ALGORITHMS FOR SOME FUZZY NETWORK PROBLEMS USING RANKING FUNCTION

Thesis submitted in partial fulfillment of the requirement for
the award of the degree of
Masters of Science
in

Mathematics and Computing

Submitted by

Neha Bhatia

Roll no. 30703012

Under the guidance of

Dr. Amit Kumar

JULY 2009

School of Mathematics and Computer Applications

Thapar University

Patiala-147004 (PUNJAB)

INDIA

DEDICATED
TO
GOD, MY PARENTS AND MY SUPERVISOR

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled **"Algorithms for some Fuzzy Network Problems using Ranking Function"** in partial fulfillment of the requirements for the award of degree of Master of Science, School of Mathematics and Computer Applications, Thapar University, Patiala is an authentic record of my own work carried out under the supervision of **Dr. Amit Kumar**.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

(Neha Bhatia)
30703012

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

14/07/09

(Dr. Amit Kumar)
Supervisor
Lecturer, SMCA
Thapar University, Patiala.

Countersigned by:

Dr. S.S. Bhatia 14.7.09
(Professor & Head)
School of Mathematics & Computer Applications
Thapar University, Patiala.

Dr. R.K. Sharma
Dean of Academic Affairs
Thapar University
Patiala.

List of Research Papers

1. "A new algorithm for the fuzzy maximal flow problem", Eighteenth International conference of Forum of Interdisciplinary Mathematics on Interdisciplinary Mathematical and Statistical Techniques, August 2-4, 2009, Jaypee University of Information Technology, Waknaghat, Distt. Solan, (Near Shimla), Himachal Pradesh, INDIA. (**Presented**)
2. "A new algorithm for the fuzzy shortest path problem", 12th International Conference on Rough Sets, Fuzzy Sets, Data Mining &Granular Computing (RSFDGrC 2009), December 16-18, 2009, I.I.T Delhi. (**Communicated**)
3. "A new algorithm for the fuzzy minimal spanning tree problem", International conference on Statistics, Probability, Operations Research, Computer Science and allied Areas in conjunction with 8th International Indian Statistical Association (IISA), January 4-8, 2010, Department of Mathematics, Andhra University, Visakhapatnam. (**Communicated**)

Chapter 1

ACKNOWLEDGMENT

First of all, I would like to thank the Almighty for granting perseverance. I would like to express my gratitude to **Dr**. **Amit Kumar**, **Lecturer**, **SMCA**, **Thapar University**, **Patiala**, for their patient guidance and support throughout this work. I was truly very fortunate to have the opportunity to work under him as a student. It was both an honour and a privilege to work with him. He also provides help in technical writing and presentation style and I found this guidance to be extremely valuable. I take this opportunity to express my sincere thanks to **Prof**. **S.S**. **Bhatia**, **Head SMCA**, **Thapar University**, **Patiala**, for their valuable support and help without which it would not have been possible for me to complete this work.

I am also thankful to all my friends who devoted their valuable time and helped me in all possible ways towards successful completion of this work. I do not find enough words with which I can express my feeling of thanks to the entire faculty and staff of SMCA, Thapar University, Patiala, for their help, inspiration and moral support which went a long way in successful completion of my work. I thank all those who have contributed directly or indirectly to this work.

Lastly, and more importantly, I would like to thank my parents for their years of unyielding love and encouragement. They have always wanted the best for me and I admire my parent's determination and sacrifice to put me through college.

Place: Patiala
Dated: July 14, 2009 (Neha Bhatia)

ABSTRACT

This thesis is devoted to different type of fuzzy networks which occur in real life problems. The analysis of these networks has been done from different angles. The main topics are Fuzzy minimal spanning tree problem, Fuzzy shortest path problem and Fuzzy maximal flow problem.

The chapter-wise summary of the thesis is as follows:

Chapter 1 is introductory in nature. This chapter includes basic definitions, operations and concepts used throughout the work.

Chapter 2 presents a brief review of the work done in the area of fuzzy minimal spanning tree problem, fuzzy shortest path problem and fuzzy maximal flow problem.

In **Chapter 3** the fuzzy minimal spanning tree problem in a given connected graph is considered. It is assumed that the edge costs are not precisely known and they are specified as triangular fuzzy numbers. A new algorithm has been proposed to characterize the optimality of edges of the graph and to choose a spanning tree under fuzzy costs and results are discussed based on the present study, conclusions are drawn.

In **Chapter 4** shortest path problem in a given connected graph is considered. Shortest path problem where the approximate costs are known is one of the most studied problems in fuzzy sets and systems area. In this chapter, we have introduced an algorithm that assumes a ranking function for comparing the fuzzy numbers.

In **chapter 5** the problem of finding the maximum flow between a source and a destination node in a network with uncertainties in its capacities is considered and an algorithm based on the classical algorithm is proposed.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction to networks

A network consists of a set of nodes linked by arcs [14]. Here we use network in more generalized context. First we define certain terms in reference to networks. The simple notation for a network is (N, A), where N is number of nodes and A is the set of arcs. For example, in Figure 1.1

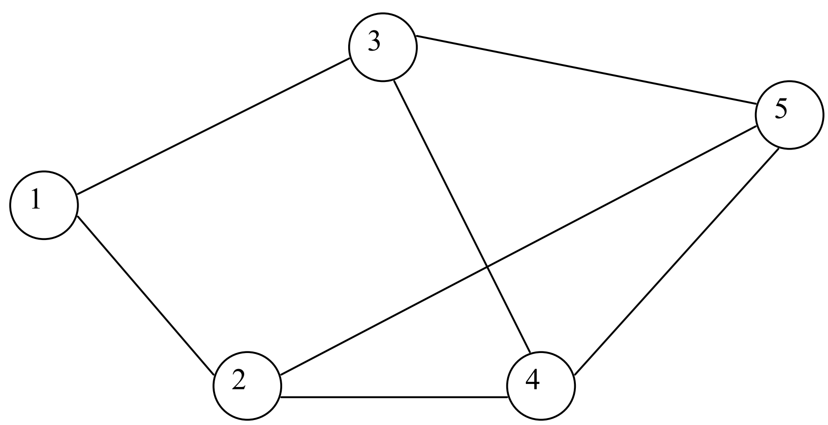


Figure 1.1

$N = \{1, 2, 3, 4, 5\}$, $A = \{(1,3),(1,2),(2,4),(2,5),(3,4),(3,5),(4,5)\}$. Each arc is assigned a capacity which may be flow, cost, distance etc. The capacity of an arc is finite or infinite. An arc is directed if it allows positive flow in one direction and zero flow in opposite direction. Hence, by directed network we mean that all arcs are directed.

1.2 Basic definitions

In this section some basic definitions [11,13,16,33] are reviewed.

Definition 1.1

A crisp set or a classical set A is defined as a collection of distinct and distinguishable objects. The objects are called elements of A.

Definition 1.2

A crisp set A, defined on the universal set X, can also be represented by $A = \{(x, \mu_A(x)); x \in X\}$ where $\mu_A : X \to \{0,1\}$ is called characteristic function defined by

$$\mu_A(X) = \begin{cases} 1 \; if \; X \in A \\ 0 \; if \; X \notin A \end{cases}$$

Definition 1.3

The characteristic function μ_A of a crisp set $A \subseteq X$ assigns a value either 0 or 1 to each member in X. This function can be generalized to a function $\mu_{\tilde{A}}$ such that the value assigned to the element of the universal set X fall within a specified range $[0,1]$ i.e. $\mu_{\tilde{A}} : X \to [0,1]$. The assigned values indicate the membership grade of the element in the set A.

The function $\mu_{\tilde{A}}$ is called the membership function and the set $\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) : x \in X\}$ defined by $\mu_{\tilde{A}}$ for each $x \in X$ is called a fuzzy set.

Definition 1.4

Let $\tilde{A}$ be a fuzzy set and α be a real number in the interval $[0,1]$. The crisp set A_α defined by $A_\alpha = \{x \in X : \mu_{\tilde{A}}(x) \geq \alpha\}$ is called $\alpha -$ **cut** of $\tilde{A}$.

The crisp set $A_{\alpha^+} = \{x \in X : \mu_{\tilde{A}}(x) > \alpha\}$ is called strong $\alpha -$ **cut** of $\tilde{A}$.

Definition 1.5

A fuzzy set $\tilde{A}$, defined on the universal set *X*, is said to be convex if

$\mu_{\tilde{A}}(\alpha_1 x_1 + \alpha_2 x_2) \geq \min\{\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2)\} \forall x_1, x_2 \in X$ and $\alpha_1, \alpha_2 \geq 0, \quad \alpha_1 + \alpha_2 = 1$.

Definition 1.6

A fuzzy set $\tilde{A}$, defined on the universal set of real numbers *R*, is said to be a fuzzy number if its membership function has the following characteristics:

1. $\mu_{\tilde{A}} : R \to [0,1]$ is continuous.
2. $\mu_{\tilde{A}}(x) = 0$ for all $x \in (-\infty, c] \cup [d, \infty)$.
3. Is strictly increasing on $[c, a]$ and strictly decreasing on $[b, d]$.
4. $\mu_{\tilde{A}}(x) = 1$ for all $x \in [a, b]$.

Definition 1.7

A fuzzy number $\tilde{A} = (a, b, c)$, shown in Figure 1.2, is said to be a triangular fuzzy number if its membership function is given by

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

where $a, b, c, d \in R$

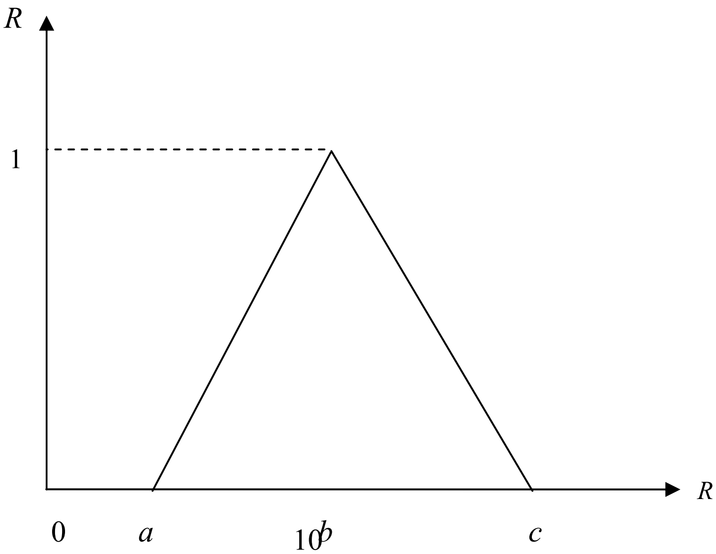


Figure 1.2 (Triangular fuzzy number)

Definition 1.8

A fuzzy number $\tilde{A} = (a,b,c,d)$, shown in Figure 1.3, is said to be a trapezoidal fuzzy number, if its membership function is given by

$$\mu_{\tilde{A}}(x) = \begin{cases} \dfrac{(x-a)}{(b-a)}, & a \le x \le b, \\ 1, & b \le x \le c \\ \dfrac{(d-x)}{(d-c)}, & \mathrm{c} \le x \le d \\ 0, & \text{otherwise} \end{cases}$$

where $a,b,c,d \in R$

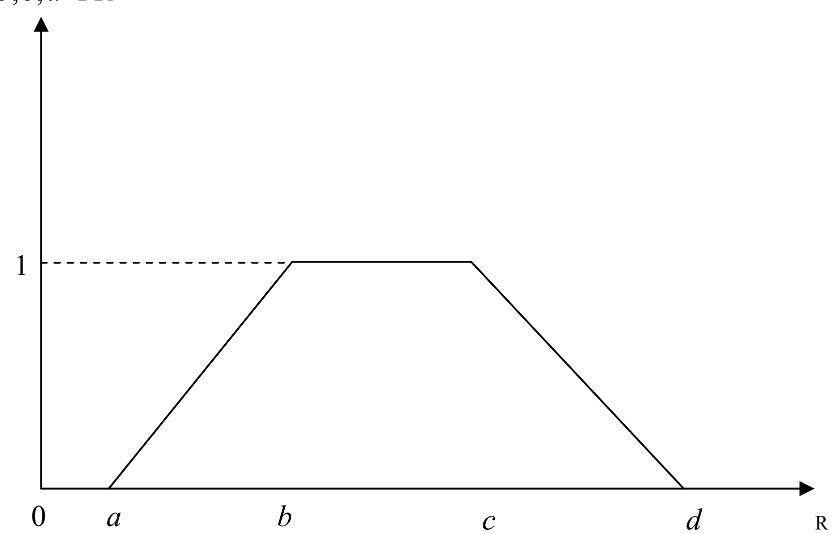


Figure 1.3 (Trapezoidal fuzzy number)

1.3 Arithmetic operations on triangular fuzzy numbers

Let $\widetilde{A} = (a_1, b_1, c_1)$ and $\widetilde{B} = (a_2, b_2, c_2)$ be two triangular fuzzy numbers then using α - cuts, $\widetilde{A}_\alpha$ and $\widetilde{B}_\alpha$ for $\alpha \in (0,1]$ one can compute $\widetilde{A} * \widetilde{B}$ where * may be any operation.

Addition ($\oplus$): $\widetilde{A} \oplus \widetilde{B} = (a_1 \oplus a_2, b_1 \oplus b_2, c_1 \oplus c_2)$

Subtraction (Θ): $\widetilde{A}\ \Theta\ \widetilde{B} = (a_1 \Theta\ c_2, b_1 \Theta\ b_2, c_1 \Theta\ a_2)$

Scalar Multiplication ($k\widetilde{A}$): $k\widetilde{A} = (ka_1, kb_1, kc_1)$ if $k > 0$ is a scalar and $k\widetilde{A} = (kc_1, kb_1, ka_1)$ if $k < 0$ is scalar.

Symmetry (or mirror image) ($-\widetilde{A}$): $-\widetilde{A} = (-c_1, -b_1, -a_1)$.

Multiplication $\widetilde{A} \otimes \widetilde{B} \approx (a', b', c')$ where $a' = \min(a_1.a_2, a_1.c_2, a_2.c_1, c_1.c_2)$, $b' = b_1.b_2$, $c' = \max(a_1.a_2, a_1.c_2, a_2.c_1, a_1.a_2, c_1.c_2)$.

Division $\widetilde{A} \div \widetilde{B} \approx (a', b', c')$ where $a' = \min\left(\frac{a_1}{a_2}, \frac{a_1}{c_2}, \frac{a_2}{c_1}, \frac{c_1}{c_2}\right)$, $b' = \frac{b_1}{b_2}$,

$$c' = \max\left(\frac{a_1}{a_2}, \frac{a_1}{c_2}, \frac{a_2}{c_1}, \frac{a_1}{a_2}, \frac{c_1}{c_2}\right), \quad a_2 > 0 \text{ or } c_2 < 0 .$$

1.4 Ranking function

A convenient method for comparing of fuzzy number is by use of ranking function [4,12,16,22]. A ranking function R: F(R) $\rightarrow$ R, where F(R) set of all fuzzy numbers defined on set of real number, maps each fuzzy number into a real number of F(R).

Let $\tilde{a}$ and $\tilde{b}$ be two triangular fuzzy numbers, then

$$(i)\ \widetilde{a} \underset{R}{\geq} \widetilde{b} \quad \text{if and only if } R(\widetilde{a}) \geq R(\widetilde{b})$$

$$(ii)\ \widetilde{a} \underset{R}{>} \widetilde{b} \quad \text{if and only if } R(\widetilde{a}) > R(\widetilde{b})$$

(iii) $\tilde{a} \underset{R}{=} \tilde{b}$ if and only if $R(\tilde{a}) = R(\tilde{b})$

1.4.1 Ranking function for triangular fuzzy number

For a triangular fuzzy number $\tilde{a} = (a, \alpha, \beta)$, ranking function [22] is given by

$R(\tilde{a}) = \dfrac{1}{2}\int_0^1 (\inf \tilde{a}_\alpha + \sup \tilde{a}_\alpha)\, d\alpha$, where $\tilde{a}_\alpha$ is $\alpha -$ cut on $\tilde{a}$. This reduces to

$R(\tilde{a}) = a + \dfrac{(\beta - \alpha)}{4}$.

Then for triangular fuzzy number $\tilde{a} = (a, \alpha, \beta)$ and $\tilde{b} = (b, \gamma, \theta)$, We have

$\tilde{a} \underset{R}{\geq} \tilde{b}$ if and only if $R(\tilde{a}) = [a + \dfrac{(\beta - \alpha)}{4}] \geq [b + \dfrac{\theta - \gamma}{4}] = R(\tilde{b})$.

Chapter 2

LITERATURE REVIEW

2.1 Fuzzy minimal spanning tree problem

One of the most fundamental problems in graph theory is the computation of a minimal spanning tree of a given undirected graph. This problem arises in a number of applications, for instance in network design. In the classical deterministic case it is assumed that the values of all edge costs are precisely known. The assumption that all the costs are precisely known may be a serious restriction in practice. In many real-world processes the exact values of the costs are not known in advance. An example was shown by Yamanet al. [31]. The minimal spanning tree can be then obtained by means of many efficient algorithms known in literature [1]. Consider a design of a telecommunication network where routing delays on links are not known with certainty. This uncertainty is caused by the time varying nature of the traffic load of the network.

It is then desirable to develop a network that hedges against all possible configurations of the costs, which may occur. One of the simplest methods of modeling the imprecise costs is to specify them as fuzzy numbers. It is then assumed that the value of each cost may fall within a given range, independently on the values of the remaining costs. A particular realization of the edge costs is called a scenario. The set of all scenarios is the Cartesian product of all the cost intervals. For this uncertainty representation two problems can be considered. The first one is the characterization of optimality of edges of a graph. Namely, we can characterize the event that a given edge will be a part of a minimal spanning tree under some scenario.

This characterization allows us to identify the most important edges, which should always be contained in a constructed solution. On the other hand, we can also identify the edges that will never be a part of a minimal spanning tree and, consequently, should be discarded while constructing a solution.

The second problem is choosing the best spanning tree under uncertainty. It is clear that an additional criterion is required to perform this task. Among criteria that can be applied to the interval uncertainty representation are the min max and min max regret. Under the min max criterion we seek a solution that minimizes the largest cost and under the min max regret one we seek a solution that minimizes the largest deviation from optimum over all scenarios.

Both criteria are described and discussed in a book by Kouvelis and Yu [19], which is entirely devoted to robust discrete optimization. The min max and min max regret criteria were first applied to the minimal spanning tree problem with interval costs by Yamanet al. [31]. The min max criterion leads to a problem that is polynomially solvable. Unfortunately, applying the second criterion, that is computing a spanning tree that minimizes the maximal regret turned out to be strongly NP-hard. This result was proved by Aron and van Hentenryck [2] and independently by Averbakh and Lebedev [3]. The problem can be solved by means of a mixed integer programming formulation proposed by Yamanet al. [31]. However, these exact methods allow us to solve the problem for rather small graphs. Finally, Kasperski and Zieli ´ nski [15] designed a simple and efficient 2-approximation algorithm for the problem.

2.2 Fuzzy shortest path problem

Let G = (V, E) be a graph, where V is the set of vertices and E is the set of edges. A path between two nodes is an alternating sequence of vertices and edges starting and ending with the vertices. The distance (cost) of a path is the sum of the weights of the edges on the path. However, since there can be more than one path between any two vertices, the problem of finding a path with the minimum cost between two specified vertices makes sense. This is the so-called shortest path problem (SPP).

The problem of finding the shortest path from a specified source node to the other nodes is a fundamental matter in graph theory and one that is currently being greatly studied [8,9,24-27]. It appears in many applications, including transportation, routing, communications, supply chain management or models involving agents.

In a network problem, the arcs are assumed to represent transportation time or cost. In the real world, the transportation time or cost may be known only approximately due to vagueness of information. To deal with this imprecise information, the probability concepts could be employed. However, to conduct probability distributions requires either a priori predictable regularity or a posteriori frequency determination. Moreover, the premise that imprecision can be equated with randomness is still questionable. As an alternative, uncertain values can be represented by membership functions under the fuzzy set theory [8]. The fuzzy shortest path problem was first discussed by Dubois and Prade [11].

Although the shortest path distance can be obtained, a corresponding shortest path cannot be identified [27]. Liu and Kao's [23] algorithm for finding the shortest path can obtain a non-dominated shortest path. For this, they need to transform all the fuzzy arcs into crisp arcs based on the Yager ranking method [30], and then solve the fuzzy shortest path problem with crisp arcs. However, 0-1 variables are needed in their approach. Okada and Soper [27] developed an algorithm based on the multiple labeling approach, by which a number of non-dominated paths can be generated. Besides, the multiple labeling approach is an exhaustive approach, and it needs to compare all the possible paths from the source node to the other nodes. Okada and Soper [27] also restricted each fuzzy number of arc length to L-R fuzzy number, thus the four objective functions are required in their approach .

2.3 Fuzzy maximal flow problem

Network flow problems have been very studied in graph theory, since they have many applications in the most diverse fields, such as: telecomunications, transportations, computation, manufacturing, etc. The maximum flow problem is an important problem in the network flow problems, it consists of sending the biggest amount of flow between two nodes (source s and destination t), holding the capacity restrictions of each arc. In the literature there are some efficient algorithms to solve the crisp problem [1]. But, there are problems that have uncertainties in their parameters (e.g.: costs, capacities and demands). This problem is called fuzzy maximum flow problem, where the graph has a crisp structure and fuzzy parameters.

In the literature, the number of papers dealing with fuzzy maximum flow problem is short [5,6,7,17]. The paper by Kim and Roush [17] is one of the first on this subject. The authors developed the fuzzy flow theory, presenting the conditions to obtain a optimal flow, by means of definitions on fuzzy matrices. But there were Chanas and Kolodziejczyk [5,6,7] who introduced the main works in the literature involving this subject. They approached this problem using the minimum cuts technique.

In the first paper [5], Chanas and Kolodziejczyk presented an algorithm for a graph with crisp structure and fuzzy capacities, i.e., the arcs have a membership function associated in their flow. This problem was studied by Chanas and Kolodzijczyk [6] again, in this paper the flow is a real number and the capacities have upper and lower bounds with a satisfaction function. Chanas and Kolodzijczyk [7] had also studied the integer flow and proposed an algorithm.

Chapter 3

FUZZY MINIMAL SPANNING TREE PROBLEM

3.1 Introduction

In this chapter the minimal spanning tree problem in a given connected graph is considered. It is assumed that the edge costs are not precisely known and they are specified as triangular fuzzy numbers. A new algorithm has been proposed to characterize the optimality of edges of the graph and to choose a spanning tree under fuzzy costs.

3.2 Fuzzy minimal spanning tree algorithm

This algorithm calculates the minimum approximate distance between nodes connected directly or indirectly. Such type of problem arises when fuzzy cable network is to be set up for certain areas in a town or city. Here, we are interested that each area is connected, and the approximate length of the cable is minimum. Another example is about the construction of paved roads between villages so that each village is connected by the paved road, and approximate distance is minimum, i.e. such linkage is economical.

Let $\{1, 2, \ldots, n\}$ be the set of nodes of the fuzzy network. Define

FC_K = Set of all the nodes that have permanently connected at iteration *K*.

$F\overline{C}_K$ = Set of all the nodes yet to be connected.

The steps of the algorithm proceed as follows:

Step 1

Start with any node $\{i\}$ from given n nodes and write $FC_1 = \{i\}$ which renders automatically $F\overline{C}_1 = n - \{i\}$. Set $K = 2$.

Step 2

Select a node *j*, in the unconnected set $F\overline{C}_{K-1}$ that gives the shortest approximate distance to a node in FC_{K-1}. This will be done by calculating rank of all the fuzzy numbers of the nodes connected to node $\{i\}$. Fuzzy number with least rank will give shortest approximate distance to the node in FC_{K-1}. Include node *j* permanently to FC_{K-1}, and remove it from $F\overline{C}_{K-1}$ to have

$$FC_K = FC_{K-1} + \{j\}, \qquad F\overline{C}_K = F\overline{C}_{K-1} - \{j\}$$

If the set of unconnected nodes FC_K is empty, then stop. Otherwise set $K = K + 1$, and repeat the Step 2 .

Example 3.1

A TV company is requested for providing cable services to six new housing development areas. Figure 3.1 depicts the potential TV linkage. The approximate length of cable in miles is represented by triangular fuzzy number and shown in Table 3.1. Determine the most economical network.

Table 3.1

Activity (i, j)	Fuzzy cable length between two houses i and j (in miles)
$(1,2)$	Approximately 1 miles $(0,1,2)$
$(1,3)$	Approximately 5 miles $(3,5,7)$
$(1,4)$	Approximately 7 miles $(5,7,9)$
$(1,5)$	Approximately 9 miles $(7,9,11)$
$(2,3)$	Approximately 6 miles $(4,6,8)$
$(2,4)$	Approximately 4 miles $(2,4,6)$
$(2,5)$	Approximately 3 miles $(1,3,5)$
$(3,4)$	Approximately 5 miles $(3,5,7)$
$(3,6)$	Approximately 10 miles $(8,10,12)$
$(4,5)$	Approximately 8 miles $(6,8,10)$
$(4,6)$	Approximately 3 miles $(1,3,5)$

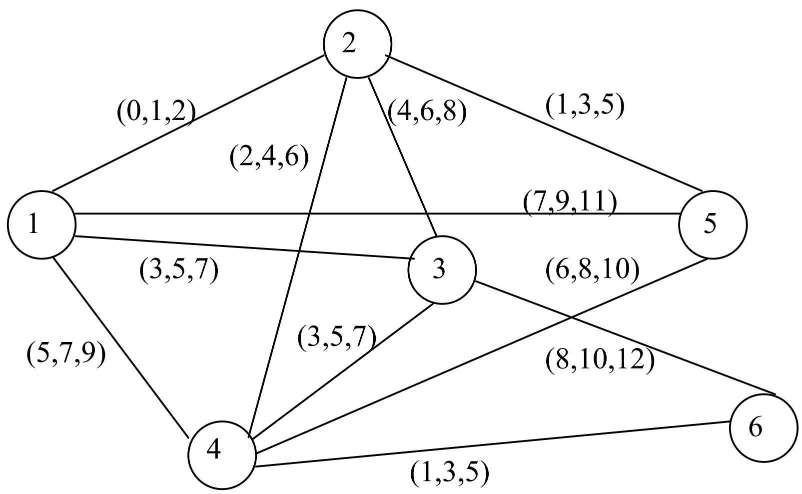


Figure 3.1

The algorithm starts at node 1 (any other node will do as well), which gives $FC_1 = \{1\}$, $F\overline{C_1} = \{2,3,4,5,6\}$. For second iteration, choose the node from $F\overline{C_1}$ which is nearest to the node $\{1\}$. According the step 2 we will first calculate the rank of all the fuzzy numbers of nodes connected to node $\{1\}$ that are $\{2,3,4\}$. Using rank $R(a, b, c) = \frac{1}{4}(a + 2 \times b + c)$.

Now $R(0,1,2) = \frac{1}{4}(0+2\times1+2) = 1$

$$R(7,9,11) = \frac{1}{4}(7+2\times9+11) = 9$$

$$R(3,5,7) = \frac{1}{4}(3+2\times5+7) = 5$$

$$R(5,7,9) = \frac{1}{4}(5+2\times7+9) = 7.$$

Thus $R(0,1,2) < R(3,5,7) < R(5,7,9) < R(7,9,11)$. Since rank of $(0,1,2)$ is minimum so node nearest to $\{1\}$ is $\{2\}$. Hence connect node 1 to 2 by dashed line, see in Figure 3.2 and construct the sets

$$FC_2 = \{1,2\}, \quad F\overline{C}_2 = \{3,4,5,6\}$$

For third iteration, choose the node from $F\overline{C}_2$ which gives the shortest approximate distance to 1 or 2. Again calculate the rank of the fuzzy numbers of the nodes nearest to 1 or 2.

$$R(5,7,9) = \frac{1}{4}(5+2\times7+9) = 7$$

$$R(4,6,8) = \frac{1}{4}(4+2\times6+8) = 6$$

$$R(1,3,5) = \frac{1}{4}(1+2\times3+5) = 3$$

$$R(3,5,7) = \frac{1}{4}(3+2\times5+7) = 5$$

$$R(2,4,6) = \frac{1}{4}(2+2\times4+6) = 4$$

$$R(7,9,11) = \frac{1}{4}(7+2\times9+11) = 9.$$

Since $R(1,3,5) < R(2,4,6) < R(3,5,7) < R(4,6,8) < R(5,7,9) < R(7,9,11)$ so shortest approximate distance will be from node 2 to 5. Hence connect node 2 to 5 by dashed line and construct sets

$$FC_3 = \{1, 2, 5\}, \qquad \overline{FC_3} = \{3, 4, 6\}.$$

For fourth iteration, choose the node from $\overline{FC_3}$ which gives the shortest approximate distance to 1 or 2 or 5. Again calculate the rank of the fuzzy numbers of the nodes nearest to 1 or 2 or 5.

$$R(5,7,9) = \frac{1}{4}(5+2\times 7+9) = 7$$

$$R(3,5,7) = \frac{1}{4}(3+2\times 5+7) = 5$$

$$R(4,6,8) = \frac{1}{4}(4+2\times 6+8) = 6$$

$$R(6,8,10) = \frac{1}{4}(6+2\times 8+10) = 8$$

$$R(2,4,6) = \frac{1}{4}(2+2\times 4+6) = 4$$

$$R(7,9,11) = \frac{1}{4}(7+2\times 9+11) = 9.$$

Since $R\ (2,4,6) < R(3,5,7) < R\ (4,6,8) < R\ (5,7,9) < R\ (6,8,10) < R\ (7,9,11)$. Thus shortest approximate distance will be from node 2 to 4. Hence connect node 2 to 4 by dashed line and construct sets

$$FC_4 = \{1,2,5,4\}, \qquad \overline{FC_4} = \{3, 6\}.$$

For fifth iteration, choose the node from $\overline{FC_4}$ which gives the shortest approximate distance to 1 or 2 or 3 or 4. Again calculate the rank of the fuzzy numbers of the nodes nearest to 1 or 2 or 3 or 4.

$$R(5,7,9) = \frac{1}{4}(5+2\times 7+9) = 7$$

$$R(3,5,7) = \frac{1}{4}(3+2\times 5+7) = 5$$

$$R(4,6,8) = \frac{1}{4}(4+2\times 6+8) = 6$$

$$R(6,8,10) = \frac{1}{4}(6+2\times 8+10) = 8$$

$$R(7,9,11) = \frac{1}{4}(7+2\times 9+11) = 9$$

$$R(1,3,5) = \frac{1}{4}(1+2\times 3+5) = 3\,.$$

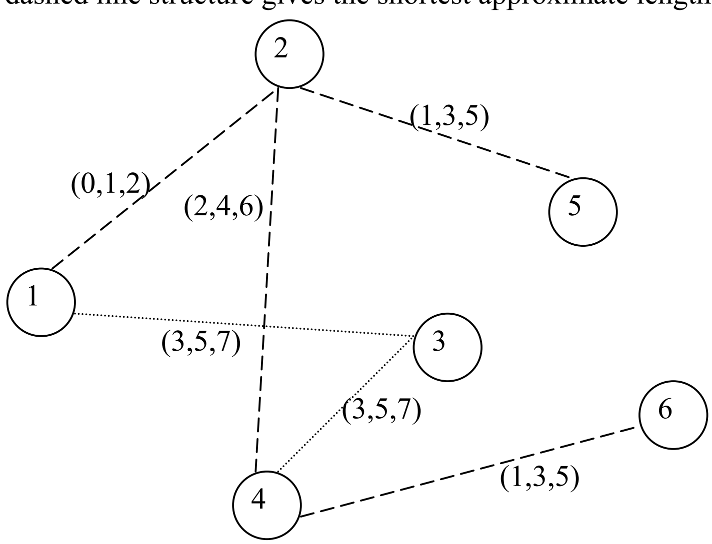
Since $R(1,3,5) < R(3,5,7) < R(4,6,8) < R(5,7,9) < R(6,8,10) < R(7,9,11)$. Thus shortest approximate distance will be from nodes 4 to 6 and 2 to 5. Hence connect nodes 4 to 6 and 2 to 5 by dashed line and construct sets

$$FC_5 = \{1, 2, 5, 4, 6\}, \qquad F\overline{C}_5 = \{3\}.$$

In last iteration the two branches are there which gives the same shortest approximate distance, i.e. there ranks are equal so connect node 3 to 4 or 3 to 1 by dashed line. Thus

$$FC_6 = \{1, 2, 5, 4, 6, 3\}, \qquad F\overline{C}_6 = \{\phi\}$$

Now the dashed line structure gives the shortest approximate length of the cable.



Total approximate shortest length =

$(0,1,2)$ + $(1,3,5)$ + $(1,3,5)$ + $(2,4,6)$ + $(3,5,7)$ = $(7,16,25)$ i.e. approximately 16 miles.

3.3 Fuzzy matrix method

The above algorithm can easily carried out distance on the fuzzy distance matrix. Execute the following steps.

Step 1

Represent the fuzzy network by fuzzy distance matrix as shown in Table 3.2. If two nodes are not connected directly, then take (∞,∞,∞) in the cell concerned.

Step 2

Let Q be initially a null set whose elements at later stage will taken as the row numbers.

Step 3

Select row one and include this in Q. Then delete 1^{st} Column of the fuzzy distance matrix. We may start with any row.

Step 4

Find the entry with smallest rank in all the rows contained in Q and starred it. In case of tie break it arbitrarily.

Step 5

Identify the column of starred entry of Step 4, and let it be *K*. Include K^{th} Row in Q.

Step 6

Delete K^{th} Column identified in Step 5.

Step 7

Check whether all the columns of the fuzzy distance matrix are deleted. If so go to Step 8, otherwise go to Step 4.

Step 8

Show the arcs in the fuzzy spanning tree corresponding to the starred entries and total of the starred entries gives the minimum approximate distance required for spanning tree. Let us rework out the above example by matrix method.

Write the fuzzy distance matrix from the network as shown in Table 3.2, initially Q $= \phi$.

Table 3.2

	1	2	3	4	5	6
1	-	(0,1,2) Rank = 1	(3,5,7) Rank = 5	(5,7,9) Rank = 7	(7,9,11) Rank = 9	(∞,∞,∞) Rank = ∞
2	(0,1,2) Rank = 1	-	(4,6,8) Rank = 6	(2,4,6) Rank = 4	(1,3,5) Rank = 3	(∞,∞,∞) Rank = ∞
3	(3,5,7) Rank = 5	(4,6,8) Rank = 6	-	(3,5,7) Rank = 5	(∞,∞,∞) Rank = ∞	(8,10,12) Rank = 10
4	(5,7,9) Rank = 7	(2,4,6) Rank = 4	(3,5,7) Rank = 5	-	(6,8,10) Rank = 8	(1,3,5) Rank = 3
5	(7,9,11) Rank = 9	(1,3,5) Rank = 3	(∞,∞,∞) Rank = ∞	(6,8,10) Rank = 8	-	(∞,∞,∞) Rank = ∞
6	(∞,∞,∞) Rank = ∞	(∞,∞,∞) Rank = ∞	(8,10,12) Rank = 10	(1,3,5) Rank = 3	(∞,∞,∞) Rank = ∞	-

Iteration 1

Set $Q = \{1\}$ (any other node will work well) and write the fuzzy distance matrix after deleting the 1^{st} Column as shown in Table 3.3. Star 1^{st} Row. Each entry of Q will be shown starred. Find the rank of all the fuzzy entries. Find the smallest fuzzy entry. The smallest fuzzy entry in 1^{st} Row is $(0,1,2)$. Starred it.

Table 3.3

	2	3	4	5	6
1^*	$(0,1,2)^*$ Rank = 1	(3,5,7) Rank = 5	(5,7,9) Rank = 7	(7,9,11) Rank = 9	(∞,∞,∞) Rank = ∞
2	-	(4,6,8) Rank = 6	(2,4,6) Rank = 4	(1,3,5) Rank = 3	(∞,∞,∞) Rank = ∞
3	(4,6,8) Rank = 6	-	(3,5,7) Rank = 5	(∞,∞,∞) Rank = ∞	(8,10,12) Rank = 10
4	(2,4,6) Rank = 4	(3,5,7) Rank = 5	-	(6,8,10) Rank = 8	(1,3,5) Rank = 3
5	(1,3,5) Rank = 3	(∞,∞,∞) Rank = ∞	(∞,∞,∞) Rank = ∞	-	(∞,∞,∞) Rank = ∞
6	(∞,∞,∞) Rank = ∞	(8,10,12) Rank = 10	(1,3,5) Rank = 3	(∞,∞,∞) Rank = ∞	-

Iteration 2

The starred entry of 1^{st} iteration occurs in 1^{st} Row and 2^{nd} Column. For 2^{nd} iteration, delete the second column of Table 3.3 and star second row, see Table 3.4. The smallest entry of 1^{st} and 2^{nd} Row is $(1,3,5)$. Starred it. $Q=\{1,2\}$.

Table 3.4

	3	4	5	6
1^*	(3,5,7) Rank = 5	(5,7,9) Rank = 7	(7,9,11) Rank = 9	(∞,∞,∞) Rank = ∞
2^*	(4,6,8) Rank = 6	(2,4,6) Rank = 4	$(1,3,5)^*$ Rank = 5	(∞,∞,∞) Rank = ∞
3	-	(3,5,7) Rank = 5	(∞,∞,∞) Rank = ∞	(8,10,12) Rank = 10
4	(3,5,7) Rank = 5	-	(6,8,10) Rank = 8	(1,3,5) Rank = 3
5	(∞,∞,∞) Rank = ∞	(6,8,10) Rank = 8	-	(∞,∞,∞) Rank = ∞
6	(8,10,12) Rank = 10	(1,3,5) Rank = 3	(∞,∞,∞) Rank = ∞	-

Iteration 3

The starred entry of 2nd iteration lies in 2nd Row and 5th Column. For executing 3rd iteration delete 5th column of Table 3.4 and star fifth row, see Table 3.5. The smallest entry in all starred rows is $(2,4,6)$. Starred it, $Q=\{1,2,5\}$.

Iteration 4

The starred entry of 3rd iteration lies in 2nd Row and 4th Column. To perform 4th iteration, delete 4th Column of Table 3.5 and star 4th Row, see Table 3.6. The smallest entry in starred rows (done so far) is $(1,3,5)$. Starred it, $Q = \{1,2,5,4\}$.

Table 3.5

	3	4	6
1^*	(3,5,7) Rank = 5	(5,7,9) Rank = 7	(∞,∞,∞) Rank = ∞
2^*	(4,6,8) Rank = 6	$(2,4,6)^*$ Rank = 4	(∞,∞,∞) Rank = ∞
3	-	(3,5,7) Rank = 5	(8,10,12) Rank = 10
4	(3,5,7) Rank = 5	-	(1,3,5) Rank = 3
5^*	(∞,∞,∞) Rank = ∞	(6,8,10) Rank = 8	(∞,∞,∞) Rank = ∞
6	(8,10,12) Rank = 10	(1,3,5) Rank = 3	-

Iteration 5

The starred entry of previous iteration is in 6th Column and hence delete the sixth column, star sixth row to perform the next iteration, see Table 3.7. The smallest fuzzy entry in starred rows is available at two positions, i.e. 1st Row and 4th Row. Thus there is a tie. Break the tie arbitrarily by taking 4th Row and starred the smallest fuzzy entry which is $(3,5,7)$, see Table 3.7. $Q = \{ 1,2,5,4,6 \}$.

Table 3.6

	3	6
1^*	(3,5,7) Rank = 5	(∞,∞,∞) Rank = ∞
2^*	(4,6,8) Rank = 6	(∞,∞,∞) Rank = ∞
3	-	(8,10,12) Rank = 10
4^*	(3,5,7) Rank = 5	$(1,3,5)^*$ Rank = 3
5^*	(∞,∞,∞) Rank = ∞	(∞,∞,∞) Rank = ∞
6	(8,10,12) Rank = 10	-

Table 3.7

	3
1^*	(3,5,7) Rank = 5
2^*	(4,6,8) Rank = 6
3	-
	$(3,5,7)^*$ Rank = 5
5^*	(∞,∞,∞) Rank = ∞
6^*	(8,10,12) Rank = 10

Iteration 6

After 5^{th} iteration, all columns have been deleted and algorithm stops, and $Q = \{1,2,5,4,3\}$. To write the minimum approximate distance of spanning tree add all starred entries, and connect nodes corresponding to them. This will give fuzzy minimal spanning tree, see Figure 3.2. The sum of starred fuzzy entries is

$(0,1,2) + (1,3,5) + (1,3,5) + (2,4,6) + (3,5,7) = (7,16,25)$ i.e. approximately 16 miles.

Remark:

The tie in 5^{th} iteration indicates that there exists an alternate fuzzy minimal spanning tree.

3.4 Result and discussion

The obtained result can be explained as follows:

1) Required length of cable is greater than 16 and less than 25.
2) Maximum number of persons are in favour that length will be 16.
3) The percentage of person increases when length of cable between two houses varies from 7 to 16 and decreases from 16 to 25.
4) The membership function for the obtained result is shown in Figure 3.3 where x-axis denotes required minimum length of cable between two houses and y-axis denotes its membership value.

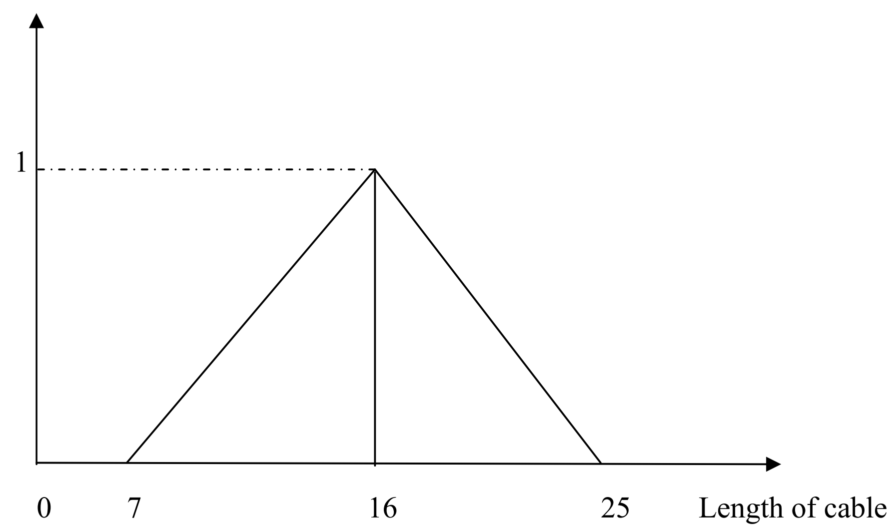


Figure 3.3 (Triangular fuzzy number)

3.5 Conclusion

In this chapter, two new algorithms have been proposed to solve the fuzzy minimal spanning tree problems occurring in real life. To illustrate the algorithms a numerical example has been solved and results are explained. If there is no uncertainty about the length then the proposed algorithm gives the same result as in crisp minimal spanning tree problem.

Chapter 4

FUZZY SHORTEST PATH PROBLEM

4.1 Introduction

In this chapter, shortest path problem in a given connected is considered. It is assumed that the approximate costs are known and they are specified as triangular fuzzy numbers. A new an algorithm has been introduced to solve the fuzzy shortest path problems.

4.2 Fuzzy dijkstra's algorithm

A network is given with different nodes connected directly or indirectly. This algorithm finds approximate shortest distance between a source (given) and any other node in the network. The algorithm advances from a node i to an immediately successive node j using a fuzzy labeling procedure. Let $\tilde{u}_i$ be approximate shortest distance from node 1 to node i and $(R(\tilde{d}_{ij})\geq 0)$ be the length of $(i, j)^{th}$ arc. Then, the fuzzy label for node j is defined as

$$[\tilde{u}_j, i] = [\tilde{u}_i \oplus \tilde{d}_{ij}, i] \quad , \qquad R(\tilde{d}_{ij}) \geq 0 .$$

Here label $[\tilde{u}_j, i]$ mean we are coming from nodes i after covering an approximate distance $\tilde{u}_j$ from the starting node. There are two types of fuzzy labels in this algorithm: *Temporary and Permanent.* A temporary fuzzy label can be replaced with another temporary fuzzy label, if shorter path to the same fuzzy node is detected. At the stage when it is certain that no better route can be found. The status of temporary node is changed to permanent.

The steps of the algorithm are summarized as follows:

Step 1

Label the source node (say node 1) with the permanently level $[(0,0,0), -]$. Set $i = 1$.

Step 2

Compute the temporary labels $[\tilde{u}_i \oplus \tilde{d}_{ij}, i]$ for each node j that can be reached from i, provided j is not permanently labeled. If node j is already labeled as $[\tilde{u}_j, k]$ through another node *k*, and if $R(\tilde{u}_i \oplus \tilde{d}_{ij}) < R(\tilde{u}_j)$ replace $[\tilde{u}_j, k]$ with $[\tilde{u}_i \oplus \tilde{d}_{ij}, i]$.

Step 3

If all the fuzzy nodes have been permanently labeled, stop. Otherwise, select the label $[\tilde{u}_r, s]$ with shortest distance $(\tilde{u}_r)$ from among all the temporary labels (break the tie arbitrarily). Set $i = r$ and repeat Step 1.

Remark:

At each iteration among all temporary nodes, make those nodes permanent which have smallest approximately distance. Note that at any iteration we can not move to permanent node, however, reverse is possible. After all the nodes have been labeled and only one temporary node remains, make it permanent.

***Example* 4.1**

In the network shown in Figure 4.1, five towns are connected through permissible routes. This approximate distance in miles between any towns are represented by triangular fuzzy numbers and is given on the arc connecting these towns. Find the approximate shortest distance of all towns from 1st town.

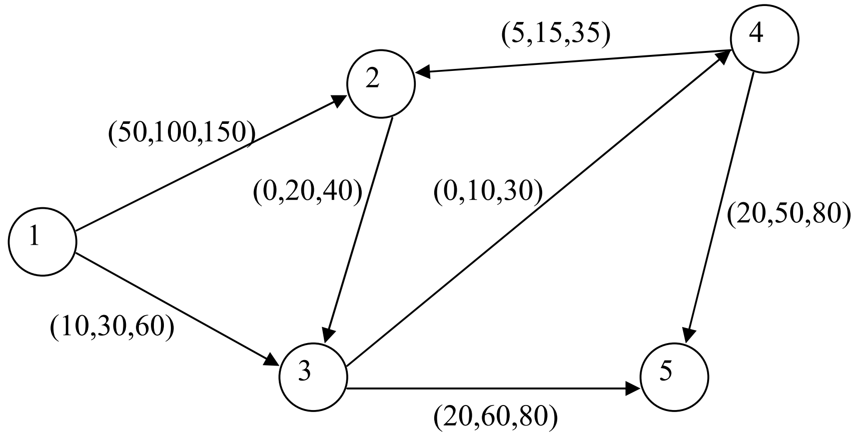


Figure 4.1

Iteration 1

Assign the permanent label $[(0,0,0),-]$ to node 1**.**

Node	Label	Status
1	$[(0,0,0),-]$	P
2	$[(50, 100, 150),1]$	T
3	$[(10,30,60),1]$	T

Table 4.1

$$R(10,\ 30,\ 60)\ =\ \frac{(10+2\times 30+60)}{4} = 32.5$$

$$R(50,100,150) = \frac{(50+2\times 100+150)}{4} = 100$$

$R(10,30,60) < R(50,100,150)$.

Iteration 2

Nodes 2 and 3 can be reached from (the last permanently labeled) node 1. Thus, label $[(50,100,150),1]_{(1)}$ and $[(10,30,60),1]_{(1)}$ to nodes 2 and 3, respectively. Since the rank of $(10,30,60)$ is less than $(50,100,150)$ so $[(10,30,60),1]$ is less than $[(50,100,150),1]$, make label $[(10,30,60),1]_{(1)}$ as permanent node, see Figure 4.2. The superscript (1) denote the number of iterations.

Nodes	Label	Status
1	$[(0,0,0),-]$	P
2	[(50, 100, 150),1]	T
3	$[(10,30,60),1]$	P
4	$[(10,40,90),3]$	T
5	$[(30,90,140),3]$	T

Table 4.2

Iteration 3

Nodes 4 and 5 can be reached from node 3, and hence label $[(10,40,90),3]_{(2)}$ and $[(30,90,140),3]_{(2)}$ to nodes 4 and 5, respectively. Since the rank of $(10,40,90)$ is less than rank of $(30,90,140)$. So $[(10,40,90),3]_{(2)} < [(30,90,140),3]_{(2)}$ make label $[(10,40,90),3]_{(2)}$ permanent.

Nodes	Label	Status
1	$[(0,0,0),-]$	P
2	$[(50,100,150),1]$	T
3	$[(10,30,60),1]$	P
4	$[(10,40,90),3]$	P
5	$[(30,90,140),3]$	T

Table 4.3

Iteration 4

So far, nodes 2 and 5 are temporary. Node 2 can also be reached from permanent node 4 and its label is changed to $[(15,55,125),4]_{(3)}$. Since no other permanent node exists from where we can reach at node 2 and since the rank of $(15,55,125)$ is less than $(50,100,150)$ so $[(15,55,125),4]$ is less than $[(50,100,150),2]$ make label $[(15,55,125)]_{(3)}$ as permanent. Note that node 5 can also reached from 4 with label $[(30,90,70),4]_{(3)}$.

Nodes	Label	Status
1	$[(0,0,0),-]$	P
2	$[(15, 55, 125), 4]$	P
3	$[(10,30,60),1]$	P
4	$[(10,40,90),3]$	P
5	$[(30,90,140),3]$	T

Table 4.4

Iteration 5

Only node 3 can be reached from node 2, but node 3 is already labeled permanently, it can not be relabeled. The only temporary node is 5 and this does not lead to any other node, its status is changed to permanent.

Nodes	Label	Status
1	$[(0,0,0),-]$	P
2	$[(15, 55, 125), 4]$	P
3	$[(10,30,60),1]$	P
4	$[(10,40,90),3]$	P
5	$[(30,90,140),3]$	P

Table 4.5

The fuzzy shortest route between the source node 1 and other node in fuzzy network is determined by starting at desire destination and backtracking through the nodes using the information given by permanently labels, see Figure 4.2. For example at fuzzy node 2 we reach as

$$2 \leftarrow 4 \leftarrow 3 \leftarrow 1$$

The desired route is $1 \rightarrow 3 \rightarrow 4 \rightarrow 2$ and the total approximate distance between 1st and 2nd node is 55 units i.e. $(15,55,125)$ units.

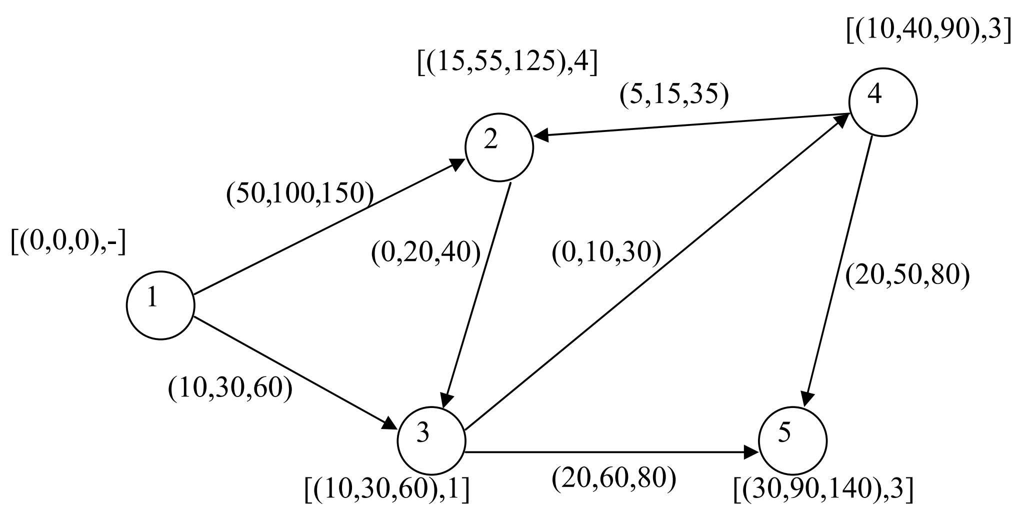


Figure 4.2

4.3 Result and discussion

The obtained result can be explained as follows:

Table 4.6

Node No. (i)	Approximate distacnce between i^{th} node and 1^{st} node	Path of i^{th} node from 1^{st} node
2	(15,55,125)	$1 \rightarrow 3 \rightarrow 4 \rightarrow 2$
3	(10,30,60)	$1 \rightarrow 3$
4	(10,40,90)	$1 \rightarrow 3 \rightarrow 4$
5	(30,90,140)	$1 \rightarrow 3 \rightarrow 5$

1. The approximate distance between the source node 1 and node 2 in fuzzy network is greater than 55 and less than 125.
2. Maximum number of persons are in favour that distance will be 55.
3. The percentage of person increases when fuzzy shortest route varies from 15 to 55 and decreases from 55 to 125.
4. The membership function for the obtained result is shown in Figure 4.3 where x-axis denotes distance between source node and other node and y-axis denotes its membership value. Similarly for the rest of nodes result can be explained.

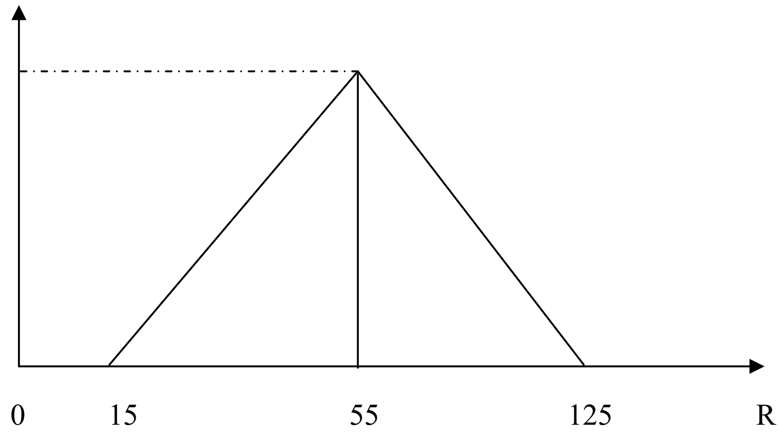


Figure 4.3 (Triangular Fuzzy Number)

4.4 Conclusion

In this chapter, a new algorithm has been proposed to solve the fuzzy shortest path problem occurring in real life. To illustrate the algorithm a numerical example has been solved and results are explained. If there is no uncertainty about the distance between different nodes then the proposed algorithm gives the same result as in crisp shortest path problem.

Chapter 5

FUZZY MAXIMAL FLOW PROBLEM

5.1 Introduction

The problem of finding the maximum flow between a source and a destination node in a network with uncertainties in its capacities is an important problem of network flows, since it has a wide range of applications in different areas (telecommunications, transportations, manufacturing, etc) and therefore deserves special attention. However, due to complexity in working with this kind of problems, there are a few algorithms in literature, which demand that the user informs the desirable maximum flow, which is difficult when the network is the large scale. In this chapter, an algorithm based on the classical algorithm is proposed.

5.2 Fuzzy maximal flow algorithm

The algorithm presented here is the labeling technique. The idea of fuzzy maximal flow algorithm is to find a breakthrough path with positive net flow that connects the source and sink nodes. Take an arc (i, j) with initial approximate capacities ($\widetilde{fc}_{ij}, \widetilde{fd}_{ji}$). As the computations of the algorithm proceed, portions of these approximate capacities will be committed to the flow in the arc. The excess approximate capacities on the arcs are then changed accordingly. For excess approximate capacities on the arc (i, j), we use the notation ($\widetilde{fc}_{ij}, \widetilde{fd}_{ji}$). The fuzzy network with the updated excess approximate capacities will be referred to as the residual fuzzy network. Define

$[\widetilde{fa}_j, i]$ = Approximate flow $\widetilde{fa}_j$ from node j to node i.

The source node is numbered 1 and the algorithm proceeds as follows:

Step 1

Let the index j refer to all nodes that can be reached directly from source node 1 by arc with positive excess capacities, i.e. $R(\widetilde{fc}_{1j}) > 0$ for all j. On the diagram of the network, we label nodes j with two numbers $[\widetilde{fa}_j, 1]$, where $\widetilde{fa}_j$ is the approximate positive excess capacity, and 1 means we are coming from node 1. If in doing this we label the sink N, so that there is a branch of approximate positive excess capacity from source to the sink, then the approximate maximal flow along the path is given by $\widetilde{f}_1 = \widetilde{fc}_{1N}$, and the excess capacity due to this breakthrough path is determined by $\widetilde{f}_1$ in the direction of the flow and is increased by $\widetilde{f}_1$ in the reverse direction. This means that for source nodes 1 and sink node N the excess flow is changed from the current

$$(\widetilde{fc}_{1N}, \widetilde{fd}_{N1}) \text{ to } (\widetilde{fc}_{1N} \ominus \widetilde{f}_1, \widetilde{fd}_{N1} \oplus \widetilde{f}_1)$$

Step 2

In case in step 1, the sink is not labeled, choose the smallest index j of the labeled nodes and search the unlabeled nodes which can be reached from j by arcs of approximate positive excess capacities. If there are no such nodes we move to the next lowest index j and repeat the process. If sink is labeled, we immediately compute

$\widetilde{f}_1$ = minimum of the excess approximate capacities on the path to the sink.

Subtract $\widetilde{f}_1$ from excess approximate capacities on the arc in the direction of path and add $\widetilde{f}_1$ from the excess approximate capacity in reverse direction. In this way we get fresh excess approximate capacities. Even, if the sink is not labeled, some unlabeled nodes (other than sink) can be reached, than using the general index *k*, we label each node as follows $[\widetilde{f}a_K, j]$, and compute $\widetilde{f}_1$.

Step 3

Steps 1 or 2 give first breakthrough. Compute freshly excess approximate capacities of all arcs which are changed due to first breakthrough. The process is repeated from step 1 to 3 until, in a finite number of steps, we reach the state so that no additional nodes can be labeled to reach sink. This is no breakthrough. The approximate maximum flow is computed by

$$\widetilde{f} = \widetilde{f}_1 \oplus \widetilde{f}_2 \oplus \widetilde{f}_3 \oplus \ldots \oplus \widetilde{f}_p$$

where p is the number of iteration to get no breakthrough.

The approximate optimal flow in the arc (i, j) is computed as

$$(\widetilde{\alpha}, \widetilde{\beta}) = (\widetilde{f}c_{ij} \ \Theta \ \widetilde{f}d'_{ij}, \widetilde{f}c_{ji} \ \Theta \ \widetilde{f}d'_{ji})$$

where $\widetilde{f}c_{ij}$ and $\widetilde{f}c_{ji}$ are the initial approximate capacities, and $\widetilde{f}d'_{ij}$, $\widetilde{f}d'_{ji}$ are the final approximate excess capacities. If $R(\widetilde{\alpha}) > 0$, the approximate optimal flow from i to j is $\widetilde{\alpha}$. Otherwise, if $R(\widetilde{\beta}) > 0$, the approximate optimal flow from j to i is $\widetilde{\beta}$. Note that $R(\widetilde{\alpha})$ and $R(\widetilde{\beta})$ can not be positive together.

Remark

During labeling process, we are not worried about whether the orientation of the arc is in the direction of our move from j to *k*. We need that the arc must have positive residual. The algorithm is illustrated by working out an example of fuzzy maximal flow problem.

Example 5.1

Consider the network shown in Figure 5.1. The bidirectional approximate capacities are shown on the respective arcs. For example, for arc $(3,4)$ the flow limit is approximately 10 say $(5,10,15)$ units from 3 to 4 and approximately 5 units from 4 to 3 say $(0,5,10)$ units. Determine the approximate maximal flow in this network between source 1 and sink 5 .

The algorithm is applied in the following manner.

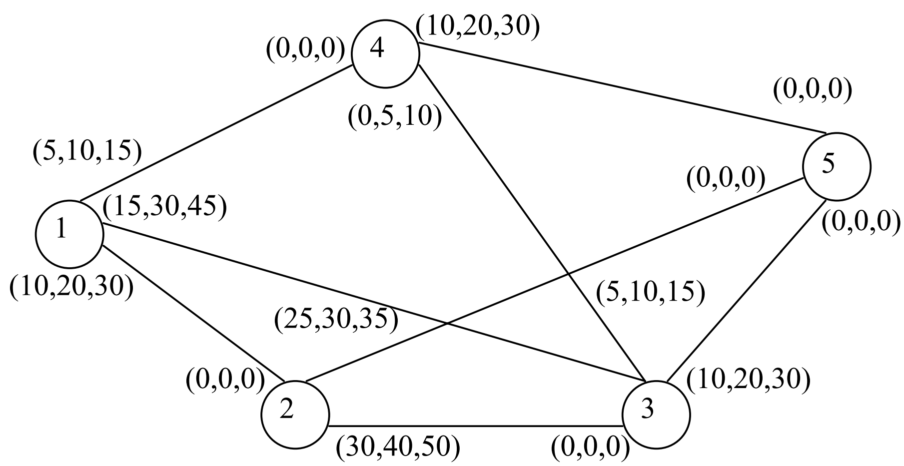


Figure 5.1

Iteration 1

At the first step, find the nodes that can be reached directly from the source by arc of positive excess approximate capacity $R(\widetilde{fc}_{ij}) > 0$. These nodes are 2, 3, 4. Label these nodes with the ordered pair of numbers $[\widetilde{fa}_j, 1]$, where $\widetilde{fa}_j = \widetilde{fc}_{1j}$ and 1 means we have reached from node 1. Firstly we will choose the path having approximate maximal flow limit. Rank of approximate flow limits from source node 1 are

$$R(5,10,15)=\frac{5+2\times10+15}{4}=10$$

$$R(15,30,45)=\frac{15+2\times30+45}{4}=30$$

$R(10,20,30)=\dfrac{10+2\times20+30}{4}=20$. Since the rank of $(15,30,45)$ is maximum so we will choose the path form node 1 to 3. Node 3 is labeled as $[\,(15,30,45),1]$. Still sink is not labeled. Again we will choose the path having maximal approximate flow limit i.e. $[(10,20,30),3]$ from node 3 to node 5. Now, sink is reached and labeling process stops as we have got first breakthrough. The approximate flow in the network can be increased by

$$\widetilde{f}_1=\min\{[(\infty,\infty,\infty),-],[(15,30,45),1],[(10,20,30),3]\}$$

That approximate flow limit will be minimum, whose rank is minimum.

$$R(15,30,45)=\frac{15+2\times30+45}{4}=30$$

$R(10,20,30)=\dfrac{10+2\times20+30}{4}=20$. Since the rank of $(10,20,30)$ is minimum so $\widetilde{f}_1$ is $(10,20,30)$. The value of $\widetilde{f}_1$ indicates that increase of approximately 20 i.e. $(10,20,30)$ units can be made along the path traced out in a move from source to sink. We can easily work backward to find the path. The label on the sink shows that we came from node 3. From node 3 it is seen that we came from node 1. The path is $1\rightarrow3\rightarrow5$. Before starting the second iteration, we first set the network showing the effect of first breakthrough. To have fresh excess approximate capacities subtract $\widetilde{f}_1$ = $(10,20,30)$ from the direction of flow in the path and add this amount in the opposite direction. Compute the approximate residual capacities of each arc on the path. After first iteration arc $(1,3)$ has fuzzy residual in direction of flow

$(15,30,45)\ominus(10,20,30)=(-15,10,35)$ and $(0,0,0)\oplus(10,20,30)=(10,20,30)$ in opposite direction. Similarly, arc $(3,5)$ has $(-20,0,20)$ and $(10,20,30)$ in the direction of flow and in opposite direction.

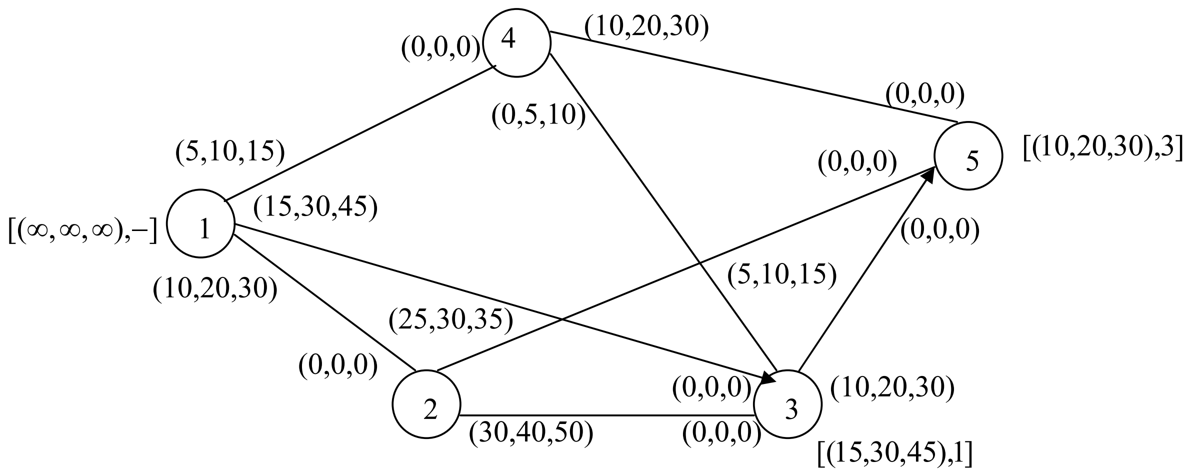


Figure 5.2

Iteration 2

Again start from source node 1 and find the nodes that can be reached directly from the source by arc of excess approximate capacities. Again we will choose the path having maximal approximate flow limit i.e. limit with maximum rank. Since the rank of $(10,20,30)$ is maximum so we will choose the path form node 1 to 2. Node 2 is labeled as $[(10,20,30),1]$. Again we will choose the path having approximate maximal flow limit i.e. $[(30,40,50),2]$ from node 2 to node 3. Still sink is not labeled. We will choose the path from node 3 to node 4 and then from node 4 to node 5. Node 3, 4 and 5 are labeled as $[(30,40,50),2]$, $[(5,10,15),3]$ and $[(10,20,30),4]$ respectively. Now, sink is reached and labeling process stops as we have got second breakthrough. The approximate flow in the network can be increased by

$\widetilde{f}_2 = \min\{[(\infty,\infty,\infty),-], [(10,20,30),1], [(30,40,50),2], [(5,10,15),3], [(10,20,30),4]\}$

That approximate flow limit will be minimum, whose rank is minimum. Since the rank of $(5,10,15)$ is minimum so $\widetilde{f}_2$ is $(5,10,15)$. The value of $\widetilde{f}_2$ indicates that increase of approximately 10 i.e. $(5,10,15)$ units can be made along the path traced out in a move from source to sink. We can easily work backward to find the path. The label on the sink shows that we came from node 4. From node 3 it is seen that we came from node 2. The path is $1 \to 2 \to 3 \to 4 \to 5$. Before starting the third iteration, we first set the network showing the effect of second breakthrough. To have fresh approximate excess capacities subtract $\widetilde{f}_2 = (5,10,15)$ from the direction of flow in the path and add this amount in the opposite direction. Compute the approximate residual capacities of each arc on the path. After second iteration arc $(1,2)$ has approximate residual in direction of flow $(10,20,30) \ominus (5,10,15) = (-5,10,25)$. And has in opposite direction $(0,0,0) \oplus (5,10,15) = (5,10,15)$. Similarly, arc $(2,3)$ has $(15,30,45)$ and $(5,10,15)$ in the direction of flow and in opposite direction and arc $(3,4)$ has $(-10,0,10)$ and $(5,15,25)$ in the direction of flow and in opposite direction and arc $(4,5)$ has $(-5,10,25)$ and $(5,10,15)$ in the direction of flow and in opposite direction.

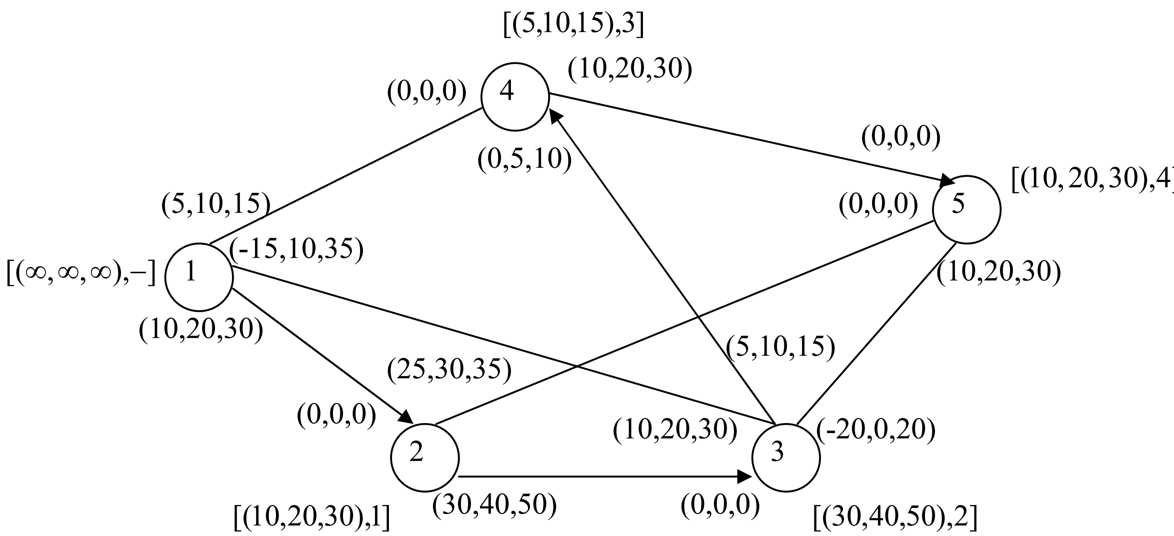


Figure 5.3

Iteration 3

Again start from source node 1 and find the nodes that can be reached directly from the source by arc of approximate excess capacities. Again we will choose the path having maximal approximate flow limit i.e. limit with maximum rank. Since the rank of all the approximate flow limits going from source node 1 are same so we can arbitrarily choose the path say from node 1 to node 2. Node 2 is labeled as $[(-5,10,25),1]$. Again rank of $(25,30,35)$ and $(15,30,45)$ are same if we choose the path from node 2 to node 3 then we can not go to node 4 and node 5 and, node 1 and 2 are already labeled so we will move from node 2 to node 5. Node 5 is labeled as $[(25,30,35),2]$. Now, sink is reached and labeling process stops as we have got third breakthrough. The approximate flow in the network can be increased by

$$\tilde{f}_3 = \min\{[(\infty,\infty,\infty),-],[(-5,10,25),1],[(25,30,35),3]\}$$

That approximate flow limit will be minimum, whose rank is minimum. Since the rank of $(-5,10,25)$ is minimum so $\tilde{f}_3$ is $(-5,10,25)$. The value of $\tilde{f}_3$ indicates that increase of approximately 10 i.e. $(-5,10,25)$ units can be made along the path traced out in a move from source to sink. We can easily work backward to find the path. The label on the sink shows that we came from node 2. From node 2 it is seen that we came from node 1. The path is $1 \to 2 \to 5$. Before starting the fourth iteration, we first set the network showing the effect of third breakthrough. To have fresh approximate excess capacities subtract $\tilde{f}_3 = (-5,10,25)$ from the direction of flow in the path and add this amount in the opposite direction. Compute the approximate residual capacities of each arc on the path. After third iteration arc $(1,2)$ has approximate residual in direction of flow $(-5,10,25)\,\Theta\,(-5,10,25) = (-30,0,30)$. And has in opposite direction $(5,10,15) \oplus (-5,10,25) = (0,20,40)$. Similarly, arc $(2,5)$ has $(0,20,40)$ and $(-5,10,25)$ in the direction of flow and in opposite direction.

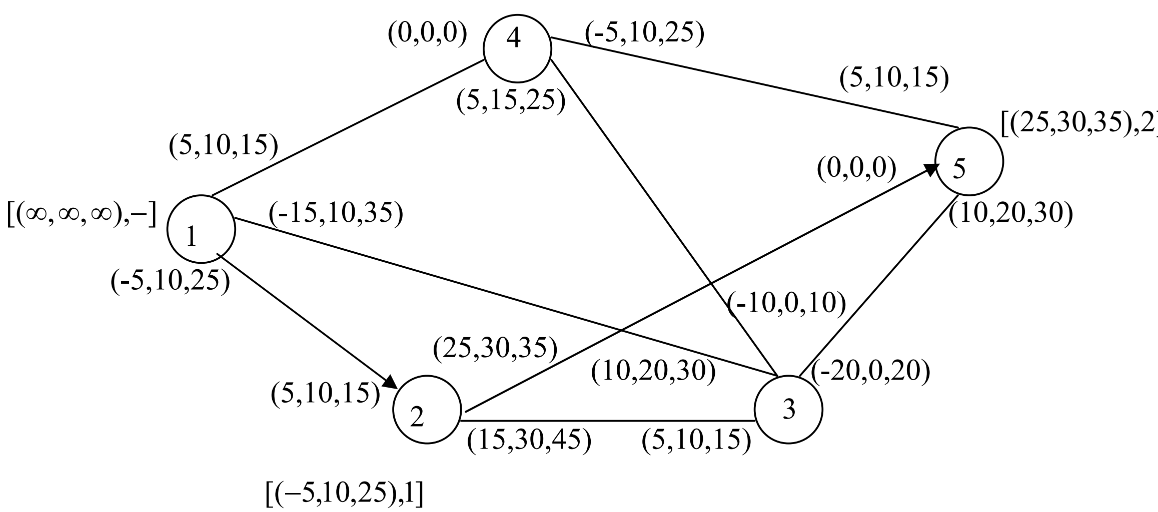


Figure 5.4

***Iteration* 4**

Again start from source node 1 and find the nodes that can be reached directly from the source by arc of approximate excess capacities. We will choose the path having maximal approximate flow limit i.e. limit with maximum rank. Since the rank of all the approximate flow limits $(5,10,15)$ and $(-15,10,35)$ going from source node 1 are same so we can arbitrarily choose the path say from node 1 to node 3. Node 3 is labeled as $[(-15,10,35),1]$. We can not go to node 1, 4 and 5 so we will move from node 3 to node 2 and then from node 2 to node 5. Nodes 3, 2 and 5 are labeled as $[(-15,10,35),1]$, $[(5,10,15),3]$ and $[(0,20,40),2]$. Now, sink is reached and labeling process stops as we have got fourth breakthrough. The approximate flow in the network can be increased by

$$\tilde{f}_4 = \min\{(\infty,\infty,\infty),-],[(5,10,15),3],[(-15,10,35),1],[(0,20,40),2]\}$$

That approximate flow limit will be minimum, whose rank is minimum. Since the rank of $(-15,10,35)$ is minimum so $\tilde{f}_4$ is $(-15,10,35)$. The value of $\tilde{f}_4$ indicates that increase of approximately 10 i.e. $(-15,10,35)$ units can be made along the path traced out in a move from source to sink. We can easily work backward to find the path. The label on the sink shows that we came from node 2. From node 2 it is seen that we came from node 3 and from node 3 it is seen that we came from node 1. The path is $1 \to 3 \to 2 \to 5$. Before starting the fifth iteration, we first set the network showing the effect of fourth breakthrough. To have fresh approximate excess capacities subtract $\tilde{f}_4$ = $(-15,10,35)$ from the direction of flow in the path and add this amount in the opposite direction. Compute the fuzzy residual capacities of each arc on the path. After fourth iteration arc $(1,3)$ has approximate residual in direction of flow $(-15,10,35)\,\Theta\,(-15,10,35) = (-50,0,50)$. And has in opposite direction $(5,10,15) \oplus (-5,10,25) = (0,20,40)$. Similarly, arc $(3,2)$ has $(-30,0,30)$ and $(0,40,80)$ in the direction of flow and in opposite direction and the arc $(2,5)$ has $(-35,10,55)$ and $(-20,20,60)$ in the direction of flow and in opposite direction.

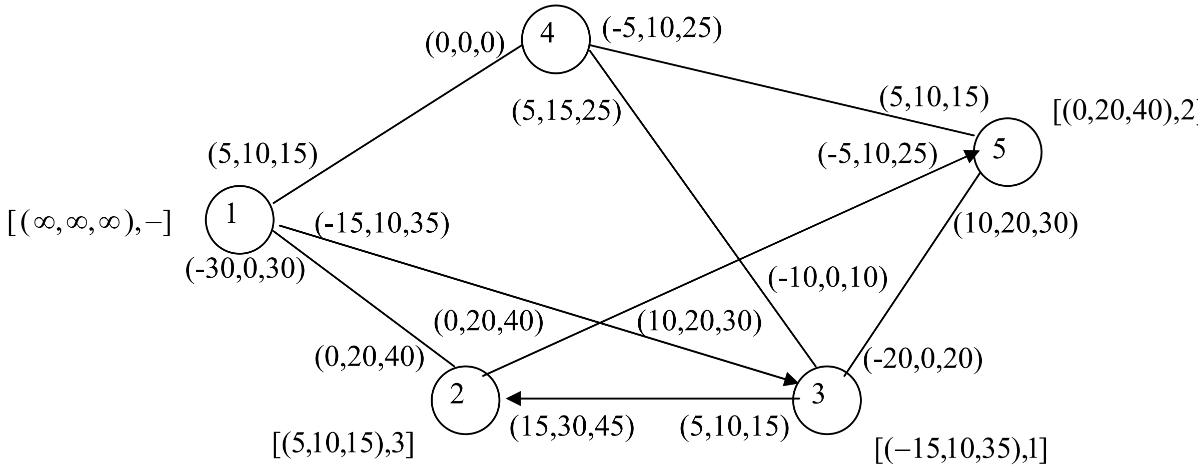


Figure 5.5

***Iteration* 5**

Again start from source node 1 and find the nodes that can be reached directly from the source by arc of approximate excess capacities. We will choose the path having maximal approximate flow limit i.e. limit with maximum rank. We can not go to node 2 and 3 because there is zero flow from node 1 to 2 and 3 so we will move from node 1 to node 4 then from node 4 to 5. Nodes 4 and 5 are labeled as $[(-15,10,35),1]$ and $[(-15,10,35),1]$. Now, sink is reached and labeling process stops as we have got fifth breakthrough. The approximate flow in the network can be increased by

$$\widetilde{f}_5 = \min\{(\infty,\infty,\infty),-],[(5,10,15),1],[(-15,10,25),4]\}$$

That approximate flow limit will be minimum, whose rank is minimum. Since the rank of $(5,10,15)$ and $(-15,10,25)$ are same so $\widetilde{f}_5$ may be $(-15,10,25)$ or $(5,10,15)$. The value of $\widetilde{f}_5$ indicates that increase of approximately 10 i.e. $(-15,10,25)$ or $(5,10,15)$ units can be made along the path traced out in a move from source to sink. We can easily work backward to find the path. The label on the sink shows that we came from node 4. From node 4 it is seen that we came from node 1. The path is $1 \to 4 \to 5$. Before starting the sixth iteration, we first set the network showing the effect of fifth breakthrough. To have fresh approximate excess capacities subtract $\tilde{f}_5 = (5,10,15)$ from the direction of flow in the path and add this amount in the opposite direction. Compute the approximate residual capacities of each arc on the path. After fifth iteration arc $(1,4)$ has approximate residual in direction of flow $(5,10,15)\,\Theta\,(5,10,15) = (-10,0,10)$. And has in opposite direction $(0,0,0) \oplus (-5,10,15) = (-5,10,15)$. Similarly, arc $(4,5)$ has $(-20,0,20)$ and $(10,20,30)$ in the direction of flow and in opposite direction.

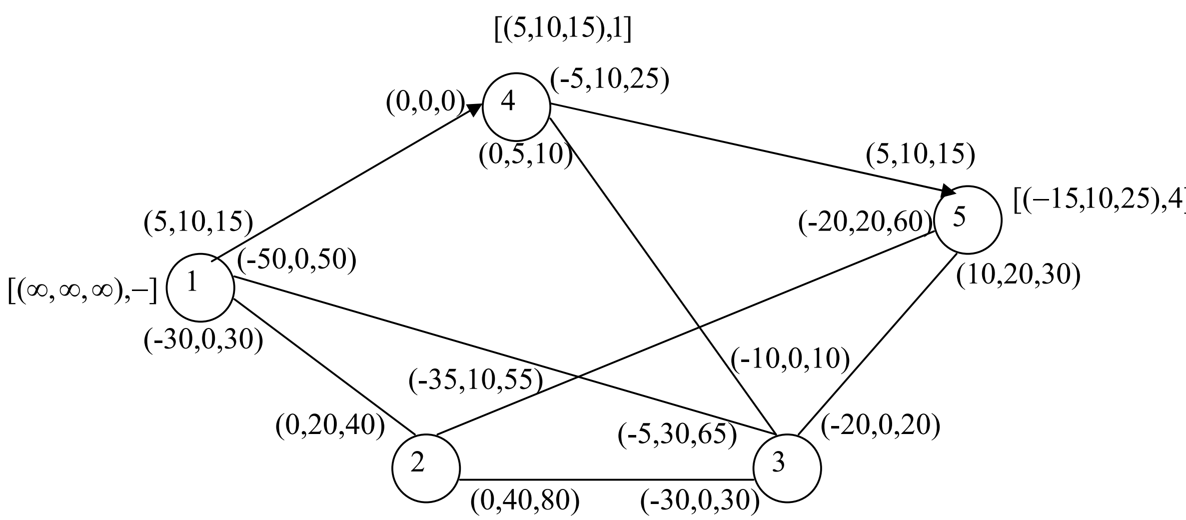


Figure 5.6

More iterations are not possible after 5th iteration a there is no way out to reach at sink from source. The approximate maximum flow is

$$\tilde{f} = \tilde{f}_1 \oplus \tilde{f}_2 \oplus \tilde{f}_3 \oplus \tilde{f}_4 \oplus \tilde{f}_5 = (10,20,30) \oplus (5,10,15) \oplus (-5,10,25) \oplus (-15,10,35) \oplus (5,10,15)$$
$$= (0,60,120)$$

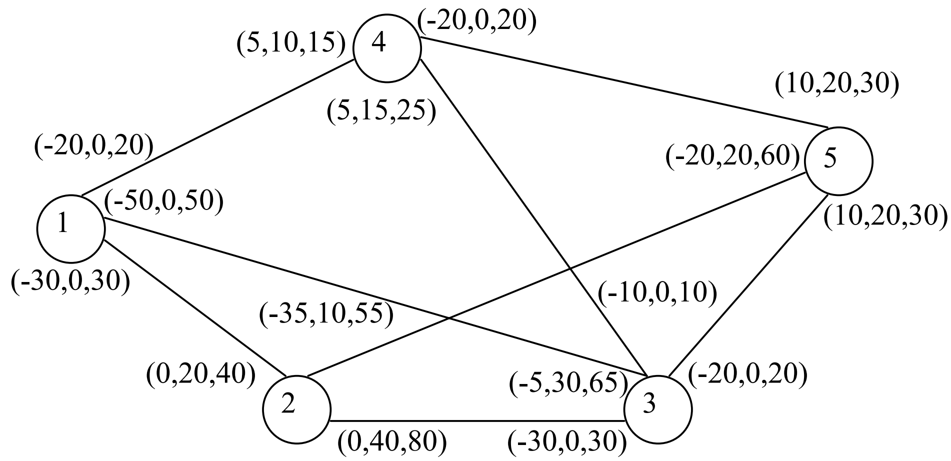


Figure 5.7 (No Break Through)

5.3 Result and discussion

The obtained result can be explained as follows:

1) The flow between source and sink is greater than 60 and less than 120.

2) Maximum number of persons are in favour that flow will be 60.

3) The percentage of person increases when flow varies from 0 to 60 and decreases from 60 to 120.

4) The membership function for the obtained result is shown in Figure 5.8 where *x*-axis denotes required maximum flow between source and other nodes and *y*-axis denotes its membership value.

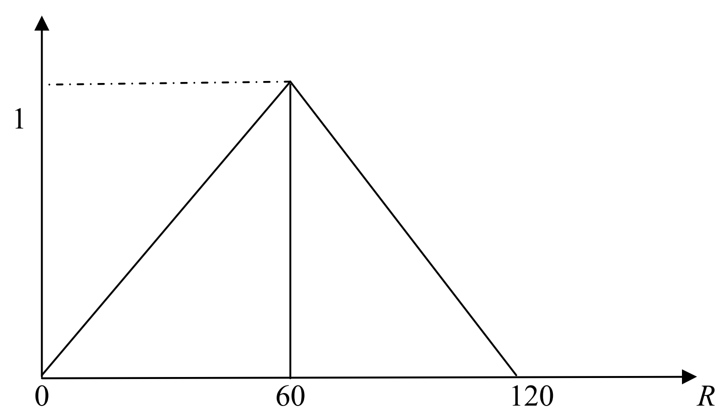


Figure 5.8 (Triangular fuzzy number)

5.4 Conclusion

In this chapter, a new algorithm has been proposed to solve the fuzzy maximal flow problem occurring in real life. To illustrate the algorithm a numerical example has been solved and result are explained. If there is no uncertainty about the flow between source and sink then the proposed algorithm gives the same result as in crisp maximal flow problem.

REFRENCES

[1] Ahuja, R. K., Magnanti, T. L., Orlin, J. B., Network flows, theory, algorithms and applications, New Jersey, Prentice Hall, 1993.

[2] Aron, I., van Hentenryck, P., On the complexity of the robust spanning tree problem with interval data. Operation Research Letters, 32 (2004), 36–40.

[3] Averbakh, I., and Lebedev, V., Interval data min max regrets network optimization problems, Discrete Applied Mathematics, 138 (2004), 289–301.

[4] Campos, L., Verdegay, J. L., Linear programming problems and ranking of fuzzy numbers, Fuzzy Sets and Systems, 32 (1989), 1-11.

[5] Chanas, S., Kolodziejczyk, W., Maximum flow in a network with fuzzy arc capacities, Fuzzy Sets and Systems, 8 (1982), 165-173.

[6] Chanas, S., Kolodziejczyk, W., Real-valued flows in a network with fuzzy arc capacities, Fuzzy Sets and Systems, 13 (1984), 139-151.

[7] Chanas, S., and Kolodziejczyk, W., Integer flows in network with fuzzy capacity constraints, Networks, 16 (1986), 17-31.

[8] Chuang, T. N., Kung, J.-Y., The fuzzy shortest path length and the corresponding shortest path in a network, Comput. Oper. Res., 32 (2005), 1409–1428.

[9] Chuang, T. N., Kung, J.-Y., A new algorithm for the discrete fuzzy shortest path problem in a network, Appl. Math. Comput., 174 (2006).

[10] Cheng. C. H, A new approach for ranking fuzzy numbers by distance method, Fuzzy Sets and Systems, 95 (1998), 307–317.

[11] Dubois, D., Prade, H., Fuzzy Sets and Systems: Theory and Applications, Academic Press, New York, 1980.

[12] Fortemps, P., Roubens, M., Ranking and defuzzification methods based on area compensation, Fuzzy Sets and Systems, 82 (1996) 319–330.

[13] Kalir, G. J., Yuhan, B., Fuzzy sets and fuzzy logic: Theory and applications, Prentice Hall, 1995.

[14] Kasana, H. S., Kumar, K. D., Introductory Operations Research Theory and Application, Springer, 2008.

[15] Kasperski, A., Zieli ´ nski, P., An approximation algorithm for interval data min max regret combinatorial optimization problems, Information Processing Letters, 97 (2006), 177–180.

[16] Kaufmann, A., Gupta, M. M., Introduction to Fuzzy Arithmetics, Theory and Application, New York, Van Nostrand Reinhold, 1991.

[17] Kim, K., Roush, F., Fuzzy flows on network, Fuzzy Sets and Systems. 8 (1982), 35-38.

[18] Klein, C. M., Fuzzy shortest paths, Fuzzy Sets and Systems, (1991), 27–41.

[19] Kouvelis, C., Yu, G., Robust discrete optimization and its applications, Boston: Kluwer Academic Publishers, 1997.

[20] Kung, J. Y., Chuang, T. N., The shortest path problems with discrete fuzzy arc lengths, Computers and Mathematics with Applications, 49 (2005), 263-270.

[21] Lin, C., Chern, M. S., The fuzzy shortest path problem and its most vital arcs, Fuzzy Sets and Systems, 58 (1993), 343–353.

[22] Liou, T. S., Wang. M. J., Ranking fuzzy numbers with integral value, Fuzzy Sets and Systems, 50 (1992), 247–255.

[23] Liu, S. T., Kao, C., Network flow problems with fuzzy arc lengths, IEEE Transactions on Systems, Man, and Cybernetics, Part B, 34 (2004), 765-769.

[24] Moreno, J. A., Moreno, J. M., Verdegay, J. L., Fuzzy location problems on networks, Fuzzy Sets and Systems, 142 (2004) 393–405.

[25] Nayeem, S. M. A., Pal. M, Shortest path problem on a network with imprecise edge weight, Fuzzy Optim. Decis. Making, 4 (2005), 293–312.

[26] Okada, S., Fuzzy shortest path problems incorporating interactivity among paths, Fuzzy Sets and Systems, 142 (2004), 335–357.

[27] Okada, S., and Soper, T., A shortest path problem on a network with fuzzy arc lengths, Fuzzy Sets and Systems, 109 (2000), 129-140.

[28] Pant, J. C., Introduction to optimization, Jain Brothers, New Delhi, 1998

[29] Sengupta, A., Pal, T. K., Solving the shortest path problem with intervals arcs, Fuzzy Optim. Decis. Making, 5 (2006), 71–89.

[30] Yager, R. R., A procedure for ordering fuzzy subsets of the unit interval, Inform. Sci., 24 (1981), 143–161.

[31] Yaman, H., Kara¸ san, O. E., Pinar, M. C., The robust spanning tree problem with interval data. Operations Research Letters, 29 (2001), 31–40.

[32] Zadeh, L., Fuzzy sets, Inf. and control, 8 (1965), 338-353.

[33] Zimmermann, H. J., Fuzzy Sets Theory and its Application, 2nd edition, (Kluwer Academic Publisher, Boston, 1991.