

# Constructing Spanning Tree in Banking Sector Problem Using Different Types of Algorithms

M.Vijaya<sup>1</sup> and B.Mohanapriya<sup>2\*</sup>

<sup>1,2</sup>P.G and Research Dept. of Mathematics, MaruduPandiyyar College, Vallam, India

Available online at: [www.isroset.org](http://www.isroset.org)

Accepted 18/Aug/2018, Online 30/Aug/2018

**ABSTRACT-** Fuzzy graphs are used as device for modeling and description of the real world. Network systems such are Banking, Transport, water, electricity, internet, work operations schemes in the process of production, construction etc.... Banking system keeps the day by day tally record as a complete banking. It can keep the information of account type, account opening form, deposit, with drawl and searching the transaction. The working of banking account system and cover the basic functionality of a bank account management system. The main focus is to a systematic study of algorithms. Basic banking itself needs to be supported by innovative strategies, in order to improve the reach and reduce the operating cost of the banks. Infrastructure sharing amongst banks and organizations will help in lowering the in commission cost and thus the cost benefit can be transferred to customers. Greater use of technology should be made by the banks to improve their reach, speed of processing, as well as to cut down the operating cost. An illustrative example is included to demonstrate the proposed algorithm. This paper is designed to find a key application of the fund available for investment for the customer to the bank using Nearest Neighbor Algorithm, Repeated Nearest Neighbor Algorithm, Cheapest Link Algorithm and Kruskal's Algorithm to a randomly weighted fuzzy graph. These algorithm continuously increases the size of a fuzzy tree, one edge at a time, starting with a fuzzy tree consisting of a single vertex, until it spans all vertices of the fuzzy graph.

**KEYWORDS:** Fuzzy Spanning Tree-Nearest Neighbor Algorithm-Repeated Nearest Neighbor Algorithm-Cheapest Link Algorithm-Kruskal's Algorithm.

## I. INTRODUCTION

In the recent years, with the increase of using internet and other new telecommunication technologies etc. has become a key area to research and improve in order to transfer data securely between two or more entities especially when the data transferred classified as a critical or important data Commercial banks constitute the most important function in the whole network of financial system for mobilization of saving, intermediation between savers and investors and allocation of credit to productive sectors and thus play a dynamic role in the economic development of a nation.

Credit is a phenomenon of the economic and banking system plays a crucial rule in this process. Credit is the important determinants of money creation, and hence of production, consumption and national income.

As the commercial banks are large organizations having a broader range of activities, they provide placement for internship in different branches and department. Most of the banks have the specified schedule for the internship and they are placed in different departments. This rotation concept helps the internship to gain overall information about the various types of functions performed in different departments.

Fuzzy graphs are excellent structures for storing; searching and retrieving large amounts of data play an important role in increasing the storage/ search efficiency of computational techniques. By reducing an instance of a problem to a undirected fuzzy graph problem. We may be able to use well known graph algorithms to provide an optimal.

To find the shortest paths in weighted graphs is one of the archetypical problems encountered in the domain of combinatorial optimization. More recently, fuzzy weighted graphs, along with generalizations of algorithm for finding optimal paths within them, have emerged as an adequate modeling tool for prohibitively complex or inherently methods used for path

comparison. The numbers of optimization criteria they impose, realistic environments are often so complex that even the most sophisticated implementation become computationally unmanageable. The method based on the very popular center of gravity defuzzification method has some defects that make it unsuitable for the purposes of the fuzzy shortest path algorithm of a defuzzification method is used as a basics for ranking fuzzy numbers a significant optimization by defuzzifying all. Fuzzy weights before algorithm execution is possible under certain contain on the involved defuzzication operators.

A less obvious application is that the minimum fuzzy spanning tree can be used to approximately solve the traveling salesman problem. A convenient formal way of defining this problem is to find the shortest path that visits each point exactly once this problem can be solved by many different algorithms, which are recently developed in research topics. There are several best algorithms depending on the following assumptions:

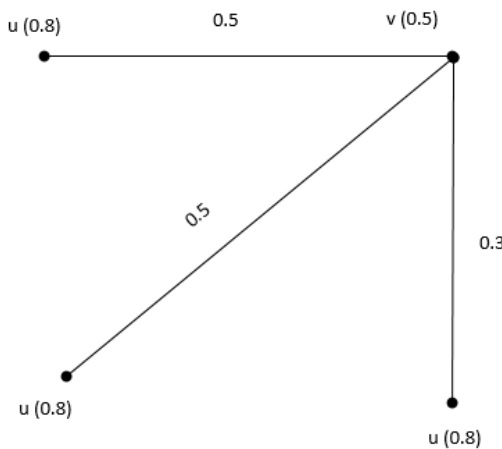
- 1) A randomized algorithm can solve it in linear expected time.
- 2) It can be solved in worst case time if the weights are large integers.
- 3) Select the minimum weightage unused edge in the fuzzy graph; highlight it.

## II. DEFINITIONS

**Definition 2.1:**

A fuzzy graph with  $V$  as the underlying set is a pair  $G: (A, \Gamma)$  where  $A: V \rightarrow [0, 1]$  is a fuzzy subset,  $\Gamma: V \times V \rightarrow [0, 1]$  is a fuzzy relation on the fuzzy subset  $A$ , such that  $\Gamma(u, v) \leq A(u) \cap A(v)$  for all  $u, v \in V$ .

**Example 2.1:**



**Fig 2.1**

**Definition 2.2:**

A directed fuzzy walk in a fuzzy graph is an alternating sequence of vertices and edges,  $x_0, e_1, x_1, \dots, e_n, x_n$  in which each edge  $e_i$  is  $x_{i-1}x_i$ .

**Definition 2.3:**

A fuzzy path is a fuzzy walk in which all vertices are distinct.

**Definition 2.4:**

A fuzzy path from  $u$  to  $v$ , the  $v$  is said to be reachable from  $u$ , and the distance,  $D(u, v)$ , from  $u$  to  $v$  is the length of any shortest such fuzzy path.

**Definition 2.5:**

A fuzzy path between the point  $s$  (source) to  $t$  (sink) of fuzzy graph  $G$  is called fuzzy policy or fuzzy tree.

**Definition 2.6**

A fuzzy Hamiltonian path that passes through each of the vertices in a fuzzy graph exactly once.

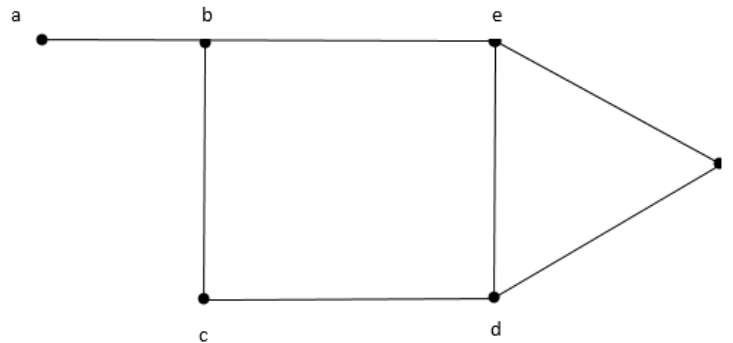
**Definition 2.7:**

A fuzzy Hamiltonian circuit is a circuit that visits every vertex in a fuzzy graph once with no repeats, being a fuzzy Hamiltonian circuit must start and end at the same vertex.

**Definition 2.8:**

A fuzzy spanning tree is a fuzzy tree which covers all the vertices of a fuzzy graph.

**Example 2.2:**



**Fig 2.2**

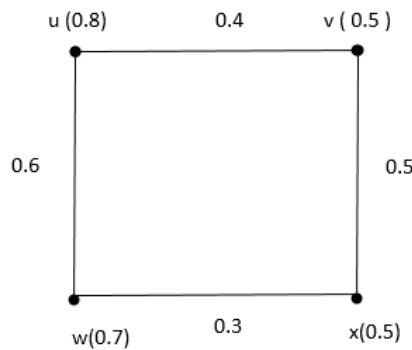
**Note 2.1**

Fuzzy trees have no circuits, and it is fine to have vertices with degree higher than two.

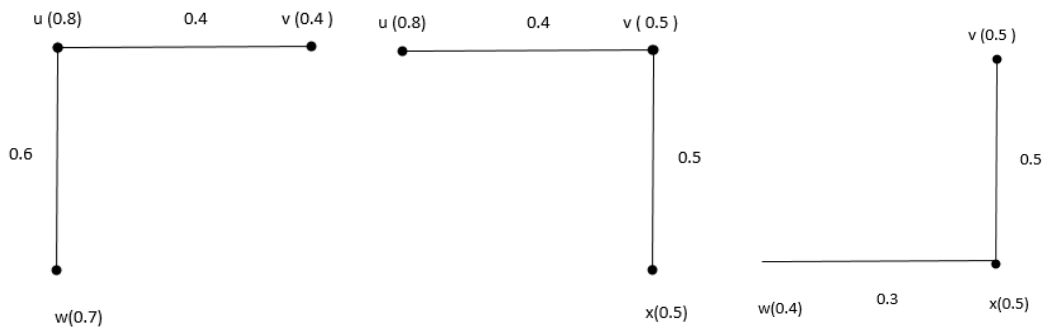
**Example 2.3**

A fuzzy spanning tree of a fuzzy graph is just a fuzzy sub graph that contains all the vertices and is a fuzzy tree. A fuzzy graph may have many fuzzy spanning trees; for instance the complete fuzzy graph on  $n$  vertices has  $n(n-1)/2$  fuzzy spanning trees:

Forming the spanning tree by finding a fuzzy sub graph a new fuzzy graph formed using all the vertices and some of the edges from the original fuzzy graph



**Fig 2.3**



**Fig a**

**Fig b**

**Fig c**

III. TYPES OF ALGORITHMS

Example 3.1

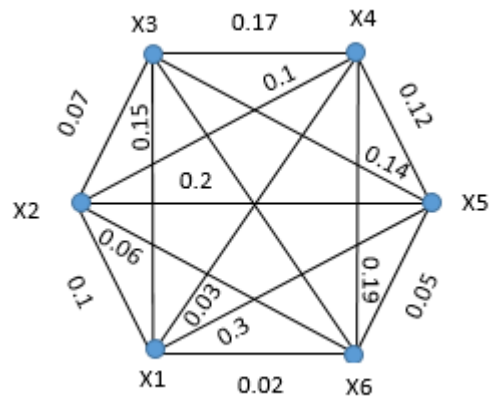


Fig 3.1

Brute Force Algorithm

- Step1: List all possible fuzzy Hamiltonian circuits.
- Step 2: Find the length of each circuit by adding the edge weights.
- Step 3: Select the circuit with minimal total weight.

SL.NO	Fuzzy Hamiltonian circuits	Weights
1.	$X_1X_2X_3X_4X_5X_6X_1$	0.53
2.	$X_1X_6X_3X_4X_2X_5X_1$	0.89
3.	$X_1X_2X_5X_6X_3X_4X_1$	0.65
4.	$X_1X_5X_3X_2X_6X_4X_1$	0.79
5.	$X_1X_5X_3X_6X_4X_2X_1$	0.93
6.	$X_1X_4X_5X_6X_2X_3X_1$	0.76

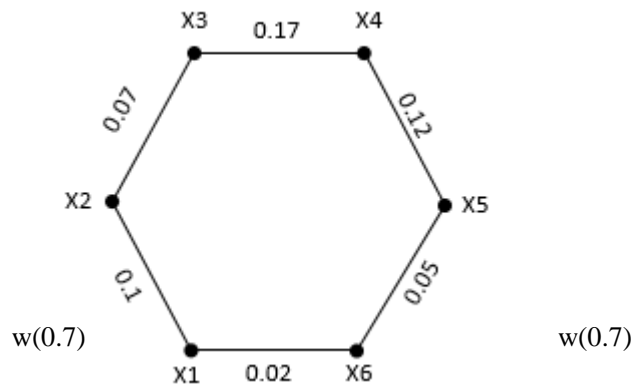


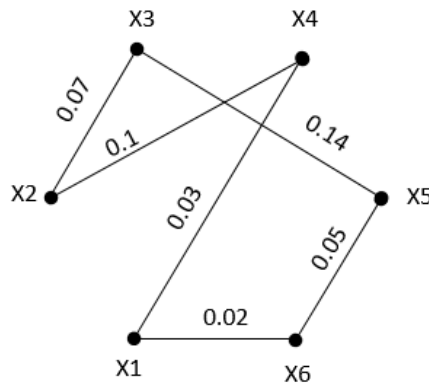
Fig 3.2

There are unique circuits on this fuzzy graph. All other circuits are the reverse of the listed ones or start at a different vertex, but result in the same weights. From this we can see that the first circuit,  $X_1X_2X_3X_4X_5X_6X_1$ , is the fuzzy optimal circuit.

**Nearest Neighbor Algorithm (RNNA)**

- Step1: Select a starting point.
- Step2: Move to the nearest unvisited vertex (the edge with Smallest weight).
- Step3: Repeat until the circuit is complete.

Consider Fig 3.1, starting at vertex  $X_4$ , the nearest neighbor is vertex  $X_1$  with a weight of 0.03. From  $X_1$ , the nearest neighbor is  $X_6$  with a weight of 0.02. From  $X_6$ , the nearest neighbor is  $X_5$  with a weight of 0.05. From  $X_5$ , the nearest neighbor is  $X_4$  with a weight of 0.12 but it complete a circuit without visiting other neighbor. So, the nearest neighbor is  $X_3$  with a weight of 0.14. From  $X_3$ , the only unvisited vertex  $X_2$  with weight 0.07. From  $X_2$ , we return to  $X_4$  with a weight 0.1. The resulting circuit  $X_4X_1X_6X_5X_3X_2X_4$  is with a total weight of 0.41.



**Fig 3.3**

**Repeated Nearest Neighbor Algorithm**

- Step1: Start at X. Apply nearest neighbor algorithm.
- Step2: Repeated process using each of the other vertices.
- Step3: Of the circuits obtained use the cheapest.
- Step4: Of the nearest neighbor circuits obtained keep the best one.
- Step5: If there is a designated starting vertex, rewrite the circuit using that vertex as the reference point.

Consider fig 3.1, starting at vertex  $X_1$ , the nearest neighbor is vertex  $X_6$  with a weight 0.02. From  $X_6$  the nearest neighbor is vertex  $X_5$  with a weight 0.05. From  $X_5$ , the nearest neighbor is vertex  $X_4$  with a weight 0.12. From  $X_4$ , the nearest neighbor is vertex  $X_3$  with a weight 0.17. From  $X_3$ , the nearest neighbor is vertex  $X_2$  with a weight 0.07. From  $X_2$ , we return to  $X_1$  with a weight 0.1. The resulting circuit  $X_1X_6X_5X_4X_3X_2X_1$  is with a total weight of 0.53.

Similarly, starting at vertex  $X_2$ , the resulting circuit  $X_2X_6X_5X_4X_1X_3X_2$  is with a total weight of 0.48. Starting at vertex  $X_3$ , the resulting circuit  $X_3X_2X_6X_5X_4X_1X_3$  is with a total weight of 0.48. Starting at vertex  $X_4$ , the resulting circuit  $X_4X_1X_6X_5X_3X_2X_4$  is with a total weight of 0.41. Starting at vertex  $X_5$ , the resulting circuit  $X_5X_4X_1X_6X_2X_3X_5$  is with a total weight of 0.44. Starting at vertex  $X_6$ , the resulting circuit  $X_6X_1X_4X_5X_3X_2X_6$  is with a total weight of 0.56.

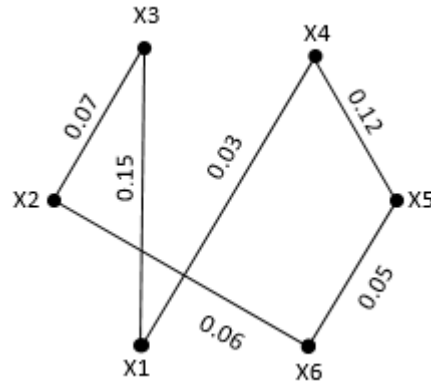


Fig 3.4

Using the Repeated Nearest Neighbor Algorithm the minimum resulting circuit is vertex  $X_4$  with a total weight of 0.41.

**Cheapest Link Algorithm**

1. Step1: Select the cheapest unused edge in the graph.
2. Step2: Repeat step1, adding the cheapest unused edge to the circuit unless:
  - a) Adding the edge would create a circuit that doesn't contain all vertices (or)
  - b) Adding the edge would give a vertex degree 3.
3. Repeat until a circuit containing all vertices is formed.

Consider fig 3.1, we have six vertex fuzzy graph, the minimum weight edge is  $X_1X_6$  with a weight of 0.02. The next fuzzy shortest edge is  $X_6X_5$  with a weight 0.05. The next fuzzy shortest edge is  $X_5X_4$  with a weight 0.12. The next fuzzy shortest edge is  $X_3X_4$  with a weight 0.17. The next fuzzy shortest edge is  $X_3X_2$  with a weight 0.07. The next fuzzy shortest edge is  $X_2X_6$ , but the edge would create a fuzzy Hamiltonian circuit  $X_1X_6X_5X_4X_3X_2X_6$ .

**Example 3.2**

In a private bank, there are connected with many services like, the Deposit Products, the Loan Products, the Investments Products, Lockers, Foreign Currency Buy/Sell, CMS Services, Tax Collection, Internet Banking, etc... We take the deposit products to plan and connect an efficient fuzzy route for raising domestic revenue and to gear up through a dynamic credit policy to ensure that the fund available for investments to the customers, it reach to the all type account once and return to the starting location (E.g.: Deposit Accounts). Consider vertex such as  $X_1$ -Deposit Accounts,  $X_2$ - Preferred/ Royale Accounts,  $X_3$ - Senior Citizens Accounts,  $X_4$ - Women's Accounts,  $X_5$ - Youth Accounts,  $X_6$ - Kids accounts,  $X_7$ - Current accounts,  $X_8$ - Fixed Deposits RD,  $X_9$ - Senior Citizens FD,  $X_{10}$ - Super Saving Accounts ,  $X_{11}$ - Tax Saving Deposits,  $X_{12}$ - Salary Accounts. The deposit percentages are represented as fuzzy weights in the tabular column:

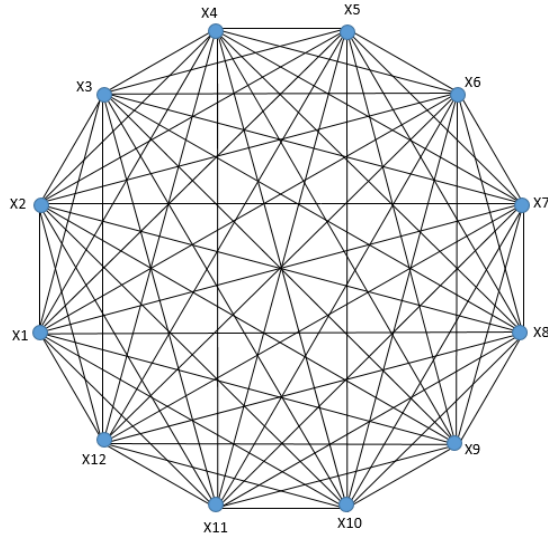


Fig 3.5

TABLE 3.1

	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	X <sub>7</sub>	X <sub>8</sub>	X <sub>9</sub>	X <sub>10</sub>	X <sub>11</sub>	X <sub>12</sub>
X <sub>1</sub>	-	0.24	0.17	0.28	0.35	0.25	0.2	0.12	0.06	0.1	0.21	0.11
X <sub>2</sub>	0.24	-	0.13	0.06	0.08	0.27	0.16	0.07	0.19	0.43	0.18	0.01
X <sub>3</sub>	0.17	0.13	-	0.15	0.34	0.12	0.03	0.11	0.25	0.05	0.27	0.31
X <sub>4</sub>	0.28	0.06	0.15	-	0.22	0.08	0.15	0.33	0.27	0.1	0.47	0.09
X <sub>5</sub>	0.35	0.08	0.34	0.22	-	0.3	0.14	0.09	0.23	0.48	0.19	0.1
X <sub>6</sub>	0.25	0.27	0.12	0.08	0.3	-	0.06	0.21	0.13	0.08	0.1	0.2
X <sub>7</sub>	0.2	0.16	0.03	0.15	0.14	0.06	-	0.11	0.02	0.22	0.12	0.17
X <sub>8</sub>	0.12	0.07	0.11	0.33	0.09	0.21	0.11	-	0.38	0.27	0.13	0.05
X <sub>9</sub>	0.06	0.19	0.25	0.27	0.23	0.13	0.02	0.38	-	0.19	0.04	0.09
X <sub>10</sub>	0.1	0.43	0.05	0.1	0.48	0.08	0.22	0.27	0.19	-	0.43	0.06
X <sub>11</sub>	0.21	0.18	0.27	0.47	0.19	0.1	0.12	0.13	0.04	0.43	-	0.03
X <sub>12</sub>	0.11	0.01	0.31	0.09	0.1	0.2	0.17	0.05	0.09	0.06	0.03	-

Using Nearest Neighbor Algorithm (NNA) starting at  $X_8$  with a fixed deposit RD. Looking in the row for  $X_8$ , the smallest distance is 0.05 to  $X_{12}$ . Following that idea, our circuit will be:

$X_8$ to $X_{12}$	0.05
$X_{12}$ to $X_2$	0.01
$X_2$ to $X_4$	0.06
$X_4$ to $X_5$	0.08
$X_5$ to $X_7$	0.14
$X_7$ to $X_9$	0.02
$X_9$ to $X_{11}$	0.1
$X_{11}$ to $X_6$	0.08
$X_6$ to $X_{10}$	0.05
$X_3$ to $X_1$	0.17
$X_1$ to $X_8$	0.12

Therefore, total connecting length is 1 (approximately).

Using Repeated Nearest Neighbor Algorithm (RNNA) starting at  $X_1$  with a super saving accounts. Looking in the row for  $X_1$ , smallest distance is 0.06 to  $X_9$ . Following that idea our circuit will be:

$X_1$ to $X_9$	0.06
$X_9$ to $X_7$	0.02
$X_7$ to $X_3$	0.03
$X_3$ to $X_{10}$	0.05
$X_{10}$ to $X_{12}$	0.06
$X_{12}$ to $X_2$	0.01
$X_2$ to $X_4$	0.06
$X_4$ to $X_6$	0.08
$X_6$ to $X_{11}$	0.1
$X_{11}$ to $X_5$	0.19
$X_5$ to $X_8$	0.09
$X_8$ to $X_1$	0.12

Therefore, the total profit of the accounts is 0.87.

Similarly starting at vertex  $X_2, X_3, X_5, X_6, X_8, X_9, X_{10}, X_{12}$ , the resulting circuit with the total profit is 1 (approximately). Starting at vertex  $X_4$ , the resulting circuit with the total profit is 0.92. Starting at vertex  $X_7, X_{11}$ , the resulting circuit with the total profit are 0.83, 0.95.

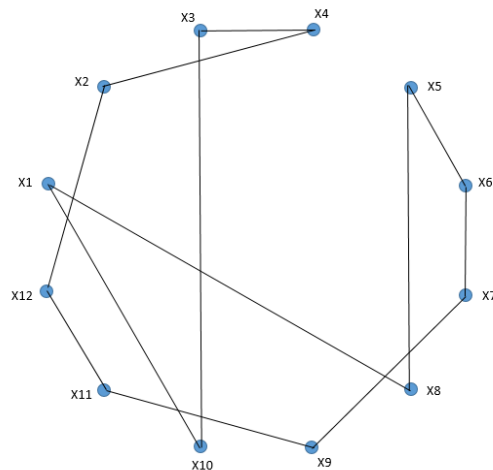


Fig 3.6

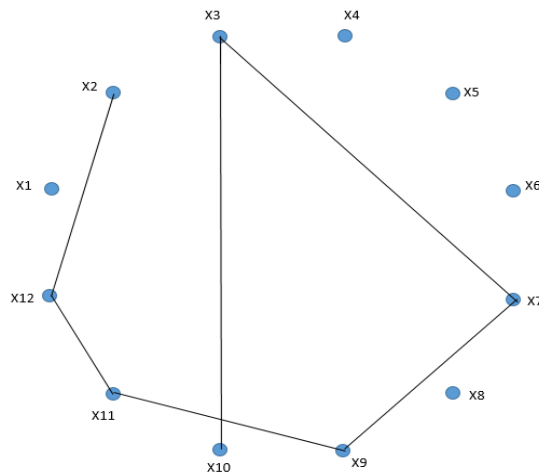


Using the Repeated Nearest Neighbor Algorithm the minimum resulting circuit is vertex  $X_7$  with a total profit of 0.83.

Using Cheapest Link Algorithm is helpful to draw a fuzzy graph perhaps by drawing vertices in a circular pattern. Adding edges to the fuzzy graph and select them with help of any fuzzy circuits or vertices with degree 3.

$X_{12}$ to $X_2$	0.01
$X_7$ to $X_9$	0.02
$X_{12}$ to $X_{11}$	0.03
$X_3$ to $X_7$	0.03
$X_{11}$ to $X_9$	0.04
$X_{10}$ to $X_3$	0.05

The fuzzy graph after adding these edges is shown in the following figure:

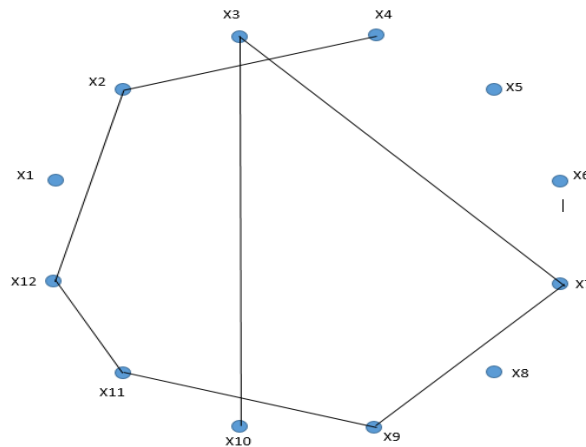


**Fig 3.7**

The next shortest edge is from  $X_8$  to  $X_{12}$  at 0.05, but adding that edge would give  $X_{12}$  degree 3. Continuing on and skip over any edge pair that contains  $X_9$  to  $X_1$ , since they both already have degree 2.

$X_4$ to $X_2$	0.06
----------------	------

The fuzzy graph after adding these edges is shown in the following figure:



**Fig 3.8**

At this point, skip over any edge pair that contains  $X_7, X_{12}, X_2$ , since they already have degree 2.

- $X_6$  to  $X_4$         0.08
- $X_6$  to  $X_{10}$       (reject – it forms fuzzy circuit)
- $X_5$  to  $X_8$         0.09
- $X_4$  to  $X_{12}$       (reject – it forms fuzzy circuit)
- $X_9$  to  $X_{12}$       (reject – it forms fuzzy circuit)
- $X_{10}$  to  $X_1$        0.1
- $X_5$  to  $X_6$         0.1

The fuzzy graph after adding these edges is shown in the following figure:

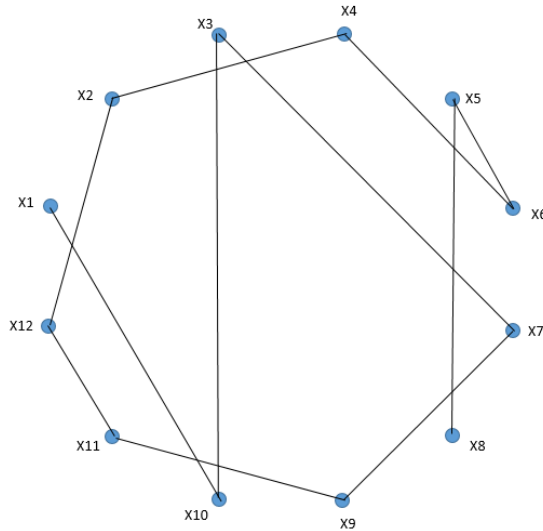


Fig 3.9

At this point, skip over any edge pair that contains  $X_{10}, X_{11}, X_{12}$ , since they already have degree 2.

At this point the only way to complete the circuit is to add:

- $X_8$  to  $X_1$         0.12

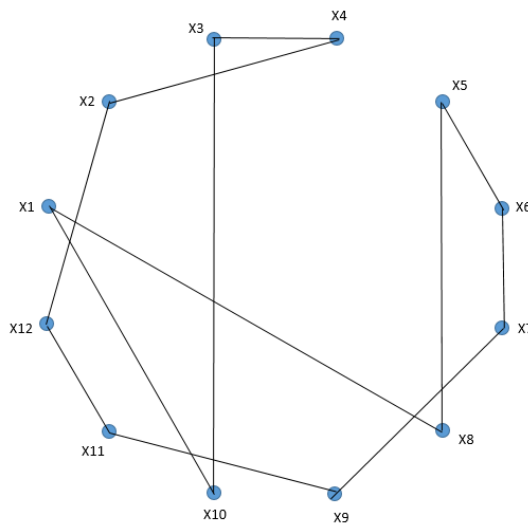


Fig 3.10

Therefore, total profit of the accounts is 0.73

#### IV. KRUSKAL'S ALGORITHM

- Step1: Select the minimum weight unused edge in the fuzzy graph; highlight it.
- Step2: Repeat step1, addition of minimum weight unused edge to the fuzzy graph,  
 Unless: Addition of edge would create a fuzzy circuit.
- Step3: Repeat until a fuzzy spanning tree is formed.

##### Example 4.1

In a bank, we take the deposit products to plan and connect an efficient fuzzy route for raising domestic revenue and to gear up through a dynamic credit policy to ensure that the fund available for investments to the customers, it reach to the all type account once and return to the starting location (E.g.: Deposit Accounts). Consider vertex such as X<sub>1</sub>-Deposit Accounts, X<sub>2</sub>- Preferred/ Royale Accounts, X<sub>3</sub>- Senior Citizens Accounts, X<sub>4</sub>- Women's Accounts, X<sub>5</sub>- Youth Accounts, X<sub>6</sub>- Kids accounts, X<sub>7</sub>- Current accounts, X<sub>8</sub>- Fixed Deposits RD, X<sub>9</sub>- Senior Citizens FD, X<sub>10</sub>- Super Saving Accounts , X<sub>11</sub>- Tax Saving Deposits, X<sub>12</sub>- Salary Accounts. The deposit percentages are represented as fuzzy weights in the tabular column3.1. The problem is to solve using Kruskal's Algorithm with fuzzy graph. Selecting the cheapest unused edge in the fuzzy graph, we have

X <sub>12</sub> to X <sub>2</sub>	0.01
X <sub>9</sub> to X <sub>7</sub>	0.02
X <sub>12</sub> to X <sub>11</sub>	0.03
X <sub>7</sub> to X <sub>3</sub>	0.03
X <sub>11</sub> to X <sub>9</sub>	0.04
X <sub>3</sub> to X <sub>10</sub>	0.05
X <sub>12</sub> to X <sub>8</sub>	0.05
X <sub>9</sub> to X <sub>1</sub>	0.06
X <sub>2</sub> to X <sub>4</sub>	0.06
X <sub>6</sub> to X <sub>7</sub>	0.06
X <sub>10</sub> to X <sub>12</sub>	(reject – it forms fuzzy circuit)
X <sub>4</sub> to X <sub>6</sub>	(reject – it forms fuzzy circuit)
X <sub>6</sub> to X <sub>10</sub>	(reject – it forms fuzzy circuit)
X <sub>2</sub> to X <sub>5</sub>	0.08

Accepted path is connected and shown in the following figure 4.1

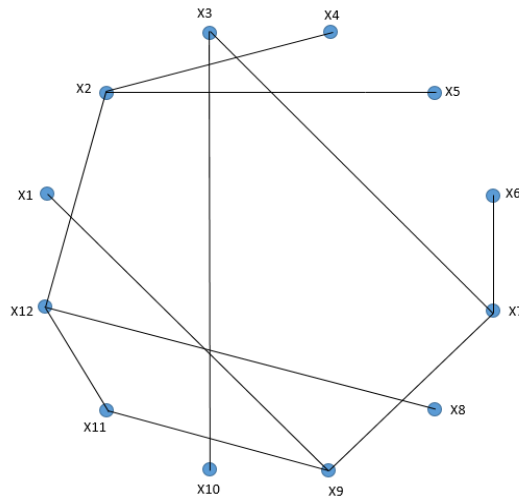


Fig 4.1

This connects the fuzzy graph. The total benefit of bank to lay would be 0.56.

## V. CONCLUSION

Bank is a business of accepting deposits and lending money. It is carried out by financial intermediaries, which performs the function of safe guarding deposits and providing loans to the public. The result on comparing with heuristic algorithm such as Nearest Neighbor Algorithm, Repeated Nearest Neighbor Algorithm, Cheapest Link Algorithm and Kruskal's Algorithm for the problem that connect an efficient fuzzy route for raising domestic income or geared up through a dynamic credit policy to make the fund available in all criteria to reach all the accounts department exactly once is Kruskal's Algorithm. The research showed that it is necessary to choose the most comfortable fuzzy route for the bank sector with increasing liberalization and higher economic growth the role of banking sector is perched to increase in the financing pattern of economic actives in the country. To assemble the growing credit demand, the banks need to organize resources from a wider deposit base and extend credit to activities previously not financed by banks. The trend of growing commercialization of agriculture and rural activities should produce greener pastures and bank should examine the benefits of increasing saturation there in. financial inclusion will strengthen financial deepening and provide resources to the banks to increase credit release. So we can conclude that Kruskal's Algorithm is the best to adopt for these types of problem.

## REFERENCES

- [1]. A.NAGOORGANI and V.T.CHANDRASEKARAN: A First Look at Fuzzy Graph Theory Allied Publishers Pvt. Ltd.
- [2]. A.NAGOORGANI and V.T.CHANDRASEKARAN: An Algorithm on domination Number of a graph, International Journal of Mathematics and Computer Science, Lebanon, 2(2007), no.3,231-235.
- [3]. DR.G.NIRMALA and K.UMA: Fizzy Shortest Route Algorithm For Telephone Line Connection, International Journal of Scientific and Research Publication, Volume 2, Issue 8, August2012.
- [4]. H.J.ZIMMERMANN: Fuzzy set theory and its Application. Allied Publishers Limited, 1996.
- [5]. L.A.ZADEH: Fuzzy logic and the calculi of fuzzy rules and fuzzy graphs, a précis, Multi. Val. Logic, 1:1-38, 1996.
- [6]. CHRISTOFIDES, N: Graph Theory: an algorithm approach. Academic Press, London, 1975.
- [7]. HALPERN.J, MAIMAON.O: Algorithm for the m-center problems; a survey. European J. Operational Res. 10, 1982, 90-99.
- [8]. T.N. Chuang, J.Y.Kung, The fuzzy shortest path length and the corresponding shortest path in a network, Computers & Operations Research 32 (2005) 1409-1428.
- [9]. C.lin, M.S. Chern, The fuzzy shortest path problem and its most vital arcs, Fuzzy Sets and Systems 58 (1993) 343-353.