Intern. J. Fuzzy Mathematical Archive
Vol. 9, No. 1, 2015, 17-25
ISSN: 2320 –3242 (P), 2320 –3250 (online)
Published on 8 October 2015
www.researchmathsci.org

International Journal of
Fuzzy Mathematical
Archive

Solving Fully Fuzzy Linear Programming Problem Using Successive Over Relaxaion Method and its Comparison with Other Strategies

***Thangaraj Beaula*[1] and *S.Rajalakshmi*[2]**

[1]Department of Mathematics, T.B.M.L. College, Porayar 609 307, TamilNadu, India
e-mail: edwinbeaula@yahoo.co.in

[2]Department of Mathematics, A.V.C.College of Engineering, Mannampandal
Mayiladuthurai, 609 305, India; e-mail: srkave@gmail.com

Received 5 September 2015; accepted 26 September 2015

Abstract. Linear programming (LP) is a widely used optimization method for solving real-life problems because of its efficiency. Linear programming is the process of optimizing a linear function subject to a finite number of line are equality and inequality constraints. Solving linear programming problems efficiently has always been a fascinating pursuit for computer scientists and mathematics. In this paper, we concentrate on solving fuzzy linear programming problems using three different methods. The algorithms of each method are given and the computational results are presented and analyzed in this paper.

Keywords: Fuzzy linear programming problem, triangular fuzzy number, type -2 fuzzy number.

AMS Mathematics Subject Classification (2010): 03E72, 03F55, 90C70, 9 0C05

1. Introduction

The concept of fuzzy numbers and fuzzy arithmetic operations were first introduced by Zadeh [9]. Negotia [8] proposed fuzzy linear programming problem with fuzzy coefficients. Malaki [4] used ranking function to solve fuzzy linear programming problem. Mishmatetal [7] Solved fuzzy number linear programming problem by lexicographic ranking function. Kumaretal [3] introduced a method for finding the fuzzy optimal solution of fully fuzzy linear programming problem with equality constraints. A type -2 fuzzy set is characterized by a fuzzy membership function [5]. Type -2 fuzzy set theory gained more and more attention from researchers in a wide range of scientific areas[6,1]. Ebrahimnejad and Nasseri [2] developed the complementary slackness theorem for solving fuzzy linear programming problem with fuzzy parameters.

2. Preliminaries

Definition 2.1. If X is a collection of objects denoted generically by x, then a **fuzzy set** A in X is defined as a set of ordered pairs $A = \{(x,\mu_A(x)) \mid x \in X\}$, where, $\mu_A(x)$ is called the **membership function** (or MF for short) for the fuzzy set A.

Definition 2.2. The **support** of a fuzzy set A is the set of all points x in X such that $\mu_A(x) > 0$.The **α − level (α −cut)** set of a fuzzy set A is a crisp subset of X and is denoted by $A_\alpha = \{x \varepsilon X / \mu_A(x)) \geq \alpha\}$

Definition 2.3. A fuzzy set A in X is **convex** *if*

$\mu_A(\lambda x + (1-\lambda)y) \geq \min\{(\mu_A(x), (\mu_A(y)\}$,for all x ,y∈ A and $\lambda \in [0,1]$.

Definition 2.4. Any triangular fuzzy number $A(x)=(a_1,a_2,a_3)$, can be written as follows:

$$A(x) = \begin{cases} (x-a_1)/(a_2-a_1), & x \in [a_1, a_2] \\ (a_3-x)/(a_3-a_2), & x \in [a_2, a_3] \\ 0, & x \leq a_1 \ \& \ x \geq a_3 \end{cases}$$

The interval level of confidence is defined as:

$A^\alpha=[A_l^\alpha, A_r^\alpha]=[a_1+\alpha(a_2-a_1), a_3-\alpha(a_3-a_2)]$ where a_1,a_2,a_3, x ε R and α ε[0,1].

Definition 2.5. If $A=(a_1,a_2,a_3)$ and $B=(b_1,b_2,b_3)$ with $a_1,a_2,a_3,b_1,b_2,b_3 \varepsilon R^+$. The arithmetic operations are defined as follows:

$A + B=(a_1+b_1,a_2+b_2,a_3+b_3)$ $A - B=(a_1-b_1,a_2-b_2,a_3-b_3)$

A. $B=(a_1b_1,a_2b_2,a_3b_3)$,A/ $B=(a_1/b_3,a_2/b_2,a_3/b_1)$.

Definition 2.6. A fuzzy eigen value λ is a fuzzy number which is a solution to the equation AX=λX where A is a n by n matrix containing fuzzy numbers and x is a non zero n by1 vector containing fuzzy numbers .

Remark 2.7. We consider $\tilde{0}$ =(0 , 0 , 0) as the zero triangular fuzzy number.

3. Type-2 fuzzy linear programming problem

Definition 3.1. A type -2 fuzzy linear programming problem is defined as

$$\begin{aligned} &\text{Max} \tilde{\tilde{Z}} = \textstyle\sum_{j=1}^{n} \tilde{\tilde{c}}_j \tilde{\tilde{x}}_j \\ &\text{sub to } \textstyle\sum_{j=1}^{n} \tilde{\tilde{a}}_{ij} \tilde{\tilde{x}}_j \leq \tilde{\tilde{b}}_i \qquad i=1,2,\ldots,m \qquad (1) \\ &\tilde{\tilde{x}}_j \geq 0 \qquad j = 1,2,\ldots,n \end{aligned}$$

where $\tilde{\tilde{A}} = (\tilde{\tilde{a}}_{ij})$,$\tilde{\tilde{c}}$,$\tilde{\tilde{b}}$,are (mxn),(1xn),(mx1) intuitionistic fuzzy matrices consisting of trapezoidal type-2 fuzzy number and $\tilde{\tilde{x}}$ is a (nx1) type-2 fuzzy variable matrix.

Definition 3.2. Consider the system $\tilde{\tilde{A}}\tilde{\tilde{X}} = \tilde{\tilde{b}}$ .with $\tilde{\tilde{X}} > 0$, where $\tilde{\tilde{A}}$ is an m × n matrix and $\tilde{\tilde{b}}$ is a m vector. Suppose that rank($\tilde{\tilde{A}}$) = m. Arranging the column of $\tilde{\tilde{A}}$ as [$\tilde{\tilde{B}}$, $\tilde{\tilde{N}}$] that $\tilde{\tilde{B}}$ is an m × m matrix. The vector $\tilde{\tilde{X}} = (\tilde{\tilde{X}}_{\tilde{\tilde{B}}}^T, \tilde{\tilde{X}}_{\tilde{\tilde{N}}}^T)$ where $\tilde{\tilde{X}}_B = B^{-1}\tilde{\tilde{b}}$ is called a basic feasible solution (BFS) of system. Here $\tilde{\tilde{B}}$ is called the basic matrix and $\tilde{\tilde{N}}$ is called the non basic matrix. The components of $\tilde{\tilde{B}}$ are called the basic variables. If $\tilde{\tilde{X}}_{\tilde{\tilde{B}}}>0$, then $\tilde{\tilde{X}}$ is called a non degenerate basic feasible solution. If $\tilde{\tilde{X}}_{\tilde{\tilde{B}}}= 0$, then $\tilde{\tilde{X}}$ is called a degenerate basic feasible solution.

Definition 3.3. Any vector $\tilde{\tilde{x}}\epsilon(F(R))^n$ which satisfies the constraints and non-negative restrictions of problem(1) is said to be a type -2 fuzzy feasible solution.

Definition 3.4. Let X be the set of all type-2 fuzzy feasible solutions of Problem(1). A type-2 fuzzy feasible solution $\tilde{\tilde{x}}_0\epsilon X$ is said to be a type-2 fuzzy optimal solution to(1) if $\tilde{\tilde{c}}\tilde{\tilde{x}}_0 \geq \tilde{\tilde{c}}\tilde{\tilde{x}}$ for all $\tilde{\tilde{x}}\epsilon X$,where
$\tilde{\tilde{c}} = (\tilde{\tilde{c}}_1, \tilde{\tilde{c}}_2, \tilde{\tilde{c}}_3, \dots\dots, \tilde{\tilde{c}}_n)and\tilde{\tilde{c}}\tilde{\tilde{x}} = \tilde{\tilde{c}}_1\tilde{\tilde{x}}_1 + \tilde{\tilde{c}}_2\tilde{\tilde{x}}_2 + \tilde{\tilde{c}}_3\tilde{\tilde{x}}_3 + \cdots.. + \tilde{\tilde{c}}_n\tilde{\tilde{x}}_n$.

4. Successive over relaxation (SOR) method

Successive Over Relaxation (SOR) method is an iterative scheme that uses a relaxation parameter ω and is a generalization of the Gauss-Seidel method in the special case $\omega = 1$. Here the coefficient matrix should be diagonally dominated. Consider a square system of n linear equations with unknown x.AX=b where $A = \begin{pmatrix} a_{11} & a_{12} & & \\ a_{21} & & & \\ & & & \\ a_{n1} & & & a_{nn} \end{pmatrix}$ $X = \begin{pmatrix} x_1 \\ x_2 \\ \\ . \\ . \\ x_n \end{pmatrix}$

and $b = \begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ b_n \end{pmatrix}$

The coefficient matrix A is decomposed into three matrices viz. be a diagonal component D, and strictly lower and upper triangular components L and U: where

$$D = \begin{pmatrix} a_{11} & 0 & 0 \\ & a_{22} & \\ & & 0 \\ 0 & 0 & a_{nn} \end{pmatrix}. \quad L = \begin{pmatrix} 0 & 0 & 0 \\ a_{21} & & \\ & & \\ a_{n1} & a_{n2} & 0 \end{pmatrix} \cdot U = \begin{pmatrix} 0 & a_{12} & a_{1n} \\ & & a_{2n} \\ & & \\ 0 & & 0 \end{pmatrix}.$$

Therefore the system of linear equations is rewritten as
$(D + \omega L)X = \omega b - [\omega U + (\omega - 1)D]X$ for a constant $\omega \succ 1$.

The method of successive over relaxation is an iterative technique that solves the left hand side of this expression for **x**, using previous value for **x** on the right hand side. Analytically, this may be written as
$X^{(k+1)} = (D + \omega L)^{-1}(\omega b - [\omega U + (\omega - 1)D]X^k)$. However, by taking advantage of the triangular form of (D+ωL), the elements of **x** $^{(k+1)}$ can be computed sequentially using forward substitution

$$x_i^{k+1} = (1-\omega)x_i^k + \frac{\omega}{a_{ij}}\left(b_i - \sum_{j \succ 1} a_{ij}x_j^k - \sum_{j \prec 1} a_{ij}x_j^{k+1}\right), \; i = 1,2,\dots,n.$$

4.1. Algorithm for solving fuzzy linear programming Problem using SOR method

Step 1. Convert the type-2 fuzzy linear programming problem into type1 fuzzy linear programming problem using ranking function.

Step 2. If the number of inequalities is equal to the number of variables, go to step 5.

Step 3. If the number of inequalities is less than the number of variables, add the inequalities in the system till the number of inequalities equals the number of variables. Go to step 5.

Step 4. If the number of inequalities is greater than the number of variables, add the number of slack variables in the appropriateinequalities and add +1 on RHS of each of those inequalities.Go tostep 5.

Step 5. Write each of the constraint as

$$x_i^{k+1} = (1-\omega)x_i^k + \frac{\omega}{a_{ij}}\left(b_i - \sum_{j\succ 1} a_{ij}x_j^k - \sum_{j\prec 1} a_{ij}x_j^{k+1}\right) \; i = 1,2,...,n$$

Step 6. Take the initial values as zero. Find the values of x_i. The elements of $\mathbf{x}^{(k+1)}$ can be computed sequentially using forward substitution.

4.2. Numerical example

Consider the fuzzy linear programming problem whose solution is to be found

$$Max \quad \tilde{\tilde{z}} = 0.\tilde{\tilde{3}}\,\tilde{\tilde{x}}_1 + 0.\tilde{\tilde{4}}\,\tilde{\tilde{x}}_2$$

$$s.t \qquad 0.4\,\tilde{\tilde{x}}_1 + 0.\tilde{\tilde{3}}\,\tilde{\tilde{x}}_2 \le 0.\tilde{\tilde{2}}$$

$$0.\tilde{\tilde{6}}\,\tilde{\tilde{x}}_1 + 0.\tilde{\tilde{8}}\,\tilde{\tilde{x}}_2 \le 0.\tilde{\tilde{6}}$$

$$with \qquad \tilde{\tilde{x}}_1 \ge \tilde{\tilde{0}},\ \tilde{\tilde{x}}_2 \ge \tilde{\tilde{0}}.$$

$$Max\ \tilde{\tilde{z}} = ((0.01,0.1,0.2),(0.2,0.3,0.4),(\,0.4,0.5,0.2,0.6))\tilde{\tilde{x}}_1 +$$
$$((0.1,0.2,0.3),(0.2,0.4,0.6),(\,0.5,0.6,0.7))\tilde{\tilde{x}}_2$$
$$s.t \qquad \Big(\,((0.1,0.3,0.5),(0.2,0.4,0.6),(\,0.3,0.5,0.7))\tilde{\tilde{x}}_1 +$$
$$(0.1,0.2,0.3),(0.2,0.3,0.4),(0.3,0.4,0.5))\tilde{\tilde{x}}_2\Big) \le$$
$$((0.01,0.1,0.2),(0.1,0.2,0.3),(\,0.2,0.3,0.4))$$
$$\Big(((0.4,0.5,0.6),(0.5,0.6,0.7),(\,0.6,0.7,0.8))\tilde{\tilde{x}}_1 +$$
$$((0.6,0.7,0.8),(0.7,0.8,0.9),(\,0.8,0.9,0.99))\tilde{\tilde{x}}_2\Big) \le$$
$$((0.1,0.2,0.3),(0.3,0.4,0.5),(\,0.5,0.6,0.7))$$
$$with \qquad \tilde{\tilde{x}}_1 \ge \tilde{\tilde{0}},\ \tilde{\tilde{x}}_2 \ge \tilde{\tilde{0}}.$$

Defuzzifying type -2 linear programming problem into type1 fuzzy linear problem

$$Max\ \tilde{z} = (0.1,0.3,0.5)\tilde{x}_1 + (0.2,0.4,0.6)\tilde{x}_2$$

$$\text{s.t} \quad (0.3,0.4,0.5)\tilde{x}_1 + (0.2,0.3,0.4)\tilde{x}_2 \le ((0.1,0.2,0.3)$$

$$(0.5,0.6,0.7)\tilde{x}_1 + (0.7,0.8,0.9)\tilde{x}_2 \le (0.2,0.4,0.6))$$

$$with \quad \tilde{x}_1 \ge 0, \tilde{x}_2 \ge 0$$

The given fuzzy linear programming problem can be written as

1. $$Max\, z = 0.1x_1 + 0.2x_2$$
$$s.t \quad 0.3x_1 + 0.2x_2 \le 0.1$$
$$0.5x_1 + 0.7x_2 \le 0.2$$
$$with \quad x_1 \ge 0, x_2 \ge 0.$$

2. $$Max\, z = 0.3x_1 + 0.4x_2$$
$$s.t \quad 0.4x_1 + 0.3x_2 \le 0.2$$
$$0.6x_1 + 0.8x_2 \le 0.4$$
$$with \quad x_1 \ge 0, x_2 \ge 0.$$

3. $$Max\, z = 0.5x_1 + 0.6x_2$$
$$s.t \quad 0.5x_1 + 0.4x_2 \le 0.3$$
$$0.7x_1 + 0.9x_2 \le 0.6$$
$$with \quad x_1 \ge 0, x_2 \ge 0.$$

The above coefficient matrix is diagonally dominated. So we can apply SOR method. Now we can solve 1 by using SOR Method

$$Max\, z = 0.1x_1 + 0.2x_2$$
$$s.t \quad 0.3x_1 + 0.2x_2 \le 0.1$$
$$0.5x_1 + 0.7x_2 \le 0.2$$
$$with . \quad x_1 \ge 0, x_2 \ge 0$$

The above constraints can be written as $x_1 = \frac{0.1 - 0.2x_2}{0.3}$, $x_2 = \frac{0.2 - 0.5x_1}{0.7}$

Using SOR method, the above equations may be written as

$$x_1^{k+1} = (1-\omega)x_1^k + \omega\left(\frac{0.1 - 0.2x_2^k}{0.3}\right), x_2^{k+1} = (1-\omega)xk_2^k + \omega\left(\frac{0.2 - 0.5x_1^k}{0.7}\right)$$

Let the initial values be zero. Assume ω=1.25. Using iteration procedure, we get the **solution** is x_1 = 0.2727 and x_2 = 0.0909. Similarly if we solve 2 and 3 set of linear programming problem then **Solution of 2** is x_1= 0.2857 and x_2 = 0.2857. **Solution of 3**. x_1= 0.1765 and x_2 = 0.5294. Hence the solution is of given type-2 fuzzy linear programming problem is x_1= 0.2857 and x_2 = 0.2857 and Max z = 2.

5. Comparison of successive over relaxation (SOR) method with breaking point technique and semi infinite programming method

5.1. Breaking point technique

The breaking points of constraints at satisfactory level γ is defined as

$a_{ij} = c - \gamma(c-b)$. The points in intersection of $a_{ij} = c - \gamma(c-b)$ lines are called the breaking points.

5.2. Algorithm for breaking point method

Step 1. Convert the type-2 fuzzy linear programming problem into type1 fuzzy linear programming problem.

Step 2. Find the breaking points of γ which are intersection points of lines $A_{ij} = A_{ij}(\gamma)$.

Step 3. Using breaking points, rearrange the coefficient of constraints as

$$a_{ij} = c - \gamma(c - b)$$

Step 4. Solve the problem using Simplex method.

5.3. Semi infinite linear programming problem method

In optimization theory, semi-infinite programming (SIP) is an optimization problem with a finite number of variables and an infinite number of constraints, or an infinite number of variables and a finite number of constraints.

5.4. Algorithm for semi infinite linear programming problem method

Step 1. Convert the type-2 fuzzy linear programming problem into type1 fuzzy linear programming Problem.

Step 2. Split this fuzzy problem into two problems by considering the left and right extremum of confidence interval. It takes the form

$$Min \sum_{j=1}^{n} L\tilde{c}_j \tilde{x}_j$$

$$s.t \quad \sum_{j=1}^{n} L\tilde{a}_{ij}(t)\tilde{x}_j \geq L\tilde{b}_i(t),\ i = 1,2,...,\ m, \forall t \in [\alpha,1]$$

$$with \quad \tilde{x}_j \geq 0, \qquad j = 1,2,......,\ n$$

and

$$Min \sum_{j=1}^{n} R\tilde{c}_j \tilde{x}_j$$

$$s.t \quad R\tilde{a}_{ij}(t)\tilde{x}_j \geq R\tilde{b}_i(t),\ i = 1,2,...,\ m, \forall t \in [\alpha,1]$$

$$with \quad \tilde{x}_j \geq 0, \qquad j = 1,2,......,\ n$$

Let $f_{ij}(t) = L_{\tilde{a}_{ij}}(t), \tilde{b}_i(t) = L_{b_i}(t)$ and $g_{ij}(t) = R_{\tilde{a}_{ij}}(t), \tilde{b}_i(t) = R_{b_i}(t)$

for i =1,2,…,m,j =1,2,…,n.

T =[α,1] $\alpha \in [0,1$ .Then the two problems are semi infinite linear programming problem.

Step 3. Fix k=1. Choose $t_i^1 \in T$,set $T_1 = \{t_1^1\}\}$.

Step 4. Solve $(LP)^K$ and obtain an optimal solution x^k (both left space and right space) and obtain an optimal solution x^k using simplex method.

Step 5. Set the constraint violation function as

$$v_i^{(k+1)}(t) = \sum_{j=1}^{n} f_{ij}(t)x_j^k - b_i(t) \quad \forall t \in T, i = 1,2,...,\ m \quad (for \quad left)$$

$$v_i^{(k+1)}(t) = \sum_{j=1}^{n} g_{ij}(t)x_j^k - b_i(t) \quad \forall t \in T, i = 1,2,...,\ m \quad (for \quad right)$$

Step 6. Find minimizers t_i^{k+1} of $v_i^{(k+1)}$ over T for i=1,2,..m.

Step 7. If $v_i^{(k+1)}(t_i^{k+1}) \geq 0$ for i=1,2,..,m,then x^k being an optimal solution. Otherwise go to step 8.

Step 8. Set $T^{k+1} = T_k \cup \{t^{k+1}\}$ *and* $k \leftarrow k+1$. Go to step (1) and repeat the procedure until an optimal solution is obtained.

6. Examples

Numerical example 4.2 is solved using breaking point technique and semi infinite programming method**.**

6.1. Breaking point technique

$$Max\ \tilde{z} = (0.1,0.3,0.5)\tilde{x}_1 + (0.2,0.4,0.6)\tilde{x}_2$$

$$\text{s.t} \quad (0.3,0.4,0.5)\tilde{x}_1 + (0.2,0.3,0.4)\tilde{x}_2 \leq ((0.1,0.2,0.3)$$

$$(0.5,0.6,0.7)\tilde{x}_1 + (0.7,0.8,0.9)\tilde{x}_2 \leq (0.2,0.4,0.6))$$

$$with \quad \tilde{x}_1 \geq 0, \tilde{x}_2 \geq 0$$

Standardize the linear programming problem by introducing slack variables $\tilde{x}_3, \tilde{x}_4 \geq 0.$

The breaking points of γ are intersection points oflines

$a_{ij} = a_{ij}$ (γ). Breaking values of γ in (0, 1) interval are 0, ½, 1.Intervals among these iterative values are [0, 1/2], [1/2, 1].

$\tilde{c}_B$	$\tilde{x}_B$	V $\tilde{x}_B$	$\tilde{x}_1$	$\tilde{x}_2$	$\tilde{x}_3$	$\tilde{x}_4$
$\tilde{0}$	$\tilde{x}_3$	$0.3 - 0.1\gamma$	$0.5 - 0.1\gamma$	$0.4 - 0.1\gamma$	$2 - \gamma$	$\tilde{0}$
$\tilde{0}$	$\tilde{x}_4$	$0.6 - 0.2\gamma$	$0.7 - 0.1\gamma$	$0.9 - 0.1\gamma$	$\tilde{0}$	$2 - \gamma$
$Max\ \tilde{z}$	$\tilde{0}$					
	$\tilde{z}_j - \tilde{c}_j$		(-0.5,-0.3,-0.1)	(-0.6,-0..4 - 0.2)	$\tilde{0}$	$\tilde{0}$

Table 1: Initial table

Using the Simplex procedure, we get the solution as

$$\tilde{x}_1 = 0,\ \tilde{x}_2 = \frac{0.6 - 0.2\gamma}{0.9 - 0.1\gamma}$$

If we substitute γ=1, Max $\tilde{z}$ = (0.1, 0.2, 0.3) x_1= 0, x2 =0.5.

If we allow x_1 to enter into the basis, we get an alternate optimal solution x_1=0.2857,x_2=0.2857.

6.2. Semi-infinite linear programming problem

Method

$$Max\,\tilde{z} = (0.1,0.3,0.5)\tilde{x}_1 + (0.2,0.4,0.6)\tilde{x}_2$$

s.t $(0.3,0.4,0.5)\tilde{x}_1 + (0.2,0.3,0.4)\tilde{x}_2 \le ((0.1,0.2,0.3)$

$(0.5,0.6,0.7)\tilde{x}_1 + (0.7,0.8,0.9)\tilde{x}_2 \le (0.2,0.4,0.6))$

with $\tilde{x}_1 \ge 0, \tilde{x}_2 \ge 0$

$$Min\,\tilde{z} = -Max(-\tilde{z}) = (\text{-}0.5,\text{-}0.4,\text{-}0.3)\tilde{x}_1 + (-0.6,-0.4,-0.2)\tilde{x}_2$$

s.t $(\text{-}0.5,\text{-}0.4,\text{-}0.3)\tilde{x}_1 + (\text{-}0.4,\text{-}0.3\text{-}\ 0.2)\tilde{x}_2 \ge (-0.3,-0.2,-0.1)$

$(-0,7,-0.6,-0.5)\tilde{x}_1 + (-0.9,-0.8,-0.7)x_2 \ge (-0.6,-0.4,-0.2)$

with $\tilde{x}_1 \ge 0, \tilde{x}_2 \ge 0$

Separating left and right space, we have the following two linear semi -infinite programming problem.

1. $$\begin{aligned} &\min \quad (-0.2t-0.1)x_1 + (-0.2-0.2)x_2 \\ &s.t \quad [-0.1t_1+0.3]x_1 + [-0.1t_1-0.2]x_2 \ge [-0.1t_1-0.1] \\ &\quad [-0.1t_2-0.5]x_1 + (-0.1t_2-0.7)x_2 \ge [-0.2t_2-0.2] \\ &with \quad x_1, x_2 \ge 0 \quad \forall t_i \in [\alpha,1] \end{aligned}$$ (left space)

and

2. $$\begin{aligned} &\min \quad (0.2t-0.5)x_1 + (0.2t-0.6)x_2 \\ &s.t \quad [0.1t-0.5_1]x_1 + [0.1-0.4t_1]x_2 \ge_\alpha [0.1t_1-0.3] \\ &\quad [0.1t_2-0.7]x_1 + [0.1t_2-0.9]x_2 \ge_\alpha [0.2t_2-0.6] \\ &with \quad x_1, x_2 \ge 0 \quad \forall t_i \in [\alpha,1] \end{aligned}$$ (right space)

Using the *algorithm* we get the solution as follows:

Left space solution is $\tilde{x}_1 = 0$, $\tilde{x}_2 = 0$=0.421

Right space solution is $\tilde{x}_1$=0 $\tilde{x}_2$=0.5 or x_1=0.2857,x_2 = 0.2857

Hence the solution is $\tilde{x}_1$ = 0, $\tilde{x}_2$ = [0.421,5].Since the problem deals with maximization, the solution is $\tilde{x}_1$=0,$\tilde{x}_2$=0.5 and $\tilde{x}_1$=0.2857,$\tilde{x}_2$=0.2857

7. Conclusion

Type- 2 fuzzy linear programming problem has been solved using SOR Method and it is compared with breaking point technique and semi-programming methods and their results are summarized as

Method Name	$\tilde{x}_1$	$\tilde{x}_2$	$Max\tilde{z}$
Semi infinite Programming Problem	0.2857	0.2857	2
Breaking Point Method	0.2857	0.2857	2
SOR method	0.2857	0.2857	2

In all these three methods, it is to be noted that the optimal solutions are equal.

REFERENCES

1. O.Castillo and P.Melin, Type-2 fuzzy logic –Theory and applications, Heidelberg Springer- Verlog Berlin, 2008.
2. A.Ebrahimnejad and S.H.Nasseri, Using complementary slackness property to solve linear programming problem with fuzzy parameters, *Fuzzy Information and Engineering,* 3 (2009) 233-245.
3. A.Kumar, J.Kaur and P.Singh, A new method for solving fully fuzzy linear programming problems, *Applied Mathematical Modeling,* 35 (2011) 817-823.
4. H.R.Malaki, Ranking function and their applications to fuzzy linear programming, *For East Journal Mathematics Sciences*, 4 (2002) 283-301.
5. J.M.Mendel, Uncertain Rule-Based fuzzy logic systems:Introduction to new directions, Prentice Hall, Upper Saddle River –NJ (2001).
6. J.M.Mendel, Advance in type-2 fuzzy sets and systems, *Information Sciences*, 177(1) (2001) 84-110
7. H.Mishmast Nehi, H.R.Maleki and M.Maohinchi, Solving fuzzy number linear programming problem by lexicographic ranking function, *Italian Journal of Pure and Applied Mathematics*,15 (2004) 19-20.
8. C.V.Negotia, Fuziness in Management, OPSA/TIMS, Miami, 1970.
9. L.A.Zadeh, The concept of a linguistic variable and its application to approximate reasoning, *Information Sciences*, 8 (1975) 199-249.