International Journal of Engineering Research and Development
ISSN: 2278-067X, Volume 1, Issue 10 (June 2012), PP.23-28
www.ijerd.com

Minimum Spanning Tree in Fuzzy Weighted Rough Graph

S. P. Mohanty[1], S. Biswal[2], G. Pradhan[3]

[1,2,3]*Department of Mathematics, College of Engineering and Technology, Techno Campus, Ghatikia, Bhubaneswar-751003, India*

Abstract—*The minimum spanning tree of a connected graph has many applications in different field of knowledge. In many real world problems the input data are often imprecise due to incomplete or non-obtainable information. This paper is intended to find minimum spanning tree on a rough graph where the edges have fuzzy weights. The Boruvka algorithm has been used to find the minimum spanning tree and signed distance ranking method has been used for ranking fuzzy numbers. Triangular fuzzy numbers are used as the weight of the edges.*

Keywords and Phrases—*Triangular fuzzy number, Ranking of fuzzy numbers, Rough set, Rough graph, Boruvka algorithm.*

I. INTRODUCTION

Many optimization problems can be tackled using graph theory. The classical graph theory may not be useful to deal with the problems where the data related to the graph such as weights attached to the vertices or edges, nature of connectivity between the vertices are imprecise. The fuzzy set theory can be effectively used to deal with these types of problems. But due to lack of proper information regarding the nature of connectivity between the vertices, the fuzzy set theory is also failed to handle the problems. *T. He et al* [5][6][7] have developed rough graph where they have applied rough set theory to attributed graphs. An attribute set was first assigned to edges of a graph. Given a subset R of the attribute set, there is associated an equivalences relation, based in which R-rough graph is defined. If the vertices of a graph are connected through edges denoting different possible relationships, it is required to quantify some attributes as the strength of relationship. The strength of relation can be deterministic where we can use crisp number, if the strength of relationship is imprecise, we can use fuzzy number.

This paper is intended to find the minimum spanning tree on a rough graph as developed by *T.He et al*, where the edges have fuzzy weights. The Boruvka algorithm has been used to find the minimum spanning tree and signed distance ranking method as suggested by *J.S. Yao and F.T Lin* [1] has been used to rank two fuzzy numbers. We have also used fuzzy triangular number as the weight of the edges.

This paper is organized as follows: some prerequisite related to fuzzy numbers, signed distance ranking methods are given in section-II. Various definitions related to rough graph are given in section III and section IV presents different algorithms to find the lower and upper approximations of rough graph and the minimum spanning tree of a fuzzy weighted rough graph. We have given one numerical example in section V to implement the algorithm. Finally in section VI, we conclude the paper with a suggestion for further work.

II. FUZZY NUMBER AND SIGNED DISTANCE RANKING

A. Fuzzy number

All definitions and discussion are based on the definition of fuzzy set, fuzzy numbers. For the sake of completeness of the work, we introduce here the following definitions related to our work.

1) ***Definition*** : The level λ triangular fuzzy number $\widehat{A}$,$0< \lambda \leq 1$, denoted by $\widehat{A}$ = (a, b, c ; λ) is a fuzzy set defined on R with the membership function defined as

$$\mu_{\widehat{A}}(x) = \begin{cases} \lambda (x-a)/(b-a), & a \leq x \leq b \\ \lambda (c-x)/(c-b), & b \leq x \leq c \\ 0 & \text{otherwise} \end{cases}$$

The family of all level λ fuzzy numbers be denoted by $F_N(\lambda) = \{ (a, b, c ; \lambda) |\ a, b, c \in R \text{ and } a < b < c \}$

In addition, for $\lambda = 1$, (a, b, c ; λ) is a triangular fuzzy number and denoted by (a, b, c).

2) ***Definition*** : A fuzzy set $\widehat{A}$ defined on R is called an interval–valued fuzzy set having the following membership function

$\widehat{A} = \{ (x , [\mu_{\widehat{A}^L}(x), \mu_{\widehat{A}^U}(x)]) \}$ where $x \in R$, $0 \leq \mu_{\widehat{A}^L}(x) \leq \mu_{\widehat{A}^U}(x) \leq 1$.

Symbolically, $\widehat{A}$ is denoted by [$\widehat{A}^L$, $\widehat{A}^U$] for all $x \in R$. The membership grade of x in $\widehat{A}$ lies in the interval $[\mu_{\widehat{A}^L}(x), \mu_{\widehat{A}^U}(x)]$. In particular if $\widehat{A}^L$ = (a, b, c ; λ) and $\widehat{A}^U$ = (p, b ,q ; ρ) where $0 < \lambda \leq \rho \leq 1$, $p < a < b < c < q$, we get $\widehat{A}$ = [a, b, c ; λ) , (p, b, q ; ρ)] which is called the level (λ , ρ) interval–valued fuzzy number . Then the membership function of $\widehat{A}$ can be defined as

$\mu_{\widehat{A}^L}(x) = \lambda (x - a) / (b - a)$, $a \le x \le b$
$= \lambda(c - x) / (c - b)$, $b \le x \le c$
$= 0$ otherwise

$\mu_{\widehat{A}^U}(x) = \rho (x - b) / (b - p)$, $p \le x \le b$
$= \rho(q - x) / (q - b)$, $b \le x \le q$
$= 0$ otherwise

The family of all (λ, ρ) interval–valued fuzzy numbers be denoted by $F_N(\lambda, \rho)$

3) ***Definition***: Let $\widehat{A}$ = (a, b, c, λ) and $\widehat{B}$ = (p, q, r, λ). We have ,

$$\widehat{A} \oplus \widehat{B} = (a+p, b+q, c+r, \lambda)$$

4) ***Definition*** : Let $\widehat{A} = [\widehat{A}^L, \widehat{A}^U] = [(a_1, b_1, c_1, \lambda), (p_1, b_1, q_1, \rho)]$ and $\widehat{B} = [\widehat{B}^L, \widehat{B}^U] = [(a_2, b_2, c_2, \lambda), (p_2, b_2, q_2, \rho)]$ be two interval- valued fuzzy numbers , then the binary operation $\oplus$ is defined by $\widehat{A} \oplus \widehat{B} = [\widehat{A}^L \oplus \widehat{B}^L, \widehat{A}^U \oplus \widehat{B}^U]$

where, $\widehat{A}^L \oplus \widehat{B}^L = (a_1 + a_2, b_1 + b_2, c_1 + c_2; \lambda)$

and $\widehat{A}^U \oplus \widehat{B}^U = (p_1 + p_2, b_1 + b_2, q_1 + q_2; \rho)$

B. Signed distance ranking of fuzzy numbers:

5) Definition: For b,0 ϵ R , the signed distance of b measured from 0 is defined by $d^*(b, 0) = b$. For any closed interval [a, b] , the signed distance of [a, b] measured from 0 is defined by $d^*([a, b], 0) = (a + b)/2$. For any two disjoint closed intervals [a, b] and [c, d] , the signed distance of [a, b] U [c, d] measured from 0 is defined by

$$d^*([a, b] \cup [c, d], 0) = [d^*([a, b], 0) + d^*([c, d], 0)] / 2$$

For $\widehat{A} = [(a, b, c; \lambda), (p, b, q; \rho)] \in F_N(\lambda, p)$, $0 < \lambda < \rho \le 1$, the signed distanced of $\widehat{A}$ measured from $\widehat{0}$ is defined by $d^*(\widehat{A}, \widehat{0}) = 1/8\,(6b + a + c + 4p + 4q) + 3\lambda/\rho\,(2b - p - q)$

For $\widehat{A} = [(a_1, b_1, c_1; \lambda), (p_1, b_1, q_1; \rho)]$ and $\widehat{B} = [(a_2, b_2, c_2; \lambda), (p_2, b_2, q_2; \rho)]$

$$d^*(\widehat{A} \oplus \widehat{B}), \widehat{0}) = d^*(\widehat{A}, \widehat{0}) + d^*(\widehat{B}, \widehat{0})$$

6) Definition : For $\widehat{A} = [(a_1, b_1, c_1; \lambda), (p_1, b_1, q_1; \rho)]$ and $\widehat{B} = [(a_2, b_2, c_2; \lambda), (p_2, b_2, q_2; \rho)]$, $0 < \lambda < \rho \le 1$, the ranking of interval-valued fuzzy numbers on $F_N(\lambda, \rho)$ are defined by

$\widehat{A} < \widehat{B}$ iff $d^*(\widehat{A}, \widehat{0}) < d^*(\widehat{B}, \widehat{0})$

$\widehat{A} \approx \widehat{B}$ iff $d^*(\widehat{A}, \widehat{0}) = d^*(\widehat{B}, \widehat{0})$

7) Definition: For $0 < \lambda \le 1$, $\widehat{A} = (a, b, c, \lambda) \in F_N(\lambda)$ the signed distance of $\widehat{A}$ measured from $\widehat{0}$ is defined by $d(\widehat{A}, \widehat{0}) = ¼\,(2b + a + c)$.

For , $\widehat{A} = (a, b, c, \lambda)$ and $\widehat{B} = (p, q, r, \lambda) \in F_N(\lambda)$,

$d(\widehat{A} \oplus \widehat{B}), \widehat{0}) = d(\widehat{A}, \widehat{0}) + d(\widehat{B}, \widehat{0})$.

8) Definition : For , $\widehat{A} = (a, b, c, \lambda)$ and $\widehat{B} = (p, q, r, \lambda) \in F_N(\lambda)$, $0 < \lambda \le 1$, the ranking of level λ fuzzy number on $F_N(\lambda)$ are defined by

$\widehat{A} < \widehat{B}$ iff $d(\widehat{A}, \widehat{0}) < d(\widehat{B}, \widehat{0})$

$\widehat{A} \approx \widehat{B}$ iff $d(\widehat{A}, \widehat{0}) = d(\widehat{B}, \widehat{0})$

III. WEIGHTED ROUGH GRAPH

In this section all definitions and discussions are based on the work of T. He *et al* [5] . For the sake of completeness we introduce here.

A. Rough graph.

9) Definition : Let U = (V, E) be the universal graph where $V = \{v_1, v_2, \ldots v_n\}$ is the set of vertices and $E = \cup\, e_k(v_i, v_j)$ is the edge set on U , where the edge e_k is endued with vertex attribute (v_i, v_j). Let $R = \{r_1, r_2 \ldots r_{|R|}\}$ be the attribute set on U. For any attributes set $R \subseteq \mathbf{R}$ on E, the elements of E can be divided into different equivalence classes R(e). For any graph T = (W,X) , where $W \subseteq V$ and $X \subseteq E$, the graph is called R-definable graph or R-exact graph if X is the sum of equivalence classes. Conversely, graph T is called R- undefinable graph or R- rough graph. For R-rough graph , two exact graphs $\underline{R}(T) = (W, \underline{R}(X))$ and $\overline{R}(T) = (W, \overline{R}(X))$ can be used to define it approximately, where

$\underline{R}(X) = \{ e \in E \mid R(e) \subseteq X \}$

$\overline{R}$ (X) = { e∈ E | R(e) ∩ X ≠ϕ }
The graph $\underline{R}$(T) and $\overline{R}$(T) are called R-Lower and R-upper approximation graph of T .The pair of graph ($\underline{R}$(T) , $\overline{R}$(T)) is called R- rough graph. The set $bn_R(X)$ = $\overline{R}$ (X) - $\underline{R}$ (X) is called the R–boundary of edge set X of T.
Property 1: The graph T = ($\underline{R}$(T) , $\overline{R}$ (T)) is exact iff $\underline{R}$(X(T))= $\overline{R}$(X(T)) and is a rough graph iff $\underline{R}$(X(T)) ≠ $\overline{R}$(X(T)).
Property 2 : The rough graph T = ($\underline{R}$(T) , $\overline{R}$(T)) is a classical graph iff all the edges of T belong to the same edge equivalence class with respect to R.

B. The class connection of rough graph

10) Definition : A (v_o,v_n) surely class walk in rough graph T = ($\underline{R}$(T) , $\overline{R}$(T)) is a finite non – null sequence $\underline{w}$ = $v_0 R[e_{v_0 v_1}] v_1\, R[e_{v_1 v_2}] v_2 .....v_{n-1} R[e_{v_{n-1} v_n}] v_n$, whose terms are alternatively vertex and edge equivalence class in $\underline{R}$(T). The surely class walk is called surely class path if $v_i \neq v_j$ for all i ≠ j where i, j = 0,1,2,3,….,n-1 .
Similarly, possibly class walk or class path can be defined analogously if the edge equivalence class are taken from $\overline{R}$(T).
11) Definition :Two vertices u,v are called surely or possibly class connected if there is a (u,v) surely or possibly class path in $\underline{R}$ (T) or $\overline{R}$ (T) respectively.
12) Definition: A (v_o,v_n) surely class path or possibly class path is called a surely class cycle or possibly class cycle if v_o = v_n.
13) Definition : A rough graph is called surely class tree, if it is surely class connected and does not contain surely class cycle. Similarly, a rough graph is called possibly class tree, if it is possibly class connected and does not contain possibly class cycle.
14) Definition: For any rough graph T = ($\underline{R}$(T) , $\overline{R}$(T)) , the surely class tree or possibly class tree H ⊆ T is called surely or possibly class spanning tree if w(T) = w(H), where w(T) is the vertices set of T and w(H) is the vertices set of H

C. Fuzzy weighted rough graph

15) Definition : For the universe graph U = (V,E), for all e∈ E, we define the mapping w: e → w(e) where w(e) is a fuzzy triangular number called the edge weight of e .
w(R(e_{uv})) = f(w(e)) is called the class weight of edge equivalence class R(e_{uv}) , where e ∈ R(e_{uv}) and f is some function. In our work f is taken as sum of all fuzzy weights of the elements of edge equivalence class.

Rough graph with the class weight of their edge equivalence class is called weighted rough graph. The weighted rough graph with fuzzy weights of the edges is called fuzzy weighted rough graph. In fuzzy weighted rough graph, the surely or possibly class spanning tree with the lightest class weight sum is called the surely or possibly class minimum spanning tree.

In this paper we have used signed distance ranking method to rank fuzzy weights and the minimum rank is considered to be the lightest weight in order to find the surely or possibly class minimum spanning tree .

IV. ALGORITHMS

In this section , we give some algorithms to classify the edge sets as edge equivalence classes, lower and upper approximate graph of rough graph and using lower and upper approximate graph, we find the surely or possibly class minimum spanning tree.

let R ={ v_1,v_2,............,v_n} be the finite attribute set on U =(V,E) , the universe graph. For any attribute set R ⊆ **R** on E, E/R is the set of all equivalence classes which makes a partition of E. Two elements e_i, e_j ∈ E where i ≠ j are called in the same kind if R(e_i) = R(e_j).

A. In this section we present the algorithm which gives the classification method to find the edges set of universe graph. In the algorithm three indexes are used : the index i points to the recent input object e_i , the index s records the classes E_1 ,E_2 …………..,E_s which has been found, and the index j is one of the 1,2,…..s which is used to check whether the recent input object e_i satisfies E_j = R (e_i) or not .

Algorithm partition :

Step 1 : i=1 ,j=1, s=1 , E_1 = { e_1 }
Step 2 : if i = | E |, then complete the classification and we have E/R = { E_1, E_2, ..., E_s } else
go to step 3
Step 3 : i = i+1 ,j =1
Step 4 : if j = s , then establish a new class
Step 5 : s = s+1 , E_s = { e_i } and go to step 2 to input the next object
Step 6 : if j < s go to step 7 .
Step 7 : j = j+1 , go to step 8.
Step 8 : if E_j = R (e_i) then e_i ∈ E_j , go to step 2 else go to step 4 to check the next E_j .

B. In this section we present algorithms to find lower and upper approximations. Given rough graph T = (W, X) where W ⊆ V and X ⊆ E , we can get the lower approximation $\underline{R}$(X) and the upper approximation $\overline{R}$ (X) of X with respect to E/R by using the following algorithm lower and algorithm upper respectively so that we can get the lower approximate graph $\underline{R}(T)$ and the upper approximate graph $\overline{R}(T)$. Given the universe graph U = (V,E) , let E/R = {E_1 , E_2 ,………, E_s} ,$1 \leq s \leq |E|$

Algorithm lower :

Step 1 : j =1 , L =ϕ .

Step 2 : if $E_j \subseteq X$, then $L = L \cup E_j$.

Step 3 : if j = s ,then stop and we have $\underline{R}(X)$ = L else go to step 4 to check next E_j

Step 4 : j = j+1 and go to step 2 .

Algorithm upper :

Step 1 : j = 1 , R = ϕ,

Step 2 : if $E_j \cap X \neq \phi$, then $R = R \cup E_j$

Step 3 : if j = s , then stop and we have $\overline{R}(X) = R$

else go to step 4 to check next E_j

Step 4 : j= j+1 and go to step 2.

C. In this section we describe Boruvka's algorithm to find the minimum spanning tree(MST). It is considered as a greedy algorithm like Kruskal's algorithm. However, unlike Kruskal's algorithm, which merges two components in a step, in a single step of Boruvka's algorithm every component is involved in a merger. The basic idea in this algorithm is to contract simultaneously the lightest edges incident to each of the vertices in the graph. For a graph G, we take a forest F initially with all the vertices of G. The algorithm continually adds edges to the forest F and contract G(and simultaneously F) across any of the edges already in F. At the end of the algorithm, F and G are eventually reduced to a single vertex. We must keep a separate list of all edges added to F. When the algorithm ends, the MST is specified by this list. In a contraction, only the lightest edge needs to be retained out of any number of multiple edges.. The process of contracting the lightest edge for each vertex in the graph is called Boruvka step.

The implementation of Boruvka step is the following:

1. Mark the edges to be contracted
2. Determine the connected components formed by the marked edges
3. Replace each connected component by a single vertex
4. Eliminate the self loops and multiple edges created by these contraction
5. If G' be the graph obtained from G after a Boruvka step. The MST of G is the union of the edges marked for contraction during this step with the edges in the MST of G'

Algorithm Boruvka step:

For any graph G = (V , E), let $G_i = (V_i , E_i)$ where $V_i \subseteq V$ and $E_i \subseteq E$

Let F be the forest of MST edges and G' be the contracted graph.

Step 1: F = φ

Step 2: for all $e_{uv} \in E_i$ do
    if $w(e_{uv})$ < u. minWeight then
      u.minWeight = $w(e_{uv})$
      u.minEdge = e_{uv}
    end if

Step 3: for all $e_{uv} \in E_i$ do
    if $w(e_{uv})$ < v. minWeight then
      v.minWeight = $w(e_{uv})$
      v.minEdge = e_{uv}
    end if

Step 4: for all v ε V do
    put v.minEdge in F
    end for

Step 5: G_i = contract (G_i)

Step 6: F = contract (F)

The graph remaining after the i^{th} iteration is the input to the (i+1)st iteration . The iteration continues till the graph left with one vertex on which case the algorithm halts. The output spanning tree is the union of the set of lightest edges selected taken over all iterations. In the implementation of the above algorithm, we need to store fields min Edge and min Edge weight for each vertex.

In our case, as the weights of edges are supposed to be fuzzy triangular numbers, the edge having lower rank between two fuzzy triangular number is considered to be the light weight edge in comparison to the others. We have used signed distance ranking method as discussed in section 2 to rank two triangular fuzzy numbers .

D. Combing the above four algorithms, we give the algorithm MST for exploring the minimum spanning tree in weighted rough graph.

Algorithm MST :

Step 1 : Input a knowledge system (U, R) where U = (V,E) , R = { r_1,r_2…………r_n} , R ⊆ **R** of the rough graph T =(W,X).

Step 2: Apply algorithm partition to E to get all edge equivalence class E/R = {E_1, E_2, …………., E_s}

Step 3 : Apply algorithm lower and algorithm upper to X respectively to get the lower approximate $\underline{R}(X)$ and the upper approximate $\overline{R}(X)$ of X .
step 4 : For the lower approximate graph $\underline{R}$ (T) = (W, $\underline{R}$ (X)) of $\underline{R}$ (T) is still class connected for some edge equivalence class E_i , apply algorithm Boruvka step to $\underline{R}$ (T) corresponding to every E_i , get the surely class spanning tree.
Step 5 : If $\underline{R}$ (T) is not class connected for all edge equivalence class E_j ,j =1,2,……..s then there is no surely class minimum spanning tree .
Step 6 : Apply algorithm Boruvka step to the upper approximate graph $\overline{R}$ (T) = (W, $\overline{R}$ (X)) corresponding to all E_j ,j =1,2,……..s to get the possibly class minimum spanning tree.

V. NUMERICAL EXAMPLE

In this section, we give one numerical example to implement the algorithm discussed. Let there are four water reservoirs A, B, C, D in a city where water is drawn from three rivers and also drawn from the ground. We have defined three kinds of relations such as, two reservoirs are related if they draw water either from the same river or from different rivers or from the ground which are denoted by $\underline{e}_i$, $\bar{e}_j$, e_k respectively. As there are three rivers each relation of first and second kind have three types of sub-relations. With the existing knowledge, we can only know about how many sub-relations exist in every kind, but we cannot distinguish the sub-relationships of the same kind. The purpose is to find the extremely conflicting relation (water is drawn from different rivers) among them. The weight corresponding to each relation is given in form of triangular fuzzy number which denotes the degree of weakness of the relation to make use of our algorithm. The class weight is considered as the sum of the edge weights and comparing the class weights, we use the signed distance ranking method as discussed in previous section. The edge having lesser rank is considered to have less weight.

***Table 1.* All the relationships among the four reservoirs**

	A	B	C	D
A		$\underline{e}_1\underline{e}_2\underline{e}_3\bar{e}_4\bar{e}_5\bar{e}_6e_7$	$\underline{e}_8\underline{e}_9\underline{e}_{10}\bar{e}_{11}\bar{e}_{12}\bar{e}_{13}e_{14}$	$\underline{e}_{15}\underline{e}_{16}\underline{e}_{17}\bar{e}_{18}\bar{e}_{19}\bar{e}_{20}e_{21}$
B	$\underline{e}_1\underline{e}_2\underline{e}_3\bar{e}_4\bar{e}_5\bar{e}_6e_7$		$\underline{e}_{22}\underline{e}_{23}\underline{e}_{24}\bar{e}_{25}\bar{e}_{26}\bar{e}_{27}e_{28}$	$\underline{e}_{29}\underline{e}_{30}\underline{e}_{31}\bar{e}_{32}\bar{e}_{33}\bar{e}_{34}e_{35}$
C	$\underline{e}_8\underline{e}_9\underline{e}_{10}\bar{e}_{11}\bar{e}_{12}\bar{e}_{13}e_{14}$	$\underline{e}_{22}\underline{e}_{23}\underline{e}_{24}\bar{e}_{25}\bar{e}_{26}\bar{e}_{27}e_{28}$		$\underline{e}_{36}\underline{e}_{37}\underline{e}_{38}\bar{e}_{39}\bar{e}_{40}\bar{e}_{41}e_{42}$
D	$\underline{e}_{15}\underline{e}_{16}\underline{e}_{17}\bar{e}_{18}\bar{e}_{19}\bar{e}_{20}e_{21}$	$\underline{e}_{29}\underline{e}_{30}\underline{e}_{31}\bar{e}_{32}\bar{e}_{33}\bar{e}_{34}e_{35}$	$\underline{e}_{36}\underline{e}_{37}\underline{e}_{38}\bar{e}_{39}\bar{e}_{40}\bar{e}_{41}e_{42}$	

***Table 2.* The actual relationships among the four reservoirs**

	A	B	C	D
A		$\underline{e}_1\underline{e}_2\bar{e}_4e_7$	$\underline{e}_9\underline{e}_{10}$	$\underline{e}_{15}\bar{e}_{18}\bar{e}_{19}e_{21}$
B	$\underline{e}_1\underline{e}_2\bar{e}_4e_7$		$\underline{e}_{22}\underline{e}_{23}\bar{e}_{25}e_{28}$	$\underline{e}_{29}e_{35}$
C	$\underline{e}_9\underline{e}_{10}$	$\underline{e}_{22}\underline{e}_{23}\bar{e}_{25}e_{28}$		$\underline{e}_{36}\underline{e}_{37}\bar{e}_{40}\bar{e}_{41}$
D	$\underline{e}_{15}\bar{e}_{18}\bar{e}_{19}e_{21}$	$\underline{e}_{29}e_{35}$	$\underline{e}_{36}\underline{e}_{37}\bar{e}_{40}\bar{e}_{41}$	

***Table 3.* Weights attached to the relations**

e_1	e_2	e_3	$\bar{e}_4$	$\bar{e}_5$	$\bar{e}_6$	e_7
(0.5,0.7,0.9)	(0.3,0.5,0.7)	(0.4,0.6,0.8)	(0.2,0.4,0.6)	(0.3,0.5,0.7)	(0.3,0.5,0.7)	(0.1,0.4,0.6)
e_8	e_9	e_{10}	$\bar{e}_{11}$	$\bar{e}_{12}$	$\bar{e}_{13}$	e_{14}
(0.3,0.5,0.7)	(0.1,0.3,0.5)	(0.2,0.4,0.6)	(0.7,0.8,0.9)	(0.1,0.3,0.5)	(0.2,0.4,0.6)	(0.4,0.6,0.8)
e_{15}	e_{16}	e_{17}	$\bar{e}_{18}$	$\bar{e}_{19}$	$\bar{e}_{20}$	e_{21}
(0.1,0.3,0.5)	(0.7,0.8,0.9)	(0.4,0.6,0.8)	(0.3,0.5,0.7)	(0.1,0.4,0.6)	(0.3,0.5,0.7)	(0.4,0.6,0.8)
e_{22}	e_{23}	e_{24}	$\bar{e}_{25}$	$\bar{e}_{26}$	$\bar{e}_{27}$	e_{28}
(0.4,0.6,0.8)	(0.2,0.4,0.6)	(0.1,0.3,0.5)	(0.7,0.8,0.9)	(0.1,0.3,0.5)	(0.3,0.5,0.7)	(0.1,0.3,0.5)
e_{29}	e_{30}	e_{31}	$\bar{e}_{32}$	$\bar{e}_{33}$	$\bar{e}_{34}$	e_{35}
(0.1,0.3,0.5)	(0.2,0.4,0.6)	(0.7,0.8,0.9)	(0.3,0.5,0.7)	(0.2,0.4,0.6)	(0.7,0.8,0.9)	(0.3,0.5,0.7)
e_{36}	e_{37}	e_{38}	$\bar{e}_{39}$	$\bar{e}_{40}$	$\bar{e}_{41}$	e_{42}
(0.7,0.8,0.9)	(0.1,0.3,0.5)	(0.2,0.4,0.6)	(0.7,0.8,0.9)	(0.2,0.5,0.7)	(0.1,0.3,0.5)	(0.1,0.3,0.5)

Taking A, B, C, D as vertices and $\underline{e}_1$ to e_{42} as edges, we can get the universe graph U. A rough graph X can be obtained according to Table 2. Further we can get weighted universe graph and weighted rough graph by making use of the data given in Table 3. Applying the algorithm MST to the graph we can find the possibly conflicting (water drawn from different rivers) minimum spanning tree among the four reservoirs.

$U = \{ \bar{e}_4, \bar{e}_5, \bar{e}_6, \bar{e}_{11}, \bar{e}_{12}, \bar{e}_{13}, \bar{e}_{18}, \bar{e}_{19}, \bar{e}_{20}, \bar{e}_{25}, \bar{e}_{26}, \bar{e}_{27}, \bar{e}_{32}, \bar{e}_{33}, \bar{e}_{34}, \bar{e}_{39}, \bar{e}_{40}, \bar{e}_{41} \}$
$X = \{ \bar{e}_4, \bar{e}_{18}, \bar{e}_{19}, \bar{e}_{25}, \bar{e}_{40}, \bar{e}_{41} \}$
$\overline{R}(X) = \{ \bar{e}_4, \bar{e}_5, \bar{e}_6, \bar{e}_{18}, \bar{e}_{19}, \bar{e}_{20}, \bar{e}_{25}, \bar{e}_{26}, \bar{e}_{27}, \bar{e}_{39}, \bar{e}_{40}, \bar{e}_{41} \}$

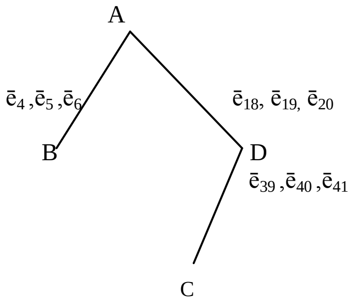

Fig.1 **Possibly conflicting MST**

VI. CONCLUSION

In this paper we have considered the weight of each edge as fuzzy triangular number . The weight can be considered as fuzzy trapezoidal number or interval – valued fuzzy number depending on the nature of the problem and impreciseness. Out of several ranking methods between fuzzy numbers we have considered the signed distance ranking method. Other methods can be used in ranking of fuzzy numbers. This paper intends the application field of rough graph with imprecise of the data.

Further, we have used Boruvka algorithm for finding minimum spanning tree as this method is more suitable for parallel implementation which can be further investigated for rough graph .

REFERENCES

[1]. Jing-Shing Yao and Feng-Tse Lin, " Fuzzy shortest path network problems with uncertain edge weight", Journal of Information Science and Engineering, Vol-19, pp 329-351, 2003
[2]. M. Liang, B. Liang, L. Wei, X. Xu, "Edge rough graph and its application", Proc. Of eighth International Conference on Fuzzy Systems and Knowledge Discovery, pp 335-338, 2011
[3]. R. L. Graham and P. Hall, "On the history of the minimum spanning tree problem", Annals of the History of Computing, Vol-7 No.1, pp 43-57, 1985
[4]. Sun Chung and Anne Condon," Parallel implementation of Boruvka's minimum spanning tree algorithm", Technical report 1297, Computer Science Department, University of Wisconsin, Madison, 1996
[5]. T. He, Y. Chan and K. Shi, "Weighted rough graph and its application", Proceedings of sixth IEEE international conference on intelligent system design and application(ISDA 2006), Vol-1, IEEE Computer society, pp 486-491, 2006
[6]. T. He and K. Shi, " Rough graph and its structure", Journal of Shandong University, Vol-6, pp 88-92, 2006
[7]. T. He, P. Xue and K. Shi, " Application of rough graph in relationship mining", Journal of System Engineering and Electronics, Vol-19, No. 4, pp 742-747, 2008
[8]. Z. Pawlak, " Rough Sets", International Journal of Parallel Programming, Vol-11, No. 5 pp 341-356, 1982.