Journal of Quality Engineering and Production Optimization

Volume 9, Issue 2, Summer & Autumn 2024

http://jqepo.shahed.ac.ir

Research Paper

DOI: 10.22070/jqepo.2025.18387.1270

Flow-shop scheduling problem with tool changes and tool wear to minimize energy consumption

Atiyeh Mohammad-Rafiei [1], Ehsan Mardan [1*], Reza Kamran Rad [1]

[1] *Department of Industrial Engineering, Faculty of Engineering, Semnan University, Semnan, Iran*

*** Corresponding Author:** *Ehsan Mardan (Email: Ehsan_Mardan@Semnan.ac.ir)*

Abstract *– Nowadays, most production units seek to minimize the amount of energy consumption due to the resulting economic pressures and the shortage of energy carriers such as electricity or fossil fuels. On the other hand, minimizing the total time to complete jobs aligns with reducing energy consumption. Therefore, efficient scheduling is one of the important concerns in industrial units. This research aims to minimize joint energy consumption and total job completion time in a flow-shop environment, considering speed level and tool wear level constraints. The increased speed level reduces the time to complete the job, and on the other hand, increases energy consumption. Moreover, the increase in the machine speed causes an increase in the level of wear and eventually a tool change. Since the flow-shop scheduling problem is NP-hard, it cannot be solved in a short time on a large scale. To cope with this issue, we solve this multi-objective optimization problem with two well-known metaheuristic methods, Non-dominated Sorting Genetic Algorithm (NSGA-II) and Multi-Objective Particle Swarm Optimization (MOPSO). The experimental results reveal that MOPSO outperforms NSGA-II in terms of MOCV and MID metrics. Conversely, NSGA-II outperforms MOPSO in terms of CPU time.*

Keywords*– Flow-shop Scheduling Problem, Optimization, Tool Change, Energy Consumption, Tool Wear, Machine Speed.*

I. INTRODUCTION

In the last decade, energy consumption management and greenhouse gas reduction have been of great importance for organizations and governments (Sola and Mota, 2020). Given that the industrial sector consumes about half of the world's energy, energy conservation is emerging as a competitive advantage (Ponzo et al., 2021). One of the most important sources of global heating is manufacturing companies (Utama et al., 2023). In industrial companies, operational planning is primarily focused on minimizing the time required to complete all tasks. However, this does not mean that other goals should be ignored (Stewart et al., 2023). Therefore, one of the essential sub-goals in manufacturing companies is to reduce energy consumption to save costs (Wu and Che, 2019; Mahdavi et al., 2023; Goli et al., 2023).

Most of the research work deals primarily with the development of a mathematical model to solve scheduling problems. These models include limitations in the production environment due to the type of problem-solving spaces

Manuscript Received: 22-March-2024 & Revised:6-May-2024 & Accepted: 12-August-2024
ISSN: 2423-3781

and the desired variables and parameters. Furthermore, over the past few decades, there has been a significant focus on energy consumption and its impact on various economic sectors and the environment. The industrial sector, in particular, is a major consumer of energy, and monetary policymakers in numerous countries have consistently prioritized efforts to optimize energy use and manage consumption effectively due to resource limitations (Goli et al., 2023).

Many studies consider the passage of time to be the only reason for tool change in the flow-shop scheduling problem. They believe that the tool is substituted when a certain level of wear occurs (Zhang et al., 2024). Most studies in this domain have primarily focused on developing mathematical models to solve scheduling problems. These models include some limitations in the production environment due to the type of problem-solving spaces and the desired variables and parameters. Currently, operational strategies such as production planning are employed to enhance the efficiency of energy consumption. This activity requires less investment compared to purchasing new machines (Yan et al., 2016). Some studies (Soleimani et al., 2020) investigate the time needed to move work from one machine to another, with the objective functions of minimizing energy consumption and the weighted sum of delays, considering the sequence of operations and the correct allocation of resources. Other studies (Chen et al., 2020) use only the speed factor to minimize the amount of energy consumption and the completion time of the last job. However, previous research (Kumar and Das, et al., 2024) shows that the machining speed plays a vital role in tool wear. According to the studies conducted in this field, the most effective factors in the amount of energy consumption are the speed of each machine and the duration of work on the machine. In this way, the higher the speed of a machine, the higher the level of wear in the tool. Therefore, it is better to consider both processing speed and processing time simultaneously. Our literature review reveals that, as the level of wear in the tool increases (Oda et al., 2015), the amount of energy consumption also increases. Hence, another objective function should be added to the previous one to minimize the makespan. On the other hand, the faster the machine, the less time it takes to prepare. This, in turn, leads to an even greater increase in energy consumption.

Since the flow-shop scheduling problem is NP-hard (Enayati et al., 2023), we will employ a high-performance metaheuristic algorithm to solve it. Specifically, we will solve a multi-objective optimization problem with two metaheuristic methods, Non-dominated Sorting Genetic Algorithm (NSGA-II) (Mousavi et al., 2024; Wang et al., 2023) and Multi-Objective Particle Swarm Optimization (MOPSO). The most important contributions of this paper are as follows:

- We formulate a new multi-objective model for the flow-shop scheduling problem. Our objective functions consider the wear level, machining speed, and tool change time in the flow-shop environment.
- Given the NP-hardness of the problem, we solve it with two well-known metaheuristic algorithms, one discrete (NSGA-II) and the other continuous (MOPSO). Then we compare the results.

The rest of this paper is organized as follows: Section 2 presents the literature review. Section 3 elaborates on the formulating the mathematical model of the flow-shop scheduling problem. Section 4 presents the problem-solving method. Section 5 explains the validation of the proposed methods and the evaluation criteria. Section 6 is devoted to performance evaluation; Finally, Section 7 concludes the paper and highlights future research trends.

II. LITERATURE REVIEW

Table I shows the most important studies on the flow shop environment. Readers interested in a more in-depth study can refer to Utama et al. (2023). Some machines may not be available for various reasons, such as the replacement of parts or energy consumption considerations. Therefore, in most studies, the joint scheduling of resource usage and availability is considered (Arasteh, 2022). Some researchers urge that scheduling should not only focus on minimizing production time but also consider minimizing energy consumption (Utama, 2021). Today, many factories are compelled to reduce energy consumption and lower greenhouse gas emissions from fossil fuels due to government regulations (Zhang et al., 2017; Zhang et al., 2023; Chen et al., 2020).

Unfortunately, classical optimization methods are unable to cope with large-scale problems, which are mostly NP-hard. For example, mixed integer linear programming techniques for a large number of variables are often unable to obtain a solution in a short time (Bruzzone et al., 2012). As an alternative, metaheuristic or heuristic methods are used in large-scale problems (Mousavi et al., 2024). Some of the most important metaheuristic algorithms used for the job shop problem are the genetic algorithm (Liou and Hsieh, 2015), particle swarm optimization (Nilakantan et al., 2015), and simulated annealing (Wang et al., 2019). Wen et al. (2023) conducted a study to perform intelligent production and minimize total energy consumption, while also considering makespan in flow shop scheduling. In their study, they proposed using robot production and coordinating the movement process of robots with the production process of machines. Shao et al. (2022) used a memetic algorithm to minimize the delay time of jobs and total carbon production. They found that by minimizing carbon production, energy consumption can be reduced. Huang et al. (2019) investigated flow shop problems by considering the sequence of jobs to minimize the makespan of the last job using a genetic algorithm.

Dai et al. (2013) investigated the minimization of energy consumption during idle time in unrelated parallel machines. They employed the switching on/off strategy in this problem. Li et al. (2018) studied a similar problem that was investigated by (Dai et al., 2013). They considered the setup time of each machine as a new job. In this vein, Ding et al. (2016) examined the bi-objective flow shop scheduling problem and introduced innovative methods as the dominant solution. They chose greenhouse gas emissions and makespan as their objectives. Singh et al. (2021) demonstrated that the contributions of the exact method and the B&B method are 13% and 7%, respectively. Wang et al. (2019) investigated flow shop problems by considering the two objectives of makespan and energy consumption in the flexible mode where they used the Pareto optimality to obtain the solution. They also used two methods of ant colony (AC) and Tabu Search (TS) to solve such problems. Geng et al. (2020) investigated worker flexibility in the flow shop environment. They set cost minimization as their first objective. Also, considering that costs are important in manufacturing industries, they set minimizing energy costs as another objective and used the evolutionary algorithm to solve these problems. Schulz et al. (2019) discussed the minimization of job completion time in a flexible flow shop environment by considering different processing speeds. According to the improvement of old machines and the replacement of new machines, they sought to minimize the makespan and energy consumption. In another study (Schulz et al., 2020), they completed the previous work by solving the problem using the EPS constraint. The authors in (Chaudhry et al., 2018) presented a genetic algorithm to minimize the total completion time by considering energy consumption in the flow shop environment. In another research (Mokhtari and Hasani, 2017), the minimization of energy consumption in the flow shop model using a genetic algorithm was investigated. Also, in (Xin et al., 2021) the genetic algorithm is used on a population monitoring plan to solve flow shop problems on a large scale.

In another study (Meng et al., 2019), a flexible flow shop scheduling algorithm for unrelated parallel machines was designed, utilizing a switching on/off strategy. To this end, the authors firstly examined the amount of energy consumption in the system. Then, by considering the mixed integer programming model, they studied two different ideas of idle time and energy consumption during idle time. Additionally, Esmaeili et al. (2021) employed the GA algorithm to solve single-machine problems, aiming to minimize the sum of tardiness and earliness. In another study (Rezvan et al., 2021), the NCDRA heuristic algorithm was employed to solve parallel machine problems using the MIP model to minimize a two-objective mathematical problem. In another study (Rastgar et al., 2021), a new mathematical model was presented to schedule hybrid flow shop problems with energy considerations.

In a study by Xiong et al. (2022), the authors reduced energy consumption in a flexible workshop environment using a two-stage mathematical model with deterministic task sizes. Todorov et al. (2019) investigated the type of alloy used in electric wires and the amount of energy loss in electric wires. Ham et al. (2021), taking into account the right of priority and delay in work and minimizing the time of doing work, set the minimization of energy consumption as their goal function. Gholizadeh et al. (2021) investigated efficient planning for maintenance activities to significantly reduce costs. In this research, mathematical modeling is designed for a flexible workshop flow environment to reduce lost energy. Sekkal and Belkaid (2023) investigated sequence-dependent multi-objective flow shop scheduling and stated that optimizing production systems has become increasingly important in industries due to increasing competition and market demand. In this research, the learning effect of employees is considered through mathematical modeling by minimizing two objective functions minimizing work time and energy consumption. Ghorbanzadeh et al. (2023) investigated flow shop problems by considering sequence-dependent start-up time, group scheduling, and restrictions. They aim to minimize the cost related to energy consumption.

Table I. Major research in the literature

Author	Scope	Number of Machines	Tool Wear	Machine Speed	Tool Change	Constraints	Objective(s)	Algorithm
Mouzon et al. (2007)	Single machine	1	-	-	-	Device idle time	Minimizing makespan and energy consumption	GA, SA, B&B
Dai et al. (2013)	Flexible flow shop	n	-	✓	-	The idle time of the device and cutting speed and volume of the machines	Minimizing makespan and energy consumption	SA & GA
Mansouri et al. (2016)	Flow shop	2	-	✓	-	Setup time and machine speed	Minimizing makespan and energy consumption	GA
Mokhtari and Hassani (2017)	Job shop	n	-	-	-	Without interruption and sequence of operations	Minimizing energy consumption	SA & GA
Wang et al. (2018)	Flow shop	2	-	-	-	Permutation	Minimizing makespan and energy consumption	Heuristic Algorithm
Li et al. (2018)	Flexible flow shop	n	-	-	-	Device idle time and cost changes due to electricity consumption at different times	Minimizing makespan and energy consumption	EA-MOA
Gadaleta et al. (2019)	Flexible flow shop	n	-	-	-	Non-dependent on the sequence and considering the transportation time between the machines	Minimizing energy consumption	GA
Soleimani et al. (2020)	Parallel machine s	n	-	-	-	The sequence of operations and appropriate allocation of resources to machines	Minimizing the weighted sum of delays and energy consumption	GA, CSO, & IABC
Geng et al. (2020)	Flexible flow shop	n	-	-	-	Flexibility of machines and workers considering the sequence of operations	Minimizing costs according to energy consumption	HEA Hybrid Evolution ary Algorithm
Shao et al. (2022)	Flexible flow shop	n	-	-	-	Considering resource constraints and reworking some stations	Minimizing energy consumption, customer dissatisfaction, and makespan	NSGA-III

Continue Table I. Major research in the literature

Author	Scope	Number of Machines	Tool Wear	Machine Speed	Tool Change	Constraints	Objective(s)	Algorithm
Chen et al. (2020)	Hybrid flow shop	n	-	✓	-	Considering speed machines	Minimizing makespan and energy consumption	NSGA-II
Schulz et al. (2020)	Flexible flow shop	n	-	✓	-	Considering speed machines	Minimizing energy consumption and delays	Eps limit
Zhang et al. (2020)	Parallel machine	n	✓	-	✓	Considering tool change and tool wear	Minimizing makespan and energy consumption	FFD
Ham et al. (2021)	Flexible flow shop	n	-	-	-	Considering priority in doing jobs	Minimizing energy consumption	NSGA-II
Shen et al. (2023)	Flexible job shop	n	-	-	-	Sequence-dependent set-up time	Minimizing makespan and cost of energy consumption	Heuristic Algorithm
Wang et al. (2023)	Hybrid flow shop	n	-	✓	-	Considering the conditions of uncertainty	Minimizing makespan and energy consumption	NSGA-II
Fontes et al. (2024)	job shop	n	-	✓	-	Transport resources by considering speed-adjustable	Minimizing makespan and energy consumption	NSGA-II
Zhang et al. (2023)	Hybrid flow shop	n	-	✓	-	Lack of consideration of heterogeneous shops	Minimizing makespan and energy consumption	Memetic, MOPSO
Our Research	Flow shop	n	✓	✓	✓	Sequence-dependent set-up time by considering the machine speed and tool change according to the level of wear	Minimizing makespan and energy consumption	NSGA-II, MOPSO

According to studies conducted in this field, the most effective factors influencing energy consumption are the speed of each machine and the duration of work on the machine. Therefore, we add another objective function to the problem to reduce the time taken to complete the last job. In this research, we also take into account the change of tools on the machine due to wear in the tools. In this way, with the passage of time and a change in the machine's speed, wear will occur on the tool, and we will need to change it. Therefore, a time, called sequence-dependent set-up time, is added to the model. Generally, the higher the machine's speed, the greater the wear on the tools. According to the above literature review, tool wear, tool change, and machine speed are among the factors affecting energy consumption, which have been less discussed. Therefore, in this research, we aim to minimize energy consumption using the above concepts, which have been the focus of many industries seeking to reduce costs and increase profitability, while also mitigating environmental effects. In summary, considering these concepts in the flow shop environment, we aim to develop an effective model to address the scheduling problem at the operational level.

III. PROPOSED MODEL

A. Problem Formulation

The number of n independent jobs consisting of $J = \{J_1.J_2.\cdots.J_n\}$ should be processed by m machines $M = \{M_1.M_2.\cdots.M_n\}$, all of which are sequentially placed behind each other. Each machine enjoys an adjustable speed of $V = \{V_0.V_1.\cdots.V_{r+1}\}$ and V_0displays the machine's standby mode.

Performing each job J_i on the machine M_j has a processing time of p_{ij}. Each job on each machine is processed at the speed of $V_{\mathrm{k}}(K = 1.2.\dots.r)$. The higher the processing speed in the machine, the shorter the processing time. In other words, if $V_{\mathrm{k}} < V_{\mathrm{k}'}$, we will have $p_{ijk} > p_{ijk'}$. In addition, pp_{jk} shows the amount of energy consumed by the machine at speed k. Thus, it is obvious if $V_{\mathrm{k}} < V_{\mathrm{k}'}$, we will have $p_{ijk} \times pp_{\mathrm{jk}} > p_{ijk'} \times pp_{\mathrm{jk}}$. This indicates that higher processing speeds result in lower processing time and higher energy consumption.

Another parameter is to consider the maximum wear level of the tool T. There is a direct relationship between the tool wear level and processing speed. In this way, the higher speed in the machine will cause more wear on the tool. When the wear level of a tool reaches T, the corresponding machine must be stopped for tool change. At this time, the device lies in the standby mode and the tool change duration is T_C. The energy consumed in the standby mode of each machine is sp_j.

After the proposed problem is solved, the order of the jobs in the machine and the processing speed of the machines are calculated at the same time. The proposed model has the following two objective functions:

1. Minimizing the completion time of the last job (C_{max})
2. Minimizing the amount of energy consumption (TEC)

Also, the assumptions considered for this problem are as follows:

- The first machine is available at zero time, and the start time of the next machine depends on the processing time of the previous machine.
- It is not allowed to turn off the device until all jobs are completed.
- Each machine can process the same job at the same speed and at the same time.
- Each job is processed by only one machine at a time.
- Each job must be processed by all machines.
- Tool change is not allowed until the work processing on the machine is completed.
- Permutation between jobs is allowed. It should be noted that Baker et al. (2013) state that it must be clarified whether a permutation is allowed or not when $F_m||C_{max}$ problem is to be solved. If this is allowed, the non-observance of a similar sequence on the machines will create sequences with a better objective function value in some cases. It has also been proven that considering one of the following two features can improve the value of the objective function:
    - Since all the criteria are considered in flow shop scheduling problems, it suffices that the sequences are the same for the first two machines.
    - According to the range of flow shop scheduling problems, those programs can be considered in which a similar sequence happens only in the last two machines.

B. Mathematical Model

This section introduces a multi-objective linear mathematical model designed for flow-shop scheduling issues. Initially, the indices, parameters, and decision variables related to the problem are outlined. Following that, the objective functions and constraints of the proposed model are detailed. Table II displays the notations utilized in this paper.

Table II. Notations and symbols

j	Machine index
i	Job index
k	Index of speed levels
h	Job sequence index
Parameters:	
n	Number of jobs $i = 1.2. \dots . n$
m	Number of machines $j = 1.2. \dots . m$
r	The number of expected speed levels of v_k so that $k = 1.2. \dots . r$
T	The upper limit of wear (Martindale)
T_c	Tool change duration
p_{ijkh}	Processing time of job i on machine j at speed v_k in sequence h (Second)
sp_j	Machine power unit j in standby mode (Kw)
pp_{jk}	The power unit of the machine j when it has speed v_k so that $k = 1.2. \dots . r$ (Kw)
w_k	Unit of tool wear associated with speed v_k (Martindale)
B	A large number
Decision variables:	
x_{ijkh}	It is a binary decision variable. If a job i is processed on a machine M_j with speed $v_k$in sequence h, it will be equal to one; otherwise, it will be zero.
y_{jh}	It is a binary decision variable. It will be equal to one if it is required to change the tool in machine j and sequence h; otherwise, it will be equal to zero.
E_{jh}	A positive value of the cumulative value of tool wear between the completion time of the previous tool and the completion time of the current job on the jth machine in sequence h.
c_{ijh}	Completion time of the job i in a machine j in sequence h.

Objective Functions and constraints of the problem are as follows:

$$min\, c_{max} \tag{1}$$

$$min\, TEC = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{r}\sum_{h=1}^{n} x_{ijkh}\, p_{ijkh} pp_{jk} + \sum_{j=1}^{m}\sum_{h=1}^{n} y_{jh}\, T_c\, sp_j \tag{2}$$

Subject to:

$$c_{max} \geq c_{ijh} \qquad \forall j = 1.2. \dots m\ i = 1.2. \dots . n. h = 1.2. \dots . n \tag{3}$$

$$\sum_{i=1}^{m}\sum_{k=1}^{r} x_{ijkh} = 1 \qquad \forall j = 1.2.\dots.m.h = 1.2.\dots.n \tag{4}$$

$$\sum_{k=1}^{r}\sum_{h=1}^{n} x_{ijkh} = 1 \qquad \forall i = 1.2.\dots.n.j = 1.2.\dots.m \tag{5}$$

$$\sum_{k=1}^{r}\sum_{h=1}^{n} x_{ijkh} \geq \sum_{k=1}^{r}\sum_{h=1}^{n} x_{i(j+1)kh} \qquad \forall i = 1.2.\dots.n.j = 1.2.\dots.m \tag{6}$$

$$E_{j(h-1)} + \sum_{i=1}^{n}\sum_{k=1}^{r} x_{ijkh}\, p_{ijkh}\, w_k \leq E_{jh} + y_{j(h-1)} * B \quad \forall j = 1.2.\dots.m\,.h = 1.2.\dots.n \tag{7}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{r} x_{ijkh} p_{ijkh} w_k \leq E_{jh} \qquad \forall j = 1.2.\dots.m\,.h = 1.2.\dots.n \tag{8}$$

$$E_{jh} \leq T \qquad \forall j = 1.2.\dots.m\,.h = 1.2.\dots.n \tag{9}$$

$$\sum_{k=1}^{r} x_{ijkh} \leq B * c_{ijh} \qquad \forall\, i = 1.2.\dots.n.j = 1.2.\dots.m.h = 1.2.\dots.n \tag{10}$$

$$c_{ijh} \leq B * \sum_{k=1}^{r} x_{ijkh} \qquad \forall\, i = 1.2.\dots.n.j = 1.2.\dots.m.h = 1.2.\dots.n \tag{11}$$

$$\sum_{i=1}^{n} c_{ij(h+1)} \geq \sum_{i=1}^{n} c_{ijh} + \sum_{i=1}^{n}\sum_{k=1}^{r} p_{\mathrm{ij}k(h+1)} x_{\mathrm{ij}k(h+1)} + y_{\mathrm{j}(h+1)} T_c \qquad \forall j = 1.2.\dots.n\,.\ h = 1.2.\dots.n \tag{12}$$

$$\sum_{h=1}^{n} c_{i(j+1)h} \geq \sum_{h=1}^{n} c_{ijh} + \sum_{h=1}^{n}\sum_{k=1}^{r} p_{i(j+1)kh} x_{i(j+1)kh} + \sum_{h=1}^{n} y_{(j+1)h} T_c \qquad \forall i = 1.2.\dots.n.j = 1.2.\dots.m \tag{13}$$

$$\sum_{h=1}^{n} c_{i1h} \geq \sum_{h=1}^{n}\sum_{k=1}^{r} x_{i1kh} p_{i1kh} \qquad \forall\, i = 1.2.\dots.n \tag{14}$$

Eqs. (1) and (2) are the objective functions that show the minimization of the job completion time and the total energy consumption. Eq. (3) indicates that the completion time of all jobs should be greater than that of each job. Eq. (4) states that no more than one job is assigned to each machine, and Eq. (5) ensures that a machine does not process more than one job at a time. Eq. (6) ascertains that some activity has been done on previous machines, and work needs to be done only on the next machine. In other words, it describes the order in which the machines process. Eqs. (7) and (8) ensure that the cumulative wear value between tools has been set correctly. In this way, Eq. (8) shows that the level of wear should not exceed the ceiling determined in a sequence. This means that if we change tools in the previous sequence, the second term of Eq. (7) is multiplied by the value *B*, and this limit will always be correct, and only the limit of Eq. (8) for wear levels in the same sequence will be obtained. However, if we do not change the tool, the wear level in the previous sequence will be added to the wear level in the current sequence, and the cumulative sum of the wear level will be obtained. Fig. 1 shows an illustration of the concept of the value of tool wear.

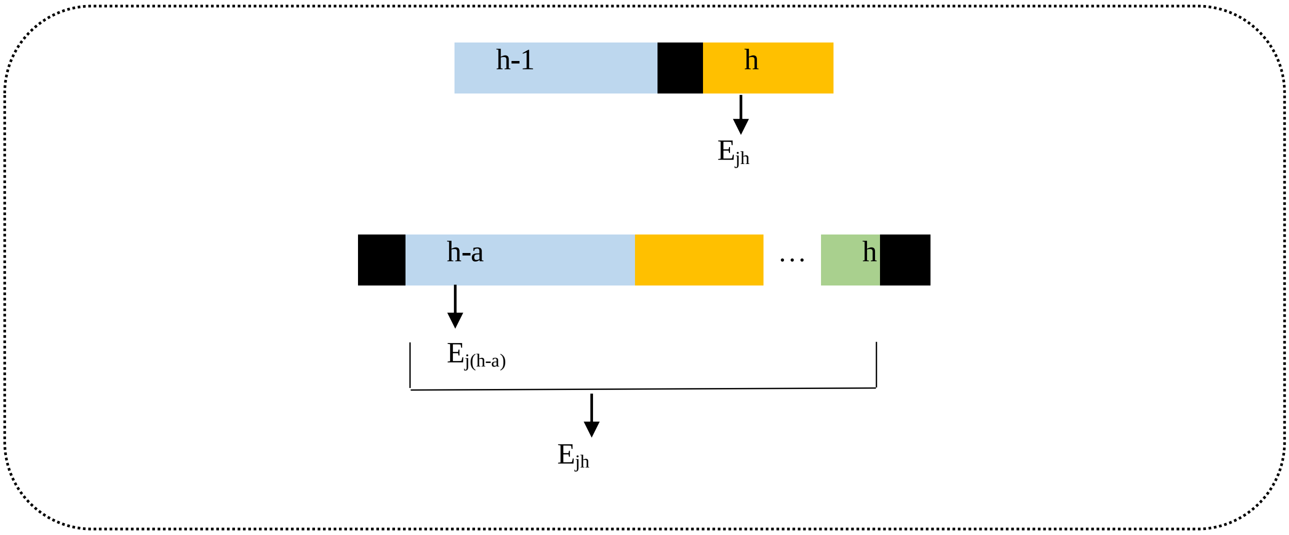

Fig. 1. The concept of the value of tool wear

Eq. (9) ensures that the cumulative value of the instrument cannot exceed its upper limit. Eqs. (10) and (11) guarantee that it is possible to calculate the completion time of a job on a machine in a specific sequence when that job is assigned to that machine in a specific sequence. Eq. (12) states that the completion time of each job in a larger sequence is equal to the sum of the completion time of the same job in the previous sequence, plus the processing time in the same sequence and the tool change time. Eq. (13) asserts that the completion time of each job on a machine is larger than or equal to the sum of the completion time of the job on the previous machine and the processing time on the same machine and tool change. Eq. (14) guarantees that the completion time of the job on the first machine is not smaller than the completion time of the same job on the first machine.

IV. PROBLEM-SOLVING METHOD

In this section, NSGA-II and MOPSO algorithms are used to solve the proposed mathematical model.

A. NSGA-II

The concept of the NSGA-II was first presented by Deb et al. (2002), and many researchers have used this algorithm to optimize multi-objective problems. This algorithm utilizes the principles of non-dominance and crowding distance to select and rank solutions. Thereafter, the two operators of intersection and mutation are applied to create a new set of solutions (children) and transfer the features to the next generation. Finally, according to the principle of non-dominance and variety, the best solutions are selected as the Pareto front.

In genetic algorithms, the solution is extracted through the coding of chromosomes. Then, extracting the solution to this problem from this chromosome is essential. The solution consists of two parts, namely *Seq* and *Sp*, where *Seq* shows the sequence of jobs and *Sp* shows the speed of processing the jobs in the sequence.

Let *I* be the number of jobs and *J* represents the number of machines. So, the proposed chromosome contains two random vectors of numbers in the interval [0, 1]. The first chromosome is in *I*1* dimension and the second chromosome is *I*J* dimension. For example, if three jobs are scheduled on two machines, the random chromosomes will be as follows:

Sequence determining chromosome: By sorting the above vector in the ascending order and determining each member of the initial vector in the sorted vector, the job sequence of all machines is obtained.

0.710	0.121	0.310

Seq1	Seq2	Seq3
3	1	2

Processing speed determining chromosome: For example, if the vector of the processing speed determining chromosome corresponds to the following vector:

0.810	0.490	0.610
0.620	0.400	0.498

To achieve the processing speed, each cell in the above matrix should be multiplied by the corresponding processing speed levels, and the result should be rounded up. In this way, the processing speed in each sequence and on each machine is also obtained.

0.810	0.490	0.610
0.620	0.400	0.498

2.430	1.470	1.830
1.860	1.200	1.494

3	2	2
2	2	2

It shows that in the first machine, job 3 should be processed in the first sequence at speed 3, job 1 should be processed in the second sequence at speed 2, and job 2 should be executed in the third sequence with speed 2. On the other hand, in the second machine, job 3 should be processed in the first sequence with speed 2, job 1 should be processed in the second sequence at speed 2, and job 2 should be processed in the third sequence at speed 2.

	Seq1	Seq2	Seq3
Machine 1	3	2	2
Machine 2	2	2	2

The arrangement of the activities on the machine and the processing speed are obtained by the above display. After determining these items, the start and end times of the jobs and the amount of cumulative erosion are obtained. This problem has only one penalty function, which calculates the amount of violation of cumulative erosion from the upper limit of erosion. Then, this value is multiplied by a large value and is added to the objective functions. If the value of the violation is equal to zero, no penalty is considered. However, if there is a violation, the large value is added to the two objective functions, deteriorating their value. Thus, we write:

$$CV = \max(0 . \frac{\max(E)}{T} - 1) \tag{15}$$

In the above expression, CV shows the violation value. Based on the violation function, the value of the objective functions is considered as follows:

$$Obj_1 = OFV_1 + M(CV) \tag{16}$$

$$Obj_2 = OFV_2 + M(CV) \tag{17}$$

In the above expressions, OFV_k denotes the obtained value of the *k*-th objective function, M is a large number, and CV represents the maximum of the *m*-th penalty function.

B. MOPSO

Coello (2002) made some changes in the PSO algorithm and developed this algorithm for multi-objective problems. The main difference between MOPSO and single-objective PSO lies in determining the best particle in the population and identifying the best personal memory for each particle. In the multi-objective particle swarm optimization algorithm, a new concept, namely the archive, has been introduced compared to the single-objective mode, which serves as a storage place for non-dominated solutions. By defining the archive in this algorithm, the concept of the best particle in the population has also changed.

In the MOPSO paradigm, particles move in the search space based on social tendencies. The position of a particle changes based on both the experiences of the particle itself and the information of neighboring particles. The exchange of information among particles is accomplished by the velocity vector, according to the following equation:

$$\overrightarrow{V_t}(t) = \mathrm{W}\overrightarrow{V_t}(t-1) + c_1 r_1 \left(\overrightarrow{X_{pbest1}} - \overrightarrow{x_t}(t)\right) + c_2 r_2 (\overrightarrow{X_{gbest}} - \overrightarrow{x_t}(t)) \tag{18}$$

The velocity vector reflects the information that has been exchanged collectively. This vector consists of three steps. In the first step, the current speed is obtained from the previous speed change. In the second step, there is a cognitive part that represents the best personal memory of the particle. In the third step, there is the social part, which represents the best collective memory of the particles. This memory has already been obtained from the collective experiences of the particles, and $r_1, r_2 \in [0,1]$ are the random values that cause diversity in the solutions. W , c_1, and c_2 are the coefficients of inertia, the best personal memory, and the best overall memory, respectively.

Coello (2002) employed the concept of Pareto optimization to obtain the optimal solution, where each particle is evaluated by all objective functions. The best positions are generated by the non-dominated approach. When only one particle is selected to update the velocity vector, there may be numerous non-dominated solutions in the neighborhood of a particle. Due to the constraint of the archive size, it is impossible to save all non-dominated solutions. In other words, the Pareto front size may become larger than the front size. For this purpose, an external archive is used to store the unfavorable solutions searched during the process. It is noteworthy that the size of the considered external archive is also limited. For this reason, legal availability is highly necessary to replace existing solutions with newly developed ones. The replacement rule in this algorithm has been considered in such a way that it improves the degree of order in the dispersion of non-dominated particles in the regions of the target space. To achieve this purpose, particles in areas with the highest particle density will have a higher probability of removal, while those in quieter areas will have a lower probability of removal. The accomplishment of this job through the crowding distance determines the probability of removing each member. However, when it is done using the *Roulette wheel*, those members will be selected that have been supposed to be removed. The display of the solution in the multi-objective particle swarm optimization algorithm is the same as that in the NSGA-II algorithm. It specifies the position of each job and the processing speed, and each chromosome in the NSGA-II algorithm represents a particle in the MOPSO algorithm. Due to space limitations, we refrain from repeating the details of this algorithm.

V. PERFORMANCE CRITERIA AND MODEL VALIDATION

To validate the proposed model, we designed a small example whose information is shown in detail in Table III. Then, we solved the proposed model as two single-objective problems for the given example. Each single-objective problem has been solved by the GAMS software using the CPLEX solver. The results of solving the two-objective model are shown in Fig. 2.

Table III. Example of the proposed model

n	5	T_C	3
m	3	p_{ijkh}	Uniform(10,25)
r	3	sp_j	1.364 2.291 3.121
T	25	pp_{jk}	Uniform(30,65)
w_k	1.545 1.032 1.792	B	500

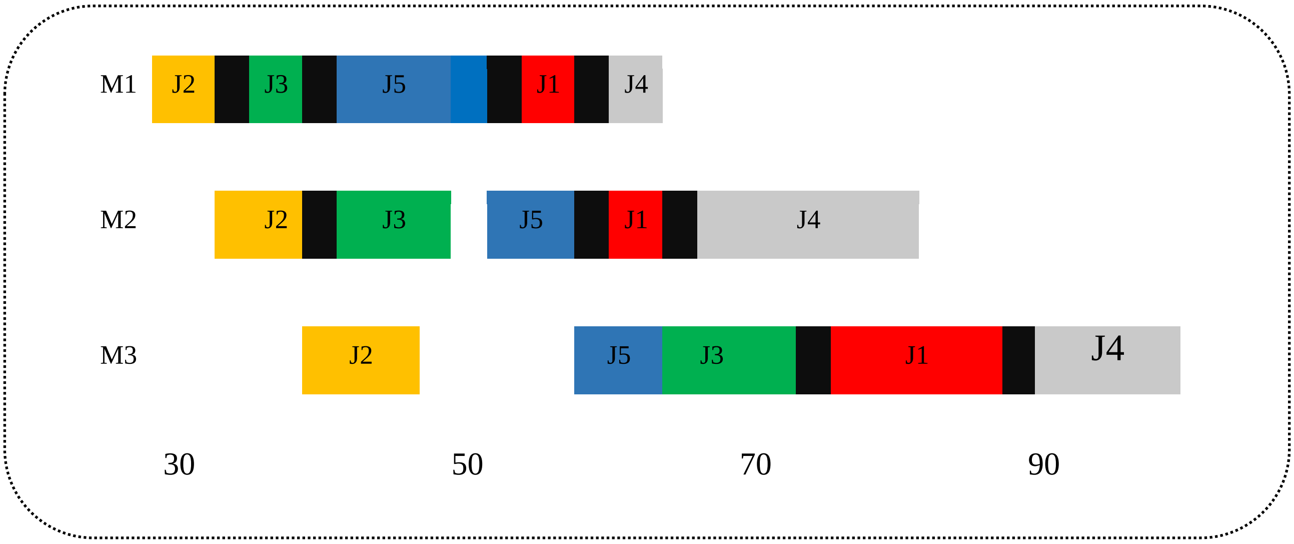

Fig. 2. The sequence obtained from solving the proposed model by the GAMS software

Fig. 2 shows the change of tool. The results from the solution in the GAMS software show the values of the first objective function (makespan) as 96.449 and the second objective function (energy consumption) as 8419.734. If we disable any of the target functions in the software and then run the program, the values of the target function will display the opposite numerical values of the obtained numbers. For example, in the case of deactivating the first objective function, which aims to minimize the makespan, the value of the second objective function is 7091.106, indicating a conflict between the objective functions. The sequence is the same in machines M1 and M2; however, in the third machine, between jobs 3 and 5, due to the permitted permutation in the problem model, displacement has occurred.

A. Parameter Setting by Taguchi Method

The Taguchi method is one of the methods for designing experiments, and it is also efficient in terms of the proposed algorithm (Unal and Dean, 1990). If the parameters of the algorithms are well-set, the efficiency of the solutions will improve. One method of parameter setting is to test all available modes, which is time-consuming and costly. Hence, the Taguchi method is used to set the parameters correctly. The Taguchi method is one of the most widely used statistical methods for analyzing the output sensitivity of a process in experimental design. This method is used when it is desired to determine the best output level of the process by performing only some part of the necessary tests. In this method, after defining the desired levels for each of the effective factors in the test, a set of designs is proposed to the examiner to determine whether it is possible to select one of the appropriate designs presented in the Taguchi method. This selected design should be consistent with the number of levels and type of experiments. Then, the examiner embarks on conducting the experiments. In the next step, the output data arising from the experiments is returned to the Taguchi design. Finally, the analysis carried out using the Taguchi method reveals the impact of each factor on the dependent variable of the process. This method converts the iterative data obtained from the experiments into an index of changes, which is called signal-to-noise ratio conversion.

Considering the parameters in each algorithm, MINITAB 17 software calculates the number of times the algorithm needs to be executed. Using the defined values, a sample problem is executed according to the levels suggested by Taguchi, and the resulting solutions are documented. To normalize the values obtained from the algorithm's execution, the Relative Percentage Deviation (RPD) index is applied. It is obtained as follows:

$$\text{RPD} = \frac{\text{the most efficient solution} - \text{execution solution}}{\text{the most efficient solution}} \times 100 \tag{19}$$

It is worth noting that the mean ideal distance is the index used for parameter setting, which has been employed in numerous studies. A lower RPD index is more favorable. Hence, the "Smaller is better" option is used when executing the Taguchi method. Moreover, a higher signal-to-noise ratio is more advantageous. As a result, the highest value of the vector is chosen for each parameter, and the associated level is regarded as the ideal level. The parameter settings of the NSGA-II and MOPSO algorithms are as follows:

- N_{pop}: refers to the number of chromosomes or the size of the genetic algorithm population.
- *Pc*: the percentage of solutions from the population of the algorithm that are intersected in each iteration of the algorithm
- *Pm*: refers to the percentage of solutions from the algorithm's population that undergo mutation during each iteration.
- max_{it}: the highest number of iterations allowed for each execution of the algorithm *W*: Inertia coefficient.
- *W:* Inertia coefficient
- C_1: Coefficient of personal best memory
- C_2: Coefficient of the best collective memory
- Swarm size (*N*): population size in MOPSO algorithm

Table IV presents the values of parameters in both algorithms. Based on the obtained results, the "Means of S/N ratios" vectors for NSGA-II and MOPSO algorithms are shown in Figs. 3 and 4, respectively. Based on the values shown in figures. Finally, the optimal values of the NSGA-II and MOPSO algorithm parameters can be summarized in Table V, items 3 and 4.

Table IV. Parameter values of NSGA-II and MOPSO algorithms

Parameter values of the NSGA-II algorithm				Parameter values of the MOPSO algorithm			
Parameter	1	2	3	Parameter	1	2	3
Npop	200	350	550	C_1	1	1.5	2
Pc	0.5	0.6	0.7	C_2	1	1.5	2
Pm	0.15	0.20	0.25	Swarm size(N)	50	75	90
MaxIt	100	200	300	MaxIt	200	400	600
				Inertia factor(W)	0.6	0.75	0.9

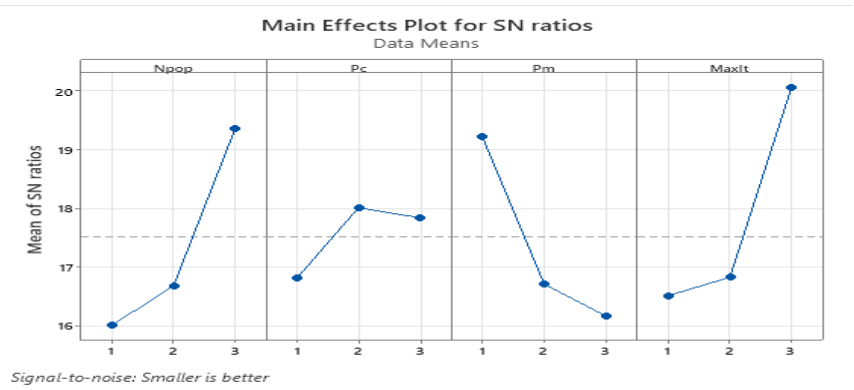

Fig. 3. S/N vector for NSGA-II algorithm

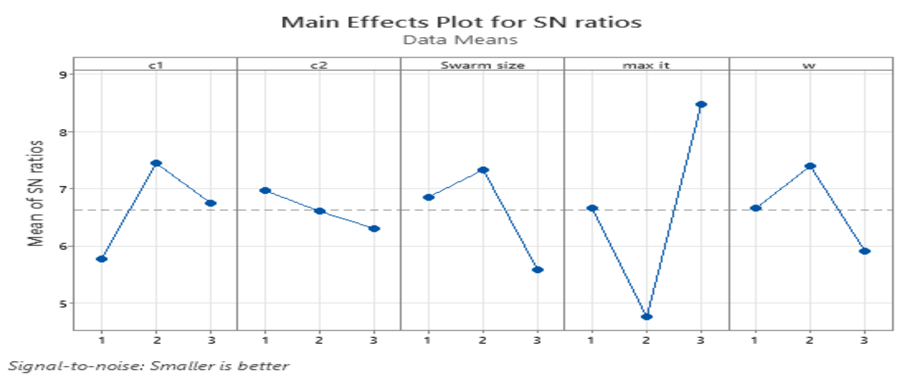

Fig. 4. S/N vector for MOPSO algorithm

Table V. Optimum hyperparameter values for NSGA-II and MOPSO algorithms

Optimal values for the parameters of NSGA-II algorithm		Optimal values for the parameters of the MOPSO algorithm	
Npop	550	C_1	1.5
Pc	0.6	C_2	1
Pm	0.15	Swarm size(N)	75
MaxIt	300	MaxIt	600
		Itertia factor(W)	0.6

B. Performance Criteria

There are different ways to evaluate the efficiency of algorithms. One of these approaches is to thoroughly investigate the solution space obtain all the non-dominated points and compare the solutions obtained from the algorithms with them. However, in practice, this is only suitable for problems with a small number of problem dimensions, and it is not suitable for problems with large dimensions. As a result, the indices used to compare multi-objective metaheuristic algorithms are employed to compare the efficiency of the algorithms with each other and to evaluate their performance. The evaluation criteria of algorithms are often divided into two categories. The first category affects the *convergence* and quality of the solutions, and the second category focuses on the *dispersion and expansion* of the solutions in the solution space. In this research, six metrics are presented for evaluating the performance of algorithms. Let us briefly review these metrics:

a) Spacing Index: This criterion, presented by Schott (1998), calculates the relative distance of consecutive solutions as follows:

$$S = \sqrt{\frac{1}{|n|}\sum_{i=1}^{n}(d_i - \bar{d})^2} \tag{20}$$

, in which

$$d_i = min_{k\in n \,\Lambda\, k\neq 1} \textstyle\sum_{m=1}^{2}\left|f_m^i - f_m^k\right| \qquad , \qquad \bar{d} = \sum_{i=1}^{n}\frac{d_i}{|n|} \tag{21}$$

The measured distance is equal to the lowest value of the sum of the absolute value of the difference among the values of the objective functions between the i-th solution and the solutions located in the final infinite set. It is noteworthy that this distance metric is different from the minimum Euclidean distance criterion between solutions. The above criterion measures the standard deviation for different d_i values. When the solutions are uniformly next to each other, then the value of s will also be small. Thus, the algorithm whose final non-dominated solutions have a small spacing value will be more desired.

b) Number of Non-dominated Solutions: This metric shows the number of members in the first front of the last solutions in the population.

c) Mean Ideal Distance (MID): In the multi-objective functions based on the Pareto approach, one of the objectives is to have the fronts as close as possible to the origin of the coordinates. Hence, this index calculates the distance of the fronts from the best value of the population (Rabiei et al., 2023). The smaller this index, the more desired it would be.

d) Dispersion Index: This metric indicates the extent of Pareto solutions generated by an algorithm. The larger values of this index indicate that the solutions are better dispersed, meaning there are fewer identical solutions and the problem has a greater variety of solutions. This metric is calculated as follows:

$$d_i' = max\left\{\sum_{m=1}^{M}(f_m^i - f_m^j)^2\right\} \tag{22}$$

$$DM = \sqrt{\sum_{i=1}^{N} d_i'} \tag{23}$$

In the above formula, f_m^i and f_m^j are the *m*-th objective function values of the two Pareto solutions *i* and *j*.

e) Multi-objective Coefficient of Variation (MOCV): This metric, which was presented by Rahmati et al. (2013), is calculated by dividing the Mean Ideal Distance (MID) by the Diversity Metric (DM). Lower values of this metric are more desired for comparing metaheuristic algorithms.

VI. PERFORMANCE EVALUATION

A. Experimental Setting

Since the proposed model is new, 30 sample problems are used in Table VI, based on the central limit theorem and the normal distribution. These problems have been randomly generated and are used to solve the proposed algorithms. In Table VI, it can be observed that the dimensions have increased over time to measure the efficiency of the algorithms in high dimensions.

Table VI. Features of random sample problems

Sample problem No.	Number of activities	Number of Machines
1	3	3
2	8	3
3	11	3
4	12	4
5	10	5
6	12	5
7	15	5
8	22	5
9	15	6
10	14	7
11	16	7
12	8	8

Continue Table VI. Features of random sample problems

Sample problem No.	Number of activities	Number of Machines
13	5	8
14	13	8
15	20	4
16	20	6
17	30	6
18	22	5
19	28	5
20	25	7
21	32	8
22	34	8
23	30	9
24	36	9
25	36	10
26	38	10
27	40	11
28	42	11
29	40	12
30	45	12

The range of random number generation is also presented in Table VII. It is noteworthy that the tool change time is 3, and the upper limit of wear has been considered to be 50.

Table VII. Random number generation interval

Parameter	Range
p_{ijkh}	Uniform[10.40]
sp_j	Uniform[1.3]
pp_{jk}	Uniform[30.65]
w_k	Uniform[2.5]
r	3

B. Analysis of Results

All algorithms mentioned in this research have been programmed and executed using MATLAB R2022a software in a Windows 10 (64-bit) environment. In the following, the performance of each of these two algorithms is compared with respect to the evaluation criteria.

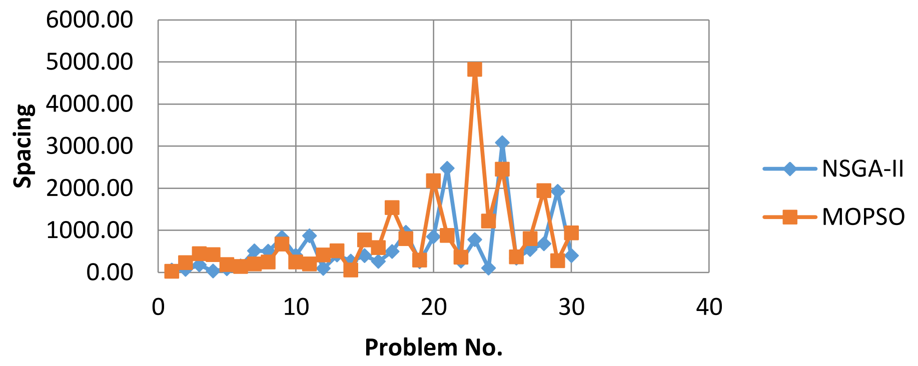

Fig. 5. Spacing index for NSGA-II and MOPSO algorithms

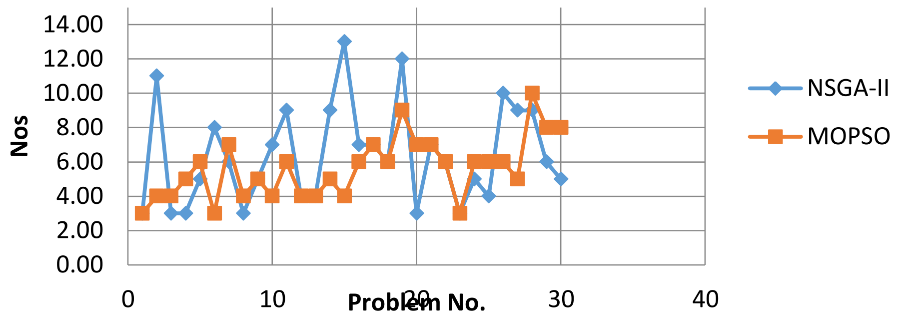

Fig. 6. NOS index for NSGA-II and MOPSO algorithms

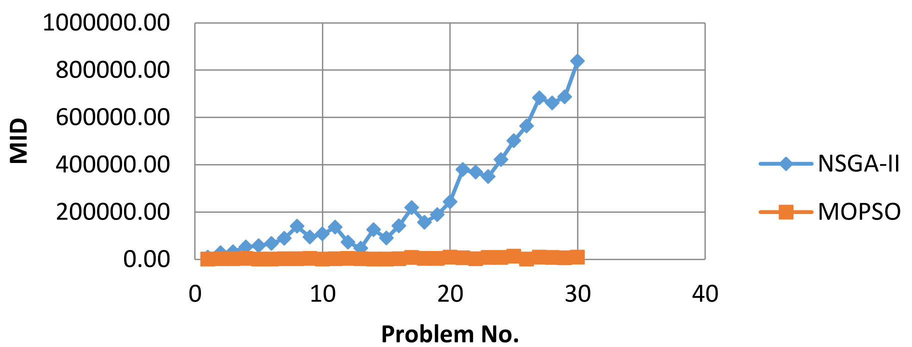

Fig. 7. MID index for NSGA-II and MOPSO algorithms

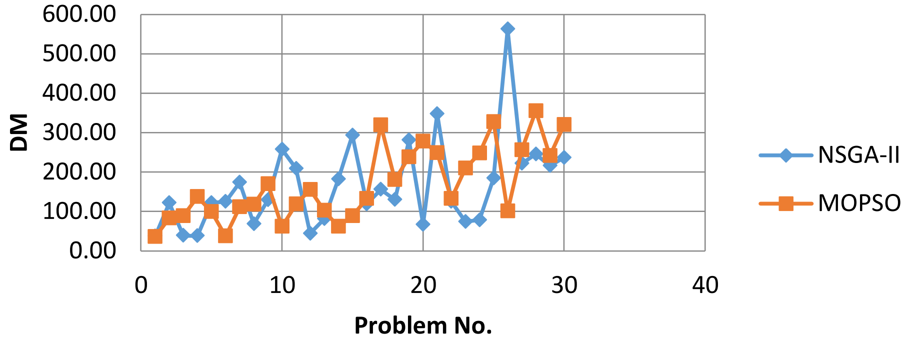

Fig. 8. DM index for NSGA-II and MOPSO algorithms

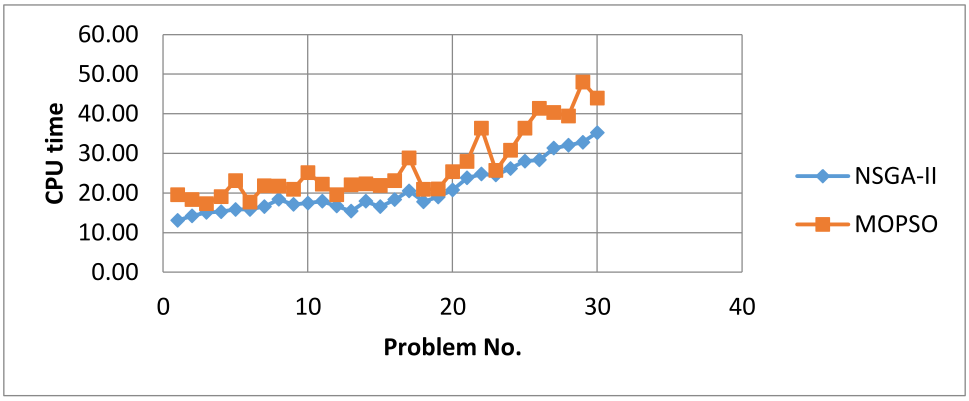

Fig. 9. CPU time index for NSGA-II and MOPSO algorithms

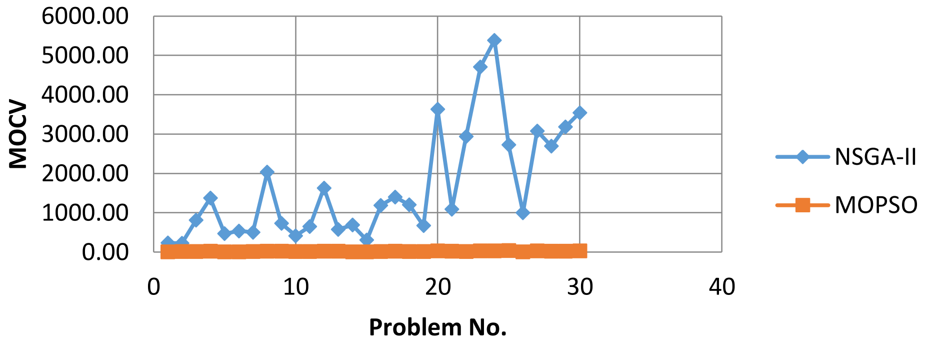

Fig. 10. MOCV index for NSGA-II and MOPSO algorithms

Fig. 5 illustrates the spacing criterion in 30 problem examples, presented as a diagram. A visual inspection of this graph shows almost no difference in this criterion until the execution of 20 tests. Still, from the 20th test onwards, MOPSO is superior in some cases, while NSGA-II is superior in others. It should be noted that the lower this criterion is, the better. Fig. 6 illustrates the number of proposed solutions in each algorithm along the Pareto front. The decision-maker can choose any of the solutions for implementation according to the existing conditions. Fig. 6 shows that NSGA-II gives better results in most cases. The MID metric shows the distance from the ideal point. Therefore, the lower this criterion, the better. Fig. 7 shows that in 30 of the test cases of the problem, this criterion is superior to the NSGA-II algorithm in most of the MOPSO tests. Fig. 8 shows the dispersion metric. In other words, higher values of this criterion indicate greater diversity in the solutions. Examining this criterion according to Fig. 8 shows that NSGA-II is superior in some cases, while MOPSO is superior in others. Fig. 8 shows that the NSGA-II algorithm is superior to MOPSO in almost all test cases. The MOCV criterion is illustrated in Fig. 10. As evident from the figure, the MOPSO algorithm yields lower values than NSGA-II in most of the tested cases. Here, the MOPSO algorithm is superior to the NSGA-II.

Now, to inspect the superiority of each of the algorithms in each evaluation metric, the Analysis of Variance (ANOVA) should be conducted. In each test, if the *p*-value is smaller than 0.05, it means that there is a significant difference between the two algorithms in that metric. In such a case, the performance of one of the algorithms is better than the other one regarding the specified metric; otherwise, there is no significant difference between the two algorithms.

Table VIII. Variance analysis of Spacing metric for NSGA-II and MOPSO algorithms

Source	SS	df	MS	F	Prob>F
Columns	586945.8	1	586945.8	0.8	0.3744
Error	42478647	58	732390.5		
Total	43065593	59			

As shown in Table VIII, the p-value is 0.3744, which is greater than 0.05. Thus, there is no significant difference between these two algorithms in the Spacing metric.

Table IX. Variance analysis of NOS metric for NSGA-II and MOPSO algorithms

Source	SS	df	MS	F	Prob>F
Columns	9.6	1	9.6	1.71	0.1967
Error	326.4	58	5.62759		
Total	336	59			

As shown in Table IX, the p-value is 0.1967, which is greater than 0.05. Thus, there is no significant difference between these two algorithms in the NOS metric.

As shown in Table X, the p-value is almost equal to zero and this value is less than 0.05. Therefore, there is a significant difference between the two algorithms concerning the MID metric. According to Fig. 11, the MOPSO algorithm has a better performance than the NSGA-II concerning the MID metric.

Table X. Variance analysis of MID metric for NSGA-II and MOPSO algorithms

Source	SS	df	MS	F	Prob>F
Columns	$9.99*10^{11}$	1	$9.99*10^{11}$	32.81	$1.82*10^{-7}$
Error	$1.62*10^{11}$	58	$2.8*10^{10}$		
Total	$2.5*10^{12}$	59			

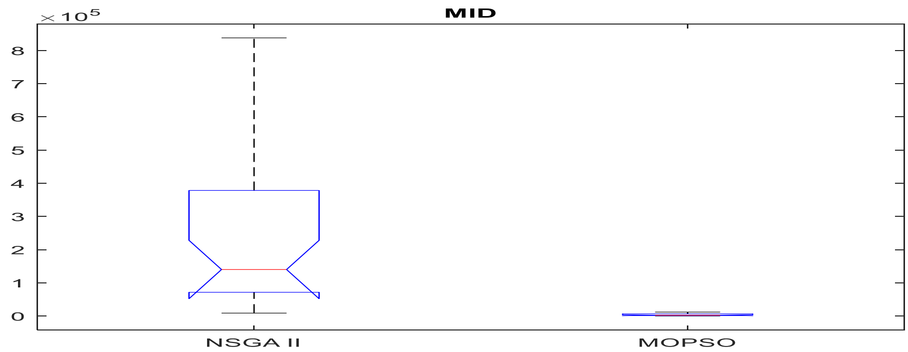

Fig. 11. MID metric obtained from NSGA-II and MOPSO algorithms

Table XI. Variance analysis of DM metric for NSGA-II and MOPSO algorithms

Columns	SS	df	MS	F	Prob>F
Columns	129.4	1	129.4	0.01	0.9128
Error	619632.1	58	10683.3		
Total	619761.5	59			

As shown in Table XI, the p-value is equal to 0.9128, which is larger than 0.05. Hence, there is no significant difference between these two algorithms concerning the DM metric.

Table XII. Variance analysis of MOCV metric for NSGA-II and MOPSO algorithms

Source	SS	df	MS	F	
Columns	4.01*107	1	40118684	40.54	3.36*10-8
Error	5.74*107	58	989642.2		
Total	9.75*107	59			

As shown in Table XII, the p-value is almost equal to zero and this value is greater than 0.05. Therefore, there is a significant difference between these two algorithms in the MOCV metric. According to Fig. 12, the MOPSO algorithm exhibits better performance in terms of the MOCV metric.

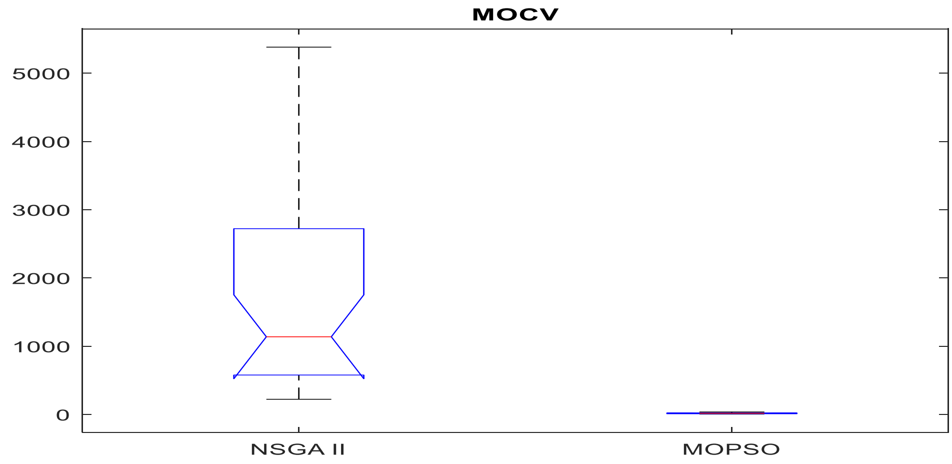

Fig. 12. MOCV metric for NSGA-II and MOPSO algorithms

Table XIII. Variance analysis of CPU time for NSGA-II and MOPSO algorithms

Source	SS	df	MS	F	Prob>F
Columns	510.65	1	510.65	8.94	0.0041
Error	3312.88	58	57.119		
Total	3823.53	59			

As shown in Table XIII, the p-value is equal to 0.0041 and this value is less than 0.05. Hence, there is a significant difference in CPU time between these two algorithms. As CPU time decreases, the algorithm becomes more efficient. According to Fig. 13, the NSGA-II algorithm performs better in this criterion.

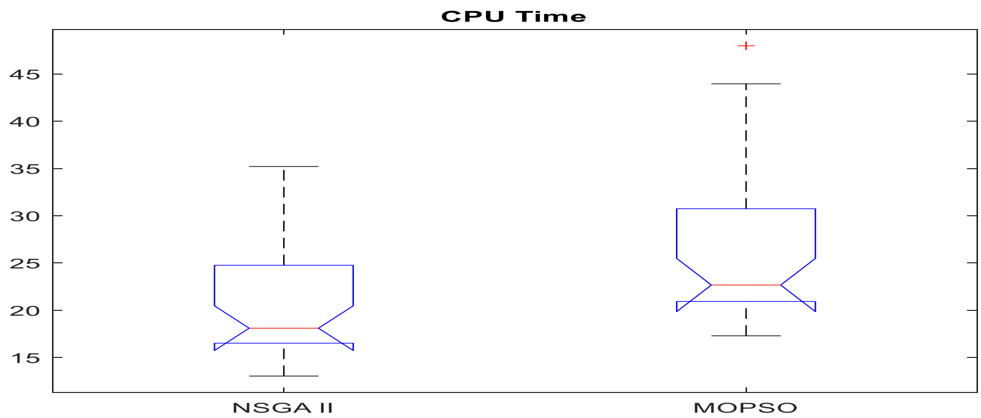

Fig. 13. The efficiency of NSGA-II and MOPSO algorithms concerning the CPU time criterion

As shown in Table XIV, in the exact solution, as the amount of wear increases, energy consumption increases, and total task time decreases. As the table shows, at a given speed, the maximum wear level increases as the speed level increases. This in turn reduces the job completion time and reduces energy consumption.

Table XIV. The impact of wear level changes on the amount of energy consumption and makespan

	T	Energy consumption	Makespan
1	25	8419.73	96.44
2	50	9019.57	89.8
3	100	9632.95	85.89

Fig. 14 shows the Pareto front diagram of the changes in energy consumption and makespan. As evident from the figure, at higher machine speeds, less time and more energy are consumed. Conversely, the lower the speed, the less energy is consumed and the longer the time to complete the work. For example, in the NSGA-II algorithm, the lowest amount of energy consumption, with a value of 8202.3, corresponds to the highest time, with a value of 124.72. The lowest time value of 103.77 jobs has the highest amount of energy consumption with a value of 8871.16. The same principle applies to the MOPSO algorithm.

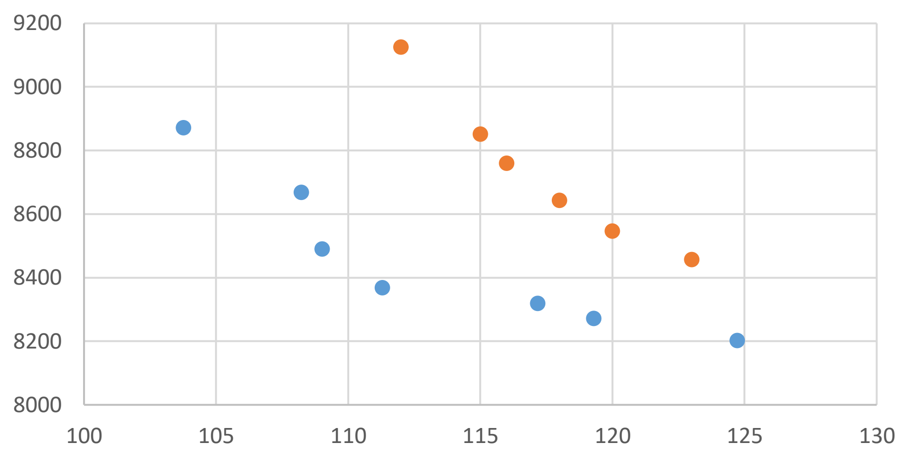

Fig. 14. Pareto front diagram of the changes in energy consumption and makespan

Table XV. The results obtained from the solution NSGA-II and MOPSO algorithms

NSGA-II			MOPSO		
Solution ID	C_{max}	TEC	Solution ID	C_{max}	TEC
1	103.77	8871.6	1	112.34	9125.2
2	108.23	8667.7	2	115.46	8851.3
3	109.02	8489.6	3	116.45	8758.9

Continue Table XV. The results obtained from the solution NSGA-II and MOPSO algorithms

NSGA-II			MOPSO		
C_{max}	TEC	C_{max}	TEC	C_{max}	TEC
4	111.29	8368.2	4	117.89	8642.8
5	117.17	8318.3	5	120.02	8546.3
6	119.29	8270.9	6	123.12	8457.6
7	124.72	8202.3	-	-	-

C. Discussion and Managerial Insights

By using the proposed modeling, the amount of energy consumption and the completion time of the last job are set to optimal values. Therefore, industrial units can achieve good results in saving costs and reducing environmental effects. Additionally, by reducing the time required to complete the last job, more products can be produced in a shorter timeframe. Therefore, our proposed mathematical model will increase the profitability of the industrial unit by reducing the costs associated with energy consumption and increasing production time. With this modeling, the right time will be spent on the right activity, and while increasing productivity, it will also cause the correct prioritization of the work. It should be noted that the implementation of this mathematical model will lead to the timely delivery of the product, and on the other hand, the timely replacement of tools will increase the level of product quality. This will increase customer satisfaction, sell more products, and ultimately increase the profitability of the industrial unit.

Our analysis has revealed that more energy is consumed in a shorter period of time, and less energy is consumed over a longer period. Therefore, managers can use the work policies they consider in each solution. Our results showed that MOPSO outperforms NSGA-II in terms of MOCV and MID metrics. Conversely, NSGA-II outperforms MOPSO in terms of CPU time. In other words, if managers' goal is to reduce the distance from the ideal solution or reduce the multi-objective coefficient of variation, MOPSO is recommended.

VII. CONCLUSION

This research addressed the effect of real changes due to tool wear with tool changes in the flow-shop environment. First, the relationship between tool changes, energy consumption, and job completion time was formulated in the flow-shop scheduling problem. Then, the multi-objective scheduling problem was solved by jointly minimizing the energy consumption and the total job completion time. In this regard, the machine processing speed and the tool wear level in the flow-shop environment were considered. Due to the NP-hardness of the problem, it was solved using two well-known metaheuristic methods: the Non-dominated Sorting Genetic Algorithm (NSGA-II) and Multi-Objective Particle Swarm Optimization (MOPSO).

The experimental results revealed that MOPSO outperforms NSGA-II in terms of MOCV and MID metrics. Conversely, NSGA-II outperforms MOPSO in terms of CPU time. Also, no significant difference was observed between the two algorithms in other criteria. The results of this study reveal interesting implications for managers of manufacturing units. If the goal of managers is to reduce the distance from the ideal solution or to reduce the multi-objective coefficient of variation, it is recommended to use MOPSO. Conversely, for managers who want to spend less time, NSGA-II is recommended.

As future research, several suggestions are offered for planning managers in manufacturing and industrial units. Considering the time it takes for parts to move between machines may have a significant impact on scheduling in the real world. Also, considering unexpected events such as machine failure is a suggestion for future research. The planning for dynamic flow-shop environments can also be an interesting area of research. Additionally, utilizing new metaheuristic methods that can establish a better tradeoff between exploration and exploitation may lead to further improvements in the quality of solutions.

REFERENCES

Arasteh, A., (2022). Mathematical modeling of flexible production lines with different part types on unreliable machines by a priority rule. *Journal of Quality Engineering and Production Optimization*, *7*(2), 34-59.

Baker, K. R. (2013). Computational results for the flowshop tardiness problem. *Computers & Industrial Engineering*, *64*(3), 812-816.

Bruzzone, A. A., Anghinolfi, D., Paolucci, M., & Tonelli, F. (2012). Energy-aware scheduling for improving manufacturing process sustainability: A mathematical model for flexible flow shops. *CIRP annals, 61(1), 459-462.*

Chaudhry, I. A., Elbadawi, I. A., Usman, M., & Tajammal Chugtai, M. (2018). Minimising total flowtime in a no-wait flow shop (NWFS) using genetic algorithms. *Ingeniería e Investigación, 38(3): 68-79.*

Chen, T. L., Cheng, C. Y., & Chou, Y. H. (2020). Multi-objective genetic algorithm for energy-efficient hybrid flow shop scheduling with lot streaming. *Annals of Operations Research*, *290*, 813-836.

Coello, C. A. C. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering, 191*(11-12), 1245-1287.

Dai, M., Tang, D., Giret, A., Salido, M. A., & Li, W. D. (2013). Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. *Robotics and Computer-Integrated Manufacturing, 29(5), 418-429.*

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

Ding, J. Y., Song, S., & Wu, C. (2016). Carbon-efficient scheduling of flow shops by multi-objective optimization. *European Journal of Operational Research, 248(3), 758-771.*

Enayati, M., Yousefi Nejad Attari, M., & Lotfian Delouyi, F. (2023). Optimizing Open Shop Scheduling: Minimizing Makespan through Whale Optimization Algorithm and Transportation Time Consideration. *Journal of Quality Engineering and Production Optimization*, *8*(1), 133-150.

Esmaeili, M., Ahmadizar, F., & Sadeghi, H. (2021). Minimizing the sum of earliness and tardiness in single-machine scheduling. *Journal of Quality Engineering and Production Optimization, 6(2),* 59-78.

Fontes, D.B., Homayouni, S.M. & Fernandes, J.C., (2024). Energy-efficient job shop scheduling problem with transport resources considering speed adjustable resources. *International Journal of Production Research*, *62*(3), 867-890.

Gadaleta, M., Pellicciari, M., & Berselli, G. (2019). Optimization of the energy consumption of industrial robots for automatic code generation. *Robotics and Computer-Integrated Manufacturing, 57,* 452-464.

Geng, K., Ye, C., Dai, Z. H., & Liu, L. (2020). Bi-Objective Re-Entrant Hybrid Flow Shop Scheduling considering Energy Consumption Cost under Time-of-Use Electricity Tariffs. *Complexity*, *2020*(1), 8565921.

Gholizadeh, H., Fazlollahtabar, H., Fathollahi-Fard, A. M., & Dulebenets, M. A. (2021). Preventive maintenance for the flexible flowshop scheduling under uncertainty: A waste-to-energy system. *Environmental Science and Pollution Research,* 1-20.

Ghorbanzadeh M., & Ranjbar, M. (2023). Energy-aware production scheduling in the flow shop environment under sequence-dependent setup times, group scheduling and renewable energy constraints. *European Journal of Operational Research, 307(2),* 519-537.

Goli, A., Ala, A., & Hajiaghaei-Keshteli, M. (2023). Efficient multi-objective meta-heuristic algorithms for energy-aware non-permutation flow-shop scheduling problem. *Expert Systems with Applications*, *213*, 119077.

Ham, A., Park, M. J., & Kim, K. M. (2021). Energy-Aware Flexible Job Shop Scheduling Using Mixed Integer Programming and Constraint Programming. *Mathematical Problems in Engineering*, *2021*(1), 8035806.

Huang, J., Pan, Q., & Chen, Q. (2019, October). A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem with sequence-dependent setup times. In *IOP conference series: materials science and engineering* (Vol. 646, No. 1, p. 012037).

IOP Publishing.Kumar, N. & Das, A.K., (2024). Critical review on tool wear and its significance on the µ-ECSM process. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *238*(4), pp.907-925.

Li, J. Q., Sang, H. Y., Han, Y. Y., Wang, C. G., & Gao, K. Z. (2018). Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions. *Journal of Cleaner Production, 181, 584-598.*

Liou, C. D., & Hsieh, Y. C. (2015). A hybrid algorithm for the multi-stage flow shop group scheduling with sequence-dependent setup and transportation times. *International Journal of Production Economics, 170, 258-267.*

Mahdavi, K., Mohammadi, M. & Ahmadizar, F., (2023). Efficient scheduling of a no-wait flexible job shop with periodic maintenance activities and processing constraints. *Journal of Quality Engineering and Production Optimization*, *8*(1), 33-56.

Mansouri, S. A., Aktas, E., & Besikci, U. (2016). Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption. *European Journal of Operational Research, 248*(3), 772-788.

Meng, L., Zhang, C., Shao, X., Ren, Y., & Ren, C. (2019). Mathematical modelling and optimisation of energy-conscious hybrid flow shop scheduling problem with unrelated parallel machines. *International Journal of Production Research, 57(4),* 1119-1145.

Mokhtari, H., & Hasani, A. (2017). An energy-efficient multi-objective optimization for flexible job-shop scheduling problem. *Computers & Chemical Engineering*, *104,* 339-352.

Mousavi, T.S., Shankar, A., Rezvani, M.H. & Ghadiri, H. (2024). Entropy-aware energy-efficient virtual machine placement in cloud environments using type information. *Concurrency and Computation: Practice and Experience*, *36*(15), e7950.

Mouzon, G., Yildirim, M. B., & Twomey, J. (2007). Operational methods for minimization of energy consumption of manufacturing equipment. *International Journal of Production Research, 45*(18-19), 4247-4271.

Nilakantan, J. M., Huang, G. Q., & Ponnambalam, S. G. (2015). An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems. *Journal of Cleaner Production, 90, 311-325.*

Oda, Y., Fujishima, M. & Takeuchi, Y., (2015). Energy-saving machining of multi-functional machine tools. *International Journal of Automation Technology*, *9*(2), 135-142.

Ponzo, F. C., Antonio, D. C., Nicla, L., & Nigro, D. (2021). Experimental estimation of energy dissipated by multistorey post-tensioned timber framed buildings with anti-seismic dissipative devices. *Sustain. Struct*, *1*(000007).

Rabiei, P., Arias-Aranda, D. & Stantchev, V. (2023). Introducing a novel multi-objective optimization model for Volunteer assignment in the post-disaster phase: Combining fuzzy inference systems with NSGA-II and NRGA. *Expert Systems with Applications*, *226*, 120142.

Rahmati, S. H. A., Hajipour, V., & Niaki, S. T. A. (2013). A soft-computing Pareto-based meta-heuristic algorithm for a multi-objective multi-server facility location problem. *Applied Soft Computing, 13*(4), 1728-1740.

Rastgar, I., Rezaean, J., Mahdavi, I., & Fattahi, P. (2021). Opportunistic maintenance management for a hybrid flow shop scheduling problem. *Journal of Quality Engineering and Production Optimization, 6*(2), 17-30.

Rezvan, M. T., Gholami, H., & Zakerian, R. (2021). A new algorithm for solving the parallel machine scheduling problem to maximize benefit and the number of jobs processed. *Journal of Quality Engineering and Production Optimization, 6(2),* 115-142.

Schott, J. R. (1998). Estimating correlation matrices that have common eigenvectors. *Computational statistics & data analysis, 27*(4), 445-459.

Schulz, S., Buscher, U., & Shen, L. (2020). Multi-objective hybrid flow shop scheduling with variable discrete production speed levels and time-of-use energy prices. *Journal of Business Economics, 90, 1315-1343.*

Schulz, S., Neufeld, J. S., & Buscher, U. (2019). A multi-objective iterated local search algorithm for comprehensive energy-aware hybrid flow shop scheduling. *Journal of cleaner production, 224, 421-434.*

Sekkal, D. N. & Belkaid, F. (2023). A multi-objective optimization algorithm for flow shop group scheduling problem with sequence dependent setup time and worker learning. *Expert Systems with Applications, 233,* 120878.

Shao, W., Shao, Z., & Pi, D. (2022). A multi-neighborhood-based multi-objective memetic algorithm for the energy-efficient distributed flexible flow shop scheduling problem. *Neural Computing and Applications, 34(24), 22303-22330.*Shen, L., Dauzère-Pérès, S., & Maecker, S. (2023). Energy cost efficient scheduling in flexible job-shop manufacturing systems. *European Journal of Operational Research, 310*(3), 992-1016.

Singh, H., Oberoi, J. S., & Singh, D. (2021). Multi-objective permutation and non-permutation flow shop scheduling problems with no-wait: a systematic literature review. *RAIRO-Operations Research, 55(1), 27-50.*

Sola, A. V., & Mota, C. M. (2020). Influencing factors on energy management in industries. *Journal of Cleaner Production, 248,* 119263.

Soleimani, H., Ghaderi, H., Tsai, P. W., Zarbakhshnia, N., & Maleki, M. (2020). Scheduling of unrelated parallel machines considering sequence-related setup time, start time-dependent deterioration, position-dependent learning and power consumption minimization. *Journal of Cleaner Production*, *249*, 119428.

Stewart, R., Raith, A., & Sinnen, O. (2023). Optimising makespan and energy consumption in task scheduling for parallel systems. *Computers & Operations Research*, *154,* 106212.

Todorov, G. N., Volkova, E. E., Vlasov, A. I., & Nikitina, N. I. (2019). Modeling energy-efficient consumption at industrial enterprises. *International Journal of Energy Economics and Policy, 9(2),* 10-18.

Unal, R. & Dean, E.B. (1990, January). Taguchi approach to design optimization for quality and cost: an overview. In *1991 Annual conference of the international society of parametric analysts*.

Utama, D. M. (2021, March). Minimizing number of tardy jobs in flow shop scheduling using a hybrid whale optimization algorithm. In *Journal of Physics: Conference Series* (Vol. 1845, No. 1, p. 012017). IOP Publishing.

Utama, D. M., Primayesti, M. D., Umamy, S. Z., Kholifa, B. M. N., & Yasa, A. D. (2023). A systematic literature review on energy-efficient hybrid flow shop scheduling. *Cogent Engineering*, *10*(1), 2206074.

Wang, F., Deng, G., Jiang, T., & Zhang, S. (2018). Multi-objective parallel variable neighborhood search for energy consumption scheduling in blocking flow shops. *Ieee Access*, *6,* 68686-68700.

Wang, S., Yao, J., Zhang, Z., & Zhang, K. (2019). Design and numerical simulations of a temperature tunable hybrid structure metamaterials. *Journal of Nanophotonics, 13(3), 036019-036019.*

Wang, Y. J., Wang, G. G., Tian, F. M., Gong, D. W., & Pedrycz, W. (2023). Solving energy-efficient fuzzy hybrid flow-shop scheduling problem at a variable machine speed using an extended NSGA-II. *Engineering Applications of Artificial Intelligence*, *121*, 105977.

Wen, X., Sun, Y., Ma, H. L., & Chung, S. H. (2023). Green smart manufacturing: energy-efficient robotic job shop scheduling models. *International Journal of Production Research, 61(17), 5791-5805.*

Wu, X., & Che, A. (2019). A memetic differential evolution algorithm for energy-efficient parallel machine scheduling. Omega, 82, 155-165.

Xin, X., Jiang, Q., Li, S., Gong, S., & Chen, K. (2021). Energy-efficient scheduling for a permutation flow shop with variable transportation time using an improved discrete whale swarm optimization. *Journal of Cleaner Production, 293*, 126121.

Xiong, H., Shi, S., Ren, D., & Hu, J. (2022). A survey of job shop scheduling problem: The types and models. *Computers & Operations Research*, *142*, 105731.

Yan, J., Li, L., Zhao, F., Zhang, F., & Zhao, Q. (2016). A multi-level optimization approach for energy-efficient flexible flow shop scheduling. *Journal of cleaner production*, *137*, 1543-1552.

Zhang, H., Deng, Z., Fu, Y., Lv, L., & Yan, C. (2017). A process parameters optimization method of multi-pass dry milling for high efficiency, low energy and low carbon emissions. *Journal of Cleaner Production 148: 174-184.*

Zhang, J., Li, R., Yang, H., Qing, Z. & Chen, S. (2024). Tool wear characteristics of interrupted-cutting Zr-based bulk metallic glasses with cemented carbide tool. *The International Journal of Advanced Manufacturing Technology*, *133*(5), 2623-2637.

Zhang, L., Deng, Q., Gong, G. & Han, W., (2020). A new unrelated parallel machine scheduling problem with tool changes to minimise the total energy consumption. *International Journal of Production Research*, *58*(22), 6826-6845.

Zhang, W., Geng, H., Li, C., Gen, M., Zhang, G. & Deng, M., (2023). Q-learning-based multi-objective particle swarm optimization with local search within factories for energy-efficient distributed flow-shop scheduling problem. *Journal of Intelligent Manufacturing*, *36*(1), 185-208.