Exploring Critical Path Solving Methods under Neutrosophic

M. Navya Pratyusha[1], **Ranjan Kumar**[1,*]
[1]VIT-AP University, Inavolu, Beside AP Secretariat, Amaravati AP, India
Emails: navyapratyusha.miriyala@gmail.com; ranjank.nit52@gmail.com

Abstract

Over the past few decades, the traditional critical path method and its various generalizations have become the most popular technique for managing complex projects. It plays a crucial role in differentiating between critical and non-critical tasks to enhance project schedules. For the first time in the literature, our proposed model implements two algorithms for the study of the critical path method, each addressing an advanced framework in the form of a single-valued triangular neutrosophic. The proposed algorithm 1 utilizes Python to extended Dijkstra's algorithm under the neutrosophic framework, while the proposed algorithm 2 employs linear programming for optimality checks, which is solved using LINGO. Our comparison with previous research on the critical path method shows that the proposed algorithms are better at dealing with uncertainty, making project schedules more reliable and flexible. The findings lead to the proposed algorithm framework, combined with Python and LINGO, to enhance decision-making and improve the accuracy and efficiency of critical path identification in complex project environments.

Keywords: Critical Path Method; Uncertainty; Neutrosophic Set; Neutrosophic Dijkstra's Algorithm; Single Valued Triangular Neutrosophic

1 Introduction

Zadeh[1] established the theory of uncertainty, a valuable tool for handling imprecise data in various real-life situations. Practically, there are specific scenarios in which often encounter the observation that objectify similar minimum, maximum, and best possible approaches to current difficulties. However, most of today's data hold uncertain, imprecise, and incomplete information, which reflects an inconsistent outcome. This led to the development of uncertainty theory. Researchers in many areas of network problems have become very interested in fuzzy optimization and network study over the last few decades. Examples include the Critical Path Method (CPM) using acceptable probability,[2] the Project Evaluation and Review Technique (PERT) using the fuzzy Delphi technique,[3] the multi-objective minimum cost flow problem,[4] and the genetic algorithm-based problem of finding the shortest path.[5]

Network optimization[6,7] plays a large part in combinatorial optimization. In today's competitive environment, an increasing number of tasks in the network could effectively track progress while adhering to cost, time, and overall duration performance. Managing complex tasks can often pose challenges. While the project durations are recognized to be known and deterministic, the CPM and PERT have been confirmed to be useful tools in managing complex tasks. Over the years, several approaches have been proposed for finding the fuzzy critical path. Chanas and Kamburowski[8] proposed the first method, known as FPERT. Since then, many authors have tested the CPM problem in various fuzzy environments, such as type-2 fuzzy,[9] intuitionistic fuzzy,[10,11] Pythagorean fuzzy,[12,13] and interval type-2 fuzzy.[14] Different approaches have been suggested to solve different types of problems, such as the CPM ranking,[15] the multicriteria decision-making problem,[16] the identifying critical path by using the integrated fuzzy analytic hierarchy process (AHP) and TOPSIS methods, the linear

DOI: https://doi.org/10.54216/IJNS.250430

programming problem,[17] the Dijkstra algorithm,[18] and the time cost trade-off.[19] Handling uncertainty parameters in fuzzy, the research has fewer boundaries, where the fuzzy takes only one membership degree. To overcome this disadvantage, the extension principle offers a variety of uncertainty parameters, as shown in figure 1.

Smarandache[20] introduced a novel theory of neutrosophic sets (NS) which is a generalization of both fuzzy sets[1] and intuitionistic sets.[21] The neutrosophic set model faces uncertain, indeterminate, and contradictory data, with a well-defined qualification of the degree of indeterminacy. It helps to overcome several difficulties of the existing methods for modelling uncertain decision information. Many researchers have been interested in using the NS model to solve the CPM in the set of criteria mentioned, which are depicted in table 1.

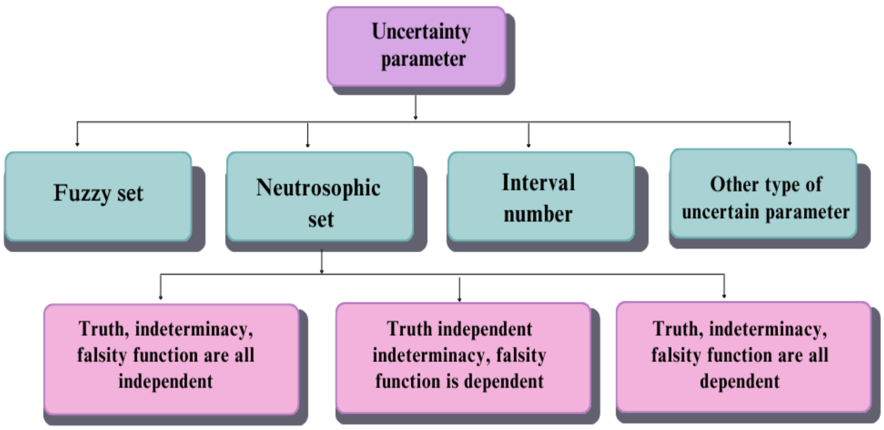


Figure 1: Classification of uncertainty parameters in neutrosophic theory

Table 1: Significant influences of the different authors towards CPM under various environments

Authors	Year	Significant Influences
Chakraborty et al.[22]	2020	Solving cylindrical neutrosophic single-valued numbers in different problems like critical path method, multi-criteria decision making, and spanning tree problem.
Chakraborty et al.[23]	2020	Developing a score function; and solving the network problem in pentagonal neutrosophic numbers.
Hema and Rajeshwari[24]	2023	Solving network models under an octagonal neutrosophic environment to find the critical path and minimum spanning tree.
Sinika and Ramesh[25]	2024	Solving the PERT problem using trapezoidal neutrosophic and inter-valued neutrosophic by employing a novel interval de-neutrosophication technique.
Abdel-Basset et al.[26]	2024	Implementing and using the modeling parameter trapezoidal neutrosophic number with the time-cost trade-off to reduce the time and the cost of the project.

Based on the previous discussions on CPM and currently available literature, there are fewer approaches available to solve CPM under neutrosophic theory. For the first time, we employ uncertainty parameter modelling of a single-valued triangular neutrosophic number in our work to solve the extended Dijkstra's algorithm and concurrently verify its optimality through linear programming. The paper offers the following contributions:

- Develop an algorithm that integrates neutrosophic values (truth, indeterminacy, and falsity) into the extended Dijkstra's algorithm, addressing a new set of problems beyond those found in the existing literature.

- Implementing the linear programming model to find the optimality check.
- A conduction to a comparative analysis between the existing literature and our proposed methods for solving CPM. This comparison not only demonstrates the efficacy of our approach but also prioritizes the role of computational techniques.
- The superior performance of our algorithms in handling complex, uncertain scenarios underscores the critical importance of advanced computational tools, i.e., Python and LINGO, in modern problem-solving.

Further sections of the paper are structured as follows: Section 2 discusses preliminary formulated ideas, Section 3 proposes implementation of extended Dijkstra algorithm under proposed algorithm 1, and optimality check of linear programming model under proposed algorithm 2. Section 4 verifies the CPM technique in a project scenario numerical study by using the proposed algorithms, followed by the conclusion with a list of references.

2 Preliminary

Definition 2.1. "Fuzzy set":[27]
Let ϖ be set of universal and let $\widehat{fs}$ be a fuzzy subset of ϖ. For each $\tau \in \varpi$, there exists a number $\mu_{\widehat{fs}} \in [0, 1]$ provided to represent the degree of membership of τ in $\widehat{fs}$.

Definition 2.2. "Fuzzy Number":[27]
A fuzzy number $\widehat{f_n}$ is the fuzzy set that satisfies both the normality and convexity conditions within the convex fuzzy set $\widetilde{C}$.

i. **Normality condition:** There exists at least one $\tau \in \mathbb{R}$ such that the membership function reaches its maximum value of 1. Formally, this can be expressed as :

$$\tau \in \mathbb{R} \text{ such that } \bigvee_{\tau} \mu_{\widehat{f_n}} = 1$$

ii. **Convexity condition:** For any two elements $\tau_1, \tau_2 \in \widetilde{C}$ and for all $\lambda \in [0,1]$, the membership function of the convex combination of τ_1 and τ_2 is at least the minimum of their individual membership degrees. Formally, this can be expressed as :

$$\forall \tau_1, \tau_2 \in \widetilde{C},\ \forall \lambda \in [0,1],\ \mu_{\widehat{f_n}}(\lambda\tau_1 + (1-\lambda)\tau_2) \geq \min(\mu_{\widehat{f_n}}(\tau_1), \mu_{\widehat{f_n}}(\tau_2))$$

Definition 2.3. "Single-valued Neutrosophic set (SVNS)"[28]:
A SVNS is defined as $S\dot{V}NS = (\langle \tau; \varsigma_n(\tau), \varphi_n(\tau), \gamma_n(\tau) \rangle, \tau \in \varpi)$, where ϖ is the universal set, and the membership degrees satisfy the following conditions as $\varsigma_n(\tau), \varphi_n(\tau), \gamma_n(\tau) \in [0, 1]$. Moreover, the sum of these membership degrees must satisfy:

$$0 \leq \varsigma_n(\tau) + \varphi_n(\tau) + \gamma_n(\tau) \leq 3 \text{ for all } \tau \in \varpi$$

Definition 2.4. "Single-Valued Triangular Neutrosophic Number (SVTrN)"[29]:
A SVTrN is defined as $\widetilde{SV} = \langle (SV_1, SV_2, SV_3; \varsigma_{(\tau)}, \varphi_{(\tau)}, \gamma_{(\tau)}) \rangle$ on real number $\mathbb{R}$, where truth, indeterminacy, and falsity membership functions are given as follows:

$$T(\varsigma_{(\tau)}) = \begin{cases} \dfrac{(\tau - SV_1)\,\varsigma_{(\tau)}}{SV_2 - SV_1} & SV_1 \leq \tau \leq SV_2 \\ \varsigma_{(\tau)} & \tau = SV_2 \\ \dfrac{(SV_3 - \tau)\,\varsigma_{(\tau)}}{SV_3 - SV_2} & SV_2 < \tau \leq SV_3 \\ 0 & \text{Otherwise,} \end{cases}$$

$$I\left(\varphi_{(\tau)}\right)=\begin{cases}\dfrac{(SV_2-\tau+(\tau-SV_1)\varphi_{(\tau)})}{SV_2-SV_1} & SV_1\leq\tau\leq SV_2\\ \varphi_{(\tau)} & \tau=SV_2\\ \dfrac{\left(\tau-SV_2+(SV_3-\tau)\,\varphi_{(\tau)}\right)}{SV_3-SV_2} & SV_2<\tau\leq SV_3\\ 1 & \text{Otherwise,}\end{cases}$$

$$F\left(\gamma_{(\tau)}\right)=\begin{cases}\dfrac{(SV_2-\tau+(\tau-SV_1)\gamma_{(\tau)})}{SV_2-SV_1} & SV_1\leq\tau\leq SV_2\\ \gamma_{(\tau)} & \tau=SV_2\\ \dfrac{\left(\tau-SV_3+(SV_3-\tau)\,\gamma_{(\tau)}\right)}{SV_3-SV_2} & SV_2<\tau\leq SV_3\\ 1 & \text{Otherwise.}\end{cases}$$

Definition 2.5. "Comparison between two SVTrN"[30]:
Let $\widehat{f}=\left\langle\left(f_1,g_1,h_1;\varsigma_{\widehat{f}}(\tau),\varphi_{\widehat{f}}(\tau),\gamma_{\widehat{f}}(\tau)\right)\right\rangle$ and $\widehat{g}=\langle(f_2,g_2,h_2;\varsigma_{\widehat{g}}(\tau),\varphi_{\widehat{g}}(\tau),\gamma_{\widehat{g}}(\tau))\rangle$ be the two SVTrN followed by the addition and subtraction operators:

- $\widehat{f}+\widehat{g}=\left\langle\left(f_1+f_2,g_1+g_2,h_1+h_2;\varsigma_{\widehat{f}}(\tau)\wedge\varsigma_{\widehat{g}}(\tau),\varphi_{\widehat{f}}(\tau)\vee\varphi_{\widehat{g}}(\tau),\gamma_{\widehat{f}}(\tau)\vee\gamma_{\widehat{g}}(\tau)\right)\right\rangle$
- $\widehat{f}-\widehat{g}=\left\langle\left(f_1-f_2,g_1-g_2,h_1-h_2;\varsigma_{\widehat{f}}(\tau)\wedge\varsigma_{\widehat{g}}(\tau),\varphi_{\widehat{f}}(\tau)\vee\varphi_{\widehat{g}}(\tau),\gamma_{\widehat{f}}(\tau)\vee\gamma_{\widehat{g}}(\tau)\right)\right\rangle$
- $\delta\cdot\widehat{f}=\begin{cases}\left(\delta f_1,\delta g_1,\delta h_1;\varsigma_{\widehat{f}}(\tau),\varphi_{\widehat{f}}(\tau),\gamma_{\widehat{f}}(\tau)\right) & (\delta>0)\\ \left(\delta h_1,\delta g_1,\delta f_1;\varsigma_{\widehat{f}}(\tau),\varphi_{\widehat{f}}(\tau),\gamma_{\widehat{f}}(\tau)\right) & (\delta<0)\end{cases}$
- $\widehat{f}^{-1}=\left\langle\left(\frac{1}{h_1},\frac{1}{g_1},\frac{1}{f_1};\varsigma_{\widehat{f}}(\tau),\varphi_{\widehat{f}}(\tau),\gamma_{\widehat{f}}(\tau)\right)\right\rangle\quad(\widehat{f}\neq 0)$

Definition 2.6. "Score function between the comparison of two SVTrN"[31]:
Let $\widehat{f}=\left\langle\left(f_1,g_1,h_1;\varsigma_{\widehat{f}}(\tau),\varphi_{\widehat{f}}(\tau),\gamma_{\widehat{f}}(\tau)\right)\right\rangle$ be the SVTrN, then

- Score function : $e(\widehat{f})^n=\frac{1}{4}\left[f_1+2g_1+h_1\right]*\frac{2+\varsigma_{\widehat{f}}(\tau)-\varphi_{\widehat{f}}(\tau)-\gamma_{\widehat{f}}(\tau)}{3}$
- Accuracy function: $I(\widehat{f})^n=\frac{1}{4}\left[f_1+2g_1+h_1\right]*\frac{2+\varsigma_{\widehat{f}}(\tau)-\varphi_{\widehat{f}}(\tau)+\gamma_{\widehat{f}}(\tau)}{3}$

Definition 2.7. "Order relation between two SVTrN"[32]:
Let $\widehat{f}=\left\langle\left(f_1,g_1,h_1;\varsigma_{\widehat{f}}(\tau),\varphi_{\widehat{f}}(\tau),\gamma_{\widehat{f}}(\tau)\right)\right\rangle$ and $\widehat{g}=\langle(f_2,g_2,h_2;\varsigma_{\widehat{g}}(\tau),\varphi_{\widehat{g}}(\tau),\gamma_{\widehat{g}}(\tau))\rangle$ defines the order relation between based on the score and the accuracy:

1. If $e(\widehat{f})^n>e(\widehat{g})^n$, implies $\widehat{f}>\widehat{g}$
2. If $e(\widehat{f})^n<e(\widehat{g})^n$, implies $\widehat{f}<\widehat{g}$
3. If $e(\widehat{f})^n=e(\widehat{g})^n$, implies
   - i. If $I(\widehat{f})^n>I(\widehat{g})^n$, implies $\widehat{f}>\widehat{g}$
   - ii. If $I(\widehat{f})^n<I(\widehat{g})^n$, implies $\widehat{f}<\widehat{g}$
   - iii. If $I(\widehat{f})^n=I(\widehat{g})^n$, implies $\widehat{f}=\widehat{g}$

3 Proposed Algorithm

In this section, we explore two distinct proposed algorithms for solving the CPM problem. The first proposed algorithm employs a extended version of the Dijkstra algorithm (DA), while the second proposed algorithm uses the linear programming (LP) technique to find the optimality check specifically tailored for CPM.

3.1 Algorithm 1:

Notations

$(et)^n$ indicates "earliest times of activities"

$(eft)^n$ indicates "earliest finish times of activities"

${D_{ij}}^n$ indicates "SVTrN duration of activities"

$(lt)^n$ indicates "latest times of activities"

(lft^n) indicates "latest finish time"

$\left(\widetilde{TC}\right)^n$ indicates "total project duration"

Δ indicates "starting node"

∇ indicates "destination node"

$\sum_{i=1}^{n} x_{ij}$ indicates "total flow out of node n"

$\sum_{j=1}^{n} x_{ji}$ indicates "total inflow into node n"

CP_{ij}^N indicates "objective function in neutrosophic environment"

CP_{ij} indicates "objective function in crisp"

$\widetilde{K}_i$ indicates "net flow requirement for different types of nodes i"

Algorithm 1 Solving CPM under neutrosophic theory using the proposed DA

Start

Step 1: Provide the input parameters for the network project. Select the type of network graph and identify the relationship between the predecessor and the successor to determine the duration of the activity, starting from the start node $(start)^n$ and ending at the end node $(end)^n$.

Step 2: The network operates using the activity-on-arrow (AOA) method, which represents the project as a graph with n-nodes, labelled as $\{N_1, N_2, \ldots, N_n\}$. The edges, denoted as pairs of nodes (e_i, e_j), represent connections between the nodes. Each edge is associated with a duration D_{ij}^n, expressed as SVTrN.

Step 3: Compute the neutrosophic earliest time parameters $(et)^n$ for each activity. Assign the earliest finish time $(eft)^n$ of the starting node N_1 as a permanent node $\langle(0,0,0;1,0,0)\rangle^*$ and label all other nodes N_{n+1} temporarily as $\langle(0,0,0;1,0,0)\rangle$.

Step 4: For each activity (i,j), compute the earliest finish time $(eft)_j^n$ using the formula:

$$(eft)_j^n = \max\left((eft)_i + D_{ij}^n\right), \text{ for all } (i,j) \in \text{activities}$$

This calculates the maximum time required to complete all preceding activities.

Step 5: Once the earliest finish time $(eft)^n$ is identified for an activity (i,j), mark the activity as fixed (denoted as $(*)$). If there are edges not connected to nodes i and j, assign their duration as $D_{ij}^n = \infty$, indicating no direction.

Step 6: Repeat steps 4 and 5 for all activities until the end node N_n is reached.

Step 7: Compute the latest time parameters $(lt)^n$ for each activity. Set the latest finish time (lft^n) of the final activity N_n equal to its earliest finish time and mark the node as permanent. This establishes $(eft^n) = (lft^n)$ for the final node.

Step 8: Allocate and assign temporary latest finish times (lft^n) to all nodes of N_{n-1} by considering their succeeding nodes, if the edges are not connected to i and j.

Step 9: For each succeeding vertex i, calculate the (lft^n) from the end node $(end)^n$ to the start node $(start)^n$:

$$lft_i^n = \min\left(lft_j^n - D_{ij}^n\right), \text{ for all } (i,j) \in \text{activities}$$

Step 10: Repeat step 9, moving backward through the network, until it reaches the start node $(start)^n$.

Step 11: Identify the critical path and non-critical path and calculate the total project duration $\widetilde{TC}^n$.

Stop

3.2 Algorithm 2

Algorithm 2 Solving CPM under neutrosophic theory using proposed LP

Start

Step 1: Consider the neutrosophic model that is as follows:

$$\text{Maximize} \sum_{i=1}^{n}\sum_{j=1}^{n} CP_{ij}^{N} x_{ij} \tag{3.1}$$

Subject to constraints:

$$\sum_{i=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = \widetilde{K}_i \tag{3.2}$$

for all $x_{ij} \in \mathbb{R},$ non-negative, where $i, j = 1, 2, \ldots, n$

$$\widetilde{K}_i = \begin{cases} 1 & \text{if} \quad i = \Delta, \\ 0 & \text{if} \quad i = \Delta + 1, \Delta + 2, \ldots, \nabla - 1, \\ -1 & \text{if} \quad i = \nabla. \end{cases} \tag{3.3}$$

Here, $CP_{ij}^{N} = \langle (SV_1, SV_2, SV_3; \varsigma_{(\tau)}, \varphi_{(\tau)}, \gamma_{(\tau)}) \rangle$ represents the SVTrN, where SV_1, SV_2, SV_3 are the components of the triangular neutrosophic number, and $\varsigma_{(\tau)}, \varphi_{(\tau)}, \gamma_{(\tau)}$ are the truth, indeterminacy, and falsity membership functions, respectively.

Step 2: Utilizing the ranking function defined in definition 2.6, the implementation of the NCPM into a CCPM[33] is as follows:

$$\text{Maximize} \sum_{i=1}^{n}\sum_{j=1}^{n} CP_{ij} x_{ij} \tag{3.4}$$

Subject to constraints:

$$\sum_{i=1}^{n} x_{ij} - \sum_{j=1}^{n} x_{ji} = \widetilde{K}_i \tag{3.5}$$

for all $x_{ij} \in \mathbb{R},$ non-negative, where $i, j = 1, 2, \ldots, n$

$$\widetilde{K}_i = \begin{cases} 1 & \text{if} \quad i = \Delta, \\ 0 & \text{if} \quad i = \Delta + 1, \Delta + 2, \ldots, \nabla - 1, \\ -1 & \text{if} \quad i = \nabla. \end{cases} \tag{3.6}$$

Step 3: Find the objective and its corresponding variable values using any of the optimal software such as Lingo, TORA, or MATLAB, and find all the values of CP$_{ij}$.

Step 4: After executing step 3 substitute the variable value in the respective objective equation 3.1,3.2,3.3 to obtain the form SVTrN as follows $CP_{ij}^{N} = \langle (SV_1, SV_2, SV_3; \varsigma_{(\tau)}, \varphi_{(\tau)}, \gamma_{(\tau)}) \rangle$.

Stop

4 Illustrative Example

The following subsection highlights some adaptations in network studies by drawing on various examples from existing literature, thereby complementing the potential applications of proposed algorithms 1 and 2.

Example 4.1. Considering a network[34] shown in figure 2, by having the weighted network $G = (V, E)$, where V represents $n-vertices$ from $A, B, .., F$; where the node A is the start node $(start)^n$ and node F be the end node $(end)^n$ with the corresponding edge weights having the possible connections from $(A, B), .., (E, F)$ as activity durations as shown in table 2. For the network, the triplet $\varsigma_{(\tau)}$, $\varphi_{(\tau)}$, and $\psi_{(\tau)}$ is determined as $(1, 0, 0)$, indicating the certainty in the truth membership function.

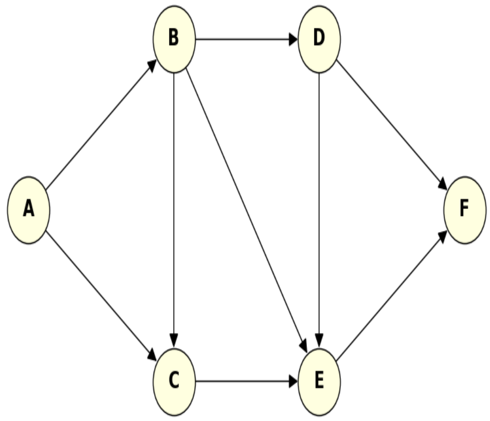


Figure 2: Network diagram (6 nodes with 9 arc lengths[34])

Table 2: Activity durations with predecessors

Activity	Predecessor	Activity Durations
B	A	$(6, 12, 18)$
C	A	$(13, 25, 33)$
C	B	$(15, 20, 29)$
D	B	$(2, 11, 20)$
E	B	$(7, 16, 25)$
E	C	$(17, 25, 33)$
E	D	$(8, 9, 10)$
F	D	$(15, 21, 31)$
F	E	$(4, 11, 14)$

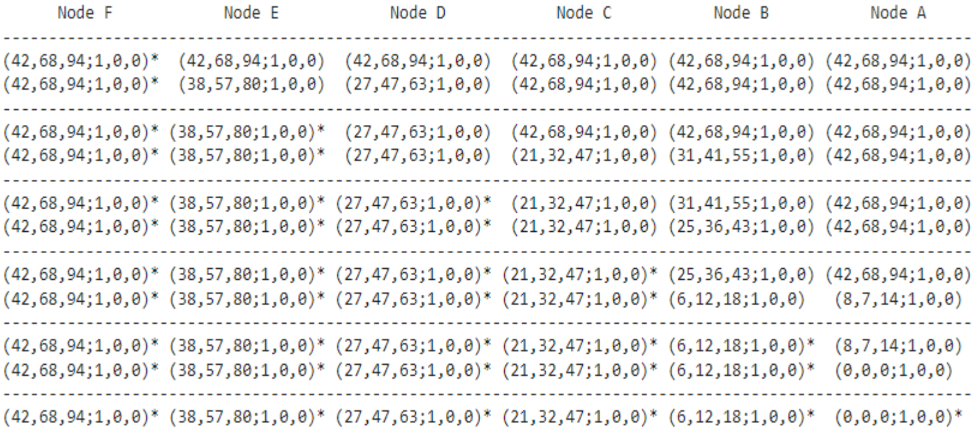
Solution: (i) Proposed algorithm 1 is executed numerically as follows:

Based on the information provided in figure 2 and table 2, the implementation of steps 1 and 2 on the AOA network is identified. Further, using steps 3 to 6 from algorithm 1, an analysis of the forward computation calculation approach is conducted using the Python programming language using the data taken from table 2 and the findings are visualized in figure 3.

```
     Node A          Node B            Node C            Node D            Node E            Node F
---------------------------------------------------------------------------------------------------------
(0,0,0;1,0,0)*  (0,0,0;1,0,0)    (0,0,0;1,0,0)    (0,0,0;1,0,0)     (0,0,0;1,0,0)     (0,0,0;1,0,0)
(0,0,0;1,0,0)* (6,12,18;1,0,0)   (13,25,33;1,0,0)       ∞                 ∞                 ∞
---------------------------------------------------------------------------------------------------------
(0,0,0;1,0,0)* (6,12,18;1,0,0)*  (13,25,33;1,0,0)       ∞                 ∞                 ∞
(0,0,0;1,0,0)* (6,12,18;1,0,0)*  (21,32,47;1,0,0) (8,23,38;1,0,0)   (13,28,43;1,0,0)        ∞
---------------------------------------------------------------------------------------------------------
(0,0,0;1,0,0)* (6,12,18;1,0,0)* (21,32,47;1,0,0)* (8,23,38;1,0,0)   (13,28,43;1,0,0)        ∞
(0,0,0;1,0,0)* (6,12,18;1,0,0)* (21,32,47;1,0,0)* (8,23,38;1,0,0)   (38,57,80;1,0,0)        ∞
---------------------------------------------------------------------------------------------------------
(0,0,0;1,0,0)* (6,12,18;1,0,0)* (21,32,47;1,0,0)* (8,23,38;1,0,0)*  (38,57,80;1,0,0)        ∞
(0,0,0;1,0,0)* (6,12,18;1,0,0)* (21,32,47;1,0,0)* (8,23,38;1,0,0)*  (38,57,80;1,0,0)  (23,44,69;1,0,0)
---------------------------------------------------------------------------------------------------------
(0,0,0;1,0,0)* (6,12,18;1,0,0)* (21,32,47;1,0,0)* (8,23,38;1,0,0)* (38,57,80;1,0,0)*  (23,44,69;1,0,0)
(0,0,0;1,0,0)* (6,12,18;1,0,0)* (21,32,47;1,0,0)* (8,23,38;1,0,0)* (38,57,80;1,0,0)* (42,68,94;1,0,0)
---------------------------------------------------------------------------------------------------------
(0,0,0;1,0,0)* (6,12,18;1,0,0)* (21,32,47;1,0,0)* (8,23,38;1,0,0)* (38,57,80;1,0,0)* (42,68,94;1,0,0)*
```

Figure 3: Dijkstra's earliest time using a forward pass method

From step 7 to Step 10, the calculation of the backward pass method is also conducted in Python programming language using the data from table 2 and using the network from the figure 2; and the results is shown in figure 4.



Node F	Node E	Node D	Node C	Node B	Node A
(42,68,94;1,0,0)*	(42,68,94;1,0,0)	(42,68,94;1,0,0)	(42,68,94;1,0,0)	(42,68,94;1,0,0)	(42,68,94;1,0,0)
(42,68,94;1,0,0)*	(38,57,80;1,0,0)	(27,47,63;1,0,0)	(42,68,94;1,0,0)	(42,68,94;1,0,0)	(42,68,94;1,0,0)
(42,68,94;1,0,0)*	(38,57,80;1,0,0)*	(27,47,63;1,0,0)	(42,68,94;1,0,0)	(42,68,94;1,0,0)	(42,68,94;1,0,0)
(42,68,94;1,0,0)*	(38,57,80;1,0,0)*	(27,47,63;1,0,0)	(21,32,47;1,0,0)	(31,41,55;1,0,0)	(42,68,94;1,0,0)
(42,68,94;1,0,0)*	(38,57,80;1,0,0)*	(27,47,63;1,0,0)*	(21,32,47;1,0,0)	(31,41,55;1,0,0)	(42,68,94;1,0,0)
(42,68,94;1,0,0)*	(38,57,80;1,0,0)*	(27,47,63;1,0,0)*	(21,32,47;1,0,0)	(25,36,43;1,0,0)	(42,68,94;1,0,0)
(42,68,94;1,0,0)*	(38,57,80;1,0,0)*	(27,47,63;1,0,0)*	(21,32,47;1,0,0)*	(25,36,43;1,0,0)	(42,68,94;1,0,0)
(42,68,94;1,0,0)*	(38,57,80;1,0,0)*	(27,47,63;1,0,0)*	(21,32,47;1,0,0)*	(6,12,18;1,0,0)	(8,7,14;1,0,0)
(42,68,94;1,0,0)*	(38,57,80;1,0,0)*	(27,47,63;1,0,0)*	(21,32,47;1,0,0)*	(6,12,18;1,0,0)*	(8,7,14;1,0,0)
(42,68,94;1,0,0)*	(38,57,80;1,0,0)*	(27,47,63;1,0,0)*	(21,32,47;1,0,0)*	(6,12,18;1,0,0)*	(0,0,0;1,0,0)
(42,68,94;1,0,0)*	(38,57,80;1,0,0)*	(27,47,63;1,0,0)*	(21,32,47;1,0,0)*	(6,12,18;1,0,0)*	(0,0,0;1,0,0)*

Figure 4: Dijkstra's earliest time using a backward pass method

Based on the forward and backward analysis from figures 3 and 4; the obtained results generated as $e_A = l_A = \langle(0,0,0;1,0,0)\rangle$, $e_B = l_B = \langle(6,12,18;1,0,0)\rangle$, $e_C = l_C = \langle(21,32,47;1,0,0)\rangle$, $e_E = l_E = \langle(38,57,80;1,0,0)\rangle$, $e_F = l_F = \langle(42,68,94;1,0,0)\rangle$. The pictorial representation of the critical path which refers to the red line, and the non-critical path which refers to the black line in figure 5 obtained as the suggested neutrosophic critical path (SNCP) as $A \to B \to C \to E \to F$ by it's membership grade SNCPL is shown in figure 6.

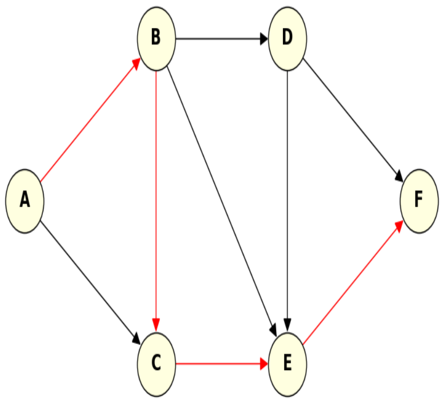


Figure 5: SNCP for example 4.1

From the results the suggested neutrosophic critical network length (SNCPL) is defined as $\langle(42,68,94;1,0,0)\rangle$ that is depicted in figure 6 indicating the range between 42 and 94, and the maximum possibility of the completion is 68 days.

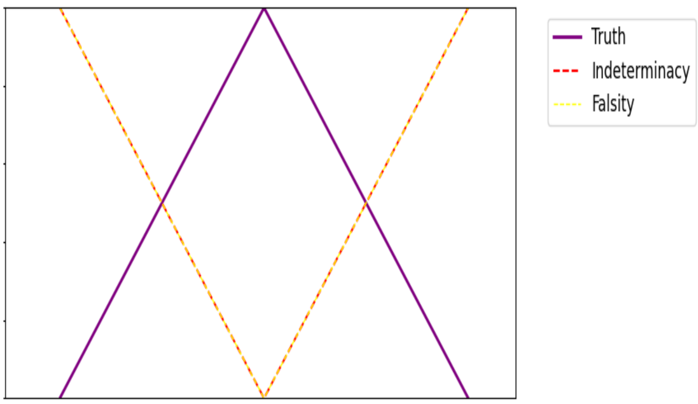


Figure 6: Determined membership grade solution (SNCPL) from example 4.1

(ii) Proposed algorithm 2 is executed numerically as follows:

Step 1: The SVTrN model solved by LP is detailed as follows:

$$\text{Max (CP)}^N = (6,12,18)\cdot x_{AB} + (13,25,33)\cdot x_{AC} + (15,20,29)\cdot x_{BC} + (2,11,20)\cdot x_{BD} + (7,16,25) \cdot x_{BE} + (17,25,33)\cdot x_{CE} + (8,9,10)\cdot x_{DE} + (15,21,31)\cdot x_{DF} + \langle(4,11,14)\rangle \cdot x_{EF}$$

Subject to the constraints:

$$x_{AB} + x_{AC} = 1;\ x_{BC} + x_{BE} + x_{BD} = x_{AB};\ x_{CE} = x_{BC} + x_{AC};\ x_{DE} + x_{DF} = x_{BD}; \quad x_{EF} = x_{CE} + x_{BE} + x_{DE};\ x_{DF} + x_{EF} = 1; v$$

After executing from steps 2 to 4 from the proposed algorithm 2; the SNCP results to the optimal path follows $x_{AB} \rightarrow x_{BC} \rightarrow x_{CE} \rightarrow x_{EF}$ and its SNCPL obtained of $\langle(42,68,94;1,0,0)\rangle$, which proportionally matches with the solution obtained in algorithm 1. The execution of the proposed model details the comparison in the table 3 that shows both proposed algorithms 1 and 2 yield consistent results with the existing fuzzy literature demonstrating the effectiveness of the neutrosophic approach in handling uncertainties by obtaining the same SNCPL on using both the proposed algorithms.

Table 3: Comparison with existing method (ref- example 4.1)

Siripurapu and Shankar[34]	**Our Proposed Model**	
	Algorithm 1	**Algorithm 2**
FCP: $A \rightarrow B \rightarrow C \rightarrow E \rightarrow F$	SNCP: $A \rightarrow B \rightarrow C \rightarrow E \rightarrow F$	SNCP: $x_{AB} \rightarrow x_{BC} \rightarrow x_{CE} \rightarrow x_{EF}$
FCPL: $(42,68,94)$	SNCPL: $\langle(42,68,94;1,0,0)\rangle$	SNCPL: $\langle(42,68,94;1,0,0)\rangle$

Example 4.2. Analysis of the study on the subsequent level of the project network, by discovering the outcomes of altering constant in $\left(\varsigma_{(\tau)}, \varphi_{(\tau)}, \gamma_{(\tau)}\right)$ as $(0.75, 0.1, 0.25)$ and maintaining (SV_1, SV_2, SV_3) parameters constant in the network framework[18] shown in figure 7 and it's activity durations is depicted in table 4. This scenario permits us to have a look at different stages of variation that can affect path analysis and criticality length and influence the time for completion of the project.

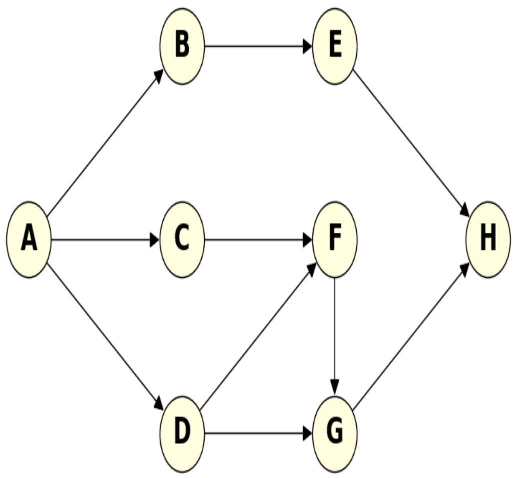


Figure 7: Network diagram (8 nodes with 10 arc lengths) (ref[18])

Table 4: Constant variation in SVTrN activity duration (ref[18])

Activity	Predecessor	Activity Durations
B	A	$(5, 6, 7)$
C	A	$(1, 3, 5)$
D	A	$(1, 4, 7)$
E	B	$(1, 2, 3)$
F	C	$(1, 2, 9)$
F	D	$(1, 5, 9)$
G	D	$(2, 2, 8)$
G	F	$(4, 4, 10)$
H	E	$(2, 2, 5)$
H	G	$(2, 2, 8)$

Solution: Applying steps 1-12 in the proposed algorithm 1, the obtained result of SNCP $A \to D \to F \to G \to H$ is shown in figure 8 and the SNCPL is shown in figure 9 as $\langle(8, 15, 34; 0.75, 0.1, 0.25)\rangle$, while implementing the proposed algorithm 1, to check the obtained optimality from the proposed algorithm 2 using from the proposed steps the obtained SNCP is $x_{AD} \to x_{DF} \to x_{FG} \to x_{GH}$ and SNCPL remains the same as algorithm 1 of example 4.2.

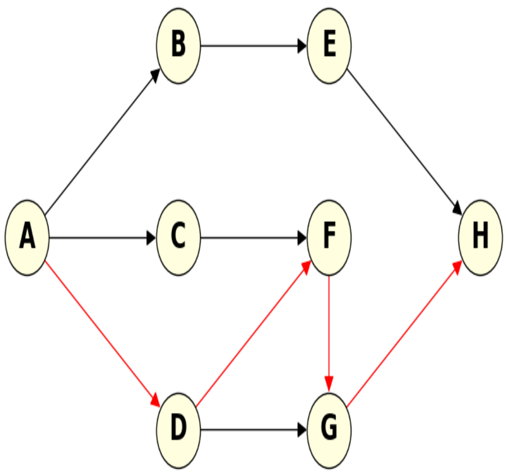


Figure 8: SNCP for example 4.2

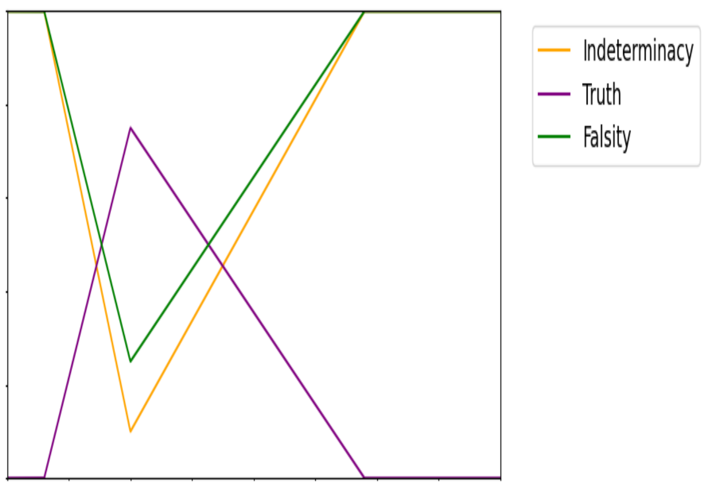


Figure 9: Determined SNCPL from example 4.2

Example 4.3. Using the network framework[18] shown in figure 7, the outcomes of altering $\left(\varsigma_{(\tau)}, \varphi_{(\tau)}, \gamma_{(\tau)}\right)$ and maintaining (SV_1, SV_2, SV_3)parameters is shown in table 5. This scenario permits an analysis of different stages of variation affecting path analysis and criticality length and influencing the time for project completion.

Table 5: Change in SVTrN activity duration (ref[18])

Activity	Predecessor	Activity Durations
B	A	$\langle(5, 6, 7; 0.85, 0.12, 0.15)\rangle$
C	A	$\langle(1, 3, 5; 0.92, 0.1, 0.08)\rangle$
D	A	$\langle(1, 4, 7; 0.9, 0.05, 0.1)\rangle$
E	B	$\langle(1, 2, 3; 0.82, 0.11, 0.18)\rangle$
F	C	$\langle(1, 2, 9; 0.8, 0.09, 0.2)\rangle$
F	D	$\langle(1, 5, 9; 0.93, 0.18, 0.07)\rangle$
G	D	$\langle(2, 2, 8; 0.86, 0.11, 0.14)\rangle$
G	F	$\langle(4, 4, 10; 0.99, 0.21, 0.01)\rangle$
H	E	$\langle(2, 2, 5; 0.88, 0.16, 0.12)\rangle$
H	G	$\langle(2, 2, 8; 0.8, 0.01, 0.2)\rangle$

Solution: Applying steps 1-12 in the proposed algorithm 1, the SNCPL results to $\langle(8, 15, 34; 0.8, 0.21, 0.2)\rangle$ shown in figure 10, and the obtained SNCP is $A \to D \to F \to G \to H$ was shown in figure 8. Using the algorithm 2, the completion SNCPL obtains the same as $\langle(8, 15, 34; 0.8, 0.21, 0.2)\rangle$ corresponding to its SNCP as $x_{AD} \to x_{DF} \to x_{FG} \to x_{GH}$.

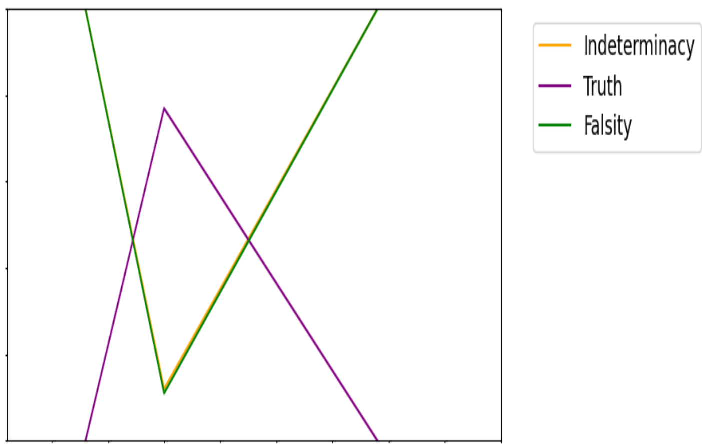


Figure 10: Determined SNCPL from example 4.3

From the above studies, both algorithms verify the accuracy and reliability of our approach. The consistency in the SNCPL demonstrates that our methodologies are robust and effective for handling the complexities and uncertainties inherent in project scheduling. The findings not only validate our use of neutrosophic sets in critical path analysis but also reinforce the applicability of using both proposed approaches in neutrosophic environments.

Contingent study

The section presents the examination of contingency for the proposed algorithms. Initially, the comparison of the existing fuzzy and neutrosophic environment is shown in table 3 for validation. From the results, it is evident that the proposed approach can further solve a new set of problems under the neutrosophic criteria, as shown in example 4.2 and example 4.3. Further, our results demonstrate the same criticality route from the networks,[18,34] and using the proposed algorithms, serve better optimality in solving new set of problems that wasn't addressed by the existing study has shown in table 6.

From our numerical analysis, using both proposed algorithms, the computed lengths identify as same that are used in each example of 4.1,4.2 and 4.3. The advantages of this proposed model include its flexibility and applicability to different uncertainty parameters, such as trapezoidal neutrosophic numbers, interval-valued

DOI: https://doi.org/10.54216/IJNS.250430

neutrosophic numbers and so on, making it a robust tool for various project management scenarios. However, there are some limitations to consider:

1. When there is any change in the network, the response will be slow as this modification will propagate node by node.
2. If a node failure occurs, there may be routing loops, further complicating the path finding process.

In conclusion, various methods for comparing neutrosophic numbers provide unique advantages and meet different desirable criteria. The proposed model, observing both NCP and SNCPL, achieves precise optimality in the evaluated approaches. This emphasizes the critical importance of selecting the right techniques tailored to specific project requirements and desired outcomes.

Table 6: Results and discussion

Ex	**Existing Literature Siripurapu et al.[18]**	**Our Proposed Model**	
		Example 4.2	**Example 4.3**
Algorithm 1	**NA**	SNCPL: $\langle(8, 15, 34; 0.75, 0.1, 0.25)\rangle$ SNCP: $A \to D \to F \to G \to H$	SNCPL: $\langle(8, 15, 34; 0.8, 0.21, 0.2)\rangle$ SNCP: $A \to D \to F \to G \to H$
Algorithm 2		SNCPL: $\langle(8, 15, 34; 0.75, 0.1, 0.25)\rangle$ SNCP: $x_{AD} \to x_{DF} \to x_{FG} \to x_{GH}$	SNCPL: $\langle(8, 15, 34; 0.8, 0.21, 0.2)\rangle$ SNCP: $x_{AD} \to x_{DF} \to x_{FG} \to x_{GH}$

Conclusion

The proposed model effectively integrates proposed algorithms under neutrosophic criteria to solve the CPM. The study of three examples highlights the model's robustness and adaptability to varying degrees of uncertainty. In example 4.1, showed validation of fuzzy confirming the model's performance under neutrosophic. Further, example 4.2 demonstrated the model's capability by directly applying it to a neutrosophic environment, achieving reliable results. Example 4.3 explored the variability of truth, indeterminacy, and falsity, illustrating the different parameters impact the outcomes. To enhance computational efficiency, the proposed model was implemented using Python and LINGO for proposed algorithms. These programming languages not only improve computation speed but also provide robust frameworks for handling complex calculations. Additionally, the flexibility of software tools allows for the incorporation of various uncertainty theories, facilitating comprehensive time analysis under different scenarios. This further emphasizes the importance of selecting appropriate techniques and tools based on specific project requirements and desired outcomes for uncertainty handling. Future research could explore the application of this model to the extended fuzzy principles by expanding its utility.

References

[1] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

[2] V. Gacevski, M. Lazarevska, and Z. Zafirovski. Ranking of total time reserves for determination of the critical path in fuzzy network plan. *Journal of Applied Engineering Science*, 22(2):297–302, 2024.

[3] Z. K. Nemaa and G. K. Aswed. Forecasting construction time for road projects and infrastructure using the fuzzy pert method. *Iop Conference Series: Materials Science and Engineering*, 1076(1):012123, 2021.

[4] M. G. Alharbi, H. A. E. W. Khalifa, and E. E. Ammar. An interactive approach for solving the multiobjective minimum cost flow problem in the fuzzy environment. *Journal of Mathematics*, 2020:1–7, 2020.

[5] L. Lin, C. Wu, and L. Ma. A genetic algorithm for the fuzzy shortest path problem in a fuzzy network. *Complex & Intelligent Systems*, 7(1):225–234, 2021.

[6] C.A. Oliveira and P.M. Pardalos. *Mathematical aspects of network routing optimization*. Springer, Berlin/Heidelberg, Germany, 2011.

[7] B. Korte and J. Vygen. *Combinatorial optimization: theory and algorithms*. Springer-Verlag, Berlin, Germany, 4th edition, 2008.

[8] S. Chanas and J. Kamburowski. The use of fuzzy variables in pert. *Fuzzy Sets and Systems*, 5(1):11–19, 1981.

[9] Y. Dorfeshan and S.M. Mousavi. A new interval type-2 fuzzy decision method with an extended relative preference relation and entropy to project critical path selection. *International Journal of Fuzzy System Applications (ijfsa)*, 8(1):19–47, 2019.

[10] N. Rameshan and D.S. Dinagar. A method for finding critical path with symmetric octagonal intuitionistic fuzzy numbers. *Advances in Mathematics: Scientific Journal*, 9(11):9273–9286, 2020.

[11] T. Yogashanthi, K. Prabakaran, and K. Ganesan. Application of intuitionistic fuzzy critical path method on airfreight ground operation systems. *Journal of Mathematical and Computational Science*, 11(4):4518–4534, 2021.

[12] M. Akram, A. Habib, and M. Deveci. Application of critical path method in epropertywatch plan using gaussian pythagorean fuzzy numbers. *Ieee Transactions On Fuzzy Systems*, 2023.

[13] A. Chakraborty, S. P. Mondal, S. Alam, D. Pamucar, and D. Marinkovic. A new idea to evaluate networking problem and mcgdm problem in parametric interval valued pythagorean arena. *Discrete Dynamics in Nature and Society*, 2022(1):1–20, 2022.

[14] E. Akan and S. Bayar. Interval type-2 fuzzy program evaluation and review technique for project management in shipbuilding. *Ships and Offshore Structures*, 17(8):1872–1890, 2022.

[15] R. J. Mitlif and F. A. Sadiq. Finding the critical path method for fuzzy network with development ranking function. *Journal of Al-qadisiyah for Computer Science and Mathematics*, 13(3):98–106, 2021.

[16] M.K. Mehlawat and N. Grover. Intuitionistic fuzzy multi-criteria group decision making with an application to critical path selection. *Annals of Operations Research*, 269:505–520, 2018.

[17] S. M. Younis and A. Y. Yousif. Finding the fuzzy critical path with octagonal fuzzy numbers using linear programming model. In *7th International Conference on Contemporary Information Technology and Mathematics (ICCITM)*, 261–266. IEEE, 2021.

[18] A. Siripurapu, R.S. Nowpada, and K. S. Rao. Improving dijkstra's algorithm for estimating project characteristics and critical path. *Reliability: Theory & Applications*, 17(4(71)):65–73, 2022.

[19] B. Abinaya and E.C.H. Amirtharaj. An alternative method for finding the critical path of the network in fuzzy time cost trade off problem. *Indian Journal of Science and Technology*, 17(10):949–954, 2024.

[20] F. Smarandache. A unifying field in logics: neutrosophic logic. In *Philosophy*, 1–141. American Research Press, 1999.

[21] K.T. Atanassov. Intuitionistic fuzzy sets. *Intuitionistic Fuzzy Sets: Theory and Applications*, 1–137, 1999.

[22] A. Chakraborty, S. P. Mondal, S. Alam, and A. Mahata. Cylindrical neutrosophic single-valued number and its application in networking problem, multi-criterion group decision-making problem and graph theory. *Caai Transactions on Intelligence Technology*, 5(2):68–77, 2020.

[23] A. Chakraborty. A new score function of pentagonal neutrosophic number and its application in networking problem. *International Journal of Neutrosophic Science*, 1(1):40–51, 2020.

[24] N. Hema and S. Rajeswari. An approach to solve network problem under octogonal neutrosophic environment. In *AIP Conference Proceedings*, volume 2649, 030015(1–11), 2023.

[25] S. Sinika and G. Ramesh. Trapezoidal neutrosophic program evaluation and review technique using interval arithmetic operations. *Iaeng International Journal of Applied Mathematics*, 54(3), 2024.

[26] M. Abdel-Basset, M. Ali, and A. Atef. Uncertainty assessments of linear time-cost tradeoffs using neutrosophic set. *Computers & Industrial Engineering*, 141:106286–106301, 2020.

[27] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, Orlando, FL, 1980.

[28] H. Wang, F Smarandache, Y. Q. Zhang, and R. Sunderraman. Single valued neutrosophic sets. *Multispace and Multistructure*, 4:410–413, 2010.

[29] I. Deli and Y. Subaş. Single valued neutrosophic numbers and their applications to multicriteria decision making problem. *Neutrosophic Sets and Systems*, 2(1):1–13, 2014.

[30] M. Abdel-Basset, M. Mohamed, and F. Smarandache. Linear fractional programming based on triangular neutrosophic numbers. *International Journal of Applied Management Science*, 11(1):1–20, 2019.

[31] S. Dhouib. Optimization of travelling salesman problem on single valued triangular neutrosophic number using dhouib-matrix-tsp1 heuristic. *International Journal of Engineering*, 34(12):2642–2647, 2022.

[32] B. E. El-Demerdash, S. A. Edalatpanah, and H. A. E. W. Khalifa. On treating input oriented data envelopment analysis model under neutrosophic environment. *Neutrosophic Sets and Systems*, 60(1):17, 2023.

[33] S.-P. Chen. Analysis of critical paths in a project network with fuzzy activity times. *European Journal of Operational Research*, 183(1):442–459, 2007.

[34] A. Siripurapu and R.S. Nowpada. Project characteristics with triangular fuzzy number. *Reliability: Theory & Applications*, 18(2 (73)):359–369, 2023.