

<https://doi.org/10.48047/AFJBS.6.10.2024.6501-6513>



## ROLE OF R PROGRAMMING AND LINGO SOFTWARE IN FUZZY LINEAR PROGRAMMING PROBLEMS IN DATA ANALYSIS

**First Author: Mohamed Salih Mukthar M \***

Department of Mathematics, C. Abdul Hakeem College (Autonomous),  
(Affiliated to Thiruvalluvar University, Serkkadu, Vellore – 632115), Melvisharam-632509, India;  
Email ID: [salih.mat@cahc.edu.in](mailto:salih.mat@cahc.edu.in) , Orcid ID: <https://orcid.org/0000-0002-2457-5314>

**Second Author: Ramathilagam S**

Department of Mathematics, Periyar Arts College,  
(Affiliated to Thiruvalluvar University, Serkkadu, Vellore – 632115), Cuddalore-607001, India;  
Email ID: [ramathilagam@pacc.in](mailto:ramathilagam@pacc.in), Orcid ID: <https://orcid.org/0009-0002-4145-5629>

\* *Correspondence:* [salih.mat@cahc.edu.in](mailto:salih.mat@cahc.edu.in)

### Article Info

Volume 6, Issue 10, 2024

Received: 20 May 2024

Accepted: 15 June 2024

Doi:  
10.48047/AFJBS.6.10.2024.6501-  
6513

### ABSTRACT

The paper begins with an introduction to the concept of fuzzy optimization, providing an in-depth explanation of its significance and applications. It then proceeds to define two distinct ranking functions that is Yager's and  $\nu_R$  – ranking functions within the context of fuzzy optimization along with the properties of  $\nu_R$ - ranking function. Following this, a comprehensive overview of Lingo software and R programming is presented, delving into their respective features and capabilities. This paper further delves into a detailed comparison of Lingo and R programming, offering a thorough exploration of their strengths and limitations through the use of example problems. These examples illustrate the process of computing optimal solutions for fully fuzzy linear programming problems using both Lingo and R programming, providing a comprehensive and practical comparison of their effectiveness in this context.

**Keywords:** Fuzzy optimization, Lingo, R programming, Fully fuzzy linear programming

## 1. INTRODUCTION

Optimization is the process of finding the best solution, often involving minimizing or maximizing the value of the objective. Optimization problems revolve around maximizing or minimizing a function within a given set, typically representing a range of available choices in a specific situation. This function allows for the comparison of different choices to determine the best option. The fuzzy concept addresses issues related to ambiguous, subjective, and imprecise judgments. It quantifies the linguistic aspect of available data and preferences for individual or group decision-making.

The concept of solving fuzzy optimization, particularly fuzzy linear programming problems, has garnered attention in the academic community since the first works in fuzzy optimization were published in 1974. Fuzzy Linear Programming model was introduced by

Zimmermann, who proposed a method to solve Linear Programming problems with fuzzy linear constraints. His contributions have influenced many other works and opened the door to involve fuzzy reasoning in optimization. Therefore, it is important to refer to related fuzzy mathematical models and methods. Numerous related works, extensions, and applications have been published after Zimmermann's seminal works in 1977. Various scholars, including mathematicians, economists, and engineers, have contributed to the development of fuzzy mathematical programming based on the principles given by Bellman and Zadeh, who provided a unified framework for fuzzy decision-making. More specific reviews were published by Ebrahimnejad and Verdegay [1], [2] who reviewed several solution methods for FLPs according to the classification proposed by Shams et al. [3], who published a brief review on several FLP models and suggested three classes only; Ghanbari et al. [4], who recently published an extensive review of methods for solving FLPs and classified them into five categories depending on fuzzy parameters; and Sotoudeh-Anvari [5], who has made an interesting criticism about some drawbacks and mathematical incorrect assumptions in fuzzy OR methods, including fuzzy linear programming problems from 2010 to 2020.

## 2. MATERIALS

In this section, first, some basic concepts will be discussed.

### Definition 2.1

A fuzzy number  $\widetilde{A}_T = (p, q, r, s)$  is said to be a trapezoidal fuzzy number [6] if its

membership function is given by: 
$$\mu_{\widetilde{A}_T}(x) = \begin{cases} \frac{x-p}{q-p}, & p \leq x \leq q \\ 1, & q \leq x \leq r \\ \frac{x-s}{r-s}, & r \leq x \leq s \\ 0, & \text{else} \end{cases}$$

### Definition 2.2

A trapezoidal fuzzy number  $\widetilde{A}_T = (p_T^L, p_T^U, p_1, p_2)$  can be shown by support and core as in the Figure -1.

Its support is  $[p_T^L - p_1, p_T^U + p_2]$  and core is  $[p_T^L, p_T^U]$ .

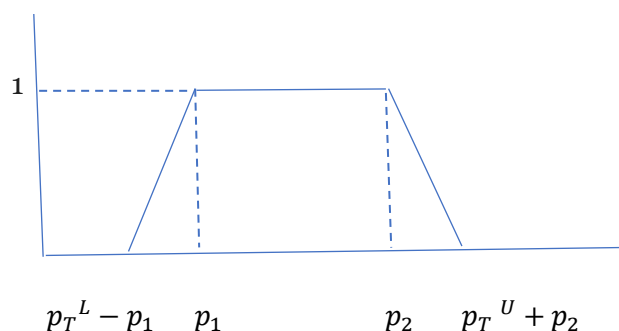


Figure-1

**Definition 2.3**

A fuzzy linear programming problem in which it contains all the parameters are fuzzy is called a fully fuzzy linear programming problem.

It can be defined as follows [6]:

$$\text{Maximize (or Minimize) } \tilde{z} = \sum_{j=1}^n \tilde{c}_j \tilde{x}_j$$

Subject to the constraints

$$\sum_{j=1}^n \tilde{a}_{ij} \tilde{x}_j \leq, \geq, = \tilde{b}_i, \quad i = 1, 2, 3, \dots, m$$

$$\tilde{x}_j \geq 0, \text{ for all } j, \text{ where } \tilde{a}_{ij}, \tilde{x}_j, \tilde{c}_j, \tilde{b}_i \text{ are trapezoidal fuzzy numbers.}$$

**Definition 2.4**

Yager’s ranking function [7] of a trapezoidal fuzzy number  $\tilde{A}_T = (p_T^L, p_T^U, p_1, p_2)$

$$\text{is defined as } \mathfrak{R}(\tilde{A}_T) = \frac{1}{2} \left( p_T^L + p_T^U - \frac{4}{5} p_1 + \frac{2}{3} p_2 \right)$$

**Definition 2.5**

$v_R$ - ranking function of a trapezoidal fuzzy number  $\tilde{A}_T = (p_T, q_T, r_T, s_T)$  is defined as  $\vartheta_{\mathcal{R}}(\tilde{A}_T) = \frac{p_T+2q_T+2r_T+s_T}{6}$ , which is being computed by Mary George and Savitha M.T [8] using the idea of Adrian.I Ban and Lucian Coroianu [9].

**Proposition 2.5.1**

For any trapezoidal fuzzy numbers  $\tilde{A}_T$  and  $\tilde{B}_T$ ,  $\vartheta_{\mathcal{R}}(\tilde{A}_T + \tilde{B}_T) = \vartheta_{\mathcal{R}}(\tilde{A}_T) + \vartheta_{\mathcal{R}}(\tilde{B}_T)$

**Proof**

Let  $\tilde{A}_T = (p_T, q_T, r_T, s_T)$  and  $\tilde{B}_T = (u_T, w_T, x_T, y_T)$

$$\tilde{A}_T + \tilde{B}_T = (p_T, q_T, r_T, s_T) + (u_T, w_T, x_T, y_T)$$

$$= (p_T + u_T, q_T + w_T, r_T + x_T, s_T + y_T)$$

$$\text{By definition, } \vartheta_{\mathcal{R}}(\tilde{A}_T) = \frac{p_T+2q_T+2r_T+s_T}{6}$$

$$\begin{aligned} \vartheta_{\mathcal{R}}(\tilde{A}_T + \tilde{B}_T) &= \vartheta_{\mathcal{R}}(p_T + u_T, q_T + w_T, r_T + x_T, s_T + y_T) \\ &= \frac{(p_T + u_T) + 2(q_T + w_T) + 2(r_T + x_T) + (s_T + y_T)}{6} \\ &= \frac{(p_T+2q_T+2r_T+s_T)+(u_T+2w_T+2x_T+y_T)}{6} \\ &= \frac{p_T + 2q_T + 2r_T + s_T}{6} + \frac{u_T + 2w_T + 2x_T + y_T}{6} \end{aligned}$$

$$= \vartheta_{\mathcal{R}}(\widetilde{A}_T) + \vartheta_{\mathcal{R}}(\widetilde{B}_T)$$

### Proposition 2.5.2

For any finite trapezoidal fuzzy numbers  $\widetilde{A}_{T1}, \widetilde{A}_{T2}, \dots$ ,  $\vartheta_{\mathcal{R}}(\widetilde{A}_{T1} + \widetilde{A}_{T2} + \dots) = \vartheta_{\mathcal{R}}(\widetilde{A}_{T1}) + \vartheta_{\mathcal{R}}(\widetilde{A}_{T2}) + \dots$

**Proof** is consequence of proposition 2.5.1.

### Proposition 2.5.3

For any trapezoidal fuzzy number  $\widetilde{A}_T$ ,  $\vartheta_{\mathcal{R}}(\alpha \widetilde{A}_T) = \alpha \vartheta_{\mathcal{R}}(\widetilde{A}_T)$ , where  $\alpha$  is a scalar.

**Proof**

$$\text{Let } \widetilde{A}_T = (p_T, q_T, r_T, s_T)$$

$$\text{By definition, } \vartheta_{\mathcal{R}}(\widetilde{A}_T) = \frac{p_T + 2q_T + 2r_T + s_T}{6}$$

$$\begin{aligned} \vartheta_{\mathcal{R}}(\alpha \widetilde{A}_T) &= \vartheta_{\mathcal{R}}(\alpha(p_T, q_T, r_T, s_T)) \\ &= \vartheta_{\mathcal{R}}(\alpha p_T, \alpha q_T, \alpha r_T, \alpha s_T) \\ &= \frac{\alpha p_T + 2\alpha q_T + 2\alpha r_T + \alpha s_T}{6} \\ &= \alpha \left( \frac{p_T + 2q_T + 2r_T + s_T}{6} \right) \\ &= \alpha \vartheta_{\mathcal{R}}(\widetilde{A}_T) \end{aligned}$$

## 3. OVERVIEW OF LINGO AND R PROGRAMMING

The LINGO software [10] is a powerful tool specifically designed to effectively handle a wide range of optimization problems. It is capable of addressing linear, non-linear, mixed integer linear, and non-linear programming challenges. Within LINGO, the Solver command is utilized to find solutions to these optimization problems, employing three distinct algorithms: Simplex, Dual Simplex, and Barrier.

It is crucial to understand that LINGO is adept at solving crisp linear programming problems but does not offer direct support for solving fuzzy linear programming problems. However, it is possible to leverage LINGO to obtain optimal solutions for fuzzy linear programming problems by first converting them into crisp linear programming problems.

R is a programming language [11] primarily used for statistical computation, graphical representation, visualization, and data analysis. It runs on multiple platforms, such as Windows,

Mac, Linux, etc. Additionally, R is an open-source software and it has an extensive R archive network that contains more than 100000 packages.

The techniques used to solve fully fuzzy linear programming problems vary depending on the specific nature of the problem. The fuzzy algebraic method and fuzzy simplex method are two prominent approaches employed in certain cases. However, it is important to acknowledge that no single method can comprehensively address all types of fully fuzzy linear programming problems. In addition to these two methods, there are various other specialized techniques available for addressing specific scenarios. Currently, R programming is increasingly being used to solve most types of fully fuzzy linear programming problems.

### 3.1 R COMMANDS FOR FUZZY LINEAR PROGRAMMING

The following are the R commands [12] for applying fuzzy linear programming problems.

#### FuzzyLP

It is a package to solve fuzzy linear programming problems in R.

#### FCLP.classicObjective

It is used to find the solution of the linear programming problem which contains fuzzy constraints and crisp objective.

#### FCLP.fixedBeta

It solves a fuzzy linear programming problem with fuzzy constraints.

#### FOLP.multiObj

It solves a fuzzy objective linear programming problem using the Representation Theorem through a multiobjective approach.

#### FOLP.posib

It solves a fuzzy objective linear programming problem using the Representation Theorem through a possibilistic approach.

#### FOLP.ordFun

It solves a fuzzy objective linear programming problem using ordering functions

#### GFLP

It is used to solve fully fuzzy linear programming problems that have fuzzy coefficients in the constraints, the objective function and the technological matrix.

#### FuzzyNumbers

It is an open source (LGPL 3) package for R. It provides S4 classes and methods to deal with fuzzy numbers.

#### TrapezoidalFuzzyNumber()

It represents trapezoidal fuzzy numbers in R.

#### 4. PROBLEM STATEMENTS AND SOLUTIONS

##### Problem Statement 4.1

A software company produces two types of system-level solutions. It has been estimated that the production of a system-level solution of type 1 needs around 3 working months (w.m.) for the construction of the knowledge base, about 3.5 w.m. for the selection of the suitable reasoning and inference methodology, and nearly 1 w.m. for the choice of the proper intelligent authoring shells. Also, the production of a system-level solution of type 2 needs around 1, 3, and 2 w.m. respectively, for each of the above procedures. According to the company's existing number of specialized staff, at most 20 w.m. per year can be spent for the construction of the knowledge base, at most 30 w.m. for the selection of the reasoning and inference methodology, and at most 18 w.m. for the selection of the intelligent authoring shells. If the net profit from the sale of a system-level solution of type 1 nears 300 thousand euros and of a system-level solution of type 2 nears 400 thousand euros, find how many system-level solutions of type 1 and type 2 should be produced per year to maximize the company's total profit.

##### Solution

This problem is having uncertainty at various parameters. So, using trapezoidal fuzzy numbers, it can be formulated into a fully fuzzy linear programming problem.

Let  $q_1$  and  $q_2$  represent the quantities of system-level solutions of type 1 and type 2 respectively to be produced per year.

Then, the problem is mathematically formulated as follows:

$$\begin{aligned} \text{Max } Z &= (2.6, 2.8, 3.0, 3.2) q_1 + (3.6, 3.8, 4.0, 4.2) q_2 \\ (2.6, 2.8, 3.0, 3.2) q_1 + (0.6, 0.8, 1.0, 1.2) q_2 &\leq (19.6, 19.8, 20.0, 20.2) \\ (3.1, 3.3, 3.5, 3.7) q_1 + (2.6, 2.8, 3.0, 3.2) q_2 &\leq (29.6, 29.8, 30.0, 30.2) \\ (0.6, 0.8, 1.0, 1.2) q_1 + (1.6, 1.8, 2.0, 2.2) q_2 &\leq (17.6, 17.8, 18.0, 18.2) \\ q_1, q_2 &\geq 0 \end{aligned}$$

To use LINGO software for this problem, first convert it into a crisp linear programming problem by using  $v_R$ - ranking function  $\vartheta_{\mathcal{R}}(\widetilde{A}_T) = \frac{p+2q+2r+s}{6}$ .

Corresponding Crisp LPP is

$$\begin{aligned} \text{Max } Z &= 2.9 q_1 + 3.9 q_2 \\ \text{Subject to the constraints} \\ 2.9 q_1 + 0.9 q_2 &\leq 19.9 \\ 3.4 q_1 + 2.9 q_2 &\leq 29.9 \\ 0.9 q_1 + 1.9 q_2 &\leq 17.9 \\ q_1, q_2 &\geq 0 \end{aligned}$$

Applying Lingo software, the output will be as follows:

```
Global optimal solution found.
Objective value:                38.08182
Infeasibilities:                0.000000
Total solver iterations:        2
Elapsed runtime seconds:        0.71

Model Class:                    LP

Total variables:                2
Nonlinear variables:            0
Integer variables:              0

Total constraints:              4
Nonlinear constraints:          0

Total nonzeros:                 8
Nonlinear nonzeros:            0
```

Variable	Value	Reduced Cost
Q1	1.272727	0.000000
Q2	8.818182	0.000000

Row	Slack or Surplus	Dual Price
1	38.08182	1.000000
2	8.272727	0.000000
3	0.000000	0.5194805
4	0.000000	1.259740

Hence the optimal solution has been achieved with quantities of system-level solution type 1 at 1.272727 and type 2 at 8.818182 to maximize company profit of 38%.

Next, the same problem will be solved by using the R program.

For this, the following are the R commands to enter into the software:

```
> obj=c(FuzzyNumbers::TrapezoidalFuzzyNumber(2.6,2.8,3,3.2),FuzzyNumbers::
TrapezoidalFuzzyNumber(3.6,3.8,4,4.2))
> a11=FuzzyNumbers::TrapezoidalFuzzyNumber(2.6,2.8,3,3.2)
> a12=FuzzyNumbers::TrapezoidalFuzzyNumber(0.6,0.8,1,1.2)
> a21=FuzzyNumbers::TrapezoidalFuzzyNumber(3.1,3.3,3.5,3.7)
> a22=FuzzyNumbers::TrapezoidalFuzzyNumber(2.6,2.8,3,3.2)
> a31=FuzzyNumbers::TrapezoidalFuzzyNumber(0.6,0.8,1,1.2)
> a32=FuzzyNumbers::TrapezoidalFuzzyNumber(1.6,1.8,2,2.2)
> A=matrix(c(a11,a21,a31,a12,a22,a32),nrow = 3)
> dir=c("<=","<=","<=")
> b=c(FuzzyNumbers::TrapezoidalFuzzyNumber(19.6,19.8,20,20.2),FuzzyNumbers
::TrapezoidalFuzzyNumber(29.6,29.8,30,30.2),FuzzyNumbers::TrapezoidalFuzzy
Number(17.6,17.8,18,18.2))
> t=c(FuzzyNumbers::TrapezoidalFuzzyNumber(1,1,1,1),FuzzyNumbers::Trapezoi
dalFuzzyNumber(1,1,1,1),FuzzyNumbers::TrapezoidalFuzzyNumber(1,1,1,1))
```

```
> sol=FuzzyLP::GFLP(obj,A,dir,b,t,maximum = TRUE,ordf_obj = "Yager1",ordf_
res = "Yager3")
> sol
```

Press the enter key to obtain the following output.

```
      beta x1      x2
[1,] 0      1.012987 9.467532
[2,] 0.25  1.077922 9.305195
[3,] 0.5   1.142857 9.142857
[4,] 0.75  1.207792 8.980519
[5,] 1     1.272727 8.818182
```

```
sol[,"objective"]
```

```
[[1]]
Trapezoidal fuzzy number with:
  support=[36.7169,43.0052],
  core=[38.813,40.9091].
```

```
[[2]]
Trapezoidal fuzzy number with:
  support=[36.3013,42.5312],
  core=[38.3779,40.4545].
```

```
[[3]]
Trapezoidal fuzzy number with:
  support=[35.8857,42.0571],
  core=[37.9429,40].
```

```
[[4]]
Trapezoidal fuzzy number with:
  support=[35.4701,41.5831],
  core=[37.5078,39.5455].
```

```
[[5]]
Trapezoidal fuzzy number with:
  support=[35.0545,41.1091],
  core=[37.0727,39.0909].
```

Optimum solution

$z = (37.0727, 39.0909, 2.0812, 2.0812)$

For defuzzification, use the following Yager's ranking function

$$\mathfrak{R}(\widetilde{A}_T) = \frac{1}{2} \left( p_T^L + p_T^U - \frac{4}{5} p_1 + \frac{2}{3} p_2 \right)$$

Therefore,  $Z = 37.9431$ .

From the R program, the output is  $Z = 37.9469$  when  $q_1 = 1.272727$  and  $q_2 = 8.818182$ .

Similar optimal solutions have been obtained using Lingo software with the same quantities of system-level solution type 1 at 1.272727 and type 2 at 8.818182 in order to maximize the company's profit by 38%.



### Problem Statement 4.2

A social network under construction is planning to use three types of hardware, denoted as  $F_1$ ,  $F_2$ , and  $F_3$ . The cost of each unit varies between 3800 - 4200 euros for  $F_1$ , 1700 - 2300 euros for  $F_2$ , and 5500 - 6500 euros for  $F_3$ . It has been estimated that each unit of  $F_1$  has the capacity to satisfy the needs of about 150 - 250 male visitors and 400 - 600 female visitors per day. Each unit of  $F_2$  can accommodate about 320 - 480 male visitors and 60 - 140 female visitors, while each unit of  $F_3$  can cater to about 170 - 230 male visitors and 80 - 120 female visitors per day. The network is expected to have at least 2400 male and 800 female visitors per day. Determine the number of units of each type that should be ordered to minimize the cost of the hardware.

### Solution

In this problem, each unit cost varies. Types of hardware and network capacities are uncertain.

So, using trapezoidal fuzzy numbers, it can be formulated into a fully fuzzy linear programming problem as follows:

Let  $h_1$ ,  $h_2$  and  $h_3$  respectively be the units of the hardware  $F_1$ ,  $F_2$  and  $F_3$  that should be ordered.

Then

$$\text{Min } z = (36, 38, 40, 42) h_1 + (17, 19, 21, 23) h_2 + (56, 59, 62, 65) h_3$$

Subject to the constraints

$$(1.5, 1.8, 2.1, 2.4) h_1 + (3.2, 3.7, 4.2, 4.7) h_2 + (1.7, 1.9, 2.1, 2.3) h_3 \geq (23.0, 23.5, 24.0, 24.5)$$

$$(4.0, 4.7, 5.4, 6.1) h_1 + (0.6, 0.9, 1.2, 1.5) h_2 + (0.8, 1.0, 1.2, 1.4) h_3 \geq (7.0, 7.5, 8.0, 8.5)$$

$$h_1, h_2, h_3 \geq 0$$

To use LINGO software, first convert this problem into a crisp linear programming problem by using  $v_R$ - ranking function  $\vartheta_{\mathcal{R}}(\widetilde{A}_T) = \frac{p+2q+2r+s}{6}$ .

$$\text{Min } z = 39 h_1 + 20 h_2 + 60.5 h_3$$

Subject to the constraints

$$1.95 h_1 + 3.95 h_2 + 2 h_3 \geq 23.75$$

$$5.05 h_1 + 1.05 h_2 + 1.1 h_3 \geq 7.75$$

$$h_1, h_2, h_3 \geq 0$$

When using Lingo software, the output will be as follows:

Global optimal solution found.

Objective value:

129.4874

```

Infeasibilities:                                0.000000
Total solver iterations:                        2
Elapsed runtime seconds:                       0.04

Model Class:                                   LP

Total variables:                               3
Nonlinear variables:                           0
Integer variables:                             0

Total constraints:                             3
Nonlinear constraints:                         0

Total nonzeros:                               9
Nonlinear nonzeros:                           0

```

Variable	Value	Reduced Cost
H1	0.3170391	0.000000
H2	5.856145	0.000000
H3	0.000000	46.72039

Row	Slack or Surplus	Dual Price
1	129.4874	-1.000000
2	0.000000	-3.354749
3	0.000000	-6.427374

Thus, the optimum solution is  $h_1 = 0.3170391$ ,  $h_2 = 5.856145$ ,  $h_3 = 0$  in order to minimize the cost of hardware by 129.4874.

Next, this problem will be solved by using the R program.

The following R commands are to be used for finding the optimum solution.

```

> obj=c(FuzzyNumbers::TrapezoidalFuzzyNumber(36,38,40,42),FuzzyNumbers::TrapezoidalFuzzyNumber(17,19,21,23),FuzzyNumbers::TrapezoidalFuzzyNumber(56,59,62,65))
> a11=FuzzyNumbers::TrapezoidalFuzzyNumber(1.5,1.8,2.1,2.4)
> a12=FuzzyNumbers::TrapezoidalFuzzyNumber(3.2,3.7,4.2,4.7)
> a13=FuzzyNumbers::TrapezoidalFuzzyNumber(1.7,1.9,2.1,2.3)
> a21=FuzzyNumbers::TrapezoidalFuzzyNumber(4,4.7,5.4,6.1)
> a22=FuzzyNumbers::TrapezoidalFuzzyNumber(0.6,0.9,1.2,1.5)
> a23=FuzzyNumbers::TrapezoidalFuzzyNumber(0.8,1,1.2,1.4)
> A=matrix(c(a11,a21,a12,a22,a13,a23),nrow = 2)
> dir=c(">=", ">=")
> b=c(FuzzyNumbers::TrapezoidalFuzzyNumber(23,23.5,24,24.5),FuzzyNumbers::TrapezoidalFuzzyNumber(7,7.5,8,8.5))
> t=c(FuzzyNumbers::TrapezoidalFuzzyNumber(0,0,0,0),FuzzyNumbers::TrapezoidalFuzzyNumber(0,0,0,0))
> sol=FuzzyLP::GFLP(obj,A,dir,b,t,maximum = FALSE,ordf_obj = "Yager1",ordf_res = "Yager3")
> sol

```

Press the enter key to obtain the following output.

```

      beta x1      x2      x3
[1,] 0      0.1550279 5.682961 0
[2,] 0.25 0.1955307 5.726257 0
[3,] 0.5  0.2360335 5.769553 0
[4,] 0.75 0.2765363 5.812849 0
[5,] 1    0.3170391 5.856145 0
    
```

```

> sol[,"objective"]
[[1]]
Trapezoidal fuzzy number with:
  support=[102.191,137.219],
  core=[113.867,125.543].

[[2]]
Trapezoidal fuzzy number with:
  support=[104.385,139.916],
  core=[116.229,128.073].

[[3]]
Trapezoidal fuzzy number with:
  support=[106.58,142.613],
  core=[118.591,130.602].

[[4]]
Trapezoidal fuzzy number with:
  support=[108.774,145.31],
  core=[120.953,133.131].

[[5]]
Trapezoidal fuzzy number with:
  support=[110.968,148.007],
  core=[123.314,135.661].
    
```

Optimum solution is

$$z = (123.314, 135.661, 12.346, 12.346)$$

For defuzzification, use the following Yager's ranking function

$$\mathfrak{R}(\tilde{A}_T) = \frac{1}{2} \left( p_T^L + p_T^U - \frac{4}{5} p_1 + \frac{2}{3} p_2 \right)$$

$$\text{Hence, } Z = 128.6644$$

Thus, the hardware cost reduction of 128.6644 will occur when ordering  $F_1$  is 0.3170391,  $F_2$  is 5.856145 and  $F_3$  is 0.

## 5. DISCUSSION

In comparison, the R programming language can be directly used to solve fully fuzzy linear programming problems, while the Lingo software requires these problems to be converted into crisp linear programming problems before they can be solved. In both models (problems 4.1 and 4.2), it has been observed that the optimal solutions obtained using Lingo software and R programming are approximately similar.

## 6. CONCLUSION

In the initial sections of the paper, the discussion centered around the significance of fuzzy optimization concepts and their practical applications. This involved a detailed exploration of two distinct ranking functions, along with a comprehensive explanation of their respective properties. Furthermore, the paper delved into an in-depth analysis of the features associated with Lingo software and R programming.

Following this, the Lingo software was leveraged to identify the optimal solutions for sample models of fully fuzzy linear programming problems. This process involved the conversion of these models into crisp linear programming problems, culminating in the successful determination of optimal solutions. Similarly, the R programming language was employed to derive optimal solutions for the same models.

Subsequently, a comparative analysis was conducted to juxtapose the outcomes obtained through both methodologies, thereby providing a comprehensive assessment of their respective efficacy and applicability.

## CONFLICT OF INTEREST:

The authors have no conflicts of interest to declare that are relevant to the content of this article.

## REFERENCES

1. Ebrahimnejad and Verdegay, Fuzzy Logic in its 50<sup>th</sup> year, A Survey on Models and Methods for Solving Fuzzy linear programming problems, Studies in Fuzziness and Soft Computing, Vol.539, Springer-Verlag, 2014, pp.327-368 [DOI: [https://doi.org/10.1007/978-3-319-31093-0\\_15](https://doi.org/10.1007/978-3-319-31093-0_15)]
2. Ebrahimnejad and Verdegay, Fuzzy sets-Based methods and techniques for modern analytics, Studies in Fuzziness and Soft computing. Springer, 2018, Vol 364. [DOI: <https://doi.org/10.1007/978-3-319-73903-8>]
3. Shams Hesam, Mogouee Masouemeh Doosti, Jamali F, Haji A, A survey on fuzzy linear programming, Am J Sci Res, 2012, Vol. 75, 117-133. [URL: <http://www.eurojournals.com/ajsr.htm> ]
4. R. Ghanbari et al., Fuzzy linear programming problems: models and solutions, Soft Computing, 2020, 24: pp.10043 – 10073. [DOI: <https://doi.org/10.1007/s00500-019-04519-w> ]
5. Sotoudeh-Anvari A, A critical review on theoretical drawbacks and mathematical incorrect assumptions in fuzzy or methods: Review from 2010 to 2020, Applied Soft Computing, Vol.93, 106354, 2020. [DOI: <http://dx.doi.org/10.1016/j.asoc.2020.106354> ]
6. Mohamed Salih Mukthar. M and Ramathilagam. S, Modified operations of trapezoidal fuzzy numbers for solving fuzzy linear programming problems, Fuzzy Mathematical

Analysis and Advances in Computational Mathematics, Studies in Fuzziness and Soft Computing, Springer, 2022, Vol 419, pp.33-41.

[DOI: [http://dx.doi.org/10.1007/978-981-19-0471-4\\_2](http://dx.doi.org/10.1007/978-981-19-0471-4_2)]

7. Karyati,D.U. Wutsqa and N, Insani, Yager's ranking method for solving the trapezoidal fuzzy number linear programming, IOP Conference series: Journal of physics,2018, Vol 983.[DOI: [10.1088/1742-6596/983/1/012135](https://doi.org/10.1088/1742-6596/983/1/012135)]
8. Mary George and Savitha M.T, New Methods for Ranking of Trapezoidal fuzzy numbers, Advances in Fuzzy Mathematics, 2017; Vol.12, No.5, pp. 1159-1170.  
[URL: [https://www.ripublication.com/afm17/afmv12n5\\_08.pdf](https://www.ripublication.com/afm17/afmv12n5_08.pdf)]
9. Adrian.I Ban and Lucian Coroianu, Existence, Uniqueness and Continuity of trapezoidal approximations of fuzzy numbers under a General condition, Fuzzy Sets and Systems, Elsevier, 2013; pp. 1-20. [DOI: <https://doi.org/10.1016/j.fss.2013.07.004>]
10. <https://www.lindo.com/index.php>
11. <https://cran.r-project.org/web/packages/FuzzyLP/vignettes/FuzzyLP.pdf>
12. <https://github.com/olbapjose/FuzzyLP>