Granular Computing (2023) 8:1097–1120
https://doi.org/10.1007/s41066-023-00377-y

ORIGINAL PAPER

A priority-based heuristic approach for solving flexible flow-shop with parallel machine scheduling in a fuzzy environment

Syed Abou Iltaf Hussain[1] · Ranbir Kalita[2] · Uttam Kumar Mandal[3]

Received: 18 January 2023 / Accepted: 9 March 2023 / Published online: 5 April 2023

Abstract
The prime objective of the manufacturing industry is to meet the non-decreasing demand of customers with quality products. Evaluation of optimal job sequence can largely increase the productivity, thereby fulfilling the aim of an enterprise. Usually, determining the optimal job sequence is an arduous task due to the fact that unit increase in an input exponentially increases the problem size. Scheduling problems is a case of non-deterministic polynomial (NP) hard problem which implies it is impractical to compute optimal job sequence within feasible time. Therefore, in the present study, a novel two-phase heuristic algorithm is proposed for multi-stage scheduling problem. The first phase of the proposed model is to compute the job and machine priority. Job priority is the measure of the total work remaining and time taken for processing and completion of the job. On the other hand, machine priority determines the machine that shall take the job for processing. The job prioritization is computed by hybridizing the completion time (CT), processing time (PT) and total work remaining (TWR) for a job. Whereas, machines in each stage of the multi-stage scheduling problem is prioritized by a novel multi-criteria decision-making (MCDM) method which is based on the concept of risk minimization. In the proposed MCDM model, the risk is defined as the loss for choosing an unreliable machine to process a job. The second phase of the proposed method involves assigning of the jobs to the machines based on their priority. The potentiality of a proposed algorithm lies in the practicality and robustness of the model. Hence, it is applied in a flexible flow-shop scheduling (FFSS) problem of a medium-sized manufacturing industry. The performance of the model is statistically tested by Wilcoxon signed-rank test on the basis of make-span and execution. Finally, the proposed approach is validated by comparing the result with some benchmark problems from the literatures.

Keywords Flexible flow shop · Parallel machine scheduling · Heuristic technique · Job and machine prioritization · Fuzzy processing time

1 Introduction

In this technology-driven era, the sustainability of an industry is solely dependent on the customer-based market where the paradigm of an ideal product is often changing (Kim et al. 2020), and the demand for customized products is at its record height (Novshek and Thoman 2006). To thrive in this environment where the manufacturers' prime objective is to meet the ever-increasing demand of the customers with best quality, a new research dimension has been added to the field of industrial engineering. The prime aim of this branch of engineering is to increase the productivity of the industry (Sakamoto 2010). Ascertaining the optimum job-processing sequence is one of the many ways for improving productivity (Arashpour et al. 2016). The method for computing the optimal sequence is called

✉ Syed Abou Iltaf Hussain
syedaboui8@gmail.com

Ranbir Kalita
ranbirkalita@gmail.com

Uttam Kumar Mandal
uttamkumar_mandal@yahoo.co.in

1 Apex Institute of Technology, Department of Management, Chandigarh University, Sahibzada Ajit Singh Nagar, Mohali 140413, Punjab, India

2 Mechanical Engineering Department, Garmur, Jorhat 785007, Assam, India

3 Department of Production Engineering, National Institute of Technology Agartala, Jirania, Agartala 799046, Tripura, India

sequencing and scheduling (SS), which involves the integration of engineering and numerical computation with managerial activities (Kumar et al. 2017). Sequencing is defined as assigning of jobs on a group of machines whereas scheduling consists in arranging, controlling and optimizing work and workloads in a manufacturing unit.

In industry, SS aims in maximizing efficiency and free flow of production and also focuses on minimizing the make-span and cost of production (Mokhtari and Hasani 2017). The determination of this optimum sequence facilitates the efficient use of its resources viz. processing machines, available floor space, material handling and storage, work hour, etc. It also reduces the wastage of time due to lack of co-ordination and thereby enables the company to deliver the products to its end-users at the promised date. SS also establishes inter-department co-ordination (Sly et al. 2017).

SS are broadly classified into two categories: 1. Job shop scheduling (JSS) and 2. Flow shop scheduling (FSS). In general, if a set of jobs are to be processed in a set of machines such that each job has a pre-specified order or route of visit on machines such type of schedule is called JSS. Whereas if every job follows the same route of visit on machines, such type of schedule is called FSS. In an industrial scenario, encountering of single machine is a rare sight to witness. In a multi-stage production shop, it is a very common approach to install identical parallel machines that can operate the same set of operations at every stage in the production shop (Naderi et al. 2014). The objective of such an arrangement is to increase the floor-space utilization and decrease the bottleneck formation. Integration of parallel machines with FSS is called Flexible flow-shop scheduling (FFSS) (Neufeld et al. 2016). FFSS is a generalized form of FSS problems and it is capable of processing more number of jobs at time. Due to these properties FFSS is mostly applied in manufacturing industries. FFSS is widely used in industry that does batch production such as printer manufacturing industry, fabrication industry, car repairing, circuit board manufacturing industry etc. Above that, FFSS is characterized with unlimited buffer space between different stages (Almeder and Hartl 2013). Hence, FFSS problems from an industry perspective are of keen research interest to the present-day researchers and academicians maximizing efficiency and productivity of an industry.

Recently, many researchers and academicians are attracted to the objective of increasing productivity by minimizing the maximum completion time (CT) called make-span (Gao et al. 2016), total lateness or tardiness and number of tardy jobs (Gholami and Zandieh 2009). These performance parameters in a scheduling problem are a function of processing time (PT) which is defined as the time taken for processing a job on a machine (Choi et al. 2010). In most of the literature, PT is considered deterministic, which implies that the time taken for processing a job in every production cycle on a machine is known with absolute accuracy. However, in the practical scenario, there is some uncertainty associated with the PT. The reason for uncertainty in PT is due to the different factors affecting the physical nature of the scheduling problem such as the efficiency of the workers, disruption during machining, loading and unloading time of jobs, machine breakdown, etc. contributes to the uncertainty in PT (Ahmadi et al. 2016).

Zadeh 1965 developed the concept of fuzzy sets (FS) which was adopted by (Prade 1979) to quantify the uncertainty in PT to develop the fuzzy PT (FPT). Dubois and Prade (1982) utilized the FPT and integrated it with the scheduling algorithm developed in (Erschler et al. 1976) to develop the first FPT-based heuristic algorithm. Since then a lot of research development is made. Some of the recent notable literature that provides a comprehensive view about FPT-based heuristic algorithm can be found in (Behnamian 2016; Liu et al. 2017; Arık and Toksarı 2018).

FPT is applied for scheduling identical parallel machines with different capacities (Jia et al. 2019), uniform parallel machines (Li et al. 2019) and non-identical parallel machines (Alcan and BaşLıGil 2012). For a multi-stage production shop, where the path of every job is pre-determined when hybridized with such arrangement is identified as flexible flow-shop scheduling (FFSS). Salvador (1973) was the first to study the FFSS problem. Some of the state-of-the-art literature on the development of FFSS problems can be found in (Ruiz and Vázquez-Rodríguez 2010; Neufeld et al. 2016).

Initially, researchers focused on developing exact methods for solving scheduling problems (Tran et al. 2016) but optimization for FFSS problems are the case of NP-hard problem (Wang and Li 2002). Therefore, researchers shifted their focus towards developing heuristic algorithms to compute a near-optimal schedule for the problems. However, unit increase in the input may exponentially slow down the devised technique. Therefore, researchers shifted their focus on developing fast heuristic techniques that can return a good schedule but not optimal (Asadzadeh and Zamanifar 2010).

In multi-stage scheduling, one of the essential decision-making is to prioritize the jobs that are waiting for processing on a machine. Such type of decision-making regulations is termed as dispatching rule. Shortest processing time (SPT) rule (Schultz 1989) is the simplest form of dispatching rule. According to this rule, the job with the shortest PT will be processed first in a machine. Unlike SPT, longest processing time (LPT) rule prioritizes the job that requires the highest time for processing (Della and Scatamacchia 2020). One of the most common forms of dispatching rules applied in multi-stage medium enterprise is the first come first serve (FCFS) rule. In this rule, the job that arrives first in the working station is taken up for processing irrespective of its importance (Schwiegelshohn and Yahyapour 1998). Earliest release date (ERD) is similar to FCFS rule. The ERD rule is mostly applied when the prime objective of the schedule is to minimize the waiting time of jobs (Pinedo and Chao 1999). Total work remaining (TWR) is a dispatching rule that prioritize the job on the basis of work remaining. Number of unfinished parts (NUP) is a dispatching rule, where the job with maximum unfinished parts is preferred over other jobs. Earliest completion time (ECT) and latest completion time (LCT) are two dispatching rules associated with completion time. The former rule prioritize the job that is expected for earlier completion; whereas, the later prioritize the job that is expected for later completion (Jungwattanakit et al. 2008). Earliest due date (EDD) rule prioritize the jobs in increasing order of due dates. The EDD rule is mostly applied when the objective of the schedule is to minimize the maximum tardiness (Pinedo and Chao 1999). Two variations of the EDD rule are minimum slack time rule (MST) and slack time per processing time rule (SP). Slack time is defined as the difference between the due dates and processing time required. In the MST rule, the job with the least slack time is preferred for processing over other jobs. In SP rule, the quotient obtained by dividing slack time by processing time is arranged in non-decreasing order and prioritized accordingly. It is shown in the paper (Kaban et al. 2012) that hybridizing PT and TWR and combination of TWR and CT works best for minimizing the job-flow time and make-span, respectively. Jungwattanakit et al. (2008) proposed a heuristic algorithm by hybridizing the SPT and EDD rule for minimizing make-span and total number of tardy jobs. There exist several survey literature on dispatching rules that are applied to multi-stage scheduling problems (Nguyen et al. 2017a, b).

Moreover, many new dispatching rules were formulated for prioritizing jobs. Priority dispatching rule falls in the category of completely reactive of the three groups as identified in (Ouelhadj and Petrovic 2009). Some of the priority dispatching algorithms formulated for FFSS environment are Palmer heuristic, Campbell, Dudek, and Smith (CDS) algorithm, Gupta algorithm (GUP), Dannenbring algorithm, Nawaz, Encore and Ham (NEH) heuristics, etc. Palmer heuristic (Palmer 1965) developed the concept of 'Slope Order Index (SOI)' for the jobs. Jobs are arranged in an increasing order of the SOI and the sequencing of the jobs is done accordingly. The Palmer heuristic is later applied to fuzzy FFSS problem by (Hong and Wang 1999). Campbell et al. (1970) proposed a priority-based dispatching rule termed as CDS algorithm. The algorithm is an extension of the Johnson algorithm. The algorithm creates many schedules for a FFSS problem and then Johnson's algorithm is applied to it. Out of the various schedules, the best computed by Johnson's algorithm is chosen (Gozali 2019). Dannenbring (1977) hybridized the advantages of the CDS and Palmer heuristic. The NEH algorithm developed in the year 1983 considered that the job with highest total operating time must be placed in higher order of the job sequence (Nawaz et al. 1983). Above these, there are many other priority dispatching rule proposed for the FFSS problem which falls within the category of completely reactive. Amin and El-Bouri (2018) defined the term completely reactive as decision-making under the real-time scenario and identified 27 sets of priority-based dispatching rules. Literature where multi-criteria decision-making (MCDM) methods were employed for developing priority-based dispatching rules (Subramaniam et al. 2000; Petroni and Rizzi 2002). For optimizing the performance parameters, dispatching rules were hybridized with different optimization algorithms in the literature (Joo and Kim 2015; Nguyen et al. 2017a, b). Many pieces of literature were reviewed for the present study but limited with the most recent and significant state-of-the-art.

1.1 Motivation and novelties

In a multi-stage manufacturing industry, one of the most important decision-making activity is to choose the next job for processing in a machine. But most of the time, it is been observed that choosing a wrong job may affect the free flow of production. Above that, for increasing the production capacity of the industry resulted in installation of number parallel machines. Even though the parallel machines are replication of each other, yet their working condition may vary depending upon their state of condition. The product quality is greatly affected depending on the condition of the processing machines. It is observed that the quality of product fails to meet the customer expectation if the jobs are assigned to machines that are deemed unfit or imperfect from the perspective of reliability. The prime need of the manufacturing industry is to design an effective SS algorithm that not only maximizes the efficiency and free flow of production but also increases the production of quality products.

Motivated by the problem, this paper proposes a novel two-phase heuristic scheduling algorithm to compute the performance measures of a flexible flow-shop environment. The proposed scheduling algorithm considers FPT for scheduling the jobs on the machines. FPT is applied to model the uncertainty associated with time taken for processing a job on a machine. The first phase of the method is setting the priority of the jobs and the machines. Job prioritization (JP) decides the sequence in which jobs shall be taken up by the machines in each stage of the FFSS problem. JP is done by hybridizing three well-established dispatching rules viz. CT, PT and TWR. Whereas machine prioritization (MP) is a way of deciding which available machine shall take up the jobs for processing by abiding the assumption that no machine shall remain idle in any stage of the FFSS environment. MP is a decision-making process which is to be evaluated based on several criteria. Hence, MCDM method became handy in setting machine preference in each stage. A novel MCDM model is proposed in the study, which is based on the concept of risk minimization. The model perceives risk as choosing an unreliable machine over the reliable one. The second phase of the heuristic algorithm is assigning of the jobs to the appropriate machines. The proposed approach is applied to schedule a 72-job and 5-stage FFSS problem of a medium-sized manufacturing industry. The performance of the model is statistically tested by Wilcoxon signed-rank test on the basis of make-span and execution. Also, the feasibility of the proposed heuristic algorithm is tested by comparing the results of the performance measures with that obtained from some other established heuristic algorithm. Moreover, an in-depth study about the heuristic algorithm is conducted, and the important points that are observed are explained elaborately in this paper.

The organization of the remaining paper is done as: Sect. 2 recalls some preliminary concepts and their definitions followed by Sect. 3 that describes the case study. Section 4 describes the methodology proposed, which is followed by results and discussions obtained for the case study in Sect. 5. Finally, Sect. 6 is the conclusion of the paper.

2 Preliminaries

In this section, we shall discuss the key concepts employed for developing the heuristic proposed in the article. We shall start by recalling a few definitions.

2.1 Definitions

2.1.1 Triangular fuzzy number

In the universe of discourse U, $\widetilde{A}$ is termed as FS, if it is typified by membership value $\left(\mu_{\widetilde{A}}\right)$ that maps every element of U to a real-valued number in $[0, 1]$ (Hussain et al. 2018).

$$\widetilde{A} = \left\{x, \mu_{\widetilde{A}}(x) | x \in U\right\},$$

where $\mu_{\widetilde{A}}(x)$ denotes the membership value of $x \in U$. $\widetilde{A}$ is called TFN if it is represented by triplets $(\underline{a}, a, \overline{a})$ such that $(\underline{a} \le a \le \overline{a})$ and the triplets represent a fuzzy subset of a real line whose membership value is defined as (Molinari 2016):

$$\mu_{\tilde{A}}(x) = \begin{cases} \dfrac{x-\underline{a}}{a-\underline{a}}, & \underline{a} \le x \le a \\ \dfrac{\overline{a}-x}{\overline{a}-a}, & a \le x \le \overline{a} \\ 0, & otherwise. \end{cases}$$

2.1.2 Arithmetic operations and defuzzification of TFN

Arithmetic operations for two TFNs and defuzzification of TFN by the center of gravity approach are defined in (Hussain et al. 2019). The expression for inverse operation and defuzzification of a TFN $\widetilde{A} = (\underline{a}, a, \overline{a})$ in Eqs. (1–5).

$$\text{Inverse of}: {}^{1}/_{\tilde{A}} = \left({}^{1}/_{\overline{a}}, {}^{1}/_{a}, {}^{1}/_{\underline{a}}\right), [(\underline{a}, a, \overline{a}) > 0], \tag{1}$$

$${}^{1}/_{\tilde{A}} = \left({}^{1}/_{\underline{a}}, {}^{1}/_{\overline{a}}, {}^{1}/_{a}\right), [\underline{a}\langle 0; (a, \overline{a})\rangle 0], \tag{2}$$

$${}^{1}/_{\tilde{A}} = \left({}^{1}/_{a}, {}^{1}/_{\underline{a}}, {}^{1}/_{\overline{a}}\right), [(\underline{a}, a)\langle 0; \overline{a}\rangle 0], \tag{3}$$

$${}^{1}/_{\tilde{A}} = \left({}^{1}/_{\underline{a}}, {}^{1}/_{a}, {}^{1}/_{\overline{a}}\right), [(\underline{a}, a, \overline{a}) < 0]. \tag{4}$$

$$\text{Defuzzification of } \tilde{A}: A = \frac{1}{3}(\underline{a} + a + \overline{a}), \tag{5}$$

where A is the defuzzified value of $\widetilde{A}$.

2.2 Processing time measurement

The PT considered in this paper is the sum of loading time, machining time and unloading time. The loading time of a job is defined as the time taken to set up a job on $thel^{th}$ machine of the k^{th} stage of the job-processing shop. Machining time is defined as the time taken to machine the loaded job, whereas unloading time is defined as the time taken to unload the machined job. Since job setup, machining and changeover are conducted manually. Therefore, the time taken for processing a similar job on the same machine varies for different cycles. Hence, job PT is uncertain. Therefore, FSs is used to model the uncertainty associated with job PT. TFN is employed to quantify the uncertainty associated with the PT. The simplicity of the membership function is the prime reason for applying TFN the fuzziness of PT.

In the study, fuzzy PT is represented as $\widetilde{t} = (\underline{t}, t, \overline{t})$. The triplet of fuzzy PT represents the optimistic time, most likely time and pessimistic time taken for processing a job on a machine in definite stage. For computing the triplet of $\widetilde{t}$, the time taken for processing ten similar jobs on the same machine at the same stage is recorded. The triplets are computed using Eqs. (6–8).

$$\underline{t} = min\{t_1, t_2, t_3 \cdots t_{10}\}, \tag{6}$$

$$t = \frac{1}{10}\sum(t_1 + t_2 + t_3 + \cdots + t_{10}), \tag{7}$$

$$\bar{t} = max\{t_1, t_2, t_3 \cdots t_{10}\}. \tag{8}$$

The most likely value t has the highest membership value $\mu_{\tilde{A}}(x) = 1$, where $x = t$. The value of membership functions $\mu_{\tilde{A}}(x)$ decreases uniformly when the value of x approaches $\underline{t}$ and $\bar{t}$. Since PT is considered in a fuzzy environment, hence, all the scheduling parameters considered in the study that are functions of PT are fuzzy. Ahmad and Cheng (2022) applied TFN to quantify the uncertainty in the processing time to develop a fuzzy control chart. (Kang et al. 2023) and (Zhou et al. 2022) developed a hybrid optimization and Pareto-based discrete optimization algorithm, respectively, for solving problems with FPT.

3 Case study

The problem considered in this paper is a case study of 72-job and 5-stage FFSS problem of a medium-sized manufacturing industry. The first major department is the manufacturing shop where the 72 jobs are produced in 5 manufacturing units. The manufacturing of the jobs is initiated by a Kanban system. The jobs are then passed to the second department that is the machining and job-processing shop. The job-processing shop is divided into 5 stages and each stage comprises of different numbers of identical parallel machines. The jobs from each manufacturing unit follow a unique route of visit on the machines. Thereby making the scheduling of jobs in the job-processing shop a FFSS problem. The jobs are processed in batches and continuous in nature. Since the time taken for setting up of a job, machining the job and changeover are conducted manually. Therefore, the time taken for processing similar jobs on the same machine varies for different cycles. There is a fuzziness associated with the processing time. The bottleneck formation of job flow in the industry is materialized because of this department. The jobs coming out of this department go to the third department, i.e., the pre-fitting. The detailed job-flow diagram of the job-processing shop is shown in Fig. 1.

3.1 Problem description

In this study, the objective of the scheduling problem under consideration is to achieve a continuous and smooth flow of parts to the pre-fitting area. This objective can be obtained by minimizing the value of make-span, total tardiness and total number of tardy jobs. J^1, J^2, J^3, J^4 and J^5 are a collection of 16, 17, 22, 8 and 9 jobs that are manufactured in 5 manufacturing units.

The total number of schedules possible for FFSS problem considered in this paper is the sum of the schedules for each stage. For a FFSS problem having n-stages with m-machines in each stage needed to process j-jobs the total number of schedules is computed according to Eq. (9).

$$\text{Total number of schedules} = \left[\sum_n (j!)^m\right]. \tag{9}$$

The total number of schedules computed according Eq. (9) is 4.746×10^{99}.

3.2 Assumptions made for the problem considered

To achieve the aim of the scheduling problem, certain assumptions that are made for processing of the jobs are as follows:

i. All the machines, in a stage, are capable of performing the same set of operations on the jobs.
ii. The sequential order of job flow is pre-defined and strictly follows it, as shown in Fig. 1.
iii. Processing of a job once started on a machine in a stage it will not pass to the next stage unless it is completed in the present stage.
iv. No pre-emption of the job is allowed in any stage of the job-processing shop.
v. Splitting of jobs is not allowed.
vi. All the 72 semi-finished jobs are readily available for processing before starting of the current cycle.
vii. All the machines in every stage are available before starting the current cycle.
viii. No machine will remain idle if jobs are available.

3.3 Proof of NP-hard

In the literatures, it is established that make-span optimization for FFSS environment is non-deterministic polynomial (NP) complete problem. It implies that the problem is both NP and NP-hard. Therefore, there does not exist a polynomial time algorithm for solving such type of problem (Jungwattanakit et al. 2008).

The problem considered in the study can be proved as NP-hard if and only if all NP-hard problem is effectively reducible to the present problem in polynomial time. However, the easiest way of proving that the present problem is NP-hard by reducing a known FFSS problem into the present problem in polynomial time by polynomial transformations also known as Karp reduction.

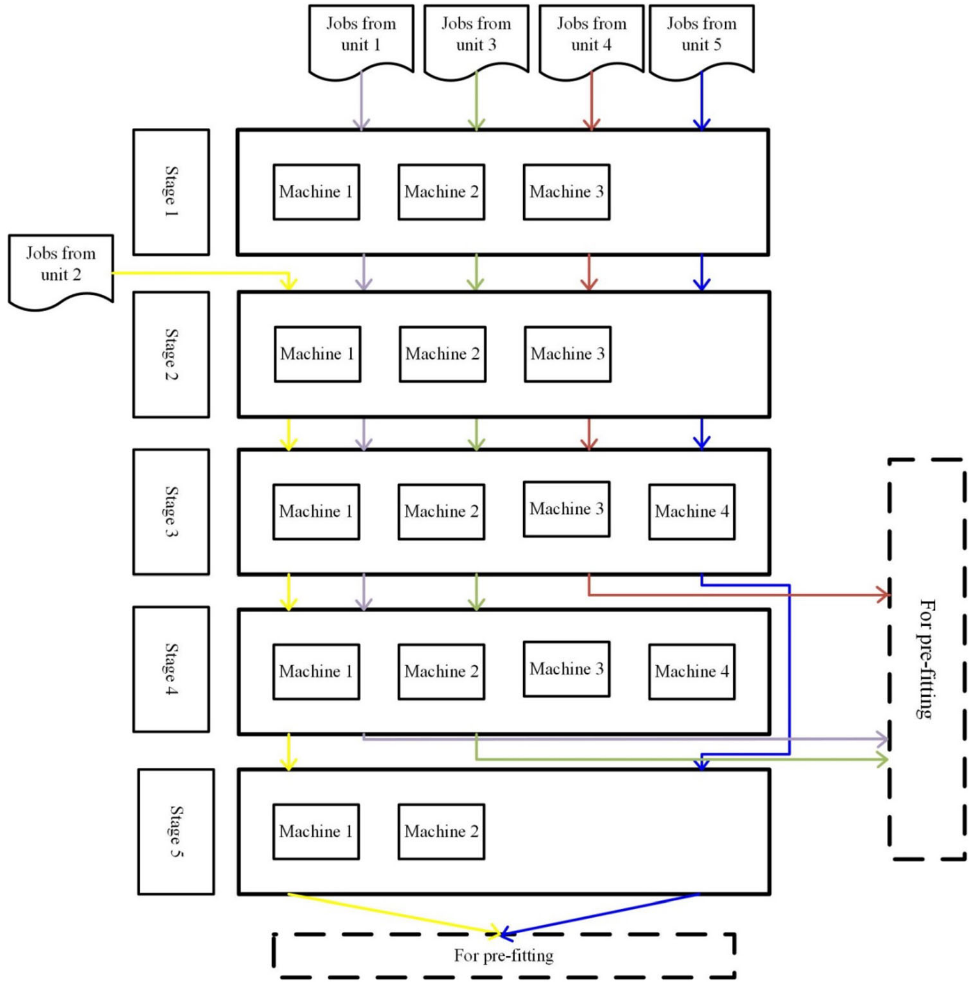


Fig. 1 Detailed job-flow diagram of job-processing shop

Considering $A$and$B$ is used to denote a general FFSS and the problem considered in this study, respectively. The steps of proving B as NP-hard can be summarized as follows (Bürgisser 2013):

- Transform inputs: Transform the inputs for problem $A(I_A)$ into the inputs for $B(I_B)$ in polynomial time represented as $I_A \underset{P}{\rightarrow} I_B$.
- Applying blackbox for solving problem B: Considering that there exists a blackbox that is capable of solving both the problem A as well as$B$.
- Transform output: Transforming the output for problem $B(O_B)$ into the output of problem $A(O_A)$ in polynomial time represented as $O_B \underset{P}{\rightarrow} O_A$.

Following the above steps, input for problem A is identified as number of stages, number of machines in each stage and number of jobs processing. After transforming the inputs of A into inputs of B are tabulated in Table 1.

Considering that there exists a polynomial time algorithm that can compute the value of make-span for problem A and the value is C (say). The polynomial time algorithm considered in the study is developed by (Aydilek and Allahverdi 2013) for problem A. When applied the same polynomial algorithm, the make-span value computed for the problem B is 2071 min, i.e., $C = 2071$ minutes. Thus, the known NP-hard problem A is reduced to problem B. Hence, it can be concluded that the problem considered in this study is also a NP-hard problem.

4 Proposed model

The comprehensive intention of the paper is to develop a heuristic algorithm that is efficient as well as effective in returning a nearly optimal result for the FFSS problem. In such type of scheduling problem, the most difficult task is computing the optimal job sequence and allocating the jobs to the appropriate machines simultaneously (Chen 2004). Hence, a heuristic algorithm is developed that focuses on job scheduling by allocating the resource to the most appropriate machine in each stage of an FFSS problem. The proposed heuristic is a double-phase procedure that

Table 1 Inputs of B in the form of inputs of A

Sl. No	Stage (k)	MCs in the stage (M^k)	Jobs passing through the stage	Total number of jobs
1	1	3	J^1, J^3, J^4 and J^5	55
2	2	4	J^1, J^2, J^3, J^4 and J^5	72
3	3	4	J^1, J^2, J^3, J^4 and J^5	72
4	4	3	J^1, J^2 and J^3	55
5	5	2	J^1 and J^5	25

initially prioritizes the jobs and machines and finally, allocates the jobs to the machines in order of the computed preference score.

The first phase of the proposed heuristic algorithm is the prioritization of the jobs and the machines in each stage. JP is a machine-independent way of deciding the next job in the queue, waiting for processing (Chua et al. 2011). Certain disadvantages are identified in the literature for the dispatching rules that consider one factor for prioritizing jobs. Therefore, in this paper, an attempt is made to prioritize the jobs by combining the three factors CT, PT and TWR.

The second step of the first phase is the prioritization of the machines in each stage of the job-processing shop. It is a way of deciding the machine that shall take the next job for processing in case two or more machines are available at the moment (Chua et al. 2011). Selecting the suitable machine for processing the jobs is an MCDM problem that involves decision-making from the point of view of reliability, productivity, efficiency, revenue generation and total costing (Karim and Karmaker 2016). From the literature survey, the criteria chosen for selecting machine are tabulated in Table 2.

The final phase of the proposed heuristic is sequencing the jobs with the most suitable machine available in order of the preference score. Before explaining the proposed approach, we shall start by defining the symbols and notations that are used for the formulation. The list of symbols used in formulating the heuristic is shown in Table 3.

4.1 Job prioritization

JP is machine-independent rules for deciding the next job in the sequence. In this study, the jobs priority is determined based on the preference score for the jobs which is computed according to the Eq. (10).

$$\widetilde{\wp}_{(j)} = w_1.\widetilde{C}_{(j)} + w_2.\widetilde{t}_{(j)} + w_3.\widetilde{T}_{(j)}, \tag{10}$$

where w_1, w_2andw_3 are the weights for the CT, PT and TWR, respectively. The computed value of $\widetilde{\wp}_{(j)}$ is in the form of TFN which is defuzzified, according to Eq. (5). The job with the highest preference score is given the priority for processing on the available machine from a set of jobs that are awaiting service. Higher the value of preference score, more prioritization is given to the job as it implies more work remaining, more time taken for processing and completion. The advantage of hybridizing the single-factor rules according to Eq. (10) is that the preference score integrates the conceptual attributes of the individual dispatching rules to the level of their importance in attaining the objectives. The value of $C_{(j)}$and$T_{(j)}$ is calculated according to Eqs. (11) and (12), respectively.

$$C_{(j)} = \sum_{k=1}^{K} t_{(j;k)}, \tag{11}$$

$$T_{(j)} = \sum_{k=k}^{K} t_{(j;k)}. \tag{12}$$

Table 2 List of criteria for machine selection

Sl. no	Criterion	Sl. no	Criterion	Sl. no	Criterion
1	Setup time (Cr_1)	2	Parts cost (Cr_2)	3	Maintenance cost (Cr_3)
4	Time taken for repairing (Cr_4)	5	Frequency of damage (Cr_5)	6	Time taken for inspection (Cr_6)
7	Ergonomically designed (Cr_7)	8	Design safety (Cr_8)		

Table 3 List of symbols

Sl. No	Symbols	Definition
1	C_{max}	Make-span
2	τ	Total lateness or tardiness
3	δ	Number of tardy jobs
4	i	Indexing for manufacturing unit
5	j	Indexing for job
6	k	Indexing for stage
7	K	Total number of stages in the job-processing unit
8	l	Indexing for machine
9	(i,j)	j^{th} job coming out of i^{th} manufacturing unit
10	(k,l)	l^{th} machine of the k^{th} stage
11	$(i,j;k,l)$	Processing of the j^{th} job coming out of i^{th} manufacturing unit done in l^{th} machine of the k^{th} stage
12	$(j;k)$	Processing of the j^{th} job on the k^{th} stage
13	$\wp_{(j)}$	Preference score of j^{th} job
14	$C_{(j)}$	Completion time for j^{th} job
15	$t_{(j)}$	Processing time for j^{th} job
16	$T_{(j)}$	Total work remaining for the j^{th} job
17	p	Indexing for decision maker
18	α	Total number of decision makers
19	q	Indexing for criterion
20	m	$\text{Totalnumberofcriteria}.m = \begin{cases} 3; & \text{Job prioritization} \\ 8; & \text{Machine prioritization} \end{cases}$
21	M^k	It is a parametric term whose value depends on the value of k. It represents total number of machines in k^{th} stage. $k \in [1, K]$
22	D	Decision matrix
23	d	Element of decision matrix D
24	$\mathbb{R}_{pql}$	Rating provided for the l^{th} machine on the basis of q^{th} criterion by p^{th} decision maker
25	R	Relative cost matrix
26	r	Element of relative cost matrix R
27	N	Normalize matrix
28	n	Element of normalize matrix N
29	W	Weighted normalize matrix
30	ω	Element of weighted normalize matrix W
31	w_q	Weightage of the q^{th} criterion
32	$\wp$	Preference score
33	z_q	Aggregated rating
34	u	Number of jobs awaiting
35	v	Number of machines available
36	$\wp_l$	Preference score for the l^{th} machine
37	$l_\wp$	Machine with preference score of $\wp$
38	$j_\wp$	Job with preference score of $\wp$
39	J^i	Collection of semi-finished jobs or parts coming out of i^{th} manufacturing unit

4.2 Machine prioritization

Machine prioritization (MP) is a way of ranking the machines available in each stage based on their reliability. MP determines the machine preference for processing a job when more than one machine is available at a time. Ranking of the available machines based on the preference score is a case of multi-criteria decision-making (MCDM) problem. Computation of the preference score for the machine is done based on the criteria as listed in Table 2 which involves making a decision based on the factors which are quantitative as well as qualitative in nature. To simplify the rating process, linguistic terms are carried out to assess the machines concerning the identified factors. Based on the experience of the decision makers, linguistic ratings are provided to the alternatives. The two main reasons for vacillation that the decision makers faced are the existence of vagueness and uncertainties in the available information about the factors and the psychological perception of the decision makers about the factors. Due to these two reasons, linguistic terminologies are the best way of rating the alternatives (Hussain et al. 2018). Linguistic terminologies are quantified using TFNs which are used for rating the alternatives (Feng et al. 2022). In the state-of-the-art literature, fuzzy sets and logic are integrated with MCDM models to develop ranking algorithm that is capable of taking into account the uncertainty that rose due to the vagueness in the nature of the problem. Some of the papers on fuzzy MCDM include (Hussain et al. 2018; Hussain et al. 2019; Akram and Bibi 2023; Luo et al. 2023).

The criteria chosen for the process of MP is mostly related with the operational, functioning and repairing of the machines. Therefore, the operators and machinists who operate and repair the machines are chosen as the decision makers. The view of the decision makers is aggregated to form the fuzzy decision matrix. The corresponding *TFNs* for the linguistic ratings and the linguistic rating-based decision matrix are shown in Tables 4 and 5, respectively.

The steps for computing the preference score of the machines in each stage are as follows:

Step 1: Aggregation of the fuzzy decision matrix

First, the linguistic ratings provided by the decision makers are quantified using *TFNs*. Then, the fuzzy decision matrix of each decision maker is aggregated to form the aggregated fuzzy decision matrix $\left(\widetilde{D}\right)$ which is computed as follows:

$$\widetilde{D} = \left[\widetilde{d}_{ql}\right]_{(m\times M^k)} = \left[\frac{\sum_{p=1}^{\alpha}\left(\widetilde{\mathbb{R}}_{pql}\right)}{\alpha}\right]_{(m\times M^k)};q\in[1,m],l\in\left[1,M^k\right]. \tag{13}$$

Step 2: Formulation of fuzzy relative cost matrix

The fuzzy relative cost matrix $\left(\widetilde{R}\right)$ is derived from the concept of risk minimization. The concept of risk pertains to for not selecting the available machine with maximum reliability. The computation of $\widetilde{R}$ is done as follows:

$$\widetilde{R} = \left[\widetilde{r}_{ql}\right]_{(m\times M^k)} = \begin{cases} \left[\max\limits_q\left(\widetilde{d}_{ql}\right) - \widetilde{d}_{ql}\right]_{(m\times M^k)}, & Forbenefit-criteria \\ \left[\widetilde{d}_{ql} - \min\limits_q\left(\widetilde{d}_{ql}\right)\right]_{(m\times M^k)}, & Forcost-criteria. \end{cases} \tag{14}$$

Step 3: Formulation of weighted normalize matrix

The weighted normalize matrix $\left(\widetilde{W}\right)$ is computed by multiplying the elements of normalize matrix and weight of the criteria. The factors based on which machines are prioritized different units, and as a result, the values are incomparable. Hence, this obstacle is tackled by normalizing the factors. Normalization of the entries of $\widetilde{R}$ is computed as follows:

$$\widetilde{N} = \left[\widetilde{n}_{ql}\right]_{(m\times M^k)} = \left[\frac{\widetilde{r}_{ql}}{\sqrt{\sum_{l=1}^{M^k}\left(\widetilde{r}_{ql}^{\,2}\right)}}\right]_{(m\times M^k)}; \quad q=\{1,2,3\cdots m\}. \tag{15}$$

The weight represents the degree of importance of the criterion for reaching at a rationale conclusion. The elements of weighted normalize matrix is computed as follows:

$$\widetilde{W} = \left[\widetilde{\omega}_{ql}\right]_{(m\times M^k)} = \left[w_q * \widetilde{n}_{ql}\right]_{(m\times M^k)}. \tag{16}$$

Step 4: Calculation of preference score

The formula for computing the preference score is as follows:

Table 4 Corresponding *TFNs* for linguistic variables

Linguistic variables	Very low (VL)	Low (L)	Medium (M)	High (H)	Very high (VH)
TFN	(0, 1, 2)	(2, 3, 4)	(4, 5, 6)	(6, 7, 8)	(8, 9, 10)

Table 5 Linguistic decision matrix

Alt	S_1			S_2			S_3				S_4				S_5	
	MC-1	MC-2	MC-3	MC-1	MC-2	MC-3	MC-1	MC-2	MC-3	MC-4	MC-1	MC-2	MC-3	MC-4	MC-1	MC-2
Cr_1	H	H	H	VH	VH	H	VH	H	VH	H	VH	VH	VH	H	M	M
Cr_2	M	L	M	L	L	M	L	H	L	L	H	M	H	M	H	VH
Cr_3	M	M	H	VL	VL	VL	H	M	H	VH	VH	VH	H	M	M	H
Cr_4	VH	H	M	VH	M	M	H	H	H	VH	L	VL	VH	VL	M	L
Cr_5	H	H	L	VH	M	L	H	H	M	VH	VL	VL	VH	VL	M	VL
Cr_6	VH	VH	M	VH	M	H	VH	VH	M	H	L	L	H	L	L	L
Cr_7	M	VH	M	L	H	M	M	H	H	L	H	L	L	M	M	L
Cr_8	H	VH	M	H	H	L	H	VH	H	M	M	H	H	M	H	H

$$\widetilde{\wp}_l = \left[\frac{\sum_{q=1}^{m}\left(\widetilde{\omega}_{ql}\right)}{\sum_{l=1}^{M^k}\sum_{q=1}^{m}\left(\widetilde{\omega}_{ql}\right)}\right]_{\left(1\times M^k\right)}; \quad l=\left\{1,2,3\cdots M^k\right\}, k=\{1,2,3,4,5\}. \tag{17}$$

The preference score as evaluated from Eq. (17) is defuzzified according to Eq. (5) and ranked in ascending order which implies lower the value of $\wp_l$ more reliable amongst the machine available for processing the jobs at the moment. Hence, the machine with the least value of $\wp_l$ will be selected to process a job in case more than one machine is available.

4.3 Computation of weight factors for the criteria

The weights signify the degree of importance of the criteria in the process of decision-making. The weights computed is the aggregate of the ratings as assigned by the decision makers. In this study, the panel of decision makers are experts from the domain of production scheduling which includes researcher, production manager, production supervisor and operational head. The experts provide their decision in linguistic rating according to their experiences in the domain of production scheduling which are quantified using *TFNs*. The aggregated rating $\left(\widetilde{z}_q\right)$ provided by the experts were computed according to the Eq. (18).

$$\widetilde{z}_q = \frac{\sum_{p=1}^{\alpha}\left(\widetilde{\mathbb{R}}_{pq}\right)}{\alpha}; q=\{1,2,3,\cdots,m\}, \tag{18}$$

where $\widetilde{\mathbb{R}}_{pq}$ represents the corresponding *TFN* for the linguistic rating given by p^{th} decision maker for the q^{th} criterion. The corresponding fuzzy ratings for linguistic terms are shown in Table 6. In Eq. (18), α represents the total number of decision makers in the panel and m is the no. of criteria, $m=3$ for JP and $m=8$ for MP in each stage. The $\widetilde{z}_q$ is defuzzified according to Eq. (5), which is then normalized to evaluate the weight for the criteria.

$$w_q = \frac{z_q}{\sum_{q=1}^{m}\left(z_q\right)}; q=\{1,2,3,\cdots m\}. \tag{19}$$

The fuzzy ratings for the criteria provided by the panel of decision makers for evaluating the weights for prioritizing the jobs and the machines are tabulated in Tables 7 and 8, respectively.

Aggregating ratings provided by the panel of decision makers and experts in the domain of production scheduling for computing the weights of the criteria for JP and MP are shown in Table 9.

4.4 Procedure for assigning of jobs to the machines

The final phase of the proposed heuristic is assigning of the jobs to the machines in the FFSS problem. Four cases may arise during the mapping of jobs to the machines:

Case i: If $\boldsymbol{u}=1$ and $\boldsymbol{v}=1$

In this case, job awaiting (j) is assigned to the available machine (l) which is represented as (j,l).

Case ii: If $\boldsymbol{u}=1$ and $\boldsymbol{v}>1$

In this case, the machine with the lowest preference score is given the priority for processing the available job. If

$$\wp_{l_{min}} = min\left\{\wp_{l_1}, \wp_{l_2}, \wp_{l_3}, \cdots, \wp_{l_{M^k}}\right\}.$$

Then the assignment is done as $\left(j, l_{\wp_{min}}\right)$, where $l_{\wp_{min}}$ is the highest prioritized machine.

Case iii: If $\boldsymbol{u}>1$ and $\boldsymbol{v}=1$

In this case, the job with the highest preference score is given priority to be the next job in the queue waiting to be processed.

Table 6 Corresponding fuzzy ratings for linguistic terms

Linguistic terms	Highly insignificant	Insignificant	Moderately significant	Significant	Highly significant
Fuzzy rating	$\widetilde{1}$	$\widetilde{2}$	$\widetilde{3}$	$\widetilde{4}$	$\widetilde{5}$
TFN	$(1,1,2)$	$(2,3,4)$	$(4,5,6)$	$(6,7,8)$	$(8,8,9)$

Table 7 Fuzzy ratings assigned to factors for JP

Factors	*DM*			
	DM_1	DM_2	DM_3	DM_4
CT	$\widetilde{5}$	$\widetilde{5}$	$\widetilde{5}$	$\widetilde{5}$
PT	$\widetilde{3}$	$\widetilde{4}$	$\widetilde{5}$	$\widetilde{3}$
TWR	$\widetilde{4}$	$\widetilde{5}$	$\widetilde{4}$	$\widetilde{4}$

Table 8 Fuzzy ratings assigned to criteria for MP

Decision maker	Cr_1	Cr_2	Cr_3	Cr_4	Cr_5	Cr_6	Cr_7	Cr_8
DM_1	$\widetilde{4}$	$\widetilde{5}$	$\widetilde{5}$	$\widetilde{3}$	$\widetilde{5}$	$\widetilde{2}$	$\widetilde{4}$	$\widetilde{4}$
DM_2	$\widetilde{3}$	$\widetilde{5}$	$\widetilde{5}$	$\widetilde{3}$	$\widetilde{5}$	$\widetilde{1}$	$\widetilde{4}$	$\widetilde{3}$
DM_3	$\widetilde{2}$	$\widetilde{5}$	$\widetilde{5}$	$\widetilde{4}$	$\widetilde{3}$	$\widetilde{1}$	$\widetilde{5}$	$\widetilde{5}$
DM_4	$\widetilde{2}$	$\widetilde{5}$	$\widetilde{5}$	$\widetilde{5}$	$\widetilde{4}$	$\widetilde{2}$	$\widetilde{5}$	$\widetilde{5}$

$$\wp_{j_{max}} = min\left\{\wp_{j_1}, \wp_{j_2}, \wp_{j_3}, \cdots, \wp_{j_{ji}}\right\}.$$

The assignment of a job to a machine is done as $\left(j_{\wp_{max}}, l\right)$, where $j_{\wp_{max}}$ is the highest prioritized job.

Case iv: If $\boldsymbol{u} > 1$ and $\boldsymbol{v} > 1$

In this case, the preference score for both jobs and machines shall come into play. Highest prioritized job is assigned to the highest prioritized machine, and the second-highest prioritized job is assigned to the second-highest prioritized machine and so on. Considering after time $\widetilde{t}$, jobs j_1, j_2, j_3 and j_4 awaiting processing in a stage where three machines are available l_1, l_2 and l_3. Assuming $\wp_{j_1} \succ \wp_{j_2} \succ \wp_{j_3} \succ \wp_{j_4}$ and $\wp_{l_1} \prec \wp_{l_2} \prec \wp_{l_3}$. The assignment of the jobs to the machines is done as follows $\left(j_{\wp_1}, l_{\wp_1}\right), \left(j_{\wp_2}, l_{\wp_2}\right)$ and $\left(j_{\wp_3}, l_{\wp_3}\right)$. The *CT* for the jobs j_1, j_2 and j_3 are computed as

$$\widetilde{C}_{\left(j_{\wp_1}, l_{\wp_1}\right)} = \widetilde{t} + \widetilde{t}_{\left(j_{\wp_1}, l_{\wp_1}\right)}; \widetilde{C}_{\left(j_{\wp_2}, l_{\wp_2}\right)} = \widetilde{t} + \widetilde{t}_{\left(j_{\wp_2}, l_{\wp_2}\right)}; \widetilde{C}_{\left(j_{\wp_3}, l_{\wp_3}\right)} = \widetilde{t} + \widetilde{t}_{\left(j_{\wp_3}, l_{\wp_3}\right)}.$$

Two sub-cases may arise for assigning of job j_4 in any one of the three machines.

Sub-case a: Job j_4 will be assigned to the machine having the least *CT*

Sub-case b: If minimum *CT* is the same for more than one machine, then assigning of the job will be done according to **Case ii**. The procedure for the solution by the proposed heuristic is shown in Fig. 2.

5 Results and discussion

In this section of the paper, the result obtained for the considered problem after applying the proposed approach is discussed.

5.1 Performance analysis of the proposed model

To corroborate the relative superior performance of the proposed heuristic approach, the performance of the proposed algorithm is compared with the performance of eighteen heuristic algorithms. The algorithms are applied to sixteen different benchmark FFSS problems. The performance table is shown in Table 10.

Subsequently, the Wilcoxon signed-rank test is employed for pairwise comparison to verify the proposed

Table 9 Computed weights of the criteria

JP		MP					
Factor	w_i	Criterion	w_i	Criterion	w_i	Criterion	w_i
CT	0.378	Cr_1	0.087	Cr_2	0.161	Cr_3	0.161
PT	0.289	Cr_4	0.123	Cr_5	0.139	Cr_6	0.042
TWR	0.333	Cr_7	0.148	Cr_8	0.139		

approach's superiority in comparison with other heuristic algorithms. Wilcoxon signed-rank test is a non-parametric dependable sample hypothesis test. This test is used to compare the performance of the repeated measurements on a single sample to analyze the divergence between their population means (Woolson 2007). The null and alternative hypotheses formulated for the Wilcoxon signed-rank test are as follows:

$(H_0)_h : \mu_{PA} = \mu_h$, i.e., there is no significant difference in the performance of the proposed algorithm and heuristic algorithm.

$(H_1)_h : \mu_{PA} \neq \mu_h$, i.e., there is a significant difference in the performance of the proposed algorithm and heuristic algorithm.Here, h in the null and alternative hypotheses stands for the eighteen different heuristic algorithm. For example: $(H_0)_{SPT}$ implies there is no significant difference in the performance of the proposed algorithm and shortest processing time algorithm. Whereas $(H_1)_{SPT}$ implies there is a significant difference in the performance of the proposed algorithm and shortest processing time algorithm.

Table 11 summarizes the results of pairwise comparison for all the heuristics for significance level of 0.05. The table consists of the sum of positive ranks (W +), sum of negative ranks (W−), Wilcoxon test (W) value and p value. The W + values indicate the sum of ranks for the heuristic algorithms for which proposed approach outperformed the compared algorithms and vice versa for W−. On the other hand 'W' measures the pairwise averages that are greater than the hypothesized median. The W− value helps to evaluate the p value. The p value measures the evidence against the null hypothesis.

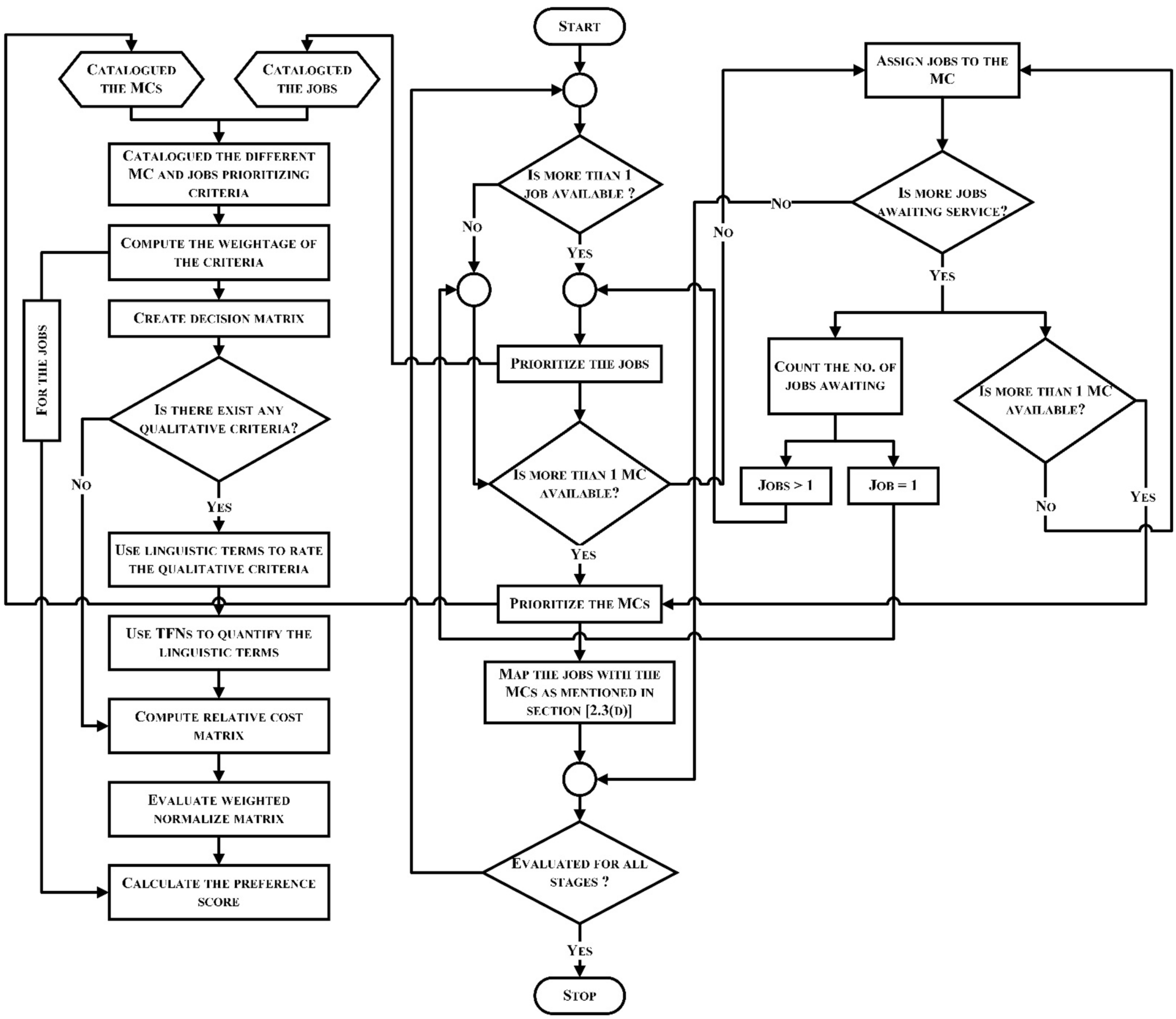


Fig. 2 Proposed heuristic algorithm

Table 10 Performance table

Prob. No	Proposed approach	SPT (Schultz 1989)	LPT (Della Croce and Scatamacchia 2020)	TWR (Karger et al. 2010)	NUP (Sumichrast et al. 1992)	EDD (Pinedo and Chao 1999)	ERD (Pinedo and Chao 1999)
1	770	866	887	870	770	831	790
2	1972	1895	2334	2082	2197	2402	2377
3	984	1228	1169	1111	1160	1022	971
4	596	833	631	690	817	637	592
5	489	580	459	561	590	528	490
6	1772	1951	1971	1730	2195	2148	1802
7	1080	1025	1060	1037	1029	1034	1056
8	988	1114	1496	1392	984	1477	1342
9	496	741	604	821	636	623	732
10	1817	1946	1897	2414	1827	1724	2182
11	1636	1575	1585	1629	1551	1608	1548
12	964	1160	924	1170	1053	1276	1104
13	523	602	540	818	619	538	711
14	1146	1715	1304	1386	1473	1647	1732
15	1056	895	1179	1350	994	1230	1338
16	1113	1120	1062	1078	1051	1113	1073
17	2041.6	2603.3	2543.6	2588.3	2363	2431.3	2072.7

Prob. No	FCFS (Schwiegelshohn and Yahyapour 1998)	ECT (Jungwattanakit et al. 2008)	LCT (Jungwattanakit et al. 2008)	MST (Davis et al. 1993)	S/P (Bari and Karande 2022)	PAL (Palmer 1965)
1	941	781	889	829	937	973
2	1932	2276	2448	2144	1735	1950
3	958	1183	1018	1238	952	1202
4	759	606	806	580	630	786
5	485	480	631	580	425	612
6	2143	1830	2020	1891	1964	2219
7	1095	1051	1053	1060	947	1092
8	1243	1068	1297	1116	1127	1237
9	507	819	668	669	804	572
10	1823	2252	1913	1912	1766	2017
11	1588	1629	1590	1572	1544	1623
12	1388	1229	1006	1322	1371	1195
13	553	856	623	606	676	725
14	1720	1717	1049	1588	1730	1690
15	1189	1151	1110	1267	901	936
16	1072	1086	1122	1081	1046	1123
17	2270.5	2628	2603.6	2357.8	2068	2048.9

Prob. No	GUP (Hong et al. 2000)	NEH (Nawaz et al. 1983)	CDS (Campbell et al. 1970)	PT + CT (Kaban et al. 2012)	CT + TWR (Kaban et al. 2012)	TWR + PT (Kaban et al. 2012)
1	926	778	845	816	743	767
2	2041	2355	1875	2471	2138	1921
3	985	1082	1136	1084	1067	1161
4	713	572	661	564	765	755
5	602	586	489	645	530	464
6	2087	1621	1892	1940	2084	1950

Table 10 (continued)

Prob. No	GUP (Hong et al. 2000)	NEH (Nawaz et al. 1983)	CDS (Campbell et al. 1970)	PT + CT (Kaban et al. 2012)	CT + TWR (Kaban et al. 2012)	TWR + PT (Kaban et al. 2012)
7	1062	1044	1059	1079	1086	1035
8	1357	1211	1094	1710	1502	1631
9	614	856	728	839	699	743
10	1920	2393	1948	2017	1788	2175
11	1575	1535	1541	1611	1541	1636
12	988	1036	887	1434	1251	902
13	757	656	890	579	690	854
14	1230	1549	1527	1134	1127	1341
15	1091	1301	910	1267	982	1168
16	1061	1108	1075	1081	1057	1120
17	2314.2	2366.5	2099.3	2352.3	2284	2307.3

From Table 11, there is enough evidence to reject the null hypotheses and accept the alternate hypotheses for the algorithms SPT, LPT, TWR, NUP, EDD, ERD, FIFO, ECT, LCT, MST, PAL, PT + CT, CT + TWR and TWR + PT for significance level of 0.05. Hence, we can conclude that there is a significant difference in the performance of the proposed algorithm and heuristic algorithm. It implies that the proposed heuristic algorithm provides better solution when applied to FFSS problems.

5.2 Results from the proposed heuristic

In this section, the result obtained from the by applying the proposed model in the problem considered for the study is discussed. The first step of the proposed heuristic is to compute the weightage of the factors and the criteria for prioritizing the jobs and the machines. The calculated weights of the criteria are shown in Table 9. The next step is computing the preference score of the jobs and machines.

In the process of JP, four decision makers are chosen and based on their experience linguistic terms are assigned to the criteria which were quantified using *TFNs*. The three factors that are identified for prioritizing the jobs are CT, PT and TWR. Based on the three factors, preference scores of the jobs are computed using Eq. (9).

For MP in each stage, decision makers assigned linguistic ratings to the machines concerning the criteria which are quantified using *TFNs*. The relative cost matrix and the normalize matrix for each stage of the problem as computed by Eqs. (13) and (14).

STAGE 1

$$R_1 = \begin{array}{c} \\ Cr_1 \\ Cr_2 \\ Cr_3 \\ Cr_4 \\ Cr_5 \\ Cr_6 \\ Cr_7 \\ Cr_8 \end{array} \begin{array}{ccc} m/c_1 & m/c_2 & m/c_3 \end{array} \begin{bmatrix} (-2,0,2) & (-2,0,2) & (-2,0,2) \\ (0,2,4) & (-2,0,2) & (0,2,4) \\ (-2,0,2) & (-2,0,2) & (0,2,4) \\ (2,4,6) & (0,2,4) & (-2,0,2) \\ (2,4,6) & (2,4,6) & (-2,0,2) \\ (2,4,6) & (2,4,6) & (-2,0,2) \\ (2,4,6) & (-2,0,2) & (2,4,6) \\ (0,2,4) & (-2,0,2) & (2,4,6) \end{bmatrix},$$

$$\widetilde{N}_1 = \begin{array}{c} \\ Cr_1 \\ Cr_2 \\ Cr_3 \\ Cr_4 \\ Cr_5 \\ Cr_6 \\ Cr_7 \\ Cr_8 \end{array} \begin{array}{ccc} m/c_1 & m/c_2 & m/c_3 \end{array} \begin{bmatrix} (-0.577,0,0.577) & (-0.577,0,0.577) & (-0.577,0,0.577) \\ (0,1,2) & (-1,0,1) & (0,1,2) \\ (-0.707,0,0.707) & (-0.707,0,0.707) & (0,0.707,1.414) \\ (0.707,1.414,2.121) & (0,0.707,1.414) & (-0.707,0,0.707) \\ (0.577,1.154,1.732) & (0.577,1.154,1.732) & (-0.577,0,0.577) \\ (0.577,1.154,1.732) & (0.577,1.154,1.732) & (-0.577,0,0.577) \\ (0.577,1.154,1.732) & (-0.577,0,0.577) & (0.577,1.154,1.732) \\ (0,0.707,1.414) & (-0.707,0,0.707) & (0.707,1.414,2.121) \end{bmatrix}.$$

Table 11 Wilcoxon signed-rank test table

Heuristic model	W +	W−	W	p	Heuristic model	W +	W−	W	p
SPT	126	10	117	0.012	MST	126	10	125	0.003
LPT	121	15	113	0.021	S/P	100	36	87	0.339
TWR	126	10	126	0.003	PAL	130	6	123	0.005
NUP	121	15	101	0.021	GUP	130	6	105	0.059
EDD	130	6	104	0.013	NEH	121	15	106	0.052
ERD	121	15	111	0.028	CDS	115	21	88	0.118
FIFO	121	15	110	0.032	PT + CT	121	15	121	0.007
ECT	126	10	122	0.006	CT + TWR	115	21	123	0.005
LCT	130	6	121	0.007	TWR + PT	121	15	101	0.021

STAGE 2

$$\widetilde{R}_2 = \begin{array}{c} \\ Cr_1 \\ Cr_2 \\ Cr_3 \\ Cr_4 \\ Cr_5 \\ Cr_6 \\ Cr_7 \\ Cr_8 \end{array} \begin{array}{c} \begin{array}{ccc} m/c_1 & m/c_2 & m/c_3 \end{array} \\ \begin{bmatrix} (0,2,4) & (0,2,4) & (-2,0,2) \\ (-2,0,2) & (-2,0,2) & (0,2,4) \\ (-2,0,2) & (-2,0,2) & (-2,0,2) \\ (2,4,6) & (-2,0,2) & (-2,0,2) \\ (4,6,8) & (0,2,4) & (-2,0,2) \\ (2,4,6) & (-2,0,2) & (0,2,4) \\ (2,4,6) & (-2,0,2) & (0,2,4) \\ (-2,0,2) & (-2,0,2) & (2,4,6) \end{bmatrix} \end{array},$$

$$\widetilde{N}_2 = \begin{array}{c} \\ Cr_1 \\ Cr_2 \\ Cr_3 \\ Cr_4 \\ Cr_5 \\ Cr_6 \\ Cr_7 \\ Cr_8 \end{array} \begin{array}{c} \begin{array}{ccc} m/c_1 & m/c_2 & m/c_3 \end{array} \\ \begin{bmatrix} (0,1,2) & (0,1,2) & (-1,0,1) \\ (-0.707,0,0.707) & (-0.707,0,0.707) & (0,0.707,1.414) \\ (-0.577,0,0.577) & (-0.577,0,0.577) & (-0.577,0,0.577) \\ (0.577,1.154,1.732) & (-0.577,0,0.577) & (-0.577,0,0.577) \\ (0.894,1.342,1.789) & (0,0.447,0.894) & (-0.447,0,0.447) \\ (0.707,1.414,2.121) & (-0.707,0,0.707) & (0,0.707,1.414) \\ (0.707,1.414,2.121) & (-0.707,0,0.707) & (0,0.707,1.414) \\ (-0.577,0,0.577) & (-0.577,0,0.577) & (0.577,1.154,1.732) \end{bmatrix} \end{array}.$$

STAGE 3

$$\widetilde{R}_3 = \begin{array}{c} \\ Cr_1 \\ Cr_2 \\ Cr_3 \\ Cr_4 \\ Cr_5 \\ Cr_6 \\ Cr_7 \\ Cr_8 \end{array} \begin{array}{c} \begin{array}{cccc} m/c_1 & m/c_2 & m/c_3 & m/c_4 \end{array} \\ \begin{bmatrix} (0,2,4) & (-2,0,2) & (0,2,4) & (-2,0,2) \\ (-2,0,2) & (2,4,6) & (-2,0,2) & (-2,0,2) \\ (0,2,4) & (-2,0,2) & (0,2,4) & (2,4,6) \\ (-2,0,2) & (-2,0,2) & (-2,0,2) & (0,2,4) \\ (0,2,4) & (0,2,4) & (-2,0,2) & (2,4,6) \\ (2,4,6) & (2,4,6) & (-2,0,2) & (0,2,4) \\ (0,2,4) & (-2,0,2) & (-2,0,2) & (2,4,6) \\ (0,2,4) & (-2,0,2) & (0,2,4) & (2,4,6) \end{bmatrix} \end{array},$$

$$\widetilde{N}_3 = \begin{array}{cc} & \begin{array}{cccc} m/c_1 & m/c_2 & m/c_3 & m/c_4 \end{array} \\ \begin{array}{c} Cr_1 \\ Cr_2 \\ Cr_3 \\ Cr_4 \\ Cr_5 \\ Cr_6 \\ Cr_7 \\ Cr_8 \end{array} & \begin{bmatrix} (0, 0.707, 1.414) & (-0.707, 0, 0.707) & (0, 0.707, 1.414) & (-0.707, 0, 0.707) \\ (-0.5, 0, 0.5) & (0.5, 1, 1.5) & (-0.5, 0, 0.5) & (-0.5, 0, 0.5) \\ (0, 0.707, 1.414) & (-0.707, 0, 0.707) & (0, 0.707, 1.414) & (0.707, 1.414, 2.121) \\ (-0.577, 0, 0.577) & (-0.577, 0, 0.577) & (-0.577, 0, 0.577) & (0, 0.577, 1.154) \\ (0, 0.707, 1.414) & (0, 0.707, 1.414) & (-0.707, 0, 0.707) & (0.707, 1.414, 2.121) \\ (0.577, 1.154, 1.732) & (0.577, 1.154, 1.732) & (-0.577, 0, 0.577) & (0, 0.577, 1.154) \\ (0, 0.577, 1.154) & (-0.577, 0, 0.577) & (-0.577, 0, 0.577) & (0.577, 1.154, 1.732) \\ (0, 0.707, 1.414) & (-0.707, 0, 0.707) & (0, 0.707, 1.414) & (0.707, 1.414, 2.121) \end{bmatrix} \end{array}.$$

STAGE 4

$$\widetilde{R}_4 = \begin{array}{cc} & \begin{array}{cccc} m/c_1 & m/c_2 & m/c_3 & m/c_4 \end{array} \\ \begin{array}{c} Cr_1 \\ Cr_2 \\ Cr_3 \\ Cr_4 \\ Cr_5 \\ Cr_6 \\ Cr_7 \\ Cr_8 \end{array} & \begin{bmatrix} (0, 2, 4) & (0, 2, 4) & (0, 2, 4) & (-2, 0, 2) \\ (0, 2, 4) & (-2, 0, 2) & (0, 2, 4) & (-2, 0, 2) \\ (2, 4, 6) & (2, 4, 6) & (0, 2, 4) & (-2, 0, 2) \\ (0, 2, 4) & (-2, 0, 2) & (6, 8, 10) & (-2, 0, 2) \\ (-2, 0, 2) & (-2, 0, 2) & (6, 8, 10) & (-2, 0, 2) \\ (-2, 0, 2) & (-2, 0, 2) & (2, 4, 6) & (-2, 0, 2) \\ (-2, 0, 2) & (2, 4, 6) & (2, 4, 6) & (0, 2, 4) \\ (0, 2, 4) & (-2, 0, 2) & (-2, 0, 2) & (0, 2, 4) \end{bmatrix} \end{array},$$

$$\widetilde{N}_4 = \begin{array}{cc} & \begin{array}{cccc} m/c_1 & m/c_2 & m/c_3 & m/c_4 \end{array} \\ \begin{array}{c} Cr_1 \\ Cr_2 \\ Cr_3 \\ Cr_4 \\ Cr_5 \\ Cr_6 \\ Cr_7 \\ Cr_8 \end{array} & \begin{bmatrix} (0, 1, 2) & (0, 1, 2) & (0, 1, 2) & (-1, 0, 1) \\ (0, 0.707, 1.414) & (-0.707, 0, 0.707) & (0, 0.707, 1.414) & (-0.707, 0, 0.707) \\ (0.577, 1.154, 1.732) & (0.577, 1.154, 1.732) & (0, 0.577, 1.154) & (-0.577, 0, 0.577) \\ (0, 0.302, 0.603) & (-0.301, 0, 0.301) & (0.905, 1.207, 1.508) & (-0.302, 0, 0.302) \\ (-0.289, 0, 0.289) & (-0.289, 0, 0.289) & (0.866, 1.155, 1.443) & (-0.289, 0, 0.289) \\ (-0.5, 0, 0.5) & (-0.5, 0, 0.5) & (0.5, 1, 1.5) & (-0.5, 0, 0.5) \\ (-0.577, 0, 0.577) & (0.577, 1.154, 1.732) & (0.577, 1.154, 1.732) & (0.577, 1.154) \\ (0, 0.707, 1.414) & (-0.707, 0, 0.707) & (-0.707, 0, 0.707) & (0, 0.707, 1.414) \end{bmatrix} \end{array}.$$

STAGE 5

$$\widetilde{R}_5 = \begin{array}{cc} & \begin{array}{cc} m/c_1 & m/c_2 \end{array} \\ \begin{array}{c} Cr_1 \\ Cr_2 \\ Cr_3 \\ Cr_4 \\ Cr_5 \\ Cr_6 \\ Cr_7 \\ Cr_8 \end{array} & \begin{bmatrix} (-2, 0, 2) & (-2, 0, 2) \\ (-2, 0, 2) & (0, 2, 4) \\ (-2, 0, 2) & (0, 2, 4) \\ (0, 2, 4) & (-2, 0, 2) \\ (2, 4, 6) & (-2, 0, 2) \\ (-2, 0, 2) & (-2, 0, 2) \\ (-2, 0, 2) & (0, 2, 4) \\ (-2, 0, 2) & (-2, 0, 2) \end{bmatrix} \end{array},$$

$$\widetilde{N}_5 = \begin{array}{cc} & \begin{array}{cc} m/c_1 & m/c_2 \end{array} \\ \begin{array}{c} Cr_1 \\ Cr_2 \\ Cr_3 \\ Cr_4 \\ Cr_5 \\ Cr_6 \\ Cr_7 \\ Cr_8 \end{array} & \begin{bmatrix} (-0.707, 0, 0.707) & (-0.707, 0, 0.707) \\ (-1, 0, 1) & (0, 1, 2) \\ (-1, 0, 1) & (0, 1, 2) \\ (0, 1, 2) & (-1, 0, 1) \\ (0.707, 1.414, 1.732) & (-0.707, 0, 0.707) \\ (-0.707, 0, 0.707) & (-0.707, 0, 0.707) \\ (-1, 0, 1) & (0, 1, 2) \\ (-0.707, 0, 0.707) & (-0.707, 0, 0.707) \end{bmatrix} \end{array}.$$

Table 12 Machine priority table

Sl. no	Stage	Machine	Preference score	Machine priority
1	STAGE 1	MC-1	0.250744	1
2		MC-2	0.600906	3
3		MC-3	0.366403	2
4	STAGE 2	MC-1	0.539193	3
5		MC-2	0.477992	1
6		MC-3	0.499668	2
7	STAGE 3	MC-1	0.37589	3
8		MC-2	0.30571	2
9		MC-3	0.38798	4
10		MC-4	0.17068	1
11	STAGE 4	MC-1	0.2799	2
12		MC-2	0.3378	3
13		MC-3	0.0630	1
14		MC-4	0.5300	4
15	STAGE 5	MC-1	0.740972	1
16		MC-2	0.772545	2

Preference score and machine priority computed following the steps of the proposed approach is shown in Table 12.

Assigning the jobs to the machines by following the steps as shown in Fig. 2, the result obtained is tabulated in Table 13. The Gantt chart for the problem considered following the proposed algorithm is shown in Fig. 3.

5.3 Validation of the proposed heuristic algorithm

Validation for a heuristic algorithm is checks the robustness of the proposed model (Singh et al. 2006). For this reason, the result obtained from the proposed heuristic algorithm is compared with the result obtained from different established algorithms. The deviation of the results is computed using Eq. (20) and the comparison values are tabulated in comparison Table 14.

$$\text{Deviation values} = \frac{y - y_{pa}}{y_{pa}} \times 100\%, \tag{20}$$

where y_{pa} is the value of the performance parameters obtained from the proposed approach whereas y is values obtained from the heuristic algorithms. The value of deviation is calculated to show how accurate the proposed model behaves with respect to the other established models. Table 14 and Fig. 4 show the comparison of the performance measures of the various heuristic algorithms.

For the problem considered in the study, the execution time for the proposed scheduling heuristic model is 27.08 s which is fairly good in comparison to the execution time of other heuristic algorithms. The result of make-span from the proposed heuristic is 2041.6 min which is far superior to the make-span value obtained from other heuristic algorithms. However, the value of total tardiness and number of tardy jobs obtained from the proposed method is the fourth best and thirteenth best value, respectively. But it should be noted that the heuristic algorithms for which performance value of total tardiness and number of tardy jobs is better than the proposed model either compute an inferior value of make-span or take more execution time.

From the overall comparison of the values of performance measures computed by different heuristic algorithms and the deviation computed, it can be concluded that proposed heuristic algorithm could be applied for solving the FFSS problem in fuzzy environment.

5.4 Discussions

The general intention of the paper is to develop a heuristic algorithm that is capable of returning the near-optimal result for FFSS problems. The proposed heuristic algorithm is a two-phase method. The first phase involves prioritization of jobs as well as machines and the second phase of the proposed heuristic involves assigning of jobs to the machines in each stage of the problem. Some of the points observed during the study are as follows:

- *JP* is defined as the sequence in which the jobs shall be processed. Dispatching rules are the most popular form of *JP*. Sequencing and scheduling done by hybridizing two or more dispatching rules performs better than single-factor regulations.
- In the study, *JP* is done by hybridizing three dispatching rules viz. processing time, completion time and total work remaining. The job with higher priority value in a stage implies comparatively more time taken for completion and processing as well as more amount of work remaining for completing the job. Hence, such a job should be processed earlier rather than other jobs.
- *MP* involves the process of decision-making from the perspective of reliability to choose the machine that shall take the job for processing in case more than one machine is available.
- A novel MCDM model is developed for computing *MP*. The model is based on the concept of risk minimization. In the model, the decision matrix is converted into a relative cost matrix which is used to compute the priority value of the machines in each stage. The model perceives risk as processing of jobs in a comparatively less reliable machine. The highest prioritized machine implies that the machine is highly reliable and the

Table 13 Sequence of assigning jobs to machine as obtained by proposed heuristic

STAGE 1			STAGE 2			STAGE 3			
MC-1	MC-3	MC-2	MC-2	MC-3	MC-1	MC-4	MC-2	MC-1	MC-3
(4,5)	(4,6)	(4, 8)	(2,15)	(2,12)	(2,16)	(2,16)	(2,12)	(2,15)	(2,6)
(4,3)	(4,7)	(4,4)	(2,2)	(2,7)	(2,6)	(2,2)	(2,7)	(2,4)	(2,14)
(3,8)	(4,2)	(4,1)	(2,1)	(2,14)	(2,4)	(2,9)	(2,1)	(2,3)	(2,17)
(3,16)	(3,10)	(3,1)	(2,5)	(2,3)	(2,9)	(2,8)	(2,5)	(2,10)	(2,11)
(3,7)	(3,14)	(3,17)	(2,8)	(2,17)	(2,10)	(3,14)	(3,1)	(2,13)	(3,22)
(3,19)	(3,13)	(3,15)	(3,1)	(2,11)	(2,13)	(3,7)	(3,8)	(3,16)	(3,10)
(3,18)	(3,22)	(3,4)	(3,14)	(3,8)	(3,16)	(3,18)	(3,13)	(4,4)	(3,9)
(3,21)	(3,9)	(3,20)	(3,7)	(3,22)	(3,13)	(3,21)	(3,15)	(3,17)	(4,2)
(3,6)	(3,3)	(3,2)	(4,4)	(3,18)	(3,10)	(3,19)	(3,4)	(3,2)	(4,1)
(3,11)	(3,5)	(3,12)	(3,21)	(3,15)	(3,9)	(3,20)	(4,3)	(3,3)	(5,1)
(5,3)	(5,2)	(5,8)	(3,4)	(3,17)	(3,19)	(3,12)	(3,6)	(4,7)	(5,8)
(5,7)	(5,5)	(5,1)	(3,2)	(4,3)	(3,20)	(3,5)	(4,8)	(4,5)	(1,2)
(5,4)	(5,9)	(5,6)	(4,2)	(3,3)	(3,12)	(3,11)	(5,2)	(5,7)	(1,13)
(1,11)	(1,12)	(1,14)	(4,7)	(3,6)	(3,5)	(4,6)	(5,5)	(1,15)	(1,9)
(1,7)	(1,15)	(1,16)	(4,6)	(4,8)	(3,11)	(5,3)	(5,6)	(1,5)	
(1,13)	(1,1)	(1,9)	(5,2)	(4,1)	(4,5)	(5,9)	(1,14)	(1,1)	
(1,3)	(1,5)	(1,2)	(5,7)	(5,5)	(5,3)	(5,4)	(1,7)	(1,4)	
(1,10)	(1,4)		(5,6)	(5,9)	(5,1)	(1,12)	(1,6)	(1,16)	
(1,6)	(1,8)		(1,15)	(5,8)	(5,4)	(1,11)	(1,3)	(1,10)	
			(1,5)	(1,1)	(1,14)	(1,8)			
			(1,7)	(1,2)	(1,12)				
			(1,4)	(1,13)	(1,6)				
			(1,16)	(1,10)	(1,11)				
			(1,3)		(1,9)				
			(1,8)						

STAGE 4				STAGE 5	
MC-3	MC-1	MC-2	MC-4	MC-1	MC-2
(2,12)	(2,15)	(2,16)	(2,6)	(2,12)	(2,15)
(2,7)	(2,2)	(2,4)	(2,1)	(2,7)	(2,16)
(2,14)	(2,3)	(2,9)	(3,1)	(2,4)	(2,6)
(2,5)	(2,10)	(2,17)	(3,8)	(2,14)	(2,1)
(2,8)	(2,13)	(3,14)	(3,16)	(2,3)	(2,5)
(2,11)	(3,22)	(3,7)	(3,9)	(2,13)	(2,8)
(3,13)	(3,10)	(3,18)	(1,12)	(2,11)	(2,9)
(3,15)	(3,17)	(3,21)	(1,3)	(2,10)	(2,2)
(3,4)	(3,2)	(3,19)		(5,2)	(2,17)
(3,20)	(3,3)	(3,12)		(5,5)	(5,3)
(3,6)	(3,5)	(1,1)		(5,6)	(5,7)
(3,11)	(1,5)	(1,2)		(5,9)	(5,1)
(1,15)	(1,7)	(1,16)			(5,8)
(1,14)	(1,6)	(1,10)			(5,4)
(1,4)	(1,11)				
(1,13)	(1,8)				
(1,9)					

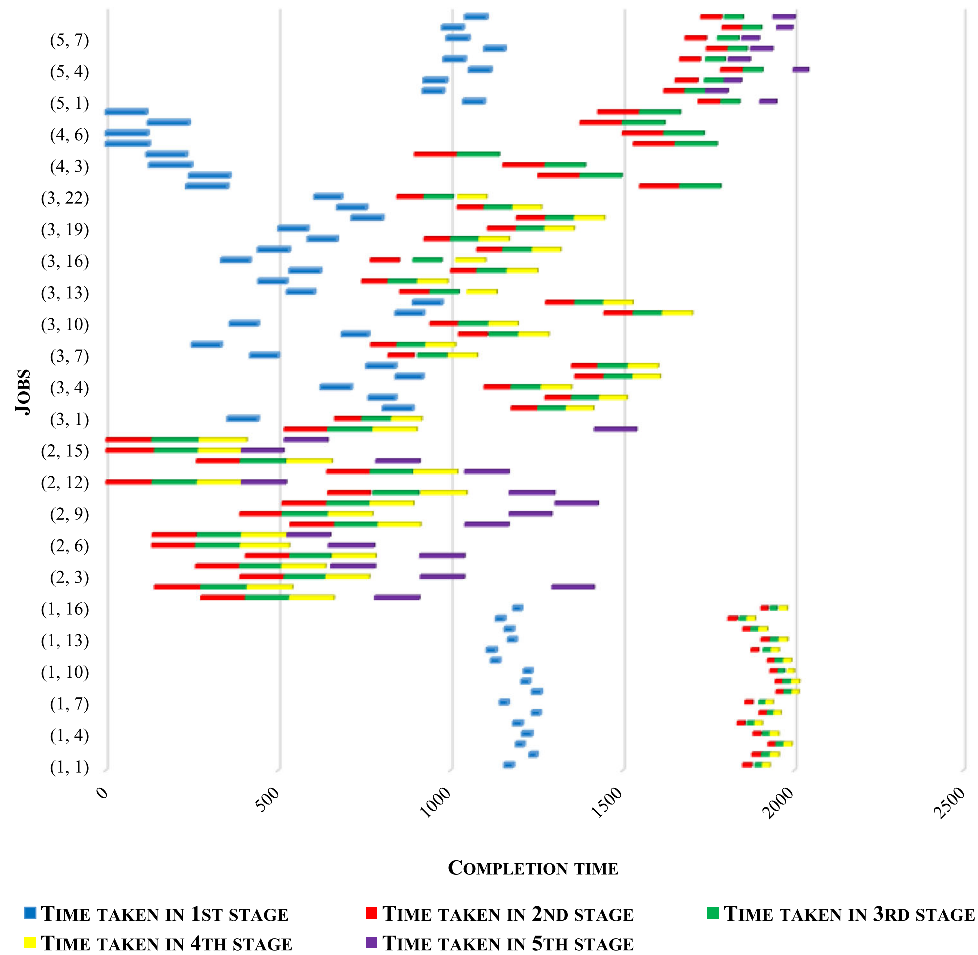


Fig. 3 Gantt chart obtained by the proposed heuristic

product obtained by manufacturing from such machine is of superior quality.

- The degree of importance of each criterion while prioritizing the jobs and machines is the aggregation of the linguistic ratings provided by the decision makers. The linguistic rating signifies the degree up to which a decision maker feels the criterion is relevant for the process of prioritization. The linguistic ratings are quantified using *TFNs*.
- The primary ambition of the proposed heuristic algorithm is to assign more number of jobs to the reliable machines in each stage of the problem. The assignment must also abide the assumptions that a low priority machine will not remain idle if jobs are available for processing at the present stage.
- It is observed that the proposed heuristic algorithm computed the best value of make-span and the fourth best value for total tardiness. Whereas the model performed poorly while minimizing the value for number of tardy jobs. However, it should be noted that average tardiness per unit tardy job is the least for the proposed model. The reason is that the model assigns more number of jobs to relatively more reliable machines.
- The proposed algorithm and eighteen other heuristic algorithms are applied to seventeen benchmark scheduling problems from the literatures. The algorithms are compared statistically on the basis of performance and execution time. Wilcoxon signed-rank test conducted at significance level of 0.05, shows that there is a significant difference in the performance of the proposed algorithm and SPT, LPT, TWR, NUP, EDD, ERD, FIFO, ECT, LCT, MST, PAL, PT + CT, CT + TWR and TWR + PT. It implies that the proposed heuristic algorithm provides better solution when applied to scheduling problems.

Some other important points that are observed during the study are:

- The *PT* for jobs on a reliable machine has values much nearer to the most likely time. In contrast, the *PT* for jobs on a relatively low reliable machines show negative skewness, i.e., most of the *PT* fall toward

Table 14 Comparison table

Sl. No	Algorithm	Performance measures			Deviation values			Execution time (s)
		C_{max} (minutes)	τ (minutes)	δ	C_{max} (%)	τ (%)	δ (%)	
1	Proposed approach	2041.6	938	103	–	–	–	27.08
2	SPT	2603.3	1235.6	91	27.51	31.73	− 11.65	22.36
3	LPT	2543.6	1018.5	98	24.59	8.58	− 4.85	25.384
4	TWR	2588.3	1135.3	105	26.78	21.03	1.94	30.725
5	NUP	2363	1016.6	105	15.74	8.38	1.94	31.34
6	EDD	2431.3	1075	86	19.09	14.61	− 16.5	29.872
7	ERD	2072.7	972.6	90	1.52	3.69	− 12.62	31.443
8	FCFS	2270.5	79 \9.6	110	11.21	− 14.75	6.8	24.561
9	ECT	2628	1374.3	111	28.72	46.51	7.77	26.395
10	LCT	2603.6	1235.6	93	27.53	31.73	− 9.71	27.02
11	MST	2357.8	1073	88	15.49	14.39	− 14.56	28.224
12	S/P	2068	909.2	78	1.29	− 3.07	− 24.27	29.325
13	PAL	2048.9	975	98	0.36	3.94	− 4.85	31.925
14	GUP	2314.2	936	101	13.35	− 0.21	− 1.94	35.623
15	NEH	2366.5	1176.6	86	15.91	25.44	− 16.5	31.24
16	CDS	2099.3	942.7	86	2.83	0.5	− 16.5	34.491
17	PT + CT	2352.3	1008.6	95	15.22	7.53	− 7.77	26.702
18	CT + TWR	2284	936	109	11.87	− 0.21	5.83	28.35
19	TWR + PT	2307.3	938	108	13.01	0	4.85	31.283

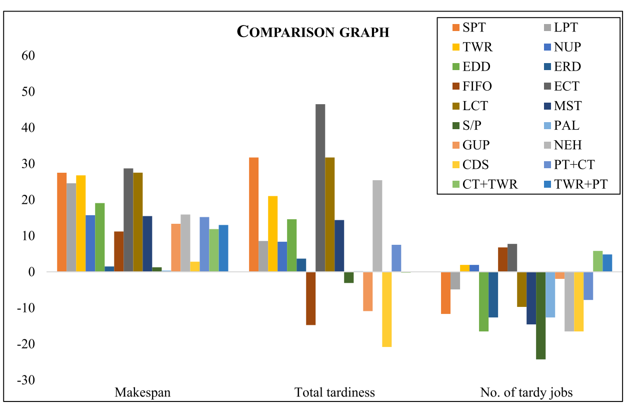


Fig. 4 Comparison chart

pessimistic value. This is also established by (Qin and Jiang 2005).

- The processing time for job $(1, 8)$ in 4th stage by 4th and 3rd machines for 100 periods is shown in Fig. 5a, b, respectively. It is observed that at an average 63%, the PT value falls toward the pessimistic time when processed on a relatively lower reliable machine. Whereas on average, the PT value for 69% concentrates about the most likely time when processed on a more reliable machine.
- In a particular stage, the difference between the pessimistic and optimistic value for processing a job on the least reliable machine for 100 cycles is around 81% more than processing the same job on a most reliable machine.
- The proposed algorithm strategically assigns more jobs to relatively more reliable machines in each stage. Due to this reason, the machine idle time for such machines is less in comparison to other machines. Since PT for reliable machines is less in comparison to less reliable machines. Hence, the make-span value obtained from the proposed heuristic is very much nearer to the optimal value. The number of jobs assigned to the machines at each stage is shown in Fig. 6.

5.5 Limitations of the proposed approach

The proposed approach computes the best result for make-span as it assigns more number of jobs to higher ranked machines. On the other hand, the model fails to compute a better result for the total tardiness and total number of tardy jobs. Above this, the execution time for the proposed model is more than some of the heuristic models in the literature.

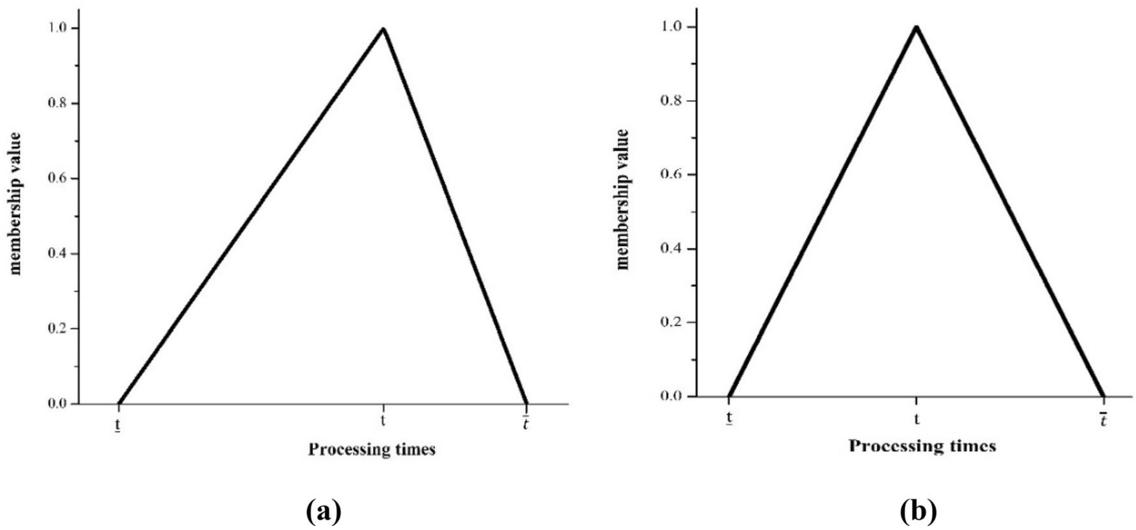
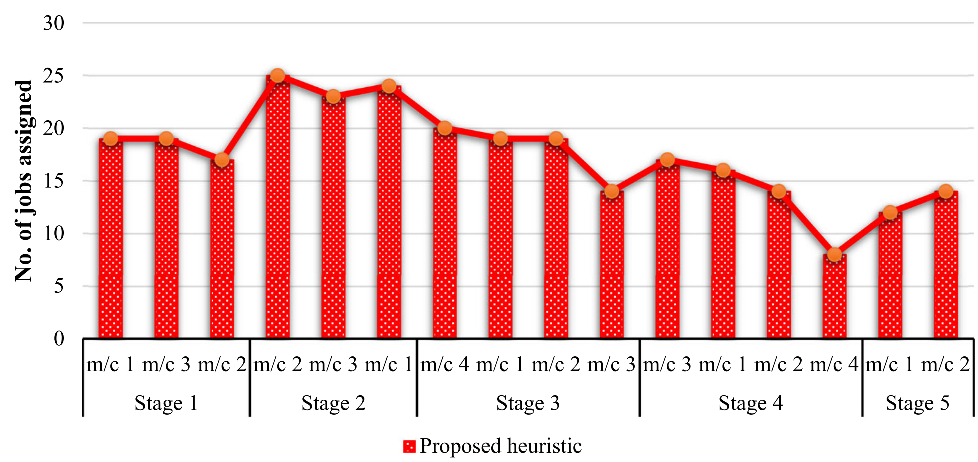


Fig. 5 Variation in *PTs* for job $(1, 8)$ in 4th stage by **a** 4th and **b** 3rd machines for 100 cycles

Fig. 6 Bar graphs representing jobs assigned to the machines



6 Conclusion

The comprehensive intention of the paper is to develop a robust scheduling heuristic algorithm capable of returning a good solution within a reasonable time-period. The study proposes a novel heuristic two-phase algorithm that prioritizes both the jobs as well as the identical parallel machines in a scheduling problem. Job prioritization is done by hybridizing the CT, PT and TWR. On the other hand, machine prioritization is done by computing the reliability of the parallel machines in the form of preference score. Ranking the identical machines based on certain criteria is an MCDM problem. A novel MCDM model is developed and proposed in this study. The MCDM model is built on the concept of risk minimization. In this study, risk is defined as the sense of regret for choosing a low reliable machine over a more reliable machine. The proposed heuristic algorithm assigns more number of jobs to a reliable machine than a less reliable machine. At the same time abides by the rule that a low priority machine shall not remain idle if jobs are available for processing at the present stage. Above this, the study employs the concept of fuzzy sets to model the uncertainty in the PT. The potentiality of a proposed heuristic algorithm lies in the practicality and robustness of the model. The proposed model is applied for scheduling jobs in the manufacturing industry of a medium enterprise in FFSS environment. Also, the same problem is solved by different heuristic algorithms. It is observed that the result obtained from the proposed model computed the best value for make-span and fourth best value for total tardiness. The heuristic algorithms for which performance value of total tardiness and number of tardy jobs is better than the proposed model either computes an inferior value of make-span or takes more execution time. From the study, it is also observed that *PT* for jobs on a more reliable machine is lesser than the *PT* for jobs on a low reliable machine. Because of this time taken for processing the jobs are relatively lesser than the time taken for processing the same job in relatively small reliable machines. The proposed approach outperformed heuristic approached for computing the make-span and therefore, the proposed approach is mostly be applied to the manufacturing industries that aims in increasing the production of the manufactured products. From the overall discussions, it can be concluded that the proposed heuristic can be applied for computing the performance measures of FFSS problem under uncertain environment.

Author contributions SAIH developed the proposed methodology, wrote the paper, and computed the solution. RK collected the data from the industry. UKM supervised and guided the project. All authors reviewed and revised the manuscript.

Funding On behalf of all authors, the corresponding author states that no funding in any form is received from any organization for carrying out the research work.

Data availability The data that support the findings of this study are enclosed as supplementary materials along with the manuscript.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1007/s41066-023-00377-y.

References

Ahmad M, Cheng W (2022) A novel approach of fuzzy control chart with fuzzy process capability indices using alpha cut triangular fuzzy number. Mathematics 10(19):3572

Ahmadi E, Zandieh M, Farrokh M, Emami SM (2016) A multi objective optimization approach for flexible job shop scheduling problem under random machine breakdown by evolutionary algorithms. Comput Oper Res 73:56–66

Akram M, Bibi R (2023) Multi-criteria group decision-making based on an integrated PROMETHEE approach with 2-tuple linguistic Fermatean fuzzy sets. Granul Comput. https://doi.org/10.1007/s41066-022-00359-6

Alcan P, BaşLıGil H (2012) A genetic algorithm application using fuzzy processing times in non-identical parallel machine scheduling problem. Adv Eng Softw 45(1):272–280

Almeder C, Hartl RF (2013) A metaheuristic optimization approach for a real-world stochastic flexible flow shop problem with limited buffer. Int J Prod Econ 145(1):88–95

Amin GR, El-Bouri A (2018) A minimax linear programming model for dispatching rule selection. Comput Ind Eng 121:27–35

Arashpour M, Wakefield R, Abbasi B, Lee EWM, Minas J (2016) Off-site construction optimization: sequencing multiple job classes with time constraints. Autom Constr 71:262–270

Arık OA, Toksarı MD (2018) Multi-objective fuzzy parallel machine scheduling problems under fuzzy job deterioration and learning effects. Int J Prod Res 56(7):2488–2505

Asadzadeh L, Zamanifar K (2010) An agent-based parallel approach for the job shop scheduling problem with genetic algorithms. Math Comput Model 52(11–12):1957–1965

Aydilek H, Allahverdi A (2013) A polynomial time heuristic for the two-machine flowshop scheduling problem with setup times and random processing times. Appl Math Model 37(12–13):7164–7173

Bari P, Karande P (2022) Ranking of sequencing rules in a job shop scheduling problem with preference selection index approach. J Decis Anal Int Comput 2(1):12–25

Behnamian J (2016) Survey on fuzzy shop scheduling. Fuzzy Optim Decis Mak 15(3):331–366

Bürgisser P (2013) Completeness and reduction in algebraic complexity theory, vol 7. Springer Science & Business Media, pp 5–6

Campbell HG, Dudek RA, Smith ML (1970) A heuristic algorithm for the n job, m machine sequencing problem. Manag Sci 16(10):B-630

Chen ZL (2004) Simultaneous job scheduling and resource allocation on parallel machines. Ann Oper Res 129(1–4):135–153

Choi BC, Leung JYT, Pinedo ML (2010) Complexity of a scheduling problem with controllable processing times. Oper Res Lett 38(2):123–126

Chua TJ, Yin X, Zhu J, Cai T (2011) U.S. Patent No. 7,904,192. Washington, DC: U.S. Patent and Trademark Office.

Dannenbring DG (1977) An evaluation of flow shop sequencing heuristics. Manag Sci 23(11):1174–1182

Davis RI, Tindell KW, Burns A (1993) Scheduling slack time in fixed priority pre-emptive systems. In: 1993 Proceedings Real-Time Systems Symposium, IEEE, pp. 222–231

Della Croce F, Scatamacchia R (2020) The longest processing time rule for identical parallel machines revisited. J Sched 23(2):163–176

Dubois D, Prade H (1982) The advantages of fuzzy approach in OR/MS demonstrated on two examples of resources allocation problems. Progress in cybernetics and systems research, vol VIII. Hemisphere, Washington, D.C, pp 491–497

Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE, pp. 39–43.

Erschler J, Roubellat F, Vernhes JP (1976) Finding some essential characteristics of the feasible solutions for a scheduling problem. Oper Res 24(4):774–783

Feng F, Zhang C, Akram M, Zhang J (2022) Multiple attribute decision making based on probabilistic generalized orthopair fuzzy sets. Granul Comput 28:1–29

Gao KZ, Suganthan PN, Pan QK, Chua TJ, Cai TX, Chong CS (2016) Discrete harmony search algorithm for flexible job shop scheduling problem with multiple objectives. J Intell Manuf 27(2):363–374

Gholami M, Zandieh M (2009) Integrating simulation and genetic algorithm to schedule a dynamic flexible job shop. J Intell Manuf 20(4):481

Gozali L, Kurniawan V, Nasution SR (2019) Design of job scheduling system and software for packaging process with SPT, EDD, LPT, CDS and NEH algorithm at PT. ACP. IOP Conf Ser: Mater Sci Eng 528(1):012045

Hong TP, Wang TT (1999) A heuristic Palmer-based fuzzy flexible flow-shop scheduling algorithm. In: FUZZ-IEEE'99. 1999 IEEE International Fuzzy Systems. Conference Proceedings (Cat. No. 99CH36315), IEEE, Vol. 3, pp. 1493–1497.

Hong TP, Wang CL, Wang SL (2000) A heuristic Gupta-based flexible flow-shop scheduling algorithm. In: Smc 2000 conference proceedings. 2000 IEEE international conference on systems, man and cybernetics. 'Cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0 (Vol. 1, pp. 319–322)). IEEE.

Hussain SAI, Mandal UK, Mondal SP (2018) Decision maker priority index and degree of vagueness coupled decision-making method: a synergistic approach. Int J Fuzzy Syst 20(5):1551–1566

Hussain SAI, Baruah D, Dutta B, Mandal UK, Mondal SP, Nath T (2019) Evaluating the impact of service quality on the dynamics of customer satisfaction in the telecommunication industry of Jorhat, Assam. Telecommun Syst 71(1):31–53

Jia Z, Yan J, Leung JY, Li K, Chen H (2019) Ant colony optimization algorithm for scheduling jobs with fuzzy processing time on parallel batch machines with different capacities. Appl Soft Comput 75:548–561

Joo CM, Kim BS (2015) Hybrid genetic algorithms with dispatching rules for unrelated parallel machine scheduling with setup time and production availability. Comput Ind Eng 85:102–109. https://doi.org/10.1016/j.cie.2015.02.029

Jungwattanakit J, Reodecha M, Chaovalitwongse P, Werner F (2008) Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. Int J Adv Manuf Technol 37(3–4):354–370

Kaban AK, Othman Z, Rohmah DS (2012) Comparison of dispatching rules in job-shop scheduling problem using simulation: a case study. Int J Simul Model 11(3):129–140

Kang L, Liu D, Wu Y, Ping G (2023) Two-stage hybrid optimization algorithm for silicon single crystal batch scheduling problem under fuzzy processing time. Math Probl Eng. https://doi.org/10.1155/2023/3816574

Karger D, Stein C, Wein J (2010) Scheduling algorithms. Algorithms and theory of computation handbook: special topics and techniques. Springer, pp 20–20

Karim R, Karmaker CL (2016) Machine selection by AHP and TOPSIS methods. Am J Ind Eng 4(1):7–13

Kim W, Kim H, Hwang J (2020) Sustainable growth for the self-employed in the retail industry based on customer equity, customer satisfaction, and loyalty. J Retail Consum Serv 53:101963

Kumar KK, Nagaraju D, Gayathri S, Narayanan S (2017) Evaluation and selection of best priority sequencing rule in job shop scheduling using hybrid MCDM technique. IOP Conf Ser Mater Sci Eng 197:1–10

Li K, Chen J, Fu H, Jia Z, Fu W (2019) Uniform parallel machine scheduling with fuzzy processing times under resource consumption constraint. Appl Soft Comput 82:105585

Liu GS, Zhou Y, Yang HD (2017) Minimizing energy consumption and tardiness penalty for fuzzy flow shop scheduling with state-dependent setup time. J Clean Prod 147:470–484

Luo Z, Yang H, Liu F (2023) An optimization-based method for eliciting priorities from fuzzy preference relations with a novel consistency index. Granul Comput. https://doi.org/10.1007/s41066-023-00361-6

Mokhtari H, Hasani A (2017) An energy-efficient multi-objective optimization for flexible job-shop scheduling problem. Comput Chem Eng 104:339–352

Molinari F (2016) A new criterion of choice between generalized triangular fuzzy numbers. Fuzzy Sets Syst 296:51–69

Naderi B, Gohari S, Yazdani M (2014) Hybrid flexible flowshop problems: models and solution methods. Appl Math Model 38(24):5767–5780

Nawaz M, Enscore EE Jr, Ham I (1983) A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega 11(1):91–95

Neufeld JS, Gupta JN, Buscher U (2016) A comprehensive review of flow shop group scheduling literature. Comput Oper Res 70:56–74

Nguyen S, Mei Y, Zhang M (2017a) Genetic programming for production scheduling: a survey with a unified framework. Complex Intell Syst 3(1):41–66

Nguyen S, Zhang M, Tan KC (2017b) Surrogate-assisted genetic programming with simplified models for automated design of dispatching rules. IEEE Trans Cybern 47(9):2951–2965

Novshek W, Thoman L (2006) Demand for customized products, production flexibility, and price competition. J Econ Manag Strateg 15(4):969–998

Ouelhadj D, Petrovic S (2009) A survey of dynamic scheduling in manufacturing systems. J Sched 12(4):417–431

Palmer DS (1965) Sequencing jobs through a multi-stage process in the minimum total time—a quick method of obtaining a near optimum. J Oper Res Soc 16(1):101–107

Petroni A, Rizzi A (2002) A fuzzy logic based methodology to rank shop floor dispatching rules. Int J Prod Econ 76(1):99–108

Pinedo M, Chao X (1999) Operations scheduling. McGraw Hill

Prade H (1979) Using fuzzy set theory in a scheduling problem: a case study. Fuzzy Sets Syst 2(2):153–165

Qin X, Jiang H (2005) A dynamic and reliability-driven scheduling algorithm for parallel real-time jobs executing on heterogeneous clusters. J Parallel Distrib Comput 65(8):885–900

Rapine C, Brauner N (2013) A polynomial time algorithm for makespan minimization on one machine with forbidden start and completion times. Discret Optim 10(4):241–250

Ruiz R, Vázquez-Rodríguez JA (2010) The hybrid flow shop-scheduling problem. Eur J Oper Res 205(1):1–18

Sakamoto S (2010) Beyond world-class productivity: Industrial engineering practice and theory. Springer Science & Business Media

Salvador MS (1973) A solution to a special class of flow shop scheduling problems. Symposium on the theory of scheduling and its applications. Springer, Berlin, Heidelberg, pp 83–91

Schultz CR (1989) An expediting heuristic for the shortest processing time dispatching rule. Int J Prod Res 27(1):31–41

Schwiegelshohn U, Yahyapour R (1998) Analysis of first-come-first-serve parallel job scheduling. SODA 98:629–638

Singh G, Singh PP, Lubana PPS, Singh KG (2006) Formulation and validation of a mathematical model of the microclimate of a greenhouse. Renew Energy 31(10):1541–1560

Sly D, Helwig M, Hu G (2017) Improving the efficiency of large manufacturing assembly plants. Proc Manuf 11:1818–1825

Subramaniam V, Lee GK, Ramesh T, Hong GS, Wong YS (2000) Machine selection rules in a dynamic job shop. Int J Adv Manuf Technol 16(12):902–908

Sumichrast RT, Russell RS, Taylor BW (1992) A comparative analysis of sequencing procedures for mixed-model assembly lines in a just-in-time production system. Int J Prod Res 30(1):199–214

Tran TT, Araujo A, Beck JC (2016) Decomposition methods for the parallel machine-scheduling problem with setups. Informs J Comput 28(1):83–95

Wang L, Li D (2002) A scheduling algorithm for flexible flow shop problem. Proceedings of the 4th world congress on intelligent control and automation (Cat. No. 02EX527), vol 4. IEEE, pp 3106–3108

Woolson RF (2007) Wilcoxon signed-rank test. Wiley encyclopedia of clinical trials, pp 1–3

Zhou W, Chen F, Ji X, Li H, Zhou J (2022) A Pareto-based discrete particle swarm optimization for parallel casting workshop scheduling problem with fuzzy processing time. Knowl-Based Syst 256:109872