NSS

Neutrosophic Sets and Systems, Vol. 65, 2024

University of New Mexico

A Novel Method for Solving the Time-Dependent Shortest Path Problem under Bipolar Neutrosophic Fuzzy Arc Values

Vidhya K[1], Saraswathi A[2,*] and Broumi Said[3]

[1]Department of Mathematics, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu - 603203,TamilNadu,India; vk3994@srmist.edu.in

[2]Department of Mathematics, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu - 603203,TamilNadu,India; saraswaa@srmist.edu.in

[3]Faculty of Science Ben MSik, University Hassan II, Casablanca, Morocco; broumisaid78@gmail.com

*Correspondence: saraswaa@srmist.edu.in;

Abstract. The Shortest path problem is highly relevant in our daily lives, addressing uncertainties like traffic conditions and weather variations. To handle such uncertainties, we utilize Fuzzy Numbers. This paper focuses on Bipolar Neutrosophic Fuzzy Numbers, which have dual positive and negative aspects. They provide a robust framework for representing arc (node/edge) weights, signifying uncertain travel times between nodes. Importantly, these weights can change over time in bipolar neutrosophic fuzzy graphs. Our study introduces an extended Bellman-Ford Algorithm for identifying optimal paths and minimum times with time-dependent Bipolar Neutrosophic Fuzzy arc weights. We demonstrate its effectiveness through a step-by-step numerical example and conduct a comparative analysis to evaluate its efficiency.

Keywords: Shortest Path Problem; Bipolar Neutrosophic Fuzzy Arc Weights; Bellman-ford Algorithm; Time-dependent Shortest path problem.

---

1. **Introduction**

The Shortest Path Problem (SPP) is a fundamental concept in graph theory and optimization, focusing on finding the most efficient route between two points in a network. It has broad applications in various fields, including transportation, logistics, telecommunications, and computer science. At its core, the Shortest Path Problem aims to identify the path with the minimum total cost or distance among all possible routes connecting two nodes in a graph. The "cost" could represent various factors, such as time, distance, financial expenses, or any other relevant metric depending on the specific context.

The need for evaluation methods in solving the Shortest Path Problem arises from its pervasive applicability and the desire to optimize resource utilization, minimize travel time, and enhance overall efficiency. Evaluation methods assess the effectiveness and performance of algorithms designed to solve the Shortest Path Problem in different scenarios. Evaluation methods for the Shortest Path Problem involve assessing the accuracy, efficiency, and scalability of algorithms developed to solve it. These methods contribute not only to practical applications but also to the broader field of algorithmic research, promoting advancements in optimization techniques and algorithm design. Real-world scenarios often introduce complexities like traffic congestion and adverse weather conditions. To address these challenges, Lotfi Zadeh (1965) pioneered the concept of fuzzy set theory [1]. Zadeh's groundbreaking work extended traditional set theory to fuzzy sets (FS), which encompass membership functions ranging from 0 to 1. He further introduced the notion of linguistic variables, representing values in the form of natural languages. When these linguistic terms are expressed using fuzzy sets defined over a universal set, they form what is known as a fuzzy linguistic variable [2]. Fuzzy sets, which allow elements to have degrees of belongingness to a set, were generalized by Atanassov [4] who introduced intuitionistic fuzzy sets (IFS). IFSs have membership and non-membership functions that add up to at most one for each element. Atanassov [4] also proposed interval-valued intuitionistic fuzzy sets (IVIFS), which are a further extension of IFSs with intervals as membership and non-membership values. IVIFS have a geometric representation and various operations defined on them. IFSs and IVIFS are widely used in many practical problems. However it does not provide the neutral or indeterminacy, when there is need in neutrality or lack of knowledge in expressing. Neutrosophic sets are a concept developed by Florentin Smarandache [8] to deal with problems that involve neutrality or indeterminacy as a key factor. Neutrosophic sets have three components: membership, non-membership, and indeterminacy. Single valued neutrosophic set (SVNS) which is a subclass of neutrosophic set, where the truth, indeterminacy, and falsity membership functions take values in the standard unit interval $[0, 1]$ and its applications is proposed by Sujit Das et al. [9]. Then, the interval-valued neutrosophic fuzzy sets (IVNFS) was implemented by Broumi et al. [10] and its relational operators were discussed. Bipolar fuzzy sets by Lee [34] - m-polarFS, built upon established fuzzy theorems, serve as a valuable framework for addressing the dual aspects of positive and negative behavior within the human mind. Later, there arise a drawback for positive and negative membership, Ali [11] proposed Bipolar Neutrosophic fuzzy sets (BNFS) and its operations in decsion-making. The application of fuzzy set theory has proven highly efficient in handling data characterized by imprecision, inaccuracy, and vagueness. One significant problem it addresses is the Fuzzy Shortest Path Problem (FSPP), which involves finding optimal paths among nodes in a graph while optimizing an objective function in a fuzzy environment. In pioneering work, Dubois [13] proposed an algorithm to solve FSPP and determine optimal weights. Subsequently, Klein [15] analyzed FSPP in terms of fuzzy mathematical programming, paving the way for further research and extensions of the concept. Expanding on these foundations, Okada and Soper [19] introduced the Multiple Label Method for large random networks, offering a solution for FSPP. To address the limitations of conventional non-interactive approaches, the concept of the degree of possibility was proposed by Okada [20] , representing arc lengths using fuzzy numbers. Nayeem et al. [18] considered networks with interval-number and triangular fuzzy numbers, providing an algorithm that accommodates both types of uncertain numbers.

Recognizing the computational complexity of FSPP, Hernandes et al. [14] presented an repetitive method utilizing a generic index ranking function to compare fuzzy numbers. This approach also accounted for graphs with negative parameters. Building on these methods, Kumar [17] tackled interval-valued fuzzy numbers within FSPP and introduced an algorithm capable of solving both fuzzy shortest path length and crisp shortest path length problems. A compartive study between the floyd warshall and rectangular algorithm under fuzzy environment is implemetnted by Vidhya et al. [24].

In a different direction, Baba [16] illustrated a technique for the Intuitionistic Fuzzy SPP. Mukherjee [21] implemented Dijkstra's algorithm for solving the shortest path with intuitionistic fuzzy arc weights in a graph. Subsequently, Broumi et al. [10] conducted a comprehensive comparative study of all existing FSPP approaches, ultimately identifying the most suitable methods for uncertain environments. Dijkstra's algorithm was expanded to address the Neutrosophic Fuzzy Shortest Path Problem (NFSPP) by Broumi et al., as documented in [22]. The arc weights are expressed as neutrosophic numbers in this extension. An interval-valued neutrosophic set (IVNFS) was introduced to expand the representation beyond single-valued neutrosophic sets (SVNS). An algorithm was devised to handle arc values of this type, as detailed in the work by Dey et al. [23] Numerous studies have been conducted on the Neutrosophic Fuzzy Shortest Path Problem (NFSPP), including works such as [26] to [31]. Janani et al. introduced the concept of bipolar neutrosophic refined sets in their work [29]. Additionally, Broumi et al. addressed the shortest path problem within the framework of interval-valued Fermatean fuzzy numbers in their study [32].

Cakir et al. proposed widening the Dijkstra algorithm in the context of the Bipolar Neutrosophic Fuzzy Shortest Path Problem (BNFSPP), as presented in [33]. This extension was exemplified with a practical example and pseudocode.

This study aims to extend the concept of the bellman-ford algorithm under bipolar neutrosophic numbers for the time-dependent SPP. The Bellman-Ford can detect the presence of negative cycles in a graph, which is a critical feature in applications where identifying and addressing negative processes, such as in-network routing, is essential to prevent instability. The

bipolar neutrosophic numbers (BNN) are used in optimization problems where the objective has both positive and negative aspects. It is applicable in various engineering and operational optimization contexts. The development of these methods has motivated us to explore the application of time-dependent SPP with bipolar neutrosophic arc values using the Bellman-Ford algorithm, which, interestingly, has not been previously employed in the context of BNFS.
This work offers several key contributions:

(i) In this paper, we incorporate Bipolar Neutrosophic values as arc values, enriching the problem domain.

(ii)Furthermore, we extend the application of time-dependent Bipolar Neutrosophic numbers (TD-BNN) to the Bellman-Ford algorithm, enhancing its versatility.

(iii)To illustrate the practicality of our approach, we present a numerical example where we successfully identify optimal results.

(iv) Additionally, we conduct a comparative analysis to demonstrate the superior efficiency of our proposed method when compared to existing approaches.

The remaining sections of the paper are structured as follows: Section 2 introduces the fundamental definitions and concepts of BNFS. In Section 3, we delve into the mathematical formulation of the problem of the SPP in a bipolar neutrosophic fuzzy context. Our preferred technique is illustrated in Section 4. A numerical example in Section 5 demonstrates how our approach can applied in practice. Section 6 compares our method with other existing methods and identifies its drawbacks. The paper concludes with Section 7.

2. Literature Review

This section introduces the fundamental concepts of the time-dependent shortest path problem and provides an overview of the relevant literature associated with this study.

Time-Dependent Shortest Path Problem

The time-dependent shortest path problem (TD-SPP) involves finding the most efficient route between two points in a network, considering variable travel times at different points in time. TD-SPP algorithms need to be adaptable to different contexts and modes of transportation. Researchers work on models that can effectively represent and predict time-dependent changes in travel times. Researchers work on models that can effectively describe and forecast time-dependent changes in travel times. The study on time-dependent shortest path problem, its theoretical aspects, and an algorithm is proposed by Dean and Brain [36]. Androutsopoulos offers a method for solving k-SPP with time-dependent cost attributes [40]. A time-dependent algorithm for vehicle routing problems is introduced by Rabie et al. [44]. Then Huang et al. [45] proposed a method for the time-dealy neural network scenario. After several studies, Hansknecht [47] proposed a dynamic approach for TD-SPP and executed it computationally to handle significant circumstances. The constrained reliable shortest path problem in stochastic time-dependent networks (CRSP-STD) extends the reliable, the time-dependent, and the constrained shortest path problem is implemented by Matthias [48].

Fuzzy Time-dependent Shortest Path Problem.

Fuzzy time-dependent models provide a means to optimize routes under uncertain conditions. Decision-makers can consider the variability and fuzziness in travel times to make more robust and reliable decisions.

Huang et al. [42] proposed the fuzzy programming model for the travel time represented by fuzzy sets and time-dependent. Then, Huang [43] prolonged it for mixed fuzzy numbers for time-dependent networks. The label correcting method for the TD-SPP is extended by Kolovsky et al. [39]. Then, Liao et al. [41] devised a genetic algorithm to solve fuzzy constrained shortest path problems. In many real-world situations, opinions, attitudes, or sentiments are positive and negative. Bipolar neutrosophic (BN) sets provide a framework to model and represent such bipolarity, capturing both positive and negative aspects of information. Real-world networks are dynamic, and the conditions of edges may change over time. The bipolar neutrosophic approach can be applied to model the uncertainty associated with dynamic environments where edge weights are subject to change. The concept of bipolar neutrosophic graphs(BNG), their properties, and their classes of single-valued BNG is introduced by Hassan et al. [38]. Broumi et al. [46] suggested an approach for Bipolar Neutrosophic SPP. The minimal spanning tree for Bipolar neutrosophic graphs is proposed by Reddy et al. [37]. Additionally, a time-dependent Dijkstra algorithm was introduced for BNFSPP by Cakir et al. in [7]. While there is a significant body of literature dedicated to the FSPP and TD-SPP, there has been comparatively limited research focused on the intersection of these domains, known as Time-Dependent Fuzzy Shortest Path Problem(TD-FSPP). The literature in this specific area is relatively sparse, indicating a gap in the exploration of solutions that simultaneously consider both FSPP constraints and time-dependent SPP. This points to a potential avenue for further investigation and development in the field of bipolar neutrosophic network optimization.

3. Preliminaries

This section introduces the basic definitions of Intuitionistic fuzzy sets, Neutrosophic fuzzy sets, and Bipolar Neutrosophic fuzzy sets. It also examines their main properties and discusses arithmetic operations related to these sets.

Definition 3.1. An IFS $\mathfrak{I}$ on the universe $\mathfrak{Q}$ is defined by: $\mathfrak{I} = \{\mathfrak{x}, \mathfrak{m}(\mathfrak{x}), \mathfrak{n}(\mathfrak{x}) \mid \mathfrak{x} \in \mathfrak{Q}\}$ where $\mathfrak{m} : \mathfrak{Q} \to [0, 1]$ and $\mathfrak{n} : \mathfrak{Q} \to [0, 1]$ represents the membership and non-membership of each

$\mathfrak{x} \in \mathfrak{Q}$, respectively. Such that $0 \leq \mathfrak{m}(\mathfrak{x}) + \mathfrak{n}(\mathfrak{x}) \leq 1$ for all $\mathfrak{x} \in \mathfrak{Q}$. Hesitation or indeterminacy part can be calculated as $\pi(\mathfrak{x}) = 1 - (\sigma(\mathfrak{x}) + \rho(\mathfrak{x})$. In some of the real-life scenarios IFS can't work when $\sigma(\mathfrak{x}) + \rho(\mathfrak{x}) > 1$ [?] named as PFS, which is also known as IFS type 2 by [3].

Definition 3.2. [6] Let $\mathfrak{X}$ be the universe of discourse. Then $\mathfrak{N} = \{\langle \mathfrak{x}, \mathfrak{T}_N(\mathfrak{x}), \mathfrak{I}_N(\mathfrak{x}), \mathfrak{F}_N(\mathfrak{x})\rangle : \mathfrak{x} \in \mathfrak{X}\}$ is defined as Neutrosophic Fuzzy Set (NFS), where the truth-membership function is represented as "$\mathfrak{T}_N : \mathfrak{X} \to]^-0, 1^+[$", an interdeterminacy-membership function "$\mathfrak{I}_N : \mathfrak{X} \to]^-0, 1^+[$" and the falsity-membership function "$\mathfrak{F}_N : \mathfrak{X} \to]^-0, 1^+[$". The sum of $\mathfrak{T}(\mathfrak{X})$, $\mathfrak{I}(\mathfrak{X})$, $\mathfrak{F}(\mathfrak{X})$ has no restrictions, So

$$0 \leq \sup\, \mathfrak{T}_N(\mathfrak{x}) + \sup\, \mathfrak{I}_N(\mathfrak{X}) + \sup\, \mathfrak{F}_N(\mathfrak{X}) \leq 3 \tag{1}$$

Definition 3.3. [6]

Let $\mathfrak{X}$ be a universe of discourse and $\mathfrak{N}$ be a Bipolar Neutrosophic Fuzzy Set (BNFS) in $\mathfrak{X}$. Then $\mathfrak{N}$ can be expressed as $\mathfrak{N} = \{\langle \mathfrak{x}, \mathfrak{T}^+_N(\mathfrak{x}), \mathfrak{I}^+_N(\mathfrak{x}), \mathfrak{F}^+_N(\mathfrak{x}), \mathfrak{T}^-_N(\mathfrak{x}), \mathfrak{I}^-_N(\mathfrak{x}), \mathfrak{F}^-_N)\rangle : \mathfrak{x} \in \mathfrak{X}\}$ Where $\mathfrak{T}^+, \mathfrak{I}^+, \mathfrak{F}^+ : \mathfrak{X} \to [0, 1]$ and $\mathfrak{T}^-, \mathfrak{I}^-, \mathfrak{F}^- : \mathfrak{X} \to [-1, 0]$. The positive-membership degrees $\mathfrak{T}^+(\mathfrak{x})$", "$\mathfrak{I}^+(\mathfrak{x})$, $\mathfrak{F}^+(\mathfrak{x})$ represent the degree of truth, indeterminacy, and falsity of the element $\mathfrak{x}$ in the BNFS $\mathfrak{N}$. The negative-membership degrees $\mathfrak{T}^-(\mathfrak{x})$, "$\mathfrak{I}^-(\mathfrak{x})$, $\mathfrak{F}^-(\mathfrak{x})$ measure the degree of truth, indeterminacy, and falsity of the element $\mathfrak{x}$ in the opposite characteristic sets related to the BNFS $\mathfrak{N}$.

Definition 3.4. [6] Let $\tilde{\mathfrak{N}}_1 = (\mathfrak{T}^+_1, \mathfrak{I}^+_1, \mathfrak{F}^+_1, \mathfrak{T}^-_1, \mathfrak{I}^-_1, \mathfrak{F}^-_1)$ and $\tilde{\mathfrak{N}}_2 = (\mathfrak{T}^+_2, \mathfrak{I}^+_2, \mathfrak{F}^+_2, \mathfrak{T}^-_2, \mathfrak{I}^-_2, \mathfrak{F}^-_2)$ be two BNFS. The operations of BNFS are :

$$\tilde{\mathfrak{N}}_1 \oplus \tilde{\mathfrak{N}}_2 = \langle \mathfrak{T}^+_1 + \mathfrak{T}^+_2 - \mathfrak{T}^+_1\mathfrak{T}^+_2, \mathfrak{I}^+_1\mathfrak{I}^+_2, \mathfrak{F}^+_1\mathfrak{F}^+_2, -\mathfrak{T}^-_1\mathfrak{T}^-_2, -(\mathfrak{I}^-_1 - \mathfrak{I}^-_2 - \mathfrak{I}^-_1\mathfrak{I}^-_2), -(\mathfrak{F}^-_1 - \mathfrak{F}^-_2 - \mathfrak{F}^-_1\mathfrak{F}^-_2)\rangle \tag{2}$$

$$\tilde{\mathfrak{N}}_1 \otimes \tilde{\mathfrak{N}}_2 = \langle \mathfrak{T}^+_1\mathfrak{T}^+_2, \mathfrak{I}^+_1 + \mathfrak{I}^+_2 - \mathfrak{I}^+_1\mathfrak{I}^+_2, -\mathfrak{F}^+_1 + \mathfrak{F}^+_2 - \mathfrak{F}^+_1\mathfrak{F}^+_2, -(\mathfrak{T}^-_1 - \mathfrak{T}^-_2 - \mathfrak{T}^-_1\mathfrak{T}^-_2) - \mathfrak{I}^-_1\mathfrak{I}^-_2, -\mathfrak{F}^-_1\mathfrak{F}^-_2\rangle \tag{3}$$

where $\lambda \geq 0$

Definition 3.5. [6] Let $\tilde{\mathfrak{N}}_1 = (\mathfrak{T}^+_1, \mathfrak{I}^+_1, \mathfrak{F}^+_1, \mathfrak{T}^-_1, \mathfrak{I}^-_1, \mathfrak{F}^-_1)$ be the BNFS. The score function "$S(\tilde{\mathfrak{N}}_1)$" and the accuracy function "$\S(\tilde{\mathfrak{N}}_1)$ of a BNFS are defined as follows:

$$S(\tilde{\mathfrak{N}}_1) = \frac{\mathfrak{T}^+_1 + 1 - \mathfrak{I}^+_1 + 1 - \mathfrak{F}^+_1 + 1 - \mathfrak{T}^-_1 - \mathfrak{I}^-_1 - \mathfrak{F}^-_1}{6} \tag{4}$$

$$A(\tilde{\mathfrak{N}}_1) = \mathfrak{T}^+_1 - \mathfrak{F}^+_1 + \mathfrak{T}^-_1 - \mathfrak{F}^+_1 \tag{5}$$

Definition 3.6. [6] Let $\tilde{\mathfrak{N}}_1 = (\mathfrak{T}_1^+, \mathfrak{I}_1^+, \mathfrak{F}_1^+, \mathfrak{T}_1^-, \mathfrak{I}_1^-, \mathfrak{F}_1^-)$ and $\tilde{\mathfrak{N}}_2 = (\mathfrak{T}_2^+, \mathfrak{I}_2^+, \mathfrak{F}_2^+, \mathfrak{T}_2^-, \mathfrak{I}_2^-, \mathfrak{F}_2^-)$ be two BNFS. The comparison of two BNFS is defined as follows:

- If $S(\tilde{\mathfrak{N}}_1) > S(\tilde{\mathfrak{N}}_2)$ then $\tilde{\mathfrak{N}}_1 \succ \tilde{\mathfrak{N}}_2$
- If $S(\tilde{\mathfrak{N}}_1) < S(\tilde{\mathfrak{N}}_2)$ then $\tilde{\mathfrak{N}}_1 \prec \tilde{\mathfrak{N}}_2$
- If $S(\tilde{\mathfrak{N}}_1) = S(\tilde{\mathfrak{N}}_2)$ then $\tilde{\mathfrak{N}}_1 = \tilde{\mathfrak{N}}_2$

(1) f $A(\tilde{\mathfrak{N}}_1) > A(\tilde{\mathfrak{N}}_2)$ then $\tilde{\mathfrak{N}}_1 \succ \tilde{\mathfrak{N}}_2$
(2) If $A(\tilde{\mathfrak{N}}_1) < A(\tilde{\mathfrak{N}}_2)$ then $\tilde{\mathfrak{N}}_1 \prec \tilde{\mathfrak{N}}_2$
(3) If $A(\tilde{\mathfrak{N}}_1) = A(\tilde{\mathfrak{N}}_2)$ then $\tilde{\mathfrak{N}}_1 = \tilde{\mathfrak{N}}_2$

Remark 3.7. [6] A bipolar fuzzy set is a special case of a BNFS, which is a more general concept.

4. Bipolar Neutrosophic Fuzzy Shortest Path Problem

In this section, we outline the mathematical formulation of the Bipolar Neutrosophic Fuzzy Shortest Path Problem (BNFSPP).

Consider a directed graph, denoted as $\mathfrak{G} = (\mathfrak{V}, \mathfrak{E})$, where $\mathfrak{V} = \mathfrak{s} = 1, 2, ..., \mathfrak{e} = \mathfrak{m}$ represents the set of vertices, and $\mathfrak{E} = (i, j) : i, j \in \mathfrak{V}, i \neq j$ represents the set of edges. In this representation, the ordered pairs (i, j) signify connections between distinct vertices within the graph, with both i and j belonging to the set of vertices $\mathfrak{V}$. It's important to note that in a connected network, there exists only one path from node i to node j, denoted as $\mathfrak{p}_{ij}$. This path comprises a sequence of arcs: $\mathfrak{p}_{ij} = (i, i_1), (i_1, i_2), ..., (i_k, j)$. Crucially, each arc starts at its source node and ends at its terminal/destination node.

The main objective is to find the best route from node $\mathfrak{S}$ (the origin/source) to node$\mathfrak{D}$ (the target/destination), considering various factors related to travel times that vary depending on the time. In the context of BNN, this parameter is represented as $\mathfrak{c}_{ij} = \langle \mathfrak{T}^+, \mathfrak{I}^+, \mathfrak{F}^+, \mathfrak{T}^-, \mathfrak{I}^-, \mathfrak{F}^- \rangle$. Here, the positive membership degrees $\mathfrak{T}^+$, $\mathfrak{I}^+$, and $\mathfrak{F}^+$ correspond to truth membership, indeterminate membership, and false membership, respectively. Similarly, the negative membership degrees $\mathfrak{T}^-$, $\mathfrak{I}^-$, and $\mathfrak{F}^-$ indicate truth membership, indeterminate membership, and false membership concerning the arc i-j. These memberships are associated with the shortest path in terms of travel time along the arc i-j. The values assigned to the arc under consideration represent the parameters associated with each edge from i to j. In this study, BNFN represent the imprecise parameters of the SPP. Consequently, the resulting problem is the BNFSPP. The mathematical model of the problem is can be expressed as:

$$\min\tilde{Z} = \sum_{i=1}^{m}\sum_{j=1}^{m} \tilde{\mathfrak{C}}^{P} x_{ij}$$
$$\text{s.t.} \sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = \begin{cases} 1 & i = 1 \\ 0 & i \neq 1, m \\ -1 & i = m \end{cases} \tag{6}$$
$$x_{ij} \geq 0, i, j = 1, 2....m$$

The arc (i, j) is in the path if and only if $x_{ij} = 1$, otherwise $x_{ij} = 0$. The set of all paths from node $\mathfrak{s}$ to node $\mathfrak{t}$ is denoted by$T_{\mathfrak{st}}$. The Bipolar Neutrosophic fuzzy travel time of the path from the node $\mathfrak{u}$ to node $\mathfrak{v}$ is $\tilde{\mathfrak{C}}^{P}_{ij}$.

5. **Proposed Algorithm**

The Bellman dynamic programming is used to determine the shortest path by the forward pass calculation. The Extended Bellman-Ford with time-dependent dynamic programming under Bipolar Neutrosophic fuzzy numbers is formulated as:

Step 1: Set the distance from the source vertex as (Departure time).

$$\text{Initialization Step: Set the Source node as } \mathfrak{t}(1) = \tilde{t}_s(\text{Departure time})$$
$$\text{Main Step: } \mathfrak{t}(\alpha) = \min_{\alpha<\beta}[\mathfrak{t}(\alpha) + \mathfrak{w}_{\alpha\beta}] \tag{7}$$

Here $\mathfrak{w}_{\alpha\beta}$ is the directed Bipolar neutrosophic fuzzy time with nodes, $\mathfrak{t}(\alpha)$ is the BNF time of the SP from $\mathfrak{S}$ to $\mathfrak{D}$. Figure 1 illustrates the flowchart of the proposed method.

5.1. *Step-by-Step Procedure*

Here is a step by step procedure for the proposed algorithm:

(1) Set the distance of the source vertex to its departure time $\tilde{t}_s$ and set the distance of all other vertices to infinity.
(2) Repeat the following steps for the number of vertices minus one times in the Bipolar Neutrosophic Graph (BNG).
(3) For each edge (α, β) in the BNG, calculate the minimum value using the score function: $\mathfrak{t}(\alpha) = \min_{\alpha<\beta}[\mathfrak{t}(\alpha) + \mathfrak{w}_{\alpha\beta}]$
(4) Relaxation:
for i from 1 to V-1: for each edge$(\alpha, \beta) in the graph : if distance[\alpha] + weight(\alpha, \beta) < distance[\beta] : distance[\beta] = distance[\alpha] + weight(\alpha, \beta)$

(5) Check for Negative Cycles:

After N-1 iterations, check for any negative cycles in the BNG. For each edge (α, β) in the BNG:

If the distance of α plus the weight of the edge is less than that of β, report an error indicating the presence of a negative cycle.

(6) Output:

If no negative cycles are found, return the distance and previous arrays as the output. The distance array contains the Bipolar Neutrosophic shortest path distance from the source to each vertex, and the previous array contains the predecessor of each vertex in the Bipolar Neutrosophic shortest path.

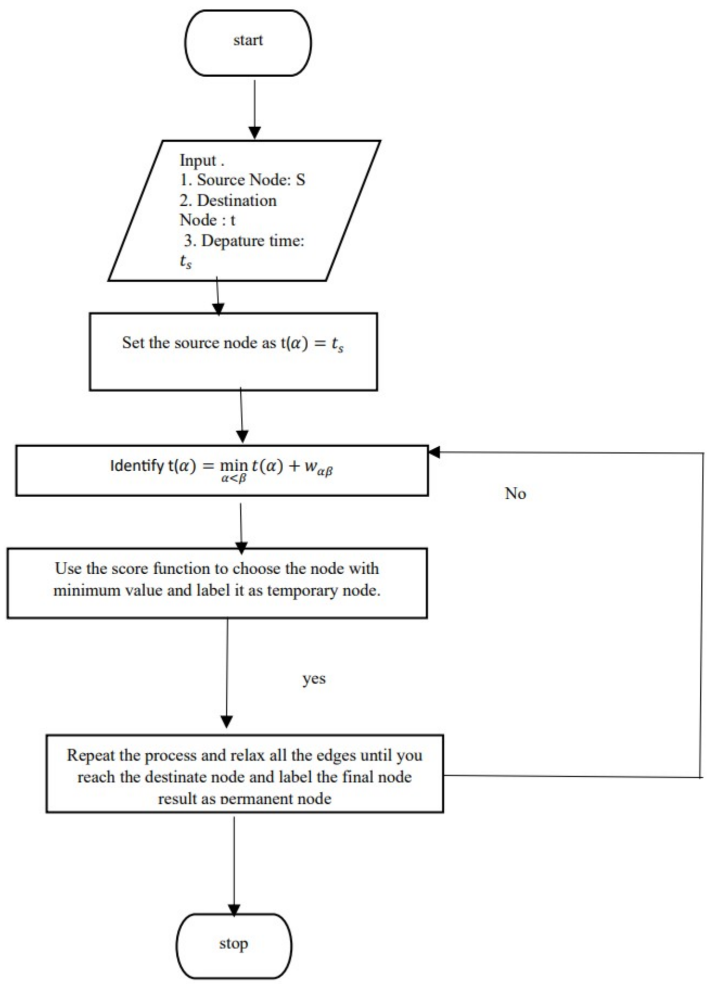

FIGURE 1. Flowchart for the proposed algorithm

5.2. *Improved Bellman-Ford Algorithm Pseudeocode*

Assume a graph G: with collections of vertices and edges. Let S be the source/origin vertex, and $\mathfrak{t}$ denote the distance array, which stores the shortest distances from S to all other vertices. Consider an array P to keep track of the predecessors of each vertex in the shortest path from S to that vertex. Fianlly , $\mathfrak{w}(U, V)$ denotes the weight of the edge connecting vertices U and V.

The pseudeocode for the proposed algorithm is given below in table 1:

Pesudeocode
function BellmanFord(G, S)
1. Initialize $\mathfrak{t}[S]$ *to* $\mathfrak{t}_s$ and $\mathfrak{t}[V]$ to infinity for all other vertices V in G
2. for each vertex V in G
3. if $V == S$ then
4. $\mathfrak{t}[V] = 0$
5. else
6.$\mathfrak{t}[V] =$ infinity
// Initialize P[V] to null for all vertices V in G for each vertex V in G
7. $P[V] = null$
8. Set a counter C to 0
9. $C = 0$
10. Repeat $\|V\| - 1$ times, where $\|V\|$ is the number of vertices in G while $C < \|V\| - 1$
11. For each edge (U, V) in G, check if $\mathfrak{t}[V]$ can be improved by using (U, V)
12. for each edge (U, V) in G
13. If $\mathfrak{t}[V] > \mathfrak{t}[U] + \mathfrak{w}(U, V)$ then update $\mathfrak{w}[V]$ and P[V]
14. if$\mathfrak{t}[V] > \mathfrak{t}[U] + \mathfrak{w}(U, V)$ then
15. $\mathfrak{t}[V] > \mathfrak{t}[U] + \mathfrak{w}(U, V)$
16. $P[V] = U$
17. Increment C by 1
18. $C = C + 1$
19. Check for negative cycles by relaxing the edges one more time for each edge (U, V) in G
20. If $\mathfrak{t}[V] > \mathfrak{t}[U] + \mathfrak{w}(U, V)$ then there is a negative weight cycle and shortest path does not exists
21. if $\mathfrak{t}[V] > \mathfrak{t}[U] + \mathfrak{w}(U, V)$ then
22. return "Negative cycle found"
23. Return the distance and parent arrays
24. return D, P

TABLE 1. Pesudecode for the Proposed Algorithm

6. **Numerical Example**

A numerical example from [7] involves a time-dependent network graph where BNFN denote the weights and the departure time $\mathfrak{t}_s = (0.2, 0.4, 0.5, -0.5, -0.7, -0.3)$

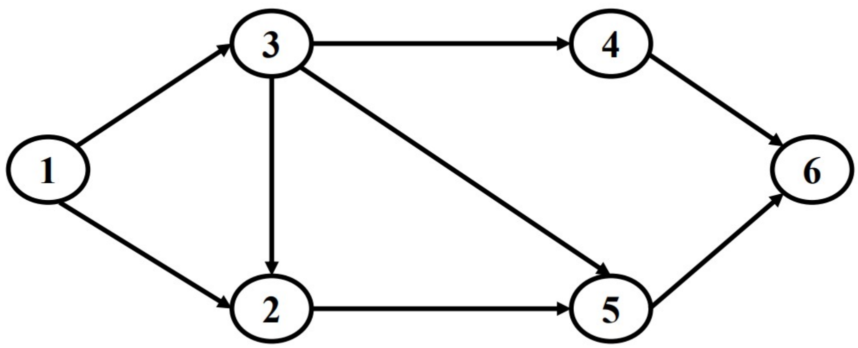

Figure 2. A Bipolar Neutrosophic Fuzzy Network Graph

Table 2. Arc values for Figure 2

Edges	**Time-dependent Bipolar Neutrosophic Fuzzy Arc Values**
$1 \to 2$	$(0.4, 0.6, 0.3, -0.5, -0.4, -0.3)$
$1 \to 3$	$(0.3, 0.8, 0.6, -0.7, -0.4, -0.2)$
$3 \to 2$	$(0.5, 0.3, 0.7, -0.4, -0.5, -0.4) - t$
$2 \to 5$	$(0.6, 0.8, 0.4, -0.7, -0.3, -0.2) * t$
$3 \to 4$	$(0.5, 0.3, 0.7, -0.4, -0.5, -0.1)$
$3 \to 5$	$(0.85, 0.3, 0.1, -0.2, -0.7, -0.8) + t$
$4 \to 6$	t
$5 \to 6$	$(0.7, 0.6, 0.2, -0.4, -0.4, -0.8)$

The best/optimal route from the $\mathfrak{S}$ to $\mathfrak{D}$ is described as follows using the suggested Algorithm 7.

Iteration 1: Begin with the source node. Assign the source node be t_s

$\mathfrak{t}(1) = (0.2, 0.4, 0.5, -0.5, -0.7, -0.3)$ and label the node as $\mathfrak{t}(1) = [(0.2, 0.4, 0.5, -0.5, -0.7, -0.3), 1]$

Iteration 2: Designate node 1 as α and node 2 as β. Proceed to ease the edges leading to node 2 employing the formula [7]. Utilize the scoring function [4] to select the minimum score and assign it as the temporary node.

$$\mathfrak{t}(2) = \min_{\alpha<2}\{\mathfrak{t}(\alpha) + \mathfrak{w}_{\alpha 2}\} = \mathfrak{t}(1) + \mathfrak{w}_{12} = (0.2, 0.4, 0.5, -0.5, -0.7, -0.3) + (0.4, 0.6, 0.3, -0.5, -0.4, -0.3)$$
$$= (0.52, 0.24, 0.15, -0.25, -0.82, -0.51)$$

$S(0.52, 0.24, 0.15, -0.25, -0.82, -0.51) = 0.615.$
Therefore, Label $\mathfrak{t}(2) = [(0.52, 0.24, 0.15, -0.25, -0.82, -0.51), 1 \to 2]$

Iteration 3: Iterate through the aforementioned procedure for node 3, relaxing all edges by applying the equation 7.

$$\mathfrak{t}(3) = \min_{\alpha<3}\{\mathfrak{t}(\alpha) + \mathfrak{w}_{\alpha 3}\} = \mathfrak{t}(1) + \mathfrak{w}_{13} = (0.2, 0.4, 0.5, -0.5, -0.7, -0.3) + (0.3, 0.8, 0.6, -0.7, -0.4, -0.2)$$
$$= (0.44, 0.8, 0.6, -0.7, -0.4, -0.2)$$

$S(0.44, 0.8, 0.6, -0.7, -0.4, -0.2) = 0.728.$
Therefore, Label $\mathfrak{t}(3) = [(0.44, 0.8, 0.6, -0.7, -0.4, -0.2), 1 \to 3]$

Iteration 4: Execute a similar sequence for node 4, wherein the edges reaching node 4 are relaxed using the equation denoted as [7]. Apply the score function identified as [4] to determine the minimum score. Subsequently, designate the node corresponding to this minimum score as the 'temporary node' for further analysis.

$$\mathfrak{t}(4) = \min_{\alpha<4}\{\mathfrak{t}(\alpha) + \mathfrak{w}_{\alpha 4}\} = \mathfrak{t}(3) + \mathfrak{w}_{34} = (0.44, 0.8, 0.6, -0.7, -0.4, -0.2) + (0.5, 0.3, 0.7, -0.4, -0.5, -0.1)$$
$$= (0.72, 0.096, 0.21, -0.14, -0.91, -0.49)$$

$S(0.72, 0.096, 0.21, -0.14, -0.91, -0.49) = 0.82.$
Therefore, Label $\mathfrak{t}(3) = [(0.72, 0.096, 0.21, -0.14, -0.91, -0.49), 1 \to 3 \to 4]$

Iteration 5: Apply the identical procedure to node 5, relaxing all edges towards node 4 with the equation [7]. Utilize the score function [4] to identify the minimum score, designating the node linked to this minimal score as the 'temporary node' for further analysis.

$$\mathfrak{t}(5) = \min_{\alpha<5}\{\mathfrak{t}(\alpha) + \mathfrak{w}_{\alpha 5}\}$$
$$= min\{\mathfrak{t}(3) + \mathfrak{w}_{35}, \mathfrak{t}(2) + \mathfrak{w}_{25}\}$$
$$= min\{(0.44, 0.8, 0.6, -0.7, -0.4, -0.2) + ((0.8, 0.3, 0.1, -0.2, -0.7, -0.8) + t),$$
$$(0.52, 0.24, 0.15, -0.25, -0.82, -0.51) + ((0.6, 0.8, 0.4, -0.7, -0.3, -0.2) * t)\}$$
$$= min\{(0.91, 0.038, 0.02, -0.04, -0.91, -0.92), (0.57, 0.21, 0.11, -0.21, -0.86, -0.54)\}$$

$S(0.91, 0.038, 0.02, -0.04, -0.91, -0.92) = 0.95.$
$S(0.57, 0.21, 0.11, -0.21, -0.86, -0.54) = 0.81.$

Therefore, the minimum values is choosen and Labeled $\mathfrak{t}(5) = [(0.57, 0.21, 0.11, -0.21, -0.86, -0.54), 1 \to 2 \to 5]$

Iteration 6: Ease all edges accessible from vertex 6 by employing the equation labeled as [7]. To determine the temporary node, apply the score function denoted as [4].

$$\begin{aligned}
\mathfrak{t}(6) &= \min_{\alpha<6}\{\mathfrak{t}(\alpha) + \mathfrak{w}_{\alpha 6}\} \\
&= min\{\mathfrak{t}(4) + \mathfrak{w}_{46}, \mathfrak{t}(5) + \mathfrak{w}_{56}\} \\
&= min\{(0.72, 0.096, 0.21, -0.14, -0.91, -0.49) + (0.2, 0.4, 0.5, -0.5, -0.7, -0.3), \\
&(0.57, 0.21, 0.11, -0.21, -0.86, -0.54 + (0.7, 0.6, 0.2, -0.4, -0.4, -0.8)\} \\
&= min\{(0.78, 0.039, 0.11, -0.07, -0.97, -0.64), (0.87, 0.13, 0.02, -0.08, -0.92, -0.91)\}
\end{aligned}$$

$S(0.78, 0.039, 0.11, -0.07, -0.97, -0.64) = 0.089.$

$S(0.57, 0.21, 0.11, -0.21, -0.86, -0.54) = 0.93.$

Therefore, the minimum values is choosen and Labeled $\mathfrak{t}(6) = [(0.78, 0.039, 0.11, -0.07, -0.97, -0.64), 1 \to 3 \to 4 \to 6]$

Hence the Optimal Path (node 1 to node 6) is $1 \to 3 \to 4 \to 6$) (Figure 3)along with travel time is (0.78,0.039,0.11,-0.07,-0.97,-0.64).

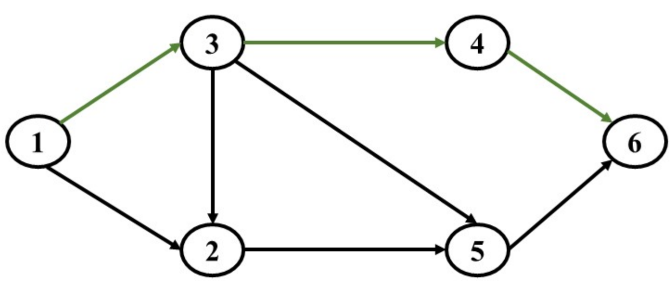

FIGURE 3. The shortest Path With Travel time using the proposed method

Table 3. Optimal Results of Shortest Travel Time for 2

Node	SP	BNFSP Travel Time	Score Value
2	$1 \rightarrow 2$	(0.52,0.24,0.15,-0.25,-0.82,-0.51)	0.615
3	$1 \rightarrow 3$	(0.44,0.8,0.6,-0.7,-0.4,-0.2)	0.738
4	$\rightarrow 3 \rightarrow 4$	(0.72,0.096,0.21,-0.14,-0.91,-0.49)	0.82
5	$1 \rightarrow 2 \rightarrow 5$	(0.57,0.21,0.11,-0.21,-0.86,-0.54)	0.81
6	$1 \rightarrow 3 \rightarrow 4 \rightarrow 6$	(0.78,0.039,0.11,-0.07,-0.97,-0.64)	0.089

7. **Sensitivity Analysis**

A sensitivity analysis was conducted to assess the performance of the proposed algorithm under various scenarios involving a time-dependent source node and variations in the arc values. The following cases were examined:

Case 1: Time-Dependent Source Node.
The algorithm was applied to a graph with a time-dependent source node.

Case 2: Zero Time-Dependent Value and Equal Arc Values
The time-dependent value was set to zero, and the arc values were kept constant as per the provided example.

Case 3: Zero Time-Dependent Value, Interchanged Arc Values.
The time-dependent value was set to zero, and the arc values (3,5) and (3,4) were interchanged.
Case 4: Time-Dependent Source Node with Interchanged Arc Values.
The source node had a time-dependent value, and the arc values (3,5) and (3,4) were interchanged.

Case 5: Time-Dependent Source Node with Interchanged Arc Value (1,2)
The source node had a time-dependent value, and the arc value (1,2) was considered in place of (2,5) and vice versa.

Case 6: Zero Time-Dependent Value with Interchanged Arc Value (1,2)
The time-dependent value was set to zero, and the arc value (1,2) was considered in place of

(2,5) and vice versa.

Table 4. Senstivity Analysis based on changing the values

Cases	SP	Bipolar Neutrosophic Number	Score Value
Case 1	$1 \to 3 \to 4 \to 6$	(0.78,0.039,0.11,-0.07,-0.97,-0.64)	0.089
Case 2	$1 \to 3 \to 4 \to 6$	(0.84,0.32,0.04,-0.17,-0.72,-0.87)	0.83
Case 3	$1 \to 3 \to 4 \to 6$	(0.91,0.04,0.02,-0.04,-0.91,-0.92)	0.79
Case 4	$1 \to 2 \to 5 \to 6$	(0.93,0.015,0.01,-0.02,-0.97,-0.94)	0.97
Case 5	$1 \to 3 \to 4 \to 6$	(0.78,0.039,0.11,-0.07,-0.97,-0.64)	0.79
Case 6	$1 \to 3 \to 4 \to 6$	(0.84,0.32,0.04,-0.17,-0.72,-0.87)	0.83

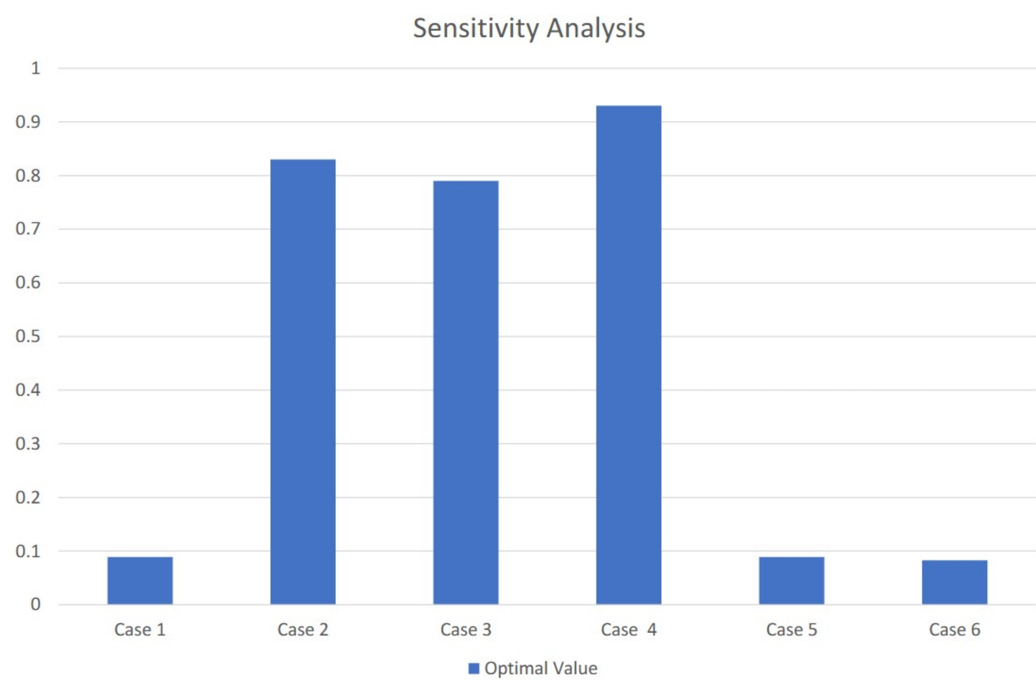

Figure 4. Senstivity Analysis for the example problem

Figure 4 illustrates the diagrammatic representation corresponding to various scenarios outlined in Table 4. The purpose of this sensitivity analysis is to assess the performance of the proposed algorithm under different conditions, encompassing variations in the time-dependent source node and alterations in arc values. Each case serves as a means to gain insights into the algorithm's robustness and adaptability across diverse input scenarios.

The outcomes derived from these individual cases play a pivotal role in comprehending the algorithm's behavior. They aid in identifying both the strengths and limitations of the algorithm when confronted with a spectrum of input configurations. This sensitivity analysis contributes valuable information for assessing the algorithm's efficacy and refining its application to different and challenging scenarios.

8. Comparison Analysis

The proposed algorithm consistently demonstrates superior performance compared to the method outlined in [7] and outperforms the existing approach presented by Broumi et al. in [46]. Notably, our method not only surpasses the performance of the cited methods but also provides accurate travel time scores, a feature absent in the existing methodology. Below table 5 shows the comparison result.

Table 5. Comparison for the proposed method

Methods	SP	Shortest Travel Time	Score of travel time
Time-Dependent Dijkstra Algorithm [7]	$1 \rightarrow 2 \rightarrow 5 \rightarrow 6$	(0.901,0.122,0.15,-0.078,-0.919,-0.912)	0.92
Exisitng Technique [46]	$1 \rightarrow 2 \rightarrow 5 \rightarrow 6$	(0.60,0.06,0.04,-0.024,-0.76,-0.85)	-
Proposed Method	$1 \rightarrow 3 \rightarrow 4 \rightarrow 6$	(0.78,0.039,0.11,-0.07,-0.97,-0.64)	0.089

The incorporation of an innovative approach for addressing uncertainty and variability in edge weights over time, utilizing time-dependent bipolar neutrosophic weights, presents a novel level of adaptability to dynamic environments. Utilizing bipolar neutrosophic numbers allows for the effective representation of imprecision and uncertainty in weight values, a crucial feature in dynamic environments where obtaining precise values may be challenging. A algorithm's capacity to manage uncertain weights in a time-dependent manner contributes to a more realistic modeling of real-world systems, accurately reflecting fluctuations in weights due to changing conditions. When the extended Bellman-Ford algorithm is modified to accommodate the time-dependent bipolar neutrosophic weights, it gains the capability to dynamically adapt to evolving network conditions. This adaptability proves advantageous in scenarios where edges undergo variations in travel times or costs owing to factors such as traffic conditions, weather, or other temporal influences. The proposed algorithm, featuring time-dependent bipolar neutrosophic weights, introduces a novel level of adaptability to dynamic environments. It exhibits the ability to dynamically respond to fluctuations in edge weights, making it well-suited for applications in which network conditions undergo frequent changes. The key advantage lies in its adaptability to changing environments, , particularly valuable in domains like transportation networks or communication networks where temporal variations can occur, and precise information may be elusive.

9. Applications of Bipolar Neutrosophic Time-Dependent Shortest Path Problem

Some applications of solving the Bipolar Neutrosophic Time-dependent Shortest Path Problem (BNTDSPP) using the Bellman-Ford algorithm:

(1) Transportation and Traffic Management:
By employing BNTDSP with the Bellman-Ford algorithm, transportation systems gain the ability to factor in unpredictable elements such as traffic uncertainty, accidents, and road closures. Additionally, the incorporation of time-dependent attributes allows for the consideration of varying traffic congestion levels at different times of the day.

(2) Network Routing in Telecommunications:
Telecommunication networks can optimize data transmission pathways by employing the Bellman-Ford algorithm adapted for BNTDSP. This optimization ensures the selection of the most dependable and time-efficient routes, considering variables like network congestion and the variable quality of network links.

(3) Autonomous Vehicles and Robotics:
Autonomous vehicles and robotic systems can enhance their navigational capabilities by incorporating BNTDSP within the Bellman-Ford framework. This adaptation enables these entities to navigate dynamic environments while accounting for obstacles that may arise unexpectedly, thus enabling real-time path adjustments for safe and efficient operations.

(4) Public Transportation Optimization:
Public transportation systems can streamline their operations by optimizing routes for buses, trams, or subways. This optimization encompasses the dynamic nature of passenger demand and varying traffic conditions, ultimately leading to more efficient and responsive public transportation networks.

The application of the Bellman-Ford algorithm adapted for BNTDSP in these scenarios empowers decision-makers to make well-informed choices by accommodating both bipolar neutrosophic elements and time-dependent attributes. This approach provides practical and adaptable solutions to real-world challenges across various domains.

9.1. *Benefits and Limitations of the proposed method*

Benefits

The proposed algorithm can handle graphs with both positive and negative weights, making it a versatile choice when the decision-maker uncertain about the weights in your graph.The optimal solution is guaranteed by the algorithm when it stops, regardless of the presence of negative weights, if there are no cycles with negative weight that can be reached from the source node. It can be elongated to

- Interval-Bipolar Neutrosophic numbers
- Fermatean Neutrosophic Fuzzy numbers
- interval-valued fermatean neutrosophic fuzzy number and so on.

Limitations

- The algorithm may require significant memory usage, especially for large graphs, which can be a constraint in memory-constrained environments.
- In dense graphs (graphs with a large number of edges), the prposed algorithm can be particularly slow due to its time complexity.
- BNN are capable of handling uncertainty, they may face challenges in effectively representing and manipulating temporal aspects or time-dependent variations in data, which are crucial in many real-world scenarios.

10. **Conclusion**

This paper introduces a novel methodology for determining optimal routes in a Bipolar Neutrosophic Graph, where travel times depend on the current time. In this unique scenario, vertex weights are expressed as fuzzy numbers to capture the inherent uncertainty and variability associated with travel times. To address this innovative problem, we extend the Bellman-Ford algorithm, originally designed for traditional graphs, to accommodate BNN. These specialized fuzzy numbers allow the representation of both positive and negative degrees of membership, thereby enhancing the algorithm's versatility. Our proposed method is applicable in transportation systems, where travel times are subject to uncertainty due to factors like traffic conditions, weather, or accidents. The BNSPP can also find applications in communication networks, supply chain optimization, critical infrastructure networks, financial networks, and more. The efficacy of our approach is evaluated through its application to a real-world case, with comparisons made against existing methods.

Notably, our method demonstrates computational efficiency, even when dealing with large graphs. However, its essential to acknowledge the limitations of our method, particularly in terms of the complexity in the representation of BNN. The three components—positive, neutral, and negative values—along with their respective membership functions, introduce intricacies in calculations and decision-making processes compared to simpler number systems. Additionally, while BNN adeptly handle uncertainty, challenges may arise in effectively representing and manipulating temporal aspects or time-dependent variations in data, crucial in many real-world scenarios. Looking ahead, we identify potential avenues for future research and applications in this promising direction, recognizing the need for further exploration and refinement.

Conflicts of Interest: The authors assert that they have no financial or affiliative interests that could present a conflict of interest regarding the submitted work.

References

1. Zadeh, L. A. (1965). Fuzzy sets. Information and control, 8(3), 338-353.
2. Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning-III. Information sciences, 9(1), 43-80. doi:10.1016/0020-0255(75)90017-1
3. Atanassov, K. T., & Atanassov, K. T. (1999). Intuitionistic fuzzy sets (pp. 1-137). Physica-Verlag HD.
4. Atanassov, K. T., & Atanassov, K. T. (1999). Interval valued intuitionistic fuzzy sets. Intuitionistic fuzzy sets: Theory and applications, 139-177.
5. Atanassov, K. T. (1989). Geometrical interpretation of the elements of the intuitionistic fuzzy objects. Preprint IM-MFAIS-1-89, Sofia, 20(S1), S27-S42. http://dx.doi.org/10.1016/S0165-0114(86)80034-3
6. Bosc, P., & Pivert, O. (2013). On a fuzzy bipolar relational algebra. Information Sciences, 219, 1-16.
7. Çakır, E., Ulukan, Z., & Acarman, T. (2022). Time-dependent Dijkstra's algorithm under bipolar neutrosophic fuzzy environment. Journal of Intelligent & Fuzzy Systems, 42(1), 227-236.
8. Smarandache, F. (1999). A unifying field in Logics: Neutrosophic Logic. In Philosophy (pp. 1-141). American Research Press.
9. Das, S., Roy, B. K., Kar, M. B., Kar, S., & Pamučar, D. (2020). Neutrosophic fuzzy set and its application in decision making. Journal of Ambient Intelligence and Humanized Computing, 11, 5017-5029.
10. Broumi, S., Deli, I., & Smarandache, F. (2014). Relations on interval valued neutrosophic soft sets. Journal of New results in Science, 3(5).
11. Ali, M., Son, L. H., Deli, I., & Tien, N. D. (2017). Bipolar neutrosophic soft sets and applications in decision making. Journal of Intelligent & Fuzzy Systems, 33(6), 4077-4087.
12. De, S. K., Biswas, R., & Roy, A. R. (2000). Some operations on intuitionistic fuzzy sets. Fuzzy sets and Systems, 114(3), 477-484.
13. Dubois, D. J. (1980). Fuzzy sets and systems: theory and applications (Vol. 144). Academic press.
14. Hernandes, F., Lamata, M. T., Verdegay, J. L., & Yamakami, A. (2007). The shortest path problem on networks with fuzzy parameters. Fuzzy sets and systems, 158(14), 1561-1570.
15. Klein, C. M. (1991). Fuzzy shortest paths. Fuzzy sets and systems, 39(1), 27-41.
16. Baba, L. (2013). Shortest path problem on intuitionistic fuzzy network. Annals of Pure and Applied Mathematics, 5(1), 26-36.
17. 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.
18. Nayeem, S. M. A., & Pal, M. (2005). Shortest path problem on a network with imprecise edge weight. Fuzzy Optimization and Decision Making, 4, 293-312.

19. Okada, S., & Soper, T. (2000). A shortest path problem on a network with fuzzy arc lengths. Fuzzy sets and systems, 109(1), 129-140.
20. Okada, S. (2004). Fuzzy shortest path problems incorporating interactivity among paths. Fuzzy Sets and Systems, 142(3), 335-357.
21. 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.
22. Broumi, S., Bakal, A., Talea, M., Smarandache, F., & Vladareanu, L. (2016, November). Applying Dijkstra algorithm for solving neutrosophic shortest path problem. In 2016 International conference on advanced mechatronic systems (ICAMechS) (pp. 412-416). IEEE.
23. Broumi, S., Dey, A., Talea, M., Bakali, A., Smarandache, F., Nagarajan, D., & Kumar, R. (2019). Shortest path problem using Bellman algorithm under neutrosophic environment. Complex & intelligent systems, 5, 409-416.
24. Kannan, Vidhya, Saraswathi Appasamy, and Ganesan Kandasamy. "Comparative study of fuzzy Floyd Warshall algorithm and the fuzzy rectangular algorithm to find the shortest path." AIP Conference Proceedings. Vol. 2516. No. 1. AIP Publishing, 2022.
25. Vidhya, K., & Saraswathi, A. (2023). An improved A* search algorithm for the shortest path under interval-valued Pythagorean fuzzy environment. Granular Computing, 8(2), 241-251.
26. Broumi, Said, et al. "Shortest path problem under triangular fuzzy neutrosophic information." 2016 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA). IEEE, 2016.
27. Broumi, S., Bakali, A., Talea, M., Smarandache, F., & Vladareanu, L. (2016, November). Computation of shortest path problem in a network with SV-trapezoidal neutrosophic numbers. In 2016 International Conference on Advanced Mechatronic Systems (ICAMechS) (pp. 417-422). IEEE.
28. 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.
29. Janani, R., & Shalini, A. F. (2023). An Introduction to Bipolar Pythagorean Refined Sets. Neutrosophic Systems with Applications, 8, 13-25.
30. Chakraborty, A. (2020). Application of pentagonal neutrosophic number in shortest path problem. International Journal of Neutrosophic Science, 3(1), 21-28.
31. Liu, R. (2020). Study on single-valued neutrosophic graph with application in shortest path problem. CAAI Transactions on Intelligence Technology, 5(4), 308-313.
32. Broumi, S., krishna Prabha, S., & Uluçay, V. (2023). Interval-Valued Fermatean Neutrosophic Shortest Path Problem via Score Function. Neutrosophic Systems with Applications, 11, 1-10.
33. ÇAKIR, E., & ULUKAN, H. Bipolar Neutrosophic Fuzzy Dijkstra Algorithm and Its Application.
34. Lee, K. M. (2000). Bipolar-valued fuzzy sets and their operations. In Proc. Int. Conf. on Intelligent Technologies, Bangkok, Thailand, 2000 (pp. 307-312).
35. Giri, P. (2021). A Study on m-polar Fuzzy Sets and m-polar Fuzzy Matrix. International Journal of Fuzzy Mathematical Archive, 19(1), 61-80.
36. Dean, B. C. (2004). Shortest paths in FIFO time-dependent networks: Theory and algorithms. Rapport technique, Massachusetts Institute of Technology, 13.
37. Reddy, G. U., Rao, T. S. N., Rao, V. V., & Rao, Y. S. (2020). Minimal Spanning tree Algorithms wrt Bipolar Neutrosophic Graphs. Infinite Study.
38. Hassan, A., & Malik, M. A. (2020). The Classes of bipolar single valued neutrosophic graphs.
39. Kolovský, F., Ježek, J., & Kolingerová, I. (2019). The ϵ-approximation of the time-dependent shortest path problem solution for all departure times. ISPRS International Journal of Geo-Information, 8(12), 538.

40. Androutsopoulos, K. N., & Zografos, K. G. (2008). Solving the k-shortest path problem with time windows in a time varying network. Operations Research Letters, 36(6), 692-695.
41. Liao, X., Wang, J., & Ma, L. (2021). An algorithmic approach for finding the fuzzy constrained shortest paths in a fuzzy graph. Complex & Intelligent Systems, 7, 17-27.
42. Huang, W., & Ding, L. (2012). The shortest path problem on a fuzzy time-dependent network. IEEE Transactions on Communications, 60(11), 3376-3385.
43. Huang, W., & Wang, J. (2016). The shortest path problem on a time-dependent network with mixed uncertainty of randomness and fuzziness. IEEE Transactions on Intelligent Transportation Systems, 17(11), 3194-3204.
44. Jaballah, R., Aerts-Veenstra, M., Callegari Coelho, L., & Renaud, J. (2021). The time-dependent shortest path and vehicle routing problem. Infor, 59(4), 592-622. https://doi.org/10.1080/03155986.2021.1973785
45. Huang, W., Yan, C., Wang, J., & Wang, W. (2017). A time-delay neural network for solving time-dependent shortest path problem. Neural Networks, 90, 21-28.
46. Broumi, S., Bakali, A., Talea, M., Smarandache, F., & Ali, M. (2017). Shortest path problem under bipolar neutrosphic setting. Applied Mechanics and Materials, 859, 59-66.
47. Hansknecht, C., Joormann, I., & Stiller, S. (2021). Dynamic shortest paths methods for the time-dependent TSP. Algorithms, 14(1), 21.
48. Ruβ, M., Gust, G., & Neumann, D. (2021). The constrained reliable shortest path problem in stochastic time-dependent networks. Operations Research, 69(3), 709-726.

Received: Dec 12, 2023. Accepted: Mar 5, 2024