

## Research Article

# A New Energy-Aware Flexible Job Shop Scheduling Method Using Modified Biogeography-Based Optimization

**Hua Zhang, Ziwei Dai, Wenyu Zhang, Shuai Zhang, Yan Wang, and Rongyu Liu**

*School of Information, Zhejiang University of Finance and Economics, Hangzhou 310018, China*

Correspondence should be addressed to Wenyu Zhang; [wyzhang@e.ntu.edu.sg](mailto:wyzhang@e.ntu.edu.sg)

Received 22 May 2017; Accepted 19 July 2017; Published 22 August 2017

Academic Editor: Thomas Hanne

Copyright © 2017 Hua Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Industry consumes approximately half of the total worldwide energy usage. With the increasingly rising energy costs in recent years, it is critically important to consider one of the most widely used energies, electricity, during the production planning process. We propose a new mathematical model that can determine efficient scheduling to minimize the makespan and electricity consumption cost (ECC) for the flexible job shop scheduling problem (FJSSP) under a time-of-use (TOU) policy. In addition to the traditional two subtasks in FJSSP, a new subtask called speed selection, which represents the selection of variable operating speeds, is added. Then, a modified biogeography-based optimization (MBBO) algorithm combined with variable neighborhood search (VNS) is proposed to solve the biobjective problem. Experiments are performed to verify the effectiveness of the proposed MBBO algorithm for obtaining an improved scheduling solution compared to the basic biogeography-based optimization (BBO) algorithm, genetic algorithm (GA), and harmony search (HS).

## 1. Introduction

Under the pressure of sustainable development, manufacturers today must consider not only production efficiency but also energy consumption. In recent years, the demand for energy and the investment in energy have continued to increase. Global energy consumption was 524 quadrillion Btu in 2010 and is expected to increase by 56.5%, to 820 quadrillion Btu, by 2040 [1]. Industry accounted for 52% of energy usage worldwide in 2010 [1]. As one of the most widely used industrial energies, electricity is an important sector that cannot be neglected. However, the rising cost of energy sources to generate electricity such as coal, natural gas, and nuclear energy leads to an increasing electricity consumption cost (ECC). Moreover, the characteristics of electricity make its storage inefficient. Although increasingly heavy investment has been made to support backup infrastructures, with the variable demand of consumers it remains difficult to achieve a trade-off between demand and supply. To address this issue, electricity suppliers have implemented demand response technology.

Time-of-use (TOU) electricity pricing, as the main method of demand response technology, means that the price

of electricity is dependent on the electricity consumption at a specific time. That is, it is related to electricity demand. The aim of this method is to encourage consumers and manufacturers to reduce electricity consumption in high-peak periods. For example, the electricity price in high-peak periods can be double that in off-peak periods. Thus, manufacturers can exploit TOU pricing and shift electricity consumption from high-peak periods to off-peak periods. Consequently, ECC, a significant industry cost, can be somewhat reduced.

Three kinds of actions can be implemented to reduce ECC [2]. Manufacturing companies can purchase new energy-efficient machines; this is expensive and has minimal impact. New product forms can be designed. This method is difficult for medium and small-sized enterprises owing to large capital investment. Compared to the aforementioned two methods, production scheduling is more reasonable and applicable because of its minimal investment and strong practicality. The flexible job shop scheduling problem (FJSSP), which has two subtasks, is an extension of the classical job shop scheduling problem. Machine selection is an essential sector in FJSSP. Processing routes are arranged after selection of the appropriate machines. In this study, we assume that each

candidate machine has adjustable speeds. Machines in high-speed mode can reduce processing time but increase energy consumption, whereas machines in low-speed mode are relatively energy saving but, more time-consuming. Reasonable arrangements of machines of different speeds in various periods can reduce ECC efficiently under the TOU pricing strategy that contributes to price fluctuations in different periods.

In this study, a new mathematical model is proposed to solve the biobjective problem, namely, makespan and ECC in FJSSP. Moreover, we present a modified biogeography-based optimization (MBBO) algorithm that adopts the extended migration model, combines elitism strategy, and employs variable neighborhood search (VNS). This kind of extension can improve the local search ability and accelerate the convergence of the original biogeography-based optimization (BBO) algorithm. A series of experiments are performed on extended well-known benchmark instances. We compare the MBBO algorithm with the basic BBO algorithm, genetic algorithm (GA), and harmony search (HS) to demonstrate the effectiveness of the proposed algorithm.

The remainder of this paper is organized as follows. In Section 2, we discuss some related works. In Section 3, the problem definition and mathematical model are presented. Section 4 describes the proposed MBBO algorithm for solving the biobjective problem that minimizes makespan and ECC in FJSSP. In Section 5, experiments are performed to validate the effectiveness of the proposed MBBO algorithm. In Section 6, we present conclusions and discuss future works.

## 2. Related Works

Both industry and academia have addressed green-manufacturing in recent years, and research concentrating on energy-efficient scheduling is gradually increasing. In the single machine environment, Mouzon et al. [3] determined that there existed considerable energy waste, accounting for 80 percent of total energy consumption. Thus, they proposed a “Turn Off/Turn On” method in the idle phase of a single CNC machine to reduce energy consumption. In their follow-up work, they considered total tardiness and energy consumption and used a novel greedy heuristic search method to solve the biobjective optimization problem [4]. Considering variable electricity price, Shrouf et al. [5] utilized GA to reduce ECC in a production-scheduling problem of a single machine. They employed a “Turn On/Turn Off” strategy to determine when to shut down the machine to save electricity cost. Che et al. [6] developed a new mathematical model to address a single machine scheduling problem under TOU electricity tariffs to minimize the total electricity cost and proposed a greedy insertion heuristic to solve it. Ding et al. [7] investigated the unrelated parallel-machine scheduling problem to optimize ECC under the TOU tariff. They considered the TOU setting where electricity prices fluctuate frequently in short periods and proposed a column generation heuristic. Further, a time-interval-based mixed-integer programming formulation was developed allowing a more efficient solution. In the flow shop environment, Bruzzone et al. [8]

established a mixed-integer programming model where energy consumption was considered without changing the original jobs’ assignment and sequencing provided by the reference schedule generated by an advanced planning and scheduling system. Lin et al. [9] proposed an integrated model for processing parameter optimization and flow shop scheduling considering makespan and carbon footprint and introduced three carbon-footprint reduction strategies. Lu et al. [10] proposed an energy-efficient permutation flow shop scheduling model considering controllable transportation times solved by backtracking search algorithm and developed a new energy saving strategy. In the job shop environment, Liu et al. [11] used a novel multiobjective GA to minimize the total nonprocessing electricity consumption and total weighted tardiness and employed a “Turn On/Turn Off” method to save electricity.

Diaz et al. [12] concluded that energy consumption is related to the cutting speed and cutting at a higher speed can save more energy than traditional-speed cutting. Fang et al. [13] proposed a new model for solving the flow shop scheduling problem to reduce peak power load, energy consumption, and carbon footprint. They assumed that jobs were operated at dynamic speeds. However, the commercial software they applied directly was only practical for a small-sized problem. Thus, there could exist a gap between the theory and application in industry. To address a parallel-machine scheduling problem considering the total weighted job tardiness and power cost, Fang and Lin [14] attempted to adjust CPU frequencies to force jobs to be processed at a variable machine speed. They assumed that higher machine speed saved time but increased power cost, whereas reduced machine speed sacrificed completion times but led to higher energy cost. They proposed two heuristic algorithms and designed a specific encoding scheme for a particle swarm optimization algorithm for solving the problem. Luo et al. [15] proposed an extended ant colony optimization metaheuristic to solve a hybrid flow shop scheduling problem under the TOU tariff. According to a parameter analysis, the combination of a high-power machine and a low-power machine could save more energy than using two middle-power machines. Sharma et al. [16] proposed a new economical scheduling model under TOU pricing where machine speeds were allowed to vary. Both economic and ecological benefits were simultaneously achieved using a multicriteria metaheuristic optimization method. For the job shop environment, Salido et al. [17] presented a biobjective problem to minimize makespan and energy consumption where machines could work at different speeds.

However, literature concentrating on energy in FJSSP is still lacking. Moon and Park [18] employed constraint programming and mixed-integer programming approaches to minimize production cost in FJSSP considering electricity costs with distributed energy resources. Zhang et al. [19] employed the extended nondominated sorting genetic algorithm II (NSGA-II), considering makespan, machine workload, and carbon footprint, for solving FJSSP. He et al. [20] utilized the nested partitions algorithm to solve a new mathematical model formulated by a mixed-integer programming to reduce energy consumption in FJSSP. Liu

and Tiwari [21] developed an optimization approach based on NSGA-II to make the scheduling plans for a carbon fiber reinforced polymer recycling workshop, considering both makespan and energy reduction under the circumstance of flexible job shop. Yang et al. [22] used NSGA-II for solving FJSSP considering makespan and total energy consumption under stochastic processing times. Lei et al. [23] proposed a shuffled frog-leaping algorithm to investigate FJSSP with the objective of minimizing workload balance and total energy consumption. Yin et al. [24] considered productivity, energy efficiency, and noise reduction in FJSSP where machining spindle speed affected the production time, power, and noise.

Metaheuristics and intelligent algorithms have been widely used in computer-integrated manufacturing. In our previous work [25], to determine the manufacturing resource allocation for supply chain deployment, extended GA was utilized for solving the multiobjective decision-making model. In our follow-up work [26], the teaching-learning-based optimization algorithm was improved to plan the distributed manufacturing resource allocation optimally. Pan et al. [27] proposed a chaotic HS to minimize makespan to solve the permutation flow shop scheduling problem considering limited buffers. Ribas et al. [28] proposed an improved artificial bee colony algorithm for solving the blocking flow shop problem by employing various strategies for each phase. Karthikeyan et al. [29] integrated discrete firefly algorithm with local search method to solve FJSSP considering makespan, the workload of the critical machine, and the total workload of all machines. Li and Gao [30] used GA for global search and tabu search for local search to achieve a balance between the intensification and diversification for solving FJSSP. As one of the metaheuristics, the BBO algorithm based on population used in this study, which aims to solve optimization problems, was first proposed by Simon [31] in 2008. Inspired by the biogeography theory, the BBO algorithm assumes that a suitable place for living has a high habitat suitability index (HSI). It is clear that HSI is related to many factors such as rainfall, temperature, and soil. These factors are defined as suitability index variables (SIVs) in the BBO algorithm. To our knowledge, a habitat with a high HSI usually has a relatively significant diversity of species, whereas species of the habitat with low HSI are minimal. Therefore, if the number of species is overly large, then the habitat with a high HSI cannot persist and migration will subsequently occur. In addition to the migration operator, mutation is another main operator that can randomly change SIVs, accordingly affecting the HSI of the specific habitat. A habitat can evolve constantly based on the two operations. The BBO algorithm has been applied and improved in scheduling problems. Rahmati and Zandieh [32] utilized the BBO algorithm to solve FJSSP and performed a comparison between the BBO algorithm and GA for an improved understanding. Wang and Duan [33] improved the standard BBO algorithm by integrating chaos theory and a strategy called “searching around the optimum,” which can prevent the local optima effectively. Lin and Zhang [34] proposed a hybrid BBO algorithm, combined with new heuristics to minimize the makespan in the distributed assembly permutation flow shop scheduling problem. However, the original

BBO algorithm has a poor local search ability and converges slowly. The proposed MBBO algorithm in this work adopts an extended migration model and is integrated with VNS, which can enhance the local search ability and accelerate the convergence. This kind of extension avoids the weaknesses of the BBO algorithm effectively.

### 3. Problem Definition and Modeling

The definition of FJSSP can be described as  $n$  independent jobs  $\{J_1, J_2, \dots, J_n\}$  to be processed on a set of  $m$  machines  $\{M_1, M_2, \dots, M_m\}$ . A job  $J_i$  consists of a sequence of operations  $O_{ij}$ . FJSSP aims to achieve the objectives by determining the appropriate machine assignment (MA) and operation sequence (OS).

FJSSP for reducing ECC in this study is an extension of classical FJSSP. It not only considers the traditional objective makespan but also ECC under the TOU tariffs. We assume that the machine frequency is adjustable and jobs can be operated at variable speeds. Thus, different from classical FJSSP with two subproblems containing MA and OS, another subproblem that considers the selection of speeds is included.

The objective of this study is to achieve a balance between makespan and ECC by assigning jobs to the appropriate machines with selected operating speeds. To make this scheduling problem more concise, there are some assumptions to be satisfied.

- (1) An operation, once started, cannot be interrupted until it is completed.
- (2) Each machine can process only one job at a time and each job can only be processed on one machine at a time.
- (3) The operations of each job have priority constraints.
- (4) All machines are available during the planning horizon and all jobs can be processed at time zero.
- (5) The setup and adjustment periods of the machines are negligible.

**3.1. Notations.** To understand the mathematical model more clearly, the corresponding parameters and decision variables are explained as follows:

*Sets*

$J$ : set of jobs  $\{J_1, J_2, \dots, J_n\}$

$M$ : set of machines  $\{M_1, M_2, \dots, M_m\}$ .

*Parameters*

$i$ : index of jobs,  $i = 1, \dots, n$

$k$ : index of machines,  $k = 1, \dots, m$

$j$ : index of operation sequences,  $j = 1, \dots, l_i$

$t$ : index of time periods,  $t = 1, \dots, r$

$s$ : index of machine processing speeds,  $s = 1, \dots, q$

$O_{ij}$ : the  $j$ th operation of  $J_i$

$P_{ij}^{sk}$ : number of time periods of processing  $O_{ij}$  on machine  $k$  at speed  $s$

$S_{ij}$ : start time of  $O_{ij}$

$C_{ij}$ : finishing time of  $O_{ij}$

$C_{\max}$ : makespan of the schedule

$E_{stk}$ : processing energy consumption on machine  $k$  at speed  $s$  at period  $t$

$E_{otk}$ : idle energy consumption on machine  $k$  at period  $t$

$EC_t$ : the electricity cost at period  $t$

ECC: total electricity cost of the schedule

### Decision Variables

$x_{ij}^{sk}$ : binary variable, if  $O_{ij}$  is processed on machine  $k$  at speed  $s$ , then  $x_{ij}^{sk} = 1$ ; otherwise,  $x_{ij}^{sk} = 0$

$y_{ij}^{skt}$ : binary variable, if  $O_{ij}$  is processed on machine  $k$  at speed  $s$  at period  $t$ , then  $y_{ij}^{skt} = 1$ ; otherwise,  $y_{ij}^{skt} = 0$

$\alpha_{stk}$ : binary variable, if machine  $k$  is at the processing status at speed  $s$  during period  $t$ , then  $\alpha_{stk} = 1$ ; otherwise  $\alpha_{stk} = 0$

$\beta_{tk}$ : binary variable, if machine  $k$  is at the standby status during period  $t$ , then  $\beta_{tk} = 1$ ; otherwise  $\beta_{tk} = 0$ .

**3.2. Formulations.** During the production process, the time and energy consumption of the setting up and transforming periods are relatively small compared to those of the processing and standby periods. In this study, we only consider three machine statuses: processing, standby, and off status.

From the following two equations, we can determine the makespan and total electricity cost:

$$C_{\max} = \max_{1 \leq i \leq n} \{C_{ii}\}, \quad (1)$$

$$ECC = \sum_{t=1}^r \sum_{k=1}^m EC_t \left( \sum_{s=1}^q E_{stk} \alpha_{stk} + E_{otk} \beta_{tk} \right).$$

Before we transform the biobjective problem, which aims to minimize the makespan and ECC into a monoobjective, the normalization method provided in [35] must be performed. Because the values of the makespan and ECC belong to intervals with different lengths, a domination could occur leading to an inefficiency of the evaluation. Thus, the normalized makespan value  $\overline{C_{\max}}$  and normalized total electricity cost value  $\overline{ECC}$  can be obtained. By setting weights  $w_1$  and  $w_2$  manually in the interval between zero and one, a balance can be achieved between time and ECC. Weight coefficients summed to one are used to reflect the respective importance of the objectives. For example, if the decision maker considers time more important than the other objectives, the weight coefficient of time can be set higher

manually. This kind of method is more flexible for different decision makers when planning production schedules.

The mathematical model is indicated as follows.

### Objective

$$\min \{w_1 \overline{C_{\max}} + w_2 \overline{ECC}\}. \quad (2)$$

### Constraints

$$C_{ij} = S_{ij} + \sum_{s=1}^q P_{ij}^{sk} x_{ij}^{sk} \quad \forall i, j, k \quad (3)$$

$$S_{i(j+1)} - S_{ij} \geq \sum_{s=1}^q P_{ij}^{sk} x_{ij}^{sk} \quad 1 \leq j \leq l_i - 1 \quad \forall i, k \quad (4)$$

$$\sum_{k=1}^m \sum_{s=1}^q y_{ij}^{skt} = 1 \quad \forall i, j, t \quad (5)$$

$$\sum_{i=1}^n \sum_{j=1}^{l_i} y_{ij}^{skt} \leq 1 \quad \forall s, k, t \quad (6)$$

$$\sum_{t=1}^r y_{ij}^{skt} = P_{ij}^{sk} x_{ij}^{sk} \quad \forall i, j, s, k \quad (7)$$

$$\sum_{s=1}^q \alpha_{stk} + \beta_{tk} \leq 1 \quad \forall t, k \quad (8)$$

$$S_{ij} \geq 0,$$

$$C_{ij} \geq 0 \quad (9)$$

$$\forall i, j$$

$$x_{ij}^{sk} \in \{0, 1\} \quad \forall i, j, s, k \quad (10)$$

$$y_{ij}^{skt} \in \{0, 1\} \quad \forall i, j, s, k, t \quad (11)$$

$$\alpha_{stk} \in \{0, 1\} \quad \forall s, t, k \quad (12)$$

$$\beta_{tk} \in \{0, 1\} \quad \forall t, k. \quad (13)$$

Equation (2) is the objective function, which is aimed at minimizing the related makespan and ECC. Constraint (3) guarantees that each operation can be completed without interruption once it starts. Constraint (4) guarantees that only when the prior operation is finished can the next operation of the same job be processed. That is, it is the priority constraint of the operations. Constraint (5) ensures that each operation cannot be processed on more than one machine at one speed at the same time. Constraint (6) ensures that each machine can perform at most one operation at any time. Constraint (7) describes the relationship between decision variables  $x_{ij}^{sk}$  and  $y_{ij}^{skt}$ . Constraint (8) confines each machine to be operating at only one specific speed at each time period and requires that each machine must be in one of the three possible states at any time. Constraint (9) imposes the nonnegative restriction

TABLE 1: Example of FJSSP with three jobs and three machines.

Job	Operation	$M_1$	$M_2$	$M_3$
$J_1$	$O_{11}$	3	6	4
	$O_{12}$	5	—	3
	$O_{21}$	4	7	3
$J_2$	$O_{22}$	—	5	3
	$O_{23}$	3	4	6
$J_3$	$O_{31}$	6	5	5
	$O_{32}$	7	6	—

on the start time and finishing time of each operation. Constraints (10)–(13) specify that the decision variables  $x_{ij}^{sk}$ ,  $y_{ij}^{skt}$ ,  $\alpha_{stk}$ , and  $\beta_{tk}$  are binary.

#### 4. MBBO Algorithm for FJSSP

In this section, the MBBO algorithm is proposed to solve the FJSSP considering both makespan and ECC. To enable the proposed approach adaptive to the investigated problem, a new three-vector string representation is employed. In addition, VNS is integrated to improve the performance of the basic BBO algorithm.

**4.1. Encoding and Decoding.** The first step of solving FJSSP using the MBBO algorithm is to obtain a reasonable encoding and decoding scheme that can effectively represent the problem. Owing to the quality of the traditional FJSSP, the encoding scheme design includes two subtasks, namely, MA, which selects the machine where the operation will be processed, and OS, which determines the specific operation sequence of all the jobs. In this study, to meet the assumption that machine frequency is adjustable, a new subtask called speed selection (SS) is introduced. It must be noted that the item called habitat in the BBO algorithm is similar to that called chromosome in GA. Accordingly, we extended the representation employed in [32] to form a three-vector string. For a clear and intuitive understanding, we present Table 1 as a simple example to describe the encoding and decoding scheme.

(1) *MA.* The length of a habitat is equal to the number of all operations to be processed. Each position means the index of the assigned machine for the corresponding operation. Using Table 1 as an example, given a habitat  $\{2\ 2\ 1\ 2\ 3\ 2\ 1\}$ , the position of the second operation of  $J_1$  is “2,” which indicates that  $O_{12}$  is processed on the second machine that can be selected from  $\{M_1, M_3\}$ , namely,  $M_3$ .

(2) *SS.* The length of SS is equal to that of MA. The positive integer number from one to three of each position indicates the selected machine speed level of the corresponding position of MA. Under the circumstance that the MA habitat is  $\{2\ 2\ 1\ 2\ 3\ 2\ 1\}$  and SS habitat is  $\{1\ 3\ 3\ 2\ 1\ 3\ 1\}$ , it can be concluded that the second operation of  $J_1$  is processed on the second machine at high speed.

(3) *OS.* The length of OS is also equal to that of MA. The positive integer value means the index of the job, and the  $p$ th occurrence of one positive integer value indicates the  $p$ th operation of one job. Further, provided that there is a habitat  $\{2\ 1\ 3\ 1\ 2\ 2\ 3\}$ , the first gene means the first operation of  $J_2$ , namely,  $O_{21}$ , and the second occurrence of “2” means the second operation of  $J_2$ .

Decoding must identify the selected machine of each operation first; then, the machine speed is identified based on the MA habitat. The operation sequence is dependent on the initial position to the end position of the habitat. That is, the operation in the first position will be scheduled first. According to the decoding scheme, a schedule plan can be confirmed.

**4.2. Migration Operator.** HSI is one of the most important factors affecting the movement of species. A habitat with a high HSI has a large number of species and will not be capable of persisting, whereas a habitat with a low HSI lacks species. To the best of our knowledge, in the real world, species can choose to migrate to other habitats once their former habitat is full (of species). Similar to the biogeography theory, migration is a vital probabilistic operator to change the SIVs of habitats by sharing features among candidate solutions in the BBO algorithm. The migration operator can maintain effective solutions and to some degree expand the search space of solutions. Ma [36] performed experiments and concluded that applying different migration models has significant effects on the BBO algorithm and that nonlinear migration models inspired by the biogeography theory generally outperform linear migration models. Based on this conclusion, we employ the following sinusoidal migration model [36]. According to (14)–(15), the immigration and emigration rate can be calculated, respectively:

$$\lambda_k = \frac{I_{\max}}{2} \left[ \cos \left( \pi \frac{S_k}{NP} \right) + 1 \right], \quad (14)$$

$$\mu_k = \frac{E_{\max}}{2} \left[ -\cos \left( \pi \frac{S_k}{NP} \right) + 1 \right], \quad (15)$$

where  $\lambda_k$  denotes the immigration rate,  $\mu_k$  denotes emigration rate,  $I_{\max}$  is the maximum immigration rate,  $E_{\max}$  is the maximum emigration rate,  $S_k$  is the number of species of habitat  $k$ , and  $NP$  is the maximum number of species that habitats can support.

**4.3. Mutation Operator.** Unpredictable natural disasters can damage the environment of habitats and destroy the ecological balance causing the HSI to subsequently change. The mutation operator represents this kind of phenomenon in nature. It is a key operator in the BBO algorithm that can change the goodness of habitats randomly and improve the diversity of solutions. It is obvious that the mutation function has an important role. The following formula [31] can calculate the probability of mutation  $m_k$  for the  $k$ th habitat:

$$m_k = m_{\max} \left( 1 - \frac{P_k}{P_{\max}} \right), \quad (16)$$

where  $m_{\max}$  denotes the maximum mutation probability and  $P_{\max} = \operatorname{argmax}_r P_r$  ( $r = 1, 2, \dots, NP$ ).  $P_k$  can be calculated according to the following formula [31]:

$$P_k = \begin{cases} -(\lambda_k + \mu_k)P_k + \mu_{k+1}P_{k+1}, & NP = 0, \\ -(\lambda_k + \mu_k)P_k + \lambda_{k-1}P_{k-1} + \mu_{k+1}P_{k+1}, & 1 \leq S_k \leq NP - 1, \\ -(\lambda_k + \mu_k)P_k + \lambda_{k-1}P_{k-1}, & S_k = NP. \end{cases} \quad (17)$$

However, it is possible for the mutation operator to decrease the goodness of solutions with a high HSI. Therefore, it is necessary to adopt the elitism strategy to retain the best  $Z_{\text{elitism}}$  solutions.

**4.4. Variable Neighborhood Search.** VNS is a metaheuristic method first proposed by Mladenović and Hansen [37]. It is utilized to solve both continuous and discrete optimization problems and has been employed in FJSSP efficiently. VNS improves solution quality by searching neighborhoods of the current solution, which avoids the possibility of falling into local optima.

**4.4.1. Neighborhood Structures.** Neighborhood structures are vital in the implementation of VNS. Limited neighborhood structures save computational time; however, it is difficult to obtain the optima and vice versa. Thus, achieving a trade-off and defining reasonable neighborhood structures are essential for VNS. The neighborhood structures used in this research are defined as follows.

#### Neighborhood Structures for Shaking

(1) *N5 Neighborhood.* N5 neighborhood was proposed by Nowicki and Smutnicki [38] and is based on the critical path theory. The critical path theory integrated with VNS has been successfully applied in FJSSP. The makespan of a schedule is equal to the length of the critical path [39]. There may exist one or more critical paths in a schedule. An operation belonging to a critical path is called a critical operation. A critical block is a set of successive critical operations processed on the same machine. To obtain a superior solution, an efficient method is to determine each critical path and perform operations on this path. An N5 neighborhood can be generated by exchanging the last two operations of the head block, interchanging the first two operations of the tail block, and swapping the first two operations or the last two operations of the middle blocks.

(2) *Machine Assignment Neighborhood.* Determine the machine that processes the most critical operations denoted as  $M_k$ . Randomly select an operation  $O_{ij}$  with one more candidate machines. From the candidate machines, select the machine whose processing time of  $O_{ij}$  is least.

#### Neighborhood Structures for Local Search

(1) *Insert Neighborhood.* Randomly select an operation and insert this operation into another position of a feasible schedule.

(2) *Swap Neighborhood.* Randomly select two operations belonging to different jobs and swap these operations in the OS habitat.

**4.4.2. Outline of VNS.** A generalization of the main procedure of VNS can be summarized as follows.

*Step 1.* Define the neighborhood structures and randomly generate a solution  $x$  denoted as the initial solution. Define the maximum outer loop iteration OL. Set  $i = 0$  and  $j_{\max} = 2$ .

*Step 2.* Set  $i = i + 1$  and  $j = 1$ .

*Step 3 (shaking).* Randomly generate a neighbor solution  $x_1$  based on the  $j$ th neighborhood structure for shaking.

*Step 4 (local search).* Obtain a local optima denoted as  $x_2$  from  $x_1$  based on the predetermined neighborhood structures for local search.

*Step 5 (move or not).* If HSI of  $x_2$  is greater than that of  $x$ , then let  $x_2$  replace  $x$  and return to Step 3. Otherwise, perform Step 6.

*Step 6.* If  $j \neq j_{\max}$ , set  $j = j + 1$ , and return to Step 3.

*Step 6.1.* If  $j = j_{\max}$  and  $i \leq OL$ , return to Step 2.

*Step 6.2.* If  $j = j_{\max}$  and  $i > OL$ , terminate the procedure.

**4.5. Time Complexity of the Proposed Algorithm.** The time complexity of the proposed algorithm is different when applied to different problems. According to the parameters explained before, let the number of all operations be  $L$ , the time complexity in one iteration of each process of the MBBO algorithm applied to the proposed model in this study is shown as follows:

(1) Migration operator is  $O(NP \times L)$ .

(2) Mutation operator is  $O(NP \times L)$ .

(3) VNS is  $O(NP \times m)$ .

(4) HSI calculation is  $O(NP \times m)$ .

Therefore, if the maximum iteration number is  $K$ , by synthesizing the time complexity of each process, the overall complexity of the proposed MBBO algorithm is  $O(K \times NP \times (m + L))$ .

## 5. Experiments

A series of experiments were performed comparing the proposed approach with the basic BBO algorithm [31] and two widely used algorithms, GA [40] and HS [41]. In this study, the C# programming language was used to test the MBBO algorithm. The experimental tests were implemented on a personal computer with an Intel(R) Core(TM) 2.3 GHz and 12 GB RAM operated on Windows 10.

**5.1. Experimental Design.** The parameters of each algorithm are shown in Table 2. The initial population size and the

TABLE 2: Parameters of each algorithm.

Algorithm	Parameters
MBBO	The maximum immigration rate $I_{\max} = 1$ , the maximum emigration rate $E_{\max} = 1$ , the maximum mutation probability $m_{\max} = 0.1$ , and the number of elites $Z_{\text{elitism}} = 2$
BBO	The maximum immigration rate $I_{\max} = 1$ , the maximum emigration rate $E_{\max} = 1$ , and the maximum mutation probability $m_{\max} = 0.1$
GA	The crossover probability $p_c = 0.85$ and the mutation probability $p_m = 0.1$
HS	The harmony memory considering rate $c_r = 0.8$ and the pitch adjusting rate $pa_r = 0.3$

TABLE 3: Time and power under different speeds in the processing status.

Speed	Time	Power
1	$T_1 = \text{rand}(1.2T_2, 1.4T_2)$	$P_1 = \text{rand}\left(\frac{P_2}{1.4}, \frac{P_2}{1.2}\right)$
2	$T_2 = \text{rand}(1.2T_3, 1.4T_3)$	$P_2 = \text{rand}\left(\frac{P_3}{1.4}, \frac{P_3}{1.2}\right)$
3	$T_3$	$P_3 = \text{rand}(4, 8)$

maximum iterations were set to 100 and 200, respectively, for all algorithms. To avoid the randomness of the results, each experiment was performed 30 times to obtain an average value.

Because there exist no benchmarks containing different machine speeds and corresponding energy consumption of FJSSP until now, to evaluate the performance of the proposed MBBO algorithm, we extended two sets of well-known benchmark problems, that is, Kacem instances [42] and Brandimarte instances [43]. The number of jobs ranged from four to ten; the number of machines ranged from five to ten in the Kacem instances ( $4 \times 5$ ,  $8 \times 8$ ,  $10 \times 7$ , and  $10 \times 10$ ); the number of jobs ranged from ten to twenty; the number of machines ranged from four to fifteen in the Brandimarte instances (MK01–MK10). Suppose that each machine owned three possible adjustable operating speeds, high, medium, and low. The processing time at the high operating speed, denoted as  $T_3$ , was equal to the processing time generated originally. The machine power during the idle time was generated randomly in the interval between two and four. The processing time of the operation  $T_s$  ( $s = 1, 2, 3$ ) and the related machine power  $P_s$  ( $s = 1, 2, 3$ ) under different operating speeds were defined in Table 3.

**5.2. Performance Evaluation.** Generally, TOU pricing policies differ in different seasons. To evaluate the performance of the model, two cases under the TOU policies in summer and winter were tested. Considering the fluctuations in electricity pricing, one work day was divided into 96 periods with the same length. Several periods can compose a duration and the electricity price of the adjacent durations was different. Based on the TOU policies in real life, Table 4 presents the specific TOU electricity prices set in this study.

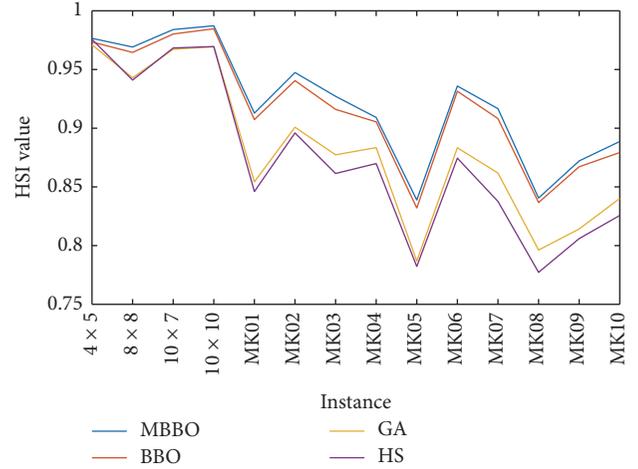
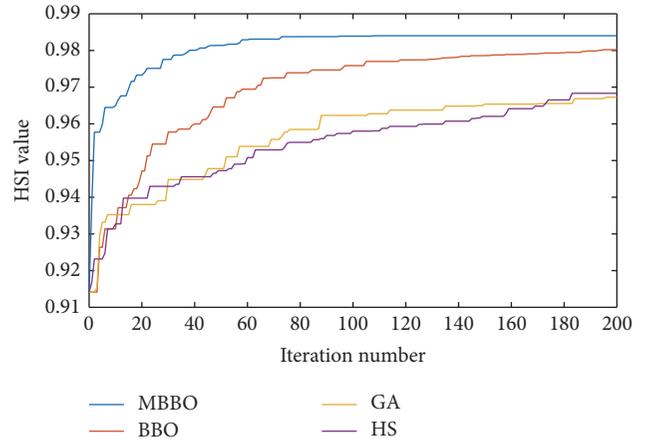


FIGURE 1: HSI value for all instances of four approaches under summer TOU electricity pricing.


 FIGURE 2: Evolution curves of four approaches for  $10 \times 7$  under summer TOU electricity pricing.

**5.2.1. Case 1.** In Case 1, we adopted TOU electricity pricing for summer; all extended benchmarks were performed to test the proposed method, and  $w_1$  was set to 0.6, and  $w_2$  was set to 0.4.

Figure 1 displays the HSI value of both Kacem and Brandimarte instances. It can be observed that the MBBO algorithm outperformed the other three methods for all instances. Based on the results obtained by the four methods, it is clear that the MBBO algorithm was more effective.

The results and evolution curves of four approaches for  $10 \times 7$  under summer TOU electricity pricing are indicated in Figure 2. According to Figure 2, the HSI value obtained by the MBBO algorithm was the highest among the four approaches. The number of iterations of the proposed approach was less than the other three approaches, which indicates its high convergence. It is clear that the performance of the MBBO algorithm was superior.

The specific operation sequence for MK02 optimized by the MBBO algorithm under TOU policy in summer is displayed in Figure 3.

TABLE 4: TOU electricity pricings in summer and winter.

	Case 1 (summer)				Case 2 (winter)				
Duration	6	48	30	12	6	24	36	24	6
Price	5	8	16	8	5	16	8	16	5

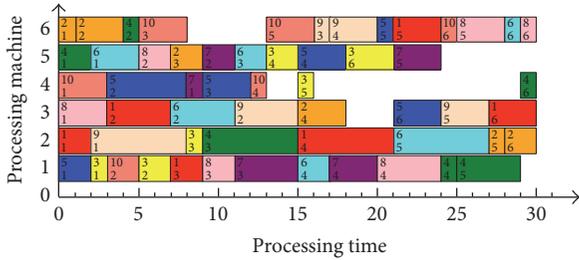


FIGURE 3: Gantt chart of an optimal solution of MK02 by MBBO algorithm under summer TOU electricity pricing.

Although Machine 3, Machine 4, and Machine 6 remain idle for several periods and the total completion time increased slightly, the objective of minimizing makespan and ECC was attained by shifting electricity consumption from high-peak periods to other periods.

To evaluate the impact of weight combination for solutions, experiments were performed and Table 5 shows the results obtained by MBBO and other algorithms with different weights (the best value is in bold).

From Table 5, it can be noted that, under summer TOU electricity pricing, the proposed MBBO algorithm can obtain better solutions for all instances than other algorithms under four different weight combinations. Furthermore, as the importance of time increases, HSI value is rising gradually, and the highest HSI value can be up to 0.99149 by the proposed algorithm in the Instance  $10 \times 10$  when  $w_1 = 0.8$  and  $w_2 = 0.2$ .

5.2.2. Case 2. In Case 2, we adopted the TOU policy for winter; the other conditions including the preference settings  $w_1 = 0.6$  and  $w_2 = 0.4$  remained unchanged.

From Figure 4, it can be seen that the proposed MBBO algorithm outperformed the other three algorithms for all instances. This indicates that the MBBO algorithm was effective under electricity pricing for both summer and winter.

According to the results and evolution curves of four approaches for  $10 \times 7$  under the winter TOU electricity pricing indicated in Figure 5, we can see that the HSI value calculated by the proposed MBBO algorithm was the highest compared to the other three approaches when the termination condition was met. Further, the number of iterations of the MBBO algorithm was less than the comparative algorithms. Thus, the proposed method was more competitive.

The specific operation sequence for MK02 optimized by the MBBO algorithm under the TOU policy for winter is presented in Figure 6.

We can see that, at an intermediary stage, the majority of machines are in idle status. This is because the machines were designed to remain idle in the high-peak periods to reduce

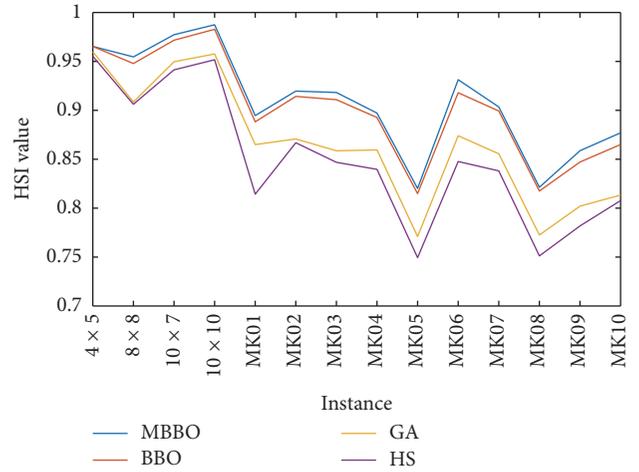


FIGURE 4: HSI value for all instances of four approaches under winter TOU electricity pricing.

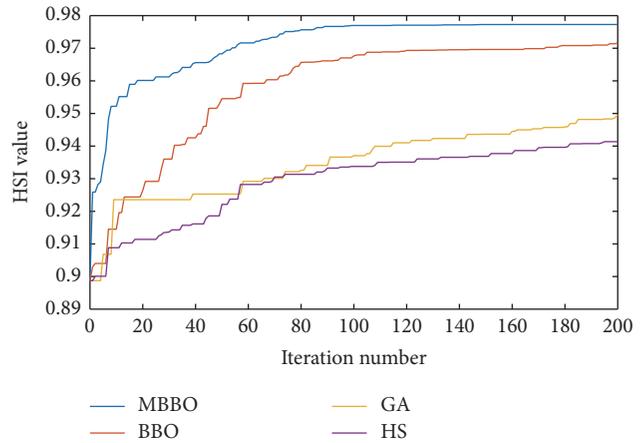


FIGURE 5: Evolution curves of four approaches for  $10 \times 7$  under winter TOU electricity pricing.

ECC and a trade-off between the makespan and ECC was achieved.

To further demonstrate the effectiveness of the MBBO algorithm, Table 6 shows the results obtained by MBBO and other algorithms with different weights (the best value is in bold).

It can be concluded from Table 6 that, under winter TOU electricity pricing with four different weight combinations, the proposed MBBO algorithm can perform better than other algorithms in terms of stability and solution quality. Furthermore, HSI value increases with the improvement of time importance, and the highest HSI value 0.99145 can be

TABLE 5: Results of Kacem and Brandimarte instances with different weights under summer TOU electricity pricing.

Instance	$w_1 = 0.2, w_2 = 0.8$			$w_1 = 0.4, w_2 = 0.6$			$w_1 = 0.6, w_2 = 0.4$			$w_1 = 0.8, w_2 = 0.2$						
	MBBO	BBO	GA	HS	MBBO	BBO	GA	HS	MBBO	BBO	GA	HS				
4 × 5	<b>0.96740</b>	0.96523	0.96679	0.96517	<b>0.96751</b>	0.96682	0.95853	0.96498	<b>0.97655</b>	0.97344	0.97104	0.97571	<b>0.98645</b>	0.98416	0.98527	0.98321
8 × 8	<b>0.94496</b>	0.94420	0.91499	0.92710	<b>0.95864</b>	0.95807	0.92819	0.93308	<b>0.96915</b>	0.96455	0.94296	0.94101	<b>0.98089</b>	0.97756	0.95790	0.96229
10 × 7	<b>0.97428</b>	0.97046	0.95653	0.95715	<b>0.97944</b>	0.97506	0.96462	0.96390	<b>0.98402</b>	0.98022	0.96726	0.96834	<b>0.98860</b>	0.98434	0.96854	0.97235
10 × 10	<b>0.98361</b>	0.98051	0.96219	0.96917	<b>0.98524</b>	0.98235	0.96332	0.96781	<b>0.98725</b>	0.98471	0.96939	0.96957	<b>0.99149</b>	0.98844	0.98234	0.97792
MK01	<b>0.83822</b>	0.82780	0.79288	0.78650	<b>0.89032</b>	0.84678	0.81475	0.81417	<b>0.91281</b>	0.90737	0.85433	0.84609	<b>0.93266</b>	0.93166	0.90572	0.90469
MK02	<b>0.91006</b>	0.87653	0.83451	0.81503	<b>0.92387</b>	0.91994	0.86659	0.87007	<b>0.94744</b>	0.94052	0.90085	0.89596	<b>0.96538</b>	0.95997	0.93434	0.91907
MK03	<b>0.89675</b>	0.89361	0.81558	0.79805	<b>0.89719</b>	0.89226	0.83237	0.82116	<b>0.92717</b>	0.91604	0.87733	0.86143	<b>0.95073</b>	0.94260	0.91115	0.89943
MK04	<b>0.83890</b>	0.83414	0.77814	0.76929	<b>0.86322</b>	0.85777	0.81870	0.80163	<b>0.90914</b>	0.90540	0.88348	0.86989	<b>0.93359</b>	0.93229	0.89804	0.88511
MK05	<b>0.75936</b>	0.75759	0.69925	0.68448	<b>0.81836</b>	0.81056	0.76774	0.73909	<b>0.83882</b>	0.83207	0.78683	0.78223	<b>0.87899</b>	0.87558	0.82883	0.82605
MK06	<b>0.90088</b>	0.89244	0.81180	0.80447	<b>0.90385</b>	0.89965	0.83318	0.83393	<b>0.93593</b>	0.93150	0.88334	0.87450	<b>0.95547</b>	0.94837	0.90930	0.89451
MK07	<b>0.87848</b>	0.86645	0.80350	0.79713	<b>0.88636</b>	0.88236	0.83275	0.81609	<b>0.91663</b>	0.90817	0.86181	0.83764	<b>0.94124</b>	0.92700	0.89149	0.89296
MK08	<b>0.73744</b>	0.73576	0.66604	0.66087	<b>0.78294</b>	0.76921	0.72914	0.70384	<b>0.84050</b>	0.83669	0.79622	0.77725	<b>0.88751</b>	0.88594	0.84900	0.83022
MK09	<b>0.79002</b>	0.78171	0.72513	0.70004	<b>0.84314</b>	0.83042	0.78290	0.75902	<b>0.87216</b>	0.86716	0.81417	0.80597	<b>0.92049</b>	0.91079	0.88276	0.87193
MK10	<b>0.82144</b>	0.81866	0.75636	0.73484	<b>0.85518</b>	0.84634	0.79271	0.77590	<b>0.88868</b>	0.87929	0.84019	0.82570	<b>0.92949</b>	0.92225	0.89135	0.87924

TABLE 6: Results of Kacem and Brandimarte instances with different weights under winter TOU electricity pricing.

Instance	$w_1 = 0.2, w_2 = 0.8$			$w_1 = 0.4, w_2 = 0.6$			$w_1 = 0.6, w_2 = 0.4$			$w_1 = 0.8, w_2 = 0.2$						
	MBBO	BBO	GA	HS	MBBO	BBO	GA	HS	MBBO	BBO	GA	HS				
4 × 5	<b>0.93825</b>	0.93249	0.92723	0.93565	<b>0.95975</b>	0.95420	0.94372	0.94692	<b>0.96928</b>	0.96748	0.96420	0.96337	<b>0.98114</b>	0.97885	0.97644	0.97699
8 × 8	<b>0.91808</b>	0.91243	0.88322	0.89265	<b>0.93583</b>	0.92791	0.90467	0.89213	<b>0.95461</b>	0.94782	0.90852	0.90618	<b>0.97281</b>	0.96160	0.93971	0.93765
10 × 7	<b>0.95635</b>	0.95068	0.93659	0.92331	<b>0.97084</b>	0.95902	0.94816	0.94205	<b>0.97728</b>	0.97166	0.94965	0.94136	<b>0.98565</b>	0.98311	0.96349	0.96037
10 × 10	<b>0.98182</b>	0.97360	0.94343	0.94521	<b>0.98317</b>	0.96014	0.93964	0.94321	<b>0.98747</b>	0.98285	0.95759	0.95169	<b>0.99145</b>	0.99015	0.96624	0.96705
MK01	<b>0.83601</b>	0.81750	0.75213	0.73110	<b>0.87045</b>	0.86263	0.82634	0.79158	<b>0.89462</b>	0.88823	0.86497	0.81430	<b>0.91429</b>	0.91179	0.89432	0.87099
MK02	<b>0.88878</b>	0.87627	0.80607	0.79735	<b>0.90070</b>	0.88977	0.86042	0.80796	<b>0.91964</b>	0.91424	0.87074	0.86685	<b>0.94458</b>	0.93821	0.92148	0.90128
MK03	<b>0.88101</b>	0.87083	0.79652	0.76992	<b>0.88340</b>	0.87743	0.80571	0.80287	<b>0.91823</b>	0.91086	0.85863	0.84685	<b>0.94397</b>	0.93540	0.90132	0.89178
MK04	<b>0.80461</b>	0.80128	0.74960	0.72476	<b>0.84858</b>	0.83296	0.78737	0.78548	<b>0.89718</b>	0.89266	0.85949	0.83966	<b>0.92629</b>	0.92445	0.88658	0.87222
MK05	<b>0.71831</b>	0.71714	0.65241	0.63486	<b>0.78537</b>	0.77733	0.72842	0.70727	<b>0.82014</b>	0.81486	0.77096	0.74939	<b>0.87052</b>	0.86346	0.81915	0.80327
MK06	<b>0.88797</b>	0.87402	0.78961	0.76749	<b>0.90050</b>	0.89262	0.82558	0.78820	<b>0.93130</b>	0.91797	0.87398	0.84766	<b>0.95312</b>	0.94519	0.90068	0.88636
MK07	<b>0.84989</b>	0.83900	0.77736	0.76118	<b>0.87229</b>	0.85748	0.80966	0.79809	<b>0.90345</b>	0.89931	0.85559	0.83810	<b>0.93634</b>	0.92645	0.89014	0.87108
MK08	<b>0.69513</b>	0.69281	0.62230	0.61341	<b>0.75231</b>	0.74188	0.70055	0.66463	<b>0.82136</b>	0.81752	0.77258	0.75119	<b>0.88063</b>	0.87245	0.84635	0.81949
MK09	<b>0.76261</b>	0.74886	0.67575	0.66019	<b>0.82182</b>	0.81046	0.74549	0.73576	<b>0.85869</b>	0.84712	0.80206	0.78183	<b>0.91223</b>	0.90961	0.87191	0.86346
MK10	<b>0.78907</b>	0.78807	0.72076	0.69542	<b>0.83747</b>	0.82573	0.76689	0.73723	<b>0.87701</b>	0.86502	0.81335	0.80772	<b>0.92236</b>	0.91647	0.87938	0.87632

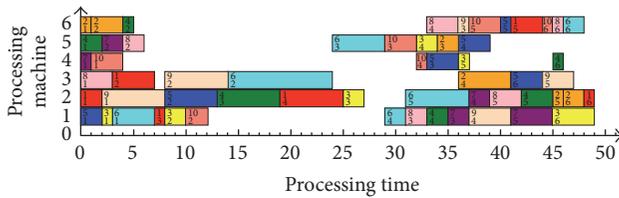


FIGURE 6: Gantt chart of an optimal solution of MK02 by MBBO algorithm under winter TOU electricity pricing.

obtained by MBBO algorithm in Instance  $10 \times 10$  when  $w_1 = 0.8$  and  $w_2 = 0.2$ .

According to the results of the performed experiments, it can be concluded that the MBBO algorithm was more effective and competitive than the basic BBO algorithm, GA, and HS for solving the proposed model under the TOU policy for both summer and winter.

## 6. Conclusion

In this study, a modified BBO algorithm integrated with VNS was proposed for solving FJSSP. In addition to the traditional objective makespan, an index addressing ECC was added. The main contributions in this study can be summarized as follows. (1) A mathematical model for the biobjective problem, to minimize makespan and ECC simultaneously, was proposed in an extended FJSSP, which considered a new subtask concerning the selection of adjustable operating speeds. (2) A modified BBO algorithm, which adopted the improved migration model combined with VNS, was applied to determine the optimal solutions effectively. (3) A series of experiments were performed on extended benchmark instances to validate that the proposed MBBO algorithm can obtain better solutions than the other comparative algorithms. From a practical point of view, this study can be applied to industry and help decision makers develop production schedules which can reduce not only time but also electricity costs. In addition, due to the conflict between time and ECC, different weight coefficient settings can affect the quality of schedules, and the experiments on different weight combinations can provide useful information for decision makers.

Because of the limitations of the proposed model and the MBBO algorithm, additional investigation can be undertaken in the future. For example, more machine statuses can be considered, for example, start-up and shut-down. Moreover, additional factors affecting electricity costs can be considered to improve the accuracy of the proposed model. Furthermore, in the practical production planning and scheduling process, many dynamic factors exist, such as machine breakdown and the insertion of jobs. Clearly, considering these factors has not only a theoretical but also a practical significance.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work has been supported by the National Natural Science Foundation of China (no. 51375429, no. 51475410) and Zhejiang Natural Science Foundation of China (no. LY17E050010).

## References

- [1] "Energy Administration Information (EIA)," *International Energy Outlook*, 2013, <http://www.eia.gov/>.
- [2] M. Dai, D. Tang, A. Giret, M. A. Salido, and W. D. Li, "Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 5, pp. 418–429, 2013.
- [3] G. Mouzon, M. B. Yildirim, and J. Twomey, "Operational methods for minimization of energy consumption of manufacturing equipment," *International Journal of Production Research*, vol. 45, no. 18-19, pp. 4247–4271, 2007.
- [4] G. Mouzon and M. B. Yildirim, "A framework to minimise total energy consumption and total tardiness on a single machine," *International Journal of Sustainable Engineering*, vol. 1, no. 2, pp. 105–116, 2008.
- [5] F. Shrouf, J. Ordieres-Meré, A. García-Sánchez, and M. Ortega-Mier, "Optimizing the production scheduling of a single machine to minimize total energy consumption costs," *Journal of Cleaner Production*, vol. 67, pp. 197–207, 2014.
- [6] A. Che, Y. Zeng, and K. Lyu, "An efficient greedy insertion heuristic for energy-conscious single machine scheduling problem under time-of-use electricity tariffs," *Journal of Cleaner Production*, vol. 129, pp. 565–577, 2016.
- [7] J.-Y. Ding, S. Song, R. Zhang, R. Chiong, and C. Wu, "Parallel machine scheduling under time-of-use electricity prices: new models and optimization approaches," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 1138–1154, 2016.
- [8] A. A. G. Bruzzone, D. Anghinolfi, M. Paolucci, and F. Tonelli, "Energy-aware scheduling for improving manufacturing process sustainability: a mathematical model for flexible flow shops," *CIRP Annals-Manufacturing Technology*, vol. 61, no. 1, pp. 459–462, 2012.
- [9] W. Lin, D. Y. Yu, C. Zhang et al., "A multi-objective teaching-learning-based optimization algorithm to scheduling in turning processes for minimizing makespan and carbon footprint," *Journal of Cleaner Production*, vol. 101, pp. 337–347, 2015.
- [10] C. Lu, L. Gao, X. Li, Q. Pan, and Q. Wang, "Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm," *Journal of Cleaner Production*, vol. 144, pp. 228–238, 2017.
- [11] Y. Liu, H. Dong, N. Lohse, and S. Petrovic, "A multi-objective genetic algorithm for optimisation of energy consumption and shop floor production performance," *International Journal of Production Economics*, vol. 179, pp. 259–272, 2016.
- [12] N. Diaz, M. Helu, A. Jarvis, S. Tönissen, D. Dornfeld, and R. Schlosser, "Strategies for minimum energy operation for precision machining," in *Proceedings of the In Proceedings of the Machine Tool Technologies Research Foundation (MTTRF2009) 2009 Annual Meeting*, pp. 47–50, Shanghai, China, 2009.

- [13] K. Fang, N. Uhan, F. Zhao, and J. W. Sutherland, "A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction," *Journal of Manufacturing Systems*, vol. 30, no. 4, pp. 234–240, 2011.
- [14] K.-T. Fang and B. M. T. Lin, "Parallel-machine scheduling to minimize tardiness penalty and power cost," *Computers and Industrial Engineering*, vol. 64, no. 1, pp. 224–234, 2013.
- [15] H. Luo, B. Du, G. Q. Huang, H. Chen, and X. Li, "Hybrid flow shop scheduling considering machine electricity consumption cost," *International Journal of Production Economics*, vol. 146, no. 2, pp. 423–439, 2013.
- [16] A. Sharma, F. Zhao, and J. W. Sutherland, "Econological scheduling of a manufacturing enterprise operating under a time-of-use electricity tariff," *Journal of Cleaner Production*, vol. 108, pp. 256–270, 2015.
- [17] M. A. Salido, J. Escamilla, A. Giret, and F. Barber, "A genetic algorithm for energy-efficiency in job-shop scheduling," *The International Journal of Advanced Manufacturing Technology*, vol. 85, no. 5-8, pp. 1303–1314, 2016.
- [18] J.-Y. Moon and J. Park, "Smart production scheduling with time-dependent and machine-dependent electricity cost by considering distributed energy resources and energy storage," *International Journal of Production Research*, vol. 52, no. 13, pp. 3922–3939, 2014.
- [19] C. Zhang, P. Gu, and P. Jiang, "Low-carbon scheduling and estimating for a flexible job shop based on carbon footprint and carbon efficiency of multi-job processing," *Journal of Engineering Manufacture*, vol. 229, no. 2, pp. 328–342, 2015.
- [20] Y. He, Y. Li, T. Wu, and J. W. Sutherland, "An energy-responsive optimization method for machine tool selection and operation sequence in flexible machining job shops," *Journal of Cleaner Production*, vol. 87, no. C, pp. 245–254, 2015.
- [21] Y. Liu and A. Tiwari, "An investigation into minimising total energy consumption and total completion time in a flexible job shop for recycling carbon fiber reinforced polymer," *Procedia CIRP*, vol. 29, pp. 722–727, 2015.
- [22] X. Yang, Z. Zeng, R. Wang, and X. Sun, "Bi-objective flexible job-shop scheduling problem considering energy consumption under stochastic processing times," *PLoS ONE*, vol. 11, no. 12, article e0167427, 2016.
- [23] D. Lei, Y. Zheng, and X. Guo, "A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption," *International Journal of Production Research*, vol. 55, no. 11, pp. 3126–3140, 2016.
- [24] L. Yin, X. Li, L. Gao, C. Lu, and Z. Zhang, "A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem," *Sustainable Computing: Informatics and Systems*, vol. 13, pp. 15–30, 2017.
- [25] W. Y. Zhang, S. Zhang, M. Cai, and J. X. Huang, "A new manufacturing resource allocation method for supply chain optimization using extended genetic algorithm," *International Journal of Advanced Manufacturing Technology*, vol. 53, no. 9-12, pp. 1247–1260, 2011.
- [26] W. Y. Zhang, S. Zhang, S. Guo, Y. Yang, and Y. Chen, "Concurrent optimal allocation of distributed manufacturing resources using extended teaching-learning-based optimization," *International Journal of Production Research*, vol. 55, no. 3, pp. 718–735, 2017.
- [27] Q.-K. Pan, L. Wang, and L. Gao, "A chaotic harmony search algorithm for the flow shop scheduling problem with limited buffers," *Applied Soft Computing Journal*, vol. 11, no. 8, pp. 5270–5280, 2011.
- [28] I. Ribas, R. Companys, and X. Tort-Martorell, "An efficient Discrete Artificial Bee Colony algorithm for the blocking flow shop problem with total flowtime minimization," *Expert Systems with Applications*, vol. 42, no. 15-16, pp. 6155–6167, 2015.
- [29] S. Karthikeyan, P. Asokan, S. Nickolas, and T. Page, "A hybrid discrete firefly algorithm for solving multi-objective flexible job shop scheduling problems," *International Journal of Bio-Inspired Computation*, vol. 7, no. 6, pp. 386–401, 2015.
- [30] X. Li and L. Gao, "An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem," *International Journal of Production Economics*, vol. 174, pp. 93–110, 2016.
- [31] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [32] S. H. A. Rahmati and M. Zandieh, "A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem," *International Journal of Advanced Manufacturing Technology*, vol. 58, no. 9–12, pp. 1115–1129, 2012.
- [33] X. Wang and H. Duan, "A hybrid biogeography-based optimization algorithm for job shop scheduling problem," *Computers and Industrial Engineering*, vol. 73, no. 1, pp. 96–114, 2014.
- [34] J. Lin and S. Zhang, "An effective hybrid biogeography-based optimization algorithm for the distributed assembly permutation flow-shop scheduling problem," *Computers and Industrial Engineering*, vol. 97, pp. 128–136, 2016.
- [35] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for Web services composition," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, 2004.
- [36] H. Ma, "An analysis of the equilibrium of migration models for biogeography-based optimization," *Information Sciences*, vol. 180, no. 18, pp. 3444–3464, 2010.
- [37] N. Mladenović and P. Hansen, "Variable neighborhood search," *Computers and Operations Research*, vol. 24, no. 11, pp. 1097–1100, 1997.
- [38] E. Nowicki and C. Smutnicki, "A fast taboo search algorithm for the job shop problem," *Management Science*, vol. 42, no. 6, pp. 797–813, 1996.
- [39] E. Balas, "Machine sequencing via disjunctive graphs: an implicit enumeration algorithm," *Operations Research*, vol. 17, no. 6, pp. 941–957, 1969.
- [40] H. Chen, J. Ihlow, and C. Lehmann, "A genetic algorithm for flexible job-shop scheduling," in *In Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1120–1125, 1999.
- [41] L. Wang, Y. Xu, Y. Mao, and M. Fei, "A discrete harmony search algorithm," *Life System Modeling and Intelligent Computing*, vol. 98, no. 2, pp. 37–43, 2010.
- [42] I. Kacem, S. Hammadi, and P. Borne, "Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic," *Mathematics and Computers in Simulation*, vol. 60, no. 3-5, pp. 245–276, 2002.
- [43] P. Brandimarte, "Routing and scheduling in a flexible job shop by tabu search," *Annals of Operations Research*, vol. 41, no. 3, pp. 157–183, 1993.



# Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

