Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 2573873, 18 pages
https://doi.org/10.1155/2019/2573873

Research Article

Multi-Objective Reentrant Hybrid Flowshop Scheduling with Machines Turning on and off Control Strategy Using Improved Multi-Verse Optimizer Algorithm

Kaifeng Geng,[1,2] **Chunming Ye**,[1] **Lei Cao**,[1] **and Li Liu**[2]

[1]*School of Business, University of Shanghai for Science and Technology, Shanghai 200093, China*
[2]*Information Construction and Management Center, Nanyang Institute of Technology, Nanyang, Henan 473004, China*

Correspondence should be addressed to Chunming Ye; yechm@usst.edu.cn

Received 13 April 2019; Accepted 4 June 2019; Published 27 June 2019

Academic Editor: Oliver Schütze

This paper focuses on the multi-objective optimization of the reentrant hybrid flowshop scheduling problem (RHFSP) with machines turning on and off control strategy. RHFSP exhibits significance in many industrial applications, but scheduling with both energy consumption consideration and reentrant concept is relatively unexplored at present. In this study, an improved Multi-Objective Multi-Verse Optimizer (IMOMVO) algorithm is proposed to optimize the RHFSP with objectives of makespan, maximum tardiness, and idle energy consumption. To solve the proposed model more effectively, a series of improved operations are carried out, including population initialization based on Latin hypercube sampling (LHS), individual position updating based on Lévy flight, and chaotic local search based on logical self-mapping. In addition, a right-shift procedure is used to adjust the start time of operations aiming to minimize the idle energy consumption without changing the makespan. Then, Taguchi method is utilized to study the influence of different parameter settings on the scheduling results of the IMOMVO algorithm. Finally, the performance of the proposed IMOMVO algorithm is evaluated by comparing it with MOMVO, MOPSO, MOALO, and NSGA-II on the same benchmark set. The results show that IMOMVO algorithm can solve the RHFSP with machines turning on and off control strategy effectively, and in terms of convergence and diversity of non-dominated solutions, IMOMVO is obviously superior to other algorithms. However, the distribution level of the five algorithms has little difference. Meanwhile, by turning on and off the machine properly, the useless energy consumption in the production process can be reduced effectively.

1. Introduction

Hybrid flowshop scheduling problem usually involves several stages, each of which contains a certain number of parallel machines and each job passes through all the stages only once in sequence. However, in some special industries, a job needs to access some stages more than once, such as semiconductor wafer manufacturing and TFT-LCD (thin film transistor liquid crystal display) panel manufacturing. RHFSP has been proved to be NP-hard [1], making it difficult to be solved by traditional methods. Therefore, it is of great theoretical and practical value to carry out research on the efficient intelligent optimization algorithm for this problem.

Since Graves and Meal et al. [2] first studied the reentrant scheduling problem in 1983, great progress has been made. Although real-world RHFSP is multi-objective by nature, many researches focus on the RHFSP with single-objective. Bertel and Billaut [3] studied the RHFSP with parallel machines and proposed a genetic algorithm to minimize the weighted number of delayed jobs. Pearn and Chung et al. [4] investigated the integrated circuits final test scheduling problem with reentrant and proposed three network algorithms to minimize the total machine load. Choi and Kim [5] proposed several improved heuristic algorithms, such as NEH and SO, to minimize the maximum completion time and compared them with the SA algorithm. The experimental results showed that the heuristic algorithms can produce better solutions in the relatively short time. Kim and Lee [6] studied the RHFSP of uncorrelated parallel machines at each stage and proposed CDS and NEH heuristic algorithms to minimize the

maximum completion time under certain delay constraints. Choi and Kim et al.[7] considered the two-stage RHFSP and minimized the completion time using the branch and bound algorithm under the constraint of maximum allowable due date. El-Khouly and El-Kilany et al. [8] proposed an optimization method for reentrant production workshop using Lagrangian decomposition with the objective of minimizing the total delay. Hekmatfar and Fatemi Ghomi et al. [9] proposed some heuristic algorithms and random key genetic algorithms (RKGA) with the objective of minimizing the maximum completion time and compared them with the hybrid genetic algorithm (HGA). Experimental results showed that the HGA was superior to other algorithms. Lin and Lee et al. [10] combined the analytical hierarchy process (AHP) and the genetic algorithm to deal with multi-objective RHFSP and applied them in a repair shop. The results showed that it is better than manual scheduling. Chen and Pan et al. [11] applied the hybrid tabu search algorithm to study RHFSP with the objectives of minimizing the maximum completion time and compared the mixed tabu search algorithm with integer programming techniques. Wu and Liu et al. [12] studied the reentrant scheduling problem considering learning effects with the objective of minimizing the maximum completion time.

In recent years, some scholars have studied the multi-objective RHFSP. Cho and Bae et al. [13] proposed a Pareto genetic algorithm to solve the biobjective RHFSP, which included the crossover operation based on Minkowski distance and local search strategy with the objectives of minimizing the maximum completion time and total delay time. Choi and Kim et al. [14] studied the real-time dynamic RHFSP with multiple optimization objectives, including maximum system output, minimum average running time, minimum average delay time, and minimum number of total lost jobs using a real-time scheduling mechanism based on decision tree which is applied to a real TFT-LCD panel production line at last. Ying and Lin et al. [15] proposed an efficient iterative Pareto greedy (IPG) algorithm based on the research in [13]. Shen and Wang et al. [16, 17] proposed an improved teaching and learning optimization algorithm and a Pareto discrete harmony search algorithm to solve the biobjective RHFSP.

At present, energy saving plays an increasingly significant role in manufacturing industries, especially energy-intensive industries. Optimizing production schedule helps to reduce unnecessary energy consumption. Luo and Du et al. [18] proposed a novel ant colony optimization algorithm for the hybrid flowshop scheduling problem considering energy consumption cost. Mansouri and Aktas et al. [19] solved the flowshop scheduling problem considering maximum completion time and total energy consumption by mixed integer multi-objective programming model and heuristic algorithm. Lei and Zheng et al. [20] proposed a hybrid frog leaping algorithm to solve the flexible flowshop scheduling problem considering energy consumption with the objective of minimizing the workload balance and total energy consumption. Wang et al. [21] used the NSGA-II algorithm to solve the problem of identical parallel machine scheduling with the objectives of reducing total energy consumption and makespan. Liu and Guo et al. [22] addressed a novel integrated green scheduling problem of flexible job shop and crane transportation to reduce the comprehensive energy consumption of the machining process and transportation process using a mixed integer programming model. Yildirim and Mouzon [23] proposed a mathematical model to minimize energy consumption and total completion time of a single machine, furthermore, introducing the dominance rules and a heuristic to increase the speed of the proposed multi-objective genetic algorithm. In [24], the single-machine scheduling problem with the objective of minimizing the energy cost under the change of energy price was studied; experiments show that energy consumption can be reduced by using the machine turning on and off control strategy.

In the current context of rising energy prices and increasingly stringent environmental concerns, it is particularly important to reduce energy cost and achieve green and sustainable development for the manufacturing industry. According to the latest data released by the State Energy Administration and China Telecom Federation, the energy consumption of manufacturing industry accounts for about one-third of the total energy consumption. However, both purchasing more energy-efficient equipment and building more energy-efficient production lines require huge financial investments. The benefits could not be easily enjoyed by most of manufacturing companies, especially those small and medium sized enterprises. In practice, it was observed that in an eight-hour shift, the bottleneck machines still stay idle 16% of the time on average. If the machines are turned off during the idle periods, 13% energy will be saved [25]. So it is significant for saving energy to turn off machines when they are idle for a certain amount of time. However, the research on energy consumption of RHFSP only focuses on the total energy consumption or idle energy consumption and seldom combines the scheduling problem with the machine turn on and off strategy. A scheduling model considering the machine turning on and off control strategy is proposed in this paper. It breaks the assumption that machines will not stop until all the jobs have been processed in RHFSP and reduces the useless energy consumption.

The remainder of this paper is organized as follows: the RHFSP is described with the makespan, max tardiness, and idle energy consumption objectives in Section 2. In Section 3, the proposed IMOMVO algorithm is introduced in detail, using an existing benchmark to present the procedure of the encoding and PS decoding. The influence of parameter settings on the IMOMVO algorithm is investigated and comparisons with other algorithms for simulated data are shown in Section 4. Finally, in Section 5, some conclusions and future work are discussed.

2. Problem Description

The RHFSP addressed in this study can be described as follows [17, 26, 27]: n jobs need to be processed in s stages in sequence. There are m_i identical parallel machines in the stage i, and each job can be processed on any machine in the corresponding stage. Due to the workflow requirement,

a job may access some stages more than once. The objective is to determine the allocation of jobs to parallel machines at each stage, as well as the sequence of the jobs assigned to each machine in order to find the no-dominated Pareto set. Using the three-field notation introduced by Pinedo [28], the RHFSP is expressed as $FFC\ |recrc|\ C_{max}, T_{max}, SEC_{min}$.

Additionally, the following assumptions are made: all the jobs and machines are ready at zero time; at any time, each machine can process at most one job, and each job can only be processed by at most one machine; all the jobs do not affect each other; the number of re-entrance of each job, the processing time of each operation, and the production route of each job through the shop are known in advance; the buffer capacity between any two consecutive processing stages is infinite; preemption is not allowed, and once the job is processed, it cannot be interrupted; the machine failure and the machine adjustment time are not considered.

2.1. Symbol Definition. The mathematical symbols involved in the model and their meanings are as follows:

n: number of jobs

j: index for jobs, $j = 1, 2, \ldots, n$

s: number of stages

i: index for stages, $i = 1, 2, \ldots, s$

m_i: number of identical parallel machines in stage i

l: index for machines in stage i, $l = 1, 2, \ldots, m_i$

N_j: number of operations for job j

k: index of operations for job j, $k = 1, 2, \ldots, N_j$

O_{jk}: the $k-th$ operation of job j

p_{jk}: processing time of O_{jk}

M: a large number

U_i: set of operations that are processed in stage i

S_{jk}: starting time of O_{jk}

d_j: due date of job j

C_j: completion time of job j

PW: energy consumption of machines per unit time at processing jobs time

PI: energy consumption of machines per unit time at idle time

$Tsetup$: turning on and off time of machines

$Setup\cos t$: energy consumption of machines at the turning on and off time

TB: the critical time to choose to shutdown machines

M_{total}: number of machines in all stages, $M_{total} = m_1 + m_2 + \ldots + m_s$

q: index of machines, $q = 1, 2, \ldots, M_{total}$; index of machine

R_q: number of operations on machine q

B_{qr}: starting time of $r-th$ operation on machine q

E_{qr}: ending time of $r-th$ operation on machine q

F_{qr} = {1, if turns off the machine after an operation; 0, otherwise}

r_{ijkl} = {1, if O_{jk} is processed on the machine l in stage i; 0, otherwise} $\forall i, j, k, l$

$Z_{jkj'k'}$ = {1, if O_{jk} procedes $O_{j'k'}$; 0, otherwise} $\forall j < j', k, k'$.

2.2. Machine Turning On and Off Control Strategy. It is generally assumed in RHFSP that the machine stops until all the jobs have been processed. Inevitably, machines will be idle during the waiting time. Kordonowy [29] analyzed the energy consumption of machine tools. It is observed that only 19.2%-65.8% of the energy is consumed in machining time. Gutowski [30] found that most of the energy is consumed while the machine is idle in the Toyota Motor Corporation. Turning off the machine in idle time can save this part of energy consumption. However, it will take considerable time and energy to turn on and off machines. So we need to switch the machine at the right time. In general, energy consumption behaviors can be conceived as cyclic processes. Take turning shop as an example, the cyclic power graph is shown as Figure 1 [31].

From Figure 1, it can be seen that energy consumption cycle can be obviously simplified into two phase. One is the idle time which is waiting for tasks, and another is the processing time which is working on tasks. Energy consumption depends on the working state of machines. It is assumed that the energy consumption per unit time is constant when the machine is idle. When the machine is in shutdown state, no energy consumption will be generated. The machine state transition diagram for an energy consumption cycle is shown in Figure 2. The text in brackets on the arc represents the transition time and energy consumption required for the transition between different states. Machine turning on and off decision variables are used to judge the state of a machine after completion of machining. We need to determine after which operations the machines should be turned off.

First of all, if turns off the machine after an operation, the conversion time should be less than the interval between the start processing time of the next job and the completion time of it, that is to say, the constraint (1).

$$B_{qr+1} - E_{qr} \geq Tsetup, \quad q = 1, 2, \ldots, M_{total},\ r = 1, 2, \ldots, R_q \tag{1}$$

Secondly, the energy consumption of starting machine is greater than that of idling state. When choosing to shut down the machine, the interval between adjacent jobs should be greater than a critical time length, that is, the constraint (2) should be satisfied.

$$B_{qr+1} - E_{qr} \geq \frac{Setup\cos t}{PI}, \quad q = 1, 2, \ldots, M_{total},\ r = 1, 2, \ldots, R_q \tag{2}$$

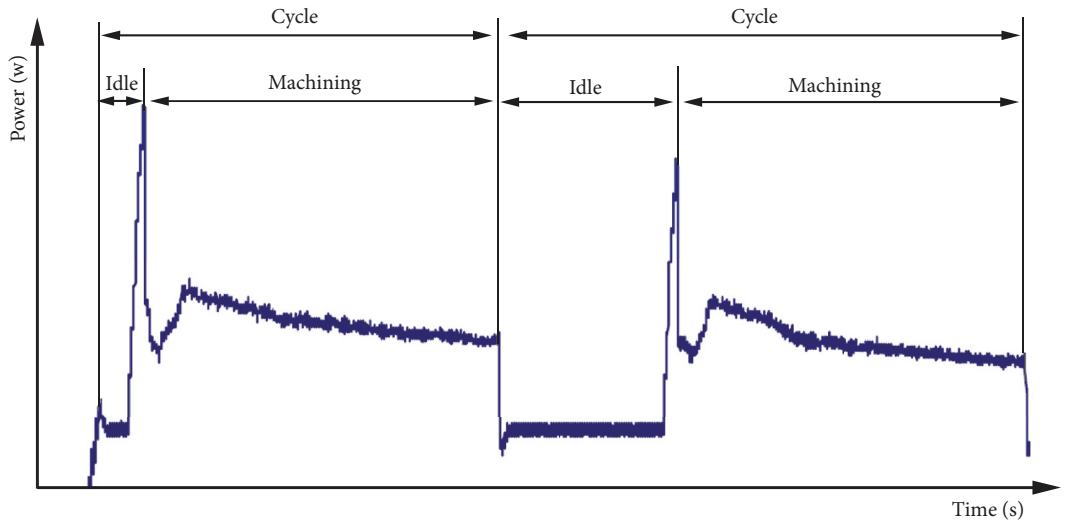

Figure 1: Power graph for turning.

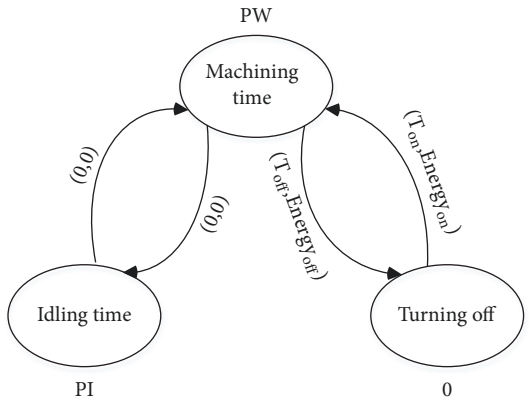

Figure 2: Machine state transition diagram.

Supposing $TB = \max(Setup\cos t/PI, Tsetup)$, the above two constraints can be combined into the following form (3):

$$B_{qr+1} - E_{qr} \geq TB, \quad q = 1, 2, \ldots, M_{total},\ r = 1, 2, \ldots, R_q \tag{3}$$

2.3. Mathematical Programming Model. Based on the existing literature [14, 15, 32], a mixed integer programming model for three-objective of *RHFSP* is presented, C_{max} denotes the makespan, T_{max} denotes the maximum tardiness time, and SEC_{min} denotes the idle energy consumption.

Objective Functions

$$f_1 = C_{\max} = \max C_j \tag{4}$$

$$f_2 = T_{\max} = \max\left(\max\left(0, C_j - d_j\right)\right) \tag{5}$$

$$\begin{aligned} f_3 &= SEC_{\min} \\ &= \sum_{q=1}^{Mtotal} \sum_{r=1}^{R_q - 1} F_{qr} Setup\cos t \\ &\quad + \sum_{q=1}^{Mtotal} \sum_{r=1}^{R_q - 1} \left(B_{qr+1} - E_{qr}\right)\left(1 - F_{qr}\right) PI \end{aligned} \tag{6}$$

In f_3, because of the existence of product term $(B_{qr+1} - E_{qr})(1 - F_{qr})$, the model is a more complex nonlinear model, so it needs to be transformed into a linear model.

$$h_{qr} = \begin{cases} B_{qr+1} - E_{qr} & if\ F_{qr} = 1 \\ 0 & if\ F_{qr} = 0 \end{cases} \tag{7}$$

Plugging $h_{qr} = (B_{qr+1} - E_{qr})F_{qr}$ into formula (6), then the nonlinear term is removed and the model is transformed into a linear form as formula (8):

$$\begin{aligned} f_3 &= SEC_{\min} \\ &= \sum_{q=1}^{Mtotal} \sum_{r=1}^{R_q - 1} F_{qr} Setup\cos t \\ &\quad + \sum_{q=1}^{Mtotal} \sum_{r=1}^{R_q - 1} \left(B_{qr+1} - E_{qr} - h_{qr}\right) PI \end{aligned} \tag{8}$$

Subject to

$$\sum_{l=1}^{m_i} r_{ijkl}\left(S_{jk} + p_{jk}\right) \leq \sum_{l=1}^{m_{i'}} r_{i'jk+1l} S_{jk+1} \quad \forall i, i', j, k \tag{9}$$

$$\sum_{l=1}^{m_i} r_{ijkl} = 1 \quad \forall i, j, O_{jk} \in U_i \tag{10}$$

$$M\left(2 - r_{ijkl} - r_{ij'k'l}\right) + M\left(1 - Z_{jkj'k'}\right) + \left(S_{j'k'} - S_{jk}\right) \geq p_{jk} \quad \forall i, j < j', l, O_{jk} \in U_i,\ O_{j'k'} \in U_i \tag{11}$$

$$M\left(2 - r_{ijkl} - r_{ij'k'l}\right) + MZ_{jkj'k'} + \left(S_{jk} - S_{j'k'}\right) \geq p_{j'k'} \quad \forall i, j < j', l, O_{jk} \in U_i,\ O_{j'k'} \in U_i \tag{12}$$

$$M\left(2 - r_{ijkl} - r_{ijk'l}\right) + \left(S_{jk'} - S_{jk}\right) \geq p_{jk} \quad \forall i, k < k', l, O_{jk} \in U_i,\ O_{jk'} \in U_i \tag{13}$$

$$S_{jk} \geq 0 \quad \forall j, k \tag{14}$$

$$C_j = \sum_{i=1}^{s}\sum_{l=1}^{m_i} r_{ijN_jl}\left(S_{jN_j} + P_{jN_j}\right) \quad \forall j \tag{15}$$

$$C_j \leq C_{\max} \quad \forall_j \tag{16}$$

Equations (4), (5), and (8) represent the three-objective functions of the optimization problem: makespan, maximum tardiness, and idle energy consumption (turning on and off energy consumption and idle energy consumption). Constraint (9) ensures that the starting time of operation O_{jk+1} is not earlier than the end time of operation O_{jk}. Constraint (10) ensures that each operation can be processed on only one machine in the required stage. Constraints (11), (12), and (13) ensure that each machine processes at most one operation at a time; Equation (14) is the nonnegative restriction for S_{jk} and constraints (15) and (16) define the maximum completion time target.

In a three-objective optimization problem, according to the definition of dominance [33], solution X_1 dominates $X_2(X_1 \prec X_2)$, if and only if $\forall i \in \{1, 2, 3\},\ f_i(X_1) \leq f_i(X_2) \wedge \exists i \in \{1, 2, 3\},\ f_i(X_1) < f_i(X_2)$. A solution X is the Pareto optimal solution (or nondominant solution) if and only if there is no solution that can dominate it. For the above RHFSP, it aims to obtain the Pareto optimal solutions or the Pareto front.

3. The Proposed IMOMVO Algorithm

Although many intelligent optimization algorithms have been used to solve the RHFSP, "No Free Lunch Theorems" [34] indicated that no algorithm can solve all the optimization problems. One algorithm is only targeted at certain specific problems and not necessarily effective for other problems.

The Multi-Verse Optimizer (MVO) [35] is a novel heuristic algorithm proposed by Professor Mirjalili in 2015, which divides the search process into two aspects: exploration and development. There is always high possibility of moving objects from a universe with high inflation rate to a universe with low inflation rate. In order to maintain the diversity of universes and perform exploitation, each universe has wormholes to transport its objects through space randomly. In 2016 Mirjalili et al. [36] proposed a Multi-objective Multi-verse Optimizer (MOMVO) algorithm. The search mechanism of MOMVO is very similar to MVO, but external archive is introduced to store the best non-dominated solutions so far. In order to further improve the quality of the solutions, the MOMVO algorithm was improved from three aspects: (1) In many algorithms, the initial population is generated in a completely random manner. Since the number of populations is much smaller than the solution space, random solutions may be concentrated in a certain local region, which is not conducive to expanding the search space and converging towards the global optimal solution. This paper adopted the Latin hypercube sampling technology to initialize the population. (2) Lévy flight was introduced into MOMVO. Both short-distance and long-distance movement are used to update the individual position, which can improve the global search ability of the algorithm. (3) Selecting partial solutions from external archives and updating the individual positon using logical self-mapping can improve the local search ability of the algorithm.

3.1. Coding and Decoding. MVO algorithm is mainly used to solve optimization problems of continuous functions, so it cannot be directly used to deal with discrete optimization problems. In this paper, we adopt random key coding. Each gene is represented by random numbers within $[0, 1]$. To get a feasible schedule, a reasonable decoding method is required to determine the processing sequence of all jobs at each stage and arrange suitable machines for the operations. Then the start time and completion time of all operations can be calculated. At last the three objectives can be obtained. In this study, we decode it according to PS method in reference [13], generate scheduling schemes according to various constraints, and ensure that the arbitrary arrangement of individuals is always a feasible scheduling. With the PS method, all jobs are allocated according to the job permutation. When it allocates a job, all its operations are assigned to the machines that can complete the operations as early as possible.

Take the RHFSP with 4 jobs and 3 stages as an example, in which the number of identical parallel machines at each stage is 3, 2, and 2, respectively. It is assumed that all the following parameters of the machines are the same, the working energy consumption per unit time PW=8, the idling energy consumption per unit time PI=2, the turning on and off cost of the machine *Setupcost*=10, the turning on and off time of the machine *Tsetup*=2. In addition, the due date of each job and the processing times are shown in Table 1.

If the processing time of one operation p_{jk}=0, it is indicated that the job is not processed in the stage of a certain pass. It can be seen from Table 1 that jobs 1, 2, and 4 are involved in two re-entrances, and job 3 is involved in one re-entrance. For job 2, its processing sequence is stage 1 ⟶ stage 2 ⟶ stage 2 ⟶ stage 3 ⟶ stage 2 ⟶ stage 3. Using the random key coding, the individual length is equal to the number of jobs, and each element is arbitrarily selected within $[0, 1]$ as shown in Table 2. It is hypothesized that an individual position vector is [0.6555, 0.3922, 0.7431, 0.1712], then the corresponding job sequence is $[4, 2, 1, 3]$ by ascending the elements of each vector, which means the processing order of the jobs is 4-2-1-3. The PS method was used to decode the job sequence into a feasible schedule. The

Table 1: The processing times of the four-job and three-stage example.

Processing time (p_{jk})	Operations (first)			Reentrant 1			Reentrant 2			Due Date
	Stage 1	Stage 2	Stage 3	Stage 1	Stage 2	Stage 3	Stage 1	Stage 2	Stage 3	
Job 1	2	2	1	0	2	1	0	1	2	13.2
Job 2	1	3	0	0	2	1	0	2	1	14.5
Job 3	3	1	2	2	1	0	0	0	0	8.6
Job 4	2	1	1	2	0	0	3	1	0	11.7

Table 2: Random key coding.

k	1	2	3	4
individual	0. 6555	0. 3922	0. 7431	0. 1712
ascending order	0.1712	0. 3922	0. 6555	0. 7431
job sequence	4	2	1	3

Gantt charts are shown in Figure 3. Firstly, all the operations of job 4 are arranged on the machines that can process it at the earliest. Then, the operations of job 2 are arranged on the machines that can process them as early as possible. If the completion time of a certain operation of job 2 is smaller than the start time of the arranged operation of job 4 on a certain machine, the operation of job 2 will be arranged before it; otherwise, it will be arranged behind it. Next, following the same procedure as before, all the operations of jobs 1 and 3 are arranged on the machines. At last, it calculates the objective values with C_{max}=13, T_{max}= 3.4, and SEC_{min}=18.

3.2. Right-Shift Procedure. In this study, we propose a right-shift procedure to further improve the quality of the solutions. On the premise of not changing the job sequence, the procedure can reduce the SEC_{min} and minimize the number of idling without affecting the makespan. Since the adjustment of the latter job will affect the adjustable range of the former job, the order of adjustment should be from back to front. The specific steps are as follows.

(1) Without considering the energy consumption cost of the machine, all the jobs will be processed as early as possible and then calculating the start and end time of each operation to generate a schedule sequence.

(2) According to the schedule sequence, the start time of each operation on each machine is adjusted from back to front. The adjustment diagram is as shown in Figure 4, in which A and B are the jobs corresponding to two operations on the machine. S_A and S_B are the start time of the corresponding operation, $S_A{}'$ is the start time of the next operation of job A, E_A, E_B, and $E_A{}'$ are the completion time of the corresponding operation, and t_1, t_2, and t_3 are the corresponding processing time. If S_B is not greater than $S_A{}'$, E_A=S_B. Otherwise, it does not need to be adjusted.

(3) Adjust the start time and completion time of all operations according to (2).

3.3. Population Initialization Based on Latin Hypercube Sampling. When the optimal solution space cannot be predicted, the solution space feature of the initial population can maximize the information of all the individuals within a limited number. Therefore, the distribution of the initial population seriously affects the convergence performance of the algorithm. In this paper, the Latin hypercube sampling technique is used to initialize the population. Assuming that m samples are extracted in an n-dimensional vector space, the steps for Latin hypercube sampling are as follows.

(1) Divide each dimension into m intervals that do not overlap so that each interval has the same probability.

(2) Randomly extract a point in each interval of each dimension.

(3) Randomly extract the points selected in (2) from each dimension and form them into vectors.

In case of 2-dimensional vector and 100 samples, the distribution of the initial population constructed by completely random and Latin hypercube sampling is shown in Figure 5. The distribution of samples constructed by Latin hypercube sampling is more uniform.

3.4. Updated Positions of Individuals Based on Lévy Flight. The Lévy distribution was a probability distribution model proposed by the famous French scientist Paul Pierre Lévy in the 1930s. Lévy flight is a random search method following the Lévy distribution. It usually moves in short distance and occasionally in long distance, so as to avoid the repeated movement in one place. The diagram is shown in Figure 6. Updating the individual position by Lévy flight can expand the search space, increase the diversity of the population, and avoid falling into local optimum in a certain range.

In this paper, the Mantegna algorithm was used to simulate the Lévy distribution. The specific principles are as follows [37]:

$$x_{t+1} = x_t + \alpha \otimes Levy\left(\beta\right) \tag{17}$$

where x_t and x_{t+1} are the current position and updated position of the individual, α is the step scaling factor, $\otimes$ is the point multiplication operation, the step size factor β of the Lévy flight ranges within [1, 2], and $Levy(\beta)$ is a Lévy random path.

$$Levy \sim u = t^{-\lambda}, \quad 1 < \lambda \leq 3 \tag{18}$$

Generating random step size according to Lévy distribution, $s = \mu/|\nu|^{1/\beta}$, where μ and v follow normal distribution, $\mu \sim N(0, \sigma_\mu^2)$ and $\nu \sim N(0, 1)$, where

$$\sigma_\mu = \left\{ \frac{\Gamma\left(1+\beta\right)\sin\left(\pi\beta/2\right)}{\beta\Gamma\left(\left(1+\beta\right)/2\right)2^{(\beta-1)/2}} \right\}^{1/\beta} \tag{19}$$

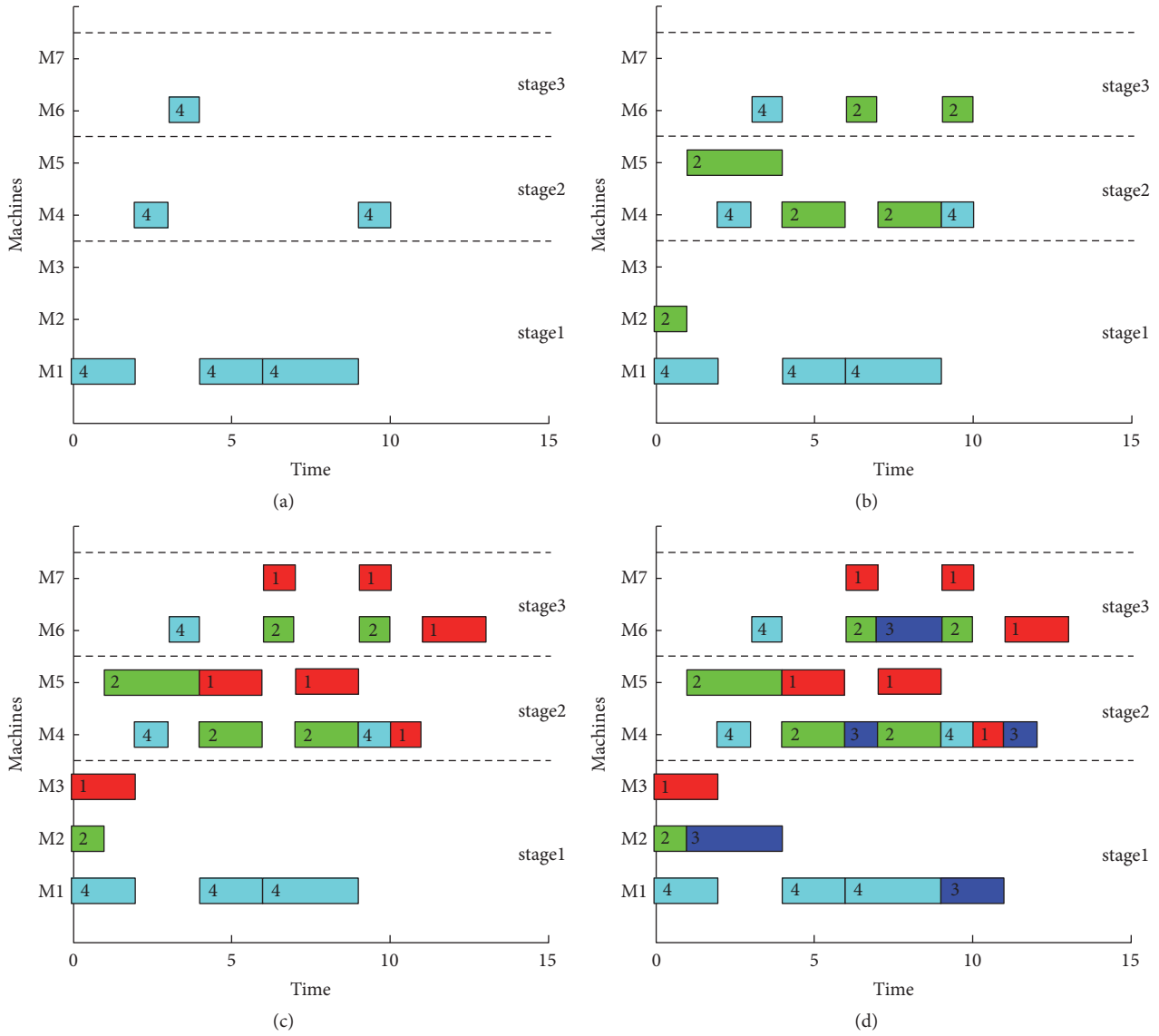

FIGURE 3: Gantt charts of the four-job and three-stage example. (a) Scheduling Job 4. (b) Scheduling Job 2. (c) Scheduling Job 1. (d) Scheduling Job 3.

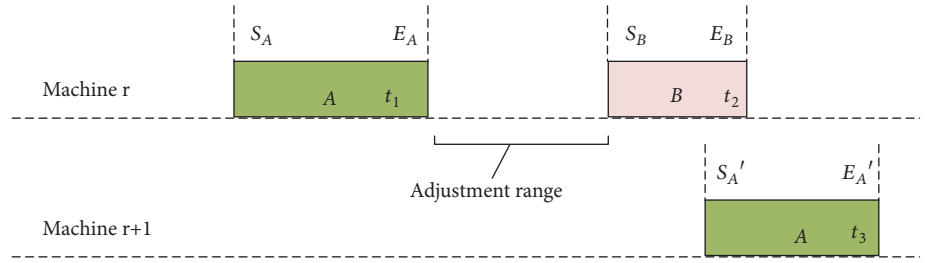

FIGURE 4: Schematic diagram of right-shift procedure.

3.5. Chaotic Local Search of Logical Self-Mapping. Introducing logical self-mapping into MOMVO algorithms, 20% of individuals in the external archives are randomly selected as elites. Then, chaotic optimization algorithm is used to search within the neighborhood of the elites, and the search space is gradually narrowed as the iteration progresses. If a better solution is detected, it will replace the solution in the external archives. If the search does not produce a better solution, it jumps to the next elite until the traversal is completed. In this paper, the logical self-mapping function is

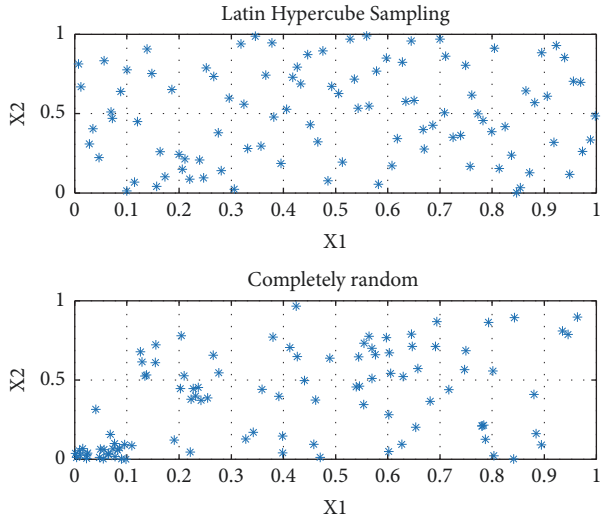

Figure 5: Distribution of complete random and Latin hypercube sampling.

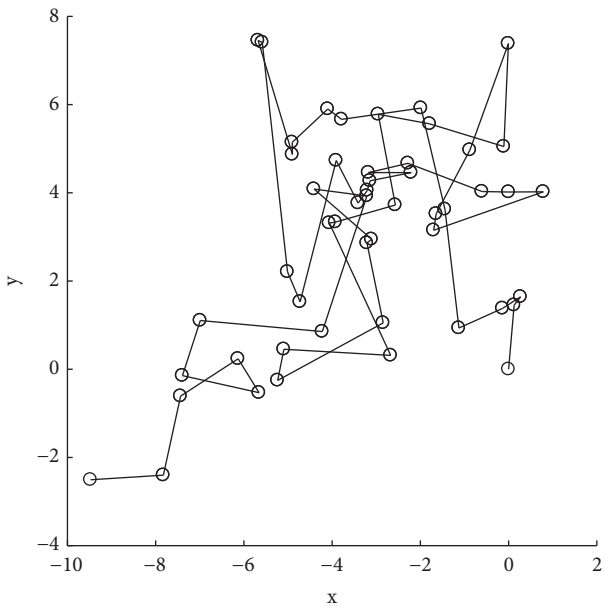

Figure 6: Lévy flight diagram.

adopted to generate the chaotic sequences. The mathematical expressions are as follows:

$$t_{i+1,d} = 1 - 2 * {t_{i,d}}^2, \quad t_{i,d} \in (0,1) \tag{20}$$

(1) Scale the elites X_i according to formula (21).

$$t_{i,d} = \frac{2\left(x_{i,d} - lb_d\right)}{\left(ub_d - lb_d\right)} - 1, \quad j = 1,2,\ldots,n \; i = 1,2,\ldots,D \tag{21}$$

(2) Substitute the scaled D-dimensional variables into logical self-mapping function to generate new D-dimensional variables. Then, transform the generated chaotic variables into the solution space X_i according to formula (22).

$$X_i' = \frac{1}{2}\left(ub_d - lb_d\right) . * t_i + \frac{1}{2}\left(ub_d + lb_d\right), \quad i = 1,2,\ldots n \tag{22}$$

(3) Calculate the function value, if $X_i' \prec X_i$, then $X_i = X_i'$ and update t_i according to formula (20); repeat (1) (2) until the maximum number of iterations is reached.

3.6. The Algorithm Flow of IMOMVO. MOMVO and MVO have similar search mechanism using white holes, black holes, and wormholes to improve the solutions. A leader selection mechanism is employed to select solutions from the archives and create tunnels between solutions. Specifically, the crowding distance between each solution in the archives is first selected, and the number of solutions in the neighborhood is counted as a measure of coverage or diversity in the approach. Then, to improve the distribution of solutions in the archives across all objectives, a roulette wheel from the less populated regions of the archives is applied to select solutions. The pseudocode of the IMOMVO algorithm is as shown in Algorithm 1.

4. Computational Results

In order to validate the effectiveness of the IMOMVO algorithm, four multi-objective optimization algorithms, including MOMVO, MOPSO, MOALO, and NSGA-II, are selected for comparative study. The simulation environment is windows 7, Intel Core i7-4770 cpu@3.40GHz, 8G memory. The algorithm is programmed by MATLAB R2017a.

4.1. Test Problems. The experiments were conducted on the benchmark set randomly generated by Cho et al. [13], which consists of 120 small-sized test problems and 120 large-sized test problems. For small-sized problems, the number of jobs is within [10, 20], the number of stages is within [5, 10], the number of re-entrance is within [1, 2], the number of parallel machines per stage is within [1, 2], and the processing time is within [1, 10]. For large-sized problems, the number of jobs is within [10, 50], the number of stages is within [5, 25], the number of re-entrance is within [1, 6], the number of parallel machines per stage is within [1, 6], and the processing time is within [1, 30].

4.2. Performance Measures. This paper selected three performance indicators: SP, GD, and IGD, in which SP and GD are described in literature [38, 39]. Since the true optimal Pareto front of the problem is unknown, the non-dominated solutions of the five algorithms are approximated as the optimal Pareto front.

SP is used to measure the distribution uniformity of the non-dominated solutions on the Pareto front. As we all know, the smaller the SP, the better the result. When SP=0,

(1) /∗Set the parameters of the IMOMVO algorithm.∗/
(2) /∗Initialize the population using Latin hypercube sampling technology∗/★
(3) /∗initialize $iter$ =1. ∗/
(4) while($iter < Max\_iter$)
(5) /∗Update WEP and TDR∗/
(6) WEP = min + iter.((max − min)/Max_iter) and $TDR = 1 - (iter)^{1/P}/Max\_iter^{1/P}$
(7) for $i = 1 : n$
(8) /∗Boundary checking of the universes∗/
(9) /∗Calculate the inflation rate (fitness) of universes∗/
(10) /∗Calculate the start processing time of each operation according to PS decoding strategy ∗/
(11) /∗Right-shift procedure∗/★
(12) /∗Calculate the inflation rate (fitness) of universes∗/
(13) end for
(14) /∗Sort fitness values∗/
(15) /∗Find the non-dominated solutions∗/
(16) /∗Normalized the inflation rates∗/
(17) /∗Add the non-dominated solutions to archive ANS∗/
(18) if size of ANS is reaches NA then
(19) /∗Delete the solutions with many neighbouring solutions from the ANS∗/
(20) /∗Applying roulette wheel and $p_i' = N_i/c$∗/
(21) end if
(22) /∗Local search with Chaotic Local Search of Logical Self-Mapping method∗/★
(23) /∗Update the position of universes ∗/
(24) $$x_i^j = \begin{cases} \begin{cases} x_j + TDR \times ((ub_j - lb_j) \times r_4 + lb_j) & r_3 < 0.5 \\ x_j - TDR \times ((ub_j - lb_j) \times r_4 + lb_j) & r_3 \geq 0.5 \end{cases} & r2 < WEP \\ x_i^j & r2 \geq WEP \end{cases}$$
(25) if $r_5 > 0.5$
(26) /∗Update the position of universes using Levy flight ∗/★
(27) end
(28) /∗Boundary checking of the universes ∗/
(29) $iter = iter + 1$
(30) end while
(31) /∗Return the solutions in ANS∗/

Algorithm 1

the non-dominated solutions on the Pareto front are evenly distributed. The SP is calculated as follows:

$$SP = \sqrt{\frac{1}{n-1}\sum_{i=1}^{N}\left(\overline{D} - D_i\right)^2} \tag{23}$$

where $D_i = \min(\sum_{k=1}^{M} |f_k^i - f_k^j|)$, $i, j = 1, 2, 3 \ldots N, i \neq j$, $\overline{D} = (1/n)\sum_{i=1}^{n} D_i$, and N is the number of non-dominated solutions.

GD is used to evaluate the approximation degree between the front obtained by the algorithm and the real Pareto front of the problem which is calculated as follows:

$$GD = \frac{\sqrt{\sum_{i=1}^{N} dist_i^2}}{N} \tag{24}$$

where N is the number of non-dominated solutions in the Pareto front and $dist_i$ is the Euclidean distance between the *i-th* solution and the nearest solution in the optimal Pareto front. A smaller GD corresponds to better convergence of the algorithm. The minimum value of GD is 0, indicating that all the non-dominated solutions in the Pareto front are included in the optimal Pareto front.

IGD is a comprehensive performance evaluation indicator which can evaluate the convergence performance of the algorithm and the diversity performance of non-dominated solutions by calculating the minimum distance between the point on each optimal Pareto front and the Pareto front obtained by some algorithm. The calculation formula is as shown in (25), where $|N^*|$ is the number of non-dominated solutions in the optimal Pareto front and N is the Pareto non-inferior solution set obtained by the algorithm. Clearly, the smaller the *IGD,* the better the convergence and distribution quality.

$$IGD = \frac{\sum_{x \in N^*} dist\,(x, N)}{|N^*|} \tag{25}$$

4.3. Parameter Setting. The performances of metaheuristics often depend on the parameter setting. The IMOMVO algorithm mainly involves three key parameters, population size N, step size factor β of Lévy flight, and local detection speed P of MOMVO. Many instances selected from the

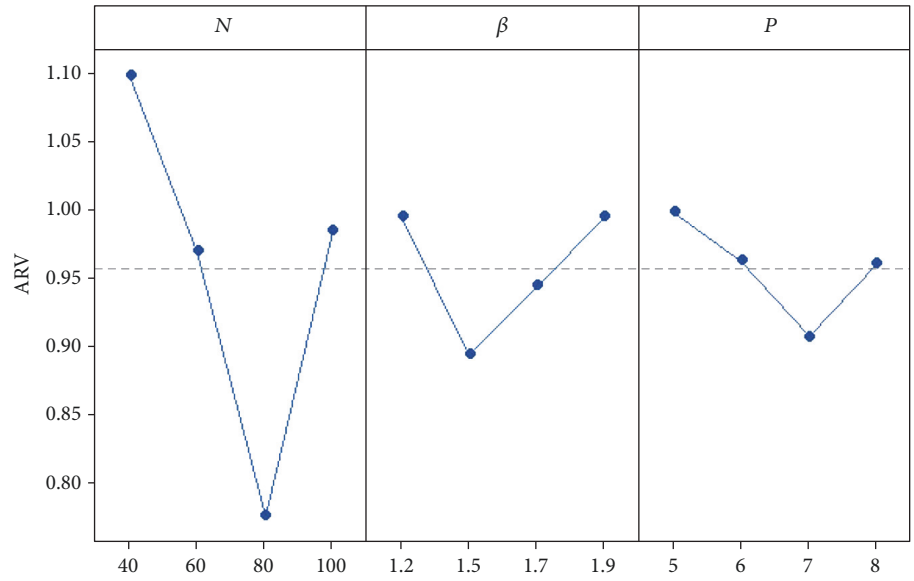

Figure 7: Main effect plot for RV.

Table 3: Levels of parameters.

Parameters	Factor level			
	1	2	3	4
N	40	60	80	100
β	1.2	1.5	1.7	1.9
P	5	6	7	8

benchmark set [13] are used for investigation, while there is little difference. In this paper, Taguchi method is used to study the effects of different parameters on the experimental results with Sproblem-04-20 as an example. The factor levels for each parameter are shown in Table 3. According to the number of factors as well as the number of factor levels, the orthogonal table L_{16} (3^4) is adopted. For each factor level combination in the orthogonal table, it runs 10 times independently. In addition, the IGD performance metrics is regarded as the response variable (RV). Clearly, the smaller the RV value, the better the combination. The experimental results are shown in Table 4. The average response value (ARV) and the importance order of each parameter are shown in Table 5. Figure 7 shows the level trend of each parameter.

From Table 5 and Figure 7, it can be seen that parameter N is the most significant one among the three parameters. Parameter β ranks second and parameter P ranks the last. Regarding parameter N, a large value of N may lead to an insufficient evolution process, while a small value may cause insufficient exploration at each generation. As for parameter β, a larger β meant larger step size, which is beneficial to global search, but not to local search. According to the above investigation, the three key parameters of the IMOMVO algorithm are set to N=80, β =1.5, and P=7.

4.4. Results and Discussion. In this paper, six small-sized and six large-sized problems are randomly selected for the benchmark set. Each problem runs 10 times independently using the five algorithms, and each run gets a set of [*SP*, *GD*, *IGD*]. The average (Avg), the standard deviations (Std), and the minimum values (Min) for the six small-sized and six large-sized problems are reported in Tables 6 and 8. The optimal results of the performance indicators are shown in bold.

The average and standard deviation can only represent the problem-solving performance from the macroscopic perspective. It can be seen from the student's t-tests whether there are significant differences between two algorithms. To show the statistical difference between the IMOMVO and other algorithms, the results are listed in Tables 7 and 9. As can be seen from Tables 7 and 9, *GD* and *IGD* of IMOMVO is better than those of the other algorithms on all the small-sized problems and most large-sized problems. Therefore, the proposed IMOMVO algorithm was significantly superior to other algorithms at 95% confidence level. However, regarding *SP*, IMOMVO was not significantly different from other algorithms. Therefore, the distribution levels of the five algorithms have little difference.

Taking Sproblem-04-02 as an example to analyze the scheduling results after adding the turning on and off control strategy. There are 16 jobs, 8 machines, 1 re-entrance, and 6 stages in the problem. Meanwhile, the number of identical parallel machines in each stage is 1, 2, 1, 2, 1, and 1. The Pareto front obtained by each of the five algorithms is shown in Figure 8. Taking a non-dominated solution 13—5—8—2—12—9—11—6—10—1—16—4—7—15—14—3 of IMOMVO algorithm as an example, its corresponding objective function values C_{max}=153, T_{max}= 110.7137, and SEC_{min}=222. In Figure 9, Gantt charts of the non-dominated explored by IMOMVO with considering the machine

Table 4: Orthogonal arrays and RV values.

Number	Factor			RV
	N	β	P	
1	40	1.2	5	1.1892
2	40	1.5	6	0.7710
3	40	1.7	7	1.0505
4	40	1.9	8	1.3805
5	60	1.2	6	1.1824
6	60	1.5	5	1.0333
7	60	1.7	8	0.8261
8	60	1.9	7	0.8349
9	80	1.2	7	0.6824
10	80	1.5	8	0.7110
11	80	1.7	5	0.8578
12	80	1.9	6	0.8531
13	100	1.2	8	0.9243
14	100	1.5	7	1.0596
15	100	1.7	6	1.0430
16	100	1.9	5	0.9107

Table 5: ARV values and significance ranks.

Level	N	β	P
1	1.0978	0.9946	0.9978
2	0.9692	0.8937	0.9624
3	0.7761	0.9444	0.9069
4	0.9844	0.9948	0.9605
Delta	0.3217	0.1011	0.0909
Rank	1	2	3

turning on and off control strategy are illustrated. Figure 10 shows the Gantt charts of the non-dominated (C_{max} = 153, T_{max}= 110.7137, and SEC_{min}=276) without considering the machine turning on and off control strategy. Comparative results show that the operations while considering the machine turning on and off control strategy is relatively centralized, and the energy consumption is reduced by 54. For machine 3, after job 4 is finished, job 14 is processed directly without waiting. In Figure 7, the double arrow indicates that the machine needs to be shut down. The first, seventh, and eighth machines do not need to shut down, while all other machines need to be shut down during the processing. For machine 5, there are five idling, all of which need to be shut down except for the fourth one. The machine turning on and off control vectors with and without considering the machine turning on and off control strategy are shown separately in formula (26) and (27), among which 1 means shutdown the machine and 0 means idle after finishing one operation. In addition, comparisons show that the number of idling of all machines is 42 and 32, respectively. So the number of idling is reduced by 10; however, both of them have 15 shutdowns during the processing.

$$U_1 = \left\{ \begin{array}{l} 0\ 0 \\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1 \\ 1\ 1\ 1\ 1 \\ 1\ 0\ 0\ 0 \\ 1\ 1\ 1\ 0\ 1 \\ 1\ 1 \\ 0\ 0\ 0 \\ 0\ 0 \end{array} \right\} \tag{26}$$

$$U_2 = \left\{ \begin{array}{l} 0\ 0 \\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1 \\ 1\ 1\ 1\ 1\ 1 \\ 0\ 0\ 1\ 0\ 0\ 0 \\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 1 \\ 1\ 1 \\ 1\ 0\ 0\ 0 \\ 0\ 0 \end{array} \right\} \tag{27}$$

Taking six small-sized problems as examples, each problem runs 10 times independently, and the relative variation of the average makespan and idle energy consumption of each problem is shown in Figure 11. The negative sign indicates the relative reduction. It can be seen from the graph that RHFSP considering machines turning on and off control strategy greatly reduces the idle energy consumption on the

Table 6: Comparisons of IMOMVO and the four algorithms for small-sized problems.

Test problems	Performance measures		IMOMVO	MOMVO	MOPSO	MOALO	NSGA-II
Sproblem-01-20	SP	Min	4.2538	3.3425	**2.6581**	4.8011	3.4495
		Avg	5.5441	6.1904	6.7477	9.2005	**5.4655**
		Std	**1.2617**	2.8282	2.9487	3.1855	1.5064
	GD	Min	**0.4243**	1.0905	1.7217	1.4066	1.3952
		Avg	**0.9150**	2.0821	2.2098	5.1492	2.0520
		Std	**0.3053**	1.1803	0.4345	2.3362	0.3902
	IGD	**Min**	**1.1835**	1.9842	2.2384	2.7163	2.1045
		Avg	**0.2990**	0.7035	0.4396	0.6491	0.5931
		Std	1.0887	**1.0087**	1.6941	1.6562	1.3959
Sproblem-02-20	SP	Min	6.4828	6.0011	7.8466	**3.8277**	3.9211
		Avg	9.7764	15.8743	14.8757	**9.0176**	14.3121
		Std	4.0321	9.8391	5.0085	**2.6563**	5.4210
	GD	Min	**0.8724**	1.5297	2.1647	3.3802	2.5898
		Avg	**1.9700**	4.0680	3.6181	5.3739	3.8907
		Std	**0.9110**	1.8535	1.2529	1.3984	1.3274
	IGD	Min	**1.0842**	2.6203	2.6529	3.3949	3.2842
		Avg	**2.4374**	5.1764	4.6116	6.1921	4.7595
		Std	**1.4366**	2.8636	1.5003	2.7036	2.1415
Sproblem-03-20	SP	Min	2.567	**0.8237**	2.7398	4.4499	2.3531
		Avg	4.9246	4.9596	**4.3365**	6.1176	6.0466
		Std	2.0407	2.2553	**1.4207**	1.9027	3.0756
	GD	Min	**0**	0.5079	1.5609	1.8643	0.9265
		Avg	**0.9232**	2.4554	2.1164	2.9399	1.8288
		Std	0.8034	1.4898	**0.4847**	0.7599	0.5991
	IGD	Min	**1.0441**	2.0295	1.3708	2.1544	1.4214
		Avg	**1.4052**	3.1044	2.7329	2.7603	2.5920
		Std	**0.2756**	1.0239	1.3596	0.5844	1.1414
Sproblem-04-20	SP	Min	4.4390	**3.0618**	4.1757	6.0383	5.036
		Avg	**6.4714**	6.7734	7.4176	9.9239	7.5878
		Std	**1.1955**	1.8944	2.2194	2.6719	2.4746
	GD	Min	**0.5672**	2.0327	1.9053	4.3381	1.8788
		Avg	**1.2277**	2.8152	3.5303	7.1940	3.0729
		Std	**0.5413**	0.6537	2.9476	2.2363	0.8076
	IGD	Min	**0.8572**	1.2991	2.1892	3.399	2.1246
		Avg	**1.5798**	2.4576	3.0508	5.4666	2.8488
		Std	**0.4280**	0.7041	0.7275	1.1314	0.7762
Sproblem-05-20	SP	Min	**1.9556**	3.7677	2.5162	3.8762	3.4323
		Avg	**5.3309**	8.5614	7.9275	7.2344	6.4903
		Std	**2.0804**	3.7584	9.4040	3.0191	3.3006
	GD	Min	0.4654	2.7581	**0**	4.2029	3.5653
		Avg	**1.7576**	4.3472	3.8166	8.6313	4.5525
		Std	**0.9737**	2.0990	1.9706	3.5671	0.9877
	IGD	Min	**1.4727**	2.0496	3.0676	5.155	2.7927
		Avg	**2.0439**	3.5092	3.8419	7.9333	4.0812
		Std	**0.5769**	1.1922	0.8280	2.9383	1.0016
Sproblem-06-20	SP	Min	5.6331	7.7566	**5.1192**	7.3206	5.7043
		Avg	10.1854	13.3589	9.5592	12.1647	**8.7989**
		Std	3.58811	5.2599	4.1399	3.5184	**2.8420**
	GD	Min	**0**	2.8996	3.3732	5.2997	3.1649
		Avg	**1.5256**	5.1802	5.5500	14.7124	5.1470
		Std	**0.9362**	1.9553	2.3292	7.9477	1.5420
	IGD	Min	**1.1463**	3.067	3.0608	4.3261	3.0502
		Avg	**2.2706**	4.6015	5.2186	12.3180	4.5948
		Std	**0.6253**	1.1447	1.3723	5.1917	0.8459

Table 7: T-test of IMOMVO vs. MOMVO, MOPSO, MOALO, and NSGA-II for small-sized problems.

Test problems	Performance measures	p value of IMOMVO vs.			
		MOMVO	MOPSO	MOALO	NSGA-II
Sproblem01-20	SP	0.518	0.258	**0.006**	0.901
	GD	**0.012**	**0**	**0**	**0**
	IGD	**0.006**	**0**	**0**	**0**
Sproblem02-20	SP	0.095	**0.022**	0.625	**0.048**
	GD	**0.005**	**0.003**	**0**	**0.001**
	IGD	**0.015**	**0.004**	**0.010**	**0.011**
Sproblem03-20	SP	0.977	0.464	0.193	0.349
	GD	**0.010**	**0.001**	**0**	**0.010**
	IGD	**0**	**0.013**	**0**	**0.010**
Sproblem04-20	SP	0.675	0.251	**0.002**	0.215
	GD	**0**	**0.026**	**0**	**0**
	IGD	**0.003**	**0**	**0**	**0**
Sproblem05-20	SP	**0.029**	0.405	0.118	0.360
	GD	**0.040**	**0.008**	**0**	**0**
	IGD	**0.004**	**0**	**0**	**0**
Sproblem06-20	SP	0.132	0.722	0.229	0.351
	GD	**0**	**0**	**0.001**	**0**
	IGD	**0**	**0**	**0**	**0**

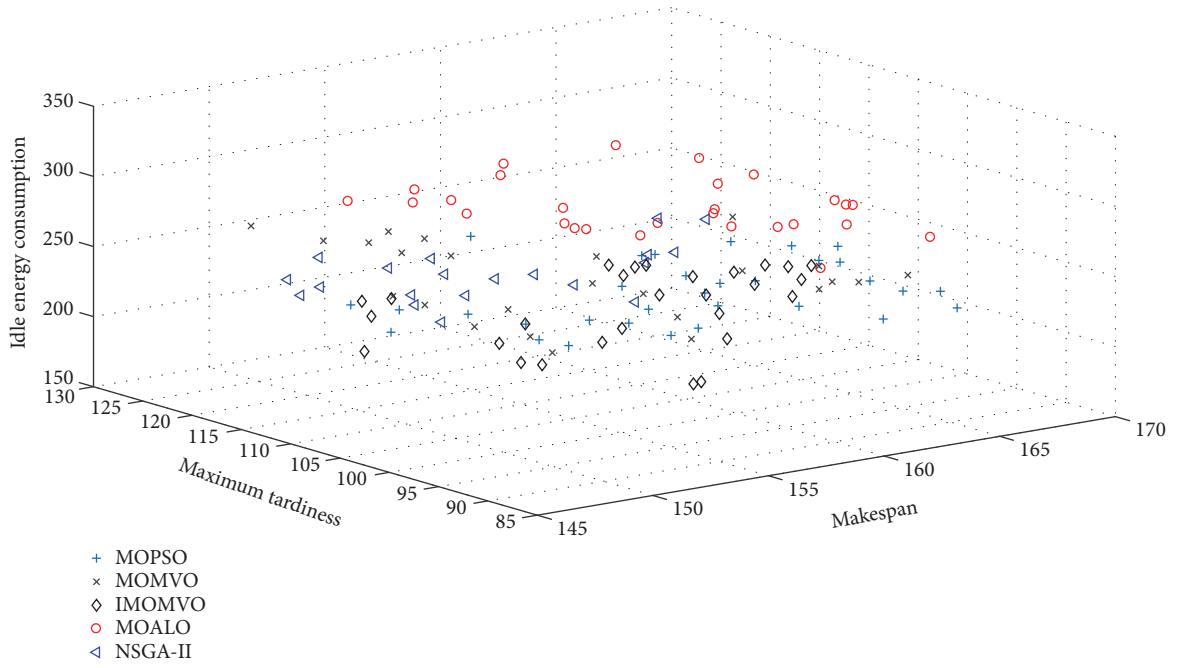

Figure 8: Pareto fronts of Sproblem-04-02.

premise of ensuring the production efficiency, which shows the effectiveness of the strategy.

To sum up, it can be seen from the experimental results that the RHFSP considering machines turning on and off control strategy reduced the total energy consumption of the machines effectively without sacrificing the production efficiency. The useless energy consumption in the production process can be reduced by reasonably setting the turning on and off of the machines. The proposed IMOMVO is more effective than other algorithms for solving the multi-objective RHFSP.

5. Conclusions

In this paper, the RHFSP with the objectives of makespan, maximum tardiness and idle energy consumption was solved

Table 8: Comparisons of IMOMVO and the four algorithms for large-sized problems.

Test problems	Performance measures		IMOMVO	MOMVO	MOPSO	MOALO	NSGA-II
Lproblem-01-20	SP	Min	**31.5046**	37.9386	39.2627	45.2234	32.5645
		Avg	**47.0788**	63.0421	48.9852	83.7999	56.2081
		Std	14.6542	24.1826	**9.5585**	24.5639	19.7613
	GD	Min	**9.6393**	20.0731	40.3205	58.4695	14.7962
		Avg	**15.0285**	33.6796	49.1722	87.9470	28.7340
		Std	**4.0260**	11.0479	8.2321	19.4371	15.4064
	IGD	Min	**11.1861**	14.2001	32.0123	36.2466	21.3965
		Avg	**13.6352**	31.7573	49.3988	63.4794	27.3624
		Std	**1.6068**	11.5299	11.0756	28.5335	4.5460
Lproblem-02-20	SP	Min	39.2138	43.3812	41.9372	43.5818	**35.7117**
		Avg	**48.0512**	80.0144	58.5973	78.0675	48.6288
		Std	**10.2653**	42.4928	11.7668	22.4166	16.4953
	GD	Min	**0**	13.7379	36.8747	31.5268	25.0612
		Avg	**13.0129**	53.0043	52.3202	52.7521	35.1891
		Std	**6.6623**	26.9211	12.8599	11.8538	16.3095
	IGD	Min	**0**	12.3588	17.6584	31.939	22.0302
		Avg	**11.7856**	50.5360	52.2226	64.0970	33.5677
		Std	**6.5477**	32.6129	33.5188	27.0935	19.0259
Lproblem-03-20	SP	Min	12.9284	14.5336	**11.0352**	17.2225	14.2715
		Avg	**17.1909**	20.6562	17.6746	28.972	17.6740
		Std	2.6819	5.0773	5.2745	16.9026	**2.6553**
	GD	Min	**0**	6.5092	**0**	20.6104	0.6099
		Avg	**1.7627**	16.8459	15.5075	32.7396	9.3958
		Std	**2.6162**	7.7080	7.1737	10.1721	5.5497
	IGD	Min	**1.3130**	10.1753	11.3452	26.5771	1.3472
		Avg	**5.2737**	17.9577	24.5695	39.4279	16.1063
		Std	**2.1392**	7.8994	20.5780	909037	3.9406
Lproblem-04-20	SP	Min	19.7206	**16.7344**	19.5014	23.1202	22.6913
		Avg	**26.3831**	31.3330	30.8081	35.5944	27.2626
		Std	**4.6792**	12.3887	9.3875	5.5558	5.7137
	GD	Min	**0**	2.0338	6.719	23.9144	8.2262
		Avg	**5.0776**	13.6868	15.6351	34.7515	15.0124
		Std	**3.5823**	10.2058	5.1480	5.7331	9.2964
	IGD	Min	**0**	5.8252	9.7547	20.8213	7.7169
		Avg	**5.8495**	12.9112	15.1598	33.2070	12.0886
		Std	**2.3550**	5.3789	4.7880	12.1977	4.7048
Lproblem-05-20	SP	Min	**0**	**0**	**0**	**0**	**0**
		Avg	19.7206	26.1458	24.2800	**18.4337**	21.7567
		Std	31.3350	33.8072	**25.3138**	31.3129	27.0370
	GD	Min	**0**	**0**	5.1640	**0**	**0**
		Avg	**4.8984**	14.9427	19.6865	26.1215	16.9535
		Std	**3.3327**	8.2743	20.7626	15.1955	8.0954
	IGD	Min	6	**0**	7	21.7401	11.3529
		Avg	**9.1573**	22.7388	25.2810	29.1370	27.4365
		Std	**1.9291**	16.6252	14.5438	3.1175	17.4456
Lproblem-06-20	SP	Min	**26.3994**	27.5350	29.1025	42.2188	33.6225
		Avg	56.4040	57.7383	**47.7134**	63.3400	52.4290
		Std	17.8382	19.7179	14.8274	19.3504	**14.5233**
	GD	Min	**0**	10.4265	4.7540	18.6695	8.2554
		Avg	**4.6472**	21.3337	11.9000	27.2603	14.1198
		Std	**2.4220**	10.5578	8.0866	8.5111	5.9599
	IGD	Min	**10.4799**	17.6675	17.8134	18.5859	16.1167
		Avg	**12.4413**	27.938	21.5491	38.2586	23.4192
		Std	**1.7633**	8.9999	4.1238	17.7398	7.8422

Table 9: T-test of IMOMVO vs. MOMVO, MOPSO, MOALO, and NSGA-II for large-sized problems.

Test problems	Performance measures	p value of IMOMVO vs.			
		MOMVO	MOPSO	MOALO	NSGA-II
Lproblem01-20	SP	0.091	**0.001**	0.256	0.734
	GD	**0**	**0**	**0.014**	**0**
	IGD	**0.001**	**0**	**0**	**0**
Lproblem02-20	SP	0.127	0.129	**0.014**	0.943
	GD	**0.014**	**0**	**0**	**0.001**
	IGD	**0.028**	**0.037**	**0.005**	**0.021**
Lproblem03-20	SP	0.072	0.799	**0.043**	0.690
	GD	**0**	**0**	**0**	**0.002**
	IGD	**0.001**	**0**	**0**	**0**
Lproblem04-20	SP	0.261	0.205	**0.001**	0.711
	GD	**0.028**	**0**	**0**	**0.005**
	IGD	**0.002**	**0**	**0**	**0.001**
Lproblem05-20	SP	0.665	0.725	0.928	0.878
	GD	**0.004**	**0.039**	**0.002**	**0**
	IGD	**0.040**	**0.013**	**0**	**0.005**
Lproblem06-20	SP	0.928	0.125	0.671	0.392
	GD	**0.001**	**0.021**	**0**	**0**
	IGD	**0**	**0**	**0.001**	**0.002**

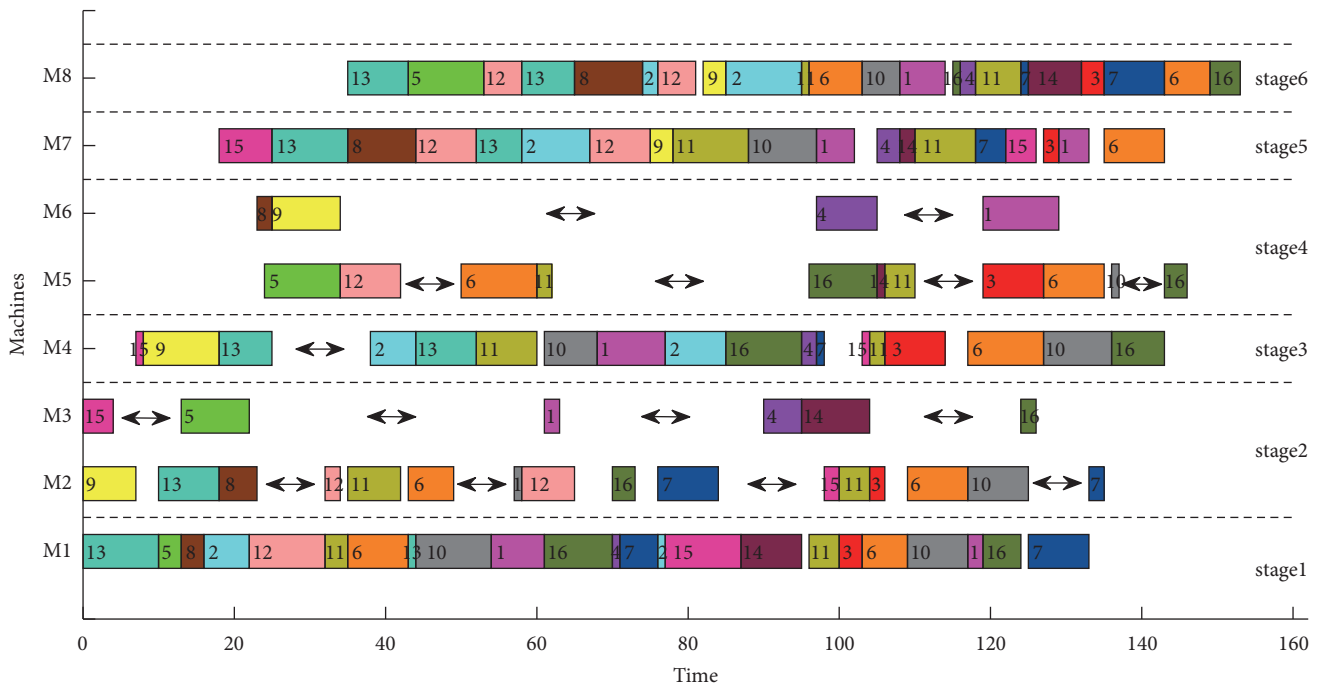

Figure 9: Gantt charts of a non-dominated solution with machines turning on and off control strategy.

by the MOMVO algorithm. In order to establish a high-performance approach for this problem, the MOMVO algorithm was improved, including population initialization based on Latin hypercube sampling (LHS), individual position update based on Lévy flight, and chaotic local search operation based on logical self-mapping. The effectiveness of the improved operations was shown by numerical tests. Experimental results demonstrated the superiority of the proposed IMOMVO to the other algorithms. Specifically, the comparative results showed that IMOMVO was significantly better than other algorithms in terms of the convergence and diversity of the non-dominated solutions. Regarding the distribution of the non-dominated solutions, there was no significant difference in the five algorithms. At present, the research on the RHFSP is not profound enough. The RHFSP model considering the machine turning on and off control

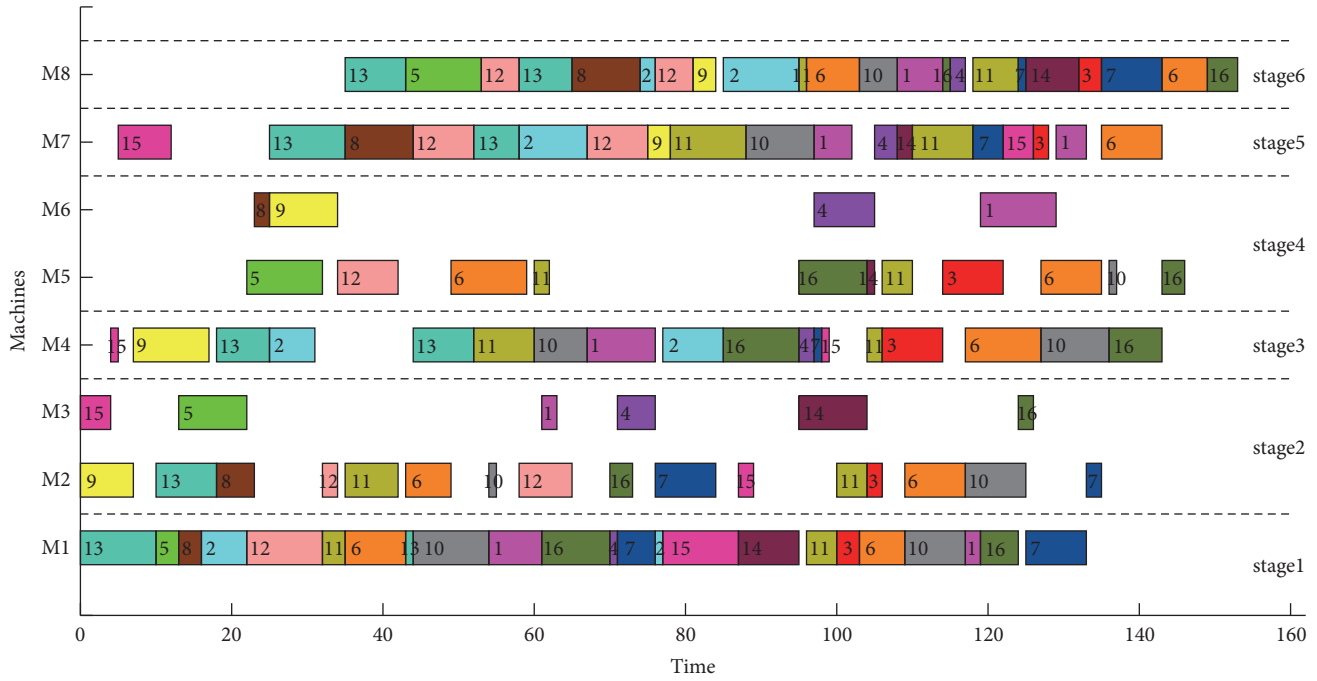

Figure 10: Gantt charts of a non-dominated solution without machines turning on and off control strategy.

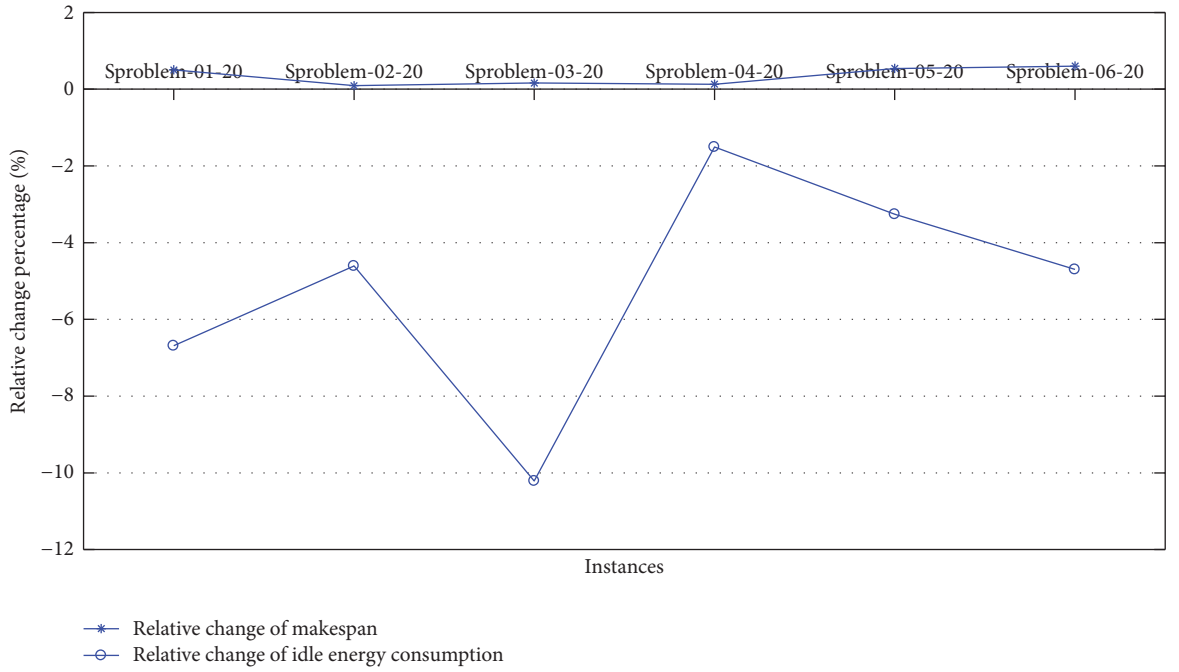

Figure 11: Relative change of makespan and idle energy consumption.

strategy in this paper can reduce the energy consumption of the machine effectively. It is mainly suitable for the scenarios in which the turning on and off is convenient and the turning on and off energy consumption is relatively low. In the future, the work could focus on the RHFSP considering time-of-use tariffs and the joint optimization considering maintenance.

Data Availability

The data used to support the findings of this study were supplied by Professor Cho HM. These datasets are cited at relevant places within the text as references [13]. Authors are not authorized to make them public.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank Professor Cho HM for providing us the data sets. The research of this paper is made possible by the generous support from National Natural Science Foundation of China (71840003); Science and Technology Development Program of University of Shanghai for Science and Technology (2018KJFZ043); Ministry of Education "Cloud Number Integration Science and Education Innovation" Fund Project (2017A01109); and Henan Province Science and Technology Research Project (182102210113).

References

[1] M. Y. Wang, S. P. Sethi, and S. L. Van De Velde, "Minimizing makespan in a class of reentrant shops," *Operations Research*, vol. 45, no. 5, pp. 702–712, 1997.

[2] S. C. Graves, H. C. Meal, D. Stefek, and A. H. Zeghmi, "Scheduling of re-entrant flow shops," *Journal of Operations Management*, vol. 3, no. 4, pp. 197–207, 1983.

[3] S. Bertel and J.-C. Billaut, "A genetic algorithm for an industrial multiprocessor flow shop scheduling problem with recirculation," *European Journal of Operational Research*, vol. 159, no. 3, pp. 651–662, 2004.

[4] W. L. Pearn, S. H. Chung, A. Y. Chen, and M. H. Yang, "A case study on the multistage IC final testing scheduling problem with reentry," *International Journal of Production Economics*, vol. 88, no. 3, pp. 257–267, 2004.

[5] S.-W. Choi and Y.-D. Kim, "Minimizing makespan on an m-machine re-entrant flowshop," *Computers & Operations Research*, vol. 35, no. 5, pp. 1684–1696, 2008.

[6] H.-W. Kim and D.-H. Lee, "Heuristic algorithms for re-entrant hybrid flow shop scheduling with unrelated parallel machines," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 223, no. 4, pp. 433–442, 2009.

[7] H. Choi, H. Kim, D. Lee, J. Yoon, C. Y. Yun, and K. B. Chae, "Scheduling algorithms for two-stage reentrant hybrid flow shops: minimizing makespan under the maximum allowable due dates," *The International Journal of Advanced Manufacturing Technology*, vol. 42, no. 9-10, pp. 963–973, 2009.

[8] I. A. El-Khouly, K. S. El-Kilany, and A. E. El-Sayed, "Modelling and simulation of re-entrant flow shop scheduling: An applicationin semiconductor manufacturing," in *Proceedings of the 2009 International Conference on Computers and Industrial Engineering, CIE 2009*, pp. 211–216, France, July 2009.

[9] M. Hekmatfar, S. M. T. Fatemi Ghomi, and B. Karimi, "Two stage reentrant hybrid flow shop with setup times and the criterion of minimizing makespan," *Applied Soft Computing*, vol. 11, no. 8, pp. 4530–4539, 2011.

[10] D. Lin, C. K. M. Lee, and Z. Wu, "Integrating analytical hierarchy process to genetic algorithm for re-entrant flow shop scheduling problem," *International Journal of Production Research*, vol. 50, no. 7, pp. 1813–1824, 2012.

[11] J. Chen, J. C. Pan, and C. Wu, "Minimizing makespan in reentrant flow-shops using hybrid tabu search," *The International Journal of Advanced Manufacturing Technology*, vol. 34, no. 3-4, pp. 353–361, 2007.

[12] C.-C. Wu, S.-C. Liu, T. C. Cheng, Y. Cheng, S.-Y. Liu, and W.-C. Lin, "Re-entrant flowshop scheduling with learning considerations to minimize the makespan," *Iranian Journal of Science & Technology, Transactions A: Science*, vol. 42, no. 2, pp. 727–744, 2018.

[13] H.-M. Cho, S.-J. Bae, J. Kim, and I.-J. Jeong, "Bi-objective scheduling for reentrant hybrid flow shop using Pareto genetic algorithm," *Computers & Industrial Engineering*, vol. 61, no. 3, pp. 529–541, 2011.

[14] H.-S. Choi, J.-S. Kim, and D.-H. Lee, "Real-time scheduling for reentrant hybrid flow shops: a decision tree based mechanism and its application to a TFT-LCD line," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3514–3521, 2011.

[15] K.-C. Ying, S.-W. Lin, and S.-Y. Wan, "Bi-objective reentrant hybrid flowshop scheduling: an iterated pareto greedy algorithm," *International Journal of Production Research*, vol. 52, no. 19, pp. 5735–5747, 2014.

[16] J.-N. Shen, L. Wang, and H.-Y. Zheng, "A modified teaching-learning-based optimisation algorithm for bi-objective re-entrant hybrid flowshop scheduling," *International Journal of Production Research*, vol. 54, no. 12, pp. 3622–3639, 2016.

[17] J. Shen, L. Wang, J. Deng, and X. Zheng, "A pareto-based discrete harmony search algorithm for bi-objective reentrant hybrid flowshop scheduling problem," *Advances in Intelligent Systems and Computing*, vol. 382, pp. 435–445, 2016.

[18] 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.

[19] S. A. Mansouri, E. Aktas, and U. Besikci, "Green scheduling of a two-machine flowshop: trade-off between makespan and energy consumption," *European Journal of Operational Research*, vol. 248, no. 3, pp. 772–788, 2016.

[20] 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.

[21] S. Wang, X. Wang, J. Yu, S. Ma, and M. Liu, "Bi-objective identical parallel machine scheduling to minimize total energy consumption and makespan," *Journal of Cleaner Production*, vol. 193, pp. 424–440, 2018.

[22] Z. Liu, S. Guo, and L. Wang, "Integrated green scheduling optimization of flexible job shop and crane transportation considering comprehensive energy consumption," *Journal of Cleaner Production*, vol. 211, pp. 765–786, 2019.

[23] M. B. Yildirim and G. Mouzon, "Single-machine sustainable production planning to minimize total energy consumption and total completion time using a multiple objective genetic algorithm," *IEEE Transactions on Engineering Management*, vol. 59, no. 4, pp. 585–597, 2012.

[24] 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.

[25] 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.

[26] L. Danping and C. K. M. Lee, "A review of the research methodology for the re-entrant scheduling problem," *International*

Journal of Production Research, vol. 49, no. 8, pp. 2221–2242, 2011.

[27] D. Lin, C. K. M. Lee, and W. Ho, "Multi-level genetic algorithm for the resource-constrained re-entrant scheduling problem in the flow shop," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 4, pp. 1282–1290, 2013.

[28] M. Pinedo and K. Hadavi, "Scheduling: theory, algorithms and systems development," in *Proceedings of the Operations Research Proceedings 1991*, pp. 35–42, 1992.

[29] D. N. Kordonowy, *A power assessment of machining tools [BSc thesis]*, Massachusetts Institute of Technology, Cambridge, UK, 2002.

[30] T. Gutowski, C. Murphy, D. Allen et al., "Environmentally benign manufacturing: observations from Japan, Europe and the United States," *Journal of Cleaner Production*, vol. 13, no. 1, pp. 1–17, 2005.

[31] Y. He, B. Liu, X. Zhang, H. Gao, and X. Liu, "A modeling method of task-oriented energy consumption for machining manufacturing system," *Journal of Cleaner Production*, vol. 23, no. 1, pp. 167–174, 2012.

[32] J. C.-H. Pan and J.-S. Chen, "Mixed binary integer programming formulations for the reentrant job shop scheduling problem," *Computers & Operations Research*, vol. 32, no. 5, pp. 1197–1212, 2005.

[33] R. Qing-dao-er-ji and Y. Wang, "Inventory based bi-objective flow shop scheduling model and its hybrid genetic algorithm," *Mathematical Problems in Engineering*, vol. 2013, Article ID 976065, 7 pages, 2013.

[34] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[35] S. Mirjalili, P. Jangir, S. Z. Mirjalili, S. Saremi, and I. N. Trivedi, "Optimization of problems with multiple objectives using the multi-verse optimization algorithm," *Knowledge-Based Systems*, vol. 134, pp. 50–71, 2017.

[36] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513, 2016.

[37] X. Zhang, T. Yang, and N. Cui, "Flame image segmentation based on the bee colony algorithm with characteristics of levy flights," *Mathematical Problems in Engineering*, vol. 2015, Article ID 805075, 8 pages, 2015.

[38] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.

[39] C. Lu, L. Gao, X. Li, Q. Wang, W. Liao, and Q. Zhao, "An efficient multiobjective backtracking search algorithm for single machine scheduling with controllable processing times," *Mathematical Problems in Engineering*, vol. 2017, Article ID 8696985, 24 pages, 2017.