One’s Fixing Method for a Distinct Symmetric Fuzzy Assignment Model

S. V. Gomathi and M. Jayalakshmi * 

Abstract: This paper hinges upon the subject of an \((n \times n)\) assignment problem and the distinct symmetric fuzzy assignment problem by assigning \(n\) machines to \(n\) jobs. One’s orientation algorithm is developed for solving the assignment problems based on the position of one’s chosen in every row as well as every column to perform allocations and obtain the assignment cost at every \((n - 1)\) reduced matrix. We also extended the two different symmetric concept to the problem to find the optimum solution based on symmetrical data and also used the ranking concept in our fuzzy assignment problem. In this proposed algorithm, the one’s position is associated with the successor of one in each iteration to obtain the optimal schedule and assignment cost. The comparative analysis is properly considered and discussed. The proposed technique is elaborated with the help of numerical computations and it gives clarity to the idea of this concept.

Keywords: symmetric triangular fuzzy number; symmetric trapezoidal fuzzy number; fuzzy assignment problem

1. Introduction

The assignment problem (AP) is commonly employed in production and service systems, as well as in a variety of other indirect applications. The assignment problem can be defined as a set of \(n\) jobs that must be completed by \(n\) workers, with costs varying according to the specific tasks. Each job must be given to one and only one worker, and each worker must complete one and only one task at a time. Then, to minimize the sum, an assignment problem is developed to choose precisely one element from each row and column. Costs, on the other hand, are not deterministic numbers in many real-world situations. Many researchers have begun to examine the assignment problem and its variants in a fuzzy environment in recent years. Ref. [1] proposed a ranking method for two types of symmetric interval-valued fuzzy numbers and obtained the optimal assignment cost. Ref. [2] proposed a ranking method and a zero suffix method to obtain an optimal solution for the two-stage fuzzy transportation problem. Ref. [3] developed a strategy for making decisions in a fuzzy environment, in which the goals or constraints define classes of possibilities and the boundaries are not well defined. Ref. [4] transformed a fuzzy assignment problem into a multi-objective linear programming problem and obtained the solution that is independent of the weights. Ref. [5] established a modified Hungarian method-based algorithm for identifying the best fuzzy AP solution presented. For a fuzzy AP, this approach is used to determine the lowest assignment cost possible. In the beginning, fuzzy numbers were used directly. Then, with the use of the cut process, the fuzzy AP was converted into an interval AP. Ref. [6] developed multi-criteria decision-making tool based on the characterization object method for solving the inter-valued triangular fuzzy number. Ref. [7] developed a ranking method to transform fuzzy assignment to crisp, solved byLINGO9.0. Ref. [8] developed an algorithm to solve multi-interval-valued fuzzy soft sets to analyze a decision problem. Ref. [9] presented

In this paper, considering the assignment problem by the proposed method, we locate the one’s position in each row as well as each column, and then the given \((n \times n)\) matrix is reduced in each and every iteration. In every iteration, one row will be assigned to one column so that we obtain the optimal schedule and assignment cost of the given problem. We applied the algorithm to symmetric interval-valued triangular and trapezoidal fuzzy assignment problems and achieved the optimal schedule and assignment cost. The proposed algorithm is not based on the Hungarian method. Illustrated examples are given for a better understanding of the solution procedure.

2. Some Preliminaries

**Definition 1.** A triangular fuzzy number is denoted by \(\tilde{A}=(r_1, r_2, r_3)\) such that \(r_1 \leq r_2 \leq r_3\) with membership functions \(\mu_{\tilde{A}}: X \to [0,1]\) (see Figure 1) is defined by

\[
\mu_{\tilde{A}}(x) = \begin{cases} 
\frac{x - r_1}{r_2 - r_1}, & x \in [r_1, r_2] \\
\frac{r_3 - x}{r_3 - r_2}, & x \in [r_2, r_3] \\
0, & \text{otherwise}
\end{cases}
\]
Figure 1. Membership function of symmetric and non-symmetric triangular fuzzy numbers.

If \( r_3 - r_2 = r_2 - r_1 \), then it is a symmetric triangular fuzzy number; otherwise, we refer to it as an asymmetric triangular fuzzy number.

**Definition 2.** A trapezoidal fuzzy number denoted by \( \tilde{A} = (p_1, p_2, p_3, p_4) \) such that \( p_1 \leq p_2 \leq p_3 \leq p_4 \) with membership functions \( \mu_A : X \rightarrow [0,1] \) (see Figure 2) is defined by

\[
\mu_A(x) = \begin{cases} 
\frac{x - p_1}{p_2 - p_1}, & p_1 < x < p_2 \\
1, & p_2 < x < p_3 \\
\frac{p_4 - x}{p_4 - p_3}, & p_3 < x < p_4 \\
0, & \text{Otherwise}
\end{cases}
\]

Figure 2. Membership function of Trapezoidal Fuzzy Number \( \tilde{A} \).

If \( p_1 - p_3 = p_2 - p_4 \), then it is a symmetric trapezoidal fuzzy number; otherwise, we refer to it as an asymmetric trapezoidal fuzzy number.

**Definition 3.** [1] A triangular interval-valued fuzzy number is denoted by \( A = \{(p_1, p_2, p_3), (q_1, q_2, q_3), w_A^l, w_A^u\} \), \( 0 \leq w_A^l \leq w_A^u \leq 1 \); then its membership functions are as follows:
A trapezoidal interval-valued fuzzy number is defined by

\[ A = \left\{ (l_1, l_2, l_3, l_4), (l_1', l_2', l_3', l_4'); w_A^l, w_A^u \right\} \]

The membership function is given by

\[
\mu_A^l (x) = \begin{cases} 
\frac{w_A^l(x-l_1)}{l_2-l_1}, & x \in [l_1, l_2] \\
\frac{w_A^l}{l_2-l_1}, & l_2 < x < l_3 \\
\frac{w_A^l(x-l_4)}{l_4-l_3}, & x \in [l_3, l_4] \\
0, & \text{otherwise}
\end{cases}
\]

\[
\mu_A^u (x) = \begin{cases} 
\frac{w_A^u(x-l_1')}{l_2'-l_1'}, & x \in [l_1', l_2'] \\
\frac{w_A^u}{l_2'-l_1'}, & l_2' < x < l_3' \\
\frac{w_A^u(x-l_4')}{l_4'-l_3'}, & x \in [l_3', l_4'] \\
0, & \text{otherwise}
\end{cases}
\]

Definition 4 [1]: A trapezoidal interval-valued fuzzy number is defined by

\[ A = \left\{ (l_1, l_2, l_3, l_4), (l_1', l_2', l_3', l_4'); w_A^l, w_A^u \right\} \]

3. Ranking Method for Symmetric Triangular Interval-Valued Fuzzy Number (STIFN) [1]:

(i). Let \( A = \left\{ (p_1, p_2, p_3), (q_1, q_2, q_3); w_A^l, w_A^u \right\} \) be a symmetric triangular interval-valued fuzzy number, then define \( C(\mu_A^l) = \left( \frac{p_1 + p_2 + p_3 \cdot w_A^l}{3} \right) \) and

\[
C(\mu_A^u) = \left( \frac{q_1 + q_2 + q_3 \cdot w_A^u}{3} \right); \text{then,}
\]

\[
R(A) = \left( \frac{w_A^l \cdot C(\mu_A^l) + w_A^u \cdot C(\mu_A^u)}{w_A^l + w_A^u} \right). \text{Here 'm' is the multiplicative operator.}
\]

(ii). Let \( A = \left\{ (l_1, l_2, l_1, l_4), (l_1', l_2', l_3', l_4'); w_A^l, w_A^u \right\} \) be a symmetric trapezoidal interval-valued fuzzy number, and then define
\[ C(\mu_1') = \left(\frac{2l_1 + 7l_2 + 7l_3 + 2l_4 * 7w_4'}{18}\right) \] and \[ C(\mu_2') = \left(\frac{2l_1' + 7l_2' + 7l_3' + 2l_4' * 7w_4''}{18}\right); \] then
\[ R(A) = \left(\frac{w_4'C(\mu_1') + w_4''C(\mu_2')}{w_4' + w_4''}\right) \] Here ‘*’ is the multiplicative operator.

\[ A = \{(d_1, \beta, \gamma), (d_1', \beta', \gamma'); w_4', w_4''\} \] and \[ B = \{(e_1, \gamma, \gamma'), (e_1', \gamma', \gamma'); w_4', w_4''\} \] are two symmetric interval-valued triangular fuzzy numbers; below are some arithmetic operations on A and B.
1. Addition
\[ A + B = \{(d_1 + e_1, \beta + \gamma, \beta + \gamma'), (d_1' + e_1', \beta' + \gamma', \beta' + \gamma'); u, v\} \] where \( u = \min(w_4', w_4''), \) \( v = \max(w_4'', w_4''') \)

2. Subtraction
\[ A - B = \{(d_1 - e_1, \beta - \gamma, \beta - \gamma'), (d_1' - e_1', \beta' - \gamma', \beta' - \gamma'); u, v\} \] where \( u = \min(w_4', w_4''), \) \( v = \max(w_4'', w_4''') \)

3. Scalar multiplication
\[ M\{(a_1, \alpha, \alpha), (a_1', \alpha', \alpha'); w_4', w_4''\} = \left\{(ma_1, m\alpha, m\alpha), (ma_1', m\alpha', m\alpha'); u, v \right\} \] \( \text{if} \ m > 0 \)
\[ M\{(a_1, -m\alpha, -m\alpha), (a_1', -m\alpha', -m\alpha'); u, v \} \] \( \text{if} \ m < 0 \)
where \( u = \min(w_4', w_4''), \) \( v = \max(w_4'', w_4''') \)

Let \[ A = \{(a_1, a_2, \alpha, \alpha), (a_1', a_2', \alpha', \alpha'); w_4', w_4''\} \] and \[ B = \{(b_1, b_2, \beta, \beta), (b_1', b_2', \beta', \beta'); w_4', w_4''\} \] be two symmetric interval-valued trapezoidal fuzzy numbers; the arithmetic operations performed on A and B are given below:
1. Addition
\[ A + B = \{(a_1 + b_1, a_2 + b_2, \alpha + \beta, \alpha + \beta), (a_1' + b_1', a_2' + b_2', \alpha' + \beta', \alpha' + \beta'); u, v\} \] where \( u = \min(w_4', w_4''), \) \( v = \max(w_4'', w_4''') \)

2. Subtraction
\[ A - B = \{(a_1 - b_2, a_2 - b_1, \alpha + \beta, \alpha + \beta), (a_1' - b_1', a_2' - b_1', \alpha' + \beta', \alpha' + \beta'); u, v\} \] where \( u = \min(w_4', w_4''), \) \( v = \max(w_4'', w_4''') \)

3. Scalar multiplication
\[ K\{(a_1, a_2, \alpha, \alpha), (a_1', a_2', \alpha', \alpha'); w_4', w_4''\} = \left\{(ka_1, ka_2, k\alpha, k\alpha), (ka_1', ka_2', k\alpha', k\alpha'); u, v \right\} \] \( \text{if} \ k > 0 \)
\[ K\{(ka_2, ka_1, -k\alpha, -k\alpha), (ka_2', ka_1', -k\alpha', -k\alpha'); u, v \} \] \( \text{if} \ k < 0 \)
where \( u = \min(w_4', w_4''), \) \( v = \max(w_4'', w_4''') \)
6. Assignment Problem

The cost of assigning the $i^{th}$ machine to the $j^{th}$ job can be expressed as an $n \times n$ matrix termed a cost matrix, where $c_{ij}$ is the cost of assigning the $i^{th}$ machine to the $j^{th}$ job. Consider a case in which you must assign $n$ machines to $n$ jobs. (one machine to one job). Let $c_{ij}$ represent the unit cost of assigning the $i^{th}$ machine to the $j^{th}$ job.

Let $x_{ij} = \begin{cases} 1, & \text{if } i^{th} \text{ machine with the } j^{th} \text{ job} \\ 0, & \text{otherwise.} \end{cases}$

<table>
<thead>
<tr>
<th>Machine1</th>
<th>Job1</th>
<th>C_{11}</th>
<th>C_{12}</th>
<th>........</th>
<th>........</th>
<th>C_{1n}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine2</td>
<td>C_{21}</td>
<td>C_{22}</td>
<td>........</td>
<td>........</td>
<td>........</td>
<td>C_{2n}</td>
</tr>
<tr>
<td>.........</td>
<td>........</td>
<td>........</td>
<td>........</td>
<td>........</td>
<td>........</td>
<td>.........</td>
</tr>
<tr>
<td>Machine$n$</td>
<td>C_{n1}</td>
<td>C_{n2}</td>
<td>........</td>
<td>........</td>
<td>........</td>
<td>C_{nn}</td>
</tr>
</tbody>
</table>

**Formulation of the fuzzy assignment problem:**

**The objective function is**

$$\text{Min } Z = \sum_{i=1}^{n} \sum_{j=1}^{n} \tilde{c}_{ij} x_{ij}$$

Subject to

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, 2, \ldots, n$$

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1, 2, \ldots, n$$

where $x_{ij} = \begin{cases} 1, & \text{if } i^{th} \text{ machine is assigned with the } j^{th} \text{ job} \\ 0, & \text{otherwise.} \end{cases}$ and $\tilde{c}_{ij}$ is the symmetric interval-valued fuzzy number.

Now, we introduce a new algorithm, namely, one’s orientation algorithm for finding an optimal solution for the assignment problem with and without fuzziness.

**One’s Orientation algorithm:**

**Step 1:** Initially, we consider an assignment problem either with fuzzy parameters or without fuzzy parameters.

**Step 2:**

(i) If the assignment problem is formulated without fuzzy parameters, then go to Step (4).

(ii) If the assignment problem formulated has only the fuzzy parameters, then go to Step (3).

**Step 3:** Convert the symmetric fuzzy assignment problem into crisp values using the ranking method.

**Step 4:** Check to see if the assignment problem is balanced.

(i) If it is balanced, go to Step 6.

(ii) If it is not balanced, go to Step 5.

**Step 5:** Add a dummy row (or) column with a cost value of zero to make the fuzzy assignment problem a balanced one.
Step 6: Divide each element of the $i$th row of the matrix by the minimum element of each row, which should be written on the right hand side of the matrix. Go to Step 8 if you get a result with ones in each row and column. If not, proceed to Step 7.

Step 7: Determine the smallest element of each column and write it below the $j$th column of the matrix; then divide each element of the $j$th column of the matrix, obtaining the assignment cost as one in each column.

Step 8: Identify the one’s position in each $(i,j)$th row and column.

Step 9: Find the value of each row’s succession of ones (after 1), select the maximum value, and delete the corresponding row and column of the matrix and perform allocations.

Step 10: In the reduced matrix, repeat step 8 until all of the rows are assigned and the optimal allocation for assignment cost is obtained.

We exposed the architecture of the proposed algorithm in Figure 3.

![Figure 3. Architecture of the proposed algorithm.](image)

Now, the proposed algorithm can be illustrated with the help of the following numerical examples. We computed four examples; first, we assumed two problems (Example 1 and Example 2) based on non-symmetric data, and we took another two problems...
(Example 3 and Example 4) based on symmetric data from [1]. Example 1.

**Find the minimum assignment cost of the matrix.**

\[
\begin{array}{cccc}
A_1 & A_2 & A_3 & A_4 \\
P_1 & 4 & 10 & 15 & 20 \\
P_2 & 12 & 6 & 16 & 8 \\
P_3 & 9 & 5 & 14 & 22 \\
P_4 & 25 & 20 & 7 & 18 \\
\end{array}
\]

**Solution:**

Step 1: Consider an assignment problem without fuzzy parameters.

Step 4: The numbers of rows and columns are equal. Therefore, it is a balanced assignment problem.

Step 6: Divide each element of the \(i\)th row of the matrix by the minimum element of each row, which should be written on the right hand side of the matrix.

\[
\begin{array}{cccc}
A_1 & A_2 & A_3 & A_4 \\
P_1 & \frac{1}{4} & 2.5 & 3.75 & 5 \rightarrow 4 \\
P_2 & \frac{2}{4} & 1 & 2.67 & 1.33 \rightarrow 6 \\
P_3 & \frac{1.8}{4} & 1 & 2.8 & 4.4 \rightarrow 5 \\
P_4 & \frac{3.57}{4} & 2.85 & 1 & 2.57 \rightarrow 7 \\
\end{array}
\]

Step 7: Here, there are no ones in fourth column, so determine the smallest element from column 4 and write it below the \(j\)th column of the matrix; then divide each element of the \(j\)th column of the matrix.

Therefore, we obtained the assignment cost as one in each row and each column.

\[
\begin{array}{cccc}
A_1 & A_2 & A_3 & A_4 \\
P_1 & 1 & 2.5 & 3.75 & 3.76 \\
P_2 & 2 & 1 & 2.67 & 1 \\
P_3 & 1.8 & 1 & 2.8 & 3.31 \\
P_4 & 3.57 & 2.86 & 1 & 1.93 \\
\end{array}
\]

Step 8: Identify the one’s position in each \((i,j)\)th row and column.

Step 9: Determine the value of each row’s succession of ones, pick the highest one, (see Table 1) and delete the associated row and column of the matrix and perform allocations.

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>(S(1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2.5(max)</td>
</tr>
<tr>
<td>2</td>
<td>2,4</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1.8</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1.93</td>
</tr>
</tbody>
</table>

Assign row 1 \(\rightarrow\) column 1 and delete the corresponding rows and columns.

In the reduced matrix, repeat step 8 until all of the rows are assigned and the optimal allocation for the assignment cost is obtained.
Now, the reduced matrix is

\[
P_2 \begin{pmatrix}
  1 & 2.67 & 1 \\
  1 & 2.8 & 3.31 \\
  2.86 & 1 & 1.93
\end{pmatrix}
\]

Locating the one's position

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>S(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2,4</td>
<td>2.67</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2.8</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1.93</td>
</tr>
</tbody>
</table>

Assign row 3 → column 2 and delete the corresponding rows and columns.

Now, the reduced matrix is

\[
P_2 \begin{pmatrix}
  2.67 & 1 \\
  1 & 1.93
\end{pmatrix}
\]

Locating the one's position

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>S(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>2.67(max)</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1.93</td>
</tr>
</tbody>
</table>

Assign row 2 → column 4 and row 4 → column 3.

Step 10: the assignment schedule is \( P_1 \to A_1, P_2 \to A_4, P_3 \to A_2, P_4 \to A_3 \)

The minimum assignment cost is 24.

**Example 2. [14]** Find the minimum assignment cost of the matrix

\[
P_1 \begin{pmatrix}
  10 & 5 & 13 \\
  3 & 9 & 18 \\
  10 & 7 & 2 \\
  7 & 11 & 9
\end{pmatrix}
\]

Solution:

Step 1: Consider the problem not in a fuzzy environment. Step 4: the assignment problem is not balanced.

Step 5, add a dummy column to make a balanced assignment problem.

\[
P_1 \begin{pmatrix}
  10 & 5 & 13 & 0 \\
  3 & 9 & 18 & 0 \\
  10 & 7 & 2 & 0 \\
  7 & 11 & 9 & 0
\end{pmatrix}
\]

Step 6: Divide each element of the i-th row of the matrix by the minimum element of each row, which should be written on the right hand side of the matrix.
Step 7: Determine the smallest element of each column and write it below the $j$th column of the matrix; then divide each element of the $j$th column of the matrix.

Therefore, we obtained the assignment cost as a one in each row and each column.

$$
\begin{bmatrix}
A_1 & A_2 & A_3 & A_4 \\
2 & 1 & 2.6 & 0 \\
1 & 3 & 6 & 0 \\
5 & 3.5 & 1 & 0 \\
1 & 1.6 & 1.3 & 0 \\
\end{bmatrix}
$$

Step 8: Identify the one’s position in each $(i,j)$th row and column.

Step 9: Determine the value of each row’s succession of ones, pick the highest one, and delete the associated row and column of the matrix and perform allocations.

Locating the One’s Position

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>S(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3.5(max)</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1.3</td>
</tr>
</tbody>
</table>

Assign row 3 $\rightarrow$ column 3 and delete the corresponding rows and columns.

In the reduced matrix, repeat step 8 until all of the rows are assigned and the optimal allocation for assignment cost is obtained.

$$
\begin{bmatrix}
A_1 & A_2 & A_4 \\
1 & 2 & 1 \\
2 & 1 & 3 \\
4 & 1 & 1.6 \\
\end{bmatrix}
$$

Now, the reduced matrix is

$$
\begin{bmatrix}
A_2 & A_4 \\
1 & 0 \\
4 & 1 \\
\end{bmatrix}
$$

Locating the one’s position

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>S(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3(max)</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1.6</td>
</tr>
</tbody>
</table>

Assign row 2 to column 1 and delete the corresponding rows and columns.

Now, the reduced matrix is

$$
\begin{bmatrix}
A_2 & A_4 \\
1 & 0 \\
4 & 1.6 \\
\end{bmatrix}
$$

Here, we got the $2 \times 2$ reduced matrix; the possible assignments are row 1 $\rightarrow$ column 2 and row 4 $\rightarrow$ column 4 (nothing is assigned to $P_4$).

Step 10: the assignment schedule is $P_1 \rightarrow A_2, P_2 \rightarrow A_1, P_3 \rightarrow A_3, P_4 \rightarrow A_4$

The minimum assignment cost is 10.

**Note:** As a result of the proposed algorithm, the assignment cost was derived under crisp parameters. Now, we apply the technique to fuzzy parameters such as triangular and trapezoidal fuzzy numbers with symmetric interval-valued fuzzy assignments.

**Example 3.** [1]
\(P_1, P_2, P_3, \text{ and } P_4\) are four professional programmers who work at a computer center. The center requires the development of four application program: \(A_1, A_2, A_3, \text{ and } A_4\). After carefully reviewing the program to be built, the director of the computer center estimates the computing time in minutes required by the experts for the application program using symmetric interval-valued triangular fuzzy numbers. Assume that \(w_{ij} = w_{ij}^u = 1\). Assign programmers to the program in a method that uses the least amount of computing time. The fuzzy assignment problem's cost matrix is

\[
\begin{bmatrix}
A_1 & A_2 & A_3 & A_4 \\
\begin{array}{cccc}
P_1 & (0.2,2),(-1,1) & (4.8,8),(3.7,7) & (28,30,30),(27,29,29) & (44,48,48),(43,47,47) \\
&P_2 & (4.8,8),(3.7,7) & (1,2,2),(0,1,1) & (33,36,36),(32,35,35) & (37,40,40),(36,39,39) \\
&P_3 & (23,25,25),(22,24,24) & (44,48,48),(43,47,47) & (1,2,2),(0,1,1) & (0,2,2),(-1,1) \\
&P_4 & (37,40,40),(36,39,39) & (4.8,8),(3.7,7) & (15,18,18),(14,17,17) & (15,18,18),(14,17,17) \\
\end{array}
\end{bmatrix}
\]

**Solution:**
Step 1: Consider the symmetric triangular fuzzy assignment problem

\[
\begin{bmatrix}
A_1 & A_2 & A_3 & A_4 \\
\begin{array}{cccc}
P_1 & (0.2,2),(-1,1) & (4.8,8),(3.7,7) & (28,30,30),(27,29,29) & (44,48,48),(43,47,47) \\
&P_2 & (4.8,8),(3.7,7) & (1,2,2),(0,1,1) & (33,36,36),(32,35,35) & (37,40,40),(36,39,39) \\
&P_3 & (23,25,25),(22,24,24) & (44,48,48),(43,47,47) & (1,2,2),(0,1,1) & (0,2,2),(-1,1) \\
&P_4 & (37,40,40),(36,39,39) & (4.8,8),(3.7,7) & (15,18,18),(14,17,17) & (15,18,18),(14,17,17) \\
\end{array}
\end{bmatrix}
\]

Step 3: Using the new ranking method [1],

\[
R(\vec{c}_{11}) = 0.275, \quad R(\vec{c}_{12}) = 2.035, \quad R(\vec{c}_{13}) = 9.505, \quad R(\vec{c}_{14}) = 15.235 \\
R(\vec{c}_{21}) = 2.035, \quad R(\vec{c}_{22}) = 0.385, \quad R(\vec{c}_{23}) = 11.385, \quad R(\vec{c}_{24}) = 12.705 \\
R(\vec{c}_{31}) = 7.865, \quad R(\vec{c}_{32}) = 15.235, \quad R(\vec{c}_{33}) = 0.385, \quad R(\vec{c}_{34}) = 0.275 \\
R(\vec{c}_{41}) = 12.705, \quad R(\vec{c}_{42}) = 2.035, \quad R(\vec{c}_{43}) = 5.445, \quad R(\vec{c}_{44}) = 5.445
\]

the above fuzzy assignment problem becomes

\[
\begin{bmatrix}
A_1 & A_2 & A_3 & A_4 \\
\begin{array}{cccc}
P_1 & 0.275 & 2.035 & 9.505 & 15.235 \\
&P_2 & 2.035 & 0.385 & 11.385 & 12.705 \\
&P_3 & 7.865 & 15.235 & 0.385 & 0.275 \\
&P_4 & 12.705 & 2.035 & 5.445 & 5.445 \\
\end{array}
\end{bmatrix}
\]

Step 4: number of rows = number of columns.
Step 5: Therefore, it is a balanced assignment problem.
Step 6: Divide each element of the \(i\)th row of the matrix by the minimum element of each row, which should be written on the right hand side of the matrix.
Step 7: Determine the smallest element of each column and write it below the \(j\)th column of the matrix; then divide each element of the \(j\)th column of the matrix.

Therefore, we obtained the assignment cost as one in each row and each column.
Step 8: Identify the one’s position in each \((i,j)\)th row and column.
Step 9: Determine the value of each row’s succession of ones, pick the highest one, and delete the associated row and column of the matrix and perform allocations.

Locating the One’s Position

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>(S(1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>7.4</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>5.29</td>
</tr>
<tr>
<td>3</td>
<td>3,4</td>
<td>28.6(max)</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2.68</td>
</tr>
</tbody>
</table>

Assign row 3 \(\rightarrow\) column 4 and delete the corresponding rows and columns.

In the reduced matrix, repeat step 8 until all of the rows are assigned and the optimal allocation for assignment cost is obtained.

Now, the reduced matrix is

\[
\begin{pmatrix}
A_1 & A_2 & A_3 \\
1 & 7.4 & 24.69 \\
5.29 & 1 & 21.12 \\
28.6 & 55.4 & 1 & 1 \\
6.24 & 1 & 2.68 & 2.68
\end{pmatrix}
\]

Locating the one’s position

<table>
<thead>
<tr>
<th>Row</th>
<th>Column</th>
<th>(S(1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>7.4(max)</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>5.29</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2.68</td>
</tr>
</tbody>
</table>

Assign row 1 \(\rightarrow\) column 1 and delete the corresponding rows and columns.

Now, the reduced matrix is

\[
\begin{pmatrix}
A_2 & A_3 \\
1 & 21.12 \\
2.68 & 1
\end{pmatrix}
\]

Again following step 6, so that matrix has ones in each and column. Then proceed to step 7 and step 8.

Assign row 2 \(\rightarrow\) column 2 & row 4 \(\rightarrow\) column 3.

Step 10: the assignment schedule is

\(P_1 \rightarrow A_1, P_2 \rightarrow A_2, P_3 \rightarrow A_4, P_4 \rightarrow A_3\)

and the total minimum assignment cost is 6.380.

Hence, the minimum symmetric fuzzy triangular assignment cost is \([(16,24,24),(12,20,20)]\).

Example 4. [1]

The Navy’s top officer has four subordinates and four responsibilities to complete. The efficiency of the subordinates varies, as do the duties’ inherent difficulty. His estimates of how long each subordinate would take to complete the each task (in hours) are expressed as symmetric interval-valued trapezoidal fuzzy numbers. Assume that
\[ w'_A = w'_B = 1. \] Determine the best assignment for reducing the total number of working hours. The fuzzy assignment problem’s cost matrix is

\[
\begin{array}{cccc}
S_1 & (1,2,3,3),(0,1,2,2) & (1,3,4,4),(0,2,3,3) & (9,11,12,12),(8,10,11,11) & (5,7,8,8),(4,6,7,7) \\
S_2 & (0,1,2,2),(-1,0,1,1) & (1,2,3,3),(0,1,2,2) & (5,6,7,7),(4,5,6,6) & (0,1,2,2),(-1,0,1,1) \\
S_3 & (3,5,6,6),(2,4,5,5) & (5,8,9,9),(4,7,8,8) & (12,15,16,16),(11,14,15,15) & (7,9,10,10),(6,8,9,9) \\
S_4 & (5,7,8,8),(4,6,7,7) & (1,5,6,6),(0,4,5,5) & (1,3,4,4),(0,2,3,3) & (1,2,3,3),(0,1,2,2) \\
\end{array}
\]

Solution:

Step 1: Consider the symmetric trapezoidal fuzzy assignment problem

\[
\begin{array}{cccc}
S_1 & (1,2,3,3),(0,1,2,2) & (1,3,4,4),(0,2,3,3) & (9,11,12,12),(8,10,11,11) & (5,7,8,8),(4,6,7,7) \\
S_2 & (0,1,2,2),(-1,0,1,1) & (1,2,3,3),(0,1,2,2) & (5,6,7,7),(4,5,6,6) & (0,1,2,2),(-1,0,1,1) \\
S_3 & (3,5,6,6),(2,4,5,5) & (5,8,9,9),(4,7,8,8) & (12,15,16,16),(11,14,15,15) & (7,9,10,10),(6,8,9,9) \\
S_4 & (5,7,8,8),(4,6,7,7) & (1,5,6,6),(0,4,5,5) & (1,3,4,4),(0,2,3,3) & (1,2,3,3),(0,1,2,2) \\
\end{array}
\]

Step 3: Using the new ranking method [1],

\[
R(\tilde{c}_{11}) = 0.735, \quad R(\tilde{c}_{12}) = 1.085, \quad R(\tilde{c}_{13}) = 4.915, \quad R(\tilde{c}_{14}) = 2.635 \\
R(\tilde{c}_{21}) = 0.345, \quad R(\tilde{c}_{22}) = 0.735, \quad R(\tilde{c}_{23}) = 2.295, \quad R(\tilde{c}_{24}) = 0.345 \\
R(\tilde{c}_{31}) = 1.86, \quad R(\tilde{c}_{32}) = 2.985, \quad R(\tilde{c}_{33}) = 5.705, \quad R(\tilde{c}_{34}) = 3.415 \\
R(\tilde{c}_{41}) = 2.635, \quad R(\tilde{c}_{42}) = 1.775, \quad R(\tilde{c}_{43}) = 1.085, \quad R(\tilde{c}_{44}) = 0.735
\]

the above fuzzy assignment problem becomes

\[
\begin{array}{cccc}
S_1 & 0.735 & 1.085 & 4.915 & 2.635 \\
S_2 & 0.345 & 0.735 & 2.295 & 0.345 \\
S_3 & 1.86 & 2.985 & 5.705 & 3.415 \\
S_4 & 2.635 & 1.775 & 1.085 & 0.735 \\
\end{array}
\]

Based on step 4 and step 5, the number of rows = the number of columns. Therefore, it is a balanced assignment problem.

Step 6: Divide each element of the ith row of the matrix by the minimum element of each row, which should be written on the right hand side of the matrix.

Step 7: Determine the smallest element of each column and write it below the jth column of the matrix; then divide each element of the jth column of the matrix.

Therefore, we obtained the assignment cost as one in each row and each column.

\[
\begin{array}{cccc}
S_1 & 1 & 1 & 3.867 & 3.585 \\
S_2 & 1 & 1.443 & 4.507 & 1 \\
S_3 & 1 & 1.087 & 2.078 & 1.836 \\
S_4 & 3.585 & 1.636 & 1 & 1 \\
\end{array}
\]

Step 8: Identify the one’s position in each (i,j)th row and column.
Step 9: Determine the value of each row’s succession of ones, pick the highest one, and delete the associated row and column of the matrix and perform allocations.

In the reduced matrix, repeat step 8 until all of the rows are assigned and the optimal allocation for the assignment cost is obtained.

Step 10: the assignment schedule is $S_1 \rightarrow T_2, S_2 \rightarrow T_4, S_3 \rightarrow T_1, S_4 \rightarrow T_3$, and the total minimum assignment cost is 4.375.

Hence, the minimum symmetric fuzzy trapezoidal assignment cost is $[(5,12,16,16),(1,8,12,12)]$.

7. Results and Discussion

This section focuses on a comparison diagram of the optimal assignment cost (see Figure 4) of the above examples of the two distinct types of assignment problems: assignment problems with a fuzzy parameter and assignment problems without a fuzzy parameter.

![Figure 4. Comparison of the optimal solution.](image)

In the above bar diagram, the existing method [1] converts the two different fuzzy numbers into crisp numbers by the new ranking method, solved by the traditional Hungarian method to obtain the assignment cost. In [10], the ranking method was used to convert the fuzzy and intuitionistic fuzzy into crisp numbers; the optimal solution was achieved using TORA software. In [14], a unique solution of the assignment problem was implemented using Python 3.8 as the programming language. In our proposed method, we first considered the assignment problem, located the one’s position in each row as well as column to perform allocations, and obtained the optimal cost. Later, we extended the algorithm for symmetric interval-valued fuzzy numbers and obtained the optimal solution. Here, we did not use the traditional Hungarian method. For the two types of problems, we obtained the optimal cost, which is the same as that in the existing method. It is easy to understand, compute, and also to interpret.

We presented a comparative analysis of the assignment problem with the crisp (balanced and unbalanced) and fuzzy (symmetric triangular number and symmetric trapezoidal number) parameters of several existing methods with the proposed approach of the corresponding optimal schedule and optimal cost.

8. Conclusions

In this work, we proposed a new algorithm to compute an optimal solution for assignment problems. This proposed method differs in the procedure of finding the optimal solution with the classical Hungarian method in allocating zero’s. This proposed al-
Algorithm requires less calculation, and the given \((n \times n)\) matrix is reduced to the \((n - 1)\) matrix in each iteration when we obtained its optimal solution. Using the proposed method, we presented four different types of assignment problems, i.e., balanced assignment problems, unbalanced assignment problems, symmetric interval-valued triangular assignment problems, and symmetric interval-valued trapezoidal assignment problems. The optimal solution for the proposed algorithm requires less calculation with less iteration when compared to the existing method. The numerical value shown in Table 2 represents a comparison between the proposed and existing methods, which shows the effectiveness of the proposed approach. Future research on the use of this proposed approach will apply this concept to symmetric intuitionistic fuzzy sets. Therefore, the proposed technique provides the optimal solution to distinct symmetric fuzzy assignment problems.

Table 2. Comparative Analysis

<table>
<thead>
<tr>
<th>Author name</th>
<th>Method</th>
<th>Optimal Schedule</th>
<th>Optimal Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Existing Method</td>
<td>Proposed Method</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Existing Method</td>
<td>Proposed Method</td>
</tr>
<tr>
<td>Senthil Kumar [10]</td>
<td>TORA Software</td>
<td>(x_{11}=4, x_{24}=8,)</td>
<td>(x_{11}=4, x_{24}=8,)</td>
</tr>
<tr>
<td>Example 8.4</td>
<td></td>
<td>(x_{32}=5, x_{43}=7,)</td>
<td>(x_{32}=5, x_{43}=7,)</td>
</tr>
<tr>
<td>Rezaul Karim [14]</td>
<td>Python 3.8 programming language</td>
<td>(x_{12}=5, x_{21}=3,)</td>
<td>(x_{12}=5, x_{21}=3,)</td>
</tr>
<tr>
<td>Example 2</td>
<td></td>
<td>(x_{33}=2, x_{44}=0,)</td>
<td>(x_{33}=2, x_{44}=0,)</td>
</tr>
<tr>
<td>Amutha, Uthra [1]</td>
<td>New Ranking method</td>
<td>(x_{11}=0.275, x_{22}=0.385,)</td>
<td>(x_{11}=0.275, x_{22}=0.385,)</td>
</tr>
<tr>
<td>Example 1</td>
<td></td>
<td>(x_{34}=0.27,)</td>
<td>(x_{34}=0.27,)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(x_{43}=5.445)</td>
<td>(x_{43}=5.445)</td>
</tr>
<tr>
<td>Uthra et al. [16]</td>
<td>Using Yager’s ranking method</td>
<td>(x_{11}=1.5, x_{22}=1.75,)</td>
<td>(x_{11}=1.5, x_{22}=1.75,)</td>
</tr>
<tr>
<td>Example 1</td>
<td></td>
<td>(x_{34}=1.5, x_{43}=17.25)</td>
<td>(x_{34}=1.5, x_{43}=17.25)</td>
</tr>
<tr>
<td>Amutha, Uthra [1]</td>
<td>New Ranking method</td>
<td>(x_{11}=0.275, x_{24}=0.385,)</td>
<td>(x_{11}=0.275, x_{24}=0.385,)</td>
</tr>
<tr>
<td>Example 1</td>
<td></td>
<td>(x_{34}=0.27,)</td>
<td>(x_{34}=0.27,)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(x_{43}=5.445)</td>
<td>(x_{43}=5.445)</td>
</tr>
</tbody>
</table>

**Author Contributions:** Conceptualization, S.V.G.; methodology, S.V.G. and M.J.; writing – original draft preparation, S.V.G.; writing review and editing, S.V.G. and M.J.; supervision, M.J.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research work is supported by Vellore Institute of Technology, Vellore.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Acknowledgement:** The authors wish to thank the management of Vellore Institute of Technology, Vellore -632014 for their continuous support and encouragement to carry out this research work.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**


