang, X., Xu, Z. (2014). Extension of TOPSIS to multiple criteria decision making with Pythagorean Fuzzy Sets. *International Journal of Intelligent Systems*, 29(12): 1061-1078. <https://doi.org/10.1002/int.21676>
- [10] Zhang, X. (2016). Multicriteria Pythagorean fuzzy decision analysis: A hierarchical QUALIFLEX approach with the closeness index-based ranking methods. *Information Sciences*, 330: 104-124. <https://doi.org/10.1016/j.ins.2015.10.012>
- [11] Fei, L., Deng, Y. (2020). Multi-criteria decision making in Pythagorean fuzzy environment. *Applied Intelligence*, 50: 537-561. <https://doi.org/10.1007/s10489-019-01532-2>
- [12] Peng, X., Ma, X. (2020). Pythagorean fuzzy multi-criteria decision making method based on CODAS with new score function. *Journal of Intelligent & Fuzzy Systems*, 38(3): 3307-3318. <https://doi.org/10.3233/jifs-190043>.
- [13] Garg, H. (2016). A novel correlation coefficients between Pythagorean Fuzzy Sets and its applications to decision-making processes. *International Journal of Intelligent Systems*, 31(12): 1234-1252. <https://doi.org/10.1002/int.21827>
- [14] Garg, H. (2016). A novel accuracy function under interval-valued Pythagorean fuzzy environment for solving multicriteria decision making problem. *Journal of Intelligent & Fuzzy Systems*, 31(1): 529-540. <https://doi.org/10.3233/IFS-162165>
- [15] Garg, H. (2018). A linear programming method based on an improved score function for interval-valued Pythagorean fuzzy numbers and its application to decision-making. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 26(1): 67-80. <https://doi.org/10.1142/s0218488518500046>
- [16] Rahman, K., Abdullah, S., Ali, A. (2019). Some induced aggregation operators based on interval-valued Pythagorean fuzzy numbers. *Granular Computing*, 4: 53-62. <https://doi.org/10.1007/s41066-018-0091-8>
- [17] Garg, H. (2016). A new generalized Pythagorean fuzzy information aggregation using Einstein operations and its application to decision making. *International Journal of Intelligent Systems*, 31(9): 886-920. <https://doi.org/10.1002/int.21809>
- [18] Parimala, M., Jafari, S., Riaz, M., Aslam, M. (2021). Applying the Dijkstra algorithm to solve a Linear Diophantine Fuzzy environment. *Symmetry*, 13(9): 1616. <https://doi.org/10.3390/sym13091616>
- [19] Liu, P., Wang, P. (2017). Some q-rung orthopair fuzzy aggregation operators and their applications to multiple-attribute decision making. *International Journal of Intelligent Systems*, 33(2): 259-280. <https://doi.org/10.1002/int.21927>.
- [20] Ali, M.I. (2018). Another view on q-rung Orthopair Fuzzy Sets. *International Journal of Intelligent Systems*, 33(11): 2139-2153. <https://doi.org/10.1002/int.22007>
- [21] Riaz, M., Hashmi, M.R. (2019). Linear Diophantine Fuzzy set and its applications towards multi-attribute decision-making problems. *Journal of Intelligent & Fuzzy Systems*, 37(4): 5417-5439. <https://doi.org/10.1155/2014/183607>
- [22] Dubois, D.J. (1980). *Fuzzy sets and systems: Theory and applications* (Vol. 144). Academic press.
- [23] Okada, S., Soper, T. (2000). A shortest path problem on a network with fuzzy arc lengths. *Fuzzy Sets and Systems*, 109(1): 129-140. [https://doi.org/10.1016/S0165-0114\(98\)00054-2](https://doi.org/10.1016/S0165-0114(98)00054-2)
- [24] Chuang, T.N., Kung, J.Y. (2005). The fuzzy shortest path length and the corresponding shortest path in a network. *Computers & Operations Research*, 32(6): 1409-1428. <https://doi.org/10.1016/j.cor.2003.11.011>
- [25] Kung, J.Y., Chuang, T.N. (2005). The shortest path problem with discrete fuzzy arc lengths. *Computers & Mathematics with Applications*, 49(2-3): 263-270. <https://doi.org/10.1016/j.camwa.2004.08.011>
- [26] Deng, Y., Chen, Y., Zhang, Y., Mahadevan, S. (2012). Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Applied Soft Computing*, 12(3): 1231-1237. <https://doi.org/10.1016/j.asoc.2011.11.011>
- [27] Gao, Y. (2011). Shortest path problem with uncertain arc lengths. *Computers & Mathematics with Applications*, 62(6): 2591-2600. <https://doi.org/10.1016/j.camwa.2011.07.058>
- [28] Mukherjee, S. (2012). Dijkstra's algorithm for solving the shortest path problem on networks under intuitionistic fuzzy environment. *Journal of Mathematical Modelling and Algorithms*, 11: 345-359. <https://doi.org/10.1007/s10852-012-9191-7>
- [29] Kannan, V., Appasamy, S., Kandasamy, G. (2022). Comparative study of fuzzy Floyd Warshall algorithm and the fuzzy rectangular algorithm to find the shortest path. In *AIP Conference Proceedings*, 2516: 1. <https://doi.org/10.1063/5.0110337>
- [30] Kumar, R., Edalatpanah, S.A., Mohapatra, H. (2020). Note on Optimal path selection approach for fuzzy reliable shortest path problem. *Journal of Intelligent & Fuzzy Systems*, 39(5): 7653-7656. <https://doi.org/10.3233/JIFS-200923>
- [31] Elizabethand, S., Sujatha, L. (2014). Fuzzy shortest path problem based on interval valued Fuzzy number matrices. *International Journal of Mathematical Sciences & Applications*, 8(1): 325-335.
- [32] Das, D., De, P.K. (2014). Shortest path problem under intuitionistic fuzzy setting. *International Journal of Computer Applications*, 105(1).
- [33] Baba, L. (2013). Shortest path problem on intuitionistic fuzzy network. *Annals of Pure and Applied Mathematics*, 5(1): 26-36.
- [34] Jayagowri, P., Ramani, G.G. (2014). Using trapezoidal intuitionistic fuzzy number to find optimized path in a network. *Advances in Fuzzy Systems*, 2014: 12-12. <https://doi.org/10.1155/2014/183607>
- [35] Gani, A.N., Jabarulla, M.M. (2010). On searching intuitionistic fuzzy shortest path in a network. *Applied Mathematical Sciences*, 4(69): 3447-3454.
- [36] Motameni, H., Ebrahimnejad, A. (2018). Constraint shortest path problem in a network with intuitionistic fuzzy arc weights. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp. 310-318. <https://doi.org/10.1007/978-3-319-91479-426>
- [37] Sujatha, L., Hyacinta, J.D. (2017). The shortest path problem on networks with intuitionistic fuzzy edge weights. *Glob J Pure Appl Math*, 13(7): 3285-3300.
- [38] Kumar, R., Jha, S., Singh, R. (2017). Shortest path problem in network with type-2 triangular fuzzy arc length. *Journal of Applied Research on Industrial Engineering*, 4(1): 1-7.



- <https://doi.org/10.22105/jarie.2017.48858>
- [39] Enayattabar, M., Ebrahimnejad, A., Motameni, H. (2019). Dijkstra algorithm for shortest path problem under interval-valued Pythagorean fuzzy environment. *Complex & Intelligent Systems*, 5(2): 93-100. <https://doi.org/10.1007/s40747-018-0083-y>
- [40] Prakash, K., Parimala, M., Garg, H., Riaz, M. (2022). Lifetime prolongation of a wireless charging sensor network using a mobile robot via Linear Diophantine Fuzzy graph environment. *Complex & Intelligent Systems*, 8(3): 2419-2434. <https://doi.org/10.1007/s40747-022-00653-5>
- [41] 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: 371-378. <https://doi.org/10.1007/s40747-019-0098-z>
- [42] Biswas, S.S., Alam, B., Doja, M.N. (2013). Generalization of Dijkstra's algorithm for extraction of shortest paths in directed multigraphs. *Journal of Computer Science*, 9(3): 377-382. <https://doi.org/10.3844/jcssp.2013.377.382>

#### NOMENCLATURE

q-ROFS	q-rung orthopair fuzzy set
PyFS	Pythagorean fuzzy set
LDFS	Linear Diophantine Fuzzy set
LDFNs	Linear Diophantine Fuzzy Numbers
SPP	Shortest Path Problem