Soft Computing Project Report

Spanning tree using Fuzzy Logic

Team Members:

T.Guru Theja Reddy – 19BCD7034

K. Nuthana – 19BCD7061

V.M Bhuvanesh – 19BCD7088

M.Vyshnavi – 19BCE7418

Gudied By:

DR. Arindam Dey

Abstarct:

The minimum spanning tree (MST) problem is a well known optimization problem in graph theory that has been used to model many real life problems, e.g., telecommunications,transportation network, routing and water supply network. The MST problems with deterministic edge costs have been worked intensively and the MST of a connected weighted graph can be determined using many efficient algorithms introduced by outstanding scientists.

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.

Introduction:

Spanning tree is a sub-graph of an undirected connected graph, which includes the vertices of the graph with a minimum number of edges. A minimum spanning tree is a spanning tree in which the sum of the edges is minimum. 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 sub graph of a graph is said to be a Spanning tree if all vertices of graph are included in it. 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. The fuzzy graph is more effective, precise and flexible for uncertainty and ambiguity.

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 a fuzzy number here, we have used Canonical Representation of operations on trapezoidal fuzzy numbers which is based on the graded mean integration representation method.

In this paper, we are going to solve numerical example in classical method by using three algorithms' Prims, Kruskal's 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.

Preliminaries:

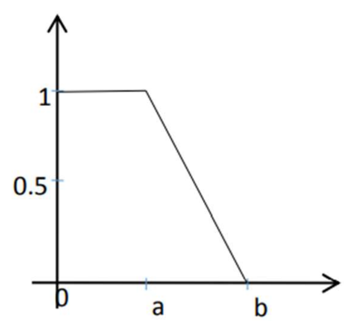
A fuzzy set is a mapping of a set of real numbers onto membership values that (generally) lie in the range [0, 1]. In this fuzzy package a fuzzy set is represented by a set of pairs ui/xi, where ui is the membership value for the real number xi. We can represent the set of values as { u1/x1 u2/x2 ... un/xn }. The x values in the set are in increasing order (x1 <= x2 <= ... <= xn). Values prior to x1 have the same membership value as x1 and values after xn have the same membership value as xn. Values between xi and xi+1 are determined by the value that lies on the straight line between the 2 consecutive points.

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

$$\mu_A(y) = \quad 1 \text{ , if } y \text{ is totally in } A$$

$$\mu_A(y) = \quad 0 \text{ , if } y \text{ is not in } A$$

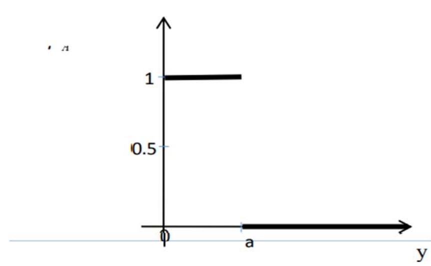
$$0 < \mu_A(y) < 1 \text{ ,if } y \text{ is partially in } A$$



Fuzzy set is a natural way to deal with the imprecision. Many real-world representations rely on significance rather than precision. Fuzzy logic is the best way to deal with them. Fuzzy set is an extension of crisp set.

The crisp set is defined as follows:

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



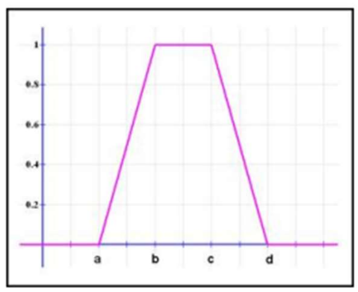
Crisp set: Degree of its member is either 0/1 or true/false.

Fuzzy Graph: A fuzzy graph (f-graph) is a pair G: (s, μ) where s is a fuzzy subset of a set S and μ is a fuzzy relation on s. A fuzzy graph H: (t, u) is called a partial fuzzy subgraph of G: (s, μ) if t (u) £ s(u) for every u and u (u, v) £ μ (u, v) for every u and v.

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 fuzzy theory, ex: triangular, trapezoidal, L-R, bell Shape. Here we are considering trapezoidal fuzzy Number

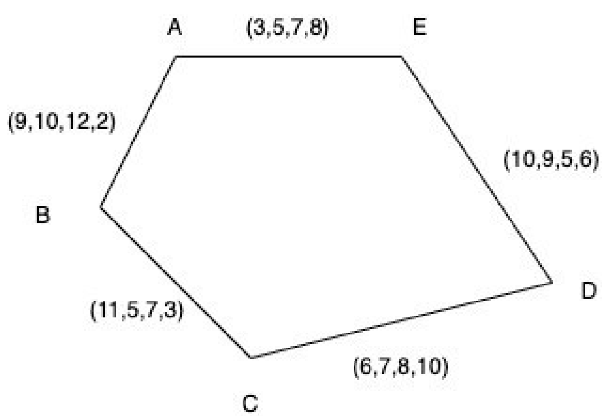
Trapezoidal function defined by a lower limit a, an upper limit d, a lower support limit b, and an upper support limit c, where a < b < c < d



Problem:

In this problem we are considering a graph with 5 vertices which are connected by 5 edges and their weights are given as follows.

Consider the Fuzzy graph G for calculations as shown in Fig



Solution:

Here we are using prim's algorithm to find the shortest path

Consider the graph G as shown in Fig. Prims algorithm is a greedy algorithm in which size of tree is increased in every step.

- Starts with a single vertex M={x} in Spanning tree T.
- Include an edge to T with minimum weight with one end vertex 'x' in step 1.
- Add next edge with minimum weight whose one end vertex must in M of T
- If 'M' contains all vertices of G then stop else go to step 2 and 3

Initially set Mnew = {y}, where y is an arbitrary vertex from M and Nnew=∅ , costFN

represents the nodes, edges and cost of the corresponding FMST in fuzzy number which we calculated by addition operation

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

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) = (1/6)(a+2b+2c+d)

P(cost(A,B)=1/6 (9+2x10+2x12+2)=9.16,

P(cost(A,E)= 1/6 (3+2x5+2x7+8)=5.83

The minimum cost is P(cost(A,E)) so add (A,B) in Enew

So Mnew = {A,E}, Nnew={(A,E)} and costFN = (3,5,7,8)

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

P(cost(A,B) = 1/6 (9+2x10+2x12+2)=9.16 P(cost(E,D) = 1/6 (10+2x9+2x5+6) = 7.33

The minimum cost is P(cost(E,D)) so add (E,D) in Enew

So Mnew = {A,E,D}, Nnew={(A,E),(E,D)} and

costFN = (3,5,7,8)+(10,9,5,6) = (13,14,12,14)

In Similar way we added other edge (D,C) ,(C,B) in spanning tree with costFN = (19,21,20,24), costFN = (30,26,27,27) in Step3,4,& 5 respectively.

So Mnew = {A,E,D,C,B}, Nnew={(A,E),(E,D),(D,C),(C,B)}

P(CostFN) = 1/6 (30+2x26+2x27+27) = 27.16

Code:

```
import sys
class Graph():
  def _init_(self, vertices):
    self.V = vertices
    self.graph = [[0 for column in range(vertices)]
          for row in range(vertices)]
  def printMST(self, parent):
    print ("Edge \tWeight")
    sum=0
    for i in range(1, self.V):
     sum=sum+self.graph[i][parent[i]]
     print (parent[i], "-", i, "\t", self.graph[i][parent[i]])
    print("Total minimum cost is ",sum)
  def minKey(self, key, mstSet):
    min = sys.maxsize
    for v in range(self.V):
      if key[v] < min and mstSet[v] == False:
```

```
            min = key[v]
            min_index = v
        return min_index

    def primMST(self):
        key = [sys.maxsize] * self.V
        parent = [None] * self.V
        key[0] = 0
        mstSet = [False] * self.V
        parent[0] = -1
        for cout in range(self.V):
            u = self.minKey(key, mstSet)
            mstSet[u] = True
            for v in range(self.V):
                if self.graph[u][v] > 0 and mstSet[v] == False and key[v] >
self.graph[u][v]:
                    key[v] = self.graph[u][v]
                    parent[v] = u
        self.printMST(parent)

n=int(input("Enter number of vertices "))
e=int(input("Enter number of edges "))
d = Graph(n)
for i in range(e):
    u=int(input("Enter source "))
    v=int(input("Enter dest "))
    q=[int(item) for item in input("Enter weight ").split()]
```

```
w=(1/6)*(q[0]+(2*q[1])+(2*q[2])+q[3])
d.graph[u][v]=w
d.graph[v][u]=w


d.primMST();
```

Results:

```
Enter number of vertices 5
Enter number of edges 5
Enter source 0
Enter dest 1
Enter weight 9 10 12 2
Enter source 1
Enter dest 2
Enter weight 11 5 7 3
Enter source 2
Enter dest 3
Enter weight 6 7 8 10
Enter source 3
Enter dest 4
Enter weight 10 9 5 6
Enter source 4
Enter dest 0
Enter weight 3 5 7 8
Edge    Weight
2 - 1    6.333333333333333
3 - 2    7.666666666666666
4 - 3    7.333333333333333
0 - 4    5.833333333333333
Total minimum cost is  27.166666666666664
```

Conclusion:

The various classical algorithms are available for solving shortest path problems which do not have uncertainty and ambiguity. So, here we applied fuzzy ranking method with Trapezoidal fuzzy number to find shortest path.

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

References:

Pettie, S., & Ramachandran, V. (2002). An optimal minimum spanning tree algorithm. *Journal of the ACM (JACM)*, *49*(1), 16-34.

Janiak, A., & Kasperski, A. (2008). The minimum spanning tree problem with fuzzy costs. *Fuzzy Optimization and Decision Making*, *7*(2), 105-118.

Dey, A., Mondal, S., & Pal, T. (2019). Robust and minimum spanning tree in fuzzy environment. *International Journal of Computing Science and Mathematics*, *10*(5), 513-524.

Dey, A., & Pal, A. (2016). Prim's algorithm for solving minimum spanning tree problem in fuzzy environment. *Annals of Fuzzy Mathematics and Informatics*, *12*(3), 419-430.

Dey, A., Son, L. H., Pal, A., & Long, H. V. (2020). Fuzzy minimum spanning tree with interval type 2 fuzzy arc length: formulation and a new genetic algorithm. *Soft computing*, *24*(6), 3963-3974.

de Almeida, T. A., Yamakami, A., & Takahashi, M. T. (2005, November). An evolutionary approach to solve minimum spanning tree problem with fuzzy parameters. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)* (Vol. 2, pp. 203-208). IEEE.

Mohanty, S. P., Biswal, S., & Pradhan, G. (2012). Minimum spanning tree in fuzzy weighted rough graph. *International Journal of Engineering Research and Development*, *1*(10), 23-28.