

## Journal of Engineering Management and Systems Science

Journal homepage: [www.jemsci-journal.org](http://www.jemsci-journal.org)  
ISSN: 3123-1624



# Post-Flood Emergency Medical Supply Transportation Problem using Dijkstra's Algorithm inspired Single-Valued Neutrosophic Graphs

Balaji S M<sup>1,\*</sup>, Vishvesh Roshan<sup>2</sup>, Sujatha R<sup>3</sup>, Tetiana Zhyber<sup>4</sup>

<sup>1</sup> Department of Mathematics, St. Joseph's College of Engineering, OMR, Chennai, Tamil Nadu, India

<sup>2</sup> Department of Computer Science, St. Joseph's College of Engineering, OMR, Chennai, Tamil Nadu, India

<sup>3</sup> School of Science and Humanities (Mathematics), Shiv Nadar University, Kalavakkam, Chennai, Tamil Nadu, India

<sup>4</sup> Department of Finance, Kyiv National Economic University, Kyiv, Ukraine

### ARTICLE INFO

#### Article history:

Received 19 May 2026

Received in revised form 16 July 2026

Accepted 2 September 2026

Available online 7 September 2026

#### Keywords:

SVN-graph; Neutrosophic fuzzy number; Shortest path; Dijkstra's algorithm; Emergency medical supply

### ABSTRACT

In this paper, arc traversal costs are described as Single-Valued Neutrosophic (SVN) Fuzzy Numbers, which concurrently capture acceptance, rejection, and indeterminacy. This provides a useful framework for solving the Shortest-Path Problem (SPP) under extreme uncertainty. The research question is how to efficiently find an economical shortest path in an acyclic directed network with such imprecise weights without requiring prohibitively large computations. Our methodology ensures tractability and resilience by successively evaluating SVN edge lengths using a heuristic ranking function and a modified Dijkstra's algorithm. A real-world case study on the transportation of emergency medical supplies during a flood disaster and a numerical comparison with exhaustive enumeration are used for validation. The results show that the heuristic performs well under highly uncertain disaster conditions and consistently finds near-optimal paths while significantly reducing runtime. These results are important because they provide a deployable, computationally effective neutrosophic SPP solution for time-sensitive logistics, improving crisis response in situations where information is conflicting and incomplete.

## 1. Introduction

A graph is a practical tool for modeling real-world problems. By modeling the graph, nodes and arcs represent the objects and their relationships. Among the graph types we need to represent the different forms of information encountered in real-world problems are fuzzy graphs, intuitionistic fuzzy graphs, and neutrosophic graph theory [1, 2]. The idea of an IFS relationship was introduced by Shannon and Atanassov. Then, in [3], they presented numerous results and explored the idea of intuitionistic-fuzzy graphs. Certain operations on single valued neutrosophic graphs are examined by Akaram et al. [4].

In operation research, the SPP is a well-known network optimization problem. The decision maker in this challenge concentrates on computing the shortest path between a certain starting node and other nodes. Economics, telecommunications, scheduling transportation, supply chain management and routing are just a few of the real-world issues that the SPP has been thought to replicate. Numer-

\*Corresponding author.

E-mail address: [balajiphd2021@gmail.com](mailto:balajiphd2021@gmail.com)

<https://doi.org/10.67334/jemsci11202630>

© The Author(s) 2026 | [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

ous scholars have conducted extensive study on SPPs with predictable edge costs. A decision maker can effectively solve the conventional SPPs by utilizing a number of well-known algorithms that were developed by some outstanding scholars. Even while the costs of the arcs are regarded as real values in typical SPP, most real-life scenarios contain a number of variables that might not always be accurate (travel demands, travel expenses, trip capacity, travel time, etc.).

The arc costs in these situations are non-deterministic. Type-1 fuzzy numbers are used by certain researchers [2] to manage the uncertainties in regular fuzzy SPP (FSPP). Because type-1 fuzzy numbers' membership degree is just a real number, the FSPP is unable to handle the many kinds of uncertainties. Few researchers used intuitionistic FSPP to address this issue. In intuitionistic FSPP, arc lengths are classified as IFNs. It can handle confusing arc length data that includes both membership and non-membership grades at the same time. The fundamental disadvantage of intuitionistic FSPP is that if the total of membership and non-membership is more than one, it is unable to handle ambiguous arc length data.

The term neutrosophic SPP [5] appears in the field of graph theory, and an appropriate modeling technique may legitimately use the neutrosophic set in such real-life settings. In contrast to the conventional SPP, which solely employs crisp numbers (fuzzy numbers), the neutrosophic-SPP involves addition and comparison operations of neutrosophic sets. The fundamental goal of locating a path between two nodes that is shorter than all the other paths in a neutrosophic-SPP, where the the length of the arc is neutrosophic numbers, is complex since the ordering of neutrosophic sets as a comparison action can be interpreted in a variety of ways. The concept of fuzzy & intuitionistic-fuzzy sets (IFS) undoubtedly offers excellent strategies for dealing with ambiguity and missing data in a range of fields, including artificial-intelligence (AI), computer science and decision-making. The NS proposes an another mathematical strategy for dealing with uncertainty, partial information, and unpredictability. The NS framework divides events into the three-subsets: true ( $\mathcal{T}$ ), false ( $\mathcal{F}$ ) and indeterminate ( $\mathcal{I}$ ). Samanta S and Pal A [6, 7] pioneered the use of planar networks in fuzzy systems and further investigated their features. Broumi Said et al. [8] presented SVN-graphs, a new aspect of graph theory that has recently evolved as an extension of fuzzy-intuitionistic-FG. A neutrophilic network is formed when there is ambiguity in the connections between components in a problem.

In a weighted graph, Dijkstra's method is a traditional graph algorithm that determines the shortest path between a beginning node, also known as the source, and every subsequent node. One of the most researched topics in the field of fuzzy sets and systems is the shortest path problem where the prices have ambiguous values [9]. Dijkstra's algorithm is a graph search technique that creates a shortest path tree by solving the single source shortest path issue for a graph with non-negative edge path costs. Edsger Dijkstra, a Dutch computer scientist, created it in 1956 and introduced it in 1959 [10, 11]. Routing and other graph algorithms commonly use this technique as a subroutine. Many authors have researched and developed Dijkstra's approach for shortest path problems [12–14]. Roushan et al. [15] concentrated on maximizing disaster relief efforts in meteorological disasters, notably flash floods caused by cloudburst events.

Dijkstra investigated the shortest path problem in 1956 while he was employed as a programmer at the Mathematical Center in Amsterdam. His objective was to demonstrate the capabilities of the new ARMAC computer. His goal was to identify an issue and a computer-based solution that non-computer users could grasp. He developed the shortest path method and then used it with ARMAC for a considerably simplified transportation map of 64 Dutch cities. Some years later, hardware engineers working on the institute's next computer presented him with another challenge: minimize the number of wires needed to connect the pins on the machine's back panel. In an attempt to find a solution, he revisited Prim's minimal spanning tree algorithm, which Prim had discovered previously.

The simplest algorithm for resolving the SPP is frequently regarded as Dijkstra's algorithm. These approach is used to solve single-source shortest path issues involving undirected or directed paths.

The Dijkstra-approach is predicated on the following: there are no negative-edge costs, the total number of vertices is finite, the source is a single vertex, and the target can be any other vertex. The main idea behind the algorithm is as follows: it starts with the source, visits the vertices in ascending order of cost, and keeps track of a set  $V$  of visited vertices whose cost from the source has been determined, as well as a tentative cost  $D(u)$  for each unvisited vertex  $u$ .

Especially for acyclic networks where topological order could be utilized, previous work has not effectively pruned the search area in an SVN context by combining a heuristic ranking function with a modified version of a classical method (like Dijkstra's). The gap is the absence of a heuristic-driven, computationally efficient algorithm that can locate near-optimal paths in an acyclic SVN-weighted network quickly enough to be used in time-sensitive, life-saving disaster response operations while still retaining a reasonable level of accuracy in comparison to exhaustive search. This work fills this gap by presenting the first heuristic-driven, computationally efficient approach created especially for acyclic directed networks with SVN-weighted arcs. Our method does not assess every conceivable path, in contrast to exhaustive enumeration. Rather, it integrates a topologically modified Dijkstra's framework with a novel heuristic ranking function that enables quick pairwise comparisons of SVN numbers. This approach guarantees a near-optimal solution within stringent time limitations while scaling the algorithm to real-world network sizes.

Adhikari, D. [16] used a fuzzy AHP-TOPSIS-COPRAS framework in 2026 to assess women's empowerment in several Indian sectors, overcoming ranking uncertainty to find the best domains for fair and timely decision-making. A framework for making decisions was proposed by Simic, V. et al. [17] to support climate change analysis in urban transportation with the help of type-2 neutrosophic uncertainty. A hybrid Multi-Criteria Decision-Making (MCDM) technique is utilized to handle this issue. The MARCOS method is used to rank the difficulties that have been discovered, and the MEREC method is used to calculate the weights of assessment criteria. Using the Uncertain MCDM Methodology, Gazi, K.H. et al. [18] discovered Humanitarian Supply Chain Management Challenges. A mathematical technique for managing uncertainty is the triangle fuzzy number (TFN). Two decision-makers provide data sets in verbal terms, which are subsequently translated into precise numerical values. Momena et al. [19] carried out research to propose a seamless way to address the difficulties and flexibility of the medicinal supply chain (MSC). Different MSC problems are taken into account as criteria that address different adaptation strategies.

### 1.1 Motivation

The main goal of applying Dijkstra's algorithm to neutrosophic graphs is to determine the optimal path in situations when there is ambiguity, indeterminacy, and insufficient information. The neutrosophic variant handles edges represented by triples  $(T, I, F)$ -membership in contrast to classical Dijkstra, which makes the assumption of exact edge weights. Using score functions, the method seeks to decrease neutrosophic distance in order to find paths that optimize reliability while lowering risk and ambiguity. The underlying drawbacks of current methods serve as the inspiration. In real-world situations with ambiguous, contradictory, or missing data, classical Dijkstra fails. Although they can handle uncertainty, fuzzy graph-based approaches are unable to represent indeterminacy, which is the state in which information is neither true nor false but rather ambiguous or conflicting. By adding the indeterminacy component, neutrosophic graphs close this crucial gap and are perfect for dynamic, complicated systems.

This work is motivated by the critical shortcomings of intuitionistic and classical fuzzy shortest-path models in capturing the simultaneous rejection, acceptance, and indeterminacy found in disaster logistics, as well as the prohibitive computational cost of current neutrosophic approaches. Our motivation stems from the pressing need to bridge the gap between advanced neutrosophic theory and life-saving operational reality by giving first responders a deployable, computationally efficient algorithm that can deliver near-optimal routes under extreme uncertainty in a matter of seconds. This

research describes a modified version of Dijkstra's algorithm for calculating the weighted aggregated neutrosophic fuzzy value of the shortest or lowest-cost path in a neutrosophic fuzzy environment.

The shortest path can be easily rebuilt by moving backward from the terminal vertex to its predecessor, tracing the series of vertices that got permanent labels. Although several approaches have been proposed in the literature to solve the Shortest-Path Problem (SPP) with fuzzy arc lengths, Neutrosophic Fuzzy Numbers (NFNs), which generalize both fuzzy and intuitionistic fuzzy numbers by incorporating truth, falsity, and indeterminacy membership degrees, have yet to be applied to this problem. To the best of our knowledge, no other study has addressed the SPP using data presented in this format, emphasizing the fundamental contribution of our work. Furthermore, the Neutrosophic-SPP has never been solved using the unique approaches suggested here, nor have the specific properties used in our approach been applied in this context. These are the core innovations and implications of this work. Table 1 presents a comparative literature review, comparing the present work with related published studies. It highlights the research gap, demonstrates the novelty of the present work, and shows how the proposed research extends existing studies.

**Table 1**  
Comparative Analysis of Present Work with Existing Literature

| Present Work                                                                                  | Related Published Work                                                                     | Comparison & Contribution                                                                                                                                                           |
|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Single-Valued Neutrosophic graph with acyclic directed networks                               | Fuzzy graphs [1, 2], Intuitionistic Fuzzy Graphs [3], Neutrosophic graphs [4, 5, 8]        | Present work extends existing graph frameworks by specifically addressing acyclic directed networks with SVN-weighted arcs, whereas prior works focused on general graph structures |
| Modified Dijkstra's algorithm with novel ranking function for SVN-numbers                     | Classical Dijkstra's algorithm [10, 11, 20], Standard FSPP algorithms [9]                  | First algorithm combining heuristic ranking function with modified Dijkstra's for SVN acyclic networks                                                                              |
| Time-sensitive disaster response operations (particularly flash floods and cloudburst events) | General SPP applications [21]: Economics, telecommunications, transportation, supply chain | Specifically addresses life-saving disaster response operations with stringent time limitations, unlike general applications in prior works                                         |
| Real-time near-optimal pathfinding with acceptable accuracy                                   | Fuzzy AHP-TOPSIS-COPRAS [16], MEREC-MARCOS [17], TFN-based MCDM [18, 19]                   | Provides algorithmic solution for path optimization rather than general decision-making frameworks                                                                                  |

## 1.2 Novelty & Research Outline

The main innovation is modeling arc traversal costs using SVN-Fuzzy Numbers. This approach is particularly well-suited for extreme uncertainty (e.g., disaster scenarios where data is conflicting, incomplete, or unreliable) because it simultaneously captures three distinct dimensions of uncertainty—acceptance, rejection, and hesitancy, in contrast to conventional fuzzy or probabilistic methods. The SPP with SVN weights is said to have the first computationally deployable solution in this research. In order to avoid the exponential complexity of exhaustive enumeration, it introduces a heuristic ranking function that converts SVN numbers into clear, comparable values, enabling the implementation of a

modified Dijkstra's algorithm. The main Algorithm Design and Experimental Setup are as follows:

- A. Algorithm Design:** To enable quick path finding, the developed cost function is integrated into a modified Dijkstra's algorithm, so sacrificing marginal optimality for notable processing efficiency.
- B. Experimental Setup:** We deploy the algorithm to an emergency medical supply chain that has been damaged by severe flooding in order to evaluate its practical applicability. Simultaneously, a controlled numerical experiment quantifies accuracy trade-offs by comparing our heuristic outputs with those obtained from exhaustive enumeration.

### 1.3 Structure of the paper

The first section begins with an introduction that lays the groundwork for the rest of our work. We also analyze the existing literature, focusing on the development of Dijkstra's algorithm in particular. Section 2 give a clear and concise explanation of the key ideas behind Dijkstra's method, which leads to the basic concepts of neutrosophic graphs. Dijkstra's algorithm for neutrosophic graphs is presented in Section 3 along with a comparison study and an example. The real-world application of emergency medical supply delivery during a flood disaster is examined in Section 4. In Section 5, the authors provide their final conclusion.

Several symbols and their associated meanings are often used in this text. Table 2 below provides a summary of these symbols together with their explanations:

**Table 2**

List of symbols and abbreviations

| Symbols                            | Abbreviations                        |
|------------------------------------|--------------------------------------|
| SVN                                | Single-Valued Neutrosophic           |
| SPP                                | Shortest-Path Problem                |
| FSPP                               | Fuzzy Shortest-Path Problem          |
| IFSPP                              | Intuitionistic Shortest-Path Problem |
| NSPP                               | Neutrosophic Shortest-Path Problem   |
| IFS                                | Intuitionistic-fuzzy sets            |
| SVNN ( $\mathcal{N}_{\tilde{N}}$ ) | Single-Valued Neutrosophic Number    |
| $\mathcal{N}(G)$                   | Single-Valued Neutrosophic Graph     |
| $S(\mathcal{N})$                   | Score function                       |
| $H(\mathcal{N})$                   | Accuracy function                    |

## 2. Preliminary concepts

Neutrosophic Sets were described by F. Smarandache as an extension of fuzzy and intuitionistic-fuzzy sets. In the general theory, each of the three independent components  $\{\mathcal{T}, \mathcal{I}, \mathcal{F}\}$  belongs to  $]0^-, 1^+[$ . Wang et al. [22] introduced the single-valued neutrosophic set to operate with standard real numbers in  $[0, 1]$  for practical purposes.

**Definition 1** [23] A Single-Valued Neutrosophic Number (SVNN) is a  $\mathcal{N}_{\tilde{N}} = (T, I, F)$  representing an ill-known or uncertain quantity through three components  $\mathcal{T}$ ,  $\mathcal{I}$  and  $\mathcal{F}$  each a real number in  $[0, 1]$ . It extends fuzzy & intuitionistic-fuzzy sets to model uncertainty, inconsistency, and indeterminacy, with  $0 \leq \mathcal{T} + \mathcal{I} + \mathcal{F} \leq 3$ .

**Definition 2** [4, 24] A single-valued neutrosophic graph  $\mathcal{N}(\mathcal{G})$  containing the pair of vertex set and edge set  $(\mathcal{W}, \mathcal{X})$ , where  $\mathcal{W} = (T_{\mathcal{W}}, I_{\mathcal{W}}, F_{\mathcal{W}})$  is a SVN-set on the vertex set  $V$  and  $\mathcal{X} = (T_{\mathcal{X}}, I_{\mathcal{X}}, F_{\mathcal{X}})$  is an SVN-relation on the edge set  $E \subseteq V \times V$ . The membership values of vertices and edges satisfy the

following conditions

$$\begin{cases} \mathcal{T}_{\mathcal{X}}(st) \leq \min \{ \mathcal{T}_{\mathcal{W}}(s), \mathcal{T}_{\mathcal{W}}(t) \}, \\ \mathcal{I}_{\mathcal{X}}(st) \leq \min \{ \mathcal{I}_{\mathcal{W}}(s), \mathcal{I}_{\mathcal{W}}(t) \}, \\ \mathcal{F}_{\mathcal{X}}(st) \leq \max \{ \mathcal{F}_{\mathcal{W}}(s), \mathcal{F}_{\mathcal{W}}(t) \}; \end{cases} \quad (1)$$

for every  $s, t \in V$ .

**Definition 3** [5, 25] Assume that SVN-number  $\mathcal{N}_{\mathcal{N}} = \{ \mathcal{T}, \mathcal{I}, \mathcal{F} \}$  then

- A. The score function of  $S(\mathcal{N})$  is defined as follows  $S(\mathcal{N}_{\mathcal{N}}) = \frac{2+\mathcal{T}-\mathcal{I}-\mathcal{F}}{3}$ ,
- B. An Accuracy function  $H(\mathcal{N})$  is defined as follows  $H(\mathcal{N}_{\mathcal{N}}) = \mathcal{T} - \mathcal{F}$ .

In Crisp Dijkstra, edge weights are exact values, presuming complete information. Fuzzy Dijkstra goes one step farther by using membership degrees to convey uncertainty. We move to Neutrosophic when the information is not just spread out, but we also have indeterminacy, allowing the algorithm to prioritize pathways that are true and certain over ones that are unknown or likely blocked.

### 3. Dijkstra's algorithm on Single-valued Neutrosophic Graph

In a crisp graph, Dijkstra's algorithm is important because it provides the fundamental concept of shortest path computation using exact numerical weights. In crisp graphs, edge weights are precise and clearly defined, which helps in understanding the core logic of distance initialization, relaxation, and greedy selection. It builds a strong foundation for advanced models like fuzzy or neutrosophic graphs. Crisp Dijkstra is frequently utilized in real-world applications such as network-routing, GPS navigation, transportation planning, and computer networks. Learning it improves algorithmic thinking, optimization skills, and understanding of graph theory basics before handling uncertainty-based models.

We study Dijkstra's algorithm in neutrosophic graphs because it enables shortest or most reliable pathfinding in systems where truth, falsity, and indeterminacy coexist — reflecting real-world complexity better than crisp or even fuzzy models. In the real world, networks often involve imprecise, incomplete, or conflicting information — something fuzzy graphs can only partly handle. Neutrosophic graphs go further by explicitly including indeterminacy.

By extending the standard Dijkstra algorithm to work on neutrosophic graphs, Neutrosophic Dijkstra is a powerful path-finding algorithm. Each edge weight is represented by the triplet  $[T, I, F]$ , which stands for truth, indeterminacy, and falsehood degrees. These degrees are independent and range from 0 to 1.

Unlike crisp Dijkstra (exact numbers) or fuzzy Dijkstra (single membership dimension), Neutrosophic Dijkstra captures three independent aspects of uncertainty simultaneously: how true a path is, how much uncertainty exists about it, and how false it might be. This allows modeling real-world complexities like conflicting information ( $1 < T + F$ ), incomplete knowledge ( $1 > T + F$ ), and complete ignorance ( $I$  high). The algorithm modifies classical Dijkstra by replacing addition with neutrosophic aggregation operators (e.g., min for  $T$ , max for  $I$  and  $F$ ) and comparison with score functions. The output provides not just the shortest path, but a comprehensive uncertainty profile-enabling better decision-making in applications like autonomous vehicles, disaster response, military logistics, and any domain where information is imperfect, inconsistent, or incomplete.

We adapt Dijkstra-algorithm to neutrosophic graphs to determine the most reliable or optimal path when:

1. Some edges are uncertain.
2. Some information about connection strength is missing or indeterminate.
3. Multiple conflicting data sources exist.

We construct the algorithm by modifying neutrosophic Dijkstra's steps:

---

```
function_neutrosophic_dijkstra(Graph, source):
```

```
for v in Graph.vertices:
```

```
dist[v] =  $\infty$  # Initially all the distances are unknown
```

```
parent[v] = None
```

```
dist[source] = 0
```

```
PQ = priority_queue() # Priority Queue for inserting source
```

```
PQ.insert(0, source) # Inserting source node to Priority Queue
```

```
while PQ is not empty:
```

```
(d, u) = PQ.extract_min() # Finding the node with minimum distance from the priorityqueue
```

```
for each edge (u, v) with neutrosophic weight (T, I, F):
```

```
cost = (2 + T - F - I)/3 # Score function
```

```
if dist[v] > dist[u] + cost : # Check whether the current path is the shortest path
```

```
dist[v] = dist[u] + cost
```

```
parent[v] = u
```

```
PQ.decrease_key(v, dist[v])
```

```
return dist, parent
```

Time Complexity:  $O((V + E)\log V)$

```
Procedure_Neutrosophic_Dijkstra(Graph, source)
```

```
----- Initialization -----
```

```
for each vertex v in Graph.vertices do
```

```
dist[v]  $\rightarrow \infty$  // Initially all nodes are not reachable
```

```
parent[v]  $\leftarrow$  NIL // no parent yet end for
```

```
dist[source]  $\rightarrow$  0
```

```
PQ  $\rightarrow$  new PriorityQueue() // min-heap keyed by distance
```

```
PQ.insert(0, source)
```

```
----- Main loop -----
```

```
while PQ is not empty do
```

```
(d, u)  $\rightarrow$  PQ.extract_min() // node u with smallest dist
```

```
for each edge (u, v) in Graph.adj[u] do
```

```
(T, I, F)  $\leftarrow$  weight(u, v) // neutrosophic weight
```

```
// Convert neutrosophic triple into scalar cost
```

```
cost  $\rightarrow$  (2 + T - F - I)/3
```

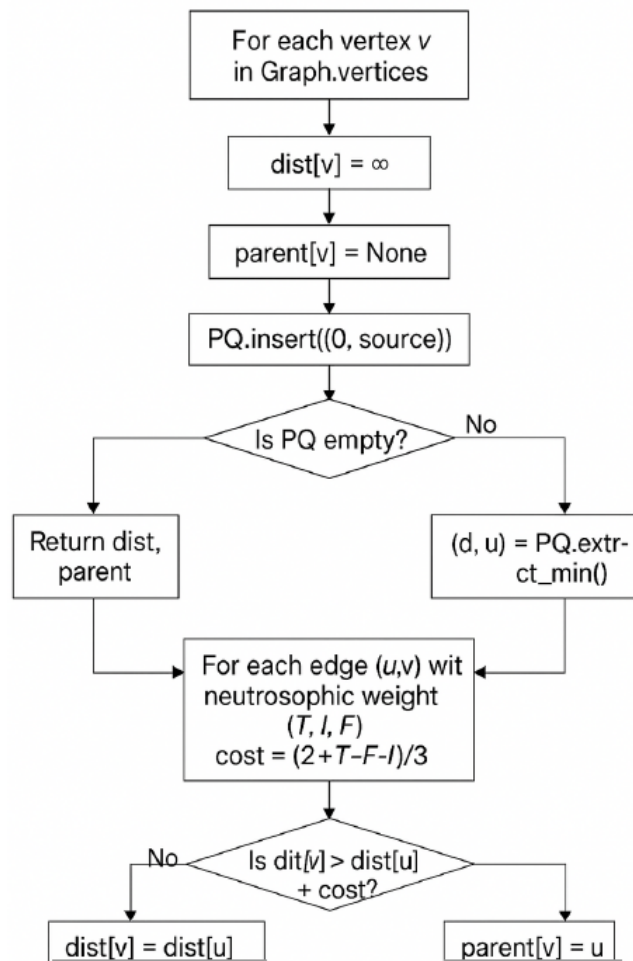
```
// Relax edge if shorter path is found if dist[v] > dist[u] + cost then
```

```
dist[v]  $\rightarrow$  dist[u] + cost
```

```
parent[v]  $\rightarrow$  u
```

---

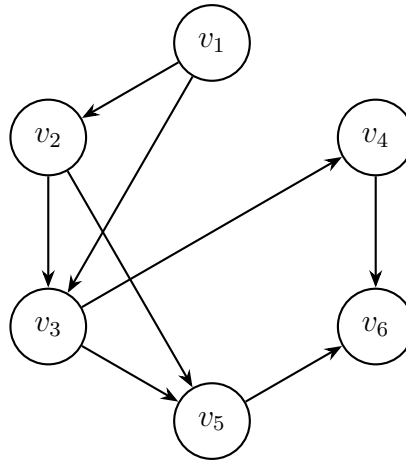
The flowchart in Figure 1 explains Dijkstra's algorithm for a neutrosophic graph. First, initialize each vertex with distance infinity and parent as None. Insert the source vertex into the priority queue with distance zero. While the priority queue is not empty, extract the vertex with minimum distance. For each adjacent edge  $(u, v)$  having neutrosophic weight  $(T, I, F)$ , compute a crisp cost using the formula  $\frac{(2+T-F-I)}{3}$ . Compare the new distance with the current distance of  $v$ . If the new value is smaller, update  $dist[v]$  and set  $parent[v] = u$ . Repeat until the queue becomes empty and return distances and parents.



**Fig. 1.** Flowchart of Dijkstra's algorithm

### 3.1 Illustrative Example

In a crisp graph, each edge has a single fixed weight. This assumes the data is exact and fully known. However, in real-world problems (transportation, medical diagnosis, communication networks), information may be uncertain. Thus, the shortest path reflects not only cost but also reliability and indeterminacy. Many practical systems involve ambiguous or partially known data. Classical Dijkstra cannot handle indeterminate information. Neutrosophic Dijkstra was motivated to incorporate uncertainty directly into path optimization, improving decision-making accuracy. Consider Figure 2 is an SVN-graph and Table 3 represents as the edge-membership value and its score value of the SVN-network in Figure 2, using the definition of the score of an edge. The aim is to compute the shortest distance and path between the source and destination nodes on the network.



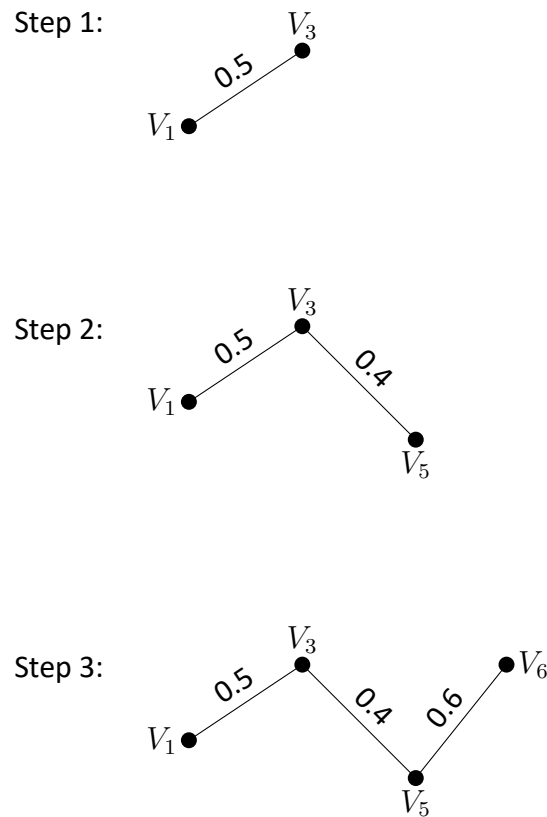
**Fig. 2.** SVN-network  $\mathring{G}_N$

**Table 3**

Score value of SVN-network  $\mathring{G}_N$

| Edges       | Edge Membership Value | Score of an Edge |
|-------------|-----------------------|------------------|
| $V_1 - V_2$ | (0.4, 0.6, 0)         | 0.6              |
| $V_1 - V_3$ | (0.2, 0.3, 0.4)       | 0.5              |
| $V_2 - V_3$ | (0.1, 0.4, 0.6)       | 0.36             |
| $V_2 - V_5$ | (0.7, 0.6, 0.8)       | 0.43             |
| $V_3 - V_4$ | (0.5, 0.3, 0.1)       | 0.7              |
| $V_3 - V_5$ | (0.3, 0.4, 0.7)       | 0.4              |
| $V_4 - V_6$ | (0.3, 0.2, 0.6)       | 0.5              |
| $V_5 - V_6$ | (0.6, 0.5, 0.3)       | 0.6              |

We use the Dijkstra method on the SVN network  $\mathring{G}_N$  to discover the shortest path between the networks, as shown below:



**Fig. 3.** Shortest path distance from  $V_1$  to  $V_6$  of Figure 2

Hence, in Figure 3, the shortest path distance from  $V_1$  to  $V_6$  using Dijkstra's algorithm is  $V_1 - V_3 - V_5 - V_6$  and the total distance is  $0.5 + 0.4 + 0.6 = 1.5$ .

### 3.2 Comparison analysis

The paper "Shortest Path Problem on SVN-graphs" [26] uses and shows SVN-networks. Said Broumi et al. [26] identified the shortest distance in the SVN network. Among others, Said Broumi used the neutrosophic shortest path algorithm. We used Dijkstra-algorithm to compute the shortest path of  $\mathring{G}_N$ , which is also part of the neutrosophic idea. Table 4 shows the score value generated for each edge.

**Table 4**

Comparison analysis

| Method                                    | Score of Shortest path distance |
|-------------------------------------------|---------------------------------|
| Neutrosophic shortest path algorithm [26] | $V_1 - V_3 - V_4 - V_6 = 1.7$   |
| Dijkstra's algorithm                      | $V_1 - V_3 - V_5 - V_6 = 1.5$   |

The comparison table shows that the neutrosophic shortest path algorithm selects the path  $V_1 - V_3 - V_4 - V_6$  with a score of 1.7, while classical Dijkstra's algorithm selects  $V_1 - V_3 - V_5 - V_6$  with a lower score of 1.5. This indicates that neutrosophic Dijkstra's algorithm is more efficient in selecting the best path under neutrosophic conditions. The neutrosophic approach evaluates truth, indeterminacy, and falsity components, providing a more realistic and reliable path under uncertainty. This highlights

the importance of the neutrosophic Dijkstra's algorithm in handling incomplete or ambiguous data. It improves decision-making accuracy and is especially useful in complex real-world networks where uncertainty plays a significant role. It is more powerful because it systematically applies a greedy strategy with proper ranking of  $(T, I, F)$  values, ensuring consistent comparison and faster convergence. It reduces computational complexity, avoids unnecessary path evaluations, and improves accuracy in decision-making. Therefore, neutrosophic Dijkstra's algorithm offers better optimization, efficiency, and reliability in uncertain network environments.

#### 4. Emergency Medical Supply Delivery During Flood Disaster Problem

##### 4.1 Problem definition

Studying the delivery of emergency medical supplies during flood incidents is crucial since the issue lies at the nexus of a potentially fatal logistical failure and a growing global threat. Supply channels are significantly disrupted by floods, and when medical supplies are delayed, avoidable deaths ensue. This complex reality can be modeled mathematically using a neutrosophic graph. This reasoning was applied to a flood scenario in my research work, which demonstrated how to compute transportation costs and penalties using neutrosophic values as opposed to fixed numbers. Decision-makers can determine the most effective routes that take into account the unpredictable and changing conditions, lowering cost and delivery time and guaranteeing a more dependable supply of essential products like food and medicine by solving the optimization issue on this graph.

##### 4.2 Data Sets

The research findings show that studies utilizing neutrosophic graphs to simulate disaster logistics frequently rely on two types of data: synthetic and literature-based data and estimates, and decision-maker inputs. The technique normally consists of two major steps:

1. Define the Network and Problem Structure.
2. Assigning Neutrosophic values to edges.

In the context of our flood research, our data collection process is most likely dependent on specialist knowledge. The unique  $(T, I, F)$  values assigned to each road are not derived from a single database. Instead, they are the result of a procedure in which a decision-maker or expert panel assesses each route's status (reliability, uncertainty, and hazard) using reports and knowledge of the flood's impact. This is precisely how models like yours use "partial reports, flooded roads, destroyed bridges, and unpredictable traffic" to calculate the best path.

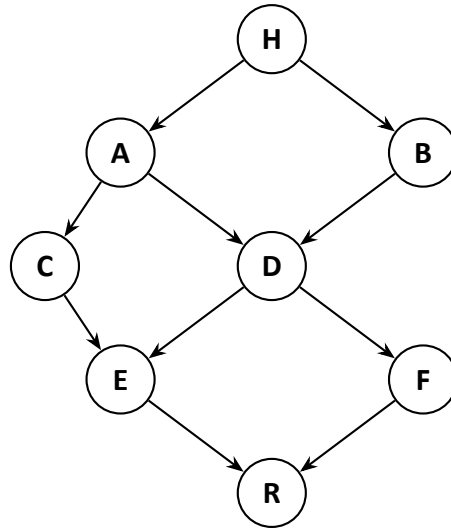
A neutrosophic fuzzy graph depicts the uncertain road conditions between Central Hospital (H) and Relief Camp (R). Each edge is classified as truth (passable), indeterminacy (unknown status), or falsity (impassable). The team calculates the best route, maximizing truth while minimizing indeterminacy and falsehood. This manages partial reports, flooded roads, destroyed bridges, and unpredictable traffic, guaranteeing that medicine is delivered reliably despite unclear and changing disaster data. Consider the eight vertices: H (Central Hospital), A (Junction-A), B (Junction-B), C (Junction-C), D (Junction-D), E (Junction-E), F (Junction-F), R (Relief Camp). Neutrosophic edge membership is represented as  $(T, I, F)$ , where  $T$  denotes road reliability,  $I$  represents uncertainty, and  $F$  indicates failure or blockage risk. The above emergency medical supply delivery during flood disaster problem is depicted in Figure 4 and its edge membership is shown in Table 5.

##### 4.3 Numerical Results

We apply Dijkstra's algorithm to emergency medical supply delivery during a flood disaster from the central hospital (H) to Relied camp (R) as follows: Initialization:

$\text{dist}[H] = 0$ , others  $= \infty$

From H:



**Fig. 4.** Emergency Medical Supply Delivery During Flood Disaster Problem

**Table 5**

Neutrosophic edge membership value and its score values

| Edge              | Neutrosophic Weight | Score value of edge |
|-------------------|---------------------|---------------------|
| $H \rightarrow A$ | (0.90, 0.05, 0.05)  | 0.93                |
| $H \rightarrow B$ | (0.70, 0.20, 0.10)  | 0.80                |
| $A \rightarrow C$ | (0.80, 0.10, 0.10)  | 0.87                |
| $A \rightarrow D$ | (0.60, 0.20, 0.20)  | 0.73                |
| $B \rightarrow D$ | (0.75, 0.15, 0.10)  | 0.83                |
| $C \rightarrow E$ | (0.85, 0.10, 0.05)  | 0.90                |
| $D \rightarrow E$ | (0.65, 0.20, 0.20)  | 0.75                |
| $D \rightarrow F$ | (0.70, 0.15, 0.15)  | 0.80                |
| $E \rightarrow R$ | (0.90, 0.05, 0.05)  | 0.93                |
| $F \rightarrow R$ | (0.60, 0.20, 0.30)  | 0.70                |

$$\begin{cases} H \rightarrow A : 0 + 0.9333 = 0.9333 \\ H \rightarrow B : 0 + 0.8000 = 0.8000 \end{cases}$$

$$dist = \{H : 0, A : 0.9333, B : 0.8000, others : \infty\}$$

Pick B (smallest : 0.8000)

From B:

$$B \rightarrow D : 0.8000 + 0.8333 = 1.6333 \text{ (better than } \infty \rightarrow \text{update D)}$$

$$dist[D] = 1.6333$$

Pick A (0.9333)

From A:

$$\begin{cases} A \rightarrow C : 0.9333 + 0.8667 = 1.8000 \end{cases}$$

$$\begin{cases} A \rightarrow D : 0.9333 + 0.7333 = 1.6666 \text{ (worse than } 1.6333 \rightarrow \text{no update)} \end{cases}$$

$$dist[C] = 1.8000$$

Pick D (1.6333)

From D:

$$\begin{cases} D \rightarrow E : 1.6333 + 0.7500 = 2.3833 \\ D \rightarrow F : 1.6333 + 0.8000 = 2.4333 \end{cases} \quad dist[E] = 2.3833, dist[F] = 2.4333$$

Pick C (1.8000)

From C:  $C \rightarrow E : 1.8000 + 0.9000 = 2.7000$  (worse than 2.3833  $\infty$  no update)

Pick E (2.3833)

From E:

$E \rightarrow R : 2.3833 + 0.9333 = 3.3166$

Pick F (2.4333), but  $F \rightarrow R : 2.4333 + 0.7000 = 3.1333$

$3.1333 < 3.3166 \rightarrow$  update R via F.

Shortest path

Path:  $H \rightarrow B \rightarrow D \rightarrow F \rightarrow R$

Total score distance = 3.1333.

#### 4.4 Discussion

- 1. Suggestions of Shortest Path:** Flooding changes frequently; thus use real-time sensor updates to refresh  $(T, I, F)$  data. Precalculate at least two backup pathways. Consider using a weighted score function to prioritize failure risk ( $F$ ) over reliability ( $T$ ) in emergency care delivery.
- 2. Merits of Shortest Path:** Reduces the risk of both blockage and compounded uncertainty. High-indeterminacy edges such as  $A \rightarrow D$  are avoided. effectively uses transformed crisp scores with Dijkstra.  $T \geq 0.6$  and  $F \leq 0.3$  for each edge guarantee that the route is free of critical failure points.
- 3. Demerits of Shortest Path:** longest topological path with four edges. Edge  $F \rightarrow R$  has the weakest link,  $F = 0.3$ . Static neutrosophic values disregard changes in floods in real time. Although the score function treats  $T$ ,  $I$ , and  $F$  equally, disaster response ought to penalize  $F$  more severely. No redundancy; the entire delivery is disrupted by a single edge failure.

In crisp graphs with exact edge weights, the traditional Dijkstra's algorithm determines the shortest path. On the other hand, uncertainty, indeterminacy, and inadequate information are present in real-world problems. Neutrosophic graphs explicitly model indeterminacy ( $I$ ), or circumstances in which information is unclear, contradictory, or ambiguous, in contrast to fuzzy graphs, which just record uncertainty. For making decisions in real time, this is essential. The algorithm can compare pathways under several competing criteria at once by minimizing neutrosophic distance using score or accuracy functions. The algorithm reduces risk in crucial applications by identifying paths with both high truth and low falsehood by taking the falsity component into account. As indeterminacy resolves, edge weights may vary over time, and the algorithm can effectively recalculate pathways. This ensures the ambulance follows the most reliable route rather than just the shortest one, saving lives by avoiding paths with high indeterminacy or falsehood, even if they are geographically shorter. By substituting neutrosophic triples  $(T, I, F)$  for crisp weights, the framework improves traditional Dijkstra and allows optimum pathfinding that directly represents truth, indeterminacy, and falsehood. This immediately enhances engineering performance in high-uncertainty emergency situations, decision quality, logistical dependability, and system resilience.

#### 5. Conclusion

The unique neutrosophic Dijkstra's algorithm was provided in this work, along with a comparative analysis and illustrative example showing its superiority over traditional techniques in uncertain contexts. In Section 4, a real-world case investigation involving the delivery of emergency medical supplies during a flood disaster was used to confirm the practical applicability of our strategy. In order to ensure that life-saving supplies reach afflicted areas via the most reliable routes rather than only the quickest ones, this application demonstrated how the algorithm efficiently prioritizes routes with high truth and low indeterminacy.

Our flood-disaster case study demonstrates how this approach can reduce route computation time from hours to seconds. For emergency medical dispatchers, this translates directly into actionable

intelligence, allowing them to dynamically reroute supplies as floodwaters rise, even when satellite data is inconsistent or delayed. In essence, we transformed an abstract mathematical model into a life-saving decision-making tool. Future study should expand this heuristic approach to general cyclic networks by including visited-node penalties or adaptive relaxation techniques, making it feasible for urban search-and-rescue operations.

The shortest path is not sufficient for real-world applications including medical logistics, autonomous driving, military navigation, and disaster response. Even if these routes are geographically longer, decision-makers require reliable paths that avoid large levels of falsehood and indeterminacy. An ambulance routing system, for example, must steer clear of highways with ambiguous obstacles or contradictory information. Therefore, the goal of neutrosophic Dijkstra is to improve safety, efficiency, and decision quality in mission-critical operations by providing robust, risk-aware, and adaptive pathfinding under real uncertainty. The author plans to use the Floyd-Warshall algorithm to investigate the shortest path problem in future work. The Floyd-Warshall approach is perfect for applications requiring all-pairs shortest path information because, in contrast to single-source algorithms like Dijkstra's, it calculates the shortest distances between each pair of vertices simultaneously.

### Acknowledgement

We are grateful to all the respected guides and teachers who helped us to complete this work.

### Conflicts of Interest

The authors declare no conflicts of interest.

### References

- [1] Ye, J. (2014). Single-valued neutrosophic minimum spanning tree and its clustering method. *Journal of intelligent Systems*, 23(3), 311–324. <https://doi.org/DOI10.1515/jisys-2013-0075>
- [2] Dey, A., Pal, A., & Pal, T. (2016). Interval type 2 fuzzy set in fuzzy shortest path problem. *Mathematics*, 4(4), 62. <https://doi.org/https://doi.org/10.3390/math4040062>
- [3] Shannon, A., & Atanassov, K. (2006). On a generalization of intuitionistic fuzzy graphs. *Notes on Intuitionistic Fuzzy Sets*, 12(1), 24–29. <http://ifigenia.org/wiki/issue:nifs/12/1/24-29>
- [4] Akram, M., & Shahzadi, G. (2017). Operations on single-valued neutrosophic graphs. *Journal of Uncertain Systems*, 11(3), 176–196. <https://jus.worldscientific.com/pdf/jusVol11No3paper02.pdf>
- [5] Broumi, S., Talea, M., Bakali, A., Smarandache, F., Nagarajan, D., Lathamaheswari, M., & Parimala, M. (2019). Shortest path problem in fuzzy, intuitionistic fuzzy and neutrosophic environment: An overview. *Complex & Intelligent Systems*, 5(4), 371–378. <https://doi.org/10.1007/s40747-019-0098-z>
- [6] Pal, A., Samanta, S., & Pal, M. (2013). Concept of fuzzy planar graphs. *2013 Science and Information Conference*, 557–563. <https://ieeexplore.ieee.org/document/6661793>
- [7] Samanta, S., Pal, M., & Pal, A. (2014). New concepts of fuzzy planar graph. *International Journal of Advanced Research in Artificial Intelligence*, 3(1), 52–59. <https://doi.org/10.14569/IJARAI.2014.030108>
- [8] Broumi, S., Smarandache, F., Talea, M., & Bakali, A. (2016). Single valued neutrosophic graphs: Degree, order and size. *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2444–2451. <https://doi.org/10.1109/FUZZ-IEEE.2016.7738000>
- [9] Hernandez, 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. <https://doi.org/10.1016/j.fss.2007.02.022>
- [10] Dijkstra, E. W. (2022). A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: His life, work, and legacy* (pp. 287–290). <https://doi.org/10.1145/3544585.3544600>
- [11] Frana, P. L., & Misa, T. J. (2010). An interview with Edsger W. Dijkstra. *Communications of the ACM*, 53(8), 41–47.
- [12] Boroujerdi, A., & Uhlmann, J. (1998). An efficient algorithm for computing least cost paths with turn constraints. *Information Processing Letters*, 67(6), 317–321. [https://doi.org/10.1016/S0020-0190\(98\)00134-3](https://doi.org/10.1016/S0020-0190(98)00134-3)
- [13] Cantone, D., & Faro, S. (2004). Two-levels-greedy: A generalization of Dijkstra's shortest path algorithm. *Electronic Notes in Discrete Mathematics*, 17, 81–86. <https://doi.org/doi:10.1016/j.endm.2004.03.019>
- [14] Knuth, D. E. (1977). A generalization of Dijkstra's algorithm. *Information Processing Letters*, 6(1), 1–5. [https://doi.org/10.1016/0020-0190\(77\)90002-3](https://doi.org/10.1016/0020-0190(77)90002-3)
- [15] Roushan, A., Das, A., Dutta, A., & Bera, U. K. (2025). A multi-objective supply chain model for disaster relief optimization using neutrosophic programming and blockchain-based smart contracts. *Supply Chain Analytics*, 10, 100107. <https://doi.org/10.1016/j.sca.2025.100107>
- [16] Adhikari, D., Gazi, K. H., Momena, A. F., Salahshour, S., Mandal, S., Giri, B. C., Dey, P., Ghosh, A., & Mondal, S. P. (2026). Sub-factors based decision making with uncertainty for sustainable women empowerment. *Soft Computing*, 30, 4497–4542. <https://doi.org/10.1007/s00500-026-11370-3>
- [17] Simic, V., Gokasar, I., Deveci, M., & Švadlenka, L. (2022). Mitigating climate change effects of urban transportation

- using a type-2 neutrosophic MEREC-MARCOS model. *IEEE Transactions on Engineering Management*, 71, 3233-3249. <https://doi.org/10.1109/TEM.2022.3207375>
- [18] Gazi, K. H., Biswas, A., Basuri, T., Ghosh, A., & Mondal, S. P. (2025). Finding humanitarian supply chain management challenges using uncertain MCDM methodology. *Spectrum of Mechanical Engineering and Operational Research*, 2(1), 248–279. <https://doi.org/10.31181/smeor21202548>
- [19] Momena, A. F., Gazi, K. H., & Mondal, S. P. (2025). Multi-criteria decision analysis for sustainable medicinal supply chain problems with adaptability and challenges issues. *Logistics*, 9(1), 31. <https://doi.org/10.3390/logistics9010031>
- [20] Misra, J. (2001). A walk over the shortest path: Dijkstra's algorithm viewed as fixed-point computation. *Information Processing Letters*, 77(2-4), 197–200. [https://doi.org/10.1016/S0020-0190\(00\)00202-7](https://doi.org/10.1016/S0020-0190(00)00202-7)
- [21] Demetrescu, C., Goldberg, A. V., & Johnson, D. S. (Eds.). (2009). The shortest path problem: Ninth DIMACS implementation challenge (Vol. 74). *American Mathematical Soc.* <https://archive.dimacs.rutgers.edu/Volumes/Vol74.html>
- [22] Wang, X., & Yang, L. (2025). Probabilistic single-valued neutrosophic set for a multidimensional evaluation of college students' mental health and academic success. *Neutrosophic Sets and Systems*, 91(1), 64. [https://digitalrepository.unm.edu/nss\\_journal/vol91/iss1/64](https://digitalrepository.unm.edu/nss_journal/vol91/iss1/64)
- [23] Akram, M. (2016). Single-valued neutrosophic planar graphs. *International Journal of Algebra and Statistics*, 5(2), 157–167. <https://doi.org/10.20454/ijas.2016.1207>
- [24] Balaji, S. M., Meiyappan, D., & Sujatha, R. (2024). Analyzing single-valued neutrosophic fuzzy graphs through matroid perspectives. *Ain Shams Engineering Journal*, 15(12), 103133. <https://doi.org/10.1016/j.asej.2024.103133>
- [25] Balaji, S. M., Meiyappan, D., Sujatha, R., & Kaviyarasu, M. (2026). Exploring relationship between neutrosophic topological indices and enhancing chemical reaction pathway optimization through f-index, edge f-index in neutrosophic graphs. *Journal of Computational and Applied Mathematics*, 117571. <https://doi.org/10.1016/j.cam.2026.117571>
- [26] Broumi, S., Talea, M., Bakali, A., Smarandache, F., & Kumar, P. K. (2017). Shortest path problem on single valued neutrosophic graphs. *2017 International Symposium on Networks, Computers and Communications (ISNCC)*, 1–6. <https://doi.org/10.1109/ISNCC.2017.8071993>