International Journal of Industrial Engineering, 30(3), 728-749, 2023

AN MCDM-BASED APPROACH TO COMPARE THE PERFORMANCE OF HEURISTIC TECHNIQUES FOR PERMUTATION FLOW-SHOP SCHEDULING PROBLEMS

Anas Ahmad Makki[1, *], Ammar Yahya Alqahtani[2], and Reda Mohamed Said Abdulaal[2]

[1]Department of Industrial Engineering
Faculty of Engineering—Rabigh, King Abdulaziz University
Jeddah, Saudi Arabia
[*]Corresponding author's e-mail: nhmakki@kau.edu.sa

[2]Department of Industrial Engineering
Faculty of Engineering, King Abdulaziz University
Jeddah, Saudi Arabia

In the industrial and manufacturing sectors, scheduling is an essential component in the process of determining crucial production cost aspects of corporate strategy. Solving flow-shop problems minimizes the makespan it takes for all jobs to be completed, reducing production costs and boosting output. Therefore, many heuristics techniques have been developed to assist in reaching a good and quick solution. However, newly developed techniques necessitate testing their performance against the classical ones. Therefore, this paper aims to conduct a comparative analytical, computational study of heuristic techniques for solving Permutation Flow-Shop Sequencing Problems and evaluating their performance. Eight techniques were compared by generating a set of problems of varying sizes and then solving them via a developed computer simulation program. Furthermore, a multi-criteria decision-making approach is followed for their performance evaluation. Results of the study revealed that based on six performance evaluation criteria, Dannenbring's technique is the first best, followed by the Slope Index technique as the second best, then the technique by Campbell, Dudek, and Smith, Hundal, the Time Deviation technique, Palmer, Gupta, and the technique by Jayasankari, Jayakumar, and Vijayaragavan, respectively. This paper puts forward a ranking of the developed techniques for flow-shop problems and a framework for the performance evaluation of new permutation flow-shop scheduling problem methods.

Keywords: Comparative; Computational; Heuristic; Multi-Criteria Decision-Making; Performance Evaluation; Permutation Flow-Shop Sequencing Problems.

(Received on November 3, 2022; Accepted on May 15, 2023)

1. INTRODUCTION

One of the most discussed issues in the field of Operations Research is the Permutation Flow-Shop Scheduling Problem (PFSP) for a reduced makespan time at a permutation flow factory (Fernandez-Viagas *et al*., 2017). In the PFSP, there are a number of machines and a number of jobs, each of which requires a number of operations. Because each machine is solely responsible for completing one job operation at a time, the jobs must all be completed in the same order as the used machines. Therefore, in the PFSP, machines must process jobs, which must be completed in a specific sequence and cannot be completed ahead of other jobs (Johnson, 1954). Thus, after knowing how long each machine will take to complete its assigned jobs, the goal is to find the ideal sequence of jobs that reduces the total makespan. Manufacturers profit financially from shorter makespans because they predict more sales volumes. However, solving the PFSP problems with more than two machines classify as a Nondeterministic Polynomial-time complete (NP-complete) problem (Garey *et al*., 1976a; Coffman, 1976; Rinnooy Kan, 1976a). Consequently, finding the best sequence of jobs in a reasonable time is usually out of the question. Scholars have presented several heuristic strategies to obtain approximate answers in practical contexts with limited time constraints (e.g., Komaki *et al*., 2019; Nawaz *et al*., 1983; Ruiz *et al*., 2005; Al Kattan and Maragoud, 2008; Muştu and Eren, 2018).

For any scheduling problem, the core of flow-shop scheduling is using the primary resource. Machines are often seen as primary resources in the scheduling issue since they are continuously used throughout the lifecycle of each job. Therefore, such scheduling is essential because of its capacity to minimize or eliminate the time spent with idle machines

DOI: 10.23055/ijietap.2023.30.3.8699 ISSN 1943-670X

(Missah 2015). However, secondary resources (such as raw materials, human resources, or equipment setup) might also be necessary to process jobs (Kempf *et al*., 1998). As an example of a secondary resource, servers are widely used in the manufacturing industry (Rahmouni Elidrissi *et al*., 2021). Secondary resources can be seen in a variety of contexts, including scheduling problems involving versatile machines and assembly components (Li *et al*., 2011), computer-controlled material handling systems (Kim and Lee, 2012), scheduling issues affecting the movement of biomass via trucks within the context of supply chain optimization (Torjai and Kruzslicz, 2016) and scheduling issues at a container terminal that affects loading and unloading containers from ships, storing containers in the terminal yard, and transporting containers utilizing a fleet of vehicles between ships and yard (Bish, 2003). Work arrangements between tasks or job families are the responsibility of servers, which are secondary resources. These servers may stand in for anything from a robot (Koulamas, 1996) to a person (Costa *et al*., 2020) to an autonomous car (Hall *et al*., 2000). This component has also been called a setup operator in the scholarly literature (see, e.g., Seeanner and Meyr, 2013; Modrák *et al*., 2012; Tempelmeier and Copil, 2016). In addition to the aforementioned practical implications, this scheduling issue is well-suited to the U-shaped manufacturing layout (Miltenburg, 2001), in which machines are distributed in a U-shape flow-shop layout, and human resources responsible for carrying out the changeovers are located in the center. Therefore, not considering the servers forms a Nondeterministic Polynomial-time hard (NP-hard) problem (Rinnooy Kan, 1976b).

An essential component of optimizing the time spent and resources utilized using a finite number of machines is to decide what jobs to do and in what sequence. Solving flow-shop problems minimizes the makespan or the time it takes for all jobs to be completed. The pursuit for better sequencing in workshop scheduling is motivated by the need to reduce production costs and boost output. Hence, many heuristics techniques have been developed to provide a good and quick solution. Eight heuristic methods include Palmer (1965), Gupta (1976), CDS (1970), Dannenbring (1977), and Hundal (1988), besides three other techniques, TD (2013), JJV (2021), and Abdulaal and Bafail (2021) are the focus. Therefore, a newly developed heuristic technique necessitates testing its performance against the classical ones previously reported in the literature. Thus, this paper aims to conduct a comparative analytical, computational study of heuristic techniques for solving PFSPs and evaluating their performance. Next, a literature review on PFSPs and developed techniques for their solving, a description of the heuristic approaches along with the newly proposed technique, the used materials and methods for achieving the objective of this study, analysis of results obtained from implementing the used methodology, a discussion and conclusions are provided in the remainder of this paper.

2. LITERATURE REVIEW

In the industrial and manufacturing sectors, scheduling is an essential component in the process of determining crucial aspects of corporate strategy. Determining how much work should be done at a given moment, where and when it should be done, and what resources should be used is an important step in the job distribution process (Brammer *et al*., 2022). The significance of making and sticking to schedules has skyrocketed in tandem with the global manufacturing spread. It is possible that a single machine, two machines, a network of machines, an open system, and other things might all be included in the job environment of a single machine. It is crucial to resolve these scheduling issues by assigning workloads to individual machines within the confines of a limited resource pool (Chen *et al*., 2009; Song and Lin, 2021). Criteria that pertain to efficiency include makespan, flow time, mean-flow time, waiting time, and idle time, amongst others. Criteria that pertain to cost include travel time, equipment maintenance, and labor charges, amongst others. Criteria that pertain to deadlines include lateness, tardiness, number of tardy jobs, etc. (Zaied *et al*., 2021). The objective of the PFSP is to ensure that a certain number of jobs (n) are finished in a predetermined order utilizing a set number of machines (m), where each machine performs precisely one operation on the jobs. In other words, the goal of the PFSP is to ensure that a certain number of jobs (n) are in the shortest total processing time (makespan). The PFSP has garnered the most attention because of the practical relevance and pervasiveness of the problem. The PFSP with the shortest makespan criterion has gained significant interest from academics and industry experts as a method for measuring the efficiency of production and service delivery (de Fátima Morais *et al*., 2022).

Previous studies have been concerned with solving the problem of scheduling large-scale job machines in flow shops based on the shortest makespan requirement for the last 50 years. Due to the exerted efforts, many heuristics and metaheuristic algorithms have been developed. Garey *et al*. (1976b) demonstrated that flow shop scheduling problems for a system with more than two machines and more than two jobs are difficult NP-complete problems. Johnson (1954) was the one who initially looked at the schedules for the two- and three-stage flow shops. Ignall and Schrage (1965) developed an m-machine system using a branch and bound approach to offer the shortest possible makespan. Page (1961) and Palmar (1965) recommended utilizing basic index-based heuristics to rank jobs in descending or ascending order with specified weights. This would allow for the most effective use of time and resources. Both Campbell *et al*. (1970) and Koulamas (1998) developed constructive heuristics for scheduling difficulties in a flow shop. They modeled their work after Johnson's two-machine method for scheduling problems. Gupta (1971) proposed a heuristic method that would be effective in overcoming the difficulties associated with scheduling large-scale flow shops that were complicated while also being applicable in real life. Bonney and Grundy (1976), Dannenbring (1977), and King and Spachis (1980) used the lowest makespan criterion to assess and analyze the effectiveness of different constructive algorithms. Stinson and Smith (1982), Nawaz *et al*. (1983), Taillard (1990), and Hundal and Rajgopal (1988), to mention just a few, are credited with having written some of the first academic works on the topic of makespan. It has been shown that the Nawaz, Enscore, and Ham (NEH) heuristic is the most effective way of resolving flow shop scheduling issues while maintaining the shortest possible makespan (Framinan *et al*., 2003). A comprehensive literature analysis on the difficulties of flow shop scheduling was provided by Reza Hejazi and Saghafian (2005), who used the makespan criterion. Ruiz and Maroto (2005) discovered that when compared to Taillard's standard, the NEH heuristic was the most effective of all the constructive heuristics.

In the NEH algorithm, sorting and reinsertion are two phases performed in succession. The first step is to construct a plan that may be implemented by prioritizing jobs according to the amount of time required to finish them. The second step involves picking out certain operations from the first sequence and rearranging them in a different order to cut the total make-time. According to the findings of Kalczynski and Kamburowski's (2007) study, the NEH heuristic suffers from a significant flaw. It was determined that the second phase's work scheduling relied too heavily on the shortest possible makespan. Chakraborty and Laha (2007) devised a heuristic strategy to reduce the time necessary to finish a make-in permutation flow shop scheduling. Dong *et al*. (2008) presented the NEHD (Nawaz-Enscore-Ham based on deviation) heuristic, which aims to effectively use the machine system by using a one-of-a-kind initial priority rule and an innovative method for tie-breaking. Kalczynski and Kamburowski (2008 and 2009) integrated NEH-KK1 and NEH-KK2 heuristics with the tie-breaking (TB) method based on Johnson's heuristic to schedule jobs in a system of machines to minimize makespan by providing weightage to the processing time. This integration and weightage were done to minimize the makespan in scheduled jobs. The innovative insertion strategies that were provided by Rad *et al*. (2009) perform much better than NEH when measured against the Taillard Benchmarks. This was found in a considerable proportion of the situations. Lin and Ying (2016) proposed a constructive heuristic method to solve the difficulties associated with makespan-related flow shop scheduling. This method defined a tie-breaking strategy based on a priority rule for the least amount of system idle time. Vasiljevic and Danilovic (2015) studied numerous strategies for dealing with ties in the NEH heuristic to address the makespan criterion for a PFSP. Liu *et al*. (2017) investigated the impact of the first four processing moments on the beginning job sequence. They proposed a new tie-breaking method for the NEH heuristic by decreasing the front delay time and the idle time before the tie position as a solution for the issue.

Arisha *et al*. (2001) are all places where various dispatching rules have been analyzed. They focused on a subset of the algorithms discussed in the overviews relevant to the issue classes that deal with flow shops or job shops to reduce the makespan. Hossain *et al*. (2014) conducted an investigation beginning with Palmer's to address a flow shop scheduling issue with four jobs and ten machines. The significant NP-hardness of the PFSP that reduces tardiness (Du and Leung, 1990; Amdouni *et al*., 2021) has led to their widespread usage of heuristic and metaheuristic approaches to solving them. In contrast, precise techniques are impracticable for medium and large examples (Sayadi *et al*., 2010; Gupta and Chauhan, 2015). Makespan reduction in flow shop scheduling issues with a non-machine resource is the focus of Laribi *et al*. (2016). An approximate solution to the n-job, m-machine flow shop issue with resource constraints may be swiftly generated by modifying the NEH heuristic to account for a second resource constraint. The Job-Shop Scheduling Problem is a more difficult simplification since the jobs vary for each primary component. Specific criteria for issue and solution formalizations are typically introduced in industrial applications (Fuchigami and Rangel, 2018). A good scheduling strategy may help industry professionals and event planners alleviate monetary worries (Cui *et al*., 2021).

As mentioned before, the permutation flow shop scheduling issue is one of the most active problems in the operation literature, with hundreds of publications tackling various versions and limitations in the classical difficulties in the past several years. There are several recent examples of solving the permutation flow shop (Wu *et al*., 2011; Alawad and Abed-alguni, 2022; Fathollahi-Fard *et al*., 2021; Fernandez-Viagas *et al*., 2022a and 2022b; Lee and Kim 2022; Morais *et al*., 2022; Doush *et al*., 2022; Rifai *et al*., 2021; Ribas *et al*., 2021; Meng *et al*., 2022; Sharma *et al*., 2022). The curious reader is directed to an in-depth analysis of the issue in various settings (Neufeld *et al*., 2022; Jayasankari, S. 2021).

In this paper, the well-known and widely used sequencing heuristic techniques for solving PFSP, including Palmer (1965), Gupta (1976), CDS (1970), Dannenbring (1977), and Hundal (1988), besides three other techniques, TD (2013), JJV (2021), and a new quick and effective computational heuristic approach proposed recently by Abdulaal and Bafail (2021) are the focus. Palmer, Gupta, CDS, Dannenbring, and Hundal approaches were chosen above the alternatives because they were founded on the same idea as the proposed technique for designing a slop index. On the other hand, the other two strategies (TD and JJV) were chosen to compare the suggested technique to those not based on the slope index. These techniques have been compared to test their performance in solving PFSPs. The methodology followed to achieve the objective and a description of the compared techniques are described subsequently.

3. MATERIALS AND METHODS

The general methodology flowchart presented in Figure 1 was followed to achieve the study's objective in three stages and two main analysis phases. In Stage 1, the eight PFSP techniques were determined for comparison after a literature review process. Subsequently, a computer simulation program was developed to generate 100,000 different PFSPs with different sizes. Solutions to the generated PFSPs using the selected eight techniques were the basis of comparison in two main analysis phases.

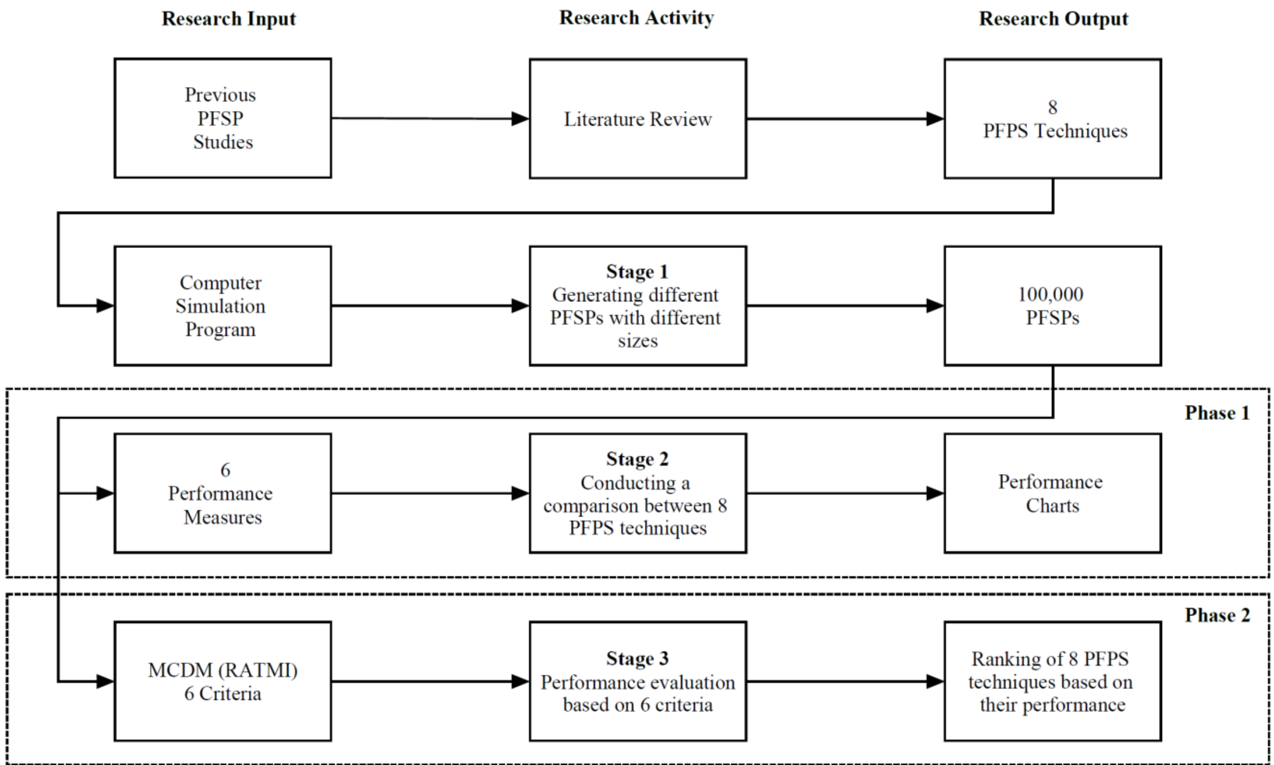


Figure 1. Methodology Flowchart

In the first phase, eight PFSP techniques were compared in Stage 2. The comparison was performed using several processing-time -based performance measures as the basis of the comparison. This is to test new techniques' validity and performance. The used measures include the total processing time (i.e., makespan), the percentage of improvement using the technique, the relative performance between the most recent proposed technique by Abdulaal and Bafail (2021) and the other seven techniques, the number of times the proposed technique was better than the other seven techniques, the percentage of errors using the technique and the number of best results obtained, and the execution time using the technique.

Subsequently, in the second phase, a performance evaluation was conducted in Stage 3 following a Multicriteria-Decision-Making (MCDM) approach using the six standard criteria for ranking the studied techniques based on their performance in solving the generated PFSPs. Those criteria included the total processing time, flow time, idle time on–jobs and –machines, machine utilization, and execution time. A description of each of the eight considered heuristic techniques in the comparisons, along with the two phases of the comparative analytical study and pertaining stages followed to achieve the objective, are all provided in the following subsections.

3.1 Eight Heuristic Techniques for Solving Permutation Flow-Shop Scheduling Problem (PFSP)

Concerns with flow shop scheduling and job sequencing have sparked intense interest in the field of Operations Research over the last several decades, leading to a seemingly endless stream of new approaches and refinements to existing ones. Maximum utilization of all available resources is essential in today's highly competitive global economy, where increased automation permeates almost every sector and area of each organization's operations. Time is the one resource that is certain to be present in every given situation. As a result, there is a pressing need for cutting-edge methods of scheduling and sequencing that can optimize a project from start to finish. Therefore, heuristic techniques for solving PFSPs were developed previously. The eight techniques listed in Table 1 are the focus of this comparative analytical, computational study are described subsequently.

Johnson (1954) was the first to examine the flow shop issue for 'n' jobs to be completed by two machines. As an objective function, the total completion time was equivalent to the job's full completion. According to Johnson's rule, work should come before another if and only if the time interval each machine must spend on the job is less than the time interval the other machine spends on the job. When the number of machines exceeds 2, the flow shop scheduling issue becomes NP-hard, hinting at its future complexity. The makespans must be larger than or equal to Palmer's bottom boundaries (1965) gave. The Palmer heuristic ranks jobs based on slope indices and schedules them in decreasing order. In its standard form, the slope index S_i is shown in equation (1).

Table 1. The Eight Heuristic Techniques for Solving Permutation Flow-Shop Scheduling Problems (PFSP)

h	Technique	Reference
1	Palmer	(Palmer, 1965)
2	CDS	(Campbell, Dudek, and Smith, 1970)
3	Gupta	(Gupta, 1971)
4	Dannenbring	(Dannenbring, 1977)
5	Hundal	(Hundal and Rajgopal, 1988)
6	Time Deviation (TD)	(Rao *et al.*, 2013)
7	Jayasankari, Jayakumar, and Vijayaragavan (JJV)	(Jayasankari *et al.*, 2021)
8	Slop Index (SI)	(Abdulaal and Bafail, 2021)

$$Si = j=1MM+2j+1\ tij \text{ for } i = 1,2, \ldots ,N, \tag{1}$$

where t_{ij} is the processing time of a job i on a machine j.

The second heuristic technique, the CDS heuristic, was created by Campbell, Dudek, and Smith (1970), who used Johnson's method as a guide. By partitioning the flow shop issue into two sets of M machines, the inventors of this heuristic can solve M-1 two-machine problems and pick the optimal schedule for each set. For the k^{th} reduced problems, g=1 or g=2 and k=1, …, M-1, the processing durations P^k_{ig} of the i^{th} job on the g^{th} machine group are shown in equation (2).

$$P^k_{i1} = \sum_j^k t_{ij} \text{ and } P^k_{i2} = \sum_j^k t_{i,m-j+1} \tag{2}$$

For the third heuristic technique, if 'n' jobs are to be completed on 'm' machines, and the workflow is unidirectional, then Gupta (1971) devised a heuristic to handle this issue. Every job and machine must follow the same technical hierarchy for this to be possible. Since machine numbers are assigned randomly, they might be chosen to represent the desired outcome—for example, jobs begin on machine 1, then go on to machines 2 and 3, etc., until they finally reach the 'mth' machine. Gupta (1976) gave each position an index and arranged them in increasing order. Equation (3) shows his summary of the index.

$$f_i = \frac{A}{(t_{im} + t_{im+1})} \text{ where } A = \begin{cases} 1 & if\ t_{im} < t_{im+1} \\ -1 & otherwise \end{cases} \tag{3}$$

The fourth heuristic technique is Rapid Access (RA), a heuristic approach first proposed by Dannenbring (1977). Dannenbring looked at the combined use of CDS heuristic approaches and Palmer's slope index. Using Palmer's slope index as a model, Dannenbring built a synthetic two-machine problem and operated Johnson's approach to solving it. The standard forms to calculate response time are presented in equation (4).

$$P_{i1} = \sum_{j=1}^{M}(M-j+1)\, t_{ij} \text{ and } P_{i2} = \sum_{j=1}^{M}(j)\, t_{ij} \text{ for } i=1, 2, \ldots, N \tag{4}$$

In an effort to find solutions rapidly, the heuristic was devised. It eliminates various flow shop issues and speeds up overall production makespan. Two auxiliary machines' downtime is calculated using Johnson's rule alone.

Fifth heuristic techniques, Hundal and Rajgopal (1988) calculated two additional sets of slope indices, which expanded on Palmer's heuristic. As a result, two more schedules are generated, from which the optimal one is chosen. These are the two groups of slope indices according to equation (5).

$$S_i = \sum_{j=1}^{M}(M-2)\, t_{ij} \text{ and } S_i = \sum_{j=1}^{M}(M-2j+2)\, t_{ij} \text{ for } i=1, 2, \ldots, N \tag{5}$$

Pascal (1973) used binomial coefficients triangle, whereas Dhanasakkaravarthi and Krishnamoorthy (2019) used a harmonic triangle. Following the work of Dhanasakkaravarthi and Krishnamoorthy, who utilized the harmonic triangle form to address PFSP by reducing 'n' jobs, 'm' machines to 'n' jobs, '2' machines, the optimal makespan could be obtained by using Johnson's rule (Ku and Niu, 1986). A modified heuristic technique, based on the time deviation (TD) technique, was developed by Rao *et al.*, 2013, which is the sixth heuristic technique considered in this paper. This approach generates a time duration table for each job vertically and horizontally. They found that the row deviation of a given cell in a time duration table equals the maximum time duration of the row minus the time duration of the cell as per equation (6).

$$P_{i1} = r_i - t_{ij}, \tag{6}$$

where r_i is the maximum time of the i^{th} row, p_{ij} is the row time deviation of the $(i, j)^{th}$ cell, and t_{ij} be the time required for processing i^{th} job on the j^{th} machine. Then, they found that the cell's column deviation in the time duration table is equal to the maximum time duration of the column minus the time duration of the cell according to equation (7).

$$C_{i1} = S_i - t_{ij} \tag{7}$$

where, S_i is the maximum time of the i^{th} column, C_{ij} is the column time deviation of the $(i, j)^{th}$ cell, and t_{ij} is the time required for processing i^{th} job on the j^{th} machine.

In the seventh heuristic technique, to reduce the overall makespan time, Jayasankari *et al.* (2021) created the JJV process, which consists of the following six steps. In step 1, create a table containing the jobs and how long they take to complete on each machine. In step 2, find the longest processing time and deduct all the other processing times from the processing times for machines M_1, M_2, M_3, …., M_n in each column of the table. One of the processing times becomes nil. Step 3 constructs group X and assigns the appropriate job if the first machine's processing time is 'zero.' Assuming it's on the machine's second half, then create group Y and assign the job to it. If it's not the first or last machine, the entry corresponding to its operation time is deleted and proceeds to the second step if that's the case. Canceling the related job in relation to the zero-processing time is step 4. Next, the remaining time in processing is used to shape the streamlined matrix. Then, all the jobs should be placed in the appropriate group, as in the previous phases. In step 5, the X and Y groups are formed. The jobs in group Y should now come first in the sequence, followed by the jobs in group X, which should now be completed last. The time elapsed may be determined in step 6 by using the acquired sequence.

Finally, the eighth heuristic technique is a recent Slope Index (SI) proposed by Abdulaal and Bafail (2021). According to them, the following are presumptions used to demonstrate the suggested heuristic method for the static flow shop:

- There are 'n' number of jobs (J) and 'm' number of machines (M).
- The order of sequence of operations of 'n' jobs on all 'm' machines is the same.
- The time required to set up is excluded from the overall processing time (makespan).

The steps of the SI heuristic technique by Abdulaal and Bafail (2021) are as follows:

Step 1: For each flow shop sequencing problem 'K' of 'n' jobs, 'm' machines, and processing time t_{ij}, determine the slope of each job's trend line T_i , along its path from the first machine to the last, using equations (8) and (9).

$$\underline{T_i} = \frac{\sum_{j=1}^{m} t_{ij}}{m} \qquad \forall\ i = 1, \dots, n \tag{8}$$

$$\underline{J} = \frac{\sum_{j=1}^{m} j}{m} \tag{9}$$

where, $\underline{T_i}$ is the average processing times for each job i, t_{ij} is the processing time of a job i on machine j, and $\underline{J}$ is the average machine numbers.

Step 2: Calculate the slope index $S_i, i = 1, \dots, n$ of each job i using equation (10).

$$S_i = \frac{\sum_{j=1}^{m} \left(\left(j - \underline{J} \right) * \left(t_{ij} - \underline{T_i} \right) \right)}{\sum_{j=1}^{m} \left(j - \underline{J} \right)^2} \qquad \forall\ i = 1, \dots, n \tag{10}$$

where, S_i: The proposed SI for each job i on machine group m.

Step 3: Rank the jobs in descending order by their indices and calculate the total processing time (makespan).

3.2 Comparative Analytical Study Stages

In flow-shop sequencing studies (Arisha *et al.*, 2002), generating a set of problems of varying sizes and then solving them with the new techniques and with one or more other proven methods designed for the same flow-shop problem is the standard approach for evaluating a heuristic or optimization model for problem-solving. A similar comparison approach is followed in this study. As shown in Figure 1, using a developed computer simulation program, 100,000 different PFSPs with different sizes were generated following the steps in Stage 1 listed below.

Stage 1: Generating different flow-shop sequencing problems with different sizes.

Step 1.1: Consider there are 'k' problem sizes, where $k = 1, \dots, K$ and $'K' = 100$, with the following: job numbers' n' equal to 4, 5, 7, 8, 10, 15, 20, 30, 50, or 80 and machine numbers' m' equal to 4, 5, 6, 10, 20, 30, 40, 70, 80, or 100. The problem sizes range from a small-size problem of 4×4 to a large-size problem of 80×100.

Step 1.2: For each problem size, 'n'×'m' generate 'r' replications, where $r = 1, \dots, R$ and $'R' = 1000$, assuming that in each replica, the processing times t_{ij} , for job i, on machine j, are uniformly distributed between 1 to 100.

Step 1.3: From the previous steps, there are $'k_r'$ different flow shop sequencing problems were created, where $k = 1, \dots, K$ and $r = 1, \dots, R$. This results in a $'K'\text{X}'R' = 100{,}000$ sequencing problems.

As shown in Figure 1, solutions for the 100,000 generated problems by the eight techniques (described in subsection 3.1) are compared in two main analysis phases. In the first phase, they are compared in terms of minimum makespan following the steps of Stage 2 presented below. In the second phase, they are evaluated to find their performance rankings following the steps of Stage 3.

3.2.1 Phase 1: Processing-Time-Based Comparative Analysis

Stage 2: Compare the eight heuristic techniques for solving Permutation Flow-Shop Scheduling Problem (PFSP).

Step 2.1: For each flow shop sequencing problem'k_r'of 'n' jobs, 'm' machines, and processing time t_{ij}, use the equations (1), (2), (3), (4), and (5) for Palmar, CDS, Gupta, Dannenbring, and Hundal, respectively, and equations (6) and (7) for TD, the six steps of JJV, and equations (8), (9), and (10) for SI. This is to identify the job sequencing eight heuristic techniques for solving PFSPs. In this case, the number of the technique under comparison is'h', where $h = 1, \dots, 8$ as listed in Table 1.

Step 2.2: Calculate the total processing time (makespan) for each sequence obtained in step 2.1 using each technique $'h'$. Let P_{hk_r} is the total processing time by technique $'h'$ for a given problem size $'k'$ at a replica $'r'$.

Step 2.3: Consider sequence problem size $'k'$ and compare the results obtained from the eight techniques with respect to the following six criteria:

$C1_{hk}$: is the average total processing time (makespan) using technique h is calculated using equation (11).

$$C1_h = \left(\sum_{r=1}^{R} P_{hk_r} \right) / R \tag{11}$$

$C2_{hk}$: is the average number of the SI technique has been better than any other seven techniques. In other words, the average number of SI techniques' overall processing time was shorter than other techniques calculated using equation (12).

$$C2_h = \left(\sum_{r=1}^{R}(P_{8k_r} < P_{hk_r})\right)/R \quad \forall\, h = 1, \dots, 7 \tag{12}$$

$C3_{hk}$: is the average percentage of improvement using the SI technique calculated using equation (13).

$$\text{For each } P_{8k_r} < P_{hk_r}, \text{ let } C3_h = \left(\sum_{r=1}^{R}\left(\frac{P_{hr_r} - P_{8k_r}}{P_{8k_r}}\right)\right)/R \quad \forall\, h = 1, \dots, 7 \tag{13}$$

$C4_{hk}$: is the average relative performance between the SI technique and the other seven techniques calculated using equation (14).

$$C4_h = \left(\sum_{r=1}^{R}\left(\frac{P_{hk_r}}{P_{8k_r}}\right)\right)/R \quad \forall\, h = 1, \dots, 7 \tag{14}$$

$C5_{hk}$: is the average percentage of error using the technique' h', and the best result obtained from all techniques under investigation is calculated using equation (15).

$$C5_h = \left(\sum_{r=1}^{R}\left(\frac{P_{hk_r} - min_h(P_{hk_r})}{min_h(P_{hk_r})}\right)\right)/R \quad \forall\, h = 1, \dots, 8 \tag{15}$$

$C6_{hk}$: is the average execution time using the technique 'h' in milliseconds calculated using equation (16).

$$C6_{hk} = \left(\sum_{r=1}^{R} E_{hk_r}\right)/R \quad \forall\, h = 1, \dots, 8 \tag{16}$$

where E_{hk_r} is the execution time it takes to compute the total processing time for problem size $'k'$ at replication $'r'$ using technique $'h'$.

Step 2.4: Repeat the above step for all sequencing problems in the list of $'k'$ before proceeding to Stage 3 below. In this study, any technique of the eight techniques can be used as a reference to compare the other seven techniques under examination. This will not have an impact on the comparison's outcomes. Here, the SI technique is chosen as a reference technique since it was the most recent proposed technique in the literature in 2021 among the other seven. Therefore, its computed performance can be charted to be visually compared with the other seven techniques according to jobs and machines. The average values for all 100,000 sequencing problems can be represented by C_1, C_2, C_3, C_4, C_5, and C_6 for the rest of the analysis instead of the preceding six criteria above that were denoted for each technique $'h'$, problem size $'k'$ and an average of 1000 replications.

3.2.2 Phase 2: Multi-Criteria Decision-Making (MCDM) Based Performance Evaluation

Stage 3: Evaluating the performance ranking of the eight heuristic techniques for solving the Permutation Flow-Shop Scheduling Problem (PFSP)

Step 3.1: Apply the MCDM technique to rank the eight techniques based on their performance. A recent MCDM tool developed by Abdulaal and Bafail (2022) was used for this step in the paper herein. This tool is known by Ranking the Alternatives using the Trace to Median Index (RATMI). According to Abdulaal and Bafail 2022, the RATMI tool was compared to seven well-known MCDM techniques, which are: Adaptive Ratio Assessment (ARAS), Simple Additive Weightage (SAW), Technique for Order Preference and Similarity to Ideal Solution (TOPSIS), Complex Proportion Assessment (COPRAS), VIseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR) in Serbian standing for Multi-criteria Optimization and Compromise Solution, Weighted Aggregated Sum Product Assessment (WASPAS), and Multi-Objective Optimization based on Ratio Analysis (MOORA). They showed the competition of the RATMI tool over these techniques. Thus, it was selected in the paper herein.

The compared eight techniques in this study were initially designed to minimize the total processing time of flow shop problems of 'n' jobs and 'm' machines. However, for their performance evaluation in this study, the six criteria, including the processing time, are considered as the RATMI ranking criteria. This is further to check their performance and validity from different performance aspects and ensure a comprehensive ranking. The used ranking criteria are as follows:

C_1: total processing time: the interval of time from the start of processing until all jobs are completed, as the starting time of the first job can be assumed as zero.
C_2: total flow time: the sum of periods in which the jobs are waiting for processing on the first machine until they are completed on the last machine.
C_3: total idle time on jobs: the period in which the jobs are waiting for processing.
C_4: total idle time on machines: the period in which the machines wait to receive the jobs in sequence.
C_5: total machine utilization: the percentage of time in which the machines are productive over the total available working time.
C_6: total execution time: the total time required to find the job sequence.

The performance criteria C_1–C_6 described above were used as the ranking criteria of the eight techniques (i.e., the alternatives), assuming equal relative importance weights with a value of 0.167 (i.e., 16.7% each) for each of the six criteria, adding up to a value of 1 (i.e., 100% for all criteria). The ranking objective was to minimize C_1, C_2, C_3, C_4, and C_6, and to maximize C_5. The required data for the RATMI is formulated using the decision-making matrix X_{ij} in equation (17).

$$\left[x_{ij}\right]_{hxz} = \begin{bmatrix} A/C & C_1 & C_2 & \dots & C_z \\ A_1 & x_{11} & x_{12} & \dots & x_{1n} \\ A_2 & x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_h & x_{m1} & x_{m2} & \dots & x_{hz} \end{bmatrix} \tag{17}$$

where,

$A = [A_1, A_2, \dots, A_h]$ is a given set of alternatives, and h is the number of techniques (i.e., A_1–A_8).
$C = [C_1, C_2, \dots, C_z]$ is a given set of criteria, and z is the total number of criteria (i.e., C_1–C_6).
$\left[x_{ij}\right]_{hxz}$ is an assessment of the alternative technique A_i with respect to a set of criteria.

Figure 2 illustrates the framework of the RATMI methodology. Results and analysis of implementing the two phases of the study and their pertaining stages and steps are provided in the subsequent section.

4. RESULTS AND DISCUSSIONS

The analysis's first phase concerns conducting a processing-time-based comparative analysis. The first two stages (i.e., Stages 1 and 2) and their pertaining steps, illustrated in Figure 1 and described in subsection 3.2.1, were implemented. A simulation experiment is employed to test the effectiveness of the eight heuristic techniques listed in Table 1 by comparing their computed makespan using the SI technique as the comparison reference. The processing times of all machines were assumed to be uniformly distributed between 1 and 40. The numbers of jobs considered are (4, 5, 7, 8, 10, 15, 20, 30, 50, 80), or 10 in total, and the numbers of machines considered are (4, 5, 6, 10, 20, 30, 40, 70, 80, 100) or 10 in total. One thousand replications are generated for each of the above (10 x 10) or 100 combinations to run the 100,000 problems. We implemented the eight heuristics in a computer simulation program. The results were computationally obtained for each of the eight techniques for each of their generated problems using equations (11–16) based on the six comparison criteria ($C1_{hk} - C6_{hk}$) described in step 2.3. Subsequently, job– and machine–specific results were charted versus each of the six computed parameters for the visual comparison.

The second phase of the analysis concerns the performance evaluation and ranking of the eight heuristic techniques under study. In this phase, Stage 3 and its pertaining steps (illustrated in Figure 1 and described in subsection 3.2.2) were implemented. This was done by following the RATMI framework in Figure 2. As described in step 3.1, the eight heuristic techniques (i.e., alternatives) were ranked based on C_1–C_6 (i.e., criteria) to obtain their performance rankings. The results of the two analysis phases are presented in the following subsections.

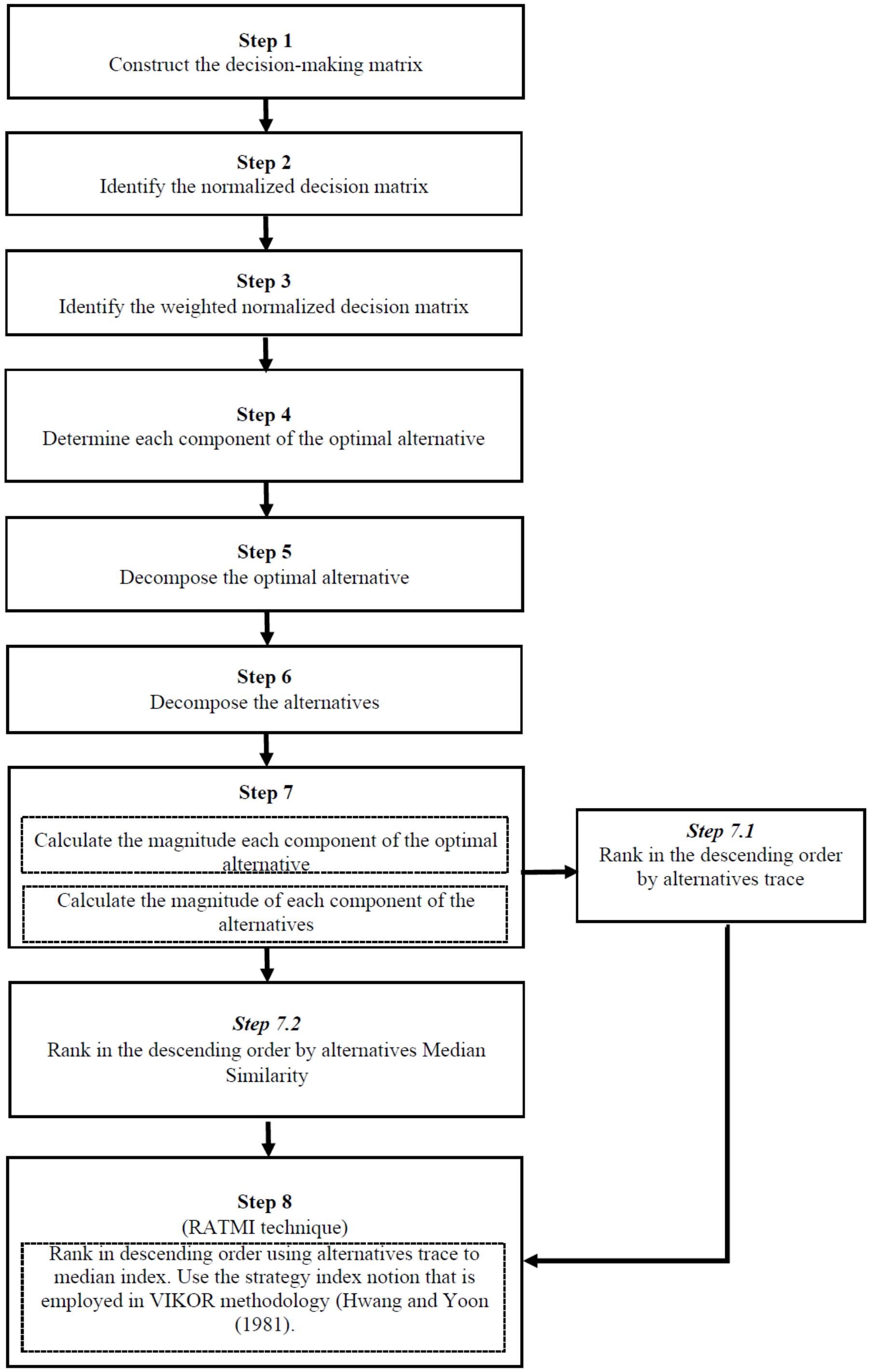
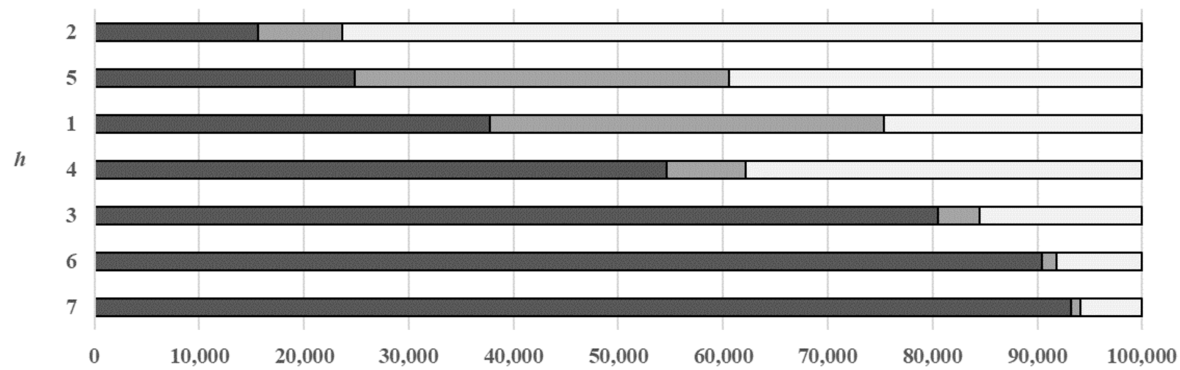


Figure 2. The Framework of The RATMI Methodology (Abdulaal and Bafail, 2022)

4.1 Results of Phase 1: Processing-Time-Based Comparative Analysis

4.1.1 Overall Results

The overall solutions by each of the seven heuristic techniques h = 1 to 7 (i.e., Palmer, CDS, Gupta, Dannenbring, Hundal, TD, and JJV), respectively, for the 100,000 generated problems are compared to the resulting solutions by h = 8 (i.e., SI technique) as the reference of the comparison, as mentioned earlier. Figure 3 summarizes and illustrates the number of times the resulting solutions by the SI technique were better, equal, or worse than the solutions of the other seven techniques in terms of shorter processing time (i.e., makespan).



h=8	7	6	3	4	1	5	2
SI (Better)	93,186	90,493	80,520	54,628	37,735	24,831	15,634
SI (Equal)	900	1,331	4,001	7,539	37,639	35,717	8,064
SI (Worse)	5,914	8,176	15,479	37,833	24,626	39,452	76,302

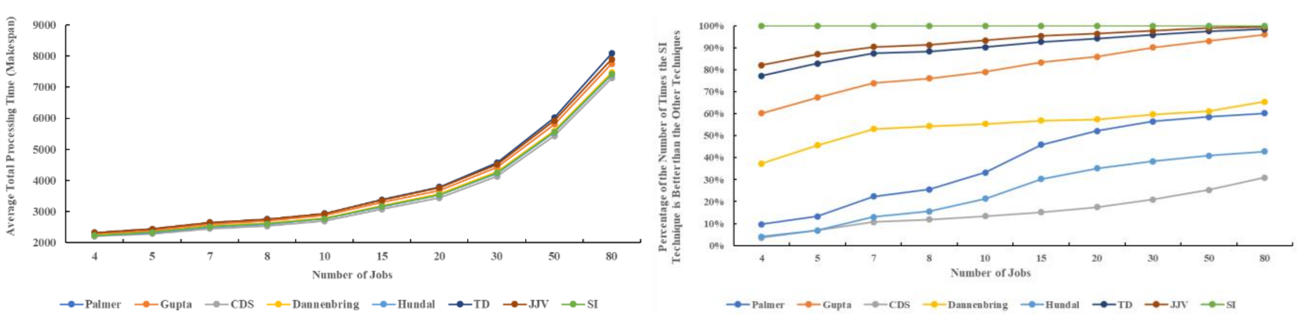
Number of Solutions (out of 100,000 Problems)

Figure 3. Overall Processing-Time-Based Comparison Results of the SI Technique Versus the Other Seven Techniques in Solving The 100,000 Problems

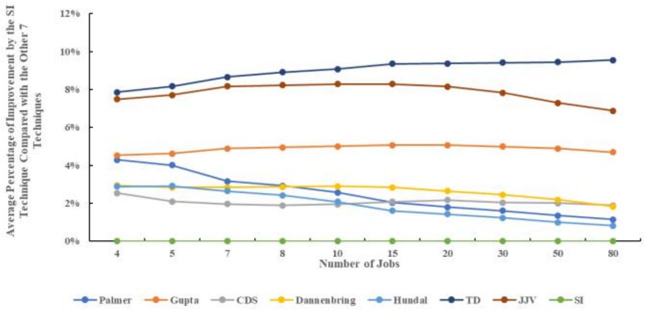
The results in Figure 3 show that, in general, *h*=8 (i.e., SI technique) outperformed or was equal to six of the studied techniques *h*=7, *h*=6, *h*=3, *h*=4, *h*=1, and *h*=5 (i.e., JJV, TD, Gupta, Dannenbring, Palmer, and Hundal), respectively. The SI technique resulted in 93.19%, 90.49%, 80.52%, 54.63%, 37.74%, and 24.83% better solutions than the other six techniques, respectively. Moreover, the SI technique resulted in 0.90%, 1.33%, 4.00%, 7.54%, 37.64%, and 35.72% equal solutions to the other six techniques, respectively. Therefore, the SI technique resulted in 94.09%, 91.82%, 84.52%, 62.17%, 75.38%, and 60.55% better and equal solutions to the other six techniques, respectively. However, the SI technique underperformed *h*=2 (i.e., CDS) and came second to it. The SI technique resulted in only 15.63% better solutions, 8.06% equal solutions, and 76.30% worse solutions than CDS. The overall results of the comparison based on processing time indicate that the recent SI technique comes second to the CDS technique in terms of processing-time performance. More detailed job- and machine-specific results are provided in the following subsections.

4.1.2 Job-Specific Results

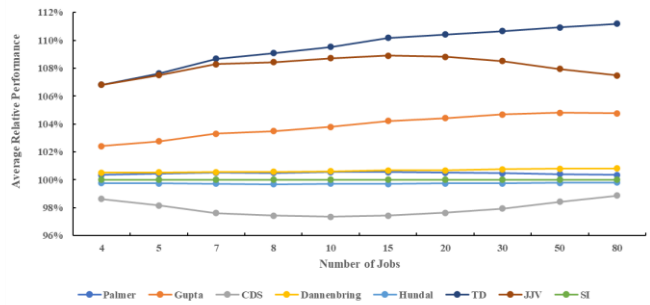
For each problem generated, the six criteria mentioned in section 3.2 have been calculated from the perspective of jobs, and the results are shown in Figure 4. Figure 4(a) (job-specific results for the average total processing time (makespan) using technique *h*), Figure 4(b) (average number of SI technique's overall processing time was shorter than other techniques), Figure 4(c) (average percentage of improvement), Figure 4(d) (average relative performance), Figure 4(e) (average percentage of error), and Figure 4(f) (average execution time).



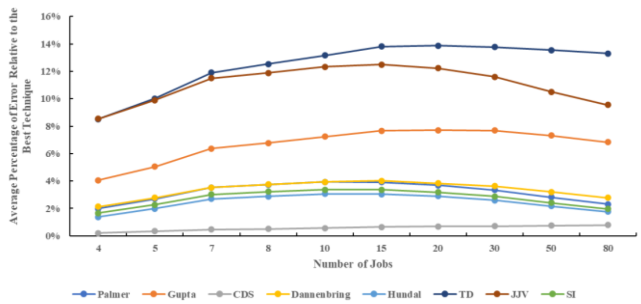
(a) C1 (b) C2



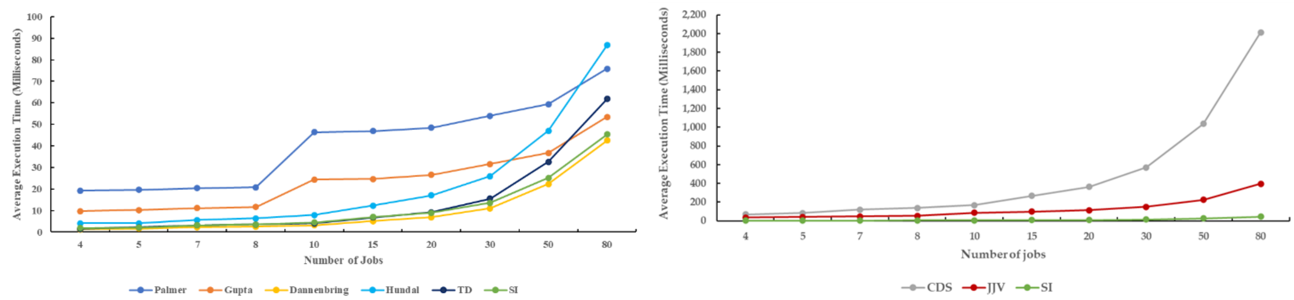
(c) C3



(d) C4



(e) C5



(f) C6

Figure 4. Number of Jobs Versus Each of The Six Comparison Criteria (C1–C6)

Figure 4(a) shows that the CDS is better than all other techniques in average processing time. The Hundal and SI heuristic techniques are very close concerning average processing time. On the other hand, Gupta, TD, and JJV are the worst tetchiness in the average makespan. In Figure 4(b), the SI heuristic is set to 100%, and the other heuristics' relative performance in terms of jobs is computed. It shows that SI performs better than all other techniques in small-size problems, and when the number of jobs in the problem increases, the percentage of the number of times the SI is better than the other seven techniques. Also, the average percentage of improvement by SI compared with seven other methods is increased concerning the number of jobs for the Gupta, TD, and JJV techniques. At the same time, it is reduced for all other heuristics techniques concerning the number of jobs. That means on large-scale PFSPs, the average percentage of improvement between Palmer, CDS, Dannenbring, Hundal, and SI is minimal, as shown in Figure 4(c). The SI heuristic has a similar average relative performance as Palmer and Dannenbring. CDS and Hundal are better in average relative performance, as shown in Figure 4(d).

Moreover, the same behavior is observed in the average error percentage between the best makespan and the one obtained from the other technique. It is clear that the CDS is the best with the lowest error percentage than Hundal and SI. Where Gupta, TD, and JJV techniques are underperforming, as shown in Figure 4(e). However, for the average execution time in milliseconds, the Dannenbring heuristic is the fastest technique, with about 9.8 milliseconds, then the SI with 11.5 milliseconds. The CDS technique took the longest execution time, averaging about 482.1 milliseconds. Although, the average execution time increases dramatically with the number of jobs, as shown in Figure 4(f). However, the average execution time of the SI heuristic is short enough for any application.

4.1.3 Machine-Specific Results

For all 100,000 generated problems, the machine-specific results series of each technique h are charted for each of the six comparison criteria as illustrated in Figure 5(a–f). The numbers of machines considered (4, 5, 6, 10, 20, 30, 40, 70, 80, 100) are charted versus the average total processing time (i.e., makespan) in Figure 5(a), the average percentage of the SI technique has been better than the other seven techniques in terms of shorter processing time in Figure 5(b), the average percentage of improvement using the SI technique in Figure 5(c), the average relative performance between the SI technique and the other seven techniques in Figure 5(d), the average percentage of error based on the best result obtained from all techniques in Figure 5(e), and the average execution time in milliseconds in Figure 5(f).

Results show that the eight compared techniques demonstrated similar average total processing time (i.e., makespan) behaviors across the experimented machine sizes, as illustrated in Figure 5(a). Results show that the SI technique outperformed all other seven techniques, with produced solutions having the shortest processing time for problems of less than ten machines. However, for problems of more than ten machines, the CDS technique outperformed all seven other techniques in producing solutions with the shortest processing time.

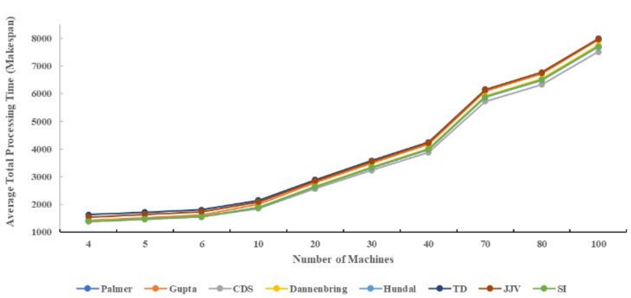
The average percentage of the SI technique has been better than the other seven techniques in terms of shorter processing time presented in Figure 5(b); the SI technique is set to 100% for the purpose of comparison. The results show that the SI technique outperformed the other seven techniques, with the JJV technique being the closest to its performance. It also could be observed that the JJV, Palmar, Hundal, and CDS techniques demonstrated similar behavior of problems with more than ten machines opposite to the remaining techniques (i.e., TD, Gupta, and Dannenbring).

For the average percentage of improvement using the SI technique presented in Figure 5(c), the SI technique is set to 0% for the purpose of comparison. The results show that the SI technique outperformed the other seven techniques with the CDS, Hundal, and Palmar techniques, demonstrating the closest performance, especially in problems of a larger number of machines.

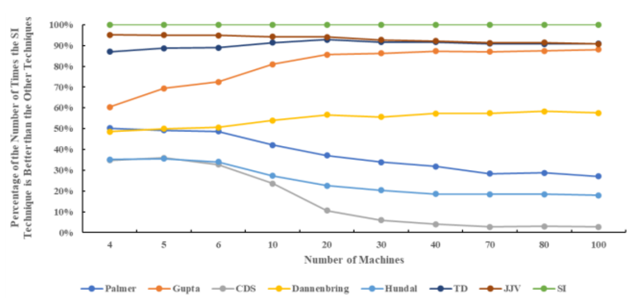
For the average relative performance between the SI technique and the other seven techniques presented in Figure 5(d), the CDS outperforms the SI technique. However, the SI technique performed similarly to the techniques by Palmar, Hundal, and Dannenbring. The results also indicate that the relative performance of all techniques gets better for problems above ten machines, opposite to Gupta's technique which performs better in smaller problems of less than ten machines.

For the average percentage of error based on the best result obtained from all eight techniques presented in Figure 5(e), the CDS technique outperformed all seven other techniques. The SI, Palmar, and Gupta came second to CDS, showing similar error percentages. The results also indicate that the percentages of errors of all techniques get lower or remain steady in problems of small to large machine sizes, opposite to Gupta's technique which performs better in smaller problems of less than ten machines.

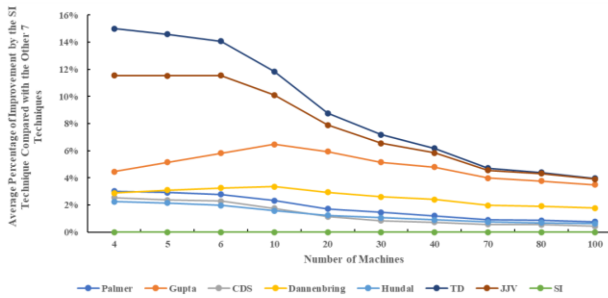
The average execution time measured in milliseconds is presented in two charts in Figure 5(f) for better visualization due to the larger execution time demonstrated by the CDS and JJV techniques. The results show that Dannenbring's technique outperformed all seven other techniques, with the SI and TD being the closest in terms of execution time. The results also indicate a general increased execution time trend of all eight techniques as the number of machines in the problems increases, which is expected. However, dramatically increased execution time trends by the CSD and JJV techniques, especially in problems with more than ten machines.



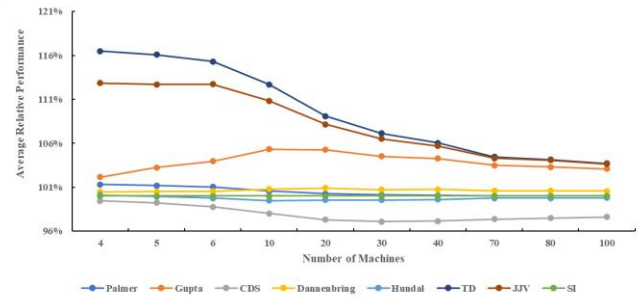
(a) C1



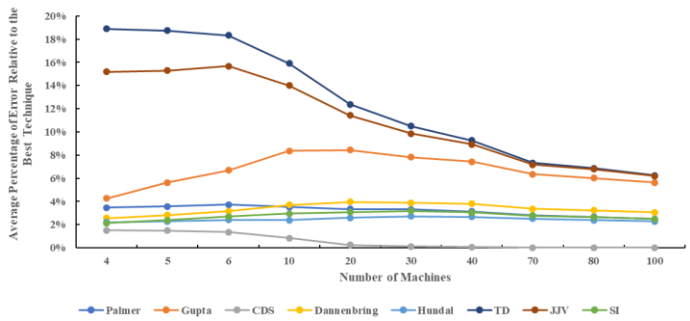
(b) C2



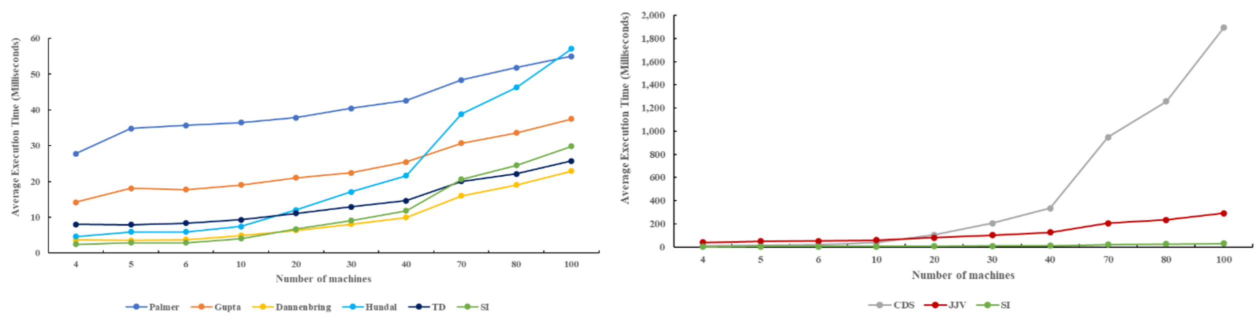
(c) C3



(d) C4



(e) C5



(f) C6

Figure 5. Number of Machines Versus Each of The Six Comparison Criteria (C1–C6)

4.2 Results of Phase 2: Multi-Criteria Decision-Making (MCDM) Based Performance Evaluation

Despite that, the overall results of the comparison based on processing time are presented in subsection 4.1.1. indicated that the recent SI technique comes second to the CDS technique; it is valuable to check how the eight studied techniques will perform when their performance is evaluated based on a set of more comprehensive criteria. Thus, the third stage and its pertaining steps (illustrated in Figure 1 and described in subsection 3.2.2) were implemented.

The second phase of the analysis was performed following the RATMI framework in Figure 2 for the MCDM performance evaluation. As described in step 3.1, the eight heuristic techniques A_1–A_8 (i.e., alternatives) were ranked based on C_1–C_6 (i.e., criteria) to obtain their performance rankings. The used ranking criteria are the total processing time, total flow time, total idle time on jobs, total idle time on machines, total machine utilization, and total execution time, respectively. As mentioned earlier, equal relative importance weights of the six criteria are assumed with a value of 0.167 (i.e., 16.7% each) for each of the six criteria, adding up to a value of 1 (i.e., 100% for all criteria). The ranking objective was to minimize C_1, C_2, C_3, C_4, and C_6 (i.e., total processing time, total flow time, total idle time on jobs, total idle time on machines, execution time) and to maximize C_5 (i.e., the total machine utilization), respectively. The required data based on the six performance criteria were computed for the 100,000 runs. Accordingly, the input decision-making matrix for the RATMI is formulated using equation (17), as presented in Table 2. Subsequently, following the RATMI framework illustrated in Figure 2, the normalized decision-making matrix and the weighted normalized decision-making matrix were developed, as shown in Tables 3 and 4, respectively. The RATMI rankings were found as presented in Table 5, and the eight heuristic techniques are ranked in descending order based on their performance in the six criteria, as shown in Table 6.

Table 2. Input Decision-Making Matrix

A_h	Criteria	C_1	C_2	C_3	C_4	C_5	C_6
	Weight	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667
	Objective	Min.	Min.	Min.	Min.	Max.	Min.
A_1		3644.23	87083.97	21818.88	163761.45	38.21%	41.079
A_2		3552.51	85678.79	19682.55	158245.13	39.08%	482.114
A_3		3780.61	88954.49	24762.03	171008.66	37.02%	23.981
A_4		3660.38	86154.18	20215.57	164888.73	38.25%	9.824
A_5		3626.62	86818.40	21434.50	163081.90	38.56%	21.678
A_6		3897.00	89633.03	20991.07	173512.89	34.45%	14.015
A_7		3856.67	89820.90	21289.24	172920.99	34.98%	124.041
A_8		3636.83	86700.69	21408.89	163651.86	38.48%	11.495

Table 3. Normalized Decision-Making Matrix

A_h	Criteria	C_1	C_2	C_3	C_4	C_5	C_6
	Weight	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667
	Objective	Min.	Min.	Min.	Min.	Max.	Min.
A_1		0.9748	0.9839	0.9021	0.9663	0.9777	0.2392
A_2		1.0000	1.0000	1.0000	1.0000	1.0000	0.0204
A_3		0.9397	0.9632	0.7949	0.9254	0.9473	0.4097
A_4		0.9705	0.9945	0.9736	0.9597	0.9788	1.0000
A_5		0.9796	0.9869	0.9183	0.9703	0.9867	0.4532
A_6		0.9116	0.9559	0.9377	0.9120	0.8815	0.7010
A_7		0.9211	0.9539	0.9245	0.9151	0.8951	0.0792
A_8		0.9768	0.9882	0.9194	0.9670	0.9846	0.8547

Table 4. Weighted Normalized Decision-Making Matrix

A_h	Criteria	C_1	C_2	C_3	C_4	C_5	C_6
	Weight	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667
	Objective	Min.	Min.	Min.	Min.	Max.	Min.
A_1		0.1625	0.1640	0.1504	0.1611	0.1630	0.0399
A_2		0.1667	0.1667	0.1667	0.1667	0.1667	0.0034
A_3		0.1566	0.1606	0.1325	0.1543	0.1579	0.0683
A_4		0.1618	0.1658	0.1623	0.1600	0.1632	0.1667
A_5		0.1633	0.1645	0.1531	0.1618	0.1645	0.0755
A_6		0.1520	0.1593	0.1563	0.1520	0.1470	0.1169
A_7		0.1536	0.1590	0.1541	0.1526	0.1492	0.0132
A_8		0.1628	0.1647	0.1533	0.1612	0.1641	0.1425

Table 5. Alternative Performance Rankings

Alternative	Alternative Trace		Alternative Median Similarity		RATMI	
	Value	Rank	Value	Rank	Value	Rank
A_1	0.1471	6	0.8831	6	0.2934	6
A_2	0.1521	3	0.9129	3	0.5113	3
A_3	0.1420	7	0.8528	7	0.0731	7
A_4	0.1633	1	0.9796	1	1.0000	1
A_5	0.1505	4	0.9034	4	0.4425	4
A_6	0.1479	5	0.8873	5	0.3282	5
A_7	0.1404	8	0.8424	8	0.0000	8
A_8	0.1583	2	0.9496	2	0.7813	2

Table 6. Overall Performance Rankings in Descending Order

Alternative	Rank
A_1	1
A_2	2
A_3	3
A_4	4
A_5	5
A_6	6
A_7	7
A_8	8

4.3 Discussion

The results of the conducted comparison based on processing time in the first phase of analysis provide evidence that the recently proposed SI technique comes second to the CDS technique based on the six comparison criteria. Also, the more detailed job- and machine-specific results indicated that some of the eight techniques were better than others in each of the six comparison criteria. It was also observed that some of the techniques behaved differently in smaller and larger problem sizes in terms of the number of jobs and machines. Results indicate that the size of problems determines the suitable technique for solving flow–shop sequencing problems, especially the number of ten jobs and ten machines that seemed like a threshold to consider.

Results of the study's second phase following the MCDM approach using RATMI revealed the performance rankings of the eight heuristic techniques. The ranking process was based on their overall performance in total processing time, total flow time, total idle time on jobs, total idle time on machines, total machine utilization, and total execution time. The objective was to find the ranking that minimizes all of them and maximizes the execution time. Results revealed that Dannenbring's technique is the first best, followed by the SI technique as the second best, followed by the CDS, Hundal, TD, Palmer, Gupta, and JJV techniques.

The results of this study help in choosing the heuristic technique that optimizes the time spent and resources utilized using a particular number of machines to decide what jobs to do and in what sequence. Solving flow-shop problems minimizes the makespan or the time it takes for all jobs to be completed. This, in turn, helps reach for better sequencing in workshop scheduling that reduces production costs and boosts output. Furthermore, the performance rankings of the techniques for solving permutation flow-shop sequencing problems provide practical insights into their performance in different problem sizes. This, in turn, helps the industrial and manufacturing sectors schedule activities efficiently and quickly to manage their resources better.

5. CONCLUSIONS

This paper focuses on comparing heuristic methods for addressing permutation flow-shop sequencing issues. Flow-shop issues reduce the time it takes to do all jobs, lowering manufacturing costs and increasing productivity. Therefore, various heuristics have been created to help find a good and fast solution. However, new methods must be tested for performance versus classical ones. Therefore, this paper aims to conduct a comparative analytical, computational study of heuristic techniques for solving PFSPs and evaluating their performance. The performance of eight PFSP methods in solving 100,000 generated problems using computer simulation software was compared in two main analysis phases.

The comparison based on processing time in the first phase of the analysis showed that the CDS technique outperforms the recently proposed SI technique on the six comparison criteria. In addition, job- and machine-specific data showed that some of the eight approaches performed better in each of the six comparison criteria. In terms of jobs and machines, several strategies operated differently in smaller and larger problem sizes. Results show that problem size should determine the best flow-shop sequencing technique to be used, especially the number of ten jobs and ten machines that appeared like a critical threshold where the performance of techniques starts behaving differently.

The eight heuristic approaches' performance rankings were disclosed in the study's second phase using RATMI. Their total processing time, flow time, idle time on jobs and machines, machine utilization, and execution time were used to rank them. The goal was to find the ranking that minimizes them all and maximizes machine utilization. Results showed that Dannenbring's method is the best, followed by the SI technique, CDS, Hundal, TD, Palmer, Gupta, and JJV approaches.

This paper puts forward a comparative analytical and computational approach, including the used comparison and performance evaluation criteria, methods, and the MCDM approach using RATMI. Moreover, the implications of this paper include the revealed performance rankings of the techniques for solving permutation flow-shop sequencing problems and the practical insights on their performance in different problem sizes. The findings of this paper assist the industrial and manufacturing sectors in scheduling activities efficiently and quickly to manage their resources better.

Despite that, the findings of this study are considered representative of the used problem sizes in terms of the number of jobs and machines; reconducting the analysis following the same or different comparison approaches considering different sets of job and machine sizes is a research direction to confirm the results further. Furthermore, the resulting performance rankings in this paper using MCDM were based on equal importance weights of the used criteria using RATMI. Therefore, reconducting the evaluation using other MCDM techniques and using the same or different set of evaluation criteria with varying weights of importance depending on the specific application context of the methods is a future research direction that might yield further insights. Another future research direction is to develop a solution method by combining decision trees and meta-heuristic algorithms. Finally, this study focused on comparing the eight PFSP techniques. Therefore, investigating the performance of other and future-developed PFSP methods is recommended for future research studies.

REFERENCES

Abdulaal, R. and Bafail, O. A. (2021). A New Slope Index for Solving Nxm Flow Shop Sequencing Problems with Minimum Makespan. *Transactions of FAMENA*, *45*(4): 0-0.

Abdulaal, R. and Bafail, O. A. (2022). Two New Approaches (RAMS-RATMI) in Multi-Criteria Decision-Making Tactics. *Journal of Mathematics*, *2022*.

Al Kattan, I. and Maragoud, R. (2008). Performance Analysis of Flowshop Scheduling Using Genetic Algorithm Enhanced with Simulation. *International Journal of Industrial Engineering: Theory, Applications and Practice*, *15*(1): 62-72.

Alawad, N. A. and Abed-Alguni, B. H. (2022). Discrete Jaya with Refraction Learning and Three Mutation Methods for The Permutation Flow Shop Scheduling Problem. *The Journal of Supercomputing*, *78*(3): 3517-3538.

Amdouni, H., Jemmali, M., Mrad, M., and Ladhari, T. (2021). An Exact Algorithm Minimizing The Makespan for The Two Machine Flowshop Scheduling Under Release Dates and Blocking Constraints. *International Journal of Industrial Engineering: Theory, Applications and Practice*, *28*(6).

Arisha, A., Young, P., and El Baradie, M. (2001, July). Job Shop Scheduling Problem: An Overview. in *International Conference for Flexible Automation and Intelligent Manufacturing (FAIM 01)* (Pp. 682-693).

Arisha, A., Young, P., and El Baradie, M. (2002, January). Flow Shop Scheduling Problem: A Computational Study. in *Proceedings of The Sixth International Conference on Production Engineering and Design for Development (PEDD6)* (Pp. 543-557).

Bish, E. K. (2003). A Multiple-Crane-Constrained Scheduling Problem in A Container Terminal. *European Journal of Operational Research*, *144*(1): 83-107.

Bonney, M. C., and Gundry, S. W. (1976). Solutions to The Constrained Flowshop Sequencing Problem. *Journal of The Operational Research Society*, *27*(4): 869-883.

Brammer, J., Lutz, B., and Neumann, D. (2022). Permutation Flow Shop Scheduling with Multiple Lines and Demand Plans Using Reinforcement Learning. *European Journal of Operational Research*, *299*(1): 75-86.

Campbell, H. G., Dudek, R. A., and Smith, M. L. (1970). A Heuristic Algorithm for The N Job, M Machine Sequencing Problem. *Management Science*, *16*(10): B-630.

Chakraborty, U. K., and Laha, D. (2007). An Improved Heuristic for Permutation Flowshop Scheduling. *International Journal of Information and Communication Technology*, *1*(1): 89-97.

Chen, J. S., Pan, J. C. H., and Lin, C. M. (2009). Solving The Reentrant Permutation Flow-Shop Scheduling Problem with A Hybrid Genetic Algorithm. *International Journal of Industrial Engineering: Theory Applications and Practice*, *16*(1): 23-31.

Coffman Jr, E. G. (1976). *Scheduling in Computer and Job Shop Systems*. Wiley, New York.

Costa, A., Fernandez-Viagas, V., and Framiñan, J. M. (2020). Solving The Hybrid Flow Shop Scheduling Problem with Limited Human Resource Constraint. *Computers & Industrial Engineering*, *146*, 106545.

Cui, L., Liu, X., Lu, S., and Jia, Z. (2021). A Variable Neighborhood Search Approach for The Resource-Constrained Multi-Project Collaborative Scheduling Problem. *Applied Soft Computing*, *107*, 107480.

Dannenbring, D. G. (1977). An Evaluation of Flow Shop Sequencing Heuristics. *Management Science*, *23*(11): 1174-1182.

De Fátima Morais, M., Ribeiro, M. H. D. M., Da Silva, R. G., Mariani, V. C., and Dos Santos Coelho, L. (2022). Discrete Differential Evolution Metaheuristics for Permutation Flow Shop Scheduling Problems. *Computers & Industrial Engineering*, *166*, 107956.

Dhanasakkaravarthi, B. and Krishnamoorthy, D. A. (2019). A New Heuristic Algorithm to Determine More Than One Sequence in Permutation Flow Shop Scheduling by Using Harmonic Triangle. *International Journal of Mechanical Engineering and Technology*, *10*(3).

Dong, X., Huang, H., and Chen, P. (2008). An Improved NEH-Based Heuristic for The Permutation Flowshop Problem. *Computers & Operations Research*, *35*(12): 3962-3968.

Doush, I. A., Al-Betar, M. A., Awadallah, M. A., Alyasseri, Z. A. A., Makhadmeh, S. N., and El-Abd, M. (2022). Island Neighboring Heuristics Harmony Search Algorithm for Flow Shop Scheduling with Blocking. *Swarm and Evolutionary Computation*, *74*, 101127.

Du, J., and Leung, J. Y. T. (1990). Minimizing Total Tardiness on One Machine Is NP-Hard. *Mathematics of Operations Research*, *15*(3): 483-495.

Fathollahi-Fard, A. M., Woodward, L., and Akhrif, O. (2021). Sustainable Distributed Permutation Flow-Shop Scheduling Model Based on A Triple Bottom Line Concept. *Journal of Industrial Information Integration*, *24*, 100233.

Fernandez-Viagas, V., Ruiz, R., and Framinan, J. M. (2017). A New Vision of Approximate Methods for The Permutation Flowshop to Minimise Makespan: State-Of-The-Art and Computational Evaluation. *European Journal of Operational Research*, *257*(3): 707-721.

Fernandez-Viagas, V., Sanchez-Mediano, L., Angulo-Cortes, A., Gomez-Medina, D., and Molina-Pariente, J. M. (2022a). The Permutation Flow Shop Scheduling Problem with Human Resources: MILP Models, Decoding Procedures, NEH-Based Heuristics, and An Iterated Greedy Algorithm. *Mathematics*, *10*(19): 3446.

Fernandez-Viagas, V., Talens, C., and Framinan, J. M. (2022b). Assembly Flowshop Scheduling Problem: Speed-Up

Procedure and Computational Evaluation. *European Journal of Operational Research*, *299*(3): 869-882.

Framinan, J. M., Leisten, R., and Rajendran, C. (2003). Different Initial Sequences for The Heuristic of Nawaz, Enscore and Ham to Minimize Makespan, Idletime Or Flowtime in The Static Permutation Flowshop Sequencing Problem. *International Journal of Production Research*, *41*(1): 121-148.

Fuchigami, H. Y., Sarker, R., and Rangel, S. (2018). Near-Optimal Heuristics for Just-In-Time Jobs Maximization in Flow Shop Scheduling. *Algorithms*, *11*(4): 43.

Garey, M. R., Graham, R. L., Johnson, D. S., and Yao, A. C. C. (1976a). Resource Constrained Scheduling as Generalized Bin Packing. *Journal of Combinatorial Theory, Series A*, *21*(3): 257-298.

Garey, M. R., Johnson, D. S., and Sethi, R. (1976b). The Complexity of Flowshop and Jobshop Scheduling. *Mathematics of Operations Research*, *1*(2): 117-129.

Gupta, A., and Chauhan, S. (2015). A Heuristic Algorithm for Scheduling in A Flow Shop Environment to Minimize Makespan. *International Journal of Industrial Engineering Computations*, *6*(2): 173-184.

Gupta, J. N. (1971). A Functional Heuristic Algorithm for The Flowshop Scheduling Problem. *Journal of The Operational Research Society*, *22*(1): 39-47.

Gupta, J. N. (1976). Optimal Flowshop Schedules with No Intermediate Storage Space. *Naval Research Logistics Quarterly*, *23*(2): 235-243.

Hall, N. G., Potts, C. N., and Sriskandarajah, C. (2000). Parallel Machine Scheduling with A Common Server. *Discrete Applied Mathematics*, *102*(3): 223-243.

Hossain, M. S., Asadujjaman, M., Nayon, M. A., and Bhattacharya, P. (2014). Minimization of Makespan in Flow Shop Scheduling Using Heuristics. in *International Conference on Mechanical, Industrial and Energy Engineering*.

Hundal, T. S., and Rajgopal, J. (1988). An Extension of Palmer's Heuristic for The Flow Shop Scheduling Problem. *International Journal of Production Research, 26*(6): 1119-1124.

Ignall, E., and Schrage, L. (1965). Application of The Branch and Bound Technique to Some Flow-Shop Scheduling Problems. *Operations Research*, *13*(3): 400-412.

Jayasankari, S. (2021). An Efficient Flow Shop Scheduling Problem with Makespan Objective. *Turkish Journal of Computer and Mathematics Education (TURCOMAT): 12*(4): 461-466.

Johnson, S. M. (1954). Optimal Two-and Three-Stage Production Schedules with Setup Times Included. *Naval Research Logistics Quarterly*, *1*(1): 61-68.

Kalczynski, P. J., and Kamburowski, J. (2007). On The NEH Heuristic for Minimizing The Makespan in Permutation Flow Shops. *Omega*, *35*(1): 53-60.

Kalczynski, P. J., and Kamburowski, J. (2008). An Improved NEH Heuristic to Minimize Makespan in Permutation Flow Shops. *Computers & Operations Research*, *35*(9): 3001-3008.

Kalczynski, P. J., and Kamburowski, J. (2009). An Empirical Analysis of The Optimality Rate of Flow Shop Heuristics. *European Journal of Operational Research*, *198*(1): 93-101.

Kempf, D. J., Sham, H. L., Marsh, K. C., Flentge, C. A., Betebenner, D., Green, B. E., ... and Norbeck, D. W. (1998). Discovery of Ritonavir, A Potent Inhibitor of HIV Protease with High Oral Bioavailability and Clinical Efficacy. *Journal of Medicinal Chemistry*, *41*(4): 602-617.

Kim, M. Y., and Lee, Y. H. (2012). MIP Models and Hybrid Algorithm for Minimizing The Makespan of Parallel

Machines Scheduling Problem with A Single Server. *Computers & Operations Research*, *39*(11): 2457-2468.

King, J. R., and Spachis, A. S. (1980). Heuristics for Flow-Shop Scheduling. *International Journal of Production Research*, *18*(3): 345-357.

Komaki, G. M., Sheikh, S., and Malakooti, B. (2019). Flow Shop Scheduling Problems with Assembly Operations: A Review and New Trends. *International Journal of Production Research*, *57*(10): 2926-2955.

Koulamas, C. (1996). Single-Machine Scheduling with Time Windows and Earliness/Tardiness Penalties. *European Journal of Operational Research*, *91*(1): 190-202.

Koulamas, C. (1998). A New Constructive Heuristic for The Flowshop Scheduling Problem. *European Journal of Operational Research*, *105*(1): 66-71.

Ku, P. S., and Niu, S. C. (1986). On Johnson's Two-Machine Flow Shop with Random Processing Times. *Operations Research*, 34(1): 130-136.

Laribi, I., Yalaoui, F., Belkaid, F., and Sari, Z. (2016). Heuristics for Solving Flow Shop Scheduling Problem Under Resources Constraints. *IFAC-Papersonline*, *49*(12): 1478-1483.

Lee, J. H., and Kim, H. J. (2022). Reinforcement Learning for Robotic Flow Shop Scheduling with Processing Time Variations. *International Journal of Production Research*, *60*(7): 2346-2368.

Li, J. Q., Pan, Q. K., and Gao, K. Z. (2011). Pareto-Based Discrete Artificial Bee Colony Algorithm for Multi-Objective Flexible Job Shop Scheduling Problems. *The International Journal of Advanced Manufacturing Technology*, *55*(9): 1159-1169.

Lin, S. W., and Ying, K. C. (2016). Minimizing Makespan for Solving The Distributed No-Wait Flowshop Scheduling Problem. *Computers & Industrial Engineering*, *99,* 202-209.

Liu, W., Jin, Y., and Price, M. (2017). A New Improved NEH Heuristic for Permutation Flowshop Scheduling Problems. *International Journal of Production Economics*, *193,* 21-30.

Meng, X., Wang, N., Liu, J., and Liu, Q. (2022). An Improved Simulated Annealing-Based Decision Model for The Hybrid Flow Shop Scheduling of Aviation Ordnance Handling. *Computational Intelligence and Neuroscience*, *2022*.

Miltenburg, J. (2001). One-Piece Flow Manufacturing on U-Shaped Production Lines: A Tutorial. *IIE Transactions*, *33*(4): 303-321.

Missah, Y. M. (2015). Empirical Bibliography on Enterprise Resources Planning Systems Implementation. *International Journal of Computer Applications*, *128*(15).

Modrák, V., Semančo, P., and Knuth, P. (2012). Alternative Constructive Heuristic Algorithm for Permutation Flow-Shop Scheduling Problem with Make-Span CRITERION. *International Journal of Industrial Engineering: Theory, Applications and Practice*, *19*(7).

Muştu, S., and Eren, T. (2018). Maximum Completion Time Under A Learning Effect in The Permutation Flowshop Scheduling Problem. *International Journal of Industrial Engineering: Theory, Applications and Practice*, *25*(2).

Nawaz, M., Enscore Jr, E. E., and Ham, I. (1983). A Heuristic Algorithm for The M-Machine, N-Job Flow-Shop Sequencing Problem. *Omega*, *11*(1): 91-95.

Neufeld, J. S., Schulz, S., and Buscher, U. (2022). A Systematic Review of Multi-Objective Hybrid Flow Shop Scheduling. *European Journal of Operational Research*.

Page, E. S. (1961). An Approach to The Scheduling of Jobs on Machines. *Journal of The Royal Statistical Society: Series B*

(Methodological): 23(2): 484-492.

Palmer, D. S. (1965). Sequencing Jobs Through A Multi-Stage Process in The Minimum Total Time—A Quick Method of Obtaining A Near Optimum. *Journal of The Operational Research Society*, *16*(1): 101-107.

Rad, S. F., Ruiz, R., and Boroojerdian, N. (2009). New High Performing Heuristics for Minimizing Makespan in Permutation Flowshops. *Omega*, *37*(2): 331-345.

Rahmouni Elidrissi, Y., and Courpasson, D. (2021). Body Breakdowns As Politics: Identity Regulation in A High-Commitment Activist Organization. *Organization Studies*, *42*(1): 35-59.

Rao, N. N., Raju, O. N., and Babu, I. R. (2013). Modified Heuristic Time Deviation Technique for Job Sequencing and Computation of Minimum Total Elapsed Time. *International Journal of Computer Science & Information Technology*, *5*(3): 67.

Reza Hejazi, S., and Saghafian, S. (2005). Flowshop-Scheduling Problems with Makespan Criterion: A Review. *International Journal of Production Research*, *43*(14): 2895-2929.

Ribas, I., Companys, R., and Tort-Martorell, X. (2021). An Iterated Greedy Algorithm for The Parallel Blocking Flow Shop Scheduling Problem and Sequence-Dependent Setup Times. *Expert Systems with Applications*, *184*, 115535.

Rifai, A. P., Mara, S. T. W., and Sudiarso, A. (2021). Multi-Objective Distributed Reentrant Permutation Flow Shop Scheduling with Sequence-Dependent Setup Time. *Expert Systems with Applications*, *183*, 115339.

Rinnooy Kan, A. H. (1976a). Machine Scheduling Problems: Classification, Complexity, and Computations. *PhD Thesis, University of Amsterdam*.

Rinnooy Kan, A. H. G. (1976b). Machine Scheduling Problems: Classification. *Complexity and Computations, Nijhoff, The Hague*.

Ruiz, R., and Maroto, C. (2005). A Comprehensive Review and Evaluation of Permutation Flowshop Heuristics. *European Journal of Operational Research*, *165*(2): 479-494.

Ruiz, R., Maroto, C., and Alcaraz, J. (2005). Solving The Flowshop Scheduling Problem with Sequence Dependent Setup Times Using Advanced Metaheuristics. *European Journal of Operational Research*, *165*(1): 34-54.

Sayadi, M., Ramezanian, R., and Ghaffari-Nasab, N. (2010). A Discrete Firefly Meta-Heuristic with Local Search for Makespan Minimization in Permutation Flow Shop Scheduling Problems. *International Journal of Industrial Engineering Computations*, *1*(1): 1-10.

Seeanner, F., and Meyr, H. (2013). Multi-Stage Simultaneous Lot-Sizing and Scheduling for Flow Line Production. *OR Spectrum*, *35*(1): 33-73.

Sharma, M., Sharma, M., and Sharma, S. (2022). Integrated Stochastic Bicriteria Flow Shop Scheduling Problem with Preventive Maintenance. *International Journal of Process Management and Benchmarking*, *12*(6): 744-784.

Smith, K. J. (1973). Pascal's Triangle. *The Two-Year College Mathematics Journal*, *4*(1): 1-13.

Song, H. B., and Lin, J. (2021). A Genetic Programming Hyper-Heuristic for The Distributed Assembly Permutation Flow-Shop Scheduling Problem with Sequence Dependent Setup Times. *Swarm and Evolutionary Computation*, *60*, 100807.

Stinson, J. P., and Smith, A. W. (1982). A Heuristic Proǵramminǵ Procedure for Sequencinǵ The Static Flowshop. *The International Journal of Production Research*, *20*(6): 753-764.

Taillard, E. (1990). Some Efficient Heuristic Methods for The Flow Shop Sequencing Problem. *European Journal of Operational Research*, *47*(1): 65-74.

Tempelmeier, H., and Copil, K. (2016). Capacitated Lot Sizing with Parallel Machines, Sequence-Dependent Setups, and A Common Setup Operator. *OR Spectrum*, *38*(4): 819-847.

Torjai, L., and Kruzslicz, F. (2016). Mixed Integer Programming Formulations for The Biomass Truck Scheduling Problem. *Central European Journal of Operations Research*, *24*(3): 731-745.

Vasiljevic, D., and Danilovic, M. (2015). Handling Ties in Heuristics for The Permutation Flow Shop Scheduling Problem. *Journal of Manufacturing Systems*, *35*, 1-9.

Wu, C. C., Chen, J. C., Wu, W. H., Hsu, P. H., and Wu, W. H. (2011). Simulated Annealing Algorithms for The Two-Machine Makespan Flowshop Scheduling with Truncated Learning Consideration. *International Journal of Industrial Engineering: Theory, Applications and Practice*, *18*(8).

Zaied, A. N. H., Ismail, M. M., and Mohamed, S. S. (2021). Permutation Flow Shop Scheduling Problem with Makespan Criterion: Literature Review. *J. Theor. Appl. Inf. Technol*, *99*(4).