Advances in Industrial Engineering, Winter 2021, 55(1): 91-113
DOI: 10.22059/jieng.2021.326682.1788
RESEARCH PAPER

Unrelated Parallel Machines Scheduling with Sequence-Dependent Setup Times to Minimize Makespan and Tariff Charged Energy Consumption

Erfaneh Karimi [a], Taha Keshavarz [b,*], Majid Shakhsi-Niaei [a]

a. *Department of Industrial Engineering, Yazd University, Yazd, Iran.*
b. *Department of Industrial Engineering, Semnan University, Semnan, Iran*.

Received: 05 July 2021, Revised: 14 August 2021, Accepted: 15 August 2021
© University of Tehran 2021

Abstract

An appropriate trade-off between total electricity costs and makespan can lead to good production planning and reduce unnecessary energy consumption. Time-of-use (TOU) electricity pricing policy has been executed in many countries, enabled industrial consumers with high energy consumption to reduce their energy costs. In this study, an unrelated parallel machines scheduling problem is considered for minimizing makespan and also energy consumption costs. Due to the importance of sequence-dependent setup times in production environments, they are considered according to the restricted duration of time periods under TOU policy. These considerations are added to the current literature. A mixed-integer bi-objective mathematical model is presented and the ε-constraint method is applied to solve small and also medium-sized instances. Because the problem is shown to be NP-hard, several large-sized instances are approximately solved using Multiple Objective Particle Swarm Optimization algorithm, and Multiple Objective Simulated Annealing algorithm. Computational experiments are conducted on randomly generated data. The results show the efficiency and appropriate performance of the proposed methods.

Keywords:
Unrelated Parallel Machines Scheduling;
Makespan;
Energy Consumption;
Time-of-Use Electricity Price;
Sequence-Dependent Setup Times

Introduction

Due to rapid economic growth and expanding population, energy demand is overgrowing. In most manufacturing industries, more than 50% of the price of a product is tied to the energy required to produce it. In 2019, the industrial sector of the U.S. accounted for 35% of total end-use energy consumption [1]. Recent researches show that more than 90% of the environmental issues are based on electrical energy consumption in the utilization phase [2]. As energy costs increase, energy consumption gains more importance in the industry.

Electricity demand during the day is unbalanced, which leads to the low efficiency of electricity consumption. To improve electricity efficiency, different tariffs have been applied to balance energy use in different time periods. Three types of time-dependent energy pricing are introduced: 1) time of use pricing (TOU), 2) real-time pricing (RTP), and 3) critical peak pricing (CPP), among which the TOU pattern is the commonly used and studied policy [3].

Based on TOU policy, electricity demand is the most important factor in determining the price of electricity and the day is divided into three periods, i.e. on-peak, mid-peak, and off-

[*] Corresponding author: (T. Keshavarz)
Email: taha_keshavarz@semnan.ac.ir

peak. The plan encourages manufacturers to prefer courses with low electricity prices to those with high prices to avoid high electricity costs.

There are various ways to minimize energy consumption which are often capitalizing and require strategic decisions. For example, buying new machines with lower energy consumption can be useful in energy saving. But, this decision requires more investment and may result in more pressure on companies [4]. One of the most important actions that does not require much investment and can only be controlled and improved by changing the production process is scheduling optimization. Incorporating energy metrics into production scheduling can reduce production costs.

In unrelated parallel machines scheduling problems, some specific machines are available from the beginning, which are the same as what machines do, and the only difference is the processing speeds of the different machines. Various processing speeds lead to different energy consumptions.

Considering setup times is another area of research that focuses on practical considerations. In most scheduling studies, setup times are neglected or considered as a portion of process time, which streamlines the analysis but makes the model unrealistic. Setup time relates to all the activities performed to prepare the main stage of the process. In general, two types of setup times are introduced. The first type just depends on the job itself, i.e., sequence-independent setup times. Another type depends on the current job and the job that has been done before it, which is called sequence-dependent setup times. As an example, in a painting process, cleaning the machine while a color change is necessary which its length depends on the previous and next colors [5].

In this research, a bi-objective mathematical model is formulated for the unrelated parallel machines scheduling with TOU energy pricing which minimizes the makespan and tariff charged energy consumption cost. Because of the importance of sequence-dependent setup times for parallel machines and the more realistic assumptions of the problem, this constraint is also considered.

The rest of the article is arranged as follows. Reviewing the relevant literature is presented in Section 2. The problem specifications are described in Section 3. The ε-constraint method, the Multiple Objective Particle Swarm Optimization (MOPSO), and the Multiple Objective Simulated Annealing (MOSA) algorithms for the research problem are introduced in Section 4. The computational results are presented in Section 5. Finally, Section 6 concludes the research and also proposes several future research directions.

Literature review

Recently, due to the increasing cost of energy consumption, energy-efficient scheduling has given noteworthy attention. In this section, the studies that have considered the energy-efficient unrelated parallel machines scheduling problem are reviewed.

Energy-efficient unrelated parallel machines scheduling without applying energy tariffs

Angel et al. [6] considered an energy-aware unrelated parallel machines scheduling problem by considering machine-dependent release date for each job to minimize energy consumption and also average weighted completion times. The LP Rounding procedure is first implemented for solving the problem. Since the proposed algorithm does not guarantee constant quality and stability in any implementation of the algorithm, so the Randomized Rounding algorithm is used for this purpose.

Liang et al. [7] considered an unrelated parallel machines scheduling problem for minimizing the weighted summation of total tardiness and energy consumption. The machine setup and idle energy consumption are considered. For each job, a specific due date is applied. Ant Colony algorithm (ACO) and innovative rules of the ATC method are used for solving this problem.

Li et al. [8] discussed unrelated parallel machines scheduling problems in three modes: machine processing, idle, and warm-up. Their objective function is minimizing total tardiness and energy consumption. Due to the complexity of the model, several heuristic algorithms are recommended.

Cota et al. [9] considered an unrelated parallel machines scheduling problem that considered sequence-dependent setup times for minimizing the makespan and the total consumption of electricity. A mixed-integer linear programming (MILP) formulation is presented which tackles independent and non-preemptive jobs. Furthermore, a novel meta-heuristic algorithm called Smart Pool is used to find a solution near the Pareto front in a limited computational budget.

Wu and Che, [4] studied unrelated parallel machines with total energy consumption and makespan minimization objectives, considering the speed-scaling of machines. A Memetic Differential Evolution algorithm is proposed as a solution approach that includes selecting and rotating operators, speed adjustment, and swapping job-machine.

Soleimani et al. [10] studied a scheduling problem of unrelated parallel machines considering the simultaneous effects of start-time-dependent deterioration, position-dependent learning, and sequence-dependent setup times. A MIP model is presented to minimize the mean weighted tardiness and power consumption. To solve the problem, Genetic Algorithm (GA), Cat Swarm Optimization (CSO), and Interactive Artificial Bee Colony (IABC), are implemented.

Zhu and Tianyu, [11] studied an unrelated parallel machine problem by considering setup times, ready times, and resource constraints. They developed a multi-objective model to minimize total energy consumption and weighted completion time. To solve the problem, an immune clone algorithm based on a non-domination rank selection strategy, by applying clone operators, neighborhood search operators, and elite preservation operators, is proposed.

Energy-efficient unrelated parallel machines scheduling with applying energy tariffs

Moon et al. [12] addressed a bi-objective unrelated parallel machines scheduling problem to minimize the total electricity cost and makespan under RTP tariffs. In the proposed model, idle time is allowed, and the production schedule is considered for one day. Genetic algorithm (GA), shifted GA (SGA), and Hybrid Inserted Genetic algorithm (HIGA) are used.

Ding et al. [13] tackled a time-interval-based mixed-integer model under the TOU tariffs for minimizing the cost of electricity consumption so that the total completion time does not exceed a predetermined production deadline. To solve the problem, they proposed a column generation (CG) method using the Dantzig–Wolfe decomposition.

Che et al. [14] considered the TOU pattern for minimizing the electricity cost. To solve small-sized instances, an improved MILP model is designed, and for large-sized instances, they suggested a two-stage heuristic. Then they validated the proposed model and heuristics with a real-world instance.

Cheng et al. [15] improved the MILP model of Ding et al. [13] for unrelated parallel machines scheduling with the TOU pattern. In the proposed model, constraints of job completion, machine availability, and non-preemption are modified to offer a more accurate and comprehensive model.

Abikarram et al. [16] considered demand charges in unrelated parallel machines environment under the RTP tariffs. Further, they investigated the sensitivity of the parameters like the number of machines, operation of machines, and so on.

Saberi-Aliabad et al. [17] presented a MILP model for unrelated parallel machines scheduling problems to minimize the cost of consuming energy with TOU energy tariffs. They presented a number of dominance rules and valid inequalities. To solve the problem, a relax heuristic algorithm is used and results are evaluated by lower bound values.

Kurniawan, [18] considered a bi-objective unrelated parallel machine scheduling problem to minimize the total tardiness and energy consumption under TOU tariffs. Two MIP models are proposed, using the time-indexed and disjunctive formulation, and to solved the problem the weighted sum method is applied. Furthermore, the effectiveness of both models is compared.

Research gap and novelties of this study

According to the literature review, most of the researches on the unrelated parallel machines scheduling only consider single-objective of energy consumption minimization or makespan minimization, and few studies have addressed both objectives simultaneously. However, in none of the researches on the energy-efficient unrelated parallel machines scheduling with energy tariffs (to minimize both costs of electricity consumption and makespan), sequence-dependent setup times are not considered while it is a vital consideration in real-world applications. So, in this study, this assumption is also considered to get closer to real-world applications. The two studies that are most similar to our research are Che et al. [14] and Wang et al. [19] which are extended to consider the above considerations.

To solve the proposed model, the ε-constraint method (for small-sized instances) and MOPSO and MOSA algorithms (for large-sized instances) are implemented. Problems with different sizes are randomly generated, and the results are presented in the upcoming sections.

Problem description

In this research, there are N jobs that must be processed on M unrelated parallel machines. All machines and also jobs are available at the starting time of the planning horizon. Each job j, is described by its process time on machine i, t_{ij}, and energy consumption rate on each machine, p_{ij}, $1 \leq i \leq M, 1 \leq j \leq N$. Each machine is able to work on just one job at a period, and each job is allowed to be allocated only to one machine at a period. Preemption is not allowed. Machines are accessible throughout the entire time horizon.

Another issue that is considered in this study is sequence-dependent setup time, s_{ihj}, that is determined based on the job done h, and its next job j, and is different for each machine i.
Also, due to TOU policy, the time horizon is split into K time periods that period $k, 1 \leq k \leq K$, has the energy price c_k. The length of period k is indicated by T_k, where $T = \sum_{k \in K} T_k$.

Also, if job j is allocated to machine i, the possible maximum time periods that job can span is indicated by τ_{ij}, that can be calculated according to the problem parameters as follows:

$$\tau_{ij} = \left(t_{ij} + \sum_{j=1}^{n} \max_{\forall h \in \{0,\dots,n\} \forall i \in \{1,\dots,m\}} s_{ihj} \right) / \min_{k \in \{1,\dots,K\}} T_k \tag{1}$$

The other used parameter is L as a large positive integer:

$$L = \frac{\left(\sum_{j=1}^{n} (\max_{\forall i \in \{1,\dots,m\}} t_{ij} + \max_{\forall h \in \{0,\dots,n\} \forall i \in \{1,\dots,m\}} s_{ihj}) \right)}{m} + \sum_{k=1}^{K} T_k \tag{2}$$

The problem considered in this research is based on the model proposed by Che et al. [14]. The decision variables are as follows:

Indices:

j, h	index of job, $1 \leq j, h \leq N$.
i	index of machine, $1 \leq i \leq M$.
k	index of time period, $1 \leq k \leq K$.

Parameters:

N	the number of jobs.
M	the number of machines.
K	the number of time periods.
t_{ij}	the processing time of job j on machine i.
p_{ij}	The energy consumption of job j on machine i.
c_k	the energy price of time period k.
T_k	The length of time period k.
τ_{ij}	The possible maximum time periods that job j can span on machine i.
s_{ihj}	Setup time for switching job h to job j on machine i.
L	a large positive integer.

Variables:

TEC	Total electricity cost
C_{max}	The makespan.
C_j	Completion time of job j.
x_{ijk}	The actual process duration of job j in k^{th} time interval of machine i.
y_{ijk}	Binary variable which is equal to 1 if job j is allocated to k^{th} time interval of machine i, 0 otherwise.
v_{ij}	Binary variable which is equal to 1 if job j is allocated to machine i, 0 otherwise.
r_{ijk}	Binary variable which is equal to 1 if the processing of job j is started in k^{th} time interval of machine i, 0 otherwise.
z_{ihj}	Binary variable which is equal to 1 if job h is the next job after job j on machine i, 0 otherwise.

The research problem is modeled via Eqs. 3-23.

$$Min\ C_{max} \tag{3}$$

$$TEC = Min\ \sum_{j=1}^{n} \sum_{i=1}^{m} \sum_{k=1}^{K} p_{ij} c_k x_{ijk} \tag{4}$$

$s.t.$

$$\sum_{k=1}^{K} \sum_{i=1}^{m} (x_{ijk}/t_{ij}) = 1 \qquad \forall j = 1, \dots, n \tag{5}$$

$$x_{ijk} \leq t_{ij} y_{ijk} \qquad \forall j = 1, \dots, n\ , \forall i = 1, \dots, m, \forall k = 1, \dots, K \tag{6}$$

$$\sum_{j=1}^{n} x_{ijk} + r_{ijk} * s_{ihj} \leq T_k \qquad \forall i = 1, \dots, m\ , \forall k = 1, \dots, K, \forall h = 0, \dots, n \tag{7}$$

$$\sum_{l=k+2}^{K} y_{ijl} \leq K\left(1 - y_{ijk} + y_{ij(k+1)}\right) \qquad \forall j = 1, \dots, n, \forall i = 1, \dots, m, \forall k = 1, \dots, K-2 \tag{8}$$

$$x_{ijk} \geq T_k\left(y_{ij(k-1)} + y_{ij(k+1)} - 1\right) \qquad \forall j = 1, \dots, n\ , \forall i = 1, \dots, m, \forall k = 2, \dots, K-1 \tag{9}$$

$$v_{ij} \geq \sum_{k=1}^{K} y_{ijk} / \tau_{ij} \qquad \forall j = 1, \dots, n\ , \forall i = 1, \dots, m \tag{10}$$

$$\sum_{i=1}^{m} v_{ij} = 1 \qquad \forall j = 1, \dots, n \tag{11}$$

$$C_j + L\left(1 - \sum_{i=1}^{m} y_{ijk}\right) \geq \sum_{l=1}^{K-1} T_l + \sum_{i=1}^{m} x_{ijk} \qquad \forall j = 1, \dots, n\ , \forall k = 2, \dots, K \tag{12}$$

$$C_j + L\left(1 - z_{ihj}\right) \geq C_h + s_{ihj} + \sum_{k=1}^{k} x_{ijk} \qquad \forall j = 1, \dots, n,\ \forall i = 1, \dots, m\ , \forall h = 0, \dots, n\ , h \neq j \tag{13}$$

$$\sum_{\substack{h=0 \\ h \neq j}}^{n} z_{ihj} = v_{ij} \qquad \forall j = 1, \dots, n\ , \forall i = 1, \dots, m \tag{14}$$

$$\sum_{\substack{j=1 \\ h\neq j}}^{n} z_{ihj} \leq v_{ih} \qquad \forall h = 0, \dots, n \ , \forall i = 1, \dots, m \qquad (15)$$

$$\sum_{k=1}^{k} r_{ijk} \leq v_{ij} \qquad \forall j = 1, \dots, n \ , \forall i = 1, \dots, m \qquad (16)$$

$$r_{ij1} = y_{ij1} \qquad \forall j = 1, \dots, n \ , \forall i = 1, \dots, m \qquad (17)$$

$$r_{ijk} \leq y_{ijk} \qquad \forall j = 1, \dots, n \ , \forall i = 1, \dots, m, \ \forall k = 2, 3, \dots, K \qquad (18)$$

$$r_{ijk} \geq 2 * y_{ijk} - 1 - r_{ij(k-1)} \qquad \forall j = 1, \dots, n \ , \forall i = 1, \dots, m \ , \ \forall k = 1, \dots, K \qquad (19)$$

$$r_{ijk} \leq L * \left(1 - r_{ij(k-1)}\right) \qquad \forall j = 1, \dots, n \ , \forall i = 1, \dots, m \ , \ \forall k = 1, \dots, K \qquad (20)$$

$$C_{max} \geq C_j \qquad \forall j = 1, \dots, n \qquad (21)$$

$$C_j \ , C_{max} \ , x_{ijk} \ , s_{ihj} \ \geq 0 \qquad \forall j = 1, \dots, n \ , \forall i = 1, \dots, m \ , \ \forall k = 1, \dots, K, \ \forall h = 0, \dots, n \qquad (22)$$

$$y_{ijk} \ , r_{ijk}, z_{ihj} \ , v_{ij} \in \{0,1\} \qquad \forall j = 1, \dots, n \ , \forall i = 1, \dots, m \ , \ \forall k = 1, \dots, K \ , \forall h = 0, \dots, n \qquad (23)$$

The objective functions (3) and (4) try to minimize the makespan and the energy consumption cost, respectively. Constraint set (5) guarantees that the total processing times of each job is equal to its processing duration. Constraint set (6) guarantees that if job j is not allocated to k^{th} period of machine i (i.e., $y_{ijk} = 0$), then the actual process duration of job j in k^{th} period of machine i will be equal to 0. Constraint set (7) does not allow the process duration of all jobs assigned in a period on each machine and their setup times to exceed the duration of that period. Constraint set (8) allows job j to be allocated to several periods of machine i only if these periods are consecutive. Due to the non-preemption assumption, constraint set (9) checks that if a job is allocated to periods $k-1$ and $k+1$, then the whole period k should be assigned to that job. Constraint set (10) guarantees the consistent values for binary variables v_{ij} and y_{ijk}. Constraint set (11) ensures that every job can only be allocated to one machine. Constraint sets (12) and (13) define the completion time of job j. Constraint sets (14) and (15) express the relationship of binary variables related to the sequence of jobs and their processing on a machine. Constraint set (16) defines the relationship between the two variables r_{ijk} and v_{ij}. Due to the limited duration of time periods, in order to consider sequence-dependent setup times, the period at which the processing of each job on each machine started must be specified. Therefore, constraint sets (17) to (20) determine the start-up period of the processes. Constraint set (21) defines C_{max}. Constraint sets (22) and (23) declare the range of decision variables.

Based on the three-field notation proposed by Graham et al. [20], our problem is related to $R, TOU|\ s_{ihj}|\{C_{max}, TEC\}$, where R represents the unrelated parallel machine, TOU represents the time-of-use electricity scheme, s_{ihj} represents considering sequence-dependent setup times, C_{max} and TEC indicate the makespan and the total energy consumption costs respectively.

If process duration and energy consumption rate of job j on all the machines are the same ($t_{ij} = t_j, p_{ij} = p_j$) and sequence-dependent setup time is not considered, our problem is reduced to ($Pm, TOU\|\{C_{max}, TEC\}$).

According to Wang et al. [19], the problem of identical parallel machines scheduling with makespan and energy consumption costs objective functions is NP-hard, so our problem is NP-hard too.

Solution Approaches

The concept of the Pareto front is used in multi-objective mathematical programming. The Pareto optimality concept states that although we cannot obtain an optimal solution for all objective functions, simultaneously, we can find several solutions that are better than others in the search space. These solutions are called Pareto-optimal while other points of the search space are the set of dominated solutions. A widely used exact method in this area is the ε-constraint method, which can solve small-sized instances in an acceptable computational time. With increasing the size of the problem and due to its NP-hardness, it becomes difficult to get the Pareto front. So, MOPSO and MOSA algorithms are developed in this research to solve large-sized instances.

The ε-constraint method

In the ε-constraint method, one of the objective functions is considered and the other objective functions are converted to the constraints, called ε-constraints [21]. For this problem, $f_2 = TEC$ is regarded as a constraint.

Multi-objective particle swarm optimization algorithm

The multi-objective particle swarm optimization algorithm (MOPSO) introduced by Coello et al. [22] is widely applied in the field of scheduling problems. For example it is used in [23,24,25,26,27].

The particle swarm optimization algorithm begins by creating a random population of particles. Each particle represents a point in the solution space of the problem. At each iteration of the algorithm, the particle is moved to a better position. Fig. 1 illustrates the flowchart of the MOPSO algorithm.

Generating initial sequences of jobs

To find the sequence of jobs and generate initial solutions, permutations of digits 1 to $n + m - 1$ are generated randomly where n shows the number of jobs and m shows the number of machines). For each generated permutation, from the beginning to the first cell whose its value is more than n, the jobs are allocated to the first machine and their sequence is determined. After this cell, up to the second cell where the numerical value is greater than n, the jobs are allocated to the second machine and their sequence is specified. In the same way, the jobs are allocated to each machine and their sequences are determined. For example, assume the following instance having 6 jobs and 3 machines, a solution can be represented as Fig. 2.

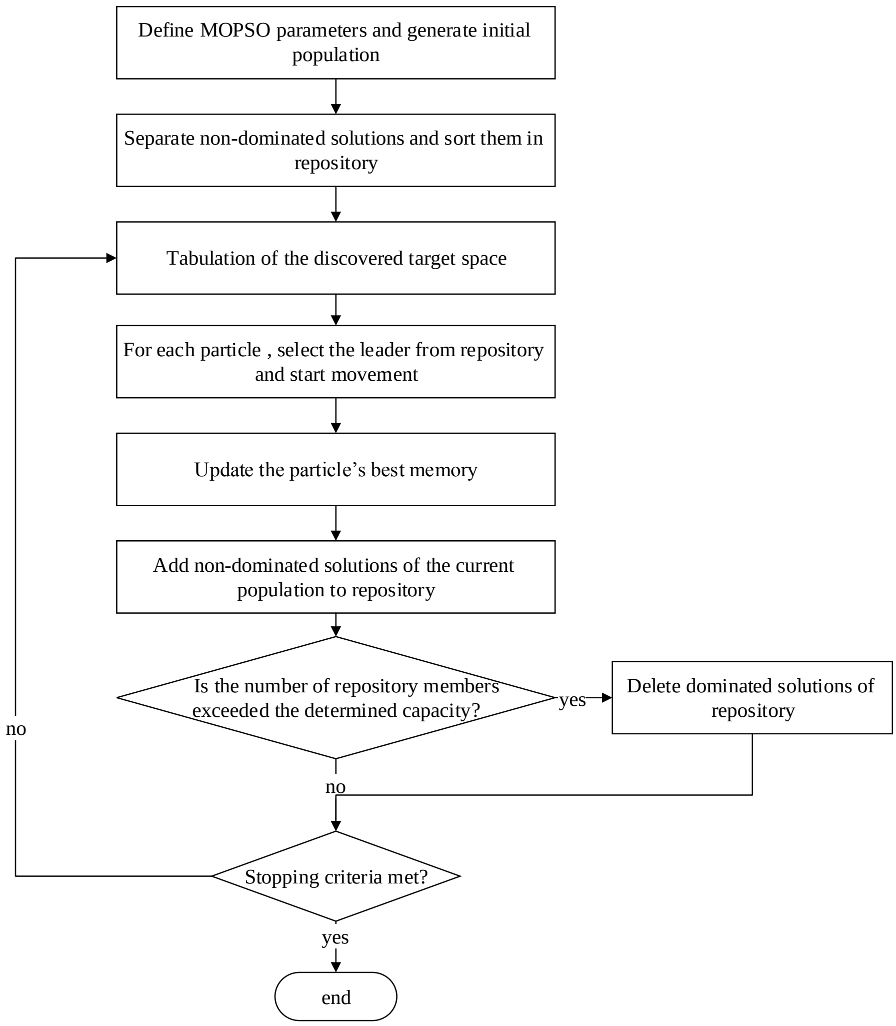

Fig. 1. Flowchart of the MOPSO algorithm

2	5	8	7	3	1	4	6

Fig. 2. Sequence pattern for an instance with 6 jobs and 3 machines

Based on the above definition, jobs 2 and 5 are allocated to machine 1, and jobs 3, 1, 4, and 6 are allocated to machine 3. In this example, no job is assigned to machine 2.

Sorting the solutions

In the multi-objective optimization, the domination of the Pareto-optimal solution is the main criterion for recognizing the quality of different solutions. In this case, it is said that the solution x_1 with objective values (C_{max1} , TEC_1) dominates the solution x_2 with objective values (C_{max2} , TEC_2), if one of the following two conditions is met:

$$\begin{aligned} &C_{max1} < C_{max2} \text{ and } TEC_1 \leq TEC_2 \\ &C_{max1} \leq C_{max2} \text{ and } TEC_1 < TEC_2 \end{aligned} \tag{24}$$

This relationship is denoted by $x_1 \prec x_2$. If one of the solutions does not dominate the other, two solutions are considered in a Pareto rank. All dominant solutions are stored in the repository.

Multi-objective simulated annealing algorithm

The multi-objective simulated annealing algorithm (MOSA) [28] is an extended version of the single-objective simulated annealing while uses the non-dominated concept. Compared to other evolutionary optimization algorithms, MOSA does not require much memory to store population information. This algorithm is utilized in scheduling studies such as [29,30,31,32,33].

The initial sequence pattern of jobs is generated in the same way as described in the MOPSO algorithm.

Neighborhood generation

In the MOSA algorithm, the new solution is randomly generated adjacent to the previous solution using a suitable neighborhood structure. Therefore, choosing the right neighborhood structure is very important to maintain the problem convergence to achieve the optimal solution. In this research, three mechanisms of neighborhood generation are used. The Swap operator first randomly chooses two jobs from the list of jobs sequence and swaps them. In the Reversion operator, two cells are selected and all the cells between them are written in reverse. In the third operator, named insertion, two cells are selected and the first selected cell is placed after the second chosen cell.

Initial temperature (T_0)

The Initial temperature should be high enough to allow movement to the neighborhood and search space. By setting too high values for initial temperature, the algorithm moves to each neighborhood and its efficiency becomes the same as a local search algorithm. Oppositely, if a too low value is set for the initial temperature, it leads to early convergence and possibly falling into the local optimal trap. One way to determine the T_0 parameter, which is used in this research, is to generate a number of consecutive solutions using the selected neighborhood structure and consider the maximum difference of the objective function between two consecutive solutions as the initial temperature.

Cooling ratio (α)

After passing the specified iterations at each temperature, the temperature should be reduced. As the algorithm progresses, the likelihood of accepting a worse solution will decrease as the algorithm proceeds to find the optimal solution. The relationship is as follows:

$$T_{n+1} = \alpha . T_n \tag{25}$$

Appropriate values for α are determined according to the problem conditions. Usually, one of these two criteria is used as the stopping condition: reaching 1) the freezing temperature, or 2) the maximum number of iterations without gaining any improvement. In this study, the second criterion is used.

Parameter setting and results

The ε-constraint procedure is implemented in GAMS while MOPSO and MOSA algorithms are implemented in MATLAB on a personal computer with a 1.80 GHz Ci5 processor and 6 GB RAM. This section presents the parameter setting procedure and compares the performance of our solution approaches.

Parameter Tuning

MOPSO and MOSA algorithms have several factors and parameters which affect the quality of results and efficiency of algorithms. In other words, setting these parameters can significantly improve the performance of algorithms. The Taguchi method is a powerful and effective technique for parameter setting. This method determines suitable levels of the design factors and the optimal parameter combination via reducing the number of experiments and also by means of the signal-to-noise ratio.

For a minimization problem, Taguchi proposed the following equation called smaller is better.

$$\text{S/N ratio} = -10 \log \textstyle\sum_{i=1}^{k} (y_i{}^2/k) \tag{26}$$

Where y_i is the response value obtained from the i^{th} instance and k is the number of instances. To convert the criteria into an answer, the Simple Additive Weighting (SAW) technique is used. The SAW technique can be described in a few steps:

1- Create a decision matrix and identify the negative or positive nature of each criterion.

2- Calculate the normalized decision matrix using Eq. 27.

$$\begin{aligned} &\text{If } r_{ij}^{+}: \quad n_{ij} = \frac{r_{ij}}{r_j^{max}} \\ &\text{If } r_{ij}^{-}: \quad n_{ij} = \frac{r_j^{min}}{r_{ij}} \end{aligned} \tag{27}$$

If the criterion is positive, each of the numbers in that column is divided by the largest number while if the criterion is negative, the minimum of that column is divided by each number.

3-Calculation of simple Additive Weighting by Eq. 28.

$$SAW_i = \textstyle\sum_j w_j . n_{ij} \qquad\qquad \sum_j w_j = 1 \tag{28}$$

In this study, the entropy method [34] is used to calculate the relative weights, w_j.

Comparison metrics of multi-objective algorithms

- **Mean Ideal Distance (MID)**

MID measures the average distance of Pareto-optimal solutions from an ideal point. So, it is clear that the lower the metric is, the better the performance of the algorithm.

Because we try to find fronts closer to the coordinate center (0,0), the MID calculates the distance between the fronts from the best population value. In this study, the ideal point is considered as the minimum value of objective functions amongst all algorithms. Eq. 29 is used to calculate the MID metric:

$$MID = \sum_{i=1}^{n} \frac{\sqrt{\left(\frac{f_{1i} - f_1^{best}}{f_1^{max} - f_1^{min}}\right)^2 + \left(\frac{f_{2i} - f_2^{best}}{f_2^{max} - f_2^{min}}\right)^2}}{n} \tag{29}$$

In Eq. 29, n indicates the number of Pareto-optimal points, f_i^{min} and f_i^{max} are maximum and minimum values of the objective values amongst all the comparable algorithms, respectively. Also (f_2^{best}, f_1^{best}) is the ideal point.

- **Spacing Metric (SM)**

The diversity of the Pareto-optimal front in the solution area can be quantified with this metric. This metric is calculated based on Eq. 30.

$$SM = \frac{\sum_{i=1}^{n-1} \left| \bar{d} - d_i \right|}{(n-1)\bar{d}} \tag{30}$$

Where d_i indicates the Euclidean distance of two successive solutions of non-dominated solutions, and $\bar{d}$ is the average of all d_i values. The lower value of SM represents the better performance of an algorithm.

- **Diversification Metric (DM)**

DM shows the amplitude of Pareto-optimal solutions of an algorithm determined by Eq. 31. The higher value of DM indicates that the algorithm performs better.

$$DM = \sqrt{\left(\frac{max\, f_{1i} - min\, f_{1i}}{f_{1.total}^{max} - f_{1.total}^{min}}\right)^2 + \left(\frac{max\, f_{2i} - min\, f_{2i}}{f_{2.total}^{max} - f_{2.total}^{min}}\right)^2} \tag{31}$$

- **CPU Time**

One of the important metrics in large-sized instances is the required CPU Time for obtaining the solution.

Parameter setting for the proposed MOPSO algorithm

The proposed MOPSO algorithm has 4 parameters that each has 3 levels according to previous researches as shown in Table 2. The sample problems used in this section are randomly generated in small and large sizes. Table 1 shows the experimental design where U represents the discrete uniform distribution. For each problem set, 5 instances are evaluated, and averages answers are considered. Table 2 shows the suggested values for MOPSO parameters.

Table 1. Experimental design

Factors	**Levels for small-scale**	**Levels for large -scale**
Number of jobs (*N*)	5, 25, 50	60, 100, 150
Number of machines (*M*)	5, 8	15, 30
processing time (t_{ij}) (hour)	U~[1, 3], U~[1,5]	U~[1, 3], U~[1, 5]
energy consumption rate (p_{ij}) (kilowatt)	U~[1, 5]	U~[1, 5]
sequence-dependent setup time (s_{hij}) (hour)	U~[1, 3]	U~[1, 3]
energy price (c_k) ($/kilowatt-hour)	U~[1, 5]	U~[1, 5]
duration of period (T_k) (hour)	{50, 100}	{150, 300}
Number of time periods (*K*)	{3, 4, 5}	{3, 4, 5}

Table 2. Parameters and candidate levels for the MOPSO algorithm

Parameter	**Small-scale levels (Instances 1-15)**	**Large -scale levels (Instances 16-25)**
Particle size	{50, 75, 100}	{100, 120, 150}
Inertia weight (W)	{0.6, 0.75, 0.9}	{0.7, 0.8, 0.9}
Personal-learning coefficient (c_1)	{1, 1.5, 2}	{1, 1.5, 2}
Global-learning coefficient (c_2)	{1, 1.5, 2}	{1, 1.5, 2}

According to the Taguchi Orthogonal Array, the L27 design can be used by considering 4 parameters and 3 levels. Experiments are performed in different cases, and metrics are calculated. To analyze different combinations in the Taguchi method, only one numerical value must be entered to determine the best combination, so the SAW method is used. The results are shown in Table 3.

Table 3. SAW values for designed experiments for the MOPSO algorithm

Exp. No	Small-scale instances				large-scale instances			
	MID	**DM**	**SM**	**SAW**	**MID**	**DM**	**SM**	**SAW**
1	0.27388	1.34863	0.5522	0.57610	0.53906	0.81094	0.45064	0.52700
2	0.33348	1.41574	0.48504	0.53610	0.43745	0.58461	0.20986	0.43065
3	0.30859	1.23353	0.51294	0.52660	0.35448	0.466	0.17456	0.38904
4	0.27764	1.17547	0.21384	0.62734	0.46811	0.56656	0.76546	0.39651
5	0.25634	1.35929	0.60634	0.59504	0.42072	0.48333	0.07767	0.42192
6	0.28280	1.11772	0.30159	0.57401	0.46444	0.38438	0.05285	0.39213
7	0.23625	1.02463	0.41777	0.59481	0.35588	0.70601	0.32777	0.50707
8	0.23238	1.15561	0.46606	0.61373	0.37360	0.91768	0.50740	0.61380
9	0.32769	2.99631	0.37566	0.79680	0.56367	0.35300	0.37299	0.27286
10	0.38813	1.23469	0.53969	0.46571	0.38316	0.43483	0.45887	0.34497
11	0.28708	1.35814	0.51319	0.56673	0.42219	0.29514	0.16729	0.27974
12	0.29900	1.25357	0.48273	0.54223	0.37123	0.78210	0.41358	0.54169
13	0.31011	1.06857	0.38180	0.51937	0.43040	0.73062	0.12743	0.52968
14	0.24685	0.94044	0.63242	0.54295	0.25464	0.22395	0.22514	0.28690
15	0.26556	0.94358	0.47945	0.53275	0.55618	0.34508	0.88024	0.26089
16	0.23097	1.42896	0.67435	0.63844	0.27211	0.36141	0.02131	0.58566
17	0.24985	1.38972	0.66577	0.60433	0.43959	0.83954	0.34361	0.56143
18	0.24382	1.21921	0.47978	0.60386	0.54707	0.56000	0.92815	0.38074
19	0.25425	1.42626	0.65379	0.60457	0.37627	0.35300	0.5588	0.29918
20	0.25452	1.17944	0.56698	0.57357	0.39035	0.36141	0.23755	0.31365
21	0.26162	1.14895	0.50194	0.56612	0.42824	0.64349	0.73051	0.44650
22	0.29132	1.11139	0.42385	0.53669	0.40832	0.83816	0.42838	0.56368
23	0.21225	1.08591	0.62948	0.62313	0.46418	0.86082	0.72123	0.56067
24	0.29283	1.20283	0.39992	0.55299	0.52264	0.57315	0.70837	0.39287
25	0.22792	1.08190	0.59216	0.59695	0.42476	0.64633	0.27099	0.46140
26	0.31730	1.20092	0.30777	0.55093	0.43836	0.68890	0.69651	0.47013
27	0.27695	1.17123	0.59754	0.54181	0.42680	1.09297	0.16449	0.72157
Average	0.27553	1.26935	0.49829	0.57791	0.42718	0.58901	0.40820	0.44267

The Taguchi design is performed as shown in Table 3. Figs. 3 and 4 show the main impacts of S/N ratio values for different levels of each parameter based on different problem sizes. According to the S/N plot of each parameter, the maximum value is selected.

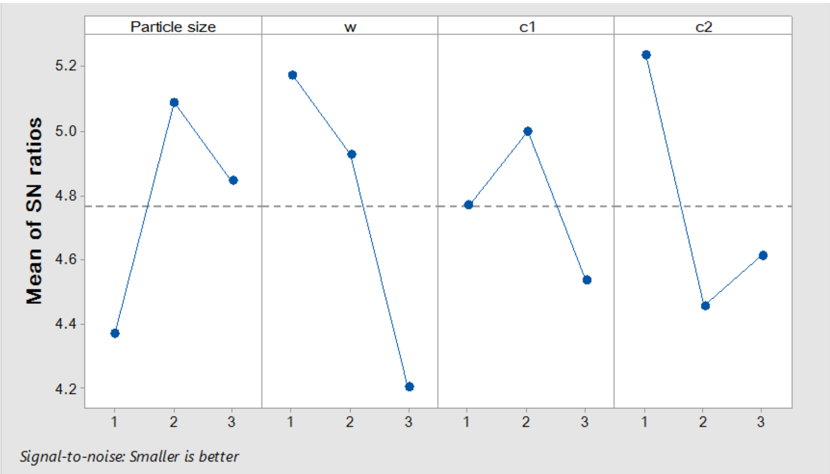

Fig. 3. Impact of S/N ratios in MOPSO on small-sized instances

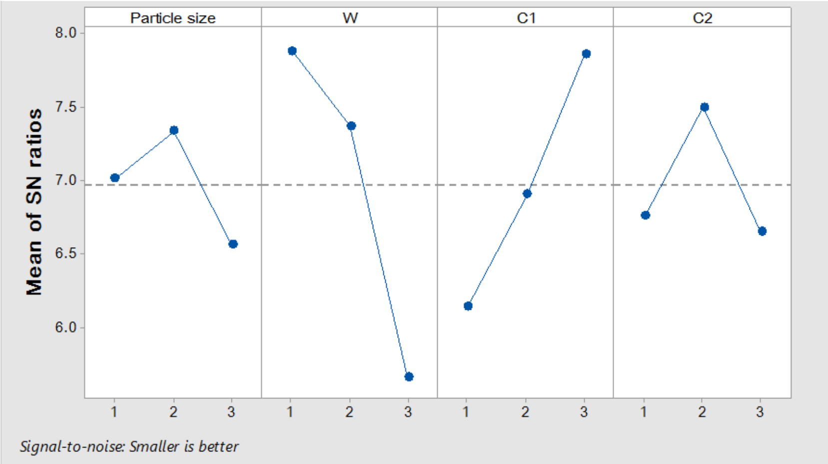

Fig. 4. Impact of S/N ratios in MOPSO on large -sized instances

According to Figs. 3 and 4, suggested parameter values for the MOPSO algorithm for small and large-sized instances are selected and shown in Table 4. Also, by testing the initial data, the number of iterations is set to 100 for small-sized and 200 for large instances.

Table 4. Optimal values for MOPSO algorithm parameters

Size	**Parameters**				
	Iteration	**Particle size**	**W**	c_1	c_2
Small	100	75	0.6	1.5	1
Large	200	120	0.7	2	1.5

Parameter setting for the proposed MOSA algorithm

As shown in Table 5, two parameters are considered as controllable factors in 3 levels for the MOSA algorithm. Similarly, the L9 design is used according to the Taguchi Orthogonal Array as shown in Table 6. The rest of the steps are the same as the previous section.

Table 5. Parameters and candidate levels for the MOSA algorithm

Parameter	**Small-scale levels (Instances 1-15)**	**Large-scale levels (Instances 16-25)**
Initial temperature (T_0)	{100, 200, 300}	{100, 200, 300}
cooling ratio (α)	{0.4, 0.6, 0.8}	{0.4, 0.6, 0.8}

Table 6. SAW values for designed experiments for the MOSA algorithm

Exp. No	**Small-sized instances**				**Large-sized instances**			
	MID	**DM**	**SM**	**SAW**	**MID**	**DM**	**SM**	**SAW**
1	0.23008	1.08766	0.24551	0.98752	0.47898	0.53036	0.44562	0.25950
2	0.21787	1.21987	0.31374	0.80641	0.51863	0.65428	0.08487	0.6407
3	0.24253	1.20272	0.54018	0.51271	0.53641	0.28133	0.66666	0.16471
4	0.21018	1.10763	0.36847	0.69635	0.51532	0.66192	0.05248	0.91222
5	0.23072	0.94221	0.31307	0.78605	0.32609	0.30655	0.26334	0.28091
6	0.22675	0.97413	0.51154	0.51994	0.33232	0.91872	0.33222	0.40451
7	0.23297	1.19653	0.46759	0.57553	0.46468	0.48429	0.46998	0.24493
8	0.23218	1.21987	0.31399	0.80430	0.46671	0.55545	0.56674	0.24917
9	0.20625	1.19265	0.59478	0.47907	0.37724	0.64547	0.43654	0.30177
Average	0.22550	1.12703	0.40765	0.68532	0.44626	0.55981	0.36871	0.38426

According to Table 5, the impact of S/N values for each size are as illustrated in Figs. 5 and 6.

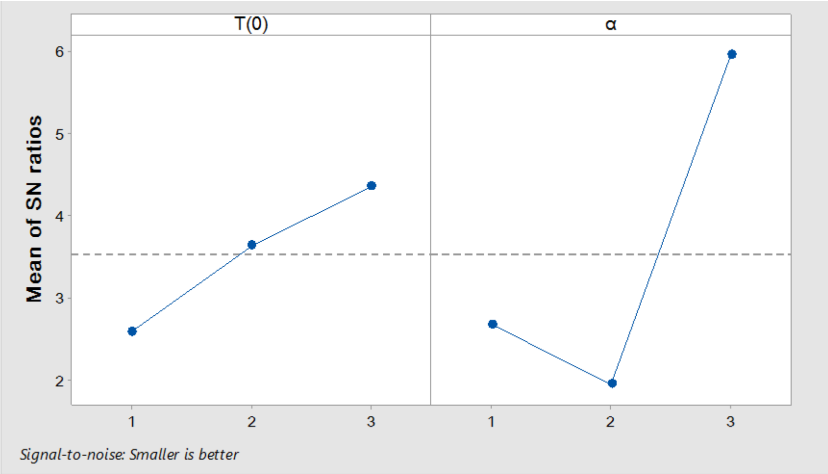

Fig. 5. Impact of S/N ratios in MOSA on small-sized instances

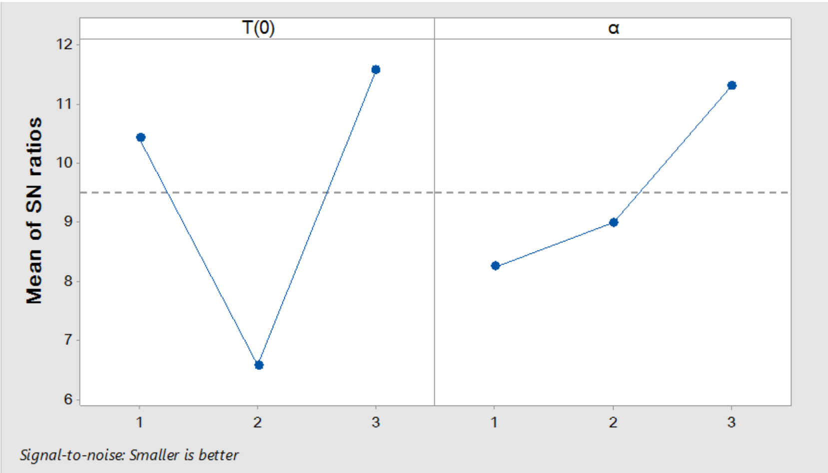

Fig. 6. Impact of S/N ratios in MOSA on large-sized instances

The suggested values of the MOSA algorithm parameters are summarized in Table 7. Also, by testing the initial data, the number of iterations is equal to 150 for small-sized instances and 200 for large-sized instances, and the number of internal loop iterations is 5 and 15 for small and large-sized instances, respectively.

Table 7. Optimal values for MOPSO algorithm parameters

Size	Parameters			
	Iteration	**Internal loop iterations**	T_0	α
Small	150	5	300	0.8
Large	200	15	300	0.8

Computational results

In this section, first, a real-world case study is investigated for validating the proposed MILP model and the proposed algorithms. Afterward, several random instances are generated for evaluating the performance of the proposed approaches.

Case study

In this section, a real-world case in Anhui Province of China is used for validating the proposed approaches. This case study is retrieved from Zhang et al. [35] which considers a batch of hollow shafts. In this system, each job undergoes two processes: the turning process, and then the milling process. At first, the turning process is performed by one of the identical parallel machines, and then, one of the unrelated milling machines mills the keyway of the job. There are 20 jobs that should be processed. There are two lathes of turning machines with different spindle speeds ($v_s \in \{0.75, 1.1\}$), and two milling machines with a fixed spindle speed.

The TOU pattern is implemented in that province as shown in Table 8. A 24-hour time horizon is considered where time zero refers to 8 am. In this study, we consider stage 2, (unrelated milling machines). Table 9 shows the processing time for these jobs on each machine and the energy consumption rate of jobs on machines. Since Zhang et al. [35] ignored the setup times, we considered them random with discrete uniform distribution, $U \sim [0, 0.3]$.

Table 8. The TOU tariffs used for experiments [35]

Period type	Time periods	Electricity price (CNY/kwh)
On-peak	9:00-12:00, 17:00-22:00	1.1236
Mid-peak	8:00-9:00, 12:00-17:00, 22:00-23:00	0.7493
Off-peak	23:00-8:00	0.4703

Table 9. The data of real-world case study [35]

Job (i)	$t_{i1}(h)$	$t_{i2}(h)$	$p_{i1}(kw)$	$p_{i2}(kw)$	Job (i)	$t_{i1}(h)$	$t_{i2}(h)$	$p_{i1}(kw)$	$p_{i2}(kw)$
1	1	1	3	17	11	1.9	0.7	4	16
2	2.1	0.7	6	16	12	0.9	1.1	3	14
3	1.9	0.8	3	14	13	1.7	1	5	19
4	0.6	0.8	7	22	14	2.5	0.5	6	14
5	1	1.1	4	19	15	1.6	0.9	7	7
6	1.4	0.6	5	14	16	1.3	1.3	5	21
7	1.5	0.9	8	21	17	0.9	1.5	6	26
8	1.1	1.3	6	9	18	1.6	1.2	7	21
9	1.3	0.7	6	16	19	2.5	0.6	4	12
10	2	0.8	7	22	20	1	0.9	5	10

- **The trade-off between TEC and C_{max}**

The trade-off between TEC and C_{max} is shown in Fig. 7. According to the Pareto front obtained from the ε-constraint method, the cost of energy consumption decreases with increasing makespan. During on-peak and mid-peak periods, the cost reduction is relatively large and it is because of the assignment of jobs to machines with lower consumption rates, which processing is done slower and needs lower energy consumption. In the off-peak period, the cost reduction is relatively small and the reason is the low cost of energy consumption in this period. According to the trade-off between our objective functions, manufacturers can determine the appropriate assignment of jobs to machines and their sequences under TOU tariffs, depending on the desired cost. Also, as shown in Fig. 7, the length of the Pareto front is longer in the off-peak period and can encourage manufacturers to move their electricity usage from on-peak to off-peak times. Applying energy-efficient scheduling under TOU tariffs reduces energy consumption and energy consumption costs which can lead to a reduction in CO_2 emission too.

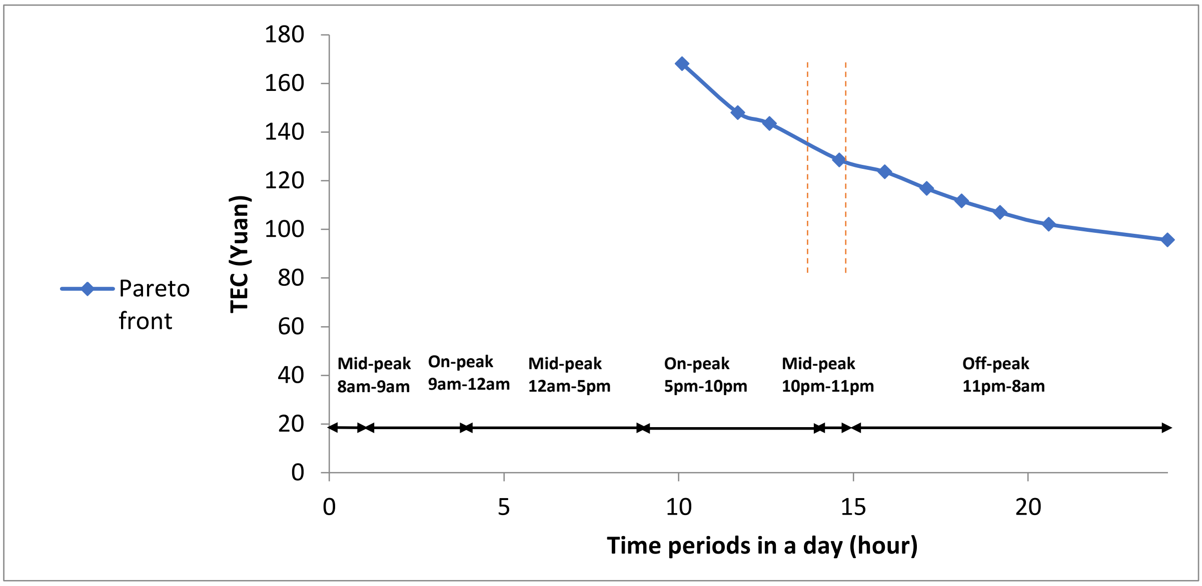

Fig. 7. The trade-off between TEC and C_{max}

- **Impact of considering sequence-dependent setup times**

The sequence of jobs based on the lowest makespan is shown in Fig. 8. Now, if we ignore sequence-dependent setup times and solve the problem, it is observed that the sequence of jobs is different from the previous case. Applying the sequence-dependent setup times for the obtained sequence in Fig. 9 will result in the sequence shown in Fig. 10 and objective functions (C_{max}=10.7, TEC=172.0124) are larger than the case in Fig. 8 (C_{max}=10.1, TEC=168.0429). So, by ignoring sequence-dependent setup times, makespan increases by 5% and energy consumption costs increase by 2% and can result in a non-optimal solution.

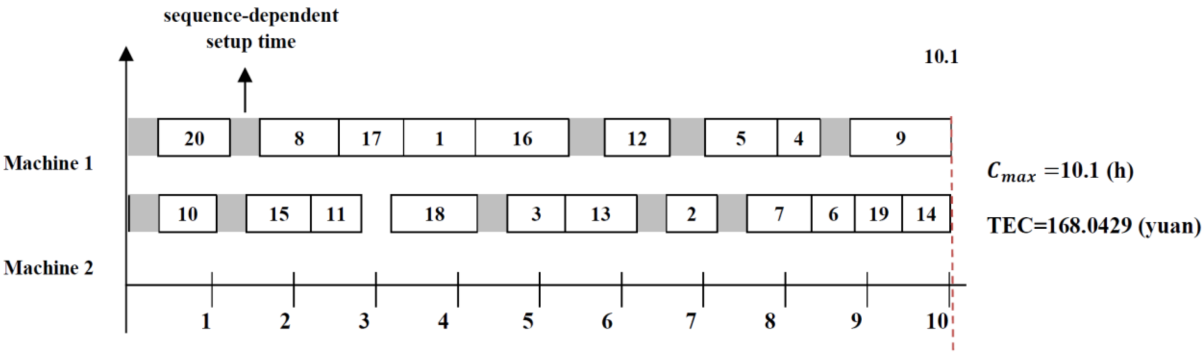

Fig. 8. The sequence of jobs with the lowest makespan

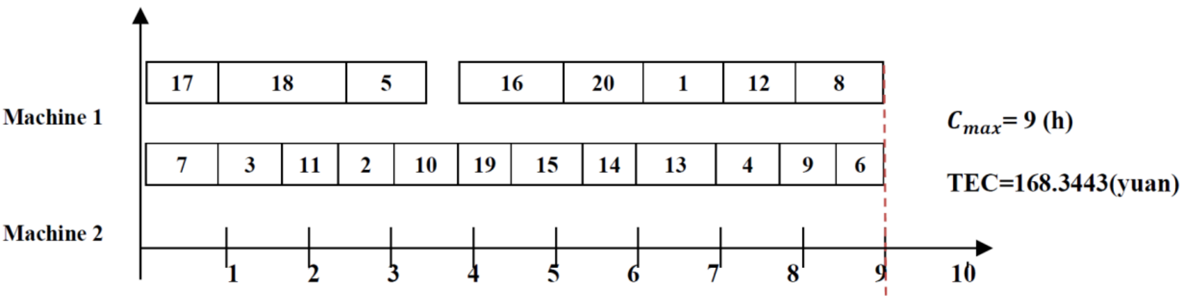

Fig. 9. The sequence of jobs without consideration of setup times

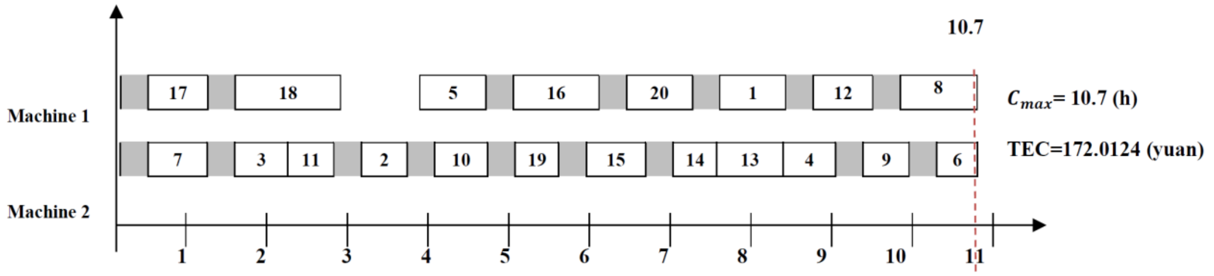

Fig. 10. The sequence of jobs in Fig. 9. with consideration of setup times

- **Comparison of the proposed algorithms**

Regarding results shown in Fig. 11, there is a logical relationship between the cost objective function and the completion time, which indicates the logical treatment of the model. In other words, the longer the completion time, the lower the energy consumption cost. According to the results, it can be concluded that the MOPSO and MOSA algorithms operate very close to the exact ε-constraint method, and there is an insignificant gap between the MOPSO and the MOSA algorithms compared to the ε-constraint method, ranging from 0 to 4.54% for MOSA, and 0 to 6.8% for MOPSO algorithm.

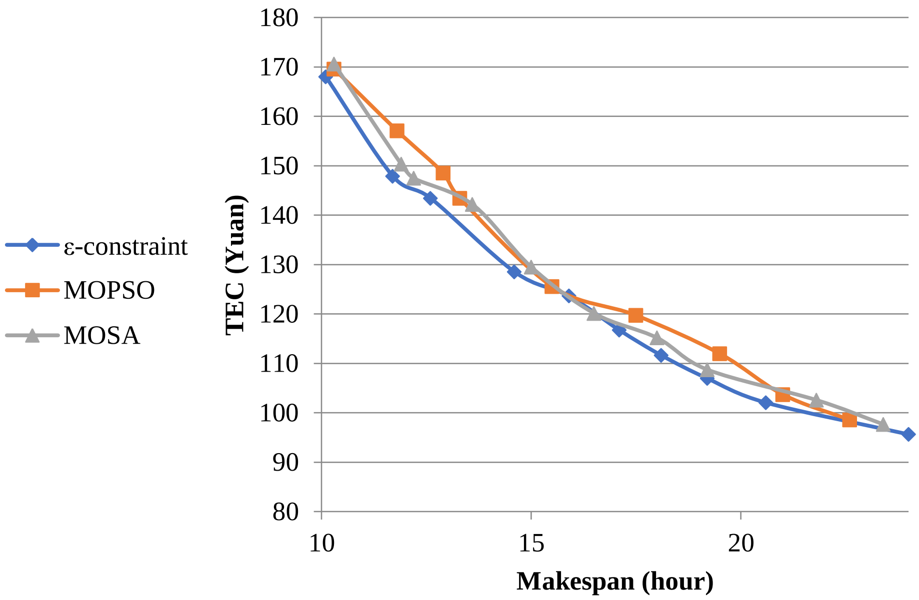

Fig. 11. The Pareto front for the real-world case study

Specifications of random test instances

To compare the results of MOPSO, MOSA, and the ε-constraint method, three random problem categories in different sizes (small, medium, and large) were generated and solved and the obtained results have been compared.

In Table 10, the number of machines and jobs ($m * n$), the process duration of each job on each machine (t_{ij}), the energy consumption rate of each job on each machine (p_{ij}), duration of the studied periods (T_k), energy price for each period (c_k) and the setup time of every job on each machine according to its previous job (s_{ihj}) are shown, respectively.

The meaning of U in Table 10 is random data with discrete uniform distribution. Also, sequence-dependent setup times, for instances 1-3 are generated from a discrete uniform distribution to analyze their effect, and for the rest of the instances are considered random and fixed.

Validation

For validating the proposed algorithms, 25 random instances are solved using the ε-constraint method, MOPSO, and MOSA. The results are presented in Table 11 and MID, SM, DM, and CPU time are reported.

Table 10. Specifications of random test instances

Size	Instance No.	m*n	t_{ij} (hour)	p_{ij} (kilowatt)	T_k (hour)	c_k ($/kilowatt-hour)	s_{ihj} (hour)
Small	1	2*5	U~[1, 5]	U~ [1, 3]	{3, 4, 5}	U~ [1, 5]	U~ [0,1]
	2	2*5	U~ [1, 5]	U~ [1, 3]	{3, 4, 5}	U~ [1, 5]	U~ [0,2]
	3	2*5	U~ [1, 5]	U~ [1, 3]	{3, 4, 5}	U~ [1, 5]	U~ [0,3]
	4	3*5	U~ [1, 5]	U~ [1, 3]	{4, 5, 6}	U~ [1, 5]	2
	5	4*5	U~ [1, 3]	U~ [1, 3]	{3, 4, 5}	U~ [1, 3]	1
	6	4*5	U~ [1, 5]	U~ [1, 3]	{3, 4, 5}	U~ [1, 5]	2
	7	4*8	U~ [1, 3]	U~ [1, 3]	{4, 5, 6}	U~ [1, 3]	1
	8	4*8	U~ [1, 3]	U~ [1, 3]	{4, 5, 6}	U~ [1, 5]	1
	9	4*10	U~ [1, 3]	U~ [1, 5]	{5, 6, 7, 8}	U~ [1, 3]	1
	10	4*10	U~ [1, 5]	U~ [1, 5]	{5, 6, 7, 8}	U~ [1, 5]	1
	11	5*10	U~ [1, 3]	U~ [1, 3]	{5, 6, 7, 8}	U~ [1, 3]	2
	12	5*10	U~ [1, 5]	U~ [1, 5]	{5, 6, 7, 8}	U~ [1, 5]	2
Medium	13	5*12	U~ [1, 5]	U~ [1, 3]	{5, 6, 7, 8}	U~ [1, 5]	2
	14	5*12	U~ [1, 5]	U~ [1, 5]	{6, 6, 8, 8}	U~ [1, 5]	2
	15	5*15	U~ [1, 5]	U~ [1, 7]	{6, 6, 8, 8}	U~ [1, 7]	2
	16	5*30	U~ [1, 3]	U~ [1, 5]	{6, 6, 8, 8}	U~ [1, 3]	1
	17	5*40	U~ [1, 3]	U~ [1, 3]	{10, 12, 15, 20}	U~ [1, 3]	1
	18	5*45	U~ [1, 5]	U~ [1, 5]	{15, 15, 20, 20}	U~ [1, 5]	1
Large	19	8*60	U~ [1,3]	U~ [1,7]	{20,20,20,25,25}	U~ [1,3]	2
	20	8*80	U~ [1, 3]	U~ [1, 7]	{20, 20, 20, 25, 25}	U~ [1,7]	2
	21	10*100	U~ [1, 5]	U~ [1, 7]	{50, 60, 70, 80, 90}	U~ [1,7]	1
	22	10*150	U~ [1, 5]	U~ [1, 7]	{50, 60, 70, 80, 90}	U~ [1,7]	1
	23	20*180	U~ [1, 5]	U~ [1, 7]	{50, 60, 70, 80, 90}	U~ [1,7]	1
	24	25*200	U~ [1, 3]	U~ [1, 7]	{50, 60, 70, 80, 90}	U~ [1,5]	1
	25	35*250	U~ [1, 5]	U~ [1, 7]	{90, 90, 90, 90, 90}	U~ [1,7]	1

As shown in Table 11, according to the values obtained by the exact method for instances 1-3 as the sequence-dependent setup times increase, MID metric increases and gets worse. In terms of DM metric, the value reported for the ε-constraint method decreases with increasing setup times. In terms of SM metric, the value of this metric gets worse with increasing setup times. CPU time decreases with increasing setup times.

According to the MID metric in Table 11, MOSA performs better than MOPSO in most of the instances and the distance of Pareto solutions from an ideal point obtained by MOSA are less than MOPSO. MOPSO algorithm has better performance than MOSA algorithm in DM metric. This means that the MOPSO algorithm can generate Pareto solutions with a wider range of possible solutions. In SM metric, the MOSA algorithm shows better performance. In other words, the distance between two consecutive Pareto solutions obtained by MOSA is less than the MOPSO algorithm.

Moreover, it can be seen that the CPU time of the ε-constraint method is increased significantly if the size of the problem is increased until in sample 16 the ε-constraint method is not able to solve the problem within the considered time limit. However, meta-heuristic algorithms can solve the problem in a much shorter time. In addition, the MOSA algorithm requires less time to find the Pareto fronts, and this can be considered as an advantage of this algorithm compared to the MOPSO. For a more detailed comparison of these algorithms, the SAW criterion is depicted in Fig. 12 for instances 1 to 15.

Table 11. Comparison the results of ε –constraint, MOPSO and MOSA

Instance No.	MID			SM			DM			CPU time (s)		
	ε-const.	MOPSO	MOSA	ε-const.	MOPSO	MOSA	ε-const.	MOPSO	MOSA	ε-constraint	MOPSO	MOSA
1	0.44	0.49	0.46	0.3	0.5	0.41	1.41	1.13	1.34	0.3	29.51	23.12
2	0.49	0.54	0.52	0.33	0.36	0.28	1.32	1.17	0.87	0.27	24.3	21.46
3	0.52	0.59	0.56	0.36	0.41	0.38	1.17	1.36	0.77	0.2	23.26	17.87
4	0.64	0.82	0.66	0.21	0.27	0.22	1.03	0.86	0.72	0.16	53.32	35.49
5	0.55	0.67	0.61	0.32	0.42	0.38	0.99	0.75	0.75	0.2	44.92	31.91
6	0.66	0.75	0.72	0.26	0.45	0.36	0.89	0.69	0.54	1.53	43.53	40.63
7	0.54	0.71	0.67	0.41	0.66	0.56	0.82	0.87	0.79	10.66	137.42	48.3
8	0.69	0.88	0.81	0.37	0.54	0.48	1	0.9	0.82	203.55	138.96	89.8
9	0.54	0.69	0.79	0.33	0.47	0.4	0.8	0.82	0.76	209.48	122.4	93.65
10	0.43	0.65	0.59	0.29	0.38	0.32	0.98	0.79	0.68	178	179.52	119.7
11	0.48	0.67	0.6	0.31	0.4	0.33	0.5	0.76	0.58	168.97	105.51	121.82
12	0.62	0.74	0.68	0.29	0.46	0.35	0.79	0.89	0.8	1896.19	104.82	125.69
13	0.68	0.74	0.7	0.27	0.58	0.37	1.4	0.86	0.64	2000.19	150.3	130.63
14	0.59	0.72	0.69	0.26	0.44	0.35	1.01	0.83	0.72	2050.28	155.79	145.24
15	0.61	0.73	0.7	0.34	0.59	0.44	0.98	0.85	0.69	3600	187.5	161.92
16	-	0.74	0.69	-	0.53	0.32	-	0.82	0.74	-	360.37	346.6
17	-	0.61	0.54	-	0.42	0.39	-	0.79	0.69	-	379.85	352.74
18	-	0.59	0.63	-	0.4	0.32	-	0.82	0.75	-	411.4	403.98
19	-	0.65	0.67	-	0.37	0.27	-	0.67	0.52	-	520.2	497.07
20	-	0.66	0.71	-	0.45	0.33	-	0.83	0.72	-	488.3	501.36
21	-	0.69	0.7	-	0.35	0.24	-	0.75	0.65	-	1778.57	706.53
22	-	0.65	0.61	-	0.42	0.34	-	0.72	0.61	-	2063.18	1021.87
23	-	0.62	0.59	-	0.37	0.32	-	0.8	0.73	-	2472.31	1789.23
24	-	0.61	0.62	-	0.4	0.32	-	0.89	0.76	-	3156.45	1985.64
25	-	0.6	0.58	-	0.49	0.36	-	0.79	0.56	-	3459.02	2607.8
Average	0.58	0.69	0.66	0.30	0.44	0.35	0.93	0.80	0.69	859.93	750.62	516.25

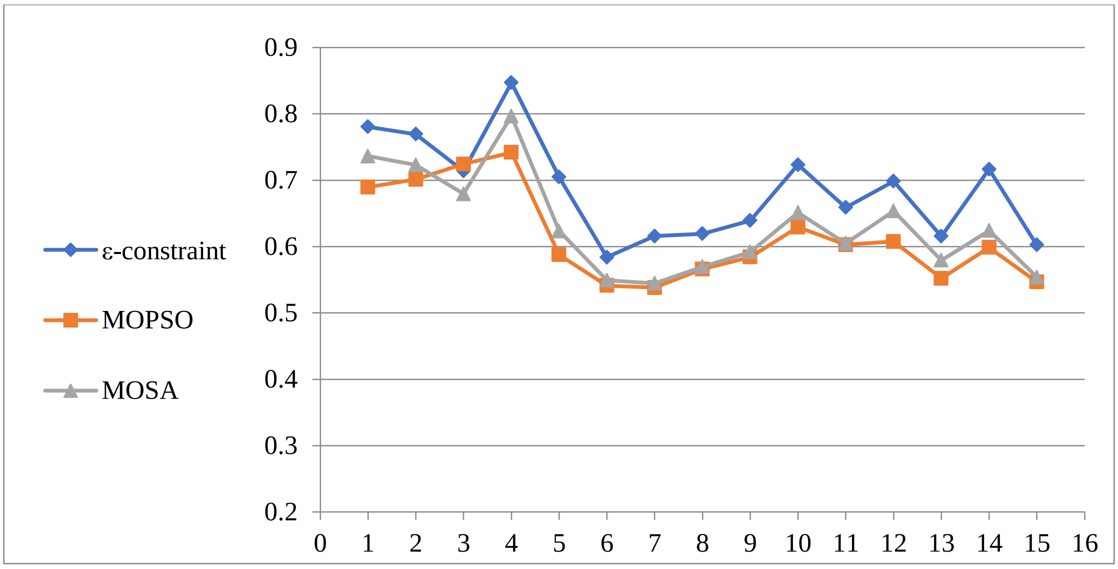

Fig. 12. The SAW criterion for instances 1 to 15

As shown in Fig. 12, it is observed that the metaheuristic algorithms used in this study are very close to the exact solution and therefore have appropriate efficiency in finding near-optimal solutions. So, they can be used for solving large-size problems where the ε-constraint method is incapable.

There is a gap between results of the ε-constraint method and those of MOPSO algorithm ranging from 0 to 12%. The MOSA algorithm shows a 0 to 9% gap. These gaps show that our proposed algorithms can yield reliable Pareto solutions.

Conclusion

In this research, an unrelated parallel machine scheduling problem under the TOU tariff in which the electricity price varies throughout a day is considered. Sequence-dependent setup time is also considered to make the problem assumptions more realistic. A bi-objective MIP model is presented for minimizing energy consumption costs and also the makespan. Also, a real-world case study is investigated for validating the proposed solution approaches. Because the ε-constraint method was incapable to solve large-sized problems in a 3600 seconds limit, MOPSO and MOSA algorithms are used to find near-optimal solutions for large-sized instances. The parameters of each algorithm are adjusted using the Taguchi method. To assess the performance of our algorithms, 25 randomly generated instances with different sizes are examined. CPU time, Mean Ideal Distance, Spacing and Diversification metrics of these algorithms are compared with the ε-constraint method. The results indicate that the MOSA performs better than the MOPSO.

In terms of practical application, the proposed model and methods provide production managers with decision-making tools to make logical trade-offs between energy consumption cost and makespan under TOU policy in production planning. An appropriate response to the TOU policy can significantly reduce energy consumption costs and help to reduce CO_2 emissions.

Since sequence-dependent setup times are unavoidable in production environments, they have been considered in the proposed mathematical modeling. By considering sequence-dependent setup times, the resulted sequence for jobs is so different from the situation that they have been ignored. This difference in sequences can lead to inaccurate and not optimal answers. So, considering sequence-dependent setup times in the production plan can help production managers get more reliable answers and prevent losses.

Considering the due date of jobs and process preemption are two possible fields for future research. Adding other objective functions such as minimizing the amount of CO_2 emitted, can be considered too. It is also possible to use other multi-objective meta-heuristic methods for the research problem and compare their performance with our proposed algorithms. Finally, considering the possibility of breakdown for the production line after processing or during processing seems interesting.

References

[1] EIA. Energy use in industry, https://www.eia.gov/energyexplained/use-of-energy/industry.php ; 2020 [Accessed 28 July 2020].
[2] Li, L., Li, C., Tang, Y., & Li, L. An integrated approach of process planning and cutting parameter optimization for energy-aware CNC machining. Journal of Cleaner Production. 2017;162: 458-73.
[3] Zeng, Y., Che, A., & Wu, X. Bi-objective scheduling on uniform parallel machines considering electricity cost. Engineering Optimization. 2018;50(1): 19-36.
[4] Wu, X., & Che, A. A memetic differential evolution algorithm for energy-efficient parallel machine scheduling. Omega. 2019; 82: 155-65.
[5] Tavakkoli-Moghaddam, R., Taheri, F., Bazzazi, M., Izadi, M., & Sassani, F. Design of a genetic algorithm for bi-objective unrelated parallel machines scheduling with sequence-dependent setup times and precedence constraints. Computers & Operations Research. 2009; 36(12): 3224-30.
[6] Angel, E., Bampis, E., & Kacem, F. Energy aware scheduling for unrelated parallel machines. IEEE International Conference on Green Computing and Communications: IEEE;2012 .p. 533-40.
[7] Liang, P., Yang, H. D., Liu, G. S., & Guo, J. H. An ant optimization model for unrelated parallel machine scheduling with energy consumption and total tardiness. Mathematical Problems in Engineering. 2015. doi: 10.1155/2015/907034
[8] Li, Z., Yang, H., Zhang, S., & Liu, G. Unrelated parallel machine scheduling problem with energy and tardiness cost. The International Journal of Advanced Manufacturing Technology. 2016; 84(1-4): 213-26.
[9] Cota, L. P., Coelho, V. N., Guimarães, F. G., & Souza, M. J. Bi- criteria formulation for green scheduling with unrelated parallel machines with sequence- dependent setup times. International Transactions in Operational Research. 2018 doi: 10.1111/itor.12566
[10] Soleimani, H., Ghaderi, H., Tsai, P. W., Zarbakhshnia, N., & Maleki, M. 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. 2020; 249: 119428.
[11] Zhu, W., & Tianyu, L. A Novel Multi-Objective Scheduling Method for Energy Based Unrelated Parallel Machines With Auxiliary Resource Constraints. IEEE Access.2019; 7: 168688-168699.
[12] Moon, J. Y., Shin, K., & Park, J. Optimization of production scheduling with time-dependent and machine-dependent electricity cost for industrial energy efficiency. The International Journal of Advanced Manufacturing Technology. 2013; 68(1-4): 523-35.
[13] Ding, J. Y., Song, S., Zhang, R., Chiong, R., & Wu, C. Parallel machine scheduling under time-of-use electricity prices: New models and optimization approaches. IEEE Transactions on Automation Science and Engineering. 2015; 13(2): 1138-54.
[14] Che, A., Zhang, S., & Wu, X. Energy-conscious unrelated parallel machine scheduling under time-of-use electricity tariffs. Journal of Cleaner Production. 2017; 156: 688-97.
[15] Cheng, J., Chu, F., & Zhou, M. An improved model for parallel machine scheduling under time-of-use electricity price. IEEE Transactions on Automation Science and Engineering. 2017; 15(2): 896-9.

[16] Abikarram, J. B., McConky, K., & Proano, R. Energy cost minimization for unrelated parallel machine scheduling under real time and demand charge pricing. Journal of Cleaner Production. 2019; 208: 232-42.

[17] Saberi-Aliabad, H., Reisi-Nafchi, M., & Moslehi, G. Energy-efficient scheduling in an unrelated parallel-machine environment under time-of-use electricity tariffs. Journal of Cleaner Production.2020;249: 119393.

[18] Kurniawan, B. Mathematical Models of Energy-Conscious Bi-Objective Unrelated Parallel Machine Scheduling. Jurnal Teknik Industri.2020; 21(2): 115-125.

[19] Wang, S., Wang, X., Yu, J., Ma, S., & Liu, M. Bi-objective identical parallel machine scheduling to minimize total energy consumption and makespan. Journal of Cleaner Production. 2018; 193: 424-40.

[20] Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. R. Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of discrete mathematics: Elsevier; 1979. p. 287-326.

[21] Ehrgott, M., & Gandibleux, X. Multiobjective combinatorial optimization—theory, methodology, and applications. Multiple criteria optimization: State of the art annotated bibliographic surveys: Springer; 2003. p. 369-444.

[22] Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. Handling multiple objectives with particle swarm optimization. IEEE Transactions on evolutionary computation. 2004; 8(3): 256-79.

[23] Wang, W., Chen, L., Jie, J., Zhao, Y., & Zhang, J. A novel multi-objective particle swarm optimization algorithm for flow shop scheduling problems. International Conference on Intelligent Computing: Springer; 2011 .p. 24-31.

[24] Torabi, S. A., Sahebjamnia, N., Mansouri, S. A., & Bajestani, M. A. A particle swarm optimization for a fuzzy multi-objective unrelated parallel machines scheduling problem. Applied Soft Computing. 2013; 13(12): 4750-62.

[25] Liao, X., Zhang, R., & Chiong, R. Multi-objective optimization of single machine scheduling with energy consumption constraints. IEEE Symposium Series on Computational Intelligence (SSCI): IEEE; 2017. p. 1-8.

[26] Manupati, V. K., Rajyalakshmi, G., Chan, F. T., & Thakkar, J. J. A hybrid multi-objective evolutionary algorithm approach for handling sequence-and machine-dependent set-up times in unrelated parallel machine scheduling problem. Sādhanā. 2017; 42(3): 391-403.

[27] Liu, Y., Liao, X., & Zhang, R. An Enhanced MOPSO Algorithm for Energy-Efficient Single-Machine Production Scheduling. Sustainability,. 2019; 11(19): 5381.

[28] Ulungu, E. L., Teghem, J. F. P. H., Fortemps, P. H., & Tuyttens, D. MOSA method: a tool for solving multiobjective combinatorial optimization problems. Journal of multicriteria decision analysis. 1999; 8(4): 221.

[29] Mansouri, S. A., Hendizadeh, S. H., & Salmasi, N. Bicriteria scheduling of a two-machine flowshop with sequence-dependent setup times. The International Journal of Advanced Manufacturing Technology. 2009; 40(11-12): 1216-26.

[30] Chyu, C. C., & Chang, W. S. Optimizing fuzzy makespan and tardiness for unrelated parallel machine scheduling with archived metaheuristics. The International Journal of Advanced Manufacturing Technology. 2011; 57(5-8): 763.

[31] Mohammadi, H., & Sahraeian, R. Bi-objective simulated annealing and adaptive memory procedure approaches to solve a hybrid flow shop scheduling problem with unrelated parallel machines. IEEE International Conference on Industrial Engineering and Engineering Management: IEEE; 2012. p. 528-32.

[32] Shoaardebili, N., & Fattahi, P. Multi-objective meta-heuristics to solve three-stage assembly flow shop scheduling problem with machine availability constraints. International Journal of Production Research. 2015; 53(3): 944-68.

[33] Tirkolaee, E. B., Goli, A., Hematian, M., Sangaiah, A. K., & Han, T. Multi-objective multi-mode resource constrained project scheduling problem using Pareto-based algorithms. Computing. 2019; 101(6): 547-70.

[34] Shannon, C. E. A mathematical theory of communication. The Bell system technical journal. 1948; 27(3): 379-423.

[35] Zhang, H., Wu, Y., Pan, R., & Xu, G. Two-stage parallel speed-scaling machine scheduling under time-of-use tariffs. Journal of Intelligent Manufacturing. 2020:1-22.

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license.