Neutrosophic Sets and Systems

Volume 68 *Neutrosophic Sets and Systems, Vol. 68, 2024*

Article 13

6-1-2024

# Evaluation of the shortest path based on the Traveling Salesman problem with a genetic algorithm in a neutrosophic environment

Prasanta Kumar Raut

Siva Prasad Behera

Said Broumi

Arindam Dey

Mohamed Talea

*See next page for additional authors*

Follow this and additional works at: https://digitalrepository.unm.edu/nss_journal

## Recommended Citation

Raut, Prasanta Kumar; Siva Prasad Behera; Said Broumi; Arindam Dey; Mohamed Talea; and Amarendra Baral. "Evaluation of the shortest path based on the Traveling Salesman problem with a genetic algorithm in a neutrosophic environment." *Neutrosophic Sets and Systems* 68, 1 (2024). https://digitalrepository.unm.edu/nss_journal/vol68/iss1/13

This Article is brought to you for free and open access by UNM Digital Repository. It has been accepted for inclusion in Neutrosophic Sets and Systems by an authorized editor of UNM Digital Repository. For more information, please contact disc@unm.edu.

# Evaluation of the shortest path based on the Traveling Salesman problem with a genetic algorithm in a neutrosophic environment


## Authors

Prasanta Kumar Raut, Siva Prasad Behera, Said Broumi, Arindam Dey, Mohamed Talea, and Amarendra Baral



This article is available in Neutrosophic Sets and Systems: https://digitalrepository.unm.edu/nss_journal/vol68/iss1/13

*Neutrosophic Sets and Systems, Vol. 68, 2024*
*University of New Mexico*

# Evaluation of the shortest path based on the Traveling Salesman problem with a genetic algorithm in a neutrosophic environment

**Prasanta Kumar Raut[1,*], Siva Prasad Behera[1], Said Broumi[2,5], Arindam Dey[3], Mohamed Talea[2], Amarendra Baral[4]**

[1]Department of Mathematics, C.V. Raman Global University, Bhubaneswar-752054, Odisha, India
[2]Laboratory of Information Processing, Faculty of Science Ben M'Sik, University Hassan II, Casablanca, Morocco
[3] School of Computer Science and Engineering, VIT-AP University, Inavolu, Beside AP Secretariat, Amaravati AP, India
[4]Trident Academy of Technology, Bhubaneswar, Odisha, India
[5]Department of Mathematics Saveetha School of Engineering, SIMATS Thandalam, Chennai – 602105 Tamilnadu, India

[1]Email:prasantaraut95@gmail.com, sivaiitkgp12@gmail.com, [2]Email: broumisaid78@gmail.com , taleamohamed@yahoo.fr

[3]Email arindam84nit@gmail.com, [4]Email: deanssh@tat.ac.in
* Correspondence prasantaraut95@gmail.com;

**Abstract:** In The traveling salesman problem (TSP) is an essential and the most popular conventional combinatorial optimization network problem in operations research, in which the TSP evaluates the shortest route or path in a network. In TSP, every node has been visited only once, excluding the starting node. In TSP, edge lengths are usually expressed to indicate journey time and expenses instead of distance from a location. The exact arc length can't be predicted because journey times and expenses vary depending on the amount of payload, climate, highway conditions, and so on. As a result, the Neutrosophic numbers introduce a new tool for dealing with unpredictability in TSP. The present article addresses TSP on a neutrosophic network where the edge weight is a neutrosophic number rather than a real number. For solving the Neutrosophic TSP, an algorithmic technique based on the genetic algorithm (GA) is proposed. We created a new mutation and crossover for our suggested GA. We used mathematical examples to show the usefulness of the algorithm that we suggested. The results of experiments suggest that the proposed GA can find the shortest path in a TSP within a neutrosophic framework. This provides valuable insights for decision-makers dealing with real-world situations characterized by imprecise and incomplete data.

**Keywords:** Connected network; Neutrosophic number; Shortest path problem; Traveling Salesman problem

## 1. Introduction

The traveling salesman problem (TSP) is one of the most essential and extensively researched systems. TSP was primarily introduced in 1930 and became extremely popular after 1950 [1–3]. Even in the past few years, optimization problems have appeared in the field of engineering research, and many papers have been published related to optimization [4-11]. The selection maker in the TSP discovers the shortest possible route or path for the salesman who visits every single node (city) exactly once (except the initial city) and returns to the initial node (city). It's a well-known optimization problem [12]. In practice, the edge cost in a traffic network path [13–16] may have

various parameters that are difficult to determine precisely (e.g., travel cost, travel time, road capacity, traffic frequency, and so on).

The decision-maker has to consider the uncertainty in travel costs and time because, depending on the climate, road conditions, expense, and time at which travel may vary, As a result, in such real-life scenarios, decision-makers cannot precisely determine the parameters (traveling time, traveling speed, and traveling cost) between two separate cities but appropriately determine the TSP. As a result, fuzzy sets can be used to deal with this type of uncertainty [17-19], allowing the decision-maker to make decisions based on uncertain data. The cost and duration of travel between two cities are primarily determined by the mode of transportation used. It generally changes daily, and experts can regard the travel costs and time of this problem as ambiguous [20- 22]. Several efforts have been made to solve the fuzzy traveling salesman problem [23–27].

The TSP [3] belongs to the complete NP problem. But it will be simple to understand but extremely difficult to solve. The computation duration for the TSP grows steadily whenever the total number of places increases. The TSP is commonly used as an example of an optimization problem to show the efficiency of a newly developed approach. Numerous heuristic and metaheuristic approaches are available to solve the TSP, including the genetic algorithm (GA), the harmony searching algorithm, and the artificial bee colony algorithms. A genetic algorithm is a type of heuristic optimization algorithm in which a chromosome describes a possible solution to a given optimization problem. The population is built from a variety of chromosomes. Chromosomes are reassembled in this algorithm to create new chromosomes. This recombination method is carried out primarily through three biological operations, i.e., selection, mutation, and crossover. The GA is used in this method to find the best solutions. The GA can solve many optimization problems [28, 29]. It is also employed in the solution of the TSP [30–35].

In this research paper, we have proposed a modified genetic algorithm for finding the shortest path using TSP in a neutrosophic environment. Initially, the GA generated by Darwin's law, primarily used for traditional TSPs when arc lengths are crisp numbers and the environment is certain, does not particularly apply to an uncertain environment. As a result, in this paper, we enhance the GA using an aspect of neutrosophic set theory. Here, the neutrosophic number represents the TSP's arc length. A mathematical model is introduced for a TSP with neutrosophic numbers as arc lengths. We present the utility of neutrosophic sets as arc lengths for TSP. We have updated our suggested GA with a new crossover as well as a mutation. We have demonstrated the effectiveness of our suggested algorithm with a numerical example. The other parts of this research paper are prepared in the following ways: Section-2 covers the fundamentals of neutrosophic sets (NSs). Section-3 proposes a GA for finding the shortest path using TSP in a neutrosophic environment. In Section-4, we present an improved pseudo-code of a GA for determining the shortest path (SP) of a connected network with respect to TSP. In Section-5 we presented a numerical example. In Section-6, we find the best route using TSP in the neutrosophic environment. In Section-7, we compare our method with other existing methods. Section-8 contains the conclusions and recommendations for additional research.

## 2. Preliminaries

**Definition-2.1**

Assume set $\ddot{X}$ is the universal set. An intuitionistic fuzzy set $\ddot{A}$ in $\ddot{X}$is written in the form: $\ddot{A} = \{\ddot{x}, \mu_{\ddot{A}}(x), \nu_{\ddot{A}}(\ddot{x})$

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 $\ddot{A}\epsilon\ \ddot{X}$, respectively, and for everyx $\epsilon\ \ddot{X}$, $\mu_{\ddot{A}}(x), \nu_{\ddot{A}}(\ddot{x}) \leq 1$.

**Definition-2.2**

Assume $\ddot{X}$ is a set of space points (objects), and $\tilde{x}$ represents the associated generic elements in $\ddot{X}$; then the element in neutrosophic set $\widetilde{A}$ has the form

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

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 $[0^-, 1^+]$ hence it's challenging to implement NSs to real-world situations.

**Definition-2.3**

Euclidean Distance: This is the most commonly used distance measure for neutrosophic numbers. It is a generalization of the Euclidean distance between two points in a multi-dimensional space. For neutrosophic numbers, you can calculate the Euclidean distance by treating each component ($\ddot{T}$, $\ddot{I}$, $\ddot{F}$) as a separate coordinate and using the standard Euclidean distance formula:

$$\text{Euclidean Distance} = \sqrt{(\ddot{T}_1 - \ddot{T}_2)^2 + (\ddot{I}_1 - \ddot{I}_2)^2 + (\ddot{F}_1 - \ddot{F}_2)^2}$$

The representation of cities as ($\ddot{T}$, $\ddot{I}$, $\ddot{F}$) in the framework of the TSP with a GA in a neutrosophic environment may have particular significance that corresponds with neutrosophic principles:

$\ddot{T}$ (truth): In a neutrosophic environment, T may represent a city's truth value. Neutrosophic is a philosophy that deals with indeterminacy and has three principles: truth ($\ddot{T}$), indeterminacy ($\ddot{I}$), and falsity ($\ddot{F}$). A city's "truth" value may indicate how precisely its location or characteristics have been identified.

$\ddot{I}$ (indeterminacy): The degree of indeterminacy or uncertainty associated with a city is represented by $\ddot{I}$. It reflects the degree of ambiguity or imprecision in the city's available information. $\ddot{I}$ denote the level of unknown or partially known information in a neutrosophic context.

$\ddot{F}$ (falsity): The degree of falsity or incorrectness associated with a city is represented by $\ddot{F}$. This value indicates how much of the available information about the city is deceptive, in error, or incorrect.

When calculating distances and making decisions in the genetic algorithm, using ($\ddot{T}$, $\ddot{I}$, $\ddot{F}$) For city representation in a TSP within a neutrosophic environment, it allows for the consideration of uncertainties and ambiguities. It recognizes that information about each city may not be entirely

true or false, but may contain varying degrees of truth, indeterminacy, and falsity, making the TSP solution more adaptable to such uncertainties.

## 3. Proposed GA for finding the shortest path using TSP

A genetic algorithm [36, 27] is an optimization algorithm inspired by natural selection. They are used to solve difficult optimization and search problems. The following are the typical steps in a GA:

**Initialization:**

Begin the NTSP's population of potential solutions with neutrosophic number representations.

**Selection:**

Evaluate the fitness of every member of the population. The fitness function assesses how well each person eliminates the problem.Choose people from the population to be the next generation's parents. Individuals with higher fitness values are more likely to be chosen, but some diversity should also be preserved.

**Crossover (Recombination):**

Take two carefully chosen parents and combine their genetic information to produce one or more offspring. Crossover occurs when parts of the parent chromosomes are exchanged or combined to create new individuals. Crossover methods, such as single-point, two-point, or uniform crossover, can vary.

**Mutation (Mutate (child)):**

Change some of the genetic information of the offspring at random. Mutation adds diversity to the population and keeps it from becoming stuck in local optima.

**Replacement (Replace Weakest (child1, child2)):**

In order to create a new population, combine the parent individuals and their progeny. You can also decide to use selection mechanisms (elitism, for example) to determine which members of the previous generation are retained in the new population, ensuring that the best solutions are not lost.

**Termination:**

Verify the termination terms, which other factors may determine, the number of generations, or the highest level of fitness attained. The algorithm terminates if the termination condition is satisfied; if not, return to the selection stage.

**Result:**

Once the algorithm terminates, the best solution found in the final population is the output of the genetic algorithm.

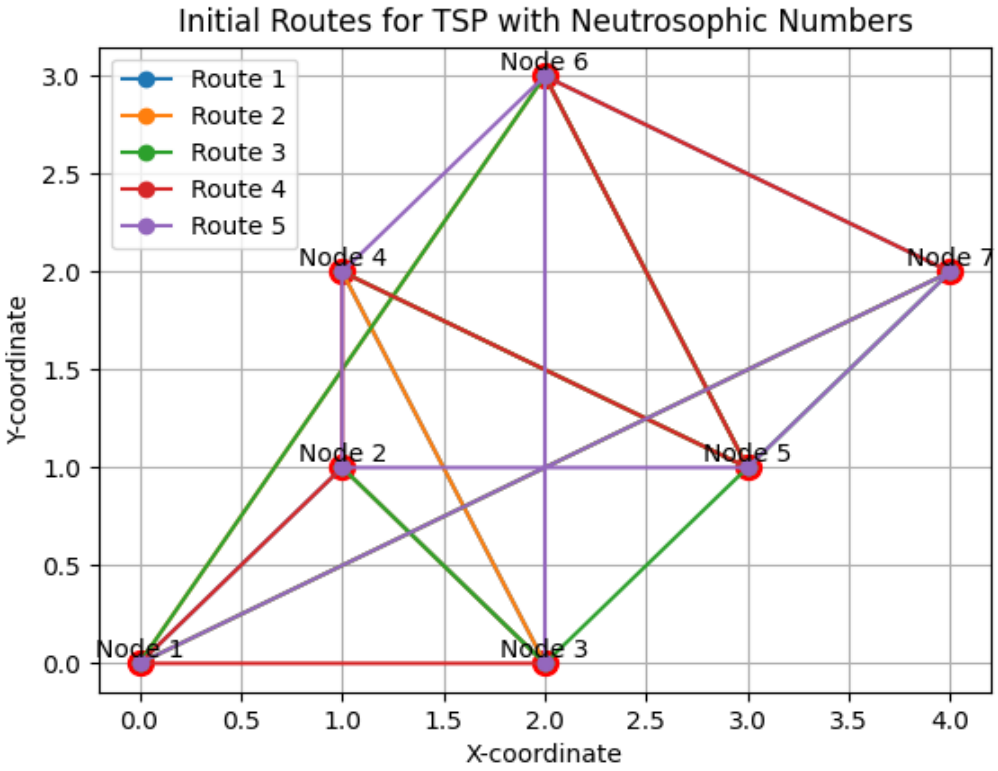
## 4. Pseudo code of genetic algorithm

Initialize Population ( )
best solution = null
best fitness = +infinity
generation = 0

```
while generation < max generations:
    fitness values = Evaluate Population()

    For i from 1 to population size by 2:
        parent1 = Select Parent()
        parent2 = Select Parent()
        child1, child2 = Crossover(parent1, parent2)
        child1 = Mutate(child1)
        child2 = Mutate(child2)
        Replace Weakest(child1, child2)

    best individual, best individual fitness = Get Best Individual()

    If best individual fitness < best fitness:
        best solution = best individual
        best fitness = best individual fitness

    generation = generation + 1

return the best solution
```

**5. Numerical example:**

In this example, we'll consider a small TSP with 7 cities, and we'll represent the cities using neutrosophic numbers in a simplified format, where each city is represented as ($\ddot{T}$, $\ddot{I}$, $\ddot{F}$) The values for $\ddot{T}$ (truth-membership), $\ddot{I}$ (indeterminacy), and $\ddot{F}$ (falsity-membership) range from 0 to 1.




**Initialization:**

In this step, we generate an initial population shown in Table-1 and we create an initial population of 5 routes in a 7-node network. Where each route represents a permutation of the 7 cities in Table 2. Each city is represented as a neutrosophic number (Ṫ, Ï, Ḟ).

Table-1. Here's a set of 7 cities with their neutrosophic number

| City | Neutrosophic number |
|---|---|
| 1 | (0.7,0.2,0.1) |
| 2 | (0.6, 0.3, 0.1) |
| 3 | (0.5, 0.3, 0.2) |
| 4 | (0.2, 0.5, 0.3) |
| 5 | (0.4, 0.4, 0.2) |
| 6 | (0.3, 0.4, 0.3) |
| 7 | (0.6, 0.3, 0.1) |

Table 2. All possible routes

| Route no | Possible route |
|---|---|
| 1 | [1, 2, 3, 4, 5, 6, 7] |

| | |
|---|---|
| 2 | [2, 3, 4, 5, 6, 1, 7] |
| 3 | [4, 5, 6, 1, 7, 3, 2] |
| 4 | [3, 1, 2, 4, 5, 6, 7] |
| 5 | [1, 7, 5, 2, 4, 6, 3] |

**Step 2: Evaluation (Fitness):**

We calculate the fitness of each route in the population using a fitness function. In this example, the fitness function takes into account the neutrosophic numbers and aims to minimize the total distance while considering indeterminacy:

$$\text{Fitness} = \sum \frac{(T_i * (distance_i))}{(I_i + 1)}$$

Where:

$T_i$ is the truth-membership of the node's highest neutrosophic number.

$I_i$ is the indeterminacy of the node's minimum neutrosophic number.

$(distance_i)$ is the distance between two consecutive nodes in the route.

**Fitness for Route 1 :( using definition-2.3)**

| **Distance between node** | **Crisp value** |
|---|---|
| 1-2 | 0.14 |
| 2-3 | 0.14 |
| 3-4 | 0.37 |
| 4-5 | 0.24 |
| 5-6 | 0.14 |
| 6-7 | 0.37 |

Total Distance =0.14+0.14+0.37+0.24+0.14+0.37=1.4

$$\text{Fitness} = \sum \frac{(0.7 * 1.4)}{(0.1 + 1)}$$

$$\text{Fitness} = \sum \frac{(0.7 * 1.4)}{(0.1 + 1)} = 0.89$$

Similarly find the fitness for Route 2, Route 3, Route 4, and Route 5

| Fitness function of Route | Minimize distance |
|---|---|
| Route 1: | 0.89 |
| Route 2: | 0.96 |

| | |
|---|---|
| Route 3: | 0.88 |
| Route 4: | 0.86 |
| Route 5: | 0.94 |

**Step 3: Selection (Choose Two Routes):**

Use a selection mechanism (e.g., roulette wheel selection or tournament selection) to choose routes to become parents for the next generation. Routes with lower fitness values have a higher chance of being selected.

Select the two routes with the lowest fitness values:

Route 4 (Fitness: 0.86)

Route 3 (Fitness: 0.88)

**Step 4: Crossover (Order Crossover):**

Apply crossover (recombination) to pairs of selected routes to create new routes (offspring). The crossover should respect the neutrosophic number representation.

Combine genetic information from Route 4 and Route 3:

Parent Route 3: [4, 5, 6, 1, 7, 3, 2]

Parent Route 4: [3, 1, 2, 4, 5, 6, 7]

Crossover Point (e.g., after Node 1):

Offspring Route: [4, 5, 6, 1| 2, 7, 3]

Complete the offspring by adding the missing nodes while avoiding duplicates:

Offspring Route: [4, 5, 6, 1, 2, 7, 3]

**Step 5: Mutation (Swap Two Random Nodes):**

Apply mutation operators to introduce small random changes in the routes. The mutation should also respect the neutrosophic numbers.

Let's swap nodes 1 and 7:

Mutated Offspring Route: [4, 5, 6, 7, 2, 1, 3]

**Step 6: Replacement (Replace One of the Parents):**

Replace some of the old routes with the newly created offspring to form the next generation.

Replace one of the parents (Route 3) with the mutated offspring:

Updated Population:

Route 1: [4, 5, 6, 3, 2, 1, 7]

Route 2: [1, 2, 3, 4, 5, 7, 6]

Route 3: [4, 5, 6, 1, 7, 3, 2]

Route 4: [4, 5, 6, 7, 2, 1, 3]

Route 5: [1, 7, 5, 2, 4, 6, 3]

**Step 7: Termination (End of One Generation):**

Choose a termination condition, such as reaching a certain number of generations or achieving a satisfactory fitness value. These steps would be repeated for several generations, with each generation improving the routes. Termination can be based on reaching a certain number of generations or a satisfactory fitness value. This example shows one generation of a GA for the TSP in a 7-node network with neutrosophic numbers. More sophisticated fitness functions and genetic operators would be used in practice.

**6. Result: The best route found in the final population is the solution to the TSP with neutrosophic numbers.**

Table 3. The Optimal result of shortest path

| **Solution using Linear programming In lingo software** | **Solution using Proposed Genetic algorithm for finding shortest path using TSP** |
|---|---|
| Minz=0.86 | The cost of NTSP=0.86<br>And shortest route=3 →1→ 2→ 4→ 5→ 6→7 |

Our suggested GAs are implemented to identify the shortest path of a neutrosophic graph's traveling salesman problem. In example 1 we take 7 nodes (Figure 1) and 5 possible paths (Table 2). We have arbitrarily chosen a possible path. Table 1 contains a description of the neutrosophic number as edge weights. This is the new concept we apply for neutrosophic numbers in the Travelling salesman problem using a genetic algorithm determining the shortest network path. In example 1, we find the optimal result of the shortest path in lingo software and our proposed method. In both cases, we got the same cost and the same shortest path in Table 3.

**7. Comparison of our methodology with other methodology**

In this section, we compare our methodology with some other existing methodologies and finally analyze our methodology for evaluating the shortest path based on the Traveling Salesman problem with a GA in a neutrosophic environment, which gives the optimal result. We discuss this in Table-4.

Table 4. Comparison of shortest path with the shortest path cost in crisp number

| **Methodology** | **Shortest path** | **shortest path cost in crisp number** |
|---|---|---|
| Traveling Salesman problem in Neutrosophic environment [38] | 3 →1→ 2→ 4→ 5→ 6→7 | The cost of NMST= 1.02 |
| Travelling salesman problem using fuzzy environment[39] | 3 →1→2→4→5→ 6→7 | 1.86 |
| Our proposed method in Neutrosophic environment | 3 →1→2→4→5→ 6→7 | 0.86 |

**8. Conclusion:**

The results of our experiments suggest that the proposed GA can find the shortest path in a TSP within a neutrosophic framework. This provides valuable insights for decision-makers dealing with real-world situations characterized by imprecise and incomplete data. Furthermore, this research opens up avenues for future exploration, such as refining the GA parameters and techniques for handling larger and more complex instances of the TSP in neutrosophic environments. Additionally, evaluating the algorithm's performance on different types of neutrosophic data and its potential integration into decision support systems could be areas for further investigation. This research serves as a valuable contribution to the field of operations research and optimization, with the potential to enhance decision-making in practical, real-world applications

**Conflicts of Interest:** The authors declare no conflict of interest.

## References


1. Bellman, R. (1962). Dynamic programming treatment of the travelling salesman problem. Journal of the ACM (JACM), 9(1), 61-63.
2. Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the travelling salesman problem. biosystems, 43(2), 73-81.
3. Langevin, A., Soumis, F., & Desrosiers, J. (1990). Classification of travelling salesman problem formulations. Operations Research Letters, 9(2), 127-132.
4. Broumi, S., krishna Prabha, S., & Uluçay, V. (2023). Interval-valued Fermatean neutrosophic shortest path problem via score function. Neutrosophic Systems with Applications, 11, 1-10.
5. Raut, P. K., Behera, S. P., Broumi, S., & Mishra, D. (2023). Calculation of shortest path on Fermatean Neutrosophic Networks. Neutrosophic Sets and Systems, 57(1), 22.
6. Broumi, S., Raut, P. K., & Behera, S. P. (2023). Solving shortest path problems using an ant colony algorithm with triangular neutrosophic arc weights. International Journal of Neutrosophic Science, 20(4), 128-28.
7. Raut, P. K., Behera, S. P., & Pati, J. K. (2021). Calculation of Shortest Path in a Closed Network in Fuzzy Environment. International Journal of Mathematics Trends and Technology-IJMTT, 67.
8. Jeyaraman, M., Iswariya, S., & Pandiselvi, R. (2023). Generalized double statistical convergence sequences on ideals in neutrosophic normed spaces. Neutrosophic Systems with Applications, 8, 50-60.
9. Mohamed, M., & El-Saber, N. (2023). Toward Energy Transformation: Intelligent Decision-Making Model Based on Uncertainty Neutrosophic Theory. Neutrosophic Systems with Applications, 9, 13-23.
10. Raut, P. K., Behera, S. P., Broumi, S., & Mishra, D. (2023). Calculation of Fuzzy shortest path problem using Multi-valued Neutrosophic number under fuzzy environment. Neutrosophic Sets and Systems, 57(1), 24.
11. Raut, P. K., & Behera, S. P. (2021). Application of Fuzzy Optimal Path Algorithm for Bus Route Expansion in Bhubaneswar city, Odisha. International Journal of Research Culture Society Issn, 5(11).
12. Zheng, W., Qiao, J., Feng, L., & Fu, P. (2018). Optimal cooperative virtual multi-input and multi-output network communication by double improved ant colony system and genetic algorithm. IAENG International Journal of Computer Science, 45(1), 89-96.
13. Yang, L., Li, B., & Xu, H. (2018). Novel power aggregation operatoroperators based on einstein operations for interval neutrosophic linguistic sets. IAENG International Journal of Applied Mathematics, 48(4), 1-10.
14. Asghar, M. Z., Subhan, F., Ahmad, H., Khan, W. Z., Hakak, S., Gadekallu, T. R., & Alazab, M. (2021). Senti-eSystem: a sentiment-based eSystem-using hybridized fuzzy and deep neural network for measuring customer satisfaction. Software: Practice and Experience, 51(3), 571-594.
15. Yang, L., & Li, B. (2018). An Extended Si nglengle-valued Neutrosophic Normalized Weighted Bonferroni Mean Einstein Aggregation Operator. Infinite Study.

16. Jhaveri, R. H., Desai, A., Patel, A., & Zhong, Y. (2018). A sequence number prediction based bait detection scheme to mitigate sequence number attacks in MANETs. Security and Communication Networks, 2018, 1-13.
17. Chen, X. E., & Liu, S. (2018). Adjacent Vertex Distinguishing Proper Edge Colorings of Bicyclic Graphs. IAENG International Journal of Applied Mathematics, 48(4).

Received: Mar 5, 2024. Accepted: May 29, 2024