Application of Dijkstra Algorithm for Solving Interval Valued Neutrosophic Shortest Path Problem

Said Broumi  
Laboratory of Information Processing, Faculty of Science  
Ben M’sik, University Hassan II, B.P 7955, Sidi Othman,  
Casablanca, Morocco  
broumisaid78@gmail.com

Mohamed Talea  
Laboratory of Information Processing, Faculty of science  
Ben M’sik, University Hassan II, B.P 7955, Sidi Othman,  
Casablanca, Morocco  
taleamohamed@yahoo.fr

Assia Bakali  
Ecole Royale Navale, Boulverad Sour Jdid, B.P 16303  
Casablanca, Morocco.  
assiabakali@yahoo.fr

Florentin Smarandache  
Department of Mathematics, University of New Mexico,  
705 Gurley Avenue, Gallup, NM87301,  
fsmarandache@gmail.com

Abstract— In this paper, the authors propose an extended version of Dijkstra’ algorithm for finding the shortest path on a network where the edge weights are characterized by an interval valued neutrosophic numbers. Finally, a numerical example is given to explain the proposed algorithm.

Keywords—Dijkstra algorithm; interval valued neutrosophic number; Shortest path problem; Network;

I. INTRODUCTION

Smarandache [1] originally proposed the concept of a neutrosophic set from a philosophical point of view. The concept of the neutrosophic set (NS for short) has the ability to handle uncertain, incomplete, inconsistent, the indeterminate in a more accurate way. The theory of neutrosophic sets are a generalization of the theory of fuzzy sets [3], intuitionistic fuzzy sets [4] and interval- valued intuitionistic fuzzy sets [6]. The concept of the neutrosophic sets is expressed by a truth-membership degree (T), an indeterminacy-membership degree (I) and a falsity--membership degree (F) independently, which are within the real standard or nonstandard unit interval ]0,1[. The concept of neutrosophic set is difficult to apply it real scientific and engineering areas. For this purpose, Smarandache [1] introduced the concept of SVNS, an instance of neutrosophic set, whose functions of truth, indeterminacy and falsity are within [0, 1]. In fact sometimes the degree of truth-membership, indeterminacy-membership and falsity-membership about a certain statement can not be defined exactly in the real situations, but expressed by several possible interval values. So the interval valued neutrosophic set (IVNS) was required. For this purpose, Wang et al.[8] introduced the concept of interval valued neutrosophic set (IVNS for short), which is more precise and more flexible than the single valued neutrosophic set. The interval valued neutrosophic sets (IVNS) is a generalization of the concept of single valued neutrosophic set, in which three membership functions are independent, and their values belong to the unite interval [0, 1].

Some more literature about neutrosophic sets, interval valued neutrosophic sets and their applications in divers fields can be found in [9]. In addition, the operations on interval valued neutrosophic sets and their ranking methods are presented in [10-11].

The selection of shortest path problem (SPP) is one of classic problems in graph theory. Several algorithms have been proposed for solving the shortest path problem in a network. The shortest path problem appears in various disciplines such as transportation, road networks and other applications. The shortest path problems could be classified into three types [12]:

1) Problem of finding shortest path from a single source in which the aim is to find shortest path from source node to all other nodes around the graph.
2) Problem of finding shortest path to a single source in which the aim is to find the shortest path between each connected pair in the graph.
3) Problem of finding shortest path between each two nodes in which the aim is to find shortest path between connected pair in the graph.

In a network, the shortest path problem concentrates at finding the path from one source node to destination node with minimum weight. The edge length of the network may represent the real life quantities such as, cost, time, etc. In classical shortest path problem, it is assumed that decision maker is certain about the parameters (time, distance, etc)
The functions \( T_A(x) \), \( I_A(x) \) and \( F_A(x) \) are real standard or non-standard subset of \( [0,1] \).

Since it is difficult to apply NSs to practical problems, Wang et al. [7] introduced the concept of a SVNS, which is an instance of Neutrosophic set and can be utilized in real scientific and engineering applications.

**Definition 2.7** [7] Let \( X \) be an universe of discourse of points (objects) with generic elements in \( X \) denoted by \( x \). The single valued neutrosophic set \( A \) (SVNS A) is characterized by truth-membership function \( T_A(x) \), an indeterminacy-membership function \( I_A(x) \), and a falsity-membership function \( F_A(x) \).

For each point \( x \) in \( X \), \( T_A(x), I_A(x), F_A(x) \in [0,1] \). A SVNS \( A \) can be expressed as

\[
A = \{ < x: T_A(x), I_A(x), F_A(x) >, x \in X \} \tag{2}
\]

**Definition 2.8.** [7] Let \( X \) be an universe of discourse of points (object) with generic elements in \( X \) denoted by \( x \). An interval valued neutrosophic set \( A \) (IVNS A) is characterized by an interval truth-membership function \( T_A(x) = [T^L_A, T^U_A] \), an interval indeterminacy-membership function \( I_A(x) = [I^L_A, I^U_A] \), and an interval falsity-membership function \( F_A(x) = [F^L_A, F^U_A] \). For each point \( x \) in \( X \), \( T_A(x), I_A(x), F_A(x) \in [0,1] \). An IVNS \( A \) can be expressed as

\[
A = \{ < x: T_A(x), I_A(x), F_A(x) >, x \in X \} \tag{3}
\]

**Definition 2.9 [11].** Let \( \tilde{A}_1 = \left[ T^L_1, T^U_1, I^L_1, I^U_1, F^L_1, F^U_1 \right] \) and \( \tilde{A}_2 = \left[ T^L_2, T^U_2, I^L_2, I^U_2, F^L_2, F^U_2 \right] \) be two interval valued neutrosophic numbers. Then, the operations for IVNNSs are defined as below:

(i)

\[
\tilde{A}_1 \odot \tilde{A}_2 = \left[ T^L_1 + T^L_2 - T^L_1 T^L_2, T^U_1 + T^U_2 - T^U_1 T^U_2, I^L_1 + I^L_2 - I^L_1 I^L_2, I^U_1 + I^U_2 - I^U_1 I^U_2, F^L_1 + F^L_2 - F^L_1 F^L_2, F^U_1 + F^U_2 - F^U_1 F^U_2 \right] \tag{4}
\]

(ii)

\[
\tilde{A}_1 \oplus \tilde{A}_2 = \left[ T^L_1 + T^U_2 + I^L_1 + I^U_2, T^U_1 + T^L_2 + I^U_1 + I^L_2, F^L_1 + F^L_2 + I^L_1 + I^L_2, F^U_1 + F^U_2 + I^U_1 + I^U_2 \right] \tag{5}
\]

(iii)

\[
\lambda \tilde{A} = \left[ \lambda T^L_1, \lambda T^U_1, \lambda I^L_1, \lambda I^U_1, \lambda F^L_1, \lambda F^U_1 \right] \tag{6}
\]

(iv)

\[
\tilde{A}^{\lambda} = \left[ (\lambda T^L_1, \lambda T^U_1, \lambda I^L_1, \lambda I^U_1, \lambda F^L_1, \lambda F^U_1) \right] \tag{7}
\]

where \( \lambda > 0 \)

**Definition 2.5 [8].** An interval valued neutrosophic number \( \tilde{A} = \left[ T^L_1, T^U_1, I^L_1, I^U_1, F^L_1, F^U_1 \right] \) is said to be empty if and only if \( T^L_1 = 0, T^U_1 = 0, I^L_1 = 1, I^U_1 = 1, F^L_1 = 1, F^U_1 = 1 \) and is denoted by.
A convenient method for comparing two interval valued neutrosophic numbers is by use of score function.

Definition 2.6 [10]. Let \( \tilde{A}_i = \left[ I_i^L, I_i^U \right], \left[ F_i^L, F_i^U \right] \) be an interval valued neutrosophic number. Then, the score function \( s(\tilde{A}_i) \) and accuracy function \( H(\tilde{A}_i) \) of an IVNN are defined as follows:

(i) \( s(\tilde{A}_i) = \left( \frac{1}{4} \right) \left\{ 2 + I_i^L + I_i^U - 2F_i^L - 2F_i^U - I_i^L - I_i^U \right\} \)  
(ii) \( H(\tilde{A}_i) = \frac{I_i^L + I_i^U - (1-I_i^L) - (1-I_i^U) - F_i^L - F_i^U - (1-F_i^L) - (1-F_i^U)}{2} \)

Definition 2.7 [10]. Let \( \tilde{A}_1 = \left[ I_1^L, I_1^U \right], \left[ F_1^L, F_1^U \right] \) and \( \tilde{A}_2 = \left[ I_2^L, I_2^U \right], \left[ F_2^L, F_2^U \right] \) be two interval valued neutrosophic numbers. Then, we define a ranking method as follows:

i. If \( s(\tilde{A}_1) > s(\tilde{A}_2) \), then \( \tilde{A}_1 \) is greater than \( \tilde{A}_2 \), that is, \( \tilde{A}_1 \) is superior to \( \tilde{A}_2 \), denoted by \( \tilde{A}_1 \triangleright \tilde{A}_2 \).

ii. If \( s(\tilde{A}_1) = s(\tilde{A}_2) \), then \( H(\tilde{A}_1) > H(\tilde{A}_2) \), then \( \tilde{A}_1 \) is greater then \( \tilde{A}_2 \), that is, \( \tilde{A}_1 \) is superior to \( \tilde{A}_2 \), denoted by \( \tilde{A}_1 \triangleright \tilde{A}_2 \).

III. NETWORK TERMINOLOGY

In this subsection, we consider a directed network \( G= (V, E) \) where \( V \) denotes a finite set of nodes \( V= \{1, 2, \ldots, n\} \) and \( E \) denotes a set of \( m \) directed edges \( E \subseteq V \times V \). Each edge is denoted by an ordered pair \( (i, j) \) where \( i, j \in V \) and \( i \neq j \). In this network, two nodes denoted \( s \) (source) and \( t \) (target) are specified, which represent the source node and the destination node. The path is defined as sequence \( P_{ij} = \{i=i_1, (i_1,i_2), i_2, \ldots, i_{t-1}, (i_{t-1},i_t), i_t=j\} \) of alternating nodes and edges. The existence of at least one \( P_{ij} \) in \( G = (V, E) \) is supposed for every \( i \in V \setminus \{s\} \).

\( d_{ij} \) denotes an interval valued neutrosophic number assigned with the edge \((i,j)\), corresponding to the length necessary to traverse \((i,j)\) from \( i \) to \( j \). In real problems, the lengths correspond to the time, the distance, the cost, etc. Hence, the interval valued neutrosophic distance along the path is denoted as \( d(P) \) and is expressed as follows:

\[
D(P) = \sum_{(i,j) \in P} d_{ij} \tag{10}
\]

Remark: A node \( i \) is called predecessor of node \( j \) if

(i) Node \( i \) and node \( j \) is connected directly.

(ii) The direction of path connected the node \( i \) and the node \( j \) is from \( i \) to \( j \).

IV. INTERVAL VALUED NEUTROSOPHIC DIJKSTRA ALGORITHM

In this subsection, we modified the fuzzy Dijkstra’s algorithm adapted from [33] for computing the shortest path on a network where the edge weights are characterized by interval valued neutrosophic numbers.

This algorithm finds the shortest path and the shortest distance between a source node and any other node in the network. The interval valued neutrosophic Dijkstra’ algorithm forwards from a node \( i \) to an immediately successive node \( j \) using a neutrosophic labeling procedure. Let \( \tilde{u}_j \) be the shortest distance from node 1 to node \( i \) and \( s(\tilde{d}_{ij}) \geq 0 \) be the length of \((i,j)\) edge. Then, the neutrosophic label for node \( j \) is defined as:

\[
[\tilde{u}_j, i] = [\tilde{u}_i \oplus \tilde{d}_{ij}, i]. \quad S(\tilde{d}_{ij}) \geq 0. \tag{11}
\]

Here label \([\tilde{u}_j, i]\) mean we are coming from nodes \( i \) after covering a distance \( \tilde{u}_j \) from the starting node. Dijkktra’ algorithm classified the nodes into two classes: temporary set \((T)\) and permanent set \((P)\). A temporary neutrosophic label can be changed into another temporary neutrosophic label, if shortest path to the same neutrosophic node is checked. If no better path can be found then, the status of temporary neutrosophic label is replaced to permanent status.

In the following, we introduce the four steps of interval valued neutrosophic Dijkstra’ algorithm as follows:

Step 1: Assign to source node (say node 1) the permanent label \([<0,0],[1,1],[1,1]>,-\). Set \( i=1 \).

Making a node permanent means that it has been included in the short path.

Step 2: Determine the temporary neutrosophic label \([\tilde{u}_i \oplus \tilde{d}_{ij}, i]\) for each node \( j \) that can be arrived from \( i \), provided \( j \) is not permanently labeled. If node \( j \) is previously labeled as \([\tilde{u}_j, k]\) through another node \( K \), and if \( S(\tilde{u}_i \oplus \tilde{d}_{ij}) < S(\tilde{u}_j) \) change \([\tilde{u}_j, k]\) with \([\tilde{u}_i \oplus \tilde{d}_{ij}, i]\).

Step 3: When all the nodes are permanently labeled, the algorithm terminates. Otherwise, choose the label \([\tilde{u}_i, s]\) with shortest distance \( \tilde{u}_i \) from the list of temporary neutrosophic labels. Set \( i=r \) and repeat step 2.

Step 4: Select the shortest path between source node 1 and the destination node \( j \) by tracing backward through the network using the label’s information.

Remark: At each iteration among all temporary nodes, make those nodes permanent which have smallest distance. Note that at any iteration we can not move to permanent node, however,
reverse is possible. After all the nodes have permanent labels and only one temporary node remains, make it permanent. After describing the interval valued neutrosophic Dijkstra’ algorithm, in next section a numerical example is given to explain the proposed algorithm. The flow diagram of interval valued neutrosophic Dijkstra algorithm is depicted in figure1.

Fig. 1. Flow diagram representing the interval valued neutrosophic Dijkstra algorithm.

V. ILLUSTRATIVE EXAMPLE

In this subsection, an hypothetical example is used to verify the proposed approach. For this, we consider the network shown in figure2, then, we computed the shortest path from node 1 to node 6 where edges is represented by an interval valued neutrosophic numbers is computed. The extended Dijkstra algorithm is applied to the following network.

Fig. 2. A network with interval valued neutrosophic weights. In this network each edge has been assigned to interval valued neutrosophic number as follows:

### Table 1. Weights of the graphs

<table>
<thead>
<tr>
<th>Edges</th>
<th>Interval valued neutrosophic distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>&lt;[0.1, 0.2], [0.2, 0.3], [0.4, 0.5]&gt;</td>
</tr>
<tr>
<td>1-3</td>
<td>&lt;[0.2, 0.4], [0.3, 0.5], [0.1, 0.2]&gt;</td>
</tr>
<tr>
<td>2-3</td>
<td>&lt;[0.3, 0.4], [0.1, 0.2], [0.3, 0.5]&gt;</td>
</tr>
<tr>
<td>2-5</td>
<td>&lt;[0.1, 0.3], [0.3, 0.4], [0.2, 0.3]&gt;</td>
</tr>
<tr>
<td>3-4</td>
<td>&lt;[0.2, 0.3], [0.2, 0.5], [0.1, 0.2]&gt;</td>
</tr>
<tr>
<td>3-5</td>
<td>&lt;[0.3, 0.6], [0.1, 0.2], [0.1, 0.4]&gt;</td>
</tr>
<tr>
<td>4-6</td>
<td>&lt;[0.4, 0.6], [0.2, 0.4], [0.1, 0.3]&gt;</td>
</tr>
<tr>
<td>5-6</td>
<td>&lt;[0.2, 0.3], [0.3, 0.5], [0.1, 0.5]&gt;</td>
</tr>
</tbody>
</table>

Following the interval valued neutrosophic Dijkstra’s algorithm, the details of calculations are defined below.

**Iteration 0:** Assign the permanent label <[0, 0], [1, 1], [1, 1]> to node 1.

**Iteration 1:** Node 2 and node 3 can be arrived from (the last permanently labeled) node 1. Hence, the list of labeled nodes (Temporary and permanently) is available in the following table.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Label</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;[0, 0], [1, 1], [1, 1]&gt;</td>
<td>P</td>
</tr>
<tr>
<td>2</td>
<td>&lt;[0.1, 0.2], [0.2, 0.3], [0.4, 0.5]&gt;</td>
<td>T</td>
</tr>
<tr>
<td>3</td>
<td>&lt;[0.2, 0.4], [0.3, 0.5], [0.1, 0.2]&gt;</td>
<td>T</td>
</tr>
</tbody>
</table>

In order to compare <[0.1, 0.2], [0.2, 0.3], [0.4, 0.5]>and <[0.2, 0.4], [0.3, 0.5], [0.1, 0.2]>, we use Eq.8.

\[ S(<[0.1, 0.2], [0.2, 0.3], [0.4, 0.5]>) = 0.1 \]

\[ S(<[0.2, 0.4], [0.3, 0.5], [0.1, 0.2]>) = 0.175 \]

Since the rank of <[0.1, 0.2], [0.2, 0.3], [0.4, 0.5]>, 1 is less than <[0.2, 0.4], [0.3, 0.5], [0.1, 0.2]>, 1. Hence, the status of node 2 is replaced by the permanent status.

**Iteration 2:** Node 3 and node 5 can be arrived from node 2. Hence, the list of labeled nodes (temporary and permanent) is available in the following table.
Nodes | Label | Status
---|---|---
1 | [0.0, 0.1, 1.1]> - | P
2 | [0.1, 0.2, 0.2, 0.3]> [0.4, 0.5]> | P
3 | [0.2, 0.4, 0.3, 0.5]> [0.1, 0.2]> | T or [0.37, 0.52]> [0.02, 0.06]> [0.12, 0.25]> | T
4 | [0.19, 0.44]> [0.06, 0.12]> [0.08, 0.15]> | T

S ( [0.37, 0.52]> [0.02, 0.06]> [0.12, 0.25] ) = 0.59
S ( [0.19, 0.44]> [0.06, 0.12]> [0.08, 0.15] ) = 0.51

Iteration 3: Node 4 and node 5 can be arrived from node 3. Hence, the list of labeled nodes (temporary and permanently) is available in the following table

Nodes | Label | Status
---|---|---
1 | [0.0, 0.1, 1.1]> - | P
2 | [0.1, 0.2, 0.2, 0.3] > [0.4, 0.5]> | P
3 | [0.2, 0.4, 0.3, 0.5]> [0.1, 0.2]> | T or [0.37, 0.52]> [0.02, 0.06]> [0.12, 0.25]> | T
4 | [0.36, 0.58]> [0.06, 0.25]> [0.04, 0.1]> | T
5 | [0.19, 0.44]> [0.06, 0.12]> [0.08, 0.15]> | T or [0.35, 0.60]> [0.01, 0.04]> [0.008, 0.075]> | T

S ( [0.36, 0.58]> [0.06, 0.25]> [0.04, 0.1] ) = 0.54 and S ( [0.19, 0.44]> [0.06, 0.12]> [0.08, 0.15] ) = 0.51

Iteration 4: Node 6 can be arrived from node 5. Hence, the list of labeled nodes (temporary and permanently) is available in the following table

Nodes | Label | Status
---|---|---
1 | [0.0, 0.1, 1.1]> - | P
2 | [0.1, 0.2, 0.2, 0.3] > [0.4, 0.5]> | P
3 | [0.2, 0.4, 0.3, 0.5]> [0.1, 0.2]> | T or [0.37, 0.52]> [0.02, 0.06]> [0.12, 0.25]> | T
4 | [0.36, 0.58]> [0.06, 0.25]> [0.04, 0.1]> | T
5 | [0.19, 0.44]> [0.06, 0.12]> [0.08, 0.15]> | T or [0.35, 0.60]> [0.01, 0.04]> [0.008, 0.075]> | T

S ( [0.36, 0.58]> [0.06, 0.25]> [0.04, 0.1] ) = 0.54 and S ( [0.19, 0.44]> [0.06, 0.12]> [0.08, 0.15] ) = 0.51

Iteration 5: The only temporary node is 4, this node can be arrived from node 3 and node 6. Hence, the list of labeled nodes (temporary and permanently) is available in the following table

Nodes | Label | Status
---|---|---
1 | [0.0, 0.1, 1.1]> - | P
2 | [0.1, 0.2, 0.2, 0.3] > [0.4, 0.5]> | P
3 | [0.2, 0.4, 0.3, 0.5]> [0.1, 0.2]> | T or [0.37, 0.52]> [0.02, 0.06]> [0.12, 0.25]> | T
4 | [0.36, 0.58]> [0.06, 0.25]> [0.04, 0.1]> | T
5 | [0.19, 0.44]> [0.06, 0.12]> [0.08, 0.15]> | T or [0.35, 0.60]> [0.01, 0.04]> [0.008, 0.075]> | T

Based on the step 4, the shortest path from node 1 to node 6 is determined using the following sequence.

(6) → [0.35, 0.60] → [0.01, 0.04] → [0.008, 0.075] → (5) → [0.19, 0.44] → (0.06, 0.12) → (0.08, 0.15) → (2) → (0.19, 0.44) → (0.06, 0.12) → (0.08, 0.15) → (1)

Hence, the required shortest path is 1 → 2 → 5 → 6.
Fig 3. Network with interval valued neutrosophic shortest distance of each node from node 1.

Where, the neutrosophic label of each node is:

\[ \begin{align*}
[a, -] &= [[0.19, 0.44], [0.06, 0.12], [0.08, 0.15]], \\
[b, 1] &= [[0.2, 0.4], [0.3, 0.5], [0.1, 0.2]], \\
[c, 1] &= [[0.4], [0.4], [0.4]], \\
[d, 3] &= [[0.36, 0.58], [0.06, 0.25], [0.04, 0.1]], \\
[e, 2] &= [[0.19, 0.44], [0.06, 0.12], [0.08, 0.15]], \\
[f, 5] &= [[0.35, 0.60], [0.01, 0.04], [0.075]],
\end{align*} \]

VI. CONCLUSION

This paper extended the single valued neutrosophic Dijkstra’s algorithm for solving the shortest path problem of a network where the edge weights are characterized by an interval valued neutrosophic number. The use of interval valued neutrosophic numbers as weights in the graph express more precision than single valued neutrosophic numbers. Finally, a numerical example has been solved to check the efficiency of the proposed method. In future, we will research the application of this algorithm.

Acknowledgment

The authors are very grateful to the chief editor and reviewers for their comments and suggestions, which is helpful in improving the paper.

REFERENCES