JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES
www.journalimcms.org

ISSN (Online) : 2454 -7190 Vol.-20, No.-8, August (2025) pp 112 - 129 ISSN (Print) 0973-8975

RENTAL COST REDUCTION IN TWO-STAGE HYBRID FSSP USING BB: A MATLAB-BASED COMPARISON WITH GA

Kanika Gupta[1], Deepak Gupta[2], Sonia Goel[3]

[1,2,3] M. M. Engineering College, Maharishi Markandeshwar (Deemed to be University), Mullana, Ambala, India.

[1]Department of Mathematics, DAV College for Girls, Yamuna Nagar, Haryana, India.

E-mail: [1]kani.ambala@gmail.com, [2]guptadeepak2003@yahoo.com, [3]sonia.mangla14@gmail.com

Corresponding Author: **Kanika Gupta**

https://doi.org/10.26782/jmcms.2025.08.00008

(Received: May 30, 2025; Revised: July 28, 2025; Accepted: August 09, 2025)

Abstract

The paper addresses the classical two-stage FSSP with a single machine in the second stage and equipotential machines in the first. The uniqueness of this problem arises from the fact that the machine at the second stage is rented, with the objective being to minimize the rental cost. Efficient scheduling of jobs is critical in such environments to optimize resource usage and reduce operational costs. A distinguishing feature of this study is the representation of processing times on both stages using trapezoidal fuzzy numbers, which better capture uncertainty and variability in processing times compared to deterministic values. This fuzzy representation aligns well with real-world scenarios where exact processing times are often unavailable or subject to fluctuations. This paper's primary contribution is the creation of an optimization algorithm that uses the branch and bound (B&B) approach to tackle the issue. By breaking the problem space down into smaller subproblems and utilizing bounds to exclude less likely solutions, the B&B technique methodically explores the solution space. This method minimizes the expense of renting the second-stage machine while guaranteeing the identification of the ideal timetable. The fuzzy nature of the problem adds complexity to the scheduling task, as it requires handling the fuzziness in processing times while maintaining optimality. To ensure the robustness of the algorithm, it is implemented in MATLAB and tested against a variety of job sequences and machine configurations, along with the comparison of results with GA.

Keywords: Idle time, Rental cost, Trapezoidal Fuzzy processing time, Utilization time.

I. Introduction

Scheduling theory is a branch of operations research and computer science that deals with the optimal allocation of resources over time to perform a collection of tasks. It entails making plans that specify when and how tasks should be completed to optimize specific parameters, including decreasing waiting times between jobs, maximizing resource use, or minimizing the total completion time. Industry scheduling is essential since it reduces expenses and delays while optimizing productivity and resource use. Companies may guarantee seamless production processes, timely completion of tasks, and fulfillment of client requests by effectively allocating tasks and resources. By minimizing downtime and bottlenecks, efficient scheduling raises output and quality standards all around. Improved inventory control is also made possible, which lowers the risk of overproduction and stockouts and gives you the adaptability to deal with unforeseen disruptions or shifts in demand. Robust scheduling procedures ultimately result in higher operational effectiveness, lower costs, and a more competitive market position. In the industrial sector, scheduling is used extensively and is essential for resource management and operations optimization. To guarantee continuous and effective output, it is used in manufacturing to schedule worker shifts, assign machine time, and arrange production processes. To offer prompt and efficient patient care, scheduling is crucial in the healthcare industry for organizing staff shifts, assigning operating room times, and managing patient appointments. In the realm of transportation, it entails organizing public transportation routes and schedules, cargo logistics, and airline operations to guarantee punctual delivery and effective vehicle utilization. Scheduling is essential to project management to assign tasks, control resources, and guarantee that projects are finished on time and within budget. In IT, scheduling controls data center operations, optimizes the use of computational resources, and guarantees the smooth operation of cloud services. All things considered, scheduling lowers expenses, increases operational effectiveness, and raises service standards in several sectors.

Flow shop scheduling theory is a subfield of scheduling that focuses on maximizing the order in which tasks are completed in a production setting where there is a required, one-way flow through several workstations or equipment. Every task in a flow shop goes through the same set of steps to achieve certain objectives, such as making the least amount of time possible overall (makespan), reducing overall lateness, or maximizing other performance indicators like throughput or machine utilization. Sequential processes, deterministic operations, single-route flow, and non-preemption are important features. Goals frequently center on throughput maximization, makespan minimization, total flow time, and total tardiness. Johnson's rule (X) for two-machine flow shops, B&B method (XII), heuristics like the NEH heuristic (XV) for more complicated circumstances, and metaheuristics like genetic algorithms (XI), and tabu search (III) for big, complex issues are examples of common methods and methodologies. Flow shop scheduling is a popular technique used in manufacturing, automotive, electronics, and food processing industries. It improves production efficiency by lowering production times, guaranteeing efficient use of resources, and increasing total productivity.

Equipotential machines, which are identical and interchangeable, simplify the initial stages of production by ensuring consistent performance and streamlining the scheduling process. This standardization helps reduce downtime and operational complexities, allowing for a more efficient production flow and ensuring that the resources used in the initial stages of production are fully optimized, as these machines perform identical functions and can be utilized to maintain a steady workflow. This prevents bottlenecks and inefficiencies in the production process. Meanwhile, the use of rented equipment for subsequent stages provides the necessary flexibility to scale production capabilities without the need for permanent investments. This model enables companies to respond to market changes and seasonal variations in demand more effectively, without the burden of long-term equipment commitments. This balanced approach ensures that production lines operate efficiently. Incorporation of rented equipment into the production process, businesses can minimize these capital costs, accessing advanced or specialized machinery without the associated upfront investment. Renting allows companies to allocate their financial resources more strategically, potentially directing funds towards other critical areas. During periods of high demand, additional rented machines can be brought in to increase production capacity. Conversely, during slower periods, the reliance on rented equipment can be reduced, allowing companies to focus on their core set of equipotential machines.

Processing times in manufacturing or service industries often depend on factors like operator skill, machine condition, raw material quality, or environmental conditions. These factors introduce variability. So, the processing times can be modeled using TFNs since they are expressed as a range instead of a single fixed value. This range gives uncertain data a more flexible and realistic depiction.

To contextualize the effectiveness of our methodology, a comparative analysis with genetic algorithms (GAs), a prominent heuristic technique known for its robustness in solving complex optimization problems, is also presented. GAs have been widely used in scheduling due to their ability to explore large solution spaces and adapt to various constraints. However, while GAs offer flexibility and can provide near-optimal solutions, they may not guarantee the optimality required for specific scheduling scenarios.

II. Literature Review

There are numerous methods given by many authors to solve FSSPs. The initial solution was given by Johnsons (X) in 1954, named as Johnson's algorithm. Another milestone was set by Lomniciki (XII) in 1965 by introducing the classical B&B method, which is the exact method used to find the solution to the problems. Ignall and Schrage (IX) also worked on the B&B method. The concept of GA was introduced by Holland (VIII) in 1992. This foundational work has led to the widespread application of GAs across various disciplines. These methods of solutions were further used with modifications according to the situations and requirements by the researchers. A hybrid GA combined with PSO was introduced by Tang et al. (XX). Umam et al. (XXII) devised an algorithm that is based on GA and tabu search for the makespan minimization. This approach effectively addressed the complexities inherent in scheduling by leveraging the strengths of both genetic algorithms and tabu search, leading to improved optimization outcomes. The paper by Tomazella and Nagano (XXI) provided a thorough examination of Branch-and-Bound algorithms specifically applied to flow shop scheduling problems. It traced the evolution of these algorithms from foundational works to contemporary advancements, offering insights into their application and effectiveness. Many comparative studies by different authors show how GA is very easy to handle and gives reliable results for the fssp. Shahsavari et al. (XVIII) demonstrated that a proposed multi-objective genetic algorithm, DIPGA, significantly enhances decision-making and achieves faster convergence and greater efficiency in solving complex flow shop scheduling problems. Compared to other popular metaheuristics such as Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO), DIPGA shows superior performance, reinforcing the effectiveness of GA-based approaches in multi-objective scheduling environments. Wang et al. (XXIII)addressed the wide usage of GA and integrated a hypothesis-testing method into a GA to reduce premature convergence and enhance population diversity. The proposed approach maintained GA's global search ability while filtering statistically similar solutions. An Improved Genetic Algorithm (IGA) using multi-crossover, multi-mutation, and hypermutation strategies has been proposed by Rajkumar and Shahabudeen (XVI) for the permutation fssp. Results on OR-Library benchmarks show that the IGA yields better makespan performance compared to earlier approaches.

Many authors have given different heuristics to minimize rental cost under different environments. Narain (XIV) gave an algorithm to optimize the renting time for a 3-machine problem. Gupta et al.(V) also worked with the problems related to rental cost. An algorithm for a three-stage problem with fuzzy processing times to minimize the makespan and rental cost was given by Alburaikan et al. (I) using fuzzy arithmetic. Sathish and Ganesan (XVII) also devised a heuristic for minimizing rental cost under a fuzzy environment without converting the fuzzy processing time to classical numbers. Another method to solve the problems related to rental cost was given by Alharbi and EL-Wahed Khalifa (II), which makes use of pentagonal fuzzy processing times. Singla et al. (XIX) also worked with the problems related to rental cost. El-Morsy et al. (IV) used the Pythagorean fuzzy numbers as the processing times and minimized the rental cost.

The issues related to equipotential machines were addressed by many researchers. Goel et al. (VI) (VII)worked on the two-stage and three-stage problems with equipotential machines using the classical B&B method. Three-stage problem was also discussed by Malhotra and Goel (XIII). In the present paper, the work by Goel et al. and Malhotra et al. is extended using the concept of rental cost.

III. Practical situations

The present model of equipotential machines at the first stage and a single rented machine at the second stage with TFN processing times is applicable at many sites. The model that integrates equipotential machines with rented resources plays a crucial role in modern industrial operations by providing significant benefits in flexibility, cost management, and efficiency. This approach is essential for businesses that face fluctuating production demands and need to adapt quickly to changing market conditions. Some of the instances where the present model can be applicable can be a pharmaceutical industry with equipotential machines to handle the capsule filling or the tablet pressing at first stage and rented high-precision coating or sterilization machine for second stage or in the agriculture equipotential machines for the cleaning and sorting of grains and the rented milling or packaging machine to process the final product or in the metal fabrication case the equipotential machine for the bending or cutting of sheets and the rented machine for polishing the final product.

- **Integration with SAP ERP and SAP ME :**

The proposed scheduling algorithm demonstrates strong potential for integration within industrial systems powered by SAP ERP and SAP ME—platforms widely used for enterprise resource planning and manufacturing operations control. In this architecture, SAP ERP is responsible for higher-level functions such as material planning, order creation, procurement, and capacity analysis, while SAP ME handles real-time shop floor execution, including work order dispatching, machine utilization, and production tracking.

By integrating the algorithm into this ecosystem, production schedules generated in ERP can be dynamically optimized and transferred to MES for execution. The algorithm can process real-time data—such as job priorities, machine availability, and rental cost variations—to make immediate decisions. For example, if a bottleneck arises or a delay is detected, the algorithm can suggest the rental of additional machines at Stage II and reschedule jobs accordingly, minimizing cost and delay. These decisions can then be fed back into ERP for financial and logistical updates, creating a closed-loop feedback system.

In real-time scheduling environments, especially in high-mix low-volume industries, machine rentals may need to be initiated or extended on the fly to meet deadlines or respond to disruptions. The algorithm supports such scenarios by continuously evaluating job progress, machine utilization, and cost trade-offs to recommend the most cost-effective rental strategy—thus supporting adaptive scheduling and on-demand resource allocation within an intelligent, responsive ERP-MES framework.

IV. Problem Formulation

In this research, we present an advanced methodology for solving flow shop scheduling problems that incorporates equipotential machines at the initial stage and a rented machine at the second stage. This approach extends the classical B&B method by integrating rental policies, which adds a layer of flexibility and adaptability to the scheduling process. The primary objective of our proposed methodology is to enhance solution efficiency and quality compared to the standard BB approach, particularly in scenarios where resource allocation involves rented equipment. The use of trapezoidal fuzzy numbers (TFN) to represent the processing time makes the model more robust to variations. This robustness is beneficial in industrial applications, where minor fluctuations in processing times can impact the total elapsed time and costs. Additionally, some presumptions must be taken into account.

Assumptions

- All first-stage equipotential devices have varying usage costs and are available at time zero.
- All tasks must be completed on the stage one machine first, followed by the stage two machine.
- It is not necessary to process jobs on every possible machine in stage one.
- Machines should be rented when needed and returned as soon as the task is finished.

The data is represented mathematically in Table 1.

Notations

f_{ij}=Operational Cost on j th machine for i th job {(i = 1, 2, ..., s); (j = 1, 2, ..., r)}

($\kappa_{i1},\kappa_{i2},\kappa_{i3},\kappa_{i4}$)= Processing time on machine P for job i

($\zeta_{i1},\zeta_{i2},\zeta_{i3},\zeta_{i4}$)= Processing time on machine Q for job i

V. Proposed Methodology

- **Step 1**: Defuzzify trapezoidal fuzzy numbers using the COG method as

$$p_i = \frac{\kappa_{i1}+2\kappa_{i2}+2\kappa_{i3}+\kappa_{i4}}{6}, q_i = \frac{\zeta_{i1}+2\zeta_{i2}+2\zeta_{i3}+\zeta_{i4}}{6} \quad (1)$$

• **Step 2**: Determining the machine P_j's optimal processing time using the Modified Distribution approach to save costs. The requirement must be met to use the MODI technique is:

$$\sum_{i=1}^{s} p_i = \sum_{j=1}^{r} \eta_j \quad (2)$$

Table 1: Mathematical representation of Data

Job/Machine	P						P.T. on P	P.T. on Q
	P_1	P_2	.	.	.	P_r		
1	f_{11}	f_{12}	.	.	.	f_{1r}	($\kappa_{11},\kappa_{12},\kappa_{13},\kappa_{14}$)	($\zeta_{11},\zeta_{12},\zeta_{13},\zeta_{14}$)
2	f_{21}	f_{22}	.	.	.	f_{2r}	($\kappa_{21},\kappa_{22},\kappa_{23},\kappa_{24}$)	($\zeta_{21},\zeta_{22},\zeta_{23},\zeta_{24}$)
3	f_{31}	f_{32}	.	.	.	f_{3r}	($\kappa_{31},\kappa_{32},\kappa_{33},\kappa_{34}$)	($\zeta_{31},\zeta_{32},\zeta_{33},\zeta_{34}$)
.	.	.				.	.	.
.	.	.				.	.	.
.	.	.				.	.	.
S	f_{s1}	f_{s2}	.	.	.	f_{sr}	($\kappa_{s1},\kappa_{s2},\kappa_{s3},\kappa_{s4}$)	($\zeta_{s1},\zeta_{s2},\zeta_{s3},\zeta_{s4}$)
	η_1	η_2	.	.	.	η_r		

• **Step 3**: Calculate

$$l_t = max\left\{\sum_{i=1}^{s} q_i + \max_{1\le j\le r} f_{tj}, \max_{1\le j\le r}\left\{\sum_{i=1}^{s} f_{ij}\right\} + \min_{r\neq t} q_r\right\} \quad (3)$$

for all the jobs t = 1, 2, ..., s. Then find the job t for which l_t is minimum. This job is to be processed first.

• **Step 4**: Considering that the work t has been processed, repeat step 2. Continue the procedure until the ideal order is obtained.

• **Step 5**: Prepare the table that represents the in-out times of jobs on the machines P_1, P_2, …, P_r and Q.

• **Step 6**: For job i on machine P_j, the in time is represented as $\overline{f_{ij}}$, while the out time is denoted as $\overline{\overline{f_{ij}}}$. On machine Q, the in time is represented as $\overline{q_i}$, and the out time is denoted as $\overline{\overline{q_i}}$.

$$Utilization\ time\ of\ machine\ Q = \bar{\bar{q}}_s - \overline{q_1} \quad (4)$$

$$Earliest\ Time = \breve{E} = \bar{\bar{q}}_s - \sum_{i=1}^{s} q_i \quad (5)$$

• **Step 7**: Now set $\overline{q_1}$= Ě and modify the in-out table accordingly.

• **Step 8**: From this modified in-out table, machine Q's utilization time can be determined and will undoubtedly be lower than the previous utilization time discovered in step 5.

$$\text{Rental Cost} = \text{Utilization Time} * \text{Rent} \quad (6)$$

V.i. Numerical Example:

An exemplary example is presented to give a good grasp of the stages involved in the suggested algorithm. This example shows how to apply the algorithm to a particular case by methodically illustrating each step of the algorithm. The problem in the depicted example consists of six parallel machines and ten jobs. Data is given in Table 2 in the format of Table 1.

Step 1: After defuzzification using the Center of Gravity (COG) method, the resulting data is summarized and presented in Table 3

Step 2: The condition for the MODI method is satisfied, and the table after using the MODI method is represented in Table 4.

Table 2: Numerical Illustration

Job/Machine	P						P.T. on P	P.T. on Q
	P_1	P_2	P_3	P_4	P_5	P_6		
1	12	18	11	15	11	16	(25, 25, 42, 49)	(3, 6, 6, 9)
2	15	15	9	11	16	17	(24, 25, 42, 42)	(6, 7, 7, 9)
3	16	17	20	16	12	15	(16, 25, 31, 39)	(5, 5, 8, 8)
4	18	16	12	13	19	13	(25, 25, 33, 40)	(5, 5, 7, 9)
5	20	19	15	19	20	17	(16, 21, 28, 37)	(4, 5, 8, 8)
6	17	21	19	14	22	10	(22, 32, 40, 54)	(5, 7, 8, 8)
7	13	24	14	10	18	19	(21, 26, 37, 51)	(5, 7, 7, 9)
8	10	14	13	12	16	13	(18, 30, 30, 41)	(5, 6, 6, 7)
9	17	16	10	18	17	11	(19, 31, 34, 46)	(1, 2, 8, 9)
10	19	11	8	14	9	15	(19, 34, 36, 47)	(4, 6, 6, 9)
	49.5	60.3	37.2	50.4	66.7	53.4		

Table 3: Table after using step 1

Job/Machine	P						P.T. on P	P.T. on Q
	P_1	P_2	P_3	P_4	P_5	P_6		
1	12	18	11	15	11	16	34.67	6.00
2	15	15	9	11	16	17	33.33	7.17
3	16	17	20	16	12	15	27.83	6.50

4	18	16	12	13	19	13	30.17	6.33
5	20	19	15	19	20	17	25.17	6.33
6	17	21	19	14	22	10	36.67	7.17
7	13	24	14	10	18	19	33.00	7.00
8	10	14	13	12	16	13	29.83	6.00
9	17	16	10	18	17	11	32.50	5.00
10	19	11	8	14	9	15	34.33	6.17
	49.5	60.3	37.2	50.4	66.7	53.4		

Table 4: Result After MODI Method

Job/Machine	P.T. on P						P.T. on Q
	P_1	P_2	P_3	P_4	P_5	P_6	
1	19.67	0	0	0	15	0	6.00
2	0	0	21.43	11.9	0	0	7.17
3	0	0	0	0	27.83	0	6.50
4	0	24.67	0	5.5	0	0	6.33
5	0	25.17	0	0	0	0	6.33
6	0	0	0	0	0	36.67	7.17
7	0	0	0	33	0	0	7.00
8	29.83	0	0	0	0	0	6.00
9	0	0	15.77	0	0	16.73	5.00
10	0	10.46	0	0	23.87	0	6.17

- **Step 3 and Step 4**: The results of Step 3 and 4 are represented using the tree in Fig. ***1*** and Fig. ***2***. The numbers given in the brackets show the weights of the respective sequence. ţ is used to represent the job 10.
- **Step 5**: The in-out table using the sequence obtained in step 4 is represented in Table 5**Table *5***.
- **Step 6**: Utilization Time using B&B= 66.81
- Earliest time = 19.87
- **Step 7**: The Modified in-out table is given in Table 6.
- **Step 8**: It is evident from Table 5 and Table 6 that the utilization of the time of machine Q is decreased by 3.14 hours using the present methodology. If we take the rent to be Rs. 100 per hour, then the total difference in rent comes out to be Rs. 314, as the rent using B&B becomes Rs. 6681, and using the present methodology, it becomes 6367.

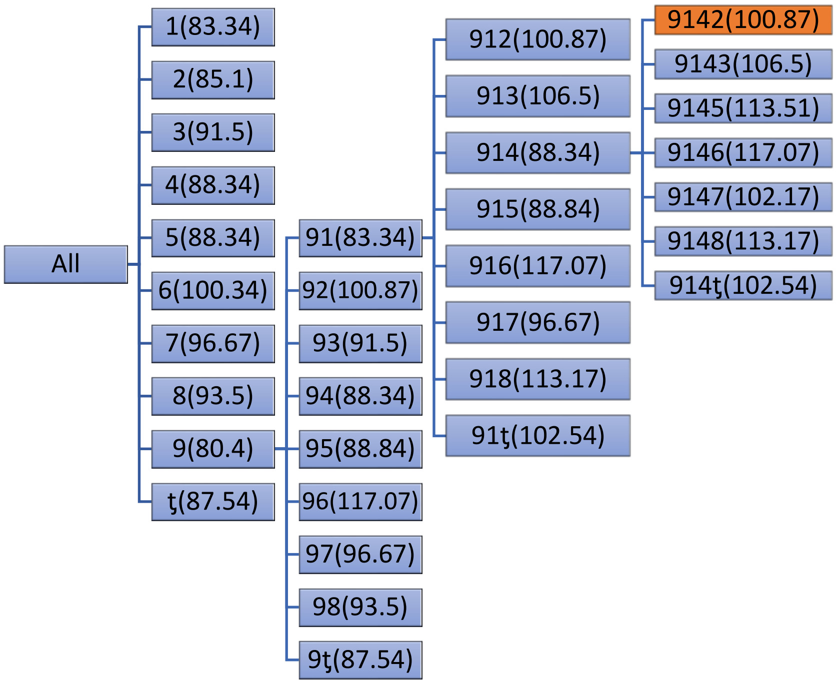

Fig. 1. Flow Chart of B&B Method

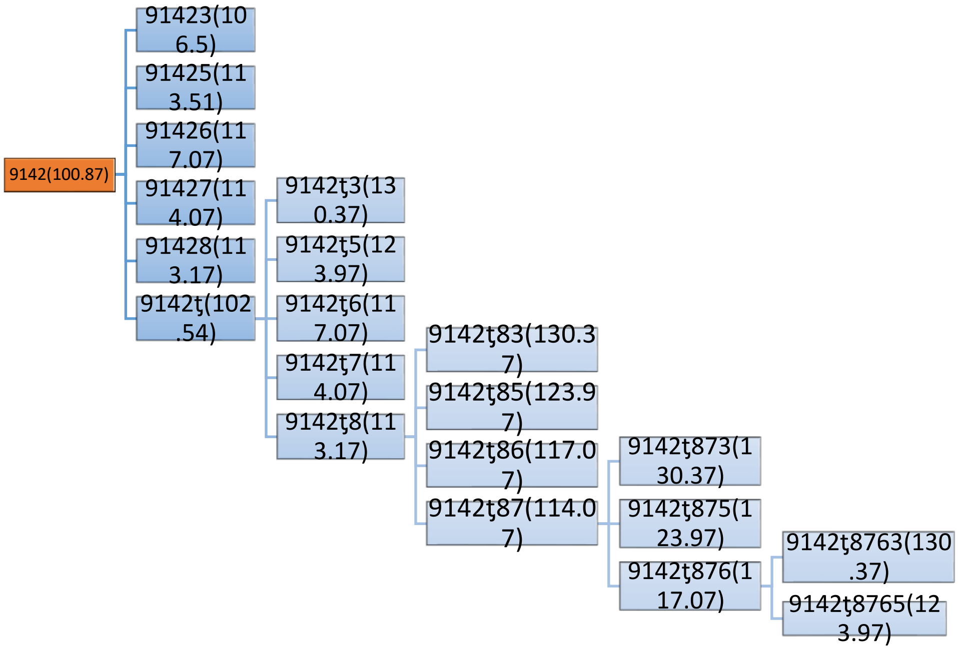

Fig. 2. Flow chart of B&B Method (Cont.)

Table 5: In-Out Table using step 4

Job/ Machine	P_1	P_2	P_3	P_4	P_5	P_6	Q
9	-	-	0-15.77	-	-	0-16.73	16.73-21.73
1	0-19.67	-	-	-	0-15	-	21.73-27.73
4	-	0-24.67	-	0-5.5	-	-	27.73-34.06
2	-	-	15.77-37.20	5.5-17.4	-	-	37.20-44.37
10	-	24.67-35.13	-	-	15-38.87	-	44.37-50.54
8	19.67-49.50	-	-	-	-	-	50.54-56.54

7	-	-	-	17.4-50.4	-	-	56.54-63.54
6	-	-	-	-	-	16.73-53.40	63.54-70.71
5	-	35.13-60.30	-	-	-	-	70.71-77.04
3	-	-	-	-	38.87-66.70	-	77.04-83.54

Table 6: Modified in-out Table

Job/ Machine	P_1	P_2	P_3	P_4	P_5	P_6	Q
9	-	-	0-15.77	-	-	0-16.73	19.87-24-87
1	0-19.67	-	-	-	0-15	-	24.87-30.87
4	-	0-24.67	-	0-5.5	-	-	30.87-37.20
2	-	-	15.77-37.20	5.5-17.4	-	-	37.20-44.37
10	-	24.67-35.13	-	-	15-38.87	-	44.37-50.54
8	19.67-49.50	-	-	-	-	-	50.54-56.54
7	-	-	-	17.4-50.4	-	-	56.54-63.54
6	-	-	-	-	-	16.73-53.40	63.54-70.71
5	-	35.13-60.30	-	-	-	-	70.71-77.04
3	-	-	-	-	38.87-66.70	-	77.04-83.54

V.ii. Evaluation of the problem using GA

Based on the ideas of genetics and natural selection, the Genetic Algorithm (GA) is an optimization method inspired by nature. It is used to find solutions to complicated problems by simulating biological evolution. Both linear and non-linear problems can be solved with GA because of its versatility and capacity to traverse wide search regions. Steps involved in GA are Selection, Crossover, and Mutation.

The pseudocode for GA is given below

Step 1: Initialize the population according to the size of the problem

Step 2: Evaluate the value of the fitness function

Step 3:Set the number of iterations as n

Repeat for n times

Step 4: Select the parent based on the fitness values of the population

Step 5: Apply the crossover operator to generate offsprings.

Step 6: Apply the mutation operator to the offspring.

Step 7: Check the fitness value of the sequence after mutation.

Step 8: If the fitness value after mutation is less than the fitness value of the parents, then set this sequence as new parent.

For the problem given in Table 2, suppose the population size is 5 and the number of iterations is 100.

Population 1: 1-4-5-2-3-8-7-10-6-9

Population 2: 2-6-5-3-1-8-10-9-7-4

Population 3: 3-5-7-1-4-10-6-2-9-8

Population 4: 7-9-10-1-4-2-6-3-8-5

Population 5: 9-6-7-4-2-5-1-3-8-10

The value of the fitness function for population 1 can be checked from Table 7, which is 101.18.

In the same way, the value of the fitness function for all the populations is listed in Table 8.

It is evident from Table 8 that the populations with the minimum fitness value are Population 2 and Population 4.

So, Population 2 and 4 are selected as parents, and the two-point crossover is applied on them to generate offsprings. The process is depicted in Fig. ***3***. The offsprings are

O1 : 2-6-5-1-3-8-10-9-7-4 O2 : 7-9-10-2-1-4-6-3-8-5

Next, the mutation is to be applied to O1 and O2. Here inversion mutation operator is used, and after the mutation, the new offspring are

O1 : 2-6-7-1-8-3-10-9-5-4 O2 : 7-9-8-2-4-1-6-3-10-5

The fitness value of O1 is 94.23, and that of O2 is 96.67. So, for the next generation, the parents selected are Parent 1 and Offspring 1. This way, after 100 iterations, the fitness value comes out to be 93.17, and the utilization time of machine Q comes out to be 71.74 hours, and thus the rental cost of machine Q becomes 7174, which is much higher than the rent using the present method.

Table 7: Fitness value for Population 1

Job/ Machine	P_1	P_2	P_3	P_4	P_5	P_6	Q
1	0-19.67	-	-	-	0-15	-	19.67-25.67
4	-	0-24.67	-	0-5.5	-	-	25.67-32
5	-	24.67-49.84	-	-	-	-	49.84-56.17
2	-	-	0-21.43	5.5-17.4	-	-	56.17-63.34
3	-	-	-	-	15-42.83	-	63.34-69.84
8	19.67-49.5	-	-	-	-	-	69.84-75.84
7	-	-	-	17.4-50.4	-	-	75.84-82.84
10	-	49.84-60.3	-	-	42.83-66.7	-	82.84-89.01
6	-	-	-	-	-	0-36.67	89.01-96.18
9	-	-	21.43-37.2	-	-	36.67-53.40	96.18-101.18

Table 8: Fitness Value

Population	Value of Fitness Function
Population 1	101.18
Population 2	93.17
Population 3	98.21
Population 4	96.97
Population 5	112.07

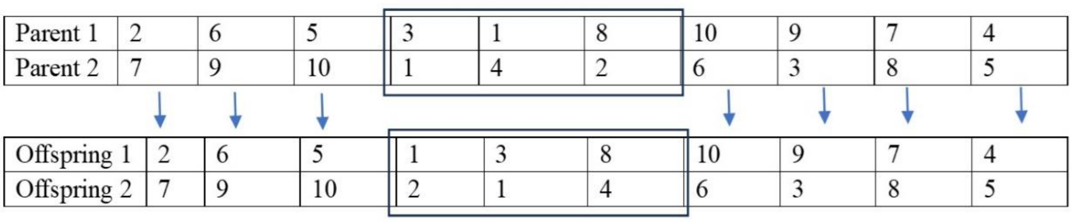

Fig. 3. Crossover Operator

VI. Computational results

The present method is tested with different combinations of the number of machines and the number of jobs, and the outcomes are listed in Table 9. The result using the GA is also compared with the results. For these experiments, MATLAB 2014(a) is used. Also, the comparative results are shown graphically in Fig. ***4*** and Fig. ***5***. The results demonstrate that the present method outperforms the GA in terms of minimizing rental costs.

Table 9: Comparison Table

No. of Jobs	No. of Machines	Rent using B&B	Rent using the present methodology	Rent using GA
20	20	316	292	354
20	60	366	285	406
20	100	385	291	450
20	40	430	311	499
20	80	484	316	554
40	80	746	623	786
40	40	674	585	757
40	60	1059	634	1055
40	20	1396	637	1424
40	100	1563	602	1633
60	60	954	848	1158
60	80	1236	853	1337
60	100	1395	877	1437
60	20	1593	892	1598
60	40	1662	901	1663
80	80	1219	1211	1229
80	60	1345	1167	1448
80	40	1886	1218	1893
80	100	1809	1163	1858
80	20	2392	1194	2417
100	40	1497	1480	1523
100	20	1543	1469	1546
100	60	2132	1444	2139
100	80	2305	1459	2308
100	100	2453	1490	2864

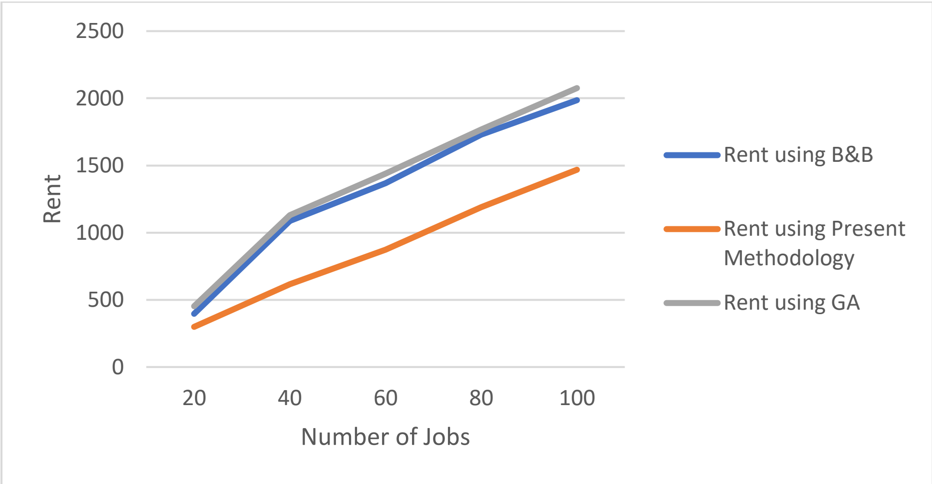

Fig. 4. Comparison of rent according to the number of Jobs

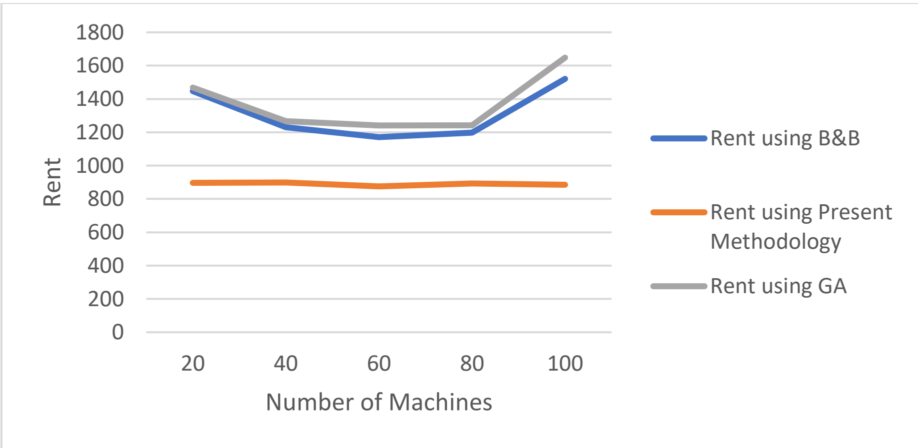

Fig. 5. Comparison of rent according to the number of machines

VII. Computational Complexity and Quality of Solution(Worst-Case Time Complexity Analysis)

The proposed methodology is designed to improve scheduling efficiency in a two-stage flow shop environment with equipotential machines at the first stage and a rented machine at the second stage. Let:

- n = number of jobs
- m = number of machines in the first stage

In the worst case, the algorithm must evaluate multiple sequences and allocate jobs across machines while optimizing rental cost and total elapsed time. Assuming the approach considers job sequencing followed by load balancing across machines, the time complexity can be estimated as:

- Job sequencing complexity: For a full enumeration of sequences, the number of permutations is O(n!).
- Machine allocation complexity: For each sequence, assigning jobs to m equipotential machines takes approximately O(n × m) time.
- Cost evaluation: Each evaluation is O(n).

Thus, the worst-case time complexity becomes approximately: O(n! × n × m)

However, since the proposed method uses a heuristic-guided branch and bound strategy, it prunes the search space significantly, reducing the actual runtime to polynomial time in practice for medium-sized instances. This is supported by the execution times reported in Table X, where even with 100 jobs and machines, the computational time remains within acceptable bounds.

While the proposed methodology consistently yields better rental cost outcomes compared to traditional methods like B&B and GA (as shown in Table 9), it is acknowledged that heuristic methods may not always guarantee optimality.

To assess the quality of the heuristic solution, the following approaches can be considered:

- RPD = $\frac{Heuristic\ solution - Best\ known\ solution}{Best\ known\ solution}\ X\ 100\%$

This metric can be computed for each test instance to quantify how close the heuristic comes to optimal.

- In all 25 cases, the proposed methodology has achieved the lowest rental cost, hence the RPD = 0% throughout.
- This indicates that the present method is consistently optimal or best-known among compared approaches, at least empirically.
- This strengthens the practical superiority and reliability of the proposed approach.

VIII. Conclusion

The study's conclusions demonstrate the effectiveness of the present methodology in minimizing rental costs for two-stage flow shop scheduling issues over the results of GA. The proposed algorithm offers a viable solution for applications where rental costs are a critical factor, and its implementation in MATLAB provides a flexible and accessible framework for further research and industrial applications.

IX. Future Work

The present study can be further extended by taking different performance measures like transportation time, weightage of jobs, setup times, job block or break down interval, etc., into consideration. The concept of equipotential machines can also be considered for both stages, and the number of stages can also be extended.

Conflict of Interest:

There was no relevant conflict of interest regarding this paper.

References

I. Alburaikan, Alhanouf, et al. “A Novel Approach for Minimizing Processing Times of Three-Stage Flow Shop Scheduling Problems under Fuzziness.” *Symmetry*, vol. 15, no. 1, Jan. 2023. 10.3390/sym15010130

II. Alharbi, Majed G., and Hamiden Abd El-Wahed Khalifa. “On a Flow-Shop Scheduling Problem with Fuzzy Pentagonal Processing Time.” *Journal of Mathematics*, vol. 2021, 2021. 10.1155/2021/6695174

III. Ben-Daya, Mohamed, and M. Al-Fawzan. “A Tabu Search Approach for the Flow Shop Scheduling Problem.” *European Journal of Operational Research*, vol. 109, no. 1, 1998, pp. 88–95. 10.1016/S0377-2217(97)00136-7

IV. El-Morsy, Salwa, et al. “On Employing Pythagorean Fuzzy Processing Time to Minimize Machine Rental Cost.” *AIMS Mathematics*, vol. 8, no. 7, 2023, pp. 17259–71. 10.3934/math.2023882.

V. Gupta, Deepak, et al. "3-stage specially structured flow shop scheduling to minimize the rental cost including transportation time, job weightage and job block criteria." *European Journal of Business and Management* 7.4 (2015): 1-7. https://www.iiste.org/Journals/index.php/EJBM/article/view/20232

VI. Gupta, Deepak, and Sonia Goel. “Three Stage Flow Shop Scheduling Model with Equipotential Machines.” *International Journal on Future Revolution in Computer Science & Communication Engineering IJFRCSCE*, 2018. https://www.ijrar.org/papers/IJRAR19J1166.pdf

VII. Gupta, Deepak, and Sonia Goel. “Two Stage Flow Shop Scheduling Model Including Transportation Time with Equipotential Machines at Every Stage.” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 12, Oct. 2019, pp. 5090–94. 10.35940/ijitee.L2740.1081219.

VIII. Holland, John H. “Genetic Algorithms.” *Scientific American*, vol. 267, no. 1, 1992, pp. 66–73. http://www.jstor.org/stable/24939139.

IX. Ignall, Edward, and Linus Schrage. “Application of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems.” *Operations Research*, vol. 13, no. 3, 1965, pp. 400–12. 10.1287/opre.13.3.400.

X. Johnson, S. M. “With Setup Times Included.” *Naval Research Logistics Quarterly*, vol. 1, 1954, pp. 61–68. **10.1002/nav.3800010110**

XI. Kahraman, Cengiz, et al. “An Application of Effective Genetic Algorithms for Solving Hybrid Flow Shop Scheduling Problems.” *International Journal of Computational Intelligence Systems*, vol. 1, no. 2, 2008, pp. 134–47. 10.2991/ijcis.2008.1.2.4

XII. Lomnicki, Z. A. “A ‘Branch-and-Bound’ Algorithm for the Exact Solution of the Three-Machine Scheduling Problem.” *Or*, vol. 16, no. 1, 1965, p. 89, 10.2307/3006687

XIII. Malhotra, Khushboo, and Sonia Goel. *Comparison of Bb With Meta-Heuristic Approach in Optimization of Three Stage Fss With Multiple Processors*. 2023, https://doi.org/10.21203/rs.3.rs-2822556/v1

XIV. Narain, Laxmi. "Optimize Renting Times of Machines in Flow-Shop Scheduling." *International Journal of Engineering and Applied Sciences*, vol. 2, no. 5, 2015, p. 257919. https://www.ijeas.org/download_data/IJEAS0205053.pdf

XV. Nawaz, Muhammad, et al. "A Heuristic Algorithm for the M-Machine, n-Job Flow-Shop Sequencing Problem." *Omega*, vol. 11, no. 1, 1983, pp. 91–95. 10.1016/0305-0483(83)90088-9

XVI. Rajkumar, R., and P. Shahabudeen. "An Improved Genetic Algorithm for the Flowshop Scheduling Problem." *International Journal of Production Research*, vol. 47, no. 1, Jan. 2009, pp. 233–49, 10.1080/00207540701523041.

XVII. Sathish, Shakeela, and K. Ganesan. "Flow Shop Scheduling Problem to Minimize the Rental Cost under Fuzzy Environment." *Journal of Natural Sciences Research*, vol. 2, no. 10, 2012, pp. 62–68. https://www.iiste.org/Journals/index.php/JNSR/article/view/3757/3806.

XVIII. Shahsavari-Pour, Nasser, et al. "A Novel Pareto-Optimal Algorithm for Flow Shop Scheduling Problem." *Mathematics 2024, Vol. 12, Page 2951*, vol. 12, no. 18, Sep. 2024, p. 2951. 10.3390/MATH12182951

XIX. Singla, Shakuntala, et al. "No Idle Flow Shop Scheduling Models with Separated Set-up Times and Concept of Job Weightage to Optimize Rental Cost of Machines." *Journal of Project Management (Canada)*, vol. 9, no. 2, Apr. 2024, pp. 101–08. 10.5267/j.jpm.2024.2.001

XX. Tang, Jianchao, et al. "Hybrid Genetic Algorithm for Flow Shop Scheduling Problem." *2010 International Conference on Intelligent Computation Technology and Automation*, vol. 2, 2010, pp. 449–52. 10.1109/ICICTA.2010.767

XXI. Tomazella, Caio Paziani, and Marcelo Seido Nagano. "A Comprehensive Review of Branch-and-Bound Algorithms: Guidelines and Directions for Further Research on the Flowshop Scheduling Problem." *Expert Systems with Applications*, vol. 158, Elsevier Ltd, 15 Nov. 2020. 10.1016/j.eswa.2020.113556

XXII. Umam, Moch Saiful, et al. "A Hybrid Genetic Algorithm and Tabu Search for Minimizing Makespan in Flow Shop Scheduling Problem." *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, Oct. 2022, pp. 7459–67, 10.1016/j.jksuci.2021.08.025

XXIII. Wang, L., et al. "A Class of Hypothesis-Test-Based Genetic Algorithms for Flow Shop Scheduling with Stochastic Processing Time." *International Journal of Advanced Manufacturing Technology*, vol. 25, no. 11–12, Jun. 2005, pp. 1157–63. 10.1007/S00170-003-1961-Y/METRICS.