HyperSoft Set Methods in Engineering

Journal Homepage: sciencesforce.com/hsse

HyperSoft Set Meth. Eng. Vol. 1 (2024) 34–45

Paper Type: Original Article

Evaluation of Shortest Path by using Breadth-First Algorithm under Neutrosophic Environment

Prasanta Kumar Raut [1], Siva Prasad Behera [1,*], Said Broumi [2], and Amarendra Baral [3]

[1] Department of Mathematics, C.V. Raman Global University, Bhubaneswar-752054, Odisha, India;
Emails: 21080012@cgu-odisha.ac.in, sivaprasad@cgu-odisha.ac.in.

[2] Laboratory of Information Processing, Faculty of Science Ben M'Sik, University Hassan II, Casablanca, Morocco;
s.broumi@flbenmsik.ma.

[3] Trident Academy of Technology, Bhubaneswar, Odisha, India; deanssh@tat.ac.in.

Received: 30 Aug 2023 **Revised:** 18 Dec 2023 **Accepted:** 10 Jan 2024 **Published:** 17 Jan 2024

Abstract

This paper recommends and designs concepts for evaluating the shortest path (SP) for a connected network using a modified breadth-first search algorithm in an uncertain environment. Evaluating the SPs of a network is an essential and extensively encountered optimization problem. Here we develop a new method for determining the SP in a neutrosophic environment in which the arc lengths are uncertain. Here, we use the parameters as neutrosophic numbers, and the new methodology, i.e., canonical representation of neutrosophic numbers in a neutrosophic environment, is used to convert neutrosophic edge lengths to crisp edge lengths and enhance the traditional breadth-first search algorithm. We employ this concept to accommodate the ambiguities in subjective decisions and address these issues. Here we demonstrate that the proposed algorithm has good stability and high efficiency for evaluating the SP. Finally, a numerical example is provided to explain the suggested algorithm.

Keywords: Breadth-First Algorithm, Connected Network, Neutrosophic Numbers, Shortest Path Problem.

1 | Introduction

Lotfi Aliasker Zadeh first suggested a new thing that deals with uncertainty, i.e., fuzzy sets [1], which include membership degrees that are independent and uncertain. Later on, Smarandache established that the neutrosophic set (NS) is a field of philosophy that surveys the existence, provenance, and extent of uniformity as well as its relationship to different theoretical peaks, including parameters such as true(T), indeterminacy (I), and false membership (F) [2]. It has been established that when the inconsistencies of a set of vertices and edges are obtained in a network, a neutrosophic number can be used to evaluate the SPP. If the connection of nodes in a network is uncertain, the NS theory is an adequate principle for addressing real-world issues in which the arc length is considered the neutrosophic number (NN) for encountering SPP [3].

SPP is a fundamental issue that arises in many areas of science and technology. The SPP's objective is to determine the minimum path distance between a given pair of nodes in a network [4]. The distance from end to end of an arc can signify actual quantities of uniqueness, for example, cost and time. The arc lengths between

Corresponding Author: sivaprasad@cgu-odisha.ac.in
https://doi.org/10.61356/j.hsse.2024.18350

nodes in a connected network are typically accurately measured using traditional SPP, with precise numerical representations in specific conditions. However, in situations of ambiguity, neutrosophic numbers are employed.

The NS is an extended version of the IFS that takes ambiguity into consideration and explains the actual issue in full depth. Too many researchers often use NSs arc length weight values to evaluate the SPP in different environments. Broumi et al. [5] suggested an SPP focusing on neutrosophic environments by using a triangular fuzzy number. Deng, Y., et al. [6] solved SPP using the Dijkstra algorithm with neutrosophic numbers. As an additional simplification of the ideas of intuitionistic fuzzy, hesitant fuzzy, and dual hesitant FSs, Ye [7] suggested a single-valued neutrosophic hesitant FS. Peng [8] suggested new methods for solving multi-valued NSs and enhanced a relationship based on intuitionistic and hesitant FSs. Using score and accuracy functions and several operational rules, Ye [9] suggested a new method to evaluate the SPP in neutrosophic environments with a trapezoidal neutrosophic number. Harish and Nancy [10] suggested an enhanced score function in the executive process to evaluate the SPP. Ridvan et al. [11] devised a technique for solving several characteristic executive problems using SVN or IVNN numbers. Deli et al. [12] created Euclidean ranking standard values for SVTN numbers.

Zhang et al. [13] recommended a cyclic uncertainty on SPP. After that, Broumi et al. [14] evaluated SPP by using SVNG. Peng and Dai [15] suggested a neutrosophic environment-based interval decision-making algorithm to calculate the SPP, and Broumi, S. et al. [16] suggested an SV-trapezoidal NN. Subas and Deli [17] suggested the SVNN ranking technique to evaluate the SP. Broumi et al. [18], [19] proposed new concepts to deal with uncertainty, i.e., the neutrosophic set. Kahraman and Bolturk [20] suggested an IVN analytical hierarchy procedure based on similarity measures. Wang et al. [21] clarified NS with interval value. Biswas et al. [22] suggested a new technique to calculate the distance, i.e., based on the TrFNN interval. Deli [23] suggested a classical representation for the diminution and extension of neutrosophic traditional sets, followed by the development of a neutrosophic traditional soft reduction technique and the presentation of a real-world example. after then Deli [24] suggested and utilised single-valued trapezoidal NN as a variable to evaluate the decision-making.

Deli and Suba [25] solved various decision-making problems by applying the weighted mathematical operator with the SVN method. Basset et al. [26], [27] suggested a new concept based on managerial technology that allows decision-makers in a neutrosophic environment to select the most appropriate project. They also suggested a new method for assessing reliability and evaluating the extent of agreement between expert opinions in a NE. Kumar et al. [28] proposed a trapezoidal fuzzy neutrosophic environment algorithm for solving SPP. Broumi et al. [29] suggested using single-valued trapezoidal NN to evaluate the SP length of a network. Tan et al. [30] suggested a complete dynamic programming technique for evaluating the SPP. Broumi et al. [31] suggested a technique score function to evaluate the SPP. Goldfarb, D. et al.[32] previously suggested for finding shortest path where crisp number are arc lengths. But in this paper we propose a method for finding SPP in uncertain environment.

The subsequent portion of the research paper is organized as follows: Section-2 elucidates the motivation and contributions of this paper. Section-3 presents definitions of existing terminologies. In Section 4, novel algorithms utilizing the proposed score function are suggested. Section 5 provides a numerical example demonstrating the calculation of NSP in neutrosophic environments. Section 6 contrasts the shortest path across various networks and parameters, along with outlining the benefits of the proposed approach. The conclusion of the proposed methodology is detailed in Section 7.

2 | Motivation

- The primary aim of this research is to introduce a productive computational approach for SPP under Bredhth's first algorithm that is adaptable to an ambiguous environment commonly faced in the procedure. The following are the paper's most significant contributions:
- We determined it by calculating the arc length on a neutrosophic network using the neutrosophic number FNSP as the vertex. Each vertex is assigned a neutrosophic number.
- Neutrosophic number NS is an extension of an intuitionistic fuzzy number.
- We introduce a novel algorithm designed to address the Shortest Path Problem (SPP) within an uncertain environment, which accurately computes the shortest path length between two specified nodes.
- In addition, we employ a selection sort algorithm to select the SP connected to the lowest rank.

3 | Preliminaries

In this part we show a general description of a Intuitionistic Fuzzy Set, neutrosophic set.

Definition 3.1 [2]: Assume set $\ddot{X}$ is the universal set. An intuitionistic fuzzy set $\ddot{A}$ in $\ddot{X}$ is written in the form:
$\ddot{A} = \{\ddot{x}, \mu_{\ddot{A}}(x), \nu_{\ddot{A}}(\ddot{x})$------------- (1)

With $\mu_{\ddot{A}}: \ddot{X} \to [0,1]$ and $\nu_{\ddot{A}}: \ddot{X} \to [0,1]$ are the functions that define the degrees of membership non-membership of $x \in \ddot{X}$ to $\ddot{A} \in \ddot{X}$, respectively, and for every $x \in \ddot{X}$, $\mu_{\ddot{A}}(x), \nu_{\ddot{A}}(\ddot{x}) \leq 1$.

Definition3.2: Let's suppose that $\ddot{X}$ represents a set of spatial points (objects), where $\tilde{x}$ denotes the corresponding generic elements in $\ddot{X}$. Then, the element in the neutrosophic set $\widetilde{A}$ takes the following form:

$\widetilde{A} = \{< \ddot{x}: \ddot{T}_{\ddot{A}}(\ddot{x}), \ddot{I}_{\ddot{A}}(\ddot{x}), \ddot{F}_{\ddot{A}}(\ddot{x}) > \ddot{x} \epsilon \ddot{X}\}$------------------------ (2)

Where three membership degree $\ddot{T}_{\ddot{A}}, \ddot{I}_{\ddot{A}}, \ddot{F}_{\ddot{A}}: \ddot{X} \to [0^-, 1^+]$ where $\ddot{T}$, $\ddot{I}$, and $\ddot{F}$ represent the truth function, the indeterminacy function, and the falsity function, respectively.

$$0^- \leq \{\ddot{T}_{\ddot{A}}(\ddot{x}) + \ddot{I}_{\ddot{A}}(\ddot{x}) + \ddot{F}_{\ddot{A}}(\ddot{x})\} \leq 3^+$$

Now $\ddot{T}_{\ddot{A}}(\ddot{x}), \ddot{I}_{\ddot{A}}(\ddot{x}), \ddot{F}_{\ddot{A}}(\ddot{x})$ are representing subsets of the interval $[0^-, 1^+]$ makes it challenging to implement neutrosophic sets in real-world situations.

Definition 3.3: Canonical representation of neutrosophic number:

If $\ddot{A} = (\ddot{p}, \ddot{q}, \ddot{r})$ is neutrosophic number. Then the Canonical representation of neutrosophic number is defined as:

$$C(\ddot{A}) = \frac{1}{6}(\ddot{p} + 4\ddot{q} + \ddot{r}) - - - - - - - - - - - - (3)$$

The canonical representation of neutrosophic numbers $\ddot{A} = (\ddot{p}_1, \ddot{q}_1, \ddot{r}_1)$ and $\ddot{B} = (\ddot{p}_2, \ddot{q}_2, \ddot{r}_2)$ is defined as follows.

$$C(\ddot{A}) = \frac{1}{6}(\ddot{p}_1 + 4\ddot{q}_1 + \ddot{r}_1)$$

$$C(\ddot{B}) = \frac{1}{6}(\ddot{p}_2 + 4\ddot{q}_2 + \ddot{r}_2)$$

The operation of adding two neutrosophic numbers can be described as:

$$C(\ddot{A} + \ddot{B}) = \frac{1}{6}(\ddot{p}_1 + 4\ddot{q}_1 + \ddot{r}_1) + \frac{1}{6}(\ddot{p}_2 + 4\ddot{q}_2 + \ddot{r}_2) --------(4)$$

The operation of multiplication two neutrosophic numbers can be described as:

$$C(\ddot{A} \times \ddot{B}) = \frac{1}{6}(\ddot{p}_1 + 4\ddot{q}_1 + \ddot{r}_1) \times \frac{1}{6}(\ddot{p}_2 + 4\ddot{q}_2 + \ddot{r}_2) --------(5)$$

4 | Proposed Method

Consider a linked network with s as the initial node and e as the final node. The objective is to determine the SP between source nodes s and e with the parameters associated with the vertices are NNs.

4.1 | The Modified neutrosophic Breadth first search Algorithm (MNBFSA)

Step 1: Take a graph $G = (V, E)$which is simple, weighted, and connected network with neutrosophic numbers as parameters for the edges. Let the source and destinations nodes be A and E, respectively.

Step 2: Then, using Breadth's first algorithm, find a different possible path from source to destination.

```
def bfs(source, destination, K):
    min_cost = dictionary representing min cost under K stops for each node
        reachable from source.
      set min_cost of source to be 0
      Q = queue()
      Q.push(source)
      stops = 0
      while Q is not empty {
           size = Q.size
           for i in range 1..size {
               element = Q.pop()
                 if element == destination then continue
                 for neighbor in adjacency list of element {
                 if stops == K and neighbor != destination   then continue
        if min_cost of neighbor improves, update and add back to the queue.
        # No need to update the minimum cost if we have already exhausted our
            K stops. if stops == K and neighbor != dst: continue
```

Step 3: Using canonical representation, we exchange neutrosophic path length values to crisp path length values.

Step 4: Finally, after getting the total path from source to destination, we apply a selection sort algorithm to find the minimum rank of the path on a given network.

5 | Numerical Illustration

Step 1: Consider the graph $G = (V, E)$as in the figure given below with the parameters associated with the edges are NNs and A and E, respectively represent the source and destination nodes.

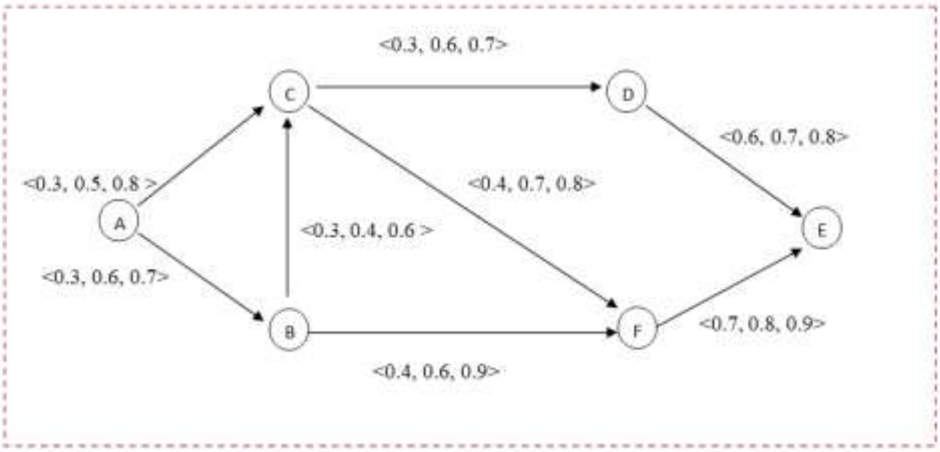


Figure 1: Connected network.

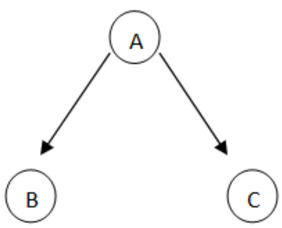
Table 1. (Arc length as neutrosophic number).

Arc Length	Edge value in neutrosophic number
A-B	<0.3, 0.6, 0.7>
A-C	<0.3, 0.5, 0.8 >
B-C	<0.3, 0.4, 0.6 >
B-F	<0.4, 0.6, 0.9>
C-D	<0.3, 0.6, 0.7>
C-F	<0.4, 0.7, 0.8>
D-E	<0.6, 0.7, 0.8>
F-E	<0.7, 0.8, 0.9>

Step 2: Now we apply BFS algorithm. Assume A is a source node and E is a Destination node

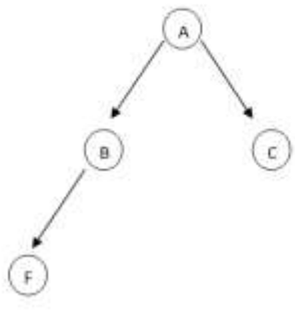
First step:

- A is the source node so it will be removed from the queue.
- A's neighbors are B and C, so it will be transited.
- B and C will be transited, which have not been transited previously.
- A Pushed to the back of the line B and C will be identified as visited.



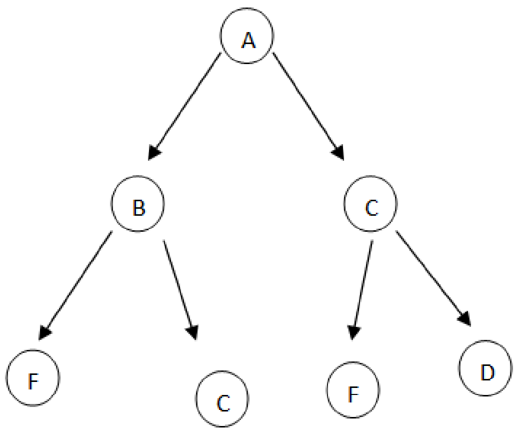
Second step:

- B is removed from the queue.
- The neighbors of B are C and F.
- C is disregarded as it has already been visited.
- F, which hasn't been explored previously, is now explored. It is then added to the queue and marked as visited.



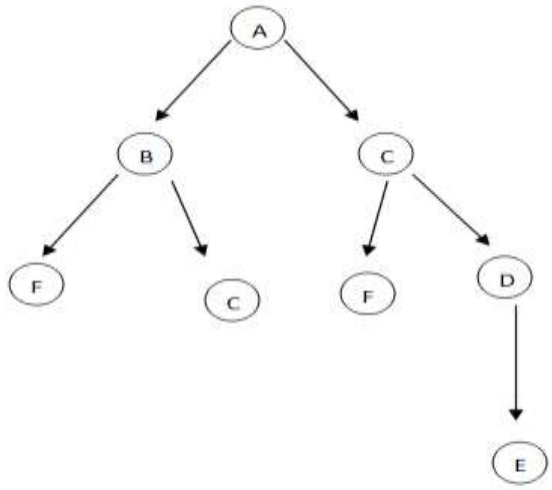
Third step:

- C is taken out of the queue.
- The neighbors of C are D and F.
- F is skipped since it has already been visited.
- D, which hasn't been explored yet, is now explored. It is then added to the queue and marked as visited.



Fourth step:

- D is taken out of the queue.
- The neighbor of D is E.
- E, which hasn't been traversed before, is now traversed. It is then added to the queue and marked as visited.



Fifth step:

- F is removed from the queue.
- The neighbor of F is E.
- E, is ignored because it is identified as 'visited'
- So in conclusion the possible path is from initial to final node is in fig: 2

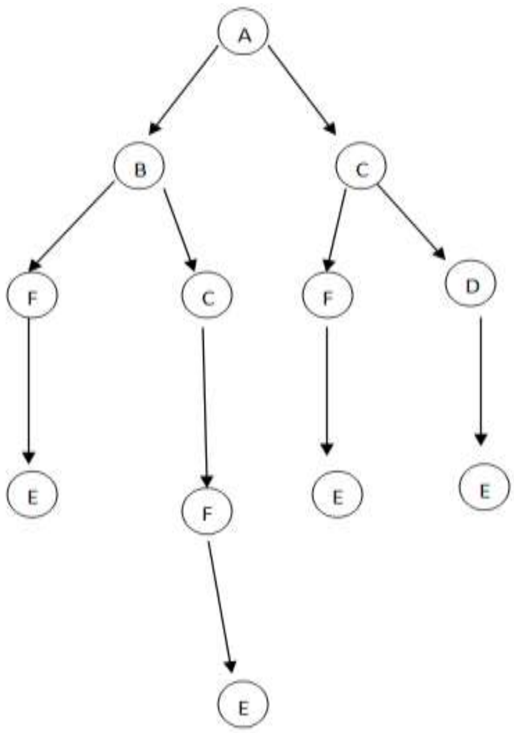


Figure 2.

Finally possible path from source to destination is in Table 2.

Table2. (Possible path).

Possible path
A-B-F-E
A-B-C-F-E
A-C-F-E
A-C-D-E

Step 3: Now by using canonical representation we find the Path distance

$$A - B - F - E = (A - B) + (B - F) + (F - E)$$

$$= \frac{1}{6}(0.3 + 4(0.6) + 0.7) + \frac{1}{6}(0.4 + 4(0.6) + 0.9) + \frac{1}{6}(0.7 + 4(0.8) + 0.9)$$

$$= \frac{1}{6}(0.3 + 2.4 + 0.7) + \frac{1}{6}(0.4 + 2.4 + 0.9) + \frac{1}{6}(0.7 + 3.2 + 0.9)$$

$$= \frac{1}{6}(3.4) + \frac{1}{6}(3.7) + \frac{1}{6}(4.8)$$

$$= 0.56 + 0.61 + 0.80$$

$$= 1.97$$

Path $A - B - C - F - E = (A - B) + (B - C) + (C - F) + (F - E)$

$$= \frac{1}{6}(0.3 + 4(0.6) + 0.7) + \frac{1}{6}(0.3 + 4(0.4) + 0.6) + \frac{1}{6}(0.4 + 4(0.7) + 0.8) + \frac{1}{6}(0.7 + 4(0.8) + 0.9)$$

$$= \frac{1}{6}(0.3 + 2.4 + 0.7) + \frac{1}{6}(0.3 + 1.6 + 0.6) + \frac{1}{6}(0.4 + 2.8 + 0.8) + \frac{1}{6}(0.7 + 3.2 + 0.9)$$

$$= \frac{1}{6}(0.3 + 2.4 + 0.7) + \frac{1}{6}(0.3 + 1.6 + 0.6) + \frac{1}{6}(0.4 + 2.8 + 0.8) + \frac{1}{6}(0.7 + 3.2 + 0.9)$$

$$= \frac{1}{6}(3.4) + \frac{1}{6}(2.5) + \frac{1}{6}(4.0) + \frac{1}{6}(4.8)$$

$$= 0.56 + 0.41 + 0.66 + 0.80$$

$$= 2.43$$

Path $A - C - F - E = (A - C) + (C - F) + (F - E)$

$$= \frac{1}{6}(0.3 + 4(0.5) + 0.8) + \frac{1}{6}(0.4 + 4(0.7) + 0.8) + \frac{1}{6}(0.7 + 4(0.8) + 0.9)$$

$$= \frac{1}{6}(0.3 + 2.0 + 0.8) + \frac{1}{6}(0.4 + 2.8 + 0.8) + \frac{1}{6}(0.7 + 3.2 + 0.9)$$

$$= \frac{1}{6}(3.1) + \frac{1}{6}(4.0) + \frac{1}{6}(4.8)$$

$$= 0.51 + 0.66 + 0.80$$

$$= 1.97$$

Path $A - C - D - E = (A - C) + (C - D) + (D - E)$

$$= \frac{1}{6}(0.3 + 4(0.5) + 0.8) + \frac{1}{6}(0.3 + 4(0.6) + 0.7) + \frac{1}{6}(0.6 + 4(0.7) + 0.8)$$

$$= \frac{1}{6}(0.3 + 2.0 + 0.8) + \frac{1}{6}(0.3 + 2.4 + 0.7) + \frac{1}{6}(0.6 + 2.8 + 0.8)$$

$$= \frac{1}{6}(3.1) + \frac{1}{6}(3.4) + \frac{1}{6}(4.2)$$

$$= 0.51 + 0.56 + 0.70$$

$$= 1.77$$

Step 4:
Input:

```
# Selection sort in Python 3 compiler
def selectionSort(array, size):

    for step in range(size):
        min_idx = step

        for i in range(step + 1, size):

            # to sort in descending order, change > to < in this line
            # select the minimum element in each loop
            if array[i] < array[min_idx]:
                min_idx = i

        # put min at the correct position
        (array[step], array[min_idx]) = (array[min_idx], array[step])


data = [1.97, 2.43, 1.97, 1.77]
size = len(data)
selectionSort(data, size)
print('Sorted the shortest path :')
print(data)
```

Output:

```
Sorted the shortest path :
[1.77, 1.97, 1.97, 2.43]
```

6 | Comparative Analysis between our Approach and the Existing Method

In this section, we compare our methodology with one other existing methodology and finally analyze our methodology for evaluating the shortest path based on the breadth first algorithm in a neutrosophic environment, which gives the optimal result. We discuss this in Table 3.

Table 3. Comparison of shortest path length.

Methods	Shortest path	shortest path length
"Shortest path problem within the framework of interval-valued neutrosophic settings [33]	A → C → D → E	[0.35,0.60],[0.01,0.04], [0.008,0.075]
Our proposed method	A → C → D → E	1.77

7 | Conclusions

In this paper, we determined the shortest path within a network operating in a neutrosophic environment, where edge weights neutrosophic number. The aforementioned paper elaborates on the benefits of employing neutrosophic numbers in NSP. The traditional breadth-first algorithm is modified to calculate the SP between a pair of given nodes. Here we illustrate the effectiveness of our suggested methodology by giving one numerical example. The most significant finding of this work is identifying an NSP algorithm in the neutrosophic environment that uses the NN as edge weights. The proposed algorithm is extremely effective for real-world application-oriented problems. In future work, one may explore different ways to resolve SPPs using and advancing our proposed algorithm.

Acknowledgments

The author is grateful to the editorial and reviewers, as well as the correspondent author, who offered assistance in the form of advice, assessment, and checking during the study period.

Author Contributaion

All authors contributed equally to this work.

Funding

This research has no funding source.

Data Availability

The datasets generated during and/or analyzed during the current study are not publicly available due to the privacy-preserving nature of the data but are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there is no conflict of interest in the research.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Zadeh, L. A. (1965). Fuzzy sets. Information and control, 8(3), 338-353.
[2] Smarandache, F. (1999). A unifying field in logics. neutrosophy: Neutrosophic probability, set and logic.
[3] Ali, M., & Smarandache, F. (2017). Complex neutrosophic set. Neural computing and applications, 28, 1817-1834.
[4] Ahuja, R. K., Mehlhorn, K., Orlin, J., & Tarjan, R. E. (1990). Faster algorithms for the shortest path problem. Journal of the ACM (JACM), 37(2), 213-223.
[5] Broumi, S., Nagarajan, D., Bakali, A., Talea, M., Smarandache, F., & Lathamaheswari, M. (2019). The shortest path problem in interval valued trapezoidal and triangular neutrosophic environment. Complex & Intelligent Systems, 5, 391-402.
[6] Deng, Y., Chen, Y., Zhang, Y., & Mahadevan, S. (2012). Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. Applied Soft Computing, 12(3), 1231-1237.
[7] Ye, J. (2015). Multiple-attribute decision-making method under a single-valued neutrosophic hesitant fuzzy environment. Journal of Intelligent Systems, 24(1), 23-36.
[8] Peng, J. J., & Wang, J. Q. (2015). Multi-valued neutrosophic sets and its application in multi-criteria decision-making problems. Neutrosophic Sets and Systems, 10(1), 6.
[9] Ye, J. (2015). Trapezoidal neutrosophic set and its application to multiple attribute decision-making. Neural Computing and Applications, 26, 1157-1166.
[10] Garg, H. (2016). An improved score function for ranking neutrosophic sets and its application to decision-making process. International Journal for Uncertainty Quantification, 6(5).
[11] Şahin, R., & Liu, P. (2016). Maximizing deviation method for neutrosophic multiple attribute decision making with incomplete weight information. Neural Computing and Applications, 27, 2017-2029.
[12] Deli, İ. (2019). A novel defuzzification method of SV-trapezoidal neutrosophic numbers and multi-attribute decision making: a comparative analysis. Soft Computing, 23(23), 12529-12545.
[13] Zhang, Y., Song, S., Shen, Z. J. M., & Wu, C. (2017). Robust shortest path problem with distributional uncertainty. IEEE transactions on intelligent transportation systems, 19(4), 1080-1090.
[14] Broumi, S., Talea, M., Bakali, A., Smarandache, F., & Kumar, P. K. (2017, May). Shortest path problem on single valued neutrosophic graphs. In 2017 international symposium on networks, computers and communications (ISNCC) (pp. 1-6). IEEE.
[15] Peng, X., & Dai, J. (2017). Algorithms for interval neutrosophic multiple attribute decision-making based on MABAC, similarity measure, and EDAS. International Journal for Uncertainty Quantification, 7(5).
[16] 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.
[17] Deli, I., & Şubaş, Y. (2017). A ranking method of single valued neutrosophic numbers and its applications to multi-attribute decision making problems. International Journal of Machine Learning and Cybernetics, 8(4), 1309-1322.
[18] Broumi, S., Bakali, A., Talea, M., Smarandache, F., Uluçay, V., Sahin, M., ... & Pramanik, S. (2018). Neutrosophic sets: An overview. Infinite Study.
[19] Broumi, S., Bakali, A., Talea, M., Smarandache, F., Kishore, K., & Şahin, R. (2016). Shortest path problem under interval valued neutrosophic setting. Infinite Study.
[20] Bolturk, E., & Kahraman, C. (2018). A novel interval-valued neutrosophic AHP with cosine similarity measure. Soft Computing, 22, 4941-4958.
[21] Wang, H., Smarandache, F., Sunderraman, R., & Zhang, Y. Q. (2005). interval neutrosophic sets and logic: theory and applications in computing: Theory and applications in computing (Vol. 5). Infinite Study.
[22] Biswas, P., Pramanik, S., & Giri, B. C. (2018). Distance measure based MADM strategy with interval trapezoidal neutrosophic numbers. Neutrosophic Sets and Systems, 19, 40-46.

[23] İrfan, D. E. L. İ. (2018). Expansions and reductions on neutrosophic classical soft set. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 22, 478-486.
[24] Deli, I. (2018). Operators on single valued trapezoidal neutrosophic numbers and SVTN-group decision making. Neutrosophic Sets and Systems, 22, 131-150.
[25] Deli, I., & Şubaş, Y. (2017). Some weighted geometric operators with SVTrN-numbers and their application to multi-criteria decision making problems. Journal of Intelligent & Fuzzy Systems, 32(1), 291-301.
[26] Abdel-Basset, M., Atef, A., & Smarandache, F. (2019). A hybrid Neutrosophic multiple criteria group decision making approach for project selection. Cognitive Systems Research, 57, 216-227.
[27] Abdel-Basset, M., Mohamed, M., & Sangaiah, A. K. (2018). Neutrosophic AHP-Delphi Group decision making model based on trapezoidal neutrosophic numbers. Journal of Ambient Intelligence and Humanized Computing, 9, 1427-1443.
[28] Kumar, R., Edaltpanah, S. A., Jha, S., Broumi, S., & Dey, A. (2018). Neutrosophic shortest path problem. Infinite Study.
[29] Broumi, S., Talea, M., Bakali, A., & Smarandache, F. (2016, December). Application of Dijkstra algorithm for solving interval valued neutrosophic shortest path problem. In 2016 IEEE symposium series on computational intelligence (SSCI) (pp. 1-6). IEEE.
[30] Tan, R. P., Zhang, W. D., & Broumi, S. (2019). Solving methods for the shortest path problem based on trapezoidal fuzzy neutrosophic numbers. Control Decis, 34(4), 851-860.
[31] Broumi, S., Nagarajan, D., Bakali, A., Talea, M., Smarandache, F., & Lathamaheswari, M. (2019). The shortest path problem in interval valued trapezoidal and triangular neutrosophic environment. Complex & Intelligent Systems, 5, 391-402.
[32] Goldfarb, D., Hao, J., & Kai, S. R. (1991). Shortest path algorithms using dynamic breadth-first search. *Networks*, *21*(1), 29-50.
[33] Broumi S, Bakali A, Talea M, Smarandache F, Kishore KK, Sahin R (2018) Shortest path problem under interval valued neutrosophic setting. J Fundam Appl Sci 10(4S):168–174