MULTI-OBJECTIVE PROBLEM WITH MULTIPLE JOBS ASSIGNED TO A SINGLE MACHINE WITHIN AVAILABLE COST UNDER UNCERTAIN ENVIRONMENT

Aamir Khan[2], Quazzafi Rabbani[2] and Ahteshamul Haq[3, *]

•

[1,2,3]Department of Mathematics &Statistics, Integral University Lucknow-226026, India
[1]aamir25dec@gmail.com, [2]quazzafi@gmail.com, [3,*]a.haq@myamu.ac.in

Abstract

The assignment problem is a key challenge in optimization and operations research, finding applications in diverse real-world scenarios. The Hungarian method is a widely employed algorithm for solving this problem, especially in its balanced form. However, for unbalanced assignment problems, where tasks outnumber resources (or vice versa), an extension is necessary. One common approach introduces a dummy resource, but this may leave tasks unassigned. The Modified Hungarian method improves upon the standard algorithm for unbalanced problems, ensuring that all tasks are assigned to real resources. This is achieved by modifying the cost matrix and algorithm steps to accommodate additional tasks and resources. Triangular fuzzy numbers are discussed when exact parameter information is undefined, and fuzzy programming is applied to determine a compromise result. Incorporating cost and profit per resource, the Modified Hungarian algorithm addresses the problem of unspecified job allocations to a single machine by introducing a cost parameter for each machine. The methodology is demonstrated on a numerical example for better comprehension.

Keywords: Triangular Fuzzy Number, Unbalanced Assignment Problem, Modified Hungarian Technique, Fuzzy Programming Approach

I. Introduction

In the contemporary landscape of industrial and operational management, the allocation of tasks to machines stands as a pivotal challenge, especially when confronted with the intricacies of handling multiple objectives concurrently. This challenge becomes particularly nuanced when multiple jobs must be assigned to a single machine, governed by predefined cost limitations, and set against the backdrop of an environment characterized by uncertainties. The fusion of these elements gives rise to a multifaceted problem, demanding sophisticated optimization techniques to strike an equilibrium among conflicting goals while ensuring resource efficiency. The optimization of job allocation on a single machine is a critical concern across various industries, spanning manufacturing, services, and beyond. The efficient utilization of resources in the face of dynamic and uncertain conditions is essential for organizations striving to enhance productivity, reduce costs, and maintain adaptability in an ever-evolving business environment. This intricate dance of optimization unfolds against the backdrop of challenges such as varying processing times, resource availability fluctuations, and external environmental factors, all of which contribute to the complexity of the multi-objective problem at hand.

A foundational element in the exploration of the multi-objective problem is the extensive body of research on multi-objective optimization in job scheduling. Studies [1] and [2] have pioneered methodologies for handling conflicting objectives, including cost reduction, and resource utilization. These works have laid the groundwork for understanding the trade-offs involved in optimizing multiple objectives simultaneously, providing essential concepts for application in the context of job allocation on a single machine. The integration of uncertainty into optimization models is a crucial aspect of addressing real-world operational challenges. Researchers, including [3, 4], have explored various approaches for modeling uncertainty in scheduling and optimization problems. Techniques such as stochastic programming, fuzzy logic, and robust optimization have been employed to account for uncertainties in processing times, resource availability, and external factors. Understanding these methodologies is vital for adapting optimization models to the uncertain environment inherent in the multi-objective problem under consideration. The interplay between cost constraints and optimization objectives has been a focal point in operations research. Works such as the study by [5] have investigated cost-sensitive optimization models, aiming to strike a balance between achieving objectives and adhering to budget limitations. These insights are particularly relevant in the context of allocating multiple jobs to a single machine, where cost constraints play a pivotal role in decision-making. Recent advancements have seen the emergence of hybrid and metaheuristic approaches in solving complex optimization problems. Research by [6, 7] exemplifies the application of genetic algorithms, simulated annealing, and other metaheuristic techniques to address combinatorial optimization problems. These approaches offer promise in handling the intricate nature of the multi-objective problem, providing effective means to navigate the solution space efficiently. To bridge the gap between theoretical models and practical implementation, several studies have presented real-world applications and case studies. Works by authors like [8, 9] have demonstrated the applicability of optimization models in industries such as manufacturing, healthcare, and logistics. Examining these cases provides valuable insights into the challenges faced by practitioners and the effectiveness of proposed methodologies in diverse operational contexts.

Our ultimate goal is to contribute to the evolving landscape of operations research and optimization by presenting novel insights and frameworks that not only tackle the complexities of the multi-objective problem but also address the uncertainties inherent in real-world industrial settings. By doing so, we aspire to furnish decision-makers and practitioners with a comprehensive toolkit that enables them to navigate the labyrinth of multi-objective optimization within the constraints of cost and uncertainty, fostering resilience and agility in their operational strategies. Through this exploration, we aim to illuminate pathways towards a more efficient, adaptive, and sustainable operational paradigm.

The assignment problem and the transportation problem [10] are both types of optimization problems in operations research and linear programming, but they are not exactly the same. The assignment problem is indeed a special case of the transportation problem, with certain constraints and characteristics that make it more specific. There are various methods to solve the assignment problem, including enumeration methods, the simplex method, and the Hungarian method [11]. In a balanced assignment problem, the number of jobs equals the number of machines, and the Hungarian method is generally very convenient and capable of finding the optimal assignment. In an unbalanced assignment problem, where the number of jobs and machines is not equal, it may not be possible to assign all jobs to machines. In such cases, some jobs may remain unassigned. It's important to note that real-world scenarios may indeed involve unbalanced assignment problems. In such situations, it might be necessary to address the unassigned jobs differently, perhaps by revising the problem formulation or considering additional constraints to handle the unbalance. Suppose, in a production factory there are five machines and eight numbers of jobs. If one machine can do only one job than remaining jobs not executed, that's create problem in manufacturing process. The Hungarian approach is improved by [12]. Kumar [13] provided a strategy for overcoming the imbalanced assignment problem that involves executing all jobs. Rabbani *et al.* [14] changed the formulation of the Hungarian approach by allowing the user to allocate several jobs to a single machine while without limiting any machine to a maximum number of jobs. The focus of this exploration lies in dissecting the intricacies of the "Multi-Objective Problem with Multiple Jobs Assigned to a Single Machine within Available Cost under Uncertain Environment." As we embark on this journey, our aim is to delve deep into the complexities of this operational puzzle, surveying existing methodologies, pinpointing gaps in current approaches, and proposing innovative frameworks that offer robust solutions to the multifaceted challenges faced by industries today.

In this paper, we will navigate through the theoretical foundations of multi-objective optimization, exploring its applications in the context of job allocation on a single machine. We will scrutinize the influence of uncertain variables on this optimization process, emphasizing the dynamic nature of real-world operational scenarios. Additionally, we will investigate the existing tools and methodologies employed in addressing similar challenges, critically evaluating their strengths and limitations. This work focuses on cost and profit management for jobs performed by machines, using the concept of [14] and providing cost parameters for constraints on each machine. Jobs are assigned to each machine based on the cost of each machine. For better comprehension, a step-by-step approach is provided and solved using a numerical example.

The motivation for studying the assignment problem originates from its importance in optimization and operations research, with numerous applications in real-life situations. While the Hungarian approach works well for balanced assignments, it struggles with imbalanced problems in which one job or resource exceeds another. The Modified Hungarian technique is unique in that it improves on the normal algorithm's ability to manage unbalanced assignments. By adjusting the cost matrix and algorithm stages, it assures that all jobs find actual resource allocations, eliminating the problem of tasks remaining unassigned while applying dummy resources.

The Modified Hungarian algorithm introduces a novel solution to the problem of unspecified job allocations to a single machine. By assigning a cost parameter to each machine and considering both cost and profit per resource, it provides a more comprehensive and realistic model for solving assignment problems. The methodology's practicality is demonstrated through a numerical example, enhancing understanding and showcasing the applicability of the Modified Hungarian algorithm in real-world scenarios. Furthermore, the use of triangular fuzzy numbers addresses cases where accurate parameter information is unavailable, resulting in a more flexible method. In such cases, the use of fuzzy programming might help identify a compromise solution.

II. Assumption of unbalanced assignment problem

- Here we consider the numbers of machines are always less than the numbers of jobs.
- Each machine is capable of performing multiple jobs.
- Each job can assign to only one machine, and no job can be assigned to multiple machines simultaneously.
- Every machine is assigned at least one task, and there are no tasks that remain unallocated to a machine.
- If a machine has multiple tasks to complete, it can perform them sequentially or in succession.
- The sum of the costs associated with the tasks assigned to each machine does not exceed the machine's available budget or cost limit.

III. Formulation

In a scenario with 'm' machines and 'n' jobs (where 'n' is greater than 'm'), each machine has a specific effectiveness or cost associated with it. The challenge here is to efficiently assign each job to one machine, ensuring that if there are more jobs than machines, the excess jobs are queued and processed subsequently.

Let's denote c_{ij} as the cost of assigning the i^{th} machine to the j^{th} job, where $i = 1,2,\cdots,m$ and $j = 1,2,\cdots,n$. Our aim to find an optimal assignment for the problem, determining the job that will be allocated to each machine, in a way to minimize the overall cost and maximize the profit incurred while performing all the tasks. It's important to note that the sum of jobs exceeds the sum of available machines, and each machine can handle multiple jobs, but a single job cannot be assigned on two machines. Additionally, each machine has a specified budget or cost that is utilized for executing the assigned jobs. This problem can be represented using a $m \times n$ cost matrix $[c_{ij}]$ (Table 1) and a profit matrix $[p_{ij}]$ (Table 2).

Table 1: *The cost matrix c_{ij} in the form of* $m \times n$

	J_1	J_2	⋯	J_n	Cost
M_1	c_{11}	c_{12}	⋯	c_{1n}	C_1
M_2	c_{21}	c_{22}	⋯	c_{2n}	C_2
⋮	⋮	⋮	⋮	⋮	⋮
M_m	c_{m1}	c_{m2}	⋯	c_{mn}	C_m

Table 2: *The profit matrix p_{ij} in the form of* $m \times n$

	J_1	J_2	⋯	J_n
M_1	p	p_{12}	⋯	p_{1n}
M_2	p_{21}	p_{22}	⋯	p_{2n}
⋮	⋮	⋮	⋮	⋮
M_m	p_{m1}	p_{m2}	⋯	p_{mn}

Let x_{ij} denote the i^{th} machine is assigned for j^{th}job such that

$$x_{ij} = \begin{cases} 1, & if\ i^{th}\ machine\ is\ assigned\ to\ j^{th} job \\ 0, & if\ i^{th}\ machine\ is\ not\ assigned\ to\ j^{th} job \end{cases}$$

The Mathematical model is stated as

$$\text{Minimize } Z_1 = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \quad (1)$$

$$\text{Maximize } Z_2 = \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} x_{ij} \quad (2)$$

subject to constraints

$$\sum_{j=1}^{n} x_{ij} \geq 1; for\ i = 1,2,\dots,m \quad (3)$$

$$\sum_{i=1}^{m} x_{ij} = 1; for\ j = 1,2 \dots,n \quad (4)$$

$$\sum_{j=1}^{n} c_{ij} x_{ij} \leq C_i;\ for\ i = 1,2,\dots,m \quad (5)$$

$$x_{ij} = 0\ or\ 1 \quad (6)$$

Eq [1-2] represents the objective functions, while Eq [3] specifies that a machine has the capability to handle multiple jobs. Eq [4] enforces the constraint that no identical job can be assigned to multiple machines, and Eq [5] signifies that the total assignment cost for any machine must not exceed its available budget. Lastly, Eq [6] indicates that only binary values are permissible in this context.

IV. Method

Solve multiple goal optimization problems by focusing on one objective at a time and disregarding the others with the restrictions provided. The following are the step-by-step procedures:

1. Examine the problem, Is the problem certain? If so, skip the defuzzification approach. Otherwise, change the problem into a neat model.
2. To reduce fuzzy problems into crisp equivalent forms, the defuzzification approach of the ranking function is applied.
   Using the defuzzification approach, we turn it into the corresponding crisp form. To get a similar crisp form, the ranking function [15, 16] was applied.
   Let$\tilde{P}_j = \left(P_j^{(1)}, P_j^{(2)}, P_j^{(3)}\right)$be a triangular fuzzy number, then the following equation must be used to compute its magnitude:
   $$M\left(\tilde{P}_j\right) = \frac{P_j^{(1)} + 4P_j^{(2)} + P_j^{(3)}}{6} \tag{7}$$
3. Input m, n
4. Find the lowest cost in each row and deduct it from the relevant row, resulting in at least one zero in each row.
5. Check all columns; if any column remains without producing a zero, choose the lowest cost of that column and subtract it from all of the values for that column to produce a zero in that column.
6. Draw the fewest lines possible to cover the zeros in order to get the optimum matrix.
7. If the number of lines does not match the number of machines, choose the least uncovered cost and deduct it from each uncovered cost before adding the intersection of lines.
8. Repeat steps 6 and 7 until the number of lines equals the number of rows.
9. To assign the job, identify the smallest number of zeros in each row or column, assign that zero to the appropriate machine, and remove the actual cost of the assigned job from the available cost for that machine.
10. We cross the remaining zeros in the relevant column after allocation, and if there will be availability of cost for that machine is completed, also cross the remaining zeros in that row (the total cost of allotted jobs to a single machine cannot exceed the availability cost of that machine).
11. In the event of a tie, i.e., two rows or columns with the same number of zeros; assign the zero with the lowest cost in the original problem. There will be no duplicate jobs assigned two separate machines, and no machine will be left without assigning a job.
12. Repeat steps 9–11 until each job have been allocated.
13. End of algorithm.

The resulting solution is the idle solution. Using the idle solution, we constructed the payoff matrix. The payoff matrix will help in the development of the desired level for each objective function.

Fuzzy Goal Programming

The Fuzzy Goal Programming is a strong and adaptable approach that may be used to a wide range of decision-making issues with multiple objectives [17]. As a result, we can take advantage of this method to get the most effective solution for the specified models. The following are the step-by-step procedure:

- The resulting solution is the idle solution. The idle solution will help in constructing the payoff matrix. Finally, the payoff matrix helps in the development of the desired level for each objective function.

- The target value is established as the objective function's goals level $(g_k, k = 1,2)$.
- We constructed the fuzzy linear membership function for the fuzzy goal of $Z(X) \preceq g$ (*i.e.,* fuzzy-min) is as follows:

$$\mu(Z(X)) = \begin{cases} 1, & if\ Z(X) \le g \\ \dfrac{U - Z(X)}{U - g}, & if\ g \le Z(X) \le U \\ 0, & if\ Z(X) \ge U \end{cases}$$

where, $U$is the fuzzy goal's the highest tolerance limit of $Z(X)$.

Moreover, if the fuzzy goal $Z(X) \succeq g$ (*i.e.,* fuzzy-max), then, the membership function is as follows:

$$\mu(Z(X)) = \begin{cases} 1, & if \quad Z(X) \ge g \\ \dfrac{Z(X) - L}{g - L}, & if \quad L \le Z(X) \le g \\ 0, & if \quad Z(X) \le L \end{cases}$$

Where L is the fuzzy goal's lower tolerance limit of $Z(X)$.

- Finally, we use the linear membership function to convert multi-objective problem into single objective problem which can be solved by using a suitable traditional optimization method. The fuzzy achievement function μ is maximized.

V. Numerical Illustration

We consider a numerical example to illustrate in which five machines are offered to complete eight jobs with related cost (in USD).
The input parameters are in the form of Triangular fuzzy number for transportation cost (Table 3) and profit (Table 4).

Table 3: *The transportation cost in the form of triangular fuzzy number*

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	Cost (USD)
M_1	26, 30, 34	24, 25, 26	38, 40, 42	45, 50, 55	33, 35, 37	24, 25, 26	39, 40, 41	24, 25, 26	26, 30, 34
M_2	38, 40, 42	26, 30, 34	22, 20, 24	22, 20, 24	22, 20, 24	41, 45, 49	22, 20, 24	24, 25, 26	38, 40, 42
M_3	22, 20, 24	38, 40, 42	26, 30, 34	38, 40, 42	26, 30, 34	45, 50, 55	26, 30, 34	38, 40, 42	45, 50, 55
M_4	24, 25, 26	22, 20, 24	33, 35, 37	26, 30, 34	24, 25, 26	26, 30, 34	33, 35, 37	26, 30, 34	57, 60, 63
M_5	33, 35, 37	33, 35, 37	45, 50, 55	38, 40, 42	38, 40, 42	57, 60, 63	45, 50, 55	38, 40, 42	74, 80, 86

Table 4: *The transportation profit in the form of triangular fuzzy number*

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8
M_1	8, 10, 12	11, 12, 13	14, 15, 16	21, 23, 25	15, 17, 19	10, 11, 12	15, 19, 23	8, 9, 10
M_2	13, 14, 15	6, 8, 10	5, 7, 9	4, 5, 6	2, 4, 6	20, 22, 24	1, 3, 5	5, 6, 7
M_3	5, 7, 9	15, 16, 17	11, 13, 15	15, 17, 19	11, 13, 15	25, 27, 29	10, 11, 12	14, 15, 16
M_4	6, 8, 10	1, 3, 5	12, 14, 16	11, 12, 13	8, 9, 10	6, 8, 10	20, 21, 22	8, 10, 12
M_5	11, 13, 15	10, 11, 12	21, 23, 25	12, 14, 16	20, 21, 22	27, 29, 31	15, 19, 23	14, 15, 16

The defuzzication approache ranking function (Equation [7]) is used to convert it into a crisp equivalent form for Table 3 & 4 will be shown in table 5 (Transportation cost) & Table 6 (Transportation profit).

Table 5: *Crisp equivalent form of transportation cost*

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	Cost (USD)
M_1	30	25	40	50	35	25	40	25	30
M_2	40	30	20	20	20	45	20	25	40
M_3	20	40	30	40	30	50	30	40	50
M_4	25	20	35	30	25	30	35	30	60
M_5	35	35	50	40	40	60	50	40	80

Table 6: *Crisp equivalent form of transportation profit*

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8
M_1	10	12	15	23	17	11	19	9
M_2	14	8	7	5	4	22	3	6
M_3	7	16	13	17	13	27	11	15
M_4	8	3	14	12	9	8	21	10
M_5	13	11	23	14	21	29	19	15

The cost associated with machines for jobs will be represented in crisp form as shown in Table 7.

Table 7: *The cost associated with machines for jobs (in USD)*

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	Cost (USD)
M_1	30	25	40	50	35	25	40	25	30
M_2	40	30	20	20	20	45	20	25	40
M_3	20	40	30	40	30	50	30	40	50
M_4	25	20	35	30	25	30	35	30	60
M_5	35	35	50	40	40	60	50	40	80

Find the simplest cost in each row and subtract it from the row that results in at least one zero in each row.
If any column remains without a zero, choose the lowest cost in that column and subtract it from all of the items in that column to obtain zero (Table 8).

Table 8: *Zeros row and column*

	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	Cost (USD)
M_1	5	0	15	25	10	0	15	0	30
M_2	20	10	0	0	0	25	0	5	40
M_3	0	20	10	20	10	30	10	20	50
M_4	5	0	15	5	5	10	15	10	60
M_5	0	0	15	5	5	25	15	5	80

Draw the number of lines required covering all zeros; in this case, four lines are required to cover all zeros, and the number of lines is not equal to the number of machines (Table 9). Go to step 7.

Table 9: *Line covered to all zeros*

	J1	J2	J3	J4	J5	J6	J7	J8	Cost (USD)
M1	5	0	15	25	10	0	15	0	30
M2	20	10	0	0	0	25	0	5	40
M3	0	20	10	20	10	30	10	20	50
M4	5	0	15	5	5	10	15	10	60
M5	0	0	15	5	5	25	15	5	80

Select the least uncovered cost, i.e. 5, and subtract it from each uncovered cost and add it to the intersection point (Table 10). Also, check that there are zeros in each row and column. Draw lines to cover all of the zeros. In this case, five lines are drawn that are equal to the number of rows to generate the needed matrix.

Table 10: *New line on updated values that covering all zeros*

	J1	J2	J3	J4	J5	J6	J7	J8	Cost (USD)
M_1	10	5	15	25	10	0	15	0	30
M_2	25	15	0	0	0	25	0	5	40
M_3	0	20	5	15	5	25	5	15	50
M_4	5	0	10	0	0	5	10	5	60
M_5	0	0	10	0	0	20	10	0	80

Begin by assigning jobs to the rows. Find a row with only one zero, assign that zero, then cross the other zeros in that column i.e. J_1 assign to M_3 and subtract the cost of the allocated job from the available cost. Here, the available cost for machine M_3 is 50 USD, and after assigning the work, the remaining cost is 30 USD. Now, verify the columns that have one zero allocated to the relevant machine i.e.,J_3,J_7 assign to M_2 and J_6 assign to M_1 and subtract the cost from the corresponding machine's available cost (Table 11).

Table 11: *Available cost of corresponding machine*

	J1	J2	J3	J4	J5	J6	J7	J8	Cost (USD)
M_1	10	5	15	25	10	0	15	0	5
M_2	25	15	0	0	0	25	0	5	0
M_3	0	20	5	15	5	25	5	15	30
M_4	5	0	10	0	0	5	10	5	40
M_5	0	0	10	0	0	20	10	0	0

In the event of a tie, i.e., two rows or columns with the same number of zeros, we allocate the zero with the lowest cost in the problem. There will be no duplicate jobs assigned two separate machines, and no machine will be left without assigning at least one job.

If the cost is not accessible to do any more jobs for that machine after it has been assigned, and there are still any position related to that machine, then cross it for not assigning any other job.

Continue step 8 to 10 until all jobs are assigned.

Table 12: *Jobs assigned to all machine*

	J1	J2	J3	J4	J5	J6	J7	J8	Cost (USD)
M_1	15	5	15	25	10	0	15	0	5
M_2	30	15	0	0	0	25	0	5	0
M_3	0	15	0	10	0	20	0	10	0
M_4	10	0	10	0	0	5	10	5	15
M_5	5	0	10	0	0	20	10	0	40

Table 13 shows the task assignment that minimizes overall cost, and Figure 1 shows a graph of job allocation to the respective machine.

Table 13: *Jobs assigned to respected machines with the associated cost*

Machine	*Jobs*	*Cost (USD)*
M_1	J_6	*25*
M_2	J_3, J_4	*20+20=40*
M_3	J_1, J_7	*20+30=50*
M_4	J_2, J_5	*20+25=45*
M_5	J_8	*40*
Total Cost		*200*

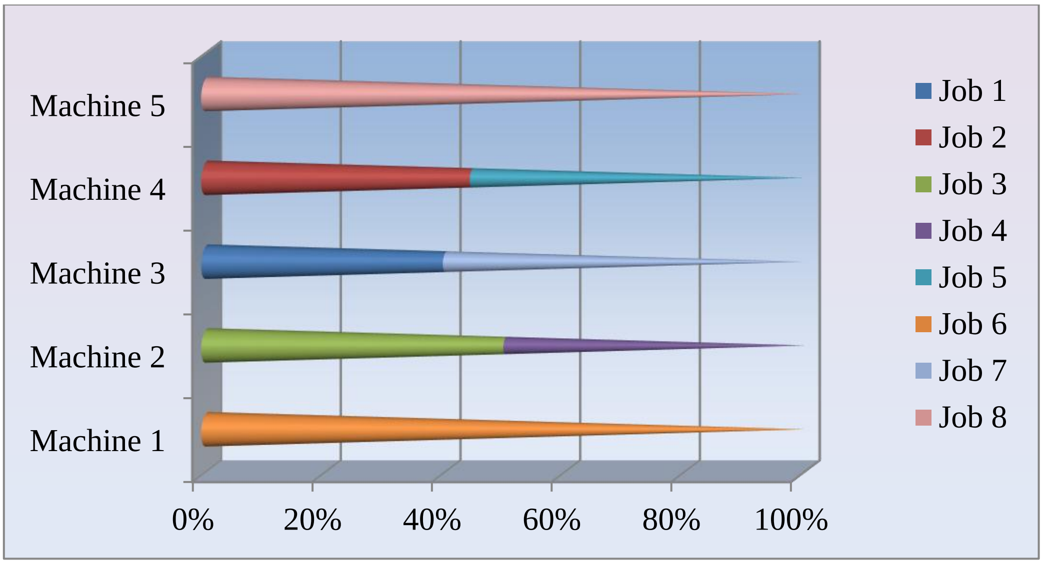

Figure 1: *Allocation of Jobs to the Machines for cost*

The second objective is maximizing type, so we select the maximum value from the Table 6 and subtract each element of the Table 6 from the selected value. Similarly, Table 14 shows the assignment of work that maximizes profit, and Figure 2 shows a graph of job allocation to the respective machine.

Table 14: *Jobs assigned to respected machines with the associated profit*

Machine	Jobs	Profit (USD)
M_1	J_2	12
M_2	J_3, J_4	7+5=12
M_3	J_6	27
M_4	J_5, J_7	9+21=30
M_5	J_1, J_8	13+15=28
Total Profit		109

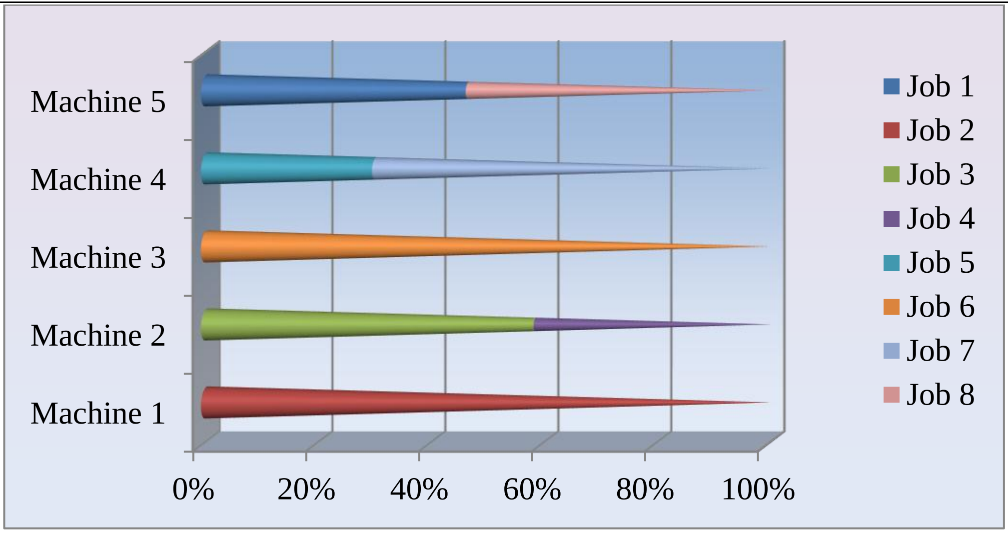

Figure 2: *Allocation of Jobs to the Machines for the profit*

After solving each objective, we have lower and upper bound of each objective such as: $200 \leq Z_1(x) \leq 250, 68 \leq Z_2(x) \leq 109$. Then, we construct the membership function of each objective are as follows:

$$\mu_1(Z_1(x)) = \begin{cases} 1, & Z_1(x) \leq 200 \\ \dfrac{250 - Z_1(x)}{250 - 200} & 200 < Z_1(x) < 250 \\ 0 & Z_1(x) \geq 250 \end{cases}$$

and

$$\mu_2(Z_2(x)) = \begin{cases} 0, & Z_2(x) \leq 68 \\ \dfrac{Z_2(x) - 68}{109 - 68}, & 68 < Z_2(x) < 109 \\ 1, & Z_2(x) \geq 109 \end{cases}$$

Use the fuzzy programming approach with these memberships, we get a compromise solution for the assignment of jobs as shown in Table 15 and graphical illustrating work allocation to the relevant machines in terms of cost and profit are shown in Figure 3 and 4 respectively.

Table 15: *Jobs assigned to respected machines with the associated profit*

Machine	Jobs	Cost (USD)	Profit (USD)
M_1	J_6	25	11
M_2	J_3, J_4	20+20=40	7+5=12
M_3	J_1, J_5	20+30=50	7+13=20
M_4	J_2, J_7	20+35=55	3+21=24
M_5	J_8	40	15
Total Profit		210	82

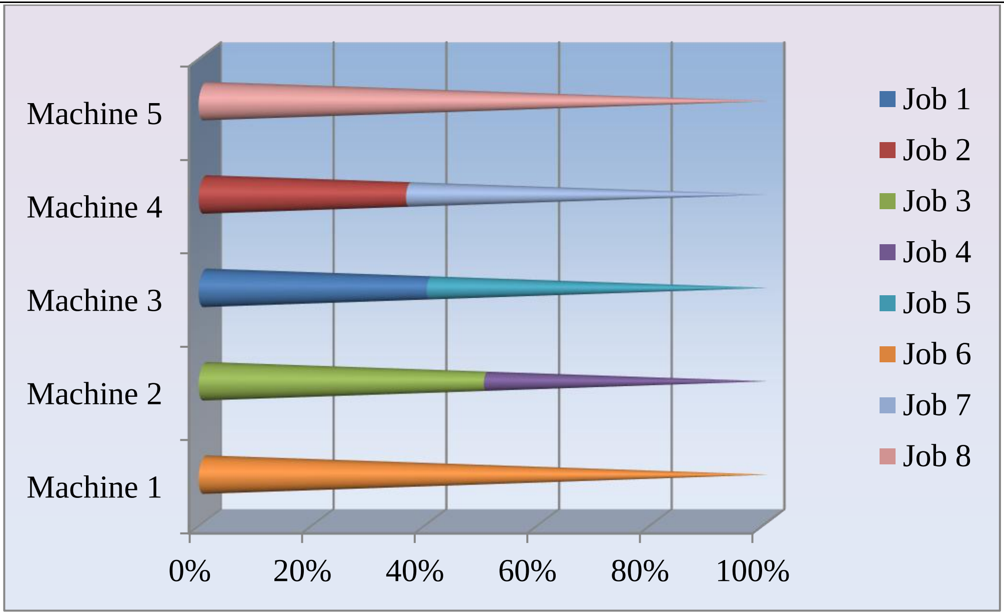

Figure 3: *Compromise allocation of Jobs to the Machines for the cost*

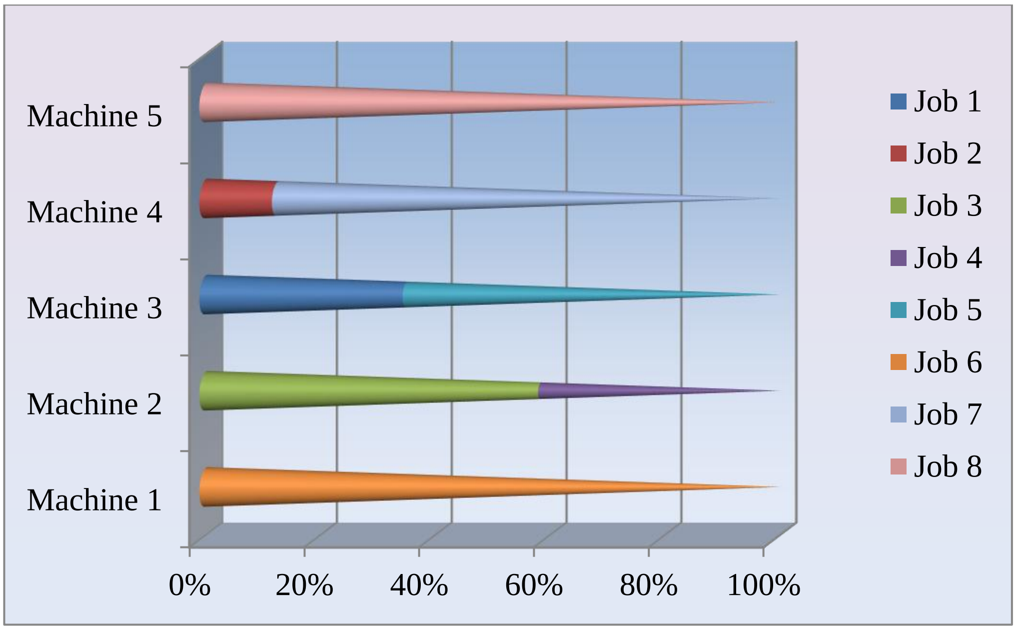

Figure 4: *Compromise allocation of Jobs to the Machines for profit*

VI. Discussion

Acknowledgment of the assignment problem's critical role in optimization and operations research, with broad applicability in real-world scenarios. Recognition of the Hungarian method's effectiveness in addressing the assignment problem in its balanced form. Expansion of the exploration to encompass unbalanced scenarios where tasks or resources outnumber each other. Introduction of a new and straightforward strategy for assigning multiple activities within given resources to achieve specific goals. Addition of two parameters, cost and profit for each resource, in the formulation to meet the demand of allocating more jobs to one resource within a specified time frame. Utilization of fuzzy programming to find compromise solutions for multi-objective problems, offering adaptability in scenarios with undefined parameter information. In-depth exploration of triangular fuzzy numbers when the exact information of parameters is not defined, showcasing a flexible approach in problem-solving. Recognition of the Modified Hungarian method as a significant advancement in rectifying the shortcomings of leaving tasks unassigned. Explanation of how modifications to the cost matrix and algorithm steps ensure the comprehensive allocation of tasks to real resources.

The role of triangular fuzzy numbers in providing adaptability in scenarios with undefined parameters. Discussion on the contribution of fuzzy programming to compromise result determination, enhancing the methodology's effectiveness. Emphasis on the incorporation of cost and profit considerations per resource, adding a practical and comprehensive dimension to the algorithm. Explanation of how the Modified Hungarian algorithm adeptly handles unspecified job allocations by introducing a cost parameter for each machine. Recognition of the methodology's practical demonstration on a numerical example as a key element for enhancing comprehension. Emphasis on how the envisioned future scope aims to ensure the continued evolution and broader applicability of the Modified Hungarian algorithm in effectively addressing intricate assignment problems across diverse and dynamic contexts.

I. Needs

- ❖ Address the fundamental need for optimization in various real-life applications.
- ❖ Fulfill the need for effective solutions in scenarios where tasks significantly outnumber resources or vice versa.
- ❖ Cater to the need for flexibility when dealing with scenarios lacking exact parameter information through the discussion of triangular fuzzy numbers.
- ❖ Meet the need for a comprehensive approach by incorporating both cost and profit considerations for each resource in the formulation.

II. Limitations

- ➢ Acknowledge the limitation of dependency on the Hungarian method, particularly in its balanced form.
- ➢ Recognize the potential limitation of leaving tasks unassigned when utilizing the common approach of introducing a dummy resource.
- ➢ Understand that the Modified Hungarian method, while addressing unbalanced problems, may introduce complexity in modifying cost matrices and algorithm steps.
- ➢ Acknowledge the subjective nature of fuzzy programming in finding a compromise result, as it relies on fuzzy logic and human judgment.

VII. Conclusion

The assignment problem, crucial in optimization and operations research, spans diverse real-world applications. While the Hungarian method adeptly tackles the problem in its balanced form, the exploration extends to address unbalanced scenarios where tasks or resources outnumber each other. In this work, a new and easy strategy for assigning multiple activities within the given resources for a particular goal is proposed. Two more parameters as cost and profit for each resource is included in the formulation. This strategy meets the demand of allocating more jobs to one resource within a certain time frame while producing the most effective results. The fuzzy programming approach is used to find the compromise solution to a multi-objective problem. We also discussed triangular fuzzy number if the exact information of the parameter is not defined. In future, researcher can used different fuzzy number for the allocation of the jobs to the machine and also used different techniques under the different circumstances.

The Modified Hungarian method, a significant advancement, rectifies the shortcomings of leaving tasks unassigned. By modifying the cost matrix and algorithm steps, it ensures all tasks find allocation to real resources. The discussion on triangular fuzzy numbers provides adaptability in scenarios with undefined parameters, and fuzzy programming contributes to compromise result determination. Incorporating cost and profit considerations per resource, the Modified Hungarian algorithm adeptly handles unspecified job allocations by introducing a cost parameter for each machine. The methodology's practical demonstration on a numerical example enhances comprehension. Looking forward, the future scope involves algorithmic refinements, dynamic resource models, integration with machine learning, validation across industries, and robustness to noisy data, ensuring the continued evolution and applicability of the Modified Hungarian algorithm in solving complex assignment problems across diverse and dynamic contexts.

Acknowledgment

The authors would like to thank the Integral University, Lucknow, India, for providing the manuscript number IU/R&D/2024-MCN0002684 to the present work.

References

[1] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2): 182-197.

[2] Coello, C. A. C. Evolutionary algorithms for solving multi-objective problems. springer.com, 2007.

[3] Li, J., Sun, K., Xu, D., & Li, H. (2010). Single machine due date assignment scheduling problem with customer service level in fuzzy environment. *Applied Soft Computing*, 10(3): 849-858.

[4] Zhou, M., Chen, Q., & Cai, Y. L. (2013). Optimizing the industrial structure of a watershed in association with economic–environmental consideration: an inexact fuzzy multi-objective programming model. *Journal of cleaner production*, 42: 116-131.

[5] Liu, M., Xu, C., Luo, Y., Xu, C., Wen, Y., & Tao, D. (2017). Cost-sensitive feature selection by optimizing F-measures. *IEEE Transactions on Image Processing*, 27(3): 1323-1335.

[6] Talbi, E. G. Metaheuristics: from design to implementation. John Wiley & Sons, 2009.

[7] Puchinger, J., Raidl, G. R., & Pferschy, U. (2010). The multidimensional knapsack problem: Structure and algorithms. *INFORMS Journal on Computing*, 22(2): 250-265.

[8] Min, Q., Lu, Y., Liu, Z., Su, C., & Wang, B. (2019). Machine learning based digital twin framework for production optimization in petrochemical industry. *International Journal of Information Management*, 49: 502-519.

[9] Wang, R. Q., Jiang, L., Wang, Y. D., & Roskilly, A. P. (2020). Energy saving technologies and mass-thermal network optimization for decarbonized iron and steel industry: A review. *Journal of Cleaner Production*, 274: 122997.

[10] Strayer, J. K., & Strayer, J. K. Transportation and Assignment Problems. Linear Programming and Its Applications, 140-184. Springer, New York, NY, 1989.

[11] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2): 83-97.

[12] Jonker, R., & Volgenant, T. (1986). Improving the Hungarian assignment algorithm. *Operations Research Letters*, 5(4): 171-175.

[13] Kumar, A. (2006). A modified method for solving the unbalanced assignment problems. *Applied mathematics and computation*, 176(1): 76-82.

[14] Rabbani, Q., Khan, A., & Quddoos, A. (2019). Modified Hungarian method for unbalanced assignment problem with multiple jobs. *Applied Mathematics and Computation*, 361: 493-498.

[15] Hsieh, C. H., & Chen, S. H. Graded mean representation distance of generalized fuzzy number. In Proceeding of sixth Conference on Fuzzy Theory and its Applications, 32: 1-5, 1998.

[16] Sen, N., Sahoo, L. and Bhunia, A.K. (2014) An application of integer linear programming problem in tea industry of barak valley of Assam, India under crisp and fuzzy environments', *Journal of Information and Computing Science*, 9(2): 132–140.

[17] Khan, M. A., Haq, A., & Ahmed, A. (2021). Multi-objective Model for Daily Diet Planning. *Reliability: Theory & Applications*, 16(1(61)): 89-97. https://doi.org/10.24412/1932-2321-2021-161-89-97