Advances in Fuzzy Sets and Systems

https://pphmjopenaccess.com
https://doi.org/10.17654/0973421X25002
Volume 30, Number 1, June 2025, Pages 11-25 **P-ISSN: 0973-421X**

CENTROID RANKING APPROACH TO SOLVE THE FUZZY ASSIGNMENT PROBLEM

Sudhir Kumar and Anita Kumari*

Department of Mathematics
Dr. Shyama Prasad Mukherjee University
Ranchi 834001, Jharkhand, India
e-mail: sudhir.kr.8826@gmail.com
mehtaanita007@gmail.com

Abstract

In this study, an optimal technique is developed to address fuzzy assignment problems utilizing the PuLP library in Python. The goal is to minimize or maximize the assignment cost within a fuzzy environment where all variables are expressed as triangular fuzzy numbers. To facilitate computation, these fuzzy values are defuzzified into crisp numbers using a ranking approach, specifically the centroid method. The complete procedure is thoroughly explained and demonstrated with a practical numerical example. This strategy

Received: June 2, 2025; Revised: June 24, 2025; Accepted: July 16, 2025
2020 Mathematics Subject Classification: 90B80, 90C70, 03E72.
Keywords and phrases: triangular fuzzy number, fuzzy assignment problem, ranking function, Python pulp.

*Corresponding author

How to cite this article: Sudhir Kumar and Anita Kumari, Centroid ranking approach to solve the fuzzy assignment problem, Advances in Fuzzy Sets and Systems 30(1) (2025), 11-25.
https://doi.org/10.17654/0973421X25002

Published Online: September 19, 2025

enables efficient one-to-one task allocation, increasing the likelihood of bidder participation while reducing both the total fuzzy assignment cost and completion time.

1. Introduction

The assignment problem is a well-known optimization problem that involves allocating various tasks (such as jobs, activities, or services) to an equal number of agents (such as workers, machines, or departments) in an efficient and coordinated manner. The primary objective is to minimize total cost or time while maximizing overall benefit, productivity, or satisfaction. Assignment models are widely utilized across disciplines including engineering, computer science, operations research, and social sciences. To effectively solve classical assignment problems, the model parameters are typically treated as precise or crisp values. However, in real-world applications, uncertainty and imprecision are unavoidable due to estimation errors, model simplifications, parameter fluctuations, computational inaccuracies, and other limitations. These uncertainties challenge the applicability of traditional assignment models, making fuzzy assignment approaches more relevant and realistic for complex, real-life scenarios. Project management, which involves planning and allocating limited organizational resources to specific tasks under time, cost, and quality constraints, also benefits from fuzzy assignment models. To optimize performance, it is crucial to allocate the best available resources to the most critical tasks, thereby minimizing costs and maximizing efficiency. Significant advancements have been made in the field of assignment problems. Lassalle and Bourgerois [1] extended the Hungarian algorithm to rectangular matrices, while Karmarkar [2] introduced a polynomial-time algorithm for linear programming. Dantzig [3] developed the simplex method, a foundational algorithm for solving linear programming problems, which has been commercially implemented for large-scale industrial applications. The Hungarian method (also referred to as the Kuhn-Munkres algorithm) remains a classical tool for solving assignment and transportation problems [4, 5]. With the development of fuzzy set theory by Zadeh [6],

ranking fuzzy numbers became an essential aspect of fuzzy decision-making. Jain [7] was among the first to propose methods for ranking fuzzy quantities to facilitate decisions under uncertainty. Since then, numerous ranking approaches have been introduced and reviewed, including works by Bortolan and Degani [8], Baldwin and Guild [9], Yager [10, 11], Adamo [12], and Dubois and Prade [13]. Mary and Selvi [14] proposed a model for solving fuzzy assignment problems using the centroid ranking method. In this paper, a novel approach is introduced to solve fuzzy assignment problems. Initially, the fuzzy parameters are converted into equivalent crisp values through a ranking function, after which the problem is formulated as a linear or integer linear programming model. This model is then solved using the Python PuLP library. Python PuLP proves particularly useful for handling complex problems involving numerous decision variables, and it supports both maximization and minimization objectives.

2. Preliminaries

Before studying the proposed algorithm and model, some key terms are defined which are as follows:

2.1. Fuzzy set

Let X be a universe of discourse and x an element of X. Then the fuzzy set A defined on X is a collection of ordered pairs, $A = ((x, u_A(x)) : x \in X)$, where $U_A(x) : X \to [0, 1]$ is called the *membership function*.

2.2. Fuzzy number

To qualify as fuzzy number, a fuzzy set A on R must pass the following three properties:

(1) A must be a normal fuzzy set.

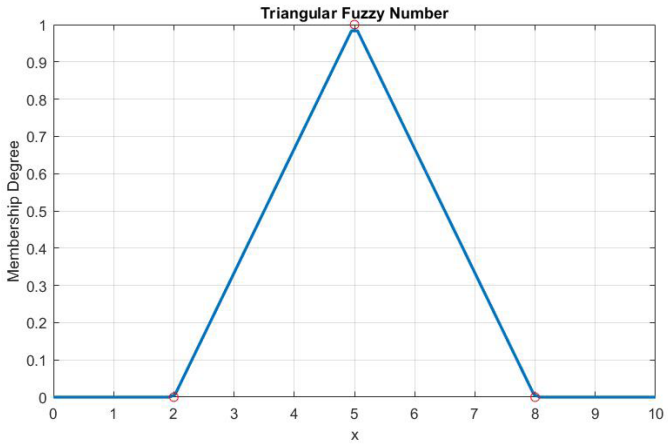
(2) A must be a convex fuzzy set.

(3) The membership function of A must be piecewise continuous.

2.3. Triangular fuzzy number

A triangular fuzzy number $P = (a, b, c)$ is specified by membership function:

$$\mu_A(x) = \begin{cases} 0; & \text{for } x \le a, \\ \dfrac{x-a}{b-a}; & \text{for } a \le x \le b, \\ \dfrac{x-c}{b-c}; & \text{for } b \le x \le c, \\ 0; & \text{for } x > c. \end{cases}$$



Graph of trian1gular fuzzy number

2.4. Fuzzy assignment problem

The mathematical model of $n \times n$ balanced AP is as follows:

(Model1) Minimize $Z = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij}$

or Maximize $Z = \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} X_{ij}$

subject to $\sum_{j=1}^{n} X_{ij} = 1$, for $i = 1, 2, ..., n$ (row restriction)

$$\sum_{i=1}^{n} X_{ij} = 1, \text{ for } j = 1, 2, ..., n \text{ (column restriction)}$$

$$x_{ij}, 0 \text{ or } 1 \text{ for all } i, j,$$

where c_{ij} is the cost of assigning the job j to the machine i. $x_{ij} = 1$ if the job j is assigned to the machine i and $x_{ij} = 0,$ otherwise.

2.5. Centroid of a generalized triangular

The centroid of a triangle fuzzy number $A = (a, b, c; w)$ where a, b and c are the lower, modal, and upper values, respectively, and w denotes the height of the fuzzy number, is given by

$$T_A = \left(\frac{a + b + c}{3}, \frac{w}{3} \right).$$

This centroid represents the defuzzified crisp value and height component of the fuzzy number. The ranking function, which transforms a generalized triangular fuzzy number into a real number for comparative purposes, is defined as

$$R(A) = \left(\frac{a + b + c}{3} \right) \cdot \left(\frac{w}{3} \right). \tag{1}$$

This function is commonly used to rank fuzzy numbers by projecting them into the real number domain, preserving both the spread and the membership strength of the fuzzy quantity [15].

2.6. Arithmetic operation

Let $f_A = (a_1, a_2, a_3, a_4, ..., a_n; w_1)$ and $f_B = (b_1, b_2, b_3, b_4, ..., b_n; w_2)$ be two fuzzy numbers, where $a_1 < a_2 < a_3 < a_4 < \cdots < a_n$ and $b_1 < b_2 < b_3 < b_4 < \cdots < b_n$. Then the arithmetic operation is defined as follows:

(1) $f_A + f_B = (a_1 + b_1, a_2 + b_2, a_3 + b_3, ..., a_n + b_n; \min(w_1, w_2))$,

(2)

$$f_A - f_B = (a_1 - b_n,\ a_2 - b_{(n-1)},\ a_3 - b_{(n-2)},\ ...,\ a_n - b_1;\ \min(w_1,\ w_2)),$$

(3) $f_A * f_B = (a_1 * b_1,\ a_2 * b_2,\ a_3 * b_3,\ ...,\ a_n * b_n;\ \min(w_1,\ w_2))$,

(4) $$KfA = \begin{cases} ka_1,\ ka_2,\ ka_3,\ ...,\ ka_n; & \text{if } k > 0, \\ ka_n,\ ka_{(n-1)},\ ka_{(n-2)},\ ...,\ ka_1; & \text{if } k < 0. \end{cases}$$

3. Algorithm

Assignment problems are a specific class of optimization problems that deal with the allocation of various tasks (such as jobs, services, or responsibilities) to an equal number of agents (such as workers, machines, or departments) in a systematic and efficient way. The core objective is to minimize the total cost or time, or to maximize the overall efficiency, benefit, or satisfaction associated with the allocation. These problems find significant application in diverse fields such as computer science, operations research, industrial engineering, and project management.

In conventional assignment problems, all model parameters are considered crisp and deterministic. However, real-world situations are often accompanied by uncertainty and imprecision due to factors such as estimation errors, environmental variability, system complexity, and human judgment. These limitations restrict the effectiveness of classical optimization techniques, making fuzzy optimization a more appropriate approach for realistic scenarios.

Fuzzy assignment problems incorporate fuzzy numbers - commonly represented as triangular fuzzy numbers - to handle the vagueness present in input data. In order to solve such problems using mathematical programming techniques, fuzzy values must first be transformed into crisp equivalents. Among various defuzzification strategies, the centroid ranking method is widely used due to its computational simplicity and effectiveness.

To address the complexity of fuzzy assignment problems, a five-step methodology is proposed in this study:

Step 1. The fuzzy numbers involved in the assignment problem are first transformed into crisp values using the centroid ranking method.

Step 2. These crisp values are then used to formulate an equivalent linear programming model.

Step 3. The resulting crisp optimization model is solved using the Python PuLP library. This step provides both the optimal solution and the corresponding objective value of the crisp model.

Step 4. Based on the results obtained in Step 3, the minimum or maximum objective value for the original fuzzy optimization problem can be calculated using equation (1).

Step 5. The final step yields the optimal assignment and allocation for the given problem.

Note. (1) A crisp optimization problem is defined as a model with all parameters represented by exact numerical values.

(2) Python PuLP is capable of solving a wide range of optimization problems, whether linear or integer-based.

The proposed method is now demonstrated through the following real-world numerical examples to highlight its applicability and effectiveness.

4. Numerical Examples

Each proposed model in this section includes concrete numerical examples drawn from real-world data.

4.1. Example 1

We are going to solve a fuzzy assignment problem that allocates four jobs to four machines, using the fuzzy assignment cost matrix C_{ij}.

Row represents four jobs $J_1,\ J_2,\ J_3,\ J_4$ and column four machines $M_1,\ M_2,\ M_3,\ M_4$. The cost matrix $[c_{ij}]4 \times 4$ is given whose elements are triangular fuzzy numbers (TFNs). The aim of the problem is to find the optimal assignment so that the total cost of job assignment becomes maximum.

Solution. The given problem possesses both the crisp and fuzzy parameters. Here, the cost of the given assignment problem is TFN and the value of the decision variable is the crisp number.

Therefore, the given problem is a fuzzy assignment problem.

Table 1. Triangular assignment model

	M_1	M_2	M_3	M_4
J_1	(10, 13, 16)	(7, 10, 13)	(8, 11, 14)	(7, 10, 13)
J_2	(9, 12, 15)	(10, 13, 16)	(8, 11, 14)	(9, 12, 15)
J_3	(5, 8, 11)	(7, 10, 13)	(7, 10, 13)	(5, 8, 11)
J_4	(8, 11, 14)	(6, 9, 12)	(8, 11, 14)	(5, 8, 11)

From Step 1, we get the following crisp assignment problem (CAP):

Table 2. Assignment model

	M_1	M_2	M_3	M_4
J_1	4.33	3.33	3.67	3.33
J_2	4.00	4.33	3.67	4.00
J_3	2.67	3.33	3.33	2.67
J_4	3.67	3.00	3.67	2.67

From Step 2, we get the following linear programming problem:

$$\begin{aligned}\text{Minimize } Z = {} & 4.33X_{11} + 3.33X_{12} + 3.67X_{13} + 3.33X_{14} + 4.00X_{21} \\ & + 4.33X_{22} + 3.67X_{23} + 4.00X_{24} + 2.67X_{31} + 3.33X_{32} \\ & + 3.33X_{33} + 2.67X_{34} + 3.67X_{41} \\ & + 3.00X_{42} + 3.67X_{43} + 2.67X_{44}\end{aligned}$$

subject to the conditions

$X_{11} + X_{12} + X_{13} + X_{14} = 1$ (row 1 restriction),

$X_{21} + X_{22} + X_{23} + X_{24} = 1$ (row 2 restriction),

$X_{31} + X_{32} + X_{33} + X_{34} = 1$ (row 3 restriction),

$X_{41} + X_{42} + X_{43} + X_{44} = 1$ (row 4 restriction),

$X_{11} + X_{21} + X_{31} + X_{41} = 1$ (column 1 restriction),

$X_{12} + X_{22} + X_{32} + X_{42} = 1$ (column 2 restriction),

$X_{13} + X_{23} + X_{33} + X_{43} = 1$ (column 3 restriction),

$X_{14} + X_{24} + X_{34} + X_{44} = 1$ (column 4 restriction),

$X_{11}, X_{12}, X_{13}, X_{14}, X_{21}, X_{22}, X_{23}, X_{24}, X_{31}, X_{32},$

$X_{33}, X_{34}, X_{41}, X_{42}, X_{43}, X_{44} \geq 0$ (non-negative restrictions).

From Step 3 by applying Python PuLP to this problem, we get the following optimal solution which is given in screenshots 1, 2 and 3 (i.e., Figures 1 to 3).

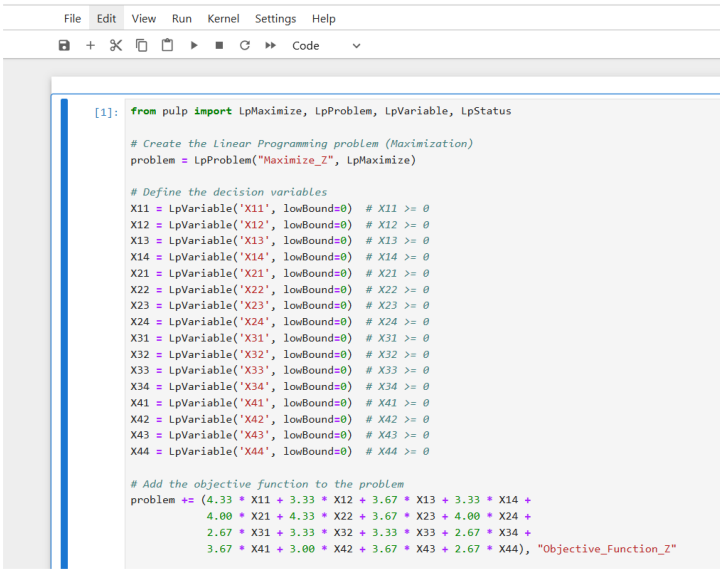


Figure 1. Out summary corresponding to the FAP and its crisp AP.

```
# Add the constraints
problem += (X11 + X12 + X13 + X14 == 1), "Constraint_1"
problem += (X21 + X22 + X23 + X24 == 1), "Constraint_2"
problem += (X31 + X32 + X33 + X34 == 1), "Constraint_3"
problem += (X41 + X42 + X43 + X44 == 1), "Constraint_4"
problem += (X11 + X21 + X31 + X41 == 1), "Constraint_5"
problem += (X12 + X22 + X32 + X42 == 1), "Constraint_6"
problem += (X13 + X23 + X33 + X43 == 1), "Constraint_7"
problem += (X14 + X24 + X34 + X44 == 1), "Constraint_8"

# Solve the problem
problem.solve()

# Print the status of the solution
print("Status:", LpStatus[problem.status])

# Calculate the final total cost (objective function value)
final_total_cost = (4.33 * X11.varValue + 3.33 * X12.varValue + 3.67 * X13.varValue + 3.33 * X14.varValue +
                    4.00 * X21.varValue + 4.33 * X22.varValue + 3.67 * X23.varValue + 4.00 * X24.varValue +
                    2.67 * X31.varValue + 3.33 * X32.varValue + 3.33 * X33.varValue + 2.67 * X34.varValue +
                    3.67 * X41.varValue + 3.00 * X42.varValue + 3.67 * X43.varValue + 2.67 * X44.varValue)

print("Optimal Solution:")
for v in problem.variables():
    print(f"{v.name} = {v.varValue}")

# Print the final total cost
```

Figure 2. Out summary corresponding to the FAP and its crisp AP.

```
# Print the final total cost
print("Final Total Cost (Z) =", final_total_cost)
```

```
Status: Optimal
Optimal Solution:
X11 = 1.0
X12 = 0.0
X13 = 0.0
X14 = 0.0
X21 = 0.0
X22 = 0.0
X23 = 0.0
X24 = 1.0
X31 = 0.0
X32 = 1.0
X33 = 0.0
X34 = 0.0
X41 = 0.0
X42 = 0.0
X43 = 1.0
X44 = 0.0
Final Total Cost (Z) = 15.33
```

```
[ ]:
```

Figure 3. Out summary corresponding to the FAP and its crisp AP.

The output image shows the optimal solution and optimal objective value of the crisp assignment problem corresponding to fuzzy assignment problem.

From Step 3, we get the following optimal solution and optimal objective value for the crisp assignment problem:

$$X_{11} = 1, \quad X_{24} = 1, \quad X_{32} = 1, \quad X_{43} = 1,$$

$$X_{12} = X_{13} = X_{14} = X_{21} = X_{22} = X_{23} = X_{31} = X_{33} = X_{34} = X_{41} = X_{42} = X_{44} = 0.$$

Therefore, the optimal assignment to the given real life problem is as follows:

$$M_1 \to J_1,\ \ M_2 \to J_4,\ \ M_3 \to J_2,\ \ M_4 \to J_3.$$

The optimal solution to given FAP and its equivalent CAP both are the same. Because, the values of decision variables in each of the problems are crisp numbers:

$$\text{Min } Z = (4.33) \times 1 + (4.00) \times 1 + (3.33) \times 1 + (3.67) \times 1$$

$$= 15.33 \text{ (optimal assignment cost).}$$

From Step 4, we can write the optimal objective value for the given FAP is as follows:

$$\text{Min } Z = (10, 13, 16) \times 1 + (9, 12, 15) \times 1 + (7, 10, 13) \times 1 + (8, 11, 14) \times 1$$

$$= (26, 35, 44).$$

5. Results and Discussion

To demonstrate the effectiveness of the proposed approach, a real-life fuzzy assignment problem involving four jobs and four machines was solved. The assignment cost data was represented using triangular fuzzy numbers, which account for the uncertainty and vagueness in real-world cost estimation. In the first step, each triangular fuzzy number was converted into a crisp value using the centroid ranking method, producing a crisp cost matrix. This matrix served as input for formulating a linear programming model under the standard assignment problem structure, with constraints ensuring one-to-one allocations and non-negativity conditions. The linear model was then solved using the Python PuLP optimization library, which effectively handles large-scale linear and integer programming problems. The solution yielded the following optimal assignments:

$$M_1 \to J_1,\ \ M_2 \to J_4,\ \ M_3 \to J_2,\ \ M_4 \to J_3.$$

These results confirm a valid one-to-one assignment where each job is uniquely allocated to a machine, and all decision variables take binary crisp values, verifying that the problem was correctly transformed and solved.

The minimum crisp total assignment cost obtained is:

$$Z = 4.33 + 4.00 + 3.33 + 3.67 = 15.33.$$

This is the precise numerical result after converting fuzzy costs into crisp values.

In addition, the fuzzy total cost of the assignment was derived from the original triangular fuzzy values associated with the selected allocations:

$$Z = (10, 13, 16) + (9, 12, 15) + (7, 10, 13) + (8, 11, 14) = (26, 35, 44).$$

Here, the triplet represents the fuzzy range - 26 being the optimistic (minimum) value, 35 the most likely (centroid), and 44 the pessimistic (maximum) estimate of the total cost. The consistency between the crisp solution and the fuzzy formulation demonstrates the accuracy of the proposed approach. Moreover, the fuzzy result provides a valuable cost range, equipping decision-makers with insights under uncertain conditions. This example illustrates that the method not only delivers optimal assignments but also enhances decision-making through efficient computation and robust uncertainty handling. It confirms the capability of the algorithm to address real-world fuzzy optimization problems and support practical applications where exact values may not always be available.

5.1. Comparative analysis of assignment methods

To evaluate the performance and effectiveness of the proposed method, a comparative analysis has been conducted between the fuzzy assignment model solved using Python PuLP and several well-established optimization approaches. These include the Hungarian method, Simplex method, and existing fuzzy assignment models that use ranking functions and heuristic algorithms.

Table 3. Comparative analysis of assignment problem solving methods

Method	Handling of fuzziness	Computational complexity	Accuracy	Scalability	Automation
Hungarian method	Not applicable (crisp only)	Low	High (for crisp cases)	Limited	Manual or tool-based
Simplex method	Not applicable (crisp only)	Medium to high	Very high	High	Solver required
Fuzzy heuristics (e.g., GA, ACO)	Yes (indirectly via fitness)	High	Medium to high	Medium	Needs algorithm tuning
Proposed method (Centroid + Python PuLP)	Yes (direct via TFNs)	Moderate	High	High	Fully automatable

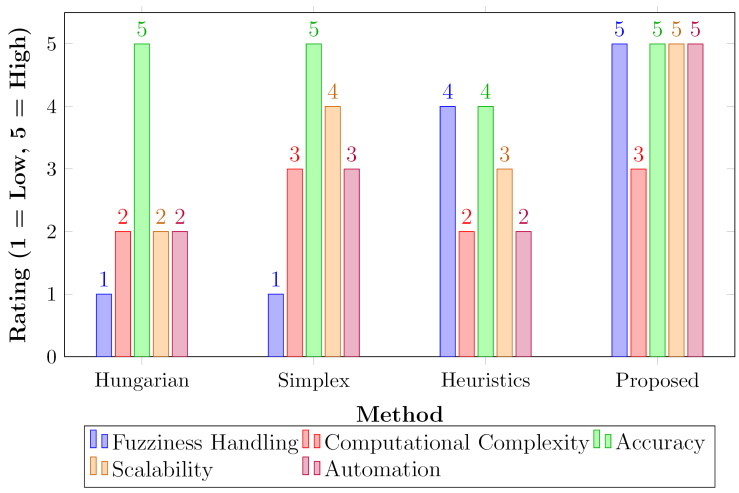


Figure 4. Graphical comparison of assignment methods based on key criteria.

6. Conclusions

In this study, the proposed method facilitates effective one-to-one allocation of individuals to tasks, aiming to minimize or maximize the total fuzzy assignment cost and overall fuzzy time within an organizational setting. The approach proves particularly beneficial in handling uncertainty, allowing decision-makers and project managers to estimate future outcomes with greater confidence. Additionally, the algorithm enhances computational

efficiency, thereby reducing the time and resources required to reach optimal solutions. The core of the method lies in transforming triangular fuzzy numbers - used to model uncertain parameters - into crisp values using the centroid ranking method. These crisp values form the basis of a linear or integer linear programming model, which is then solved using the Python PuLP optimization library. This not only provides a practical solution for fuzzy assignment problems but also ensures scalability and adaptability for large, real-world applications. The proposed algorithm not only supports accurate resource planning and allocation under uncertain conditions but also contributes to improved operational decision-making. The results obtained demonstrate the algorithm's effectiveness in minimizing the total fuzzy assignment cost, optimizing time utilization, and enhancing decision quality in uncertain environments.

Acknowledgement

The authors thank the anonymous referees for their valuable suggestions and comments which led to the improvement of the paper.

References

[1] J. Lassalle and F. Bourgerois, An extension of the Munkres algorithm for the assignment problem to rectangular matrices, Commun. ACM 14(12) (1971), 802-804.

[2] N. Karmarkar, A new polynomial-time algorithm for linear programming, Combinatorica 4(4) (1984), 373-395.

[3] George B. Dantzig, Linear Programming and Extensions, Princeton University Press, 1963, pp. 45-67.

[4] J. Munkres, Algorithm for the assignment transportation problem, Soc. Ind. Appl. Math. 5(1) (1957), 32-38.

[5] H. W. Kuhn, The Hungarian method for the assignment problem, Naval Res. Logist. Quart. 2 (1955), 83-97.

[6] L. A. Zadeh, Fuzzy sets, Information and Control 8 (1965), 338-353.

[7] R. Jain, Decision making in the presence of fuzzy variable, IEEE 6 (1976), 698-703.

[8] G. Bortolan and R. Degani, A review of some methods for ranking fuzzy subsets, Fuzzy Sets and Systems 15 (1985), 1-19.

[9] N. C. F. |Baldwin and J. F. Guild, Comparison of fuzzy sets on the same decision space, Fuzzy Sets and Systems 2(3) (1979), 213-231.

[10] R. R. Yager, On choosing between fuzzy subsets, Kybernetes 9 (1980), 151-154.

[11] R. R. Yagar, A procedure for ordering fuzzy subsets of the unit interval, Inform. Sci. 24 (1981), 143-161.

[12] J. M. Adamo, Fuzzy decision trees, Fuzzy Sets and Systems, Fuzzy Math. 4 (1980), 207-219.

[13] D. Dubois and H. Prade, Ranking fuzzy numbers in the setting of possibility theory, Inform. Sci. 30 (1983), 183-224.

[14] R. Q. Mary and D. Selvi, Solving fuzzy assignment using centroid ranking method, International Journal of Mathematics and its Applications 6(3) (2018), 9-16.

[15] Y. L. P. Thorani and N. Ravi Shanker, Fuzzy assignment problem with generalised fuzzy numbers, Appl. Math. Sci. 7(71) (2013), 3511-3537.