Neutrosophic Sets and Systems, Vol. 64, 2024
University of New Mexico

Evaluation of Shortest path on multi stage graph problem using Dynamic approach under neutrosophic environment

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

[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]Trident Academy of Technology, Bhubaneswar, Odisha, India

[1]Email:prasantaraut95@gmail.com,sivaiitkgp12@gmail.com

[2]Email: broumisaid78@gmail.com

[3]Email: deanssh@tat.ac.in

Correspondence:sivaiitkgp12@gmail.com

Abstract

The shortest path problem is a classic optimization problem in graph theory and computer technology. It involves identifying the shortest path between two nodes in a graph, where each edge has a numerical weight. In this paper, we put our effort into examining the use of the dynamic programming method to evaluate the shortest path (SP) between the two specified nodes in a multistage network where the parameter is a multi-value neutrosophic number (MVNN). Firstly, we propose an algorithm based on the forward and backward approach in an uncertain environment and also implement our approach in the Python-3 programming language. Furthermore, a numerical illustration has been provided to showcase the effectiveness and robustness of the novel model.

Keyword: dynamic programming approach; multistage graph; neutrosophic multi-value number; shortest path problem

1. Introduction:

The shortest path problem represents one of the primary network issues in graph theory, with numerous applications in computer science and several real-life applications such as transportation networks, communication networks, and pipeline distribution systems. In this paper, we propose a new idea for evaluating the shortest path of a multistage graph using fuzzy multivalued neutrosophic numbers as arc length.

A fuzzy set (FS) is used to identify and solve a wide range of real-world issues that involve uncertainty and improbability. Lotfi Aliasker Zadeh first suggested the fuzzy set [1], and then Atanassov (1988) [2] proposed intuitionistic fuzzy sets (IFs), which are the extended concept of fuzzy sets. Then Smarandache (1995) [3] first established the theme of the new idea of neutrosophic sets (NS). The NS is a collection of three parameters, namely fuzzy membership degree, fuzzy indeterminate degree, and fuzzy non-membership degree, with the addition of their weights being less than or equal to 3. The field of neutrosophic numbers extends beyond crisp numbers. Numerous research papers have addressed the computation of the fuzzy shortest path (FSP) in a single-stage network. For example, Wang proposed the idea of IVNS (2018) by generalizing SVN (2010) [4]. The IVNS [5] is a database that generalizes the idea of various types of sets in terms of intervals to denote the truth T, falsity F, and indeterminacy I of membership degrees. Many researchers have proposed various papers on neutrosophic environments (Basset (2018), Abdel-Basset (2018), and Dey (2019)) [6–14].

Many researchers have proposed new approaches for finding SPP in uncertain environments. Das and De (2015) [15] solved FSP using Bellman's dynamic programming method with intuitionistic fuzzy trapezoidal numbers as parameters. Bhincher and De (2011) [16] investigated the FSP in a connected network in which they used triangular and trapezoidal fuzzy numbers as parameters in two distinct approaches, namely the influential programming approach and the multi-objective linear programming approach. Kumar (2015) [17] developed a technique for determining the SP of a connected network using an interval intuitionistic trapezoidal number. Kaliraja and Meenakshi (2012) [18] used interval-based parameters and proposed a method to identify the SPP and model an interval-valued FSPP.

Said Broumi (2016) [19] proposed a new idea of evaluating the shortest path using the parameters SV-triangular and SV-trapezoidal fuzzy neutrosophic numbers. Then again, Said Broumi (2017) [20] suggested a new idea to evaluate the FSPP of a given connected network with neutrosophic trapezoidal numbers. Said Broumi (2017) [21] suggested an innovative method for formulating the SPP in which they use the parameters, which are bipolar neutrosophic numbers. Deivanayagam Pillai, N (2020) [22], solved the NSPP by using the score function, where the parameters are interval-valued neutrosophic trapezoidal and neutrosophic triangular numbers. Said Broumi (2019) [23] solved the SPP in a neutrosophic environment (NS) using the Bellman-Ford approach, where the parameter is interval-valued neutrosophic numbers(IVNNs).

The primary aim of this study is to determine the shortest path between the source node and the destination node using multi-value neutrosophic numbers, along with identifying the minimum cost between the source and destination nodes. The contents of the next parts of the paper are arranged in the following manner: Section-2, highlights the motivation and contribution of this paper. Section-3 highlights some definitions of some of the existing terminologies. Section-4 highlights the algorithm, i.e., the multistage network, for multi-valued neutrosophic numbers (MVNNs). Section-5 highlights a numerical example. Section-6 gives an implementation of our algorithm with the Python programming language. Section-7 provides a summary of the conclusions drawn from the study and offers recommendations for further research endeavors.

2. Motivation:

There are various algorithms and various parameters that are used to evaluate the SPP in uncertain circumstances. The key points are as follows:

- There are many methods used to solve the single-stage network, but our method is used to solve the multistage network in NSP.
- In this paper, a dynamic programming method is used to evaluate the shortest path (SP) between the two specified nodes in a multistage network, where the parameter is a multi-value neutrosophic number (MVNN). Firstly, we propose an algorithm based on the forward and backward approach in an uncertain environment and also implement our approach in the Python-3 programming language.

- In this paper, we are finding the minimum cost between the source node and the destination node.
- Moreover, here we illustrate one algorithm with the help of a numerical example.

3. Preliminaries

This section encompasses the review of literature concerning the fundamental concepts and definitions of fuzzy sets (FSs), neutrosophic sets (NSs), and MVNSs.

3.1 Fuzzy set (FS):

If $\breve{Z}$ is a generalised form of crisp set and $\breve{z}$ is a member of $\breve{Z}$, then fuzzy set $\breve{A}$ on $\breve{Z}$ is defined by a membership value $\mu_{\breve{A}}(\breve{z})$, which identifies the function that maps from every element to the interval [0, 1] and can be defined as

$A = \{(\breve{z}, \mu_{\breve{A}}(\breve{z})), \breve{z} \in \breve{Z},\}$

and $\mu_{\breve{Z}}(\breve{z}): \breve{z} \rightarrow [0,1]$

3.2 Neutrosophic set (NS):

If $\breve{X}$ is a set and $\breve{x}$ is one of its elements in $\breve{X}$; then neutrosophic set $\breve{A}$ has the form

$$\breve{A} = \{< \breve{x}: \breve{T}_{\breve{A}}(x), \breve{I}_{\breve{A}}(x), \breve{F}_{\breve{A}}(x) > \breve{x}\epsilon\breve{X}\} - - - - - - - - - - (1)$$

Where $\breve{T}$ denotes the truth degree, $\breve{I}$ denotes the indeterminacy degree and $\breve{F}$ denotes the falsity membership degree of the element $\breve{x}\epsilon\breve{X}$

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

Now $\breve{T}_{\breve{A}}(x), \breve{I}_{\breve{A}}(x), \breve{F}_{\breve{A}}(x)$ are denotes subsets of the interval $[0^-, 1^+]$.

3.3 Multi-valued neutrosophic set (MVNs):

If $\breve{X}$ is a set and $\breve{x}$ is one of its elements in $\breve{X}$. Then the multi-valued neutrosophic (MVN) set is represented as.

$\breve{A} = \breve{x}, \breve{T}_{\breve{A}}(x), \breve{I}_{\breve{A}}(x), \breve{F}_{\breve{A}}(x), \breve{x}\epsilon\breve{X}$

Then $\breve{T}_{\breve{A}}(x), \breve{I}_{\breve{A}}(x)$ and $\breve{F}_{\breve{A}}(x)$ are the membership function differentiate $\breve{A}$ in $\breve{X}$.

Where $\breve{T}_{\breve{A}}(x), \breve{I}_{\breve{A}}(x)$ and $\breve{F}_{\breve{A}}(x) \in [0,1]$ and the condition is

$0 \leq \alpha, \beta, \gamma \leq 1, 0 \leq \alpha^+, \beta^+, \gamma^+ \leq 3, \alpha \in \breve{T}_{\breve{A}}(x), \beta \in \breve{I}_{\breve{A}}(x), \gamma \in \breve{F}_{\breve{A}}(x).$

$\alpha^+ = \text{Sup}\breve{T}_{\breve{A}}(x), \beta^+ = \text{Sup}\,\breve{I}_{\breve{A}}(x), \gamma^+ = \text{Sup}\breve{F}_{\breve{A}}(x) - - - - - - - - - (2)$

The multi-valued neutrosophic (MVN) are called as single valued neutrosophic (SVN) sets if $\breve{A} = \{\breve{T}_{\breve{A}}(x), \breve{I}_{\breve{A}}(x), \breve{F}_{\breve{A}}(x)\}$ has just one value.

3.4 Operations of Neutrosophic number:

Assume that $\breve{A}_1 = \{\breve{T}_{\breve{A}_1}(x), \breve{I}_{\breve{A}_1}(x), \breve{F}_{\breve{A}_1}(x)\}$ and $\breve{A}_2 = \{\breve{T}_{\breve{A}_2}(x), \breve{I}_{\breve{A}_2}(x), \breve{F}_{\breve{A}_2}(x)\}$ are

represent two sets of neutrosophic numbers with multiple values. Subsequently, the functions for SVNNs are defined as follows:

(a) $(\breve{A}_1 + \breve{A}_2) = \{\breve{T}_{\breve{A}_1}(x) + \breve{T}_{\breve{A}_2}(x) - \breve{T}_{\breve{A}_1}(x)\breve{T}_{\breve{A}_2}(x), \breve{I}_{\breve{A}_1}(x)\breve{I}_{\breve{A}_2}(x), \breve{F}_{\breve{A}_1}(x)\breve{F}_{\breve{A}_2}(x)\}$

(b)

$(\breve{A}_1 \times \breve{A}_2) = \{\breve{T}_{\breve{A}_1}(x)\breve{T}_{\breve{A}_2}(x), \breve{I}_{\breve{A}_1}(x) + \breve{I}_{\breve{A}_2}(x) - \breve{I}_{\breve{A}_1}(x)\breve{I}_{\breve{A}_2}(x), \breve{F}_{\breve{A}_1}(x) + \breve{F}_{\breve{A}_2}(x) - \breve{F}_{\breve{A}_1}(x)\breve{F}_{\breve{A}_2}(x)\}$

(c) $\lambda\breve{A}_1 = \{1(1 - \breve{T}_{\breve{A}_1}(x))^{\lambda}, \breve{I}_{\breve{A}_1}(x)^{\lambda}, \breve{F}_{\breve{A}_1}(x)^{\lambda}\}$

(d) $\breve{A}_1^{\lambda} = \{\breve{T}_{\breve{A}_1}(x)^{\lambda}, 1 - (1 - \breve{I}_{\breve{A}_1}(x))^{\lambda}, 1 - (1 - \breve{I}_{\breve{A}_1}(x))^{\lambda}\}$

With $\lambda > 0$

3.5 Fuzzy Graded mean Integration:

If the fuzzy triangular number $\breve{A} = (\breve{l}, \breve{m}, \breve{n})$ .Then the Fuzzy graded mean

integration is expressed as:

$G(\breve{A}) = \frac{1}{6}(\breve{l}_1 + 4\breve{m} + \breve{n}) - - - - - - - - - - - - (3)$

If $\breve{A} = (\breve{l}_1, \breve{m}_1, \breve{n}_1)$ and $\breve{B} = (\breve{l}_2, \breve{m}_2, \breve{n}_2)$ is two fuzzy triangular numbers.

Then the graded mean integration representation is defined as

$$G(\breve{A}) = \frac{1}{6}(\breve{l}_1 + 4\breve{m}_2 + \breve{n}_3)$$

$$G(\breve{B}) = \frac{1}{6}(\breve{l}_1 + 4\breve{m}_2 + \breve{n}_3)$$

If $\breve{A}$ and $\breve{B}$ are two fuzzy triangular numbers then its addition is expressed as:

$$G(\breve{A} + \breve{B}) = \frac{1}{6}(\breve{l}_1 + 4\breve{m}_1 + \breve{n}_1) + \frac{1}{6}(\breve{l}_2 + 4\breve{m}_2 + \breve{n}_2) - - - - - - - - -(4)$$

If $\breve{A}$ and $\breve{B}$ are two fuzzy triangular numbers then its multiplication is expressed as:

$$G(\breve{A} \times \breve{B}) = \frac{1}{6}(\breve{l}_1 + 4\breve{m}_1 + \breve{n}_1) \times \frac{1}{6}(\breve{l}_2 + 4\breve{m}_2 + \breve{n}_2) - - - - - - - - -(5)$$

4. Algorithm: Multistage Network Utilizing Multi-Valued Neutrosophic Numbers (MVNNs)

- Step 1: Select a source and destination vertex within the provided multistage network.
- Step 2: Convert the arc length values from multi-valued neutrosophic numbers to single-value neutrosophic numbers using the fuzzy simplicity method (equation-2).
- Step 3: Convert it from single-value neutrosophic numbers to a real number using graded mean integration (definition-3).
- Step 4: Then, using a dynamic approach, i.e., a forward and backward computation approach.

Backward Approach Algorithm:

```
#Algorithm for Backward Approach

 Algorithm BGraph (G, K, n, p)
# some function as FGraph
{
B cost [1] = 0.0;
For j = 2 to n do
{
# compute b cost [j].
Let r be such that  is an edge of
G and b cost [r] + c [r, j];
D [j] = r;
}
# find a minimum cost path
P [1] = 1;                  p [k] = n;
For j = k-1 to 2 do p[j] = d [p (j+1)];
}
```

Forward Approach Algorithm:

```
#Algorithm for Forward Approach

F graph (graph G, int K, int n, int p[])
{
Float cost [max size], int d [max size], r;
Cost [n] = 0.0
For (int j = n-1; j>= 1; j--)
{
Let r be a vertex such that  is an edge of G and C[j][r] + cost[r] is minimum;
Cost [j] = C[j][r] + Cost[r]
D [j] = r
}
#Find a minimum-cost path
P [1] = 1 ,  P[k] = n
For (j = 2 ; j <= K-1; j++)
P[j] = d[P(j-1)];
}
```

➢ **Step-5:** After applying the dynamic approach, i.e., forward and backward approach if both techniques produce the equal minimum values and the shortest path, then the path yielded in the process is called the optimal path or shortest path of a network.

5. Numerical Example:

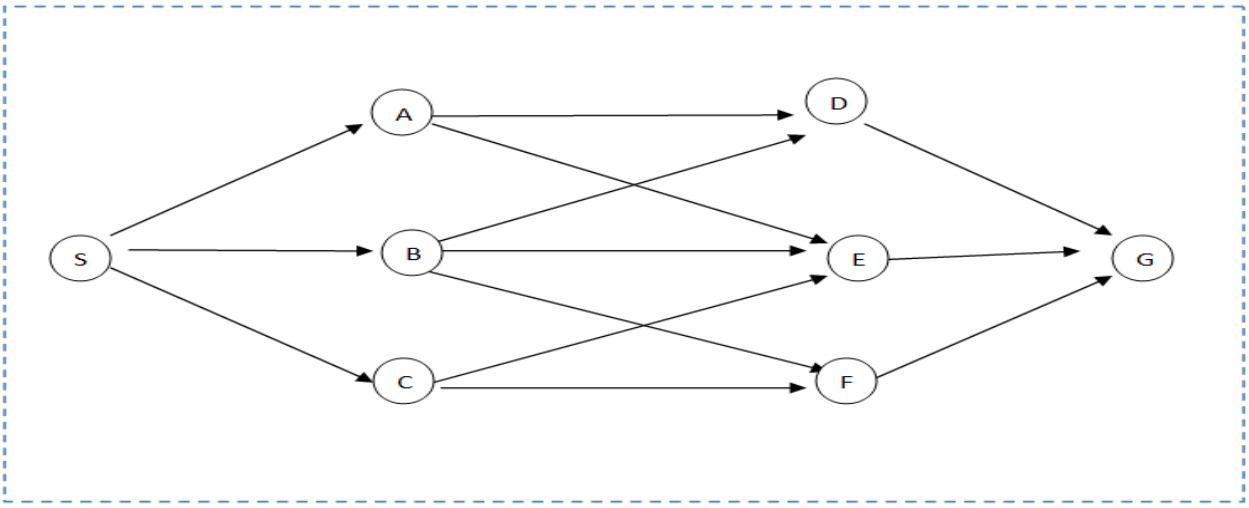

Fig- 1: Network

Arc	**Multi-membership value**
S→A	<[0.2,0.4,0.5],[0.3,0.5,0.6],[0.6,0.8,0.9]>
S→B	<[0.1,0.3,0.4],[0.3,0.4,0.7],[0.5,0.7,0.9]>
S→C	<[0.1,0.3,0.5],[0.3,0.5,0.7],[0.4,0.5,0.8]>
A→D	<[0.3,0.4,0.5],[0.4,0.5,0.6],[0.5,0.7,0.9]>

A→E	<[0.2,0.3,0.6],[0.3,0.4,0.8],[0.4,0.5,0.9]>
B→D	<[0.1,0.2,0.4],[0.3,0.4,0.6],[0.4,0.5,0.6]>
B→E	<[0.3,0.4,0.6],[0.3,0.4,0.7],[0.4,0.7,0.9]>
B→F	<[0.1,0.2,0.5],[0.2,0.4,0.5],[0.5,0.6,0.9]>
C→E	<[0.4,0.2,0.5],[0.6,0.5,0.8],[0.5,0.6,0.8]>
C→F	<[0.1,0.2,0.3],[0.2,0.5,0.6],[0.5,0.7,0.9]>
D→G	<[0.4,0.5,0.9],[0.6,0.7,0.8],[0.5,0.6,0.9]>
E→G	<[0.1,0.2,0.5],[0.2,0.4,0.5],[0.5,0.7,0.9]>
F→G	<[0.2,0.6,0.7],[0.2,0.5,0.8],[0.6,0.8,0.9]>

Table- 1: Arc weight in Multi-membership value

Implementation of Algorithm

Step-1:

From fig-1 assume that source node is S and destination node is G

Step-2:

Arc	Single Membership value
S→A	<[0.5, 0.6, 0.9]>
S→B	<[0.4,0.7,0.9]>
S→C	<[0.5,0.7,0.8]>
A→D	<[0.5,0.6, 0.9]>
A→E	<[0.6,0.8,0.9]>
B→D	<[0.4,0.6,0.6]>
B→E	<[0.6,0.7,0.9]>
B→F	<[0.5,0.5,0.9]>
C→E	<[0.5,0.8,0.8]>
C→F	<[0.3,0.6,0.9]>

D→G	<[0.9,0.8,0.9]>
E→G	<[0.5,0.5,0.9]>
F→G	<[0.7,0.8,0.9]>

Table- 2: Single Membership value

Step-3:

Converting the Single membership value into a real value by using Graded mean integration (definition-3.5)

Here $(\breve{l} = 0.5, \breve{m} = 0.6, \breve{n} = 0.9)$

$$G(\breve{A}) = \frac{1}{6}(\breve{l} + 4\breve{m} + \breve{n})$$

$$G(\breve{A}) = \frac{1}{6}(0.5 + 4 \times 0.6 + 0.9)$$

$$= 0.63$$

Similarly to find all the edge's value in Crisp number

Arc	**Single Membership value**
S→A	0.63
S→B	0.68
S→C	0.68
A→D	0.63

A→E	0.78
B→D	0.56
B→E	0.71
B→F	0.56
C→E	0.75
C→F	0.60
D→G	0.83
E→G	0.56
F→G	0.80

Table- 3: Membership value in crisp number

Step 4:

Backward Approach

In backward approach we start from source vertex, so the distance from source (S) to destination vertex (T) is (S, T) is given by

$$dis(S,G) = min\{0.63 + dis(A,G), 0.68 + dis(B,G), 0.68 + dis(C,G)\}$$

$$-------2.0$$

Now to calculate the distance (A to G), distance (B to G) and distance (C to G).

$$dis(A,G) = min\{0.63 + dis(D,G), 0.78 + dis(E,G)\}$$

$$dis(A,G) = min\{0.63 + 0.83, 0.78 + 0.56\}$$

$$dis(A,G) = min\{1.46, 1.34\}$$

$$dis(A,G) = 1.34 - - - - - - - - -2.1$$

$$dis(B,G) = min\{0.56 + dis(D,G), 0.71 + dis(E,G), 0.56 + dis(F,G)\}.$$

$$d(B,G) = min\{0.56 + 0.83, 0.71 + 0.56, 0.56 + 0.80\}$$

$$dis(B,G) = min\{1.39, 1.27, 1.36\}$$

$$d(B,G) = 1.27 - - - - - - - - -2.2$$

$$(C,G) = min\{0.75 + dis(E,G), 0.60 + dis(F,G)\}$$

$$(C,G) = min\{0.75 + 0.56, 0.60 + 0.80\}$$

$$(C,G) = min\{1.31, 1.40\}$$

$$(C,G) = 1.31 - - - - - - - - -2.3$$

Now Putting all this values in equation 2.0

$$dis(S,G) = \min\{0.63 + dis(A,G), 0.68 + dis(B,G), 0.68 + dis(C,G)\}$$

$$dis(S,G) = min\{0.63 + 1.34, 0.68 + 1.27, 0.68 + 1.31\}$$

$$dis(S,G) = min\{1.97,1.95,1.99\}$$

$$dis(S,G) = 1.95(\mathrm{S}-\mathrm{B}-\mathrm{E}-\mathrm{G})$$

Forward approach

Here $dis(S,A) = 0.63$

$$dis(S,B) = 0.68$$

$$dis(S,C) = 0.68$$

$$dis(S,D) = min\{0.63 + dis(A,D), 0.68 + dis(B,D)\}$$

$$dis(S,D) = min\{0.63 + 0.63, 0.68 + 0.56\}$$

$$dis(S,D) = min\{1.26,1.24\}$$

$$dis(S,D) = 1.24$$

$$dis(S,E) = min\{0.63 + dis(A,E), 0.68 + dis(B,E), 0.68 + dis(C,E)\}$$

$$dis(S,E) = min\{0.63 + 0.78, 0.68 + 0.71, 0.68 + 0.75\}$$

$$dis(S,E) = min\{1.41,1.39,1.43\}$$

$$dis(S,E) = 1.39$$

$$dis(S,F) = min\{0.63 + dis(B,F), 0.68 + dis(C,F)\}$$

$$dis(S,F) = min\{0.63 + 0.56, 0.68 + 0.60\}$$

$$dis(S,F) = min\{1.19, 1.28\}$$

$$dis(S,F) = 1.19$$

$$dis(S,G) = min\{dis(S,D) + dis(D,G), dis(S,E) + dis(E,G), dis(S,F) + dis(F,G)\}$$

$$dis(S,G) = min\{1.24 + 0.83, 1.39 + 0.56, 1.19 + 0.80\}$$

$$dis(S,G) = \{2.07, 1.95, 1.99\}$$

$$dis(S,G) = 1.95(\mathrm{S} \rightarrow \mathrm{B} \rightarrow \mathrm{E} \rightarrow \mathrm{G})$$

Implementation of Our algorithm with Python Programming Language

```python
#Implementation of our work with multistage graph in Python3 Compiler (shortest path).
import sys
#function for finding shortest distance inmulti-stage network
def Source_to_Destination(SG):
#list for storing shortest distance from particular node to N-1 node
    Distance=[0]*n
#finding the shortest paths
    for x in range(n-2, -1, -1):
        Distance[x]=infinity
#Checking nodes from next stages
        for y in range(n):
#condition when no edge exists
            if SG[x][y]==infinity:
                continue
#finding minimum distances
            Distance[x]=min(SG[x][y]+Distance[y],Distance[x])
    return Distance[0]
# Driver code
n=8
infinity=sys.maxsize
#Adjacency matrix for graph
SG=[[infinity, 0.63, 0.68, 0.68, infinity, infinity, infinity, infinity],
    [infinity, infinity, infinity,infinity, 0.63, 0.78, infinity, infinity],
    [infinity, infinity, infinity,infinity,0.56, 0.71, 0.56, infinity],
    [infinity, infinity, infinity, infinity, infinity,  0.75, 0.60, infinity],
    [infinity, infinity, infinity, infinity, infinity, infinity, infinity, 0.83],
    [infinity, infinity, infinity, infinity, infinity, infinity, infinity, 0.56],
    [infinity, infinity, infinity, infinity, infinity, infinity, infinity, 0.80]]
D=Source_to_Destination(SG)
print("SHORTEST PATH FROM SOURCE TO DESTINATION IS :",D)
```

Output Code:

```
SHORTEST PATH FROM SOURCE TO DESTINATION IS : 1.9500000000000002


...Program finished with exit code 0
Press ENTER to exit console.
```

Step-5:

In the dynamic approach, i.e., both forward and backward approaches have an equal minimum path value 1.95 and an equal path S-B-E-G, so this is the SP connecting the source vertex to destination vertex of this given Network.

7. Conclusion

In this paper, we find the shortest path (SP) on the multistage network by using the dynamic approach, i.e., the forward and backward approach, and then we implement our result in the Python programming language, and finally, we get the shortest path. The minimum cost between the source vertex and the destination vertex is 1.95. The most important objective of this research is to determine a new algorithm for solving multistage graphs. Right here, we propose a mathematical instance to show our new suggested method. I'm hoping that this paper will help new researchers find the SPP in multistage graphs.

References

[1] Zadeh, L. A. (1965). Fuzzy sets. Information and control, 8(3), 338-353.

[2] Atanassov, K. (1988). Review and new results on intuitionistic fuzzy sets. preprint Im-MFAIS-1-88, Sofia, 5(1).

[3] Smarandache, F., Abdel-Basset, M., & Broumi, S. (Eds.). (2021). Neutrosophic Sets and Systems, vol. 40, 2021. Infinite Study.

[4]Wang, H., Smarandache, F., Zhang, Y., & Sunderraman, R. (2010). Single valued neutrosophic sets. Infinite study.

[5]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.

[6] Basset, M. A., Mohamed, M., Sangaiah, A. K., & Jain, V. (2018). An integrated neutrosophic AHP and SWOT method for strategic planning methodology selection. Benchmarking: An International Journal.

[7] Abdel-Basset, M., Mohamed, M., & Smarandache, F. (2018). A hybrid neutrosophic group ANP-TOPSIS framework for supplier selection problems. Symmetry, 10(6), 226..

[8] Abdel-Basset, M., Mohamed, M., & Smarandache, F. (2018). An extension of neutrosophic AHP–SWOT analysis for strategic planning and decision-making. Symmetry, 10(4), 116.

[9] Abdel-Basset, M., Mohamed, M., Smarandache, F., & Chang, V. (2018). Neutrosophic association rule mining algorithm for big data analysis. Symmetry, 10(4), 106.

[10]Abdel-Basset, M., Zhou, Y., Mohamed, M., & Chang, V. (2018). A group decision making framework based on neutrosophic VIKOR approach for

e-government website evaluation. Journal of Intelligent & Fuzzy Systems, 34(6), 4213-4224.
[11] Abdel-Basset, M., Mohamed, M., Zhou, Y., & Hezam, I. (2017). Multi-criteria group decision making based on neutrosophic analytic hierarchy process. Journal of Intelligent & Fuzzy Systems, 33(6), 4055-4066.
[12] Abdel-Basset, M., & Mohamed, M. (2018). The role of single valued neutrosophic sets and rough sets in smart city: Imperfect and incomplete information systems. Measurement, 124(10.1016).
[13] Abdel-Basset, M., Manogaran, G., Gamal, A., & Smarandache, F. (2018). A hybrid approach of neutrosophic sets and DEMATEL method for developing supplier selection criteria. Design Automation for Embedded Systems, 22(3), 257-278.
[14] Dey, A., & Pal, A. (2019). Computing the shortest path with words. International Journal of Advanced Intelligence Paradigms, 12(3-4), 355-369.
[15] Vaidyanathan, R., Das, S., & Srivastava, N. (2015). Query expansion strategy based on pseudo relevance feedback and term weight scheme for monolingual retrieval. arXiv preprint arXiv:1502.05168.
[16] De, P. K., & Bhincher, A. (2011). Dynamic programming and multi objective linear programming approaches. Appl Math Inf Sci, 5(2), 253-263.
[17] Kumar, G., Bajaj, R. K., & Gandotra, N. (2015). Algorithm for shortest path problem in a network with interval-valued intuitionistic trapezoidal fuzzy number. Procedia Computer Science, 70, 123-129.
[18] Meenakshi, A. R., & Kaliraja, M. (2012). Determination of the shortest path in interval valued fuzzy networks. Int J Math Arch, 3(6), 2377-2384.
[19] 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.
[20]Book
Broumi, S., Bakali, A., Talea, M., & Smarandache, F. (2017). Shortest path problem under trapezoidal neutrosophic information. Infinite Study.
[21] Broumi, S., Bakali, A., Talea, M., Smarandache, F., & ALi, M. (2017). Shortest path problem under bipolar neutrosphic setting. In Applied Mechanics and Materials (Vol. 859, pp. 59-66). Trans Tech Publications Ltd.
[22] DeivanayagamPillai, N., Malayalan, L., Broumi, S., Smarandache, F., & Jacob, K. (2020). Application of Floyd's Algorithm in Interval Valued

Neutrosophic Setting. In Neutrosophic Graph Theory and Algorithms (pp. 77-106). IGI Global..
[23] Broumi, S., Dey, A., Talea, M., Bakali, A., Smarandache, F., Nagarajan, D., ... & Kumar, R. (2019). Shortest path problem using Bellman algorithm under neutrosophic environment. Complex & intelligent systems, 5(4), 409-416.

Received: 13 Dec ,2023 Accepted:19 Feb,2024