Journal of Industrial Intelligence
https://www.acadlore.com/journals/JII

Development of a Hybrid Model for a Single-Machine Scheduling Using Expert Systems and Search Algorithms: A Simulation Study

Abdurrahman Zubia[1], Ibrahim Badi[1*], Mouhamed Bayane Bouraima[2]

[1] School of Engineering and Applied Sciences, Libyan Academy, 21851 Misrata, Libya
[2] Department of Civil Engineering, Sichuan College of Architectural Technology, 618000 Deyang, China

[*] Correspondence: Ibrahim Badi (i.badi@lam.edu.ly)

Received: 08-14-2024 **Revised:** 09-20-2024 **Accepted:** 09-26-2024

Citation: A. Zubia, I. Badi, and M. B. Bouraima, "Development of a hybrid model for a single-machine scheduling using expert systems and search algorithms: A simulation study," *J. Ind Intell.*, vol. 2, no. 3, pp. 132–143, 2024. https://doi.org/10.56578/jii020301.

Abstract: Job scheduling for a single machine (JSSM) remains a core challenge in manufacturing and service operations, where optimal job sequencing is essential to minimize flow time, reduce delays, prioritize high-value tasks, and enhance overall system efficiency. This study addresses JSSM by developing a hybrid solution aimed at balancing multiple performance objectives and minimizing overall processing time. Eight established scheduling rules were examined through a comprehensive simulation based on randomly generated scenarios, each defined by three parameters: processing time, customer weight, and job due date. Performance was evaluated using six key metrics: flow time, total delay, number of delayed jobs, maximum delay, average delay of delayed jobs, and average weight of delayed jobs. A multi-criteria decision-making (MCDM) framework was applied to identify the most effective scheduling rule. This framework combines two approaches: the Analytic Hierarchy Process (AHP), used to assign relative importance to each criterion, and the Evaluation based on Distance from Average Solution (EDAS) method, applied to rank the scheduling rules. AHP weights were determined by surveying expert assessments, whose averaged responses formed a consensus on priority ranking. Results indicate that the Earliest Due Date (EDD) rule consistently outperformed other rules, likely due to the high weighting of delay-sensitive criteria within the AHP, which positions EDD favourably in scenarios demanding stringent adherence to deadlines. Following this initial rule-based scheduling phase, an optimization stage was introduced, involving four Tabu Search (TS) techniques: job swapping, block swapping, job insertion, and block insertion. The TS optimization yielded marked improvements, particularly in scenarios with high job volumes, significantly reducing delays and improving performance metrics across all criteria. The adaptability of this hybrid MCDM framework is highlighted as a primary contribution, with demonstrated potential for broader application. By adjusting weights, criteria, or search parameters, the proposed method can be tailored to diverse real-time scheduling challenges across different sectors. This integration of rule-based scheduling with metaheuristic search underscores the efficacy of hybrid approaches for complex scheduling problems.

Keywords: Hybrid model; Single-machine; Job scheduling rules; Multi-criteria decision-making; Tabu Search

1 Introduction

Job sequencing refers to the process of determining the most efficient order in which tasks should be processed at one or more workstations. Since workstations typically handle multiple tasks, effective sequencing is vital to minimizing costs associated with job delays and idle time at workstations [1]. Poor scheduling can lead to job congestion and long waiting queues, adding complexity and pressuring management to develop more effective scheduling solutions [2].

Job scheduling involves determining the optimal order for completing a set of jobs or orders on a single machine or a group of machines. The goal is to achieve the best possible outcome based on a specific objective [3]. By establishing an optimal sequence, it is possible to calculate start and end times for each job, along with key performance metrics for both jobs and machines. Efficient scheduling is essential for optimizing resource use, meeting customer requirements within designated timeframes, and reducing inventory levels [4].

In this context, a variety of priority sequencing rules or heuristics have been used to determine the sequence in which jobs can be processed at workstations. These rules guide decision-making regarding the allocation of workstations for further processing. The use of priority sequencing rules offers advantages, as they integrate current knowledge of operational conditions into scheduling processes [5]. The performance of a given sequence generated by a priority sequencing rule is assessed using key performance indicators, such as average job completion time, average number of jobs in the system, mean job tardiness, and the number of delayed jobs. Selecting the most appropriate sequencing rule for job processing presents a complex challenge, and no single rule can be universally regarded as the best option for all scenarios [6].

Thus, from a managerial decision-making perspective, a detailed method for selecting the optimal sequencing rule is essential. The chosen methodology should account for the problem's complexity by explicitly considering multiple criteria, leading to more informed and improved decisions. MCDM techniques address these requirements by structuring complex problems and facilitating the evaluation of multiple criteria simultaneously [7].

Despite the potential benefits of integrating MCDM techniques with sequencing rules, there has been limited research exploring this hybrid approach. Only a few studies have investigated the effectiveness of combining MCDM methods with priority sequencing rules [8]. This research gap presents an opportunity to examine the benefits of these hybrid methodologies for optimizing job sequencing across various contexts. By leveraging MCDM techniques, which allow for the explicit consideration of multiple criteria, and incorporating them into sequencing rule selection, researchers can enhance the decision-making process and improve the performance of job sequencing systems. Therefore, further exploration and empirical validation of the approach integrating MCDM with sequencing rules are necessary to advance understanding and practical applications in this area.

Classic scheduling rules have been offered to address scheduling problems. Some scenarios, however, frequently fail to perform well when numerous competing requirements are involved. As real-world scheduling problems get more complicated, hybrid approaches combining MCDM with metaheuristic algorithms like Genetic Algorithms (GA), Simulated Annealing (SA), and TS have grown in popularity [1]. These methods aim to integrate the qualities of two frameworks: MCDM techniques provide a systematic mechanism for evaluating and ranking, whereas metaheuristics methods like TS are used to refine and improve initial solutions.

Most previous studies have focused on finding solutions using traditional scheduling rules and deciding how to arrange these rules best, either manually or using mathematical models based on a single objective function by choosing one of the performance criteria. The main objective of this research is to create a more efficient framework for tackling single-machine scheduling problems by combining hybrid MCDM methodologies and meta-optimization methods. The goal is to handle the issues of balancing several conflicting performance criteria by ranking these rules based on their performance across six key criteria: total flow time, total delay, number of delayed jobs, maximum delay, average tardy, and average weight of tardy jobs. Another major objective is to employ the best ranking rule from the MCDM evaluation as the initial solution for the TS algorithm. To improve scheduling results, the study applies a variety of TS search strategies, including job swapping, block swapping, job insertion, and block insertion.

2 Methodology

Figure 1 illustrates the methodology adopted to achieve the objectives. Job parameters include processing time, customer importance, and due date. Calculations were made for eight different scheduling rules for 50 random JSSM scenarios. Hybrid MCDM was designed to analyze performance criteria, i.e., flow-time, total tardiness, maximum tardiness, number of tardy jobs, average tardiness, and average weight of tardy jobs. The best solution was then identified and used as the starting initial solution for the TS algorithm, which improves the solution in four methods. The resulted data was compared and discussed.

2.1 Scheduling Rules

The assumptions used in this research are as follows:

J : Number of jobs; $J = \{J_1, J_2, \ldots, J_n\}$

O_i : Number of operations; $i = 1$

R_i : Release date; $\{R_1, R_2, \ldots, R_n\} = 0$

S_i : Setup time; $\{S_1, S_2, \ldots, S_n\} = 0$

C_i : Cost; $\{C_1, C_2, \ldots, C_n\} = 0$

PT_i : Processing time; $PT_j = \{PT_1, PT_2, \ldots, PT_n\}$

DD_i : Due date; $DD_i = \{DD_1, DD_2, \ldots, DD_n\}$

W_i : Customer importance; $W_i = \{W_1, W_2, \ldots, W_n\}$

First-come, first-served (FCFS), sometimes known as first in, first out (FIFO), is the simplest work scheduling algorithm. It schedules jobs in the order they entered the ready queue. The average waiting time is not always the shortest, and throughput can be low because lengthy jobs might overwhelm the entire workload, forcing short jobs to wait for an extended period, and making it difficult for this system to fulfill deadlines.

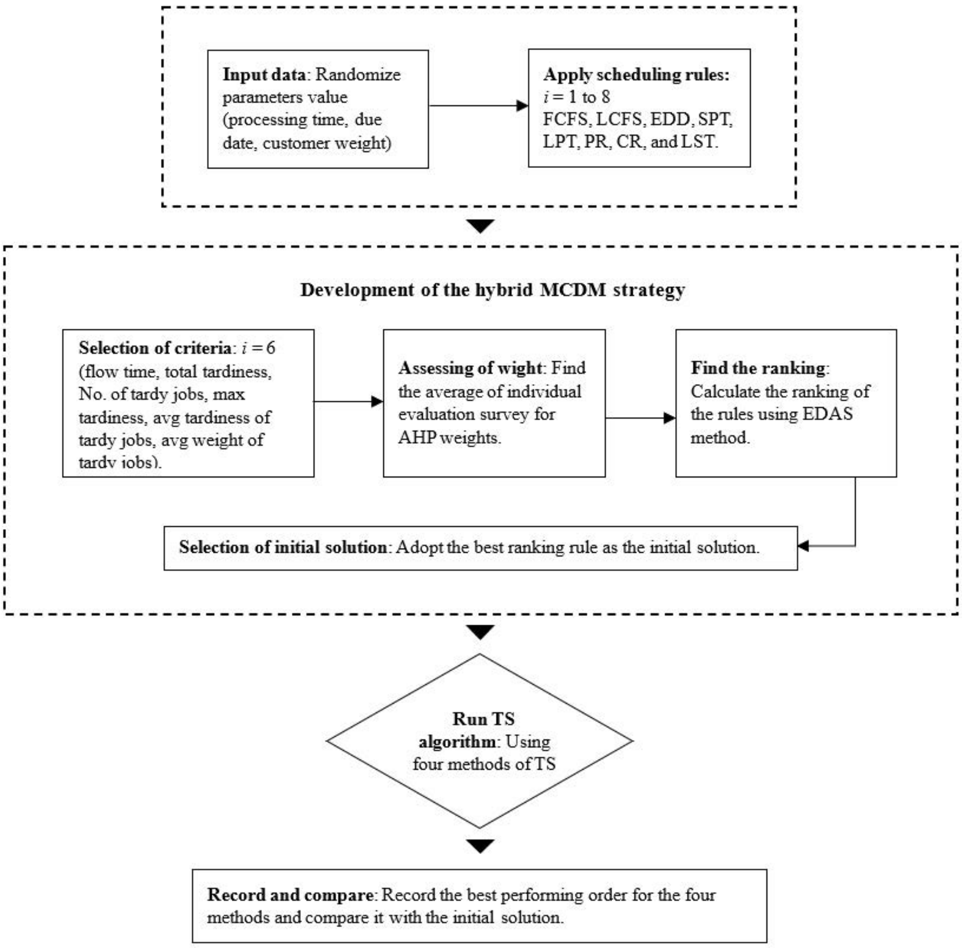

Figure 1. Development of the MCDM framework and JSSM simulation

Last-come, first-served (LCFS), also known as last in, first out (LIFO), reverses the previous rule by processing jobs in the opposite order they were received. This method can be irrational and has numerous disadvantages when it comes to deadlines.

Shortest Processing Time (SPT), also known as Shortest-Job-Next (SJN) or Shortest Job First (SJF), is a non-preemptive scheduling policy that requires work to be done based on the SPT. In other words, the jobs that have the shortest execution time are prioritized to start first. This rule is effective since it is simple and reduces the average length of time that each job must take to finish. However, if short jobs are introduced regularly, they may starve jobs that can take a long time to complete. Another drawback of using this algorithm in many industries is that the total execution time of a job must be known before execution.

Longest Processing Time (LPT), also known as Longest Job Next (LJN) or Longest Job First (LJF), is a scheduling system that arranges jobs so that each subsequent job has a shorter processing time than the previous one. When the machine is freed, the longest job ready at that time can begin working. This regulation increases the work in progress and may lead numerous short jobs to miss their deadlines.

EDD, earliest deadline first, or shortest deadline first, is a real-time scheduling method that prioritizes activities. When a scheduling event occurs (a job completes, a new job is released, etc.), the queue is searched for the process closest to the deadline, which becomes the next process scheduled for execution.

Least Slack Time (LST), also known as least slack first, is a dynamic priority scheduling technique. It prioritizes jobs depending on slack time, which is the amount of time remaining after completing a job if it begins now. It is mostly used in systems with several operations.

Priority scheduling: it is focused on prioritizing the jobs or the customers with the highest relevance or value and then moving down to the lowest. The disadvantage with this method is that it does not take into account delivery time, which can result in a substantial total delay.

Critical Ratio (CR): the CR scheduling rule determines the CR by dividing the total time remaining to the deadline by the total production time remaining. Priority is given to items in the production cycle with the smallest CR.

2.2 Mathematical Representation of Criteria

Flow time (FT) refers to the amount of time it takes for a job to be processed. This includes the time from when the job is entered until when it is done, which also means that it concerns all the jobs that were completed before this current job.

$$Flow\,Time_i = Processing\,Time_i + Flow\,Time_{i-1} \tag{1}$$

$$Total\,Flow\,Time = \sum_{i=1}^{n} Flow\,Time_i \tag{2}$$

Tardiness for each job is the difference in value between the completion time and due date, which is deemed zero if the value is negative. However, total tardiness (TT) is the aggregate of all jobs' tardiness.

$$T_i = \max\left(0, C_i - D_i\right) \tag{3}$$

where, T_i represents the tardiness of job i, C_i is the completion time of job i, and D_i is the due date of job i.

$$Total\,Tardiness = \sum_{i=1}^{n} T_i \tag{4}$$

Number of tardy jobs (NO. TJ) refers to the number of jobs that are late, which is the count of jobs whose completion time exceeds delivery time.

$$Number\,of\,Tardy\,Jobs = \sum_{i=1}^{n} I\left(T_i > 0\right) \tag{5}$$

Maximum tardiness (MAX T) refers to the maximum value of tardiness among all the tardy jobs.

$$T_{\max} = \max\left\{T_1, T_2, \ldots, T_n\right\} \tag{6}$$

Average tardiness (AVG T) describes the median delay of the tardy jobs.

$$Average\,Tardiness = \frac{\sum_{T_i>0} T_i}{\sum_{i=1}^{n} I\left(T_i > 0\right)} \tag{7}$$

Average weights of tardy jobs (AVG W) is the mean weight of the jobs that their completion time is bigger than the due date.

$$Average\,Weights\,of\,Tardy\,Jobs = \frac{\sum_{T_i>0} W_i}{\sum_{i=1}^{n} I\left(T_i > 0\right)} \tag{8}$$

2.3 Development of the Hybrid MCDM

MCDM is a strong analytical framework that helps in evaluating and prioritizing multiple, often conflicting, criteria in a decision-making process [9, 10]. In complex decision scenarios, where qualitative and quantitative factors are involved, MCDM provides a structured approach to the assessment of alternatives to optimize the decision outcome [11, 12]. MCDM uses techniques such as AHP in systematic analyses by the decision-makers, which include trade-offs among criteria [13, 14]. It is, therefore, an indispensable tool in decision-making within diverse fields, including engineering, management, and public policy, toward more informed and transparent decisions [15, 16].

This study employed two MCDM techniques: the AHP and the EDAS. The AHP was used to evaluate the six criteria and assign weights to them in order to estimate the relative importance of each criterion over another in the decision-making process. As for the EDAS, the distance between the criteria was utilized to bring the values of the criteria closer to each other, and the rules were then ranked based on this approximation.

2.3.1 AHP

The AHP is a mathematical and psychological strategy for organizing and analyzing complex decisions, as well as a precise approach to quantifying decision criteria weights. The analytic hierarchy approach was developed in the 1970s by Thomas L. Satty, who collaborated with Ernest Forman to create the Expert Choice program in 1983 [17]. Since then, it has been extensively researched and refined. The process is divided into three parts: the ultimate goal or problem to be solved; all feasible solutions, known as alternatives; and the criteria used to evaluate the alternatives.

The AHP provides a rational framework for reaching a desired decision by identifying its criteria and alternative options.

Specialists and decision-makers use a customized questionnaire to compare the relative importance of two factors at a time. The analytic hierarchy procedure translates these opinions into numerical values that can be compared to all possible criteria. This ability to measure distinguishes the AHP from other decision-making techniques. The final part of the process involves calculating numerical priorities for each of the different possibilities. These numbers represent the most desirable options, based on the values of all participants in the evaluation.

AHP is extremely useful for making decisions for complex, high-stakes problems. It stands out from other decision-making techniques because it measures criteria and options that are traditionally difficult to measure with numbers. Rather than describing the "right" decision, AHP assists decision-makers in finding the values that best fit their understanding of the problem. Involving all stakeholders is critical since various specialists can weight criteria differently. AHP also differs from classic surveys or questionnaires because it eliminates bias from the decision-making.

One of the most essential steps in the hierarchy is the binary matrix, which is often used to represent the preferences of criteria for one another and was utilized in this study to calculate the weights of criteria by comparing them to the others.

The steps are as follows:

Step 1: The criteria to be evaluated and their numbers were defined. The numbers represent the size of the binary matrix, which in this case was 6×6.

Step 2: The scale below was used to compare each pair of criteria, determining which is relatively more important than the other and to what extent it is preferred. The gray cells on the top are the inverse of the white cells at the bottom (Table 1).

Table 1. AHP binary matrix

Criteria	C_1	C_2	C_3	C_4	C_5	C_6
C_1	1	C_{21}	C_{31}	C_{41}	C_{51}	C_{61}
C_2	C_{12}	1	C_{32}	C_{42}	C_{52}	C_{62}
C_3	C_{13}	C_{23}	1	C_{43}	C_{53}	C_{63}
C_4	C_{14}	C_{24}	C_{34}	1	C_{54}	C_{64}
C_5	C_{15}	C_{25}	C_{35}	C_{45}	1	C_{65}
C_6	C_{16}	C_{26}	C_{36}	C_{46}	C_{56}	1

Step 3: The summation of factors was calculated for each criterion and the total of these values was found (Table 2).

$$S_i = \sum_{j=1}^{n} C_{ij} \tag{9}$$

$$S_{Total} = \sum_{i=1}^{n} S_i \tag{10}$$

Table 2. AHP binary matrix (summation)

Criteria	C_1	C_2	C_3	C_4	C_5	C_6	Summation
C_1	1	C_{21}	C_{31}	C_{41}	C_{51}	C_{61}	$S_1 = 1 + \{A_1 + A_2 + A_3 + A_4 + A_5\}$
C_2	C_{12}	1	C_{32}	C_{42}	C_{52}	C_{62}	$S_2 = 1 + \{B_1 + B_2 + B_3 + B_4 + B_5\}$
C_3	C_{13}	C_{23}	1	C_{43}	C_{53}	C_{63}	$S_3 = 1 + \{C_1 + C_2 + C_3 + C_4 + C_5\}$
C_4	C_{14}	C_{24}	C_{34}	1	C_{54}	C_{64}	$S_4 = 1 + \{D_1 + D_2 + D_3 + D_4 + D_5\}$
C_5	C_{15}	C_{25}	C_{35}	C_{45}	1	C_{65}	$S_5 = 1 + \{E_1 + E_2 + E_3 + E_4 + E_5\}$
C_6	C_{16}	C_{26}	C_{36}	C_{46}	C_{56}	1	$S_6 = 1 + \{F_1 + F_2 + F_3 + F_4 + F_5\}$
							S_{Total}

Step 4: The weights were determined by following the equation, as shown in Table 3:

$$W_i = \frac{S_i}{S_{Total}}, \text{ where } W_{Total} = 1 \tag{11}$$

Table 3. AHP binary matrix (weight determination)

Criteria	C_1	C_2	C_3	C_4	C_5	C_6	Summation	Weights
C_1	1	C_{21}	C_{31}	C_{41}	C_{51}	C_{61}	S_1	W_1
C_2	C_{12}	1	C_{32}	C_{42}	C_{52}	C_{62}	S_2	W_2
C_3	C_{13}	C_{23}	1	C_{43}	C_{53}	C_{63}	S_3	W_3
C_4	C_{14}	C_{24}	C_{34}	1	C_{54}	C_{64}	S_4	W_4
C_5	C_{15}	C_{25}	C_{35}	C_{45}	1	C_{65}	S_5	W_5
C_6	C_{16}	C_{26}	C_{36}	C_{46}	C_{56}	1	S_6	W_6
							S_{Total}	$W_{Total} = 1$

2.3.2 EDAS

The EDAS is an MCDM ranking mechanism that is particularly useful for evaluating and finding the rank of alternatives that have multiple weighted criteria [18].

The following steps were taken using the EDAS method [19]:

Step 1: The criteria and the alternatives were determined, which were eight rules and six criteria in this case.

Step 2: Each criterion was categorized as beneficial (where higher values desired) or non-beneficial (where lower values desired). In this case, they were all non-beneficial.

Step 3: A decision matrix was prepared and the average value of each criterion was calculated for the alternatives, as shown in Table 4.

$$AVG_j = \frac{\sum_{i=1}^{n} x_{ij}}{n} \tag{12}$$

where, i refers to criteria, and j refers to alternatives.

Table 4. Average of each criterion

Weightage	Criteria 1 $W_1\%$	Criteria 2 $W_2\%$	Criteria 3 $W_3\%$	Criteria 4 $W_4\%$	Criteria 5 $W_5\%$	Criteria 6 $W_6\%$
Alternative 1	X_{11}	X_{21}	X_{31}	X_{41}	X_{51}	X_{61}
Alternative 2	X_{12}	X_{22}	X_{32}	X_{42}	X_{52}	X_{62}
Alternative 3	X_{13}	X_{23}	X_{33}	X_{43}	X_{53}	X_{63}
Alternative 4	X_{14}	X_{24}	X_{34}	X_{44}	X_{54}	X_{64}
Alternative 5	X_{15}	X_{25}	X_{35}	X_{45}	X_{55}	X_{65}
Alternative 6	X_{16}	X_{26}	X_{36}	X_{46}	X_{56}	X_{66}
Alternative 7	X_{17}	X_{27}	X_{37}	X_{47}	X_{57}	X_{67}
Alternative 8	X_{18}	X_{28}	X_{38}	X_{48}	X_{58}	X_{68}
AVG_j	AVG_1	AVG_2	AVG_3	AVG_4	AVG_5	AVG_6

Step 4: Positive and negative distance from average determines how much better or worse the alternative is compared to the found average for each criterion. In other words, the positive distance is calculated by subtracting the value of alternative (n) at criterion (x) from the average of criterion (x), where n and x represent a specific alternative and criterion. If the difference is larger than zero, it is divided by the average value and multiplied by the weight percentage for this specific criterion (which was previously determined using AHP). If the difference is less than zero, the value is set equal to zero (Table 5).

If criterion is beneficial, then

$$PDA_{ij} = \frac{\max\left(0, (X_{ij} - AVG_j)\right)}{AVG_j} \tag{13}$$

If criterion is non-beneficial, then

$$PDA_{ij} = \frac{\max\left(0, (AVG_j - X_{ij})\right)}{AVG_j} \tag{14}$$

However, for negative distance, the average of criteria (x) is subtracted from the value of alternative (n) at criterion (x). If the difference is greater than zero, it is divided by the average value, which is then multiplied by the weight percentage for this specific criterion. Otherwise, the value is returned to zero (Table 6).

If criterion is beneficial, then

$$NDA_{ij} = \frac{\max\left(0, \left(AVG_j - X_{ij}\right)\right)}{AVG_j} \tag{15}$$

If criterion is non-beneficial, then

$$NDA_{ij} = \frac{\max\left(0, \left(X_{ij} - AVG_j\right)\right)}{AVG_j} \tag{16}$$

Table 5. Positive distance from average (PDA)

Weightage	**Criteria 1** $W_1\%$	**Criteria 2** $W_2\%$	**Criteria 3** $W_3\%$	**Criteria 4** $W_4\%$	**Criteria 5** $W_5\%$	**Criteria 6** $W_6\%$
Alternative 1	PDA_{11}	PDA_{21}	PDA_{31}	PDA_{41}	PDA_{51}	PDA_{61}
Alternative 2	PDA_{12}	PDA_{22}	PDA_{32}	PDA_{42}	PDA_{52}	PDA_{62}
Alternative 3	PDA_{13}	PDA_{23}	PDA_{33}	PDA_{43}	PDA_{53}	PDA_{63}
Alternative 4	PDA_{14}	PDA_{24}	PDA_{34}	PDA_{44}	PDA_{54}	PDA_{64}
Alternative 5	PDA_{15}	PDA_{25}	PDA_{35}	PDA_{45}	PDA_{55}	PDA_{65}
Alternative 6	PDA_{16}	PDA_{26}	PDA_{36}	PDA_{46}	PDA_{56}	PDA_{66}
Alternative 7	PDA_{17}	PDA_{27}	PDA_{37}	PDA_{47}	PDA_{57}	PDA_{67}
Alternative 8	PDA_{18}	PDA_{28}	PDA_{38}	PDA_{48}	PDA_{58}	PDA_{68}

Table 6. Negative distance from average (NDA)

Weightage	**Criteria 1** $W_1\%$	**Criteria 2** $W_2\%$	**Criteria 3** $W_3\%$	**Criteria 4** $W_4\%$	**Criteria 5** $W_5\%$	**Criteria 6** $W_6\%$
Alternative 1	NDA_{11}	NDA_{21}	NDA_{31}	NDA_{41}	NDA_{51}	NDA_{61}
Alternative 2	NDA_{12}	NDA_{22}	NDA_{32}	NDA_{42}	NDA_{52}	NDA_{62}
Alternative 3	NDA_{13}	NDA_{23}	NDA_{33}	NDA_{43}	NDA_{53}	NDA_{63}
Alternative 4	NDA_{14}	NDA_{24}	NDA_{34}	NDA_{44}	NDA_{54}	NDA_{64}
Alternative 5	NDA_{15}	NDA_{25}	NDA_{35}	NDA_{45}	NDA_{55}	NDA_{65}
Alternative 6	NDA_{16}	NDA_{26}	NDA_{36}	NDA_{46}	NDA_{56}	NDA_{66}
Alternative 7	NDA_{17}	NDA_{27}	NDA_{37}	NDA_{47}	NDA_{57}	NDA_{67}
Alternative 8	NDA_{18}	NDA_{28}	NDA_{38}	NDA_{48}	NDA_{58}	NDA_{68}

Weighted values were calculated by multiplying the weight percentage per each criterion by PDA_{ij} and NDA_{ij}, as shown in Tables 7 and 8.

$$WPDA_{ij} = W_j * PDA_{ij} \tag{17}$$

$$WNDA_{ij} = W_j * NDA_{ij} \tag{18}$$

The summation of $WPDA_{ij}$ of each alternative value was calculated as follows:

$$SP_i = \sum_{j=1}^{n} WPDA_{ij} \tag{19}$$

$$SN_i = \sum_{j=1}^{n} WNDA_{ij} \tag{20}$$

The highest values of SP_i and SN_i were extracted and normalized values of NSP_i and NSN_i were found:

$$NSPi = \frac{SP_i}{\mathrm{Max}_i\left(SP_i\right)} \tag{21}$$

$$NSNi = 1 - \frac{SN_i}{\mathrm{Max}_i\left(SN_i\right)} \tag{22}$$

Table 7. Weighted sum of PDA (WPDA)

Weightage	Criteria 1	Criteria 2	Criteria 3	Criteria 4	Criteria 5	Criteria 6	$\boldsymbol{SP_i}$
Alternative 1	$WPDA_{11}$	$WPDA_{21}$	$WPDA_{31}$	$WPDA_{41}$	$WPDA_{51}$	$WPDA_{61}$	SP_1
Alternative 2	$WPDA_{12}$	$WPDA_{22}$	$WPDA_{32}$	$WPDA_{42}$	$WPDA_{52}$	$WPDA_{62}$	SP_2
Alternative 3	$WPDA_{13}$	$WPDA_{23}$	$WPDA_{33}$	$WPDA_{43}$	$WPDA_{53}$	$WPDA_{63}$	SP_3
Alternative 4	$WPDA_{14}$	$WPDA_{24}$	$WPDA_{34}$	$WPDA_{44}$	$WPDA_{54}$	$WPDA_{64}$	SP_4
Alternative 5	$WPDA_{15}$	$WPDA_{25}$	$WPDA_{35}$	$WPDA_{45}$	$WPDA_{55}$	$WPDA_{65}$	SP_5
Alternative 6	$WPDA_{16}$	$WPDA_{26}$	$WPDA_{36}$	$WPDA_{46}$	$WPDA_{56}$	$WPDA_{66}$	SP_6
Alternative 7	$WPDA_{17}$	$WPDA_{27}$	$WPDA_{37}$	$WPDA_{47}$	$WPDA_{57}$	$WPDA_{67}$	SP_7
Alternative 8	$WPDA_{18}$	$WPDA_{28}$	$WPDA_{38}$	$WPDA_{48}$	$WPDA_{58}$	$WPDA_{68}$	SP_8

Table 8. Weighted sum of NDA (WNDA)

Weightage	Criteria 1	Criteria 2	Criteria 3	Criteria 4	Criteria 5	Criteria 6	$\boldsymbol{SN_i}$
Alternative 1	$WNDA_{11}$	$WNDA_{21}$	$WNDA_{31}$	$WNDA_{41}$	$WNDA_{51}$	$WNDA_{61}$	SN_1
Alternative 2	$WNDA_{12}$	$WNDA_{22}$	$WNDA_{32}$	$WNDA_{42}$	$WNDA_{52}$	$WNDA_{62}$	SN_2
Alternative 3	$WNDA_{13}$	$WNDA_{23}$	$WNDA_{33}$	$WNDA_{43}$	$WNDA_{53}$	$WNDA_{63}$	SN_3
Alternative 4	$WNDA_{14}$	$WNDA_{24}$	$WNDA_{34}$	$WNDA_{44}$	$WNDA_{54}$	$WNDA_{64}$	SN_4
Alternative 5	$WNDA_{15}$	$WNDA_{25}$	$WNDA_{35}$	$WNDA_{45}$	$WNDA_{55}$	$WNDA_{65}$	SN_5
Alternative 6	$WNDA_{16}$	$WNDA_{26}$	$WNDA_{36}$	$WNDA_{46}$	$WNDA_{56}$	$WNDA_{66}$	SN_6
Alternative 7	$WNDA_{17}$	$WNDA_{27}$	$WNDA_{37}$	$WNDA_{47}$	$WNDA_{57}$	$WNDA_{67}$	SN_7
Alternative 8	$WNDA_{18}$	$WNDA_{28}$	$WNDA_{38}$	$WNDA_{48}$	$WNDA_{58}$	$WNDA_{68}$	SN_8

The values of NSP_i and NSN_i were normalized, as shown in Table 9.

$$ASi = \frac{1}{2} * (NSP_i + NSN_i) \tag{23}$$

Table 9. AS_i calculations

	$\boldsymbol{SP_i}$	$\boldsymbol{SN_i}$	$\boldsymbol{NSP_i}$	$\boldsymbol{NSN_i}$	$\boldsymbol{AS_i}$
Alternative 1	SP_1	SN_1	NSP_1	NSN_1	AS_1
Alternative 2	SP_2	SN_2	NSP_2	NSN_2	AS_3
Alternative 3	SP_3	SN_3	NSP_3	NSN_3	AS_4
Alternative 4	SP_4	SN_4	NSP_4	NSN_4	AS_5
Alternative 5	SP_5	SN_5	NSP_5	NSN_5	AS_6
Alternative 6	SP_6	SN_6	NSP_6	NSN_6	AS_7
Alternative 7	SP_7	SN_7	NSP_7	NSN_7	AS_8
Alternative 8	SP_8	SN_8	NSP_8	NSN_8	AS_9

Finally, the ranking was found by arranging alternatives in descending order based on AS_i values.

2.4 TS Algorithm

TS is a metaheuristic optimization local search method that solves combinatorial problems by first implementing an initial solution and then enhancing it by making minor adjustments (or "moves") to the present solution and examining surrounding alternatives. The quality of a neighborhood is important since it influences the possibility of a local search. The chosen TS method, Tabu list, and aspiration criteria influence the algorithm's decision on which neighborhood to explore and which moves to consider, allowing it to avoid redundant or low-quality solutions. TS enables the algorithm to accept non-improving moves temporarily to escape from local optima and explore new sections of the solution space. To do this, the search employs a short-term memory structure (Tabu list) that retains recently visited solutions or movements (or characteristics of those solutions) to keep track of forbidden moves and prevent the algorithm from returning to these solutions for a certain number of iterations. However, the Tabu list is not absolute; the aspiration criterion is a rule that permits certain moves, even those on the Tabu list, to be reconsidered under certain situations. The search constantly improves the solution by testing new moves, updating the Tabu list, and applying aspiration criteria, and it will stop when a preset condition is fulfilled, such as reaching a maximum number of iterations, a time restriction, or no further improvement is discovered after a particular number of iterations.

TS steps are as follows:

Step 1: Let S be the set of all potential solutions.

Step 2: An initial solution s_o was adopted, with $s_{\text{current}} = s_o$.

Step 3: The objective function $f(s)$ was defined to evaluate the quality of the solution s, aiming to minimize the solution and find S^*, with $f(s^*) \leq f(s)$.

Step 4: The neighborhood function $N(s_{\text{current}})$ was defined.

Step 5: Iteration k and $k_{\max}$ were defined.

Step 6: The best solution was initialized and found $s^* = s_o$.

Step 7: The Tabu list $T = \{s_1, s_2, s_3, \ldots, s_i\}$ was established, where s_i is the recently visited solution, and l is the tenure length.

Step 8: The objective function for each solution in the neighborhood $f(s')$ was evaluated, where s' indicates that it is Tabu and cannot be visited for k number of iterations.

Step 9: The best non-Tabu solution was selected from the neighborhood s_{best}.

Step 10: After setting $s_{\text{current}} = s_{\text{best}}$, the best-know solution can be updated if improvement is found $f(s_{\text{current}}) < f(s)$.

Step 11: After adding solutions to the Tabu list, the oldest entry can be removed if l exceeds the maximum length specified.

Step 12: The steps should be repeated until a stopping criterion is met, such as a maximum number of iterations, a time restriction, or no improvement in the range of the set number of iterations.

TS is often applied to problems such as scheduling, routing, and assignment. Below is a breakdown of what are called operators, techniques, or methods used in TS for solving job scheduling problems.

Job swapping: It refers to exchanging two jobs in the initial solution. For example, as for the jobs $\{J_1,J_2,J_3,J_4,J_5,J_6,\ldots,J_n\}$ in a series, exchanging jobs would swap the positions of two jobs, such as switching J_2 and J_3, so that the schedule becomes $J_1, J_3, J_2, J_4, J_5, J_6, \ldots, J_n$. This method is used to investigate adjacent solutions by determining whether changing the placements of two functions improves the objective function which is the scheduling performance (by minimizing the six criteria).

Block swapping: It is similar to job swapping, except that it entails swapping a block that contains more than one job with another block. For example, as for jobs $\{J_1, J_2, J_3, J_4, J_5, J_6, \ldots, J_n\}$ in a sequence, swapping block $\{J_5, J_6\}$ with block $\{J_1, J_2\}$ may improve the solution, making the solution $\{J_5, J_6, J_3, J_4, J_1, J_2, \ldots, J_n\}$. Blocks can have more than two jobs, and this TS method is effective when tiny modifications do not result in considerable improvements.

Job insertion: It involves removing a job from its current position and inserting it in a different position in the sequence. For example, for jobs $\{J_1, J_2, J_3, J_4, J_5, J_6, \ldots, J_n\}$, removing J_3 from its current position and inserting it after J_6 in the sequence makes the solution $\{J_1, J_2, J_4, J_5, J_6, J_3, \ldots, J_n\}$. This procedure offers more comprehensive schedule alterations by moving one job to a more optimal position.

Block insertion: This method is similar to inserting jobs, but entire blocks are removed and inserted at different locations within the sequence. For jobs $\{J_1, J_2, J_3, J_4, J_5, J_6, \ldots, J_n\}$, removing $\{J_4, J_5\}$ from their current location and inserting it after J_1 in the sequence results in $\{J_1, J_4, J_5, J_2, J_3, J_6, \ldots, J_n\}$. This method is useful when a group of jobs has to stay together.

3 Development of the Hybrid MCDM and TS Model

Simulation study refers to the use of a computational model to simulate the behavior or the performance of a system under various parameters. The purpose of using a stochastic simulation (random) is to have different results due to the randomness of selecting different values of the parameters, which can lead to a better understanding of how the system behaves under these multiple scenarios and improve the processes.

To demonstrate the proposed steps, a large-scale project with a capacity of processing 50 jobs was considered. One machine must process these jobs in 30 different scenarios. A total of 1500 jobs that were generated by a simulation model select random numbers with pre-defined constraints and parameters, with job processing time from 1 to 20 days, delivery time from January 1, 2024, to March 30, 2025, and the weight of the job or customer value from a scale of 1 to 10.

3.1 AHP and EDAS

Five professionals and academics in engineering project management completed a questionnaire to assess the weights of the criteria. The average of the individuals' evaluations was then taken, producing the binary matrix in Table 10 below.

Given the weights, the EDAS method was used to rank the rules, and all 30 case studies concluded that the EDD scheduling rule was the best way to organize the 50 jobs.

Table 10. Results of the AHP binary matrix

Criteria	FT	TT	No. TJ	MAX T	AVG T	MAX W	Summation	Weights
FT	1.00	0.33	0.25	0.50	0.50	0.33	2.917	5.92%
TT	3.00	1.00	0.33	0.50	1.00	1.00	6.833	13.87%
No. TJ	4.00	3.00	1.00	0.33	0.50	0.33	9.167	18.61%
MAX T	2.00	2.00	3.00	1.00	0.50	0.33	8.833	17.94%
AVG T	2.00	1.00	2.00	2.00	1.00	0.50	8.500	17.26%
MAX W	3.00	1.00	3.00	3.00	2.00	1.00	13.000	26.40%
							49.250	1.0

3.2 TS Computational Model

The EDD scheduling rule was chosen as an initial solution to improve scheduling performance in all cases using the TS algorithm.

3.2.1 TS programming

The code was used to program TS via the Google Colab application. The steps are as follows:

Step 1: Input data was extracted, such as processing times, weights, due dates, and initial job sequence.

Step 2: A function was defined to evaluate the solution based on total flow time, tardiness, and number of tardy jobs.

Step 3: A function was defined to generate neighboring solutions by swapping, inserting jobs, or moving blocks of jobs.

Step 4: As for implementation of the TS algorithm, the best solution and Tabu list were initialized. For a fixed number of iterations, after generating neighbors, the best neighbor not in the Tabu list was evaluated and selected. After updating the best solution if the neighbor improves the result, the selected neighbor was added to the Tabu list, ensuring its size limit.

Step 5: The TS was executed using different methods (job swapping, block swapping, job insertion, block insertion).

Step 6: The results were output after the completion of the search process.

3.2.2 TS results

The above programming gives the results of TS scheduling with four Tabu methods presented previously (job swapping, block swapping, job insertion, and block insertion) for the 50 cases included in the input process. The program then compares the performance of the TS scheduling outcomes to the initial solution (EDD), which was chosen primarily because it outperformed the other scheduling rules.

The TS produced results for six performance criteria across all the simulated scenarios. Some methods outperformed others in the number of improved cases on some criteria, while others outperformed others. For one criterion, the initial solution remained the one with the highest number of cases with the best solution. This means that the solution did not improve on the Tabu.

4 Results and Discussion

JSSM was thoroughly simulated by creating 50 job determinants for 30 randomly chosen scenarios. Each work had a processing time, customer value, and a due date assigned. A separate table was utilized to schedule the jobs according to ranking rules. The rules were reviewed using the MCDM. The EDD scheduling rule proved to be the most effective in all cases. This is due to the fact that EDD is usually beneficial for reducing delays, while four of the six evaluation criteria are delay-related, with a combined weight of about 70% according to the binary evaluation matrix. Using Tabu, the model represented four Tabu approaches in comparison to the initial solution. The outcomes of each criterion improved. The values in Table 11 represent the number of cases, out of 30, in which method j has the best result for criterion i. The table shows that the initial solution using EDD produces 30 better results in terms of the maximum tardiness. It also outperforms Tabu methods in the number of cases that produce a better result in the average tardiness and the average weight of tardy jobs criteria by 15 and 11 cases, respectively. However, it was discovered that the job swapping method produces the highest score in terms of flow time by improving the solution for 20 cases, while the total tardiness and the number of tardy jobs rank second.

Block swapping improves the solution and achieves acceptable results in the average tardiness by improving the solution in nine cases, ranking second after the initial solution results, while the improvement in the solution is regarded as minor in the remaining five criteria. The job insertion method exceeds all other methods in terms of total tardiness and number of tardy jobs with 15 and 17 cases, respectively, and ranks second in terms of average weight of tardy jobs with seven cases. The block insertion method showed its highest improvement in the average weight of tardy jobs criterion with only four cases, ranking sixth overall.

Table 11. Comparison of the four methods with the initial solution

Solutions/ Performance	Total Flow Time	Total Tardiness	Number of Tardy Jobs	Maximum Tardiness	Average Tardiness	Average Weight of Tardy Jobs
Initial solution	0	0	0	30	15	11
Job swapping	20	14	8	0	3	6
Block swapping	5	0	3	0	9	2
Job insertion	4	15	17	0	1	7
Block insertion	1	1	2	0	2	4

5 Conclusions

In this study, hybrid MCDM methods based on AHP, weight assignment, and EDAS were created to evaluate the eight scheduling rules used for JSSM and select the optimal rule. The TS model worked in four ways to improve the solution and provided better solutions to the performance criteria. Tabu's results and the initial solution were compared to see how much the performance criteria were improved. The model provides flexibility in changing the proposed performance criteria (flow time, total tardiness, number of tardy jobs, maximum tardiness, average tardiness of tardy jobs, average weights of tardy jobs) and their weights based on the knowledge of experts to obtain new rankings for the rules. It also allows modification to the values of the three parameters (processing time, delivery time, job weight or importance) to generate additional case studies and thus varied results provide a better understanding and accuracy for the simulation study.

Author Contributions

Conceptualization, I.B.; methodology, A.Z.; validation, I.B. and M.B.B.; formal analysis, A.Z.; writing—original draft preparation, A.Z. and M.B.B.; writing—review and editing, I.B.; visualization, A.Z.; project administration, I.B. All authors have read and agreed to the published version of the manuscript.

Data Availability

The data used to support the research findings are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] I. Badi, M. B. Bouraima, Y. Qiu, and Ž. Stević, "Optimizing priority sequencing rules in parallel machine scheduling: An evaluation and selection approach using hybrid MCDM techniques," *Decis. Making Adv.*, vol. 2, no. 1, pp. 22–31, 2024. https://doi.org/10.31181/dma21202422

[2] P. Bari and P. Karande, "Optimal job scheduling to minimize total tardiness by dispatching rules and community evaluation chromosomes," *Decis. Making Appl. Manage. Eng.*, vol. 6, no. 2, pp. 201–250, 2023. https://doi.org/10.31181/dmame622023700

[3] J. N. D. Gupta, C. Wu, W. Lin, X. Zhang, D. Bai, B. M. T. Lin, and C. Liao, "Bicriteria single-machine scheduling with multiple job classes and customer orders," *Appl. Soft Comput.*, vol. 147, p. 110809, 2023. https://doi.org/10.1016/j.asoc.2023.110809

[4] X. He, Z. Wang, Y. Li, S. Khazhina, W. Du, J. Wang, and W. Wang, "Joint decision-making of parallel machine scheduling restricted in job-machine release time and preventive maintenance with remaining useful life constraints," *Reliab. Eng. Syst. Saf.*, vol. 222, p. 108429, 2022. https://doi.org/10.1016/j.ress.2022.108429

[5] S. Li and J. Yuan, "Single-machine scheduling with multi-agents to minimize total weighted late work," *J. Scheduling*, vol. 23, no. 4, pp. 497–512, 2020. https://doi.org/10.1007/s10951-020-00646-7

[6] X. Wang, L. Zhang, T. Lin, C. Zhao, K. Wang, and Z. Chen, "Solving job scheduling problems in a resource preemption environment with multi-agent reinforcement learning," *Rob. Comput. Integr. Manuf.*, vol. 77, p. 102324, 2022. https://doi.org/10.1016/j.rcim.2022.102324

[7] Ž. Stević, A. Bašić, S. Moslem, and K. Zhong, "An integrated ABC-FUCOM model for product classification," *Spectr. Eng. Manage. Sci.*, vol. 1, no. 1, pp. 83–91, 2023. https://doi.org/10.31181/sems1120239k

[8] V. L. A. Santos, T. F. M. Carvalho, L. P. de Assis, M. Weiss-Cohen, and F. G. Guimaraes, “Multi-objective Iterated Local Search based on decomposition for job scheduling problems with machine deterioration effect,” *Eng. Appl. Artif. Intell.*, vol. 112, p. 104826, 2022. https://doi.org/10.1016/j.engappai.2022.104826

[9] M. B. Bouraima, Y. Qiu, Ž. Stević, and V. Simić, “Assessment of alternative railway systems for sustainable transportation using an integrated IRN SWARA and IRN CoCoSo model,” *Socio Econ. Plann. Sci.*, vol. 86, p. 101475, 2023. https://doi.org/10.1016/j.seps.2022.101475

[10] Ž. Stević, N. Mujaković, A. Goli, and S. Moslem, “Selection of logistics distribution channels for final product delivery: Fucom-MARCOS model,” *J. Intell. Manage. Decis.*, vol. 2, no. 4, pp. 172–178, 2023. https://doi.org/10.56578/jimd020402

[11] M. B. Bouraima, A. Gore, E. Ayyildiz, S. Yalcin, I. Badi, C. K. Kiptum, and Y. Qiu, “Assessing of causes of accidents based on a novel integrated interval-valued Fermatean fuzzy methodology: Towards a sustainable construction site,” *Neural Comput. Appl.*, vol. 35, no. 29, pp. 21 725–21 750, 2023. https://doi.org/10.1007/s00521-023-08948-5

[12] O. Unsal, G. Demir, C. B. Karakus, and D. Pamučar, “Application of Z-number based fuzzy MCDM in solar power plant location selection problem in spatial planning,” *Energy Rep.*, vol. 12, pp. 4034–4054, 2024. https://doi.org/10.1016/j.egyr.2024.09.055

[13] A. Alosta, O. Elmansuri, and I. Badi, “Resolving a location selection problem by means of an integrated AHP-RAFSI approach,” *Reports Mech. Eng.*, vol. 2, no. 1, pp. 135–142, 2021. https://doi.org/10.31181/rme200102135a

[14] I. Badi, D. Pamučar, Ž. Stević, and L. Muhammad, “Wind farm site selection using BWM-AHP-MARCOS method: A case study of Libya,” *Sci. Afr.*, vol. 19, p. e01511, 2023. https://doi.org/10.1016/j.sciaf.2022.e01511

[15] M. B. Bouraima, Y. Qiu, B. Yusupov, and C. M. Ndjegwes, “A study on the development strategy of the railway transportation system in the West African Economic and Monetary Union (WAEMU) based on the SWOT/AHP technique,” *Sci. Afr.*, vol. 8, p. e00388, 2020. https://doi.org/10.1016/j.sciaf.2020.e00388

[16] R. Imran, K. Ullah, Z. Ali, and M. Akram, “A multi-criteria group decision-making approach for robot selection using interval-valued intuitionistic fuzzy information and Aczel-Alsina Bonferroni means,” *Spectr. Decis. Making Appl.*, vol. 1, no. 1, pp. 1–32, 2024. https://doi.org/10.31181/sdmap1120241

[17] M. Eshtaiwi, I. Badi, A. Abdulshahed, and T. E. Erkan, “Determination of key performance indicators for measuring airport success: A case study in Libya,” *J. Air Transp. Manage.*, vol. 68, pp. 28–34, 2018. https://doi.org/10.1016/j.jairtraman.2017.12.004

[18] D. D. Trung, “Application of EDAS, MARCOS, TOPSIS, MOORA and PIV methods for multi-criteria decision making in milling process,” *J. Mech. Eng.*, vol. 71, no. 2, pp. 69–84, 2021. https://doi.org/10.2478/scjme-2021-0019

[19] Ž. Stević, M. Vasiljević, A. Puška, I. Tanackov, R. Junevičius, and S. Vesković, “Evaluation of suppliers under uncertainty: A multiphase approach based on fuzzy AHP and fuzzy EDAS,” *Transport*, vol. 34, no. 1, pp. 52–66, 2019. https://doi.org/10.3846/transport.2019.7275