Fuzzy Approach to Compare a Minimal Spanning Tree Problem by Using Various Algorithms

Archana A. Deshpande¹, Onkar K. Chaudhari²

¹ Assistant Professor, Applied Mathematics, G.H.Raisoni College of Engineering, Nagpur, Maharashtra, India.
² Professor, Applied Mathematics, G.H.Raisoni College of Engineering, Nagpur, Maharashtra, India.

Abstract

In classical graph theory there are various algorithms to find Minimal Spanning tree. The Minimal Spanning tree where edges having values in the form of Fuzzy number is most studied problem in fuzzy graph theory.

In this paper we solve one problem of a Minimal Spanning tree by using various algorithms in Fuzzy environment and compare the result. So that we can tell which algorithm is best to find Minimal Spanning tree or shortest path by using Fuzzy approach.

Keywords: Minimal Spanning tree, Fuzzy graph, Trapezoidal Fuzzy number, Graded mean ranking.

1. INTRODUCTION

In Graph theory minimal spanning tree problem is the most important & fundamental concept. It has wide applications in processing of Images, transportation, wireless telecommunication, and cluster analysis.

In classical graph theory, G = (M, N) be a connected weighted graph, where M is the vertices set and N is the edges set, assigning a real number to every edge of G. A circuit less sub graph of a graph is said to be a Spanning tree if all vertices of graph are included in it. Minimal Spanning tree is the Spanning whose total weight i.e. sum of all weights of edges is minimum. In classical graph theory vertices i.e. objects or items having definite value and relations between them i.e edges are exactly known. In classical
graph theory uncertainty is not properly represent. Many of the uncertain problems come in applications of graph theory. So to tackle that uncertainty and ambiguity Fuzzy graph [2, 10, 6, 9] concept is used. The fuzzy graph is more effective, precise and flexible for uncertainty and ambiguity.

In graph theory Minimal spanning is the most typical problem. Many researchers used this in applications of communication network, statistical cluster analysis, image processing etc. In classical graph theory almost all minimal spanning tree problem has the weights assigned to edges are in real numbers but in real life practical problems the parameters have not naturally precise there exits uncertainty and ambiguity. In classical graph theory various algorithms are there to find Minimal Spanning tree such as Prims algorithm, Kruskals algorithm & Dijkstra algorithm etc. Some researchers used stochastic Minimal Spanning tree problem in which assumption may not be true in realistic applications of minimal spanning tree problem. In such cases Fuzzy variables can be used to solve Minimal Spanning tree problem.

In Fuzzy Minimal Spanning tree it involves ranking and addition operation between fuzzy numbers and comparison of weights of edges. It is distinctive from the Classical Minimal Spanning tree, involving real numbers. In the Minimal Spanning tree problem of Fuzzy environment , the numbers assigning to the edges are Fuzzy variables , and it is hard to determine which edge is smaller than all the others edges. Various papers are published on Minimal Spanning tree problem in Fuzzy environment. Itoh and Ishii [16] was first formulated a Minimum Spanning tree problem with fuzzy edge weights as a chance constrained programming model based on the necessity measure. After that Chang and Lee [11] presented three approaches based on the overall existence ranking index for ranking fuzzy edge weights of Spanning trees. De Almeida, Yamakami, and Takahashi [15] studied the Minimum Spanning tree problem with fuzzy parameters & proposed an exact algorithm and a special genetic algorithm.

In Fuzzy Minimal Spanning tree problem the important component is ranking of fuzzy number. As fuzzy numbers represent uncertainty and ambiguity in numerical values there is uncertainty between the comparisons of fuzzy numbers. There are various methods for fuzzy ranking of numbers. For ranking of fuzzy numbers here we are used Canonical Representation of operations on trapezoidal fuzzy numbers which is based on the graded mean integration representation method [5].

In this paper, we are solved a numerical example in classical method by using three algorithms Prims, Kruskals and Dijkstra's and same numerical with fuzzy numbers as a weight of edges i.e., for the FMST problem. The canonical representation of operations on fuzzy numbers is used in the algorithm for addition and comparison between fuzzy numbers.
The flow of the paper is as follows, in Section 2 some preliminaries is elaborated with definitions. In section 3 we solved a numerical on minimal spanning tree problem by classical method using Prims, Kruskals and Dijkstras algorithms. In section 4, we solve same numerical in fuzzy environment of Prim's Kruskals and Dijkstras algorithm for FMST problem. Finally, we conclude in section 5.

2. PRELIMINARIES

In this section we discussed Fuzzy number, fuzzy graph, Fuzzy set, Canonical representation of operations on fuzzy numbers and classical algorithms.

Fuzzy logic is branch of Mathematics in which degrees of membership function is used rather than a strict true/false membership. Uncertainty, Ambiguity and Imprecision can be easily handled by this tool. The real world problems mainly consisting of many other solutions which were less proven on the basis of robustness, low-cost and tractability, hence to overcome all these aspects [3] has introduced this tool. Uncertainty in many fields can be restricted by applying Fuzzy sets. Fuzzy set theory is a response to the demand for ideas and approaches for handling non statistical uncertainty. To optimize the profit in production and sales, we motivated highly to use Fuzzy logic because of its strength to handle imprecise data and ability to deal the ambiguity.

Crisp Set: A particular set, in which belongingness of any member or element in the set is either 0/1 or true/false.

The crisp set is defined as follows:

\[
\mu_A(y) = 1, \text{ if } y \in A \\
= 0, \text{ if } y \notin A
\]

Fig 1: Characteristic function for crisp set
Fuzzy Set: A Fuzzy set is well defined by its membership function. It relates to each element \( y \) in \( A \), an element \( f(y) \) in \([0,1]\). Every variable in well-defined Fuzzy set is having intermediate values or grades in between zero and one, implies that each set contain elements which is partial member of that set. The degree of belonging is called membership functions which has values from 0 to 1. If \( U \) is the Universal set and its elements are denoted as \( a \), in contrast with crisp set, then the fuzzy set \( A \) of \( U \) has characteristics function associated to it.

The Fuzzy set is represented by a Characteristic function, defined as follows:

\[
\mu_A : A \rightarrow [0,1]
\]

\[
\mu_A(y) = \begin{cases} 
1, & \text{if } y \text{ is totally in } A \\
0, & \text{if } y \text{ is not in } A \\
0 < \mu_A(y) < 1, & \text{if } y \text{ is partially in } A 
\end{cases}
\]

Fig 2: Fuzzy set Characteristic function

The value 0 means a non-belonging and 1 corresponds to the full belonging. Therefore, a fuzzy membership function \( \mu_A(y) \) represents the degree of membership of some element \( y \) of set \( A \).

Fuzzy Graph: Let \( M \) be a non-empty and finite vertices set. A fuzzy graph is a pair of functions \( G = (f, h) \), where \( f \) is a fuzzy subset of \( M \) and \( h \) is a symmetric Fuzzy relation on \( f \), i.e., \( f : M \rightarrow [0; 1] \) and \( h : M \times M \rightarrow [0; 1] \) such that \( (m; n) \leq \min(f(m); f(n)) \)

The fuzzy number is a fuzzy set with the conditions such as Convex Fuzzy set, Normalized fuzzy set, The membership function of Fuzzy number is piecewise continuous, It is defined in the real number.

Various types of Fuzzy numbers are defined in the Fuzzy theory, e.g., Triangular Fuzzy
Number, Trapezoidal Fuzzy number, L-R Fuzzy number and bell shape fuzzy number. Most fuzzy numbers get their names from the sharp of member functions. In this study, we use Trapezoidal Fuzzy number to represent the weight of edges the graph.

Triangular membership function can be calculated as shown in below Fig 3

![Triangular Membership function](image)

**Fig 3:** Triangular Membership function

Membership function $\mu_A(a)$ is represented by eq1 as follows.

$$
\mu_A(a) = \begin{cases} 
0 & \text{if } a \leq l \\
\frac{a-l}{m-l} & \text{if } l \leq a \leq m \\
\frac{n-a}{n-m} & \text{if } m \leq a \leq n \\
0 & \text{if } a \geq n 
\end{cases}
$$

---

**eq1**

Trapezoidal function defined by a lower limit a, an upper limit d, a lower support limit b, and an upper support limit c, where $a < b < c < d$ is shown in Fig 4.

![Trapezoidal Membership function](image)

**Fig 4:** Trapezoidal Membership function
The ranking of M and N is formulated as follows:

\[ M \geq N \text{ if and only if } P(M) > P(N) \text{ or } P(M) = P(N). \]

\[ M \leq N \text{ if and only if } P(M) < P(N) \text{ or } P(M) = P(N). \]

\[ M = N \text{ if and only if } P(M) = P(N). \]

Let a Trapezoidal Fuzzy number \( M = (a,b,c,d) \), the graded mean integration representation of Trapezoidal Fuzzy number M is defined as

\[ P(M) = \frac{1}{6}(a+2b+2c+d) \quad \text{---------------- eq3} \]

The addition operation on Trapezoidal M and N can be defined [1, 14]

\[ P(M \oplus N) = \frac{1}{6}(a_1+2b_1+2c_1+d_1)+\frac{1}{6}(a_2+2b_2+2c_2+d_2) \quad \text{---------------- eq4} \]

3. NUMERICAL SOLUTION BY CLASSICAL METHOD

We solved the following numerical by using classical Prims, Kruskals and Dijkstra’s algorithm.
3.1 Prims algorithm:
Consider the graph G as shown in Fig 5. Prims algorithm [12] is a greedy algorithm in which size of tree is increased in every step.

Step 1: Starts with a single vertex \( M = \{ x \} \) in Spanning tree \( T \).

Step 2: Include an edge to \( T \) with minimum weight with one end vertex ‘\( x \)’ in step 1.

Step 3: Add next edge with minimum weight whose one end vertex must in \( M \) of \( T \).

Step 4: If ‘\( M \)’ contains all vertices of \( G \) then stop else go to step 2 and 3.

3.2 Kruskals algorithm:
In this algorithm

Step 1: Arrange all the edges in ascending order weights.

Step 2: Start with an edge of minimum weight in spanning tree \( T \).

Step 3: Add next edge with minimum weight in \( T \).

Step 4: If \( T \) contains all vertices of \( G \) then stop else repeat step 3 to get minimal spanning tree.

3.3 Dijkstra’s Algorithm [7]:
In this algorithm

Step 1. Initially denote all vertices as unvisited. Denote the starting vertices as source. For each vertices assign a distance zero for the one we start from.

Step 2: Assign \( \infty \) for the remaining.

Step 3: Add a vertex with 0 distance in \( T \).

Step 4: After including a vertex in \( M \), update distance values of its adjacent vertices. Add an edge with minimum weight.

Step 5: Continue this step until all vertices are assign the number i.e. visited or the vertices with smallest distance is unvisited set is infinity, then stop Otherwise, assign vertices with minimum distance as current and go to step 4.
4. NUMERICAL SOLUTION IN FUZZY ENVIRONMENT

Consider the Fuzzy graph G for calculations as shown in Fig 7.

**4.1 Prim’s Algorithm:**
Initially set $M_{new} = \{y\}$, where $y$ is an arbitrary vertex from $M$ and $N_{new}=\emptyset$, costFN
represents the nodes, edges and cost of the corresponding FMST in fuzzy number which we calculated by addition operation [1].

Step1: start with a vertex A so \( M_{\text{new}} = \{A\} \), we find all edges with one end vertex A i.e B or F. The two edges (A,F), (A,B) connected with A. We use graded mean integration (eq3) of fuzzy numbers to find value of P(cost(A,B)) & P(cost(A,F))

\[
P(\text{cost}(A,B)) = \frac{1}{6}(3+2\times6+2\times9+10) = 7.16,
\]

\[
P(\text{cost}(A,F)) = \frac{1}{6}(9+2\times11+2\times13+14) = 11.83
\]

The minimum cost is P(cost(A,B)) so add (A,B) in E_{\text{new}}

So \( M_{\text{new}} = \{A,B\} \), \( N_{\text{new}}=\{(A,B)\} \) and cost\( FN = (3,6,9,10) \)

Step2: Now to find next edge such as one end point is either A or B. We have two vertices F or C i.e two edges (A,F) or (B,C)

\[
P(\text{cost}(A,F)) = \frac{1}{6}(9+2\times11+2\times13+14) = 11.83, \quad P(\text{cost}(B,C)) = \frac{1}{6}(4+2\times7+2\times10+11) = 7
\]

The minimum cost is P(cost(B,C)) so add (B,C) in E_{\text{new}}

So \( M_{\text{new}} = \{A,B,C\} \), \( N_{\text{new}}=\{(A,B),(B,C)\} \) and
cost\( FN = (3,6,9,10)+(4,7,10,11) = (7,13,19, 21) \)

In Similar way we added other edge (C,D) , (D,E) and (E,F) in spanning tree with cost\( FN = (12,21,30,33) \), cost\( FN = (18,30,42,47) \) and cost\( FN = (25,40,55,65) \) in Step3,4,& 5 respectively.

So \( M_{\text{new}} = \{A,B,C,D,E,F\} \), \( N_{\text{new}}=\{(A,B),(B,C),(C,D),(D,E),(E,F)\} \)

\[
P(\text{CostFN}) = \frac{1}{6}(25+2\times40+2\times55+65) = 46.66
\]

4.2 Kruskals Algorithm:

\[
P(\text{cost}(A,B)) = \frac{1}{6}(3+2\times6+2\times9+10) = 7, \quad P(\text{cost}(B,C)) = 8.16, \quad P(\text{cost}(C,D)) = 9.1
\]

\[
P(\text{cost}(D,E)) = 10.3, \quad P(\text{cost}(E,F)) = 11.33 \quad P(\text{cost}(A,F)) = 14.16
\]

Step1: Arrange the edges in ascending order. For that we find P(cost) of every edge by using graded mean integration representation (eq3) of the fuzzy numbers.

Step2: First add edge in \( E_{\text{new}} = \{(A,B)\} \) cost\( FN = (3,6,9,10) \)

Step3: Next add edge in spanning tree (B,C) so \( E_{\text{new}} = \{(A,B),(B,C)\} \)

cost\( FN = (7,13,19,21) \)
Continue this until we got spanning tree in Step 4,5,6 respectively with
\[ E_{\text{new}} = \{(A,B),(B,C),(C,D),(D,E),(E,F)\} \]
\[ \text{costFN} = (25,40,55,65) \]
\[ P(\text{costFN}) = \frac{1}{6}(25+4\times40+55) = 46.66 \]

### 4.3 Dijkstra’s Algorithm step by step:

1) Mark all vertices unvisited and make a set of them. Denote the starting vertices as Source, assign 0 distance to a vertex from which we start.

2) Denote $\infty$ for the rest.

3) Calculate distance from starting point for all unvisited neighbors of current vertices by using mean graded integration fuzzy number

4) On comparing the new value with old and assign smaller one..

5) Delete the current vertices from the unvisited set.

6) If all vertices of G are included in T or the unvisited vertices having smallest distance is infinity, then stop.

7) Otherwise, set the vertices with shortest distance as current and go to step 3.

### 5. COMPARISON OF ALGORITHMS

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Prims</th>
<th>Kruskals</th>
<th>Dijkstras</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Start with Minimum weight edge(A,B)=(3,6,9,10)</td>
<td>Arrange all the edges in ascending order</td>
<td>Start with any vertex(Source) assigning as 0</td>
</tr>
<tr>
<td>2</td>
<td>Next add edge(B,C) so CostFN=(7,13,19,21)</td>
<td>Start with minimum edge(A,B)=(3,6,9,10)</td>
<td>Remaining vertices consider at infinite distance from source.</td>
</tr>
<tr>
<td>3</td>
<td>Next add (C,D) CostFN=(12,21,30,33)</td>
<td>Next add edge(B,C) so CostFN=(7,13,19,21)</td>
<td>Add first edge(A,B)=(3,6,9,10)</td>
</tr>
<tr>
<td>4</td>
<td>Next add edge (D,E) CostFN=(18,30,42,47)</td>
<td>Next add (C,D) CostFN=(12,21,30,33)</td>
<td>Next add edge(B,C) so CostFN=(7,13,19,21)</td>
</tr>
<tr>
<td>5</td>
<td>Next add edge (E,F),costFN=(25,40,55,65)</td>
<td>Next add edge (D,E) CostFN=(18,30,42,47)</td>
<td>Next add (C,D) CostFN=(12,21,30,33)</td>
</tr>
</tbody>
</table>
6. CONCLUSION

The various classical algorithms are available for solving shortest path problems which do not have scope of any uncertainty and ambiguity. Hence we tried to apply the fuzzy ranking method with Triangular fuzzy number to find shortest path by various algorithms.

It is found that the approach of fuzzy logic gives minimum weight for spanning tree solution by considering all kind of uncertainty & ambiguity and it is also found that Prims algorithm requires less number of iterations as compared to other two for finding solution in classical and Fuzzy environment.

In future we will use different methods of ranking for shortest path and compare, so that we can find best method. Which we can use in different applications.

REFERENCES


