Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 8574892, 12 pages
https://doi.org/10.1155/2018/8574892

*Research Article*

# Energy-Efficient Scheduling for a Job Shop Using Grey Wolf Optimization Algorithm with Double-Searching Mode

**Tianhua Jiang**[1], **Chao Zhang**[2], **Huiqi Zhu**[1], **and Guanlong Deng**[3]

[1] *School of Transportation, Ludong University, Yantai, Shandong, 264025, China*
[2] *Department of Computer Science and Technology, Henan Institute of Technology, Xinxiang, 453003, China*
[3] *School of Information and Electrical Engineering, Ludong University, Yantai, 264025, Shandong, China*

Correspondence should be addressed to Tianhua Jiang; jth1127@163.com

Received 3 July 2018; Revised 17 October 2018; Accepted 23 October 2018; Published 30 October 2018

Academic Editor: Antonio M. Gonçalves de Lima



Workshop scheduling has mainly focused on the performances involving the production efficiency, such as times and quality, etc. In recent years, environmental metrics have attracted the attention of many researchers. In this study, an energy-efficient job shop scheduling problem is considered, and a grey wolf optimization algorithm with double-searching mode (DMGWO) is proposed with the objective of minimizing the total cost of energy-consumption and tardiness. Firstly, the algorithm starts with a discrete encoding mechanism, and then a heuristic algorithm and the random rule are employed to implement the population initialization. Secondly, a new framework with double-searching mode is developed for the GWO algorithm. In the proposed DMGWO algorithm, besides of the searching mode of the original GWO, a random seeking mode is added to enhance the global search ability. Furthermore, an adaptive selection operator of the two searching modes is also presented to coordinate the exploration and exploitation. In each searching mode, a discrete updating method of individuals is designed by considering the discrete characteristics of the scheduling solution, which can make the algorithm directly work in a discrete domain. In order to further improve the solution quality, a local search strategy is embedded into the algorithm. Finally, extensive simulations demonstrate the effectiveness of the proposed DMGWO algorithm for solving the energy-efficient job shop scheduling problem based on 43 benchmarks.

## 1. Instruction

Job shop scheduling problem (JSP) has a strong theoretical and applied background, which has been widely concerned by researchers. In the actual production, many problems can be taken as a job shop scheduling problem, such as workshop scheduling in the industry, departure and arrival times of logistic problems, the delivery times of orders in a company, and so on. In the manufacturing field, JSP aims to determine the processing order between jobs on each machine to acquire good production performance, e.g., makespan [1, 2], total weighted tardiness [3, 4], average flow time [5, 6], etc. In previous researches, most work about JSP only consider the time-related indicators, rather than the environmental factors, such as energy consumption, $CO_2$ emissions and carbon footprint, etc. These studies can not adequately adapt to the development needs under the global low-carbon economy.

In recent years, energy-efficient production scheduling problem has gradually attracted the researchers' attention. Yildirim and Mouzon [7] established a mathematical model of a single-machine system and proposed a multiobjective genetic algorithm to optimize energy consumption and total completion time. Shrouf et al. [8] built a mathematical model of the single-machine sustainable scheduling with variable energy prices to minimize energy consumption costs. Che et al. [9] investigated a single-machine scheduling under time-dependent electricity tariffs. A continuous-time mixed-integer linear programming model was developed, and an efficient greedy insertion algorithm was proposed to optimize the total electricity cost within a given makespan. Tang et al. [10] developed a particle swarm optimization for solving the dynamic scheduling problem in a flexible flow shop with the criterion to minimize energy consumption and makespan. Lu et al. [11] investigated an energy-efficient permutation flow shop scheduling problem with sequence-dependent setup and controllable transportation time. A hybrid multiobjective backtracking search algorithm was presented to obtain the optimal makespan and energy consumption. Ding et al. [12] addressed a permutation flow shop scheduling problem to minimize the total carbon emissions and the makespan. A multiobjective NEH algorithm and a modified multiobjective iterated greedy algorithm were proposed to solve the problem. Ai and Lei [13] proposed a new neighborhood search strategy to solve a hybrid flow shop scheduling problem with the criterion to minimize the carbon emissions.

Regarding the above literature, many energy-efficient scheduling problems concentrate on the single-machine or flow shop systems. Considering the importance of JSP, it is more practical for considering the problem with environmental metrics. Salido et al. [14] investigated an energy-efficient job shop scheduling problem, where each operation can be processed on one machine at several alternative speeds. Zhang and Chiong [15] addressed a multiobjective energy-efficient job shop scheduling problem with a machine speed scaling framework for minimizing the energy consumption and total weighted tardiness. May et al. [16] considered the effects of production scheduling scheme on the makespan and the energy consumption in a job shop. For such a problem, the introduction of environmental factor increases the number of variables and constraints and makes the problem more complex than the original JSP. It is well-known that metaheuristic algorithms have shown efficiency for solving the production scheduling problem. Therefore, the application of metaheuristics on solving the energy-efficient JSP will be also a hotspot in the area of production scheduling.

Inspired from the hunting behavior and the hierarchy structure of wolf pack in the nature, the grey wolf optimization (GWO) algorithm was developed by Mirjalili et al. in 2014 [17]. Due to some characteristics, such as high precision and fast convergence and ease of implementation, GWO and its different variants have been used to solve various optimization problems, e.g., feature selection [18], maximum power point tracking [19], UAV path planning [20], global optimization [21, 22] and power scheduling [23], etc. The experimental results in these pieces of literature indicate that GWO is competitive to other efficient algorithms, such as GA, PSO, and DE. However, for the scheduling problems, there are few pieces of literature involving the application of GWO. Lu et al. [24] developed a multiobjective discrete GWO for a real-life welding workshop with the criterion to optimize production efficiency and machine load. Lu et al. [25] presented a hybrid multiobjective GWO to deal with dynamic welding scheduling problem. Komaki and Kayvanfar [26] proposed a grey wolf optimizer to solve the scheduling problem in a two-stage assembly flow shop. Maharana and Kotecha [27] evaluated the performance of GWO on five job shop scheduling problems with parallel machines. The application of GWO on the production scheduling problems should be paid more attention. Based on the above analysis of the energy-efficient JSP and the efficiency of the GWO algorithm, we propose an improved GWO algorithm, namely, DMGWO, in this paper. The main novelties of this work are listed as follows: (1) a new framework with double-searching mode is presented in the GWO algorithm; (2) an adaptive selection method of the two searching modes is proposed to balance the ability of exploration and exploitation; (3) a new discrete individual updating method is developed based on the discrete characteristics of scheduling solution, which can make the algorithm directly work in a discrete domain. Extensive experimental results demonstrate that the proposed DMGWO algorithm is effective for the problem under study.

## 2. Problem Description

$n$ jobs are required to be processed on $m$ machines in the workshop with fixed and certain processing times and routing. In this study, we concentrate on the effects of production scheduling on the productive and environmental performances in a job shop, i.e., total tardiness cost and energy consumption cost.

Some assumptions are involved as follows:

(1) Any job can not be processed on more than one machine at the same time

(2) Each machine can only process one operation simultaneously

(3) No preemption is allowed once a job starts to be processed

(4) Setup time and breakdown of machines are not considered in this paper

(5) Each machine will not stop until the tasks assigned to it are completed. During the idle periods, each machine will be on a stand-by mode

For the energy-efficient job shop scheduling problem, May et al. [16] considered five energetic states. To simplify the problem, only two states (working and stand-by) are involved in [14, 15]. In this study, the energy consumption is also classified into two types. The first one is the processing energy consumed for dealing with jobs by machines. The other is the no-load energy consumed within the idle time interval between two successive jobs on a machine. To quantify the energy consumption cost, the average processing energy consumption cost per unit time and the average no-load energy consumption cost per unit time of each machine are predefined. Because the processing times and routing are certain, the processing energy consumption cost is fixed. Therefore, in order to control the total energy consumption cost, it is very necessary to arrange the processing sequence of jobs on each machine to reduce the no-load energy consumption cost. In addition, the tardiness of jobs is an important metrics for the classical job shop scheduling problem. Therefore, for the energy-efficient JSP under study, the optimization objective is to minimize the sum of the no-load energy consumption cost and the total tardiness cost, which is different from the objectives in [14, 15]:

$$\min CS = \sum_{k=1}^{m} \lambda_k \left(CT_k - WL_k\right) + \sum_{i=1}^{n} \eta_i \max\left\{0, C_i - d_i\right\} \quad (1)$$

$$C_{ik} - p_{ik} + Q\left(1 - x_{ijk}\right) \geq C_{ij}, \quad i = 1, 2, \cdots, n;\ j, k = 1, 2, \cdots, m \tag{2}$$

$$C_{lk} - C_{ik} + Q\left(1 - y_{ilk}\right) \geq p_{lk}, \quad i, l = 1, 2, \cdots, n;\ k = 1, 2, \cdots, m \tag{3}$$

$$C_{ik} \geq 0, \quad i = 1, 2, \cdots, n;\ k = 1, 2, \cdots, m \tag{4}$$

$$CT_k = \max\left\{C_{ik}\right\}, \quad i = 1, 2, \cdots, n \tag{5}$$

$$WL_k = \sum_{i=1}^{n} p_{ik}, \quad k = 1, 2, \cdots, m \tag{6}$$

$$x_{ijk} \in \{0, 1\}, \quad i = 1, 2, \cdots, n;\ j, k = 1, 2, \cdots, m \tag{7}$$

$$y_{ilk} \in \{0, 1\}, \quad i, l = 1, 2, \cdots, n;\ k = 1, 2, \cdots, m. \tag{8}$$

$CS$ is the total cost; $\lambda_k$ is the average no-load energy consumption cost per unit time of machine $k$; $CT_k$ defines the completion time of machine $k$; $WL_k$ is the workload of machine $k$; $\eta_i$ represents the tardiness cost per unit time of job $i$; $C_i$ defines the final completion time of job $i$; $d_i$ is the due date of job $i$; $C_{ik}$ represents the completion time of job $i$ on machine $k$; $p_{ik}$ means the processing time of job $i$ on machine $k$; $Q$ is a big positive value; $x_{ijk}$ is a 0-1 variable, if machine $j$ processes job $i$ prior to machine $k$, $x_{ijk}$=1, otherwise, $x_{ijk}$=0; $y_{ilk}$ is a 0-1 variable, if job $i$ is processed on machine $k$ prior to job $l$, $y_{ilk}$=1, otherwise, $y_{ilk}$=0.

Equation (1) is the objective function, where the first item corresponds to the no-load energy consumption cost, and the second means the tardiness cost. Constraint (2) represents the precedence relationship between operations in a job. Constraint (3) denotes that each machine can only process one operation at the same time. Constraint (4) guarantees that the completion time of each operation must be nonnegative. Constraint (5) is the completion time of machine $k$. Constraint (6) represents the workload of machine $k$. Constraints (7) and (8) represent 0-1 variables.

## 3. The Proposed DMGWO

*3.1. The Original GWO.* GWO can be viewed as a typical swarm-based intelligent algorithm, which mimics the hierarchy structure and hunting behavior of wolf pack in nature [17]. Starting with a predefined size of population, wolves are hierarchically classified into four types ($\alpha$, $\beta$, $\delta$, and $\omega$) according to their fitness values, which can be shown in Figure 1. The leader of the group ($\alpha$) on the first level has a strong ability in managing the behaviors of other wolves in the group. The second level of the group ($\beta$) can help $\alpha$ to make decisions and manage the other wolves on the lower levels. $\delta$ locates on the third level which are dominated by alpha and manage the behaviors of $\omega$. $\omega$ locates on the lowest level and obeys the other wolves on the higher levels such as $\alpha$, $\beta$, and $\delta$.

In the algorithm, $\alpha$, $\beta$, and $\delta$ correspond to the best solution, the second best solution, and the third best solution, respectively, and all the other solutions are defined as $\omega$, whose search behaviors are mainly guided by $\alpha$, $\beta$, and $\delta$. Wolves have special ability in finding the position of prey and encircling it. During the hunting process, it is assumed that $\alpha$, $\beta$, and $\delta$ can guesstimate the possible location of the prey. The three best individuals are saved and guide the others to update their positions in each iteration. The detailed description of the algorithm can be found in [17].

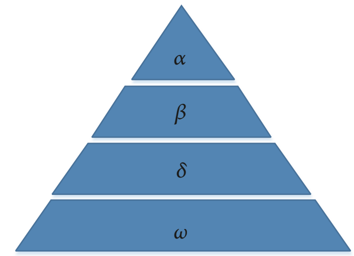


Figure 1: The hierarchy structure of the wolf pack [17].

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 1 | 2 | 3 | 3 | 1 | 1 | 3 |

Figure 2: An example of the scheduling solution.

*3.2. Encoding Method.* It is obvious that the problem under study is a typical discrete combinatorial problem. However, the original GWO was proposed for continuous optimization problems. Some adjustments should be developed to make the GWO work in a discrete search space. To implement it, the first step of constructing a GWO is to adopt an appropriate encoding method. Therefore, a discrete operation-based encoding method is first employed here. Taking a $3 \times 3$ JSP (3 jobs, 3 machines) for example, the scheduling solution can be shown in Figure 2.

*3.3. Population Initialization.* For a swarm-based intelligent algorithm, population initialization is very crucial for the performance of the algorithm because it determines the convergence speed and the solution quality to a great extent. In this paper, a heuristic algorithm is employed to generate the active schedule and no-delay schedule [28]. In addition, the random rule is also used to randomly select operations to generate the initial operation permutations.

The generation of active schedule can be shown as follows.

*Step 1.* Choose the schedulable operations to fill the set $\Omega$.

*Step 2.* Evaluate the earliest completion time $C_i$ of each operation in $\Omega$.

*Step 3.* Get the minimum value $C^* = \min\{C_i\}, i = 1, 2, \cdots, |\Omega|$ and the machine $M$ on which $C^*$ can be realized.

*Step 4.* Identify the operation set $L$ where the start time $S_i$ of each operation on machine $M$ is less than $C^*$.

*Step 5*. Choose one operation from $L$ and store it into set $SD$. If $|L| > 1$, a random number $l \in [0, 1]$ is generated. If $l < 0.5$, the SPT rule is employed to choose an operation from $L$; otherwise, the operation is chosen at random.

*Step 6*. Delete the chosen operation from $\Omega$, and add its immediate successor into $\Omega$.

*Step 7*. If all operations are not scheduled, go to Step 2; otherwise, output the scheduling result according to the selection order of operations in set $SD$.

No-delay schedules are produced by modifying Steps 2~4 in the above procedure.

*Step 2*. Evaluate the earliest start time $S_i$ of each operation in $\Omega$.

*Step 3*. Get the minimum value $S^* = \min\{S_i\}, i = 1, 2, \cdots, |\Omega|$ and the machine $M$ on which $S^*$ can be realized.

*Step 4*. Identify the operation set $L$ in which the start time $S_i$ of each operation on machine $M$ is equal to $S^*$.

*3.4. Tracing Mode.* In the original GWO, the searching behavior of each individual is mainly guided by $\alpha$, $\beta$, and $\delta$, which is called tracing mode here. As mentioned before, the original GWO can not be directly used to deal with the discrete scheduling problem. In the previous research, a discrete GWO was proposed where a new individual updating method was designed which can make the algorithm directly work in a discrete domain [29]. The updating method can be formulized by

$$\boldsymbol{X}_k(t+1) = \begin{cases} F(\boldsymbol{X}_k(t), \boldsymbol{X}_\alpha(t)), & rand \le \frac{1}{3} \\ F(\boldsymbol{X}_k(t), \boldsymbol{X}_\beta(t)), & \frac{1}{3} < rand < \frac{2}{3} \\ F(\boldsymbol{X}_k(t), \boldsymbol{X}_\delta(t)), & rand \ge \frac{2}{3} \end{cases} \quad (9)$$

$\boldsymbol{X}_k$ represents the discrete scheduling solution corresponding to $k$th wolf; $\boldsymbol{X}_\alpha, \boldsymbol{X}_\beta, \boldsymbol{X}_\delta$ define the scheduling solutions of $\alpha$, $\beta$, and $\delta$; $F$ defines the discrete crossover operation, and *rand* is a random number inside $[0, 1]$.

According to (9), the crossover operation is performed following the same probability between the current individual and the three best individuals. However, the hierarchical relationship among $\alpha$, $\beta$, and $\delta$ is not well reflected in the updating process. In this paper, we proposed an improved individual updating method in (10), by which the crossover probability will be adaptively adjusted according to the fitness values of the three best solutions:

$$\boldsymbol{X}_k(t+1) = \begin{cases} F(\boldsymbol{X}_k(t), \boldsymbol{X}_\alpha(t)), & 0 \le rand \le \tau_1 \\ F(\boldsymbol{X}_k(t), \boldsymbol{X}_\beta(t)), & \tau_1 < rand < \tau_1 + \tau_2 \\ F(\boldsymbol{X}_k(t), \boldsymbol{X}_\delta(t)), & \tau_1 + \tau_2 \le rand \le \tau_1 + \tau_2 + \tau_3 \end{cases} \quad (10)$$

$$\tau_1 = \frac{f_\alpha(t)}{f_\alpha(t) + f_\beta(t) + f_\delta(t)}$$
$$\tau_2 = \frac{f_\beta(t)}{f_\alpha(t) + f_\beta(t) + f_\delta(t)} \quad (11)$$
$$\tau_3 = \frac{f_\delta(t)}{f_\alpha(t) + f_\beta(t) + f_\delta(t)}$$

$$f = \frac{Z}{CS}, \quad Z \text{ is a constant.} \quad (12)$$

The precedence preserving order-based crossover (POX) [30] is also employed to implement the crossover operation in this study. In addition, $f$ is the fitness function, and $\tau$ is concerned with the crossover probability, i.e., $\tau_1 + \tau_2 + \tau_3 = 1$.

*3.5. Seeking Mode.* According to (10), it maintained the partial characteristics of the original GWO, but each current individual is just updated according to the information of the three best wolves $\alpha$, $\beta$, and $\delta$. It tends to result in the loss of population diversity and make the algorithm appear premature convergence. To overcome this drawback, a seeking mode is introduced to the proposed algorithm in order to enhance the randomicity and improve the global search ability. In this mode, the POX crossover is conducted between the current individual and a randomly selected individual. The updating method can be shown by (13), and $\boldsymbol{X}_{\text{rand}}$ represents the randomly selected individual:

$$\boldsymbol{X}_k(t+1) = F(\boldsymbol{X}_k(t), \boldsymbol{X}_{\text{rand}}(t)). \quad (13)$$

*3.6. Adaptive Selection Method of Searching Modes.* In this study, two searching modes are involved in our algorithm corresponding to local search and global search, respectively. It is well-known that an effective coordination between global search and local search can help the algorithm avoid the premature and obtain the rapid convergence. Therefore, we developed an adaptive selection method of searching modes, by which individuals are encouraged to explore the global search space at the early stage of the optimization, cluster around the local optimum, and exploit information to converge on the global optimum at the latter stage. Here, a selection method is developed in (14), where $pb$ is the selecting probability inside 0 and 1, $pb_{\max}$ and $pb_{\min}$ are the maximum and minimum values of $pb$. During the evolutionary process, a random number $rand' \in [0, 1]$ is generated for each individual. If $rand' \le pb$, the individual will be updated by the method in the seeking mode; otherwise, it will be updated by the method of in the tracing mode:

$$pb = pb_{\max} - (pb_{\max} - pb_{\min}) \times \frac{t}{t_{\max}}. \quad (14)$$

*3.7. Local Search.* In GWO, the search process is guided by the three best individuals ($\alpha$, $\beta$, and $\delta$) towards the potential optimum, which means that the quality of $\alpha$, $\beta$, and $\delta$ is crucial for the solution quality. Therefore, a local search strategy is performed on the three best individuals by considering their important effects. To implement the procedure, three neighborhood structures are used here, i.e., Swap, Insert, and Inverse [31]. The steps of the local search can be shown as below.

*Step 1.* Get the initial solution $\boldsymbol{X}'$, and set $g_{\max} \longleftarrow 3$, $\rho \longleftarrow 1$ and the maximum iteration $\rho_{\max}$.

*Step 2.* Set $g \longleftarrow 1$.

*Step 3.* Perform the procedure below until $g > g_{\max}$.

if $g = 1$ then $\boldsymbol{X}'' \longleftarrow \text{Swap}(\boldsymbol{X}')$
elseif $g = 2$ then $\boldsymbol{X}'' \longleftarrow \text{Insert}(\boldsymbol{X}')$
else $\boldsymbol{X}'' \longleftarrow \text{Inverse}(\boldsymbol{X}')$
endif
if $CS'(\boldsymbol{X}'') < CS'(\boldsymbol{X}')$ then $\boldsymbol{X}' \longleftarrow \boldsymbol{X}''$, $g \longleftarrow 1$
else $g \longleftarrow g + 1$
endif

*Step 4.* Set $\rho \longleftarrow \rho + 1$; if $\rho > \rho_{\max}$, go to Step 5; otherwise, go to Step 2.

*Step 5.* End the procedure and output the local optimal solution $\boldsymbol{X}'$.

*3.8. Steps of the DMGWO.* The steps of the proposed DMGWO can be shown as follows.

*Step 1.* Set the parameters of the algorithm and generate the initial population.

*Step 2.* Evaluate the individual fitness and find the three best individuals ($\alpha$, $\beta$, and $\delta$).

*Step 3.* Perform the local search to the best three individuals and update $\alpha$, $\beta$, and $\delta$.

*Step 4.* For each individual, generate a random number $rand'$. If $rand' \leq pb$, update the individual according to (13); otherwise, update the individual according to (10).

*Step 5.* Check the stopping criterion. If met, output the optimum and end the procedure; otherwise, go to Step 2.

## 4. Results and Discussion

To evaluate the performance of the proposed DMGWO, we coded the algorithm in FORTRAN and run it on VMware Workstation with 2GB main memory under WinXP.

In this section, extensive experiments have been conducted based on 43 benchmark instances to evaluate the effectiveness of our DMGWO algorithm. For each instance, 10 independent runs are conducted by different algorithms. Here, the processing times and routing are taken from the benchmark instances. In addition, $\lambda_j$ and $\eta_i$ are, respectively, selected from $[2, 5]$ and $[1, 3]$ with discrete uniform distribution. The due date data is set according to the method developed by Demirkol et al. [32], which can be shown by $d_i = (1 + (0.3 \times n)/m) \times \sum_{j=1}^{J_i} p_{ij}$, where $p_{ij}$ is the $j$th operation of Job $i$.

*4.1. Effectiveness of Improvement Strategies.* In this paper, we present several improvement strategies to enhance the search ability of the proposed algorithm, such as the double-searching mode and the new individual updating method in (10). Here, the effectiveness of these strategies is first verified in Table 1. In the table, instance names are shown in the first column, and computational results are reported in the following columns. 'SMGWO' represents the single-searching mode GWO, where only the tracing mode in (10) is used to update individuals. 'DMGWO-1' defines the algorithm with double-searching mode where (9) is used in the tracing mode. 'DMGWO' is the proposed algorithm in this study. In addition, '*Best*' is the best value in the ten runs of each algorithm. '*Avg*' is the average results of the ten runs. '*ARPD*' is the average relative percentage difference, i.e.,$ARPD = \sum_{r=1}^{R}((100 \times (Al_r - Min))/Min)/R$, where '*R*' is the number of runs, '*Min*' represents the minimum values obtained by all algorithms, and $Al_r$ is the obtained value in the $r$th run by each algorithm. '*Time*' is the average time in the ten runs (in seconds). Boldface represents the optimal value obtained by all compared algorithms. To facilitate the comparison, the same parameters are set for the three compared algorithms; i.e., population size is 200 and maximum iteration is 1000.

Seen from the experimental results in Table 1, it can be easily observed as follows. (1) In comparisons of the '*Best*' value, DMGWO obtains 22 optimal values, and the other two algorithms can only yield 12 optimal values. (2) In comparisons of the '*Avg*' value, DMGWO yields 30 optimal values, which is significantly better than other algorithms. (3) In comparisons of the '*ARPD*' value, DMGWO also performs better than other two algorithms. (4) In comparisons of the '*Time*' value, the three compared algorithms show almost the same performance.

Figures 3–5 are the Gantt chart of the DMGWO algorithm for LA17, LA30, and LA37.

To verify whether the differences from Table 1 are significant, an analysis of variance (ANOVA) test is performed in Table 2. The three compared algorithms are taken as levels and ARPD is viewed as the response variable. It is well-known that ANOVA should be conducted on the basis of some prerequisites: independent sample, homogeneity of variance, and normal distribution. As mentioned above, the compared algorithms are independently conducted to obtain the computational results. In Table 2, the variances of ARPD values obtained by these algorithms are 12.5, 16.7, and 10.9, respectively. Thus, the ratio of the greatest variance and the smallest one, i.e., 16.7/10.9=1.53, is smaller than 3, which meets the homogeneity of variance. For the normal distribution, it can be observed according to Figure 6. After checking the prerequisites, Table 2 indicates that there are significant differences among the algorithms as p-value is smaller than 0.05.

*4.2. Effectiveness of the Proposed DMGWO.* To further demonstrate the effectiveness of the proposed DMGWO, it is

Table 1: Effectiveness analysis of improvement strategy.

| Instancce | SMGWO | | | | DMGWO-1 | | | | DMGWO | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Best* | *Avg* | *ARPD* | *T* | *Best* | *Avg* | *ARPD* | *T* | *Best* | *Avg* | *ARPD* | *T* |
| FT06 | **372.3** | **378.3** | **1.61** | 5.4 | **372.3** | **378.3** | **1.61** | 5.5 | **372.3** | **378.3** | **1.61** | 5.5 |
| FT10 | 9594.7 | 10546.0 | 13.07 | 18.6 | **9327.2** | 10903.2 | 16.90 | 18.8 | 9511.2 | **10260.24** | **10.00** | 18.9 |
| FT20 | 10365.0 | 11802.3 | 15.90 | 20.8 | 10941.6 | 11898.5 | 16.84 | 20.9 | **10183.4** | **11489.3** | **12.82** | 21.4 |
| LA01 | 2932.8 | 3156.3 | 11.18 | 8.0 | **2839.0** | 3205.4 | 12.91 | 7.9 | 2865.4 | **3081.9** | **8.56** | 8.3 |
| LA02 | **2121.8** | **2415.1** | **13.82** | 7.9 | **2121.8** | 2533.0 | 19.38 | 8.1 | 2200.6 | 2529.2 | 19.20 | 8.0 |
| LA03 | **3543.6** | **3805.7** | **7.40** | 8.0 | 3812.0 | 4063.5 | 14.67 | 8.0 | 3744.2 | 3939.3 | 11.17 | 8.0 |
| LA04 | **2880.2** | 3246.6 | 12.72 | 8.0 | 3034.4 | 3276.4 | 13.76 | 8.0 | 2924.0 | **3115.9** | **8.18** | 8.0 |
| LA05 | 4005.2 | 4205.0 | 10.03 | 8.1 | **3821.8** | **4168.8** | **9.08** | 8.0 | 4059.6 | 4174.8 | 9.24 | 7.8 |
| LA06 | **6324.0** | **6935.9** | **9.68** | 13.8 | 6541.0 | 7227.8 | 14.29 | 13.8 | 6480.4 | 6947.0 | 9.85 | 13.8 |
| LA07 | **6225.3** | 7032.8 | 12.97 | 13.7 | 6325.7 | 6864.9 | 10.27 | 13.6 | 6467.9 | **6803.6** | **9.29** | 13.9 |
| LA08 | 5503.9 | 5816.1 | 12.94 | 13.6 | 5177.8 | 5944.8 | 15.44 | 13.8 | **5149.9** | **5697.1** | **10.63** | 13.1 |
| LA09 | 5929.3 | 6666.4 | 13.50 | 13.4 | 5944.5 | 6466.2 | 10.09 | 13.8 | **5873.6** | **6257.9** | **6.54** | 13.8 |
| LA10 | 7524.1 | 7900.8 | 5.41 | 13.4 | 7572.2 | 8167.5 | 8.96 | 13.7 | **7495.6** | **7853.4** | **4.77** | 13.8 |
| LA11 | **11758.6** | **12694.5** | **7.96** | 20.8 | 12073.8 | 12709.8 | 8.09 | 20.7 | 11940.6 | 12872.4 | 9.47 | 21.2 |
| LA12 | 9273.4 | 10230.3 | 10.68 | 21.0 | **9243.2** | 10013.1 | 8.33 | 20.7 | 9295.0 | **9974.3** | **7.91** | 21.3 |
| LA13 | 10760.6 | **11307.2** | **7.08** | 20.8 | 10885.8 | 11673.8 | 10.56 | 20.7 | **10559.2** | 11390.6 | 7.87 | 21.4 |
| LA14 | 15202.6 | 16236.1 | 11.24 | 21.1 | 14867.8 | 16058.8 | 10.03 | 20.9 | **14595.4** | **15989.2** | **9.55** | 21.6 |
| LA15 | 12442.0 | 13215.2 | 6.48 | 20.9 | 12504.4 | 13763.6 | 10.89 | 20.9 | **12411.4** | **13008.0** | **4.81** | 21.8 |
| LA16 | 12114.8 | 13060.7 | 9.92 | 18.8 | **11882.4** | 12761.2 | 7.40 | 19.0 | 11979.8 | **12690.6** | **6.80** | 19.5 |
| LA17 | 11141.3 | 11889.0 | 12.10 | 18.4 | 10882.0 | 11667.9 | 10.01 | 18.8 | **10606.0** | **11558.0** | **8.98** | 18.7 |
| LA18 | 10241.1 | 10856.3 | 15.11 | 18.9 | 9901.8 | **10472.8** | **11.04** | 18.3 | **9431.4** | 10564.0 | 12.01 | 18.6 |
| LA19 | 8654.7 | 9450.7 | 11.94 | 18.6 | 8701.7 | 9489.2 | 12.39 | 18.8 | **8443.0** | **9156.8** | **8.45** | 18.6 |
| LA20 | 13571.8 | 14711.0 | 15.66 | 18.3 | **12719.0** | 14440.0 | 13.53 | 18.9 | 13320.0 | **14149.4** | **11.25** | 18.6 |
| LA21 | 12742.3 | 13942.8 | 16.04 | 32.3 | **12015.5** | 13866.3 | 15.40 | 33.0 | 12320.6 | **13344.7** | **11.06** | 34.1 |
| LA22 | 13293.2 | 14564.3 | 10.61 | 33.0 | 13243.3 | 14618.8 | 11.02 | 33.5 | **13167.4** | **14102.6** | **7.10** | 33.6 |
| LA23 | 10561.8 | 11179.8 | 11.16 | 32.9 | 10852.2 | 11428.7 | 13.64 | 33.6 | **10057.2** | **11101.7** | **10.39** | 33.5 |
| LA24 | 10071.8 | **11378.3** | **13.49** | 32.8 | 11250.8 | 12485.2 | 24.53 | 33.4 | **10025.5** | 11501.1 | 14.72 | 33.6 |
| LA25 | **11742.4** | 13810.7 | 17.61 | 32.4 | 11795.1 | **13763.7** | **17.21** | 32.7 | 12546.1 | 13800.9 | 17.53 | 33.6 |
| LA26 | 16026.0 | **16958.9** | **6.43** | 48.6 | 16170.2 | 17820.8 | 11.84 | 48.3 | **15934.2** | 17142.0 | 7.58 | 50.7 |
| LA27 | 16684.2 | 18170.7 | 11.08 | 49.7 | **16358.4** | 17703.3 | 8.22 | 49.7 | 16796.6 | **17551.7** | **7.29** | 50.0 |
| LA28 | 17072.8 | 18582.3 | 16.71 | 45.3 | 16470.2 | 18515.4 | 16.29 | 49.5 | **15922.4** | **18244.7** | **14.59** | 49.7 |
| LA29 | 19252.2 | 20467.4 | 9.49 | 48.4 | 18907.8 | 20670.0 | 10.57 | 49.2 | **18694.2** | **20205.0** | **8.08** | 49.3 |
| LA30 | **18855.8** | 21493.5 | 13.99 | 49.1 | 19047.0 | 21798.4 | 15.61 | 49.4 | 20023.4 | **20896.1** | **10.82** | 49.2 |
| LA31 | 36280.3 | 37608.1 | 8.11 | 87.7 | 36145.3 | 38073.7 | 9.45 | 87.5 | **34787.6** | **36839.3** | **5.90** | 88.2 |
| LA32 | **34144.8** | **36493.9** | **6.88** | 89.4 | 35691.7 | 37758.0 | 10.58 | 88.2 | 36402.4 | 37161.4 | 8.83 | 87.5 |
| LA33 | **32832.7** | 35563.9 | 8.32 | 89.2 | 33602.5 | 36491.1 | 11.14 | 88.3 | 33466.2 | **35538.7** | **8.24** | 87.4 |
| LA34 | 35653.3 | 37131.8 | 8.96 | 89.7 | 34552.2 | 36547.6 | 7.25 | 92.5 | **34077.1** | **36436.9** | **6.92** | 92.2 |
| LA35 | 38859.1 | 41906.7 | 8.82 | 90.5 | 38769.9 | 40991.1 | 6.44 | 91.5 | **38511.5** | **40783.8** | **5.90** | 93.3 |
| LA36 | 32196.6 | 33621.6 | 8.21 | 59.2 | 32513.8 | 33776.8 | 8.71 | 59.6 | **31070.4** | **33365.9** | **7.39** | 60.6 |
| LA37 | 32316.5 | 34618.5 | 9.53 | 58.2 | **31605.3** | 33625.9 | 6.39 | 58.6 | 32082.3 | **32980.9** | **4.35** | 59.9 |
| LA38 | **28785.5** | **30382.9** | **5.55** | 57.1 | 28418.5 | 31090.0 | 8.01 | 56.8 | 29835.1 | 30725.1 | 6.74 | 58.8 |
| LA39 | 23805.1 | 25619.0 | 11.92 | 57.9 | **22890.1** | **24903.7** | **8.80** | 57.4 | 24139.1 | 24966.4 | 9.07 | 58.0 |
| LA40 | 26144.0 | 28031.2 | 16.36 | 59.7 | 26108.8 | 27191.0 | 12.87 | 58.4 | **24091.0** | **26508.0** | **10.03** | 58.3 |

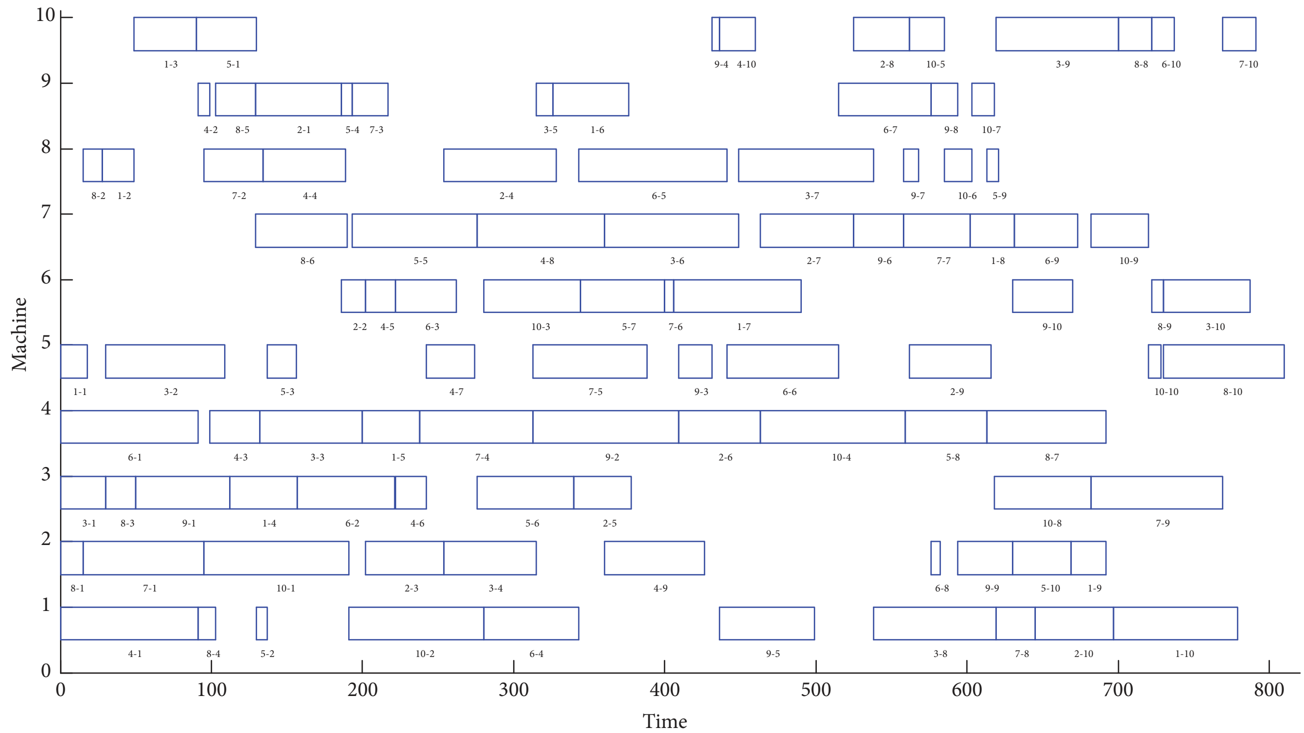


FIGURE 3: The Gantt chart of LA17.

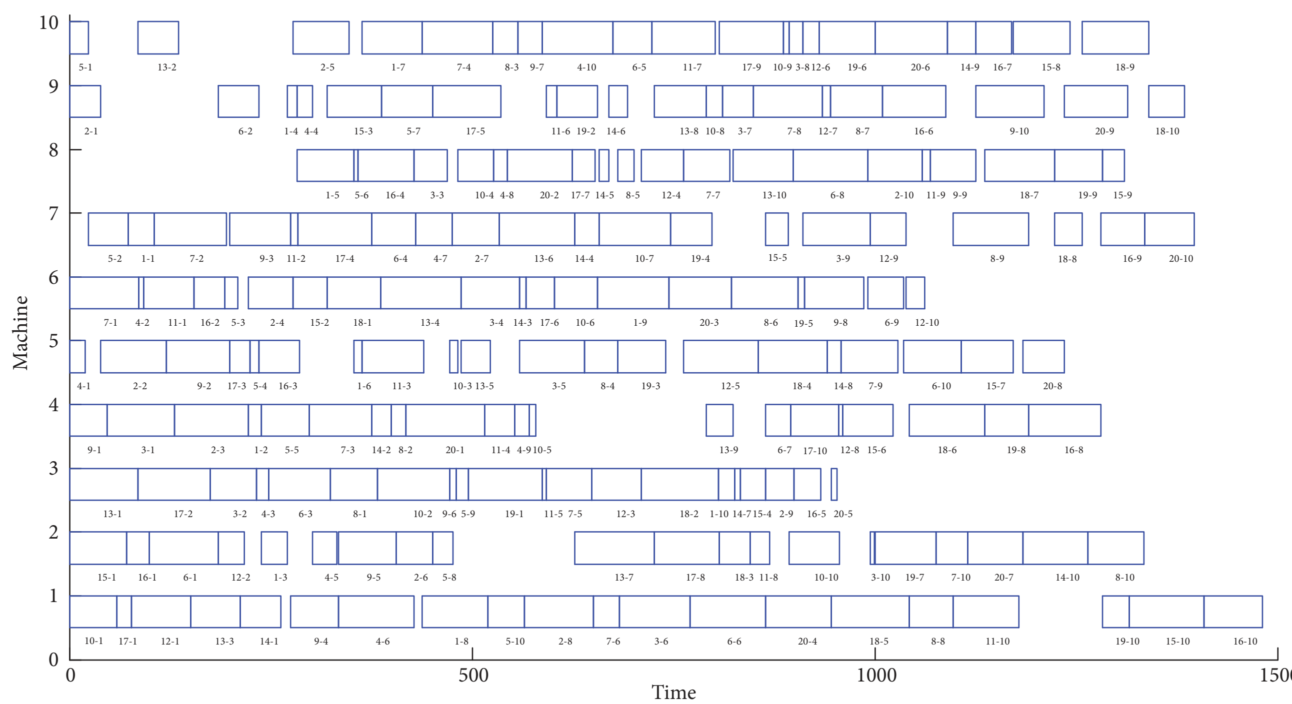


FIGURE 4: The Gantt chart of LA30.

compared with TLBO and HGWO in this subsection. TLBO is the teaching-learning based optimization algorithm for job shop scheduling problem proposed by Baykasoglu et al. [2]. HGWO is a modified version of the hybrid GWO algorithm for flexible job shop scheduling problem proposed by Jiang [33], where the machine selection segment is excluded from the scheduling solution, the population initialization method in this paper is adopted to create the initial solutions, and the variable neighborhood search (VNS) takes place by the local search in [33] to enhance the local search ability. To

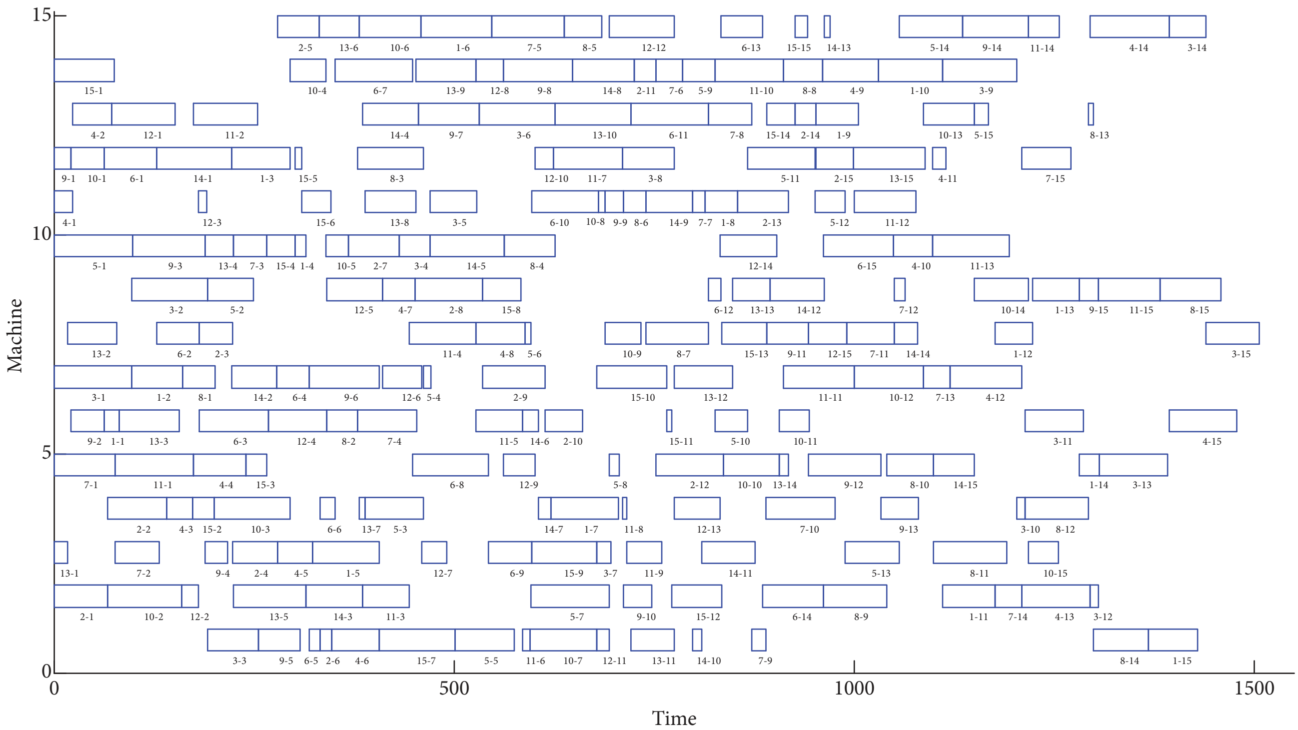


Figure 5: The Gantt chart of LA37.

Table 2: ANOVA for ARPD of the compared algorithms.

| Source | DF | Sum of Squares | Mean Square | F | p-value |
|---|---|---|---|---|---|
| Factor | 2 | 145.32843 | 72.66422 | 5.44155 | 0.00541 |
| Error | 126 | 1682.55354 | 13.3536 | | |
| Total | 128 | 1827.88197 | | | |

facilitate the comparison, the population size and the maximum iteration of the compared algorithms are the same as those of DMGWO; i.e., population size is 200, and the maximum of iteration is 1000. In addition, the crossover and mutation rates of HGWO are 0.8 and 0.1, and the maximum iteration of local search is 20.

Seen from the experimental results in Table 3, it can be easily observed as follows. (1) In comparisons of the ‘*Best*’ value, DMGWO yields 32 optimal values, which is significantly better than the other two algorithms. (2) In comparisons of the ‘*Avg*’ value, DMGWO yields 34 optimal values, which is far more than those of the other algorithms. (3) In comparisons of the ‘*ARPD*’ value, DMGWO also performs better than the other two algorithms and obtains 34 optimal values. (4) In comparisons of the ‘*Time*’ value, the proposed DMGWO spends a shorter time for each instance.

To statistically analyze the results in Table 3, an analysis of variance (ANOVA) test is conducted in Table 4. Each algorithm is taken as a level and ARPD is the response variable. The prerequisites of ANOVA can be checked by the above method. Here, to satisfy the homogeneity of variance, the logarithmic transformation is first executed on the ARPD values obtained by the three algorithms. The variances of converted values are 0.22, 0.25, and 0.25, respectively. Thus, the ratio of the greatest variance and the smallest one, i.e., 0.25/0.22=1.14, is smaller than 3, which meets the homogeneity of variance. In addition, Figure 7 is the histogram of the ARPD, which meets the normal distribution. After checking the prerequisites, Table 4 indicates that there are significant differences among the algorithms because p-value is equal to 0.

## 5. Conclusions

In this paper, a kind of grey wolf optimization algorithm with double-searching mode (DMGWO) is presented to solve the energy-efficient job shop scheduling problem.

(1) At the initialization phase, a discrete encoding mechanism is first employed to represent the scheduling solution, and a heuristic-based initialization method is used to ensure the quality and diversity of the initial population. Two searching modes, named tracing mode and seeking mode, are used to perform global search and local search simultaneously. An adaptive selection method of search modes is developed to balance the exploration and exploitation during

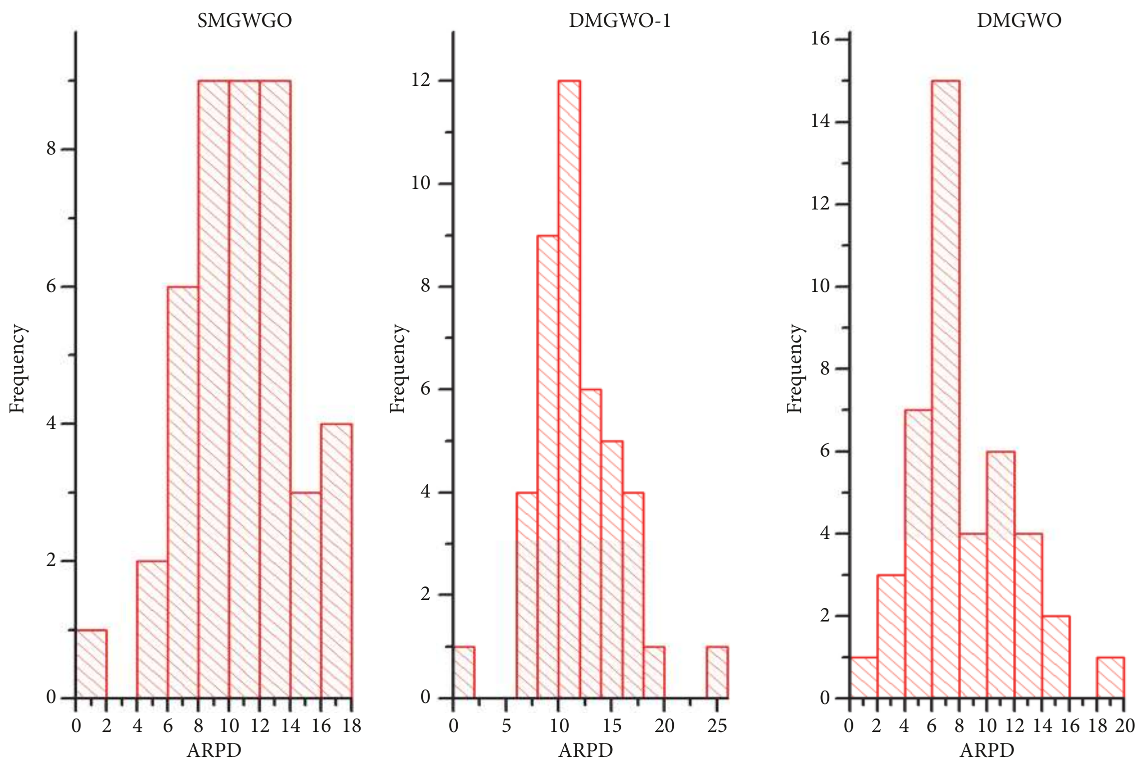


Figure 6: Histogram of the ARPD obtained by SMGWO, DMGWO-1, and DMGWO.

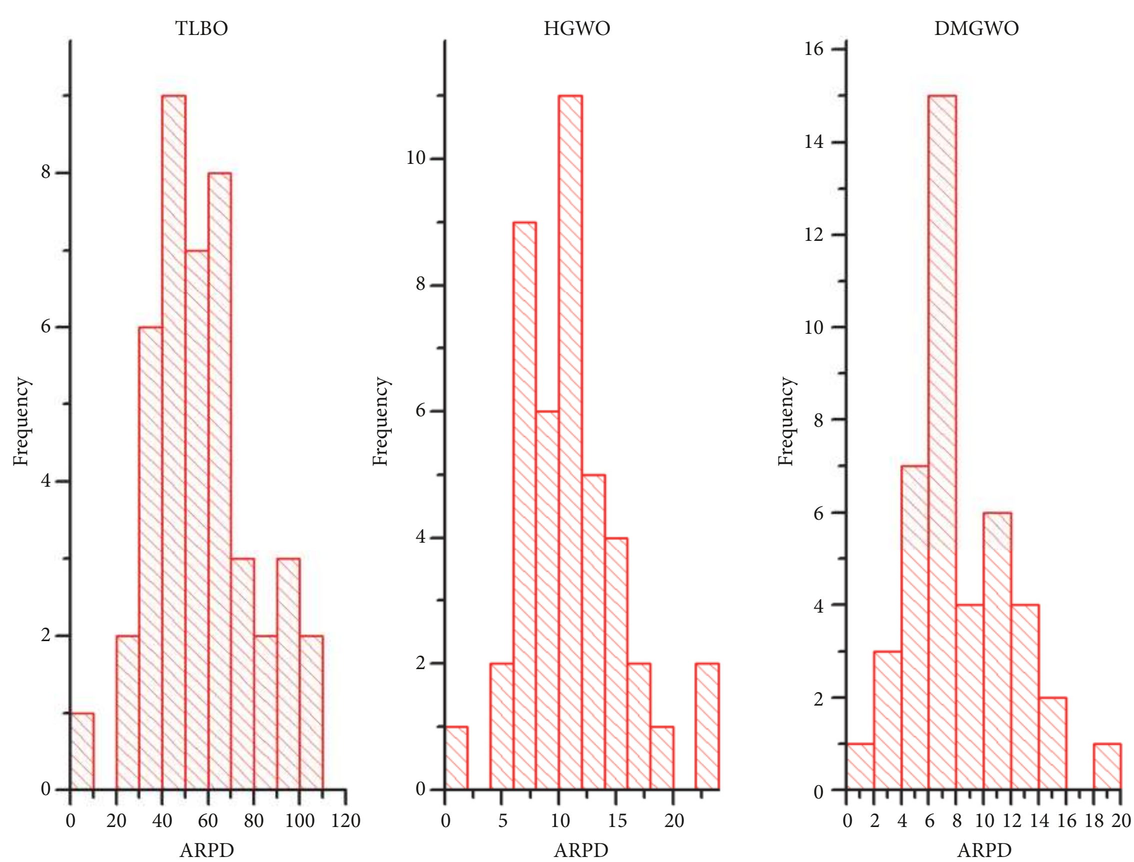


Figure 7: Histogram of the ARPD obtained by TLBO, HGWO, and DMGWO.

Table 3: Comparison results for different algorithms.

| Index | Instancce | TLBO | | | | HGWO | | | | DMGWO | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *Best* | *Avg* | *ARPD* | *T* | *Best* | *Avg* | *ARPD* | *T* | *Best* | *Avg* | *ARPD* | *T* |
| 1 | FT06 | **372.3** | 400.3 | 7.52 | 37.1 | **372.3** | **376.5** | **1.13** | 13.0 | **372.3** | 378.3 | 1.61 | 5.5 |
| 2 | FT10 | 13155.8 | 14211.8 | 49.42 | 189.6 | 9822.7 | 10634.3 | 11.81 | 65.0 | **9511.2** | **10260.24** | **7.88** | 18.9 |
| 3 | FT20 | 14375.4 | 17241.7 | 69.31 | 345.6 | 11516.8 | 12552.7 | 23.27 | 66.0 | **10183.4** | **11489.3** | **12.82** | 21.4 |
| 4 | LA01 | 3963.2 | 4727.1 | 64.97 | 80.0 | **2865.4** | 3130.9 | 9.27 | 21.4 | **2865.4** | **3081.9** | **7.56** | 8.3 |
| 5 | LA02 | 3531.8 | 3956.3 | 86.46 | 79.4 | **2121.8** | **2509.1** | **18.25** | 21.3 | 2200.6 | 2529.2 | 19.20 | 8.0 |
| 6 | LA03 | 4777.2 | 5112.9 | 47.58 | 77.0 | **3464.6** | **3908.6** | **12.82** | 21.1 | 3744.2 | 3939.3 | 13.70 | 8.0 |
| 7 | LA04 | 4364.8 | 4622.5 | 58.09 | 77.8 | 2934.2 | 3262.3 | 11.57 | 21.0 | **2924.0** | **3115.9** | **6.56** | 8.0 |
| 8 | LA05 | 4547.0 | 4755.8 | 22.93 | 81.2 | **3868.6** | **4149.4** | **7.26** | 21.2 | 4059.6 | 4174.8 | 7.92 | 7.8 |
| 9 | LA06 | 8135.4 | 9242.1 | 42.62 | 178.2 | 6942.3 | 7253.5 | 11.93 | 40.2 | **6480.4** | **6947.0** | **7.20** | 13.8 |
| 10 | LA07 | 8899.9 | 9418.7 | 45.62 | 176.4 | 6665.3 | 7078.2 | 9.44 | 40.3 | **6467.9** | **6803.6** | **5.19** | 13.9 |
| 11 | LA08 | 7777.6 | 8474.6 | 64.56 | 184.1 | 5381.6 | 5855.6 | 13.70 | 40.3 | **5149.9** | **5697.1** | **10.63** | 13.1 |
| 12 | LA09 | 8593.2 | 9496.0 | 61.67 | 177.4 | 6004.5 | 6779.3 | 15.42 | 39.6 | **5873.6** | **6257.9** | **6.54** | 13.8 |
| 13 | LA10 | 9648.7 | 10303.0 | 38.17 | 171.9 | **7456.6** | 7917.5 | 6.18 | 39.5 | 7495.6 | **7853.4** | **5.32** | 13.8 |
| 14 | LA11 | 15544.2 | 16590.8 | 38.94 | 333.2 | 12795.2 | 13203.2 | 10.57 | 64.7 | **11940.6** | **12872.4** | **7.80** | 21.2 |
| 15 | LA12 | 12664.4 | 13228.8 | 42.32 | 331.4 | 9416.6 | **9901.5** | **6.53** | 65.7 | **9295.0** | 9974.3 | 7.31 | 21.3 |
| 16 | LA13 | 15006.2 | 15563.2 | 47.39 | 338.8 | 10808.4 | 11580.1 | 9.67 | 64.9 | **10559.2** | **11390.6** | **7.87** | 21.4 |
| 17 | LA14 | 17479.0 | 18805.6 | 29.54 | 333.9 | **14517.4** | **15542.4** | **7.06** | 65.5 | 14595.4 | 15989.2 | 10.14 | 21.6 |
| 18 | LA15 | 16251.8 | 16993.9 | 36.92 | 334.7 | 12425.8 | 13400.6 | 7.97 | 64.4 | **12411.4** | **13008.0** | **4.81** | 21.8 |
| 19 | LA16 | 14492.9 | 15869.3 | 32.47 | 182.0 | **11979.8** | 13086.8 | 9.24 | 63.1 | **11979.8** | **12690.6** | **5.93** | 19.5 |
| 20 | LA17 | 13146.6 | 14391.7 | 35.69 | 182.9 | **10606.0** | **11490.1** | **8.34** | 62.0 | **10606.0** | 11558.0 | 8.98 | 18.7 |
| 21 | LA18 | 12855.4 | 14588.5 | 54.68 | 183.7 | 10241.1 | 10688.2 | 13.33 | 62.2 | **9431.4** | **10564.0** | **12.01** | 18.6 |
| 22 | LA19 | 12475.0 | 13215.9 | 56.53 | 185.6 | **8443.0** | 9484.7 | 12.34 | 61.9 | **8443.0** | **9156.8** | **8.45** | 18.6 |
| 23 | LA20 | 15653.6 | 17613.8 | 32.24 | 181.3 | 13521.8 | 14362.1 | 7.82 | 63.1 | **13320.0** | **14149.4** | **6.23** | 18.6 |
| 24 | LA21 | 21510.9 | 22242.2 | 86.38 | 426.6 | **11934.1** | 13853.9 | 16.09 | 122.7 | 12320.6 | **13344.7** | **11.82** | 34.1 |
| 25 | LA22 | 21873.5 | 22429.0 | 70.34 | 422.8 | 13476.6 | 14466.0 | 9.86 | 116.3 | **13167.4** | **14102.6** | **7.10** | 33.6 |
| 26 | LA23 | 19924.8 | 21070.1 | 109.50 | 417.2 | 10311.1 | 11248.6 | 11.85 | 117.0 | **10057.2** | **11101.7** | **10.39** | 33.5 |
| 27 | LA24 | 18822.2 | 20179.1 | 101.28 | 431.0 | 10271.5 | 12389.9 | 23.58 | 119.2 | **10025.5** | **11501.1** | **14.72** | 33.6 |
| 28 | LA25 | 20193.5 | 22058.7 | 75.82 | 419.4 | 13110.0 | 14392.6 | 14.72 | 117.7 | **12546.1** | **13800.9** | **10.00** | 33.6 |
| 29 | LA26 | 27738.2 | 30390.6 | 99.74 | 817.7 | **15215.0** | 17455.7 | 14.73 | 197.7 | 15934.2 | **17142.0** | **12.67** | 50.7 |
| 30 | LA27 | 27596.2 | 30652.0 | 91.42 | 800.3 | **16013.2** | 17738.3 | 10.77 | 203.2 | 16796.6 | **17551.7** | **9.61** | 50.0 |
| 31 | LA28 | 28440.4 | 31144.0 | 96.00 | 803.4 | **15889.8** | **18211.8** | **14.61** | 203.9 | 15922.4 | 18244.7 | 14.82 | 49.7 |
| 32 | LA29 | 28901.6 | 31343.6 | 69.21 | 790.3 | **18524.0** | 20412.4 | 10.19 | 200.5 | 18694.2 | **20205.0** | **9.07** | 49.3 |
| 33 | LA30 | 32110.6 | 33845.3 | 69.03 | 805.0 | 20429.0 | 21392.4 | 6.84 | 207.1 | **20023.4** | **20896.1** | **4.36** | 49.2 |
| 34 | LA31 | 50140.7 | 54193.6 | 55.78 | 2137.1 | 36974.3 | 39046.6 | 12.24 | 434.4 | **34787.6** | **36839.3** | **5.90** | 88.2 |
| 35 | LA32 | 55044.1 | 56783.2 | 55.99 | 2011.3 | 37168.4 | 38154.3 | 4.81 | 444.6 | **36402.4** | **37161.4** | **2.09** | 87.5 |
| 36 | LA33 | 49378.8 | 52924.8 | 58.14 | 1674.4 | 34788.7 | 36123.5 | 7.94 | 425.7 | **33466.2** | **35538.7** | **6.19** | 87.4 |
| 37 | LA34 | 51422.3 | 54523.5 | 60.00 | 1857.8 | 34494.4 | 37743.5 | 10.76 | 434.0 | **34077.1** | **36436.9** | **6.92** | 92.2 |
| 38 | LA35 | 57561.1 | 59842.1 | 55.39 | 1794.1 | 40544.9 | 42996.0 | 11.64 | 434.4 | **38511.5** | **40783.8** | **5.90** | 93.3 |
| 39 | LA36 | 43779.1 | 45924.3 | 47.81 | 587.3 | 32033.0 | 34732.9 | 11.79 | 251.7 | **31070.4** | **33365.9** | **7.39** | 60.6 |
| 40 | LA37 | 43986.0 | 46232.3 | 44.11 | 596.9 | 32552.6 | 33660.8 | 4.92 | 253.5 | **32082.3** | **32980.9** | **2.80** | 59.9 |
| 41 | LA38 | 39819.45 | 42084.6 | 47.51 | 634.3 | **28530.4** | **30706.1** | **7.63** | 249.2 | 29835.1 | 30725.1 | 7.69 | 58.8 |
| 42 | LA39 | 39469.9 | 42439.5 | 75.81 | 611.3 | 24615.9 | 26750.9 | 10.82 | 254.4 | **24139.1** | **24966.4** | **3.43** | 58.0 |
| 43 | LA40 | 38279.0 | 40019.4 | 66.12 | 640.3 | 24942.3 | 28096.8 | 16.63 | 255.4 | **24091.0** | **26508.0** | **10.03** | 58.3 |

Table 4: ANOVA for converted values of ARPD.

| Source | DF | Sum of Squares | Mean Square | F | p-value |
|---|---|---|---|---|---|
| Factor | 2 | 96.71517 | 48.35758 | 203.28664 | 0 |
| Error | 126 | 29.97273 | 0.23788 | | |
| Total | 128 | 126.6879 | | | |

the evolutionary process. In addition, a local search strategy is embedded to further enhance the solution quality of the algorithm.

(2) A number of experiments based on 43 benchmark instances are carried out. The effectiveness of improvement strategies, e.g., a double-searching mode and a new individual updating method, is first verified by extensive experiments. Then the proposed DMGWO is compared with two published algorithms. According to the comparisons results, the proposed DMGWO is effective for solving the energy-efficient job shop scheduling problem under study.

(3) In the future work, the energy-efficient JSP will be further studied by considering some practical constraints, e.g., adjustable processing speed of machines, time-of-use electricity policy, etc. In addition, the energy-efficient scheduling problem will be considered in some more complex workshop, such as flexible job shop and assembly job shop, and so on.

## Data Availability

The data of benchmark instances for the purposes of optimization are included in the OR-Library [34].

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

This work is supported in part by the Training Foundation of Shandong Natural Science Foundation of China under Grant ZR2016GP02, the National Natural Science Foundation Project of China under Grant 61403180, the Project of Shandong Province Higher Educational Science and Technology Program under Grant J17KA199, the Special Research and Promotion Program of Henan Province under Grant 182102210257, the Project of Henan Province Higher Educational Key Research Program under Grant 16A120011, and the Talent Introduction Research Program of Ludong University under Grant 32860301.

## References

[1] R. Zhang, S. Song, and C. Wu, "A hybrid artificial bee colony algorithm for the job shop scheduling problem," *International Journal of Production Economics*, vol. 141, no. 1, pp. 167–178, 2013.

[2] A. Baykasoğlu, A. Hamzadayi, and S. Y. Köse, "Testing the performance of teaching-learning based optimization (TLBO) algorithm on combinatorial problems: flow shop and job shop scheduling cases," *Information Sciences*, vol. 276, pp. 204–218, 2014.

[3] C. Bierwirth and J. Kuhpfahl, "Extended GRASP for the job shop scheduling problem with total weighted tardiness objective," *European Journal of Operational Research*, vol. 261, no. 3, pp. 835–848, 2017.

[4] R. Zhang and J. Kuhpfahl, "Erratum: A simulated annealing algorithm based on block properties for the job shop scheduling problem with total weighted tardiness objective," *Computers & Operations Research*, vol. 38, no. 5, pp. 854–867, 2013.

[5] R. Tavakkoli-Moghaddam, M. Azarkish, and A. Sadeghnejad-Barkousaraie, "A new hybrid multi-objective Pareto archive PSO algorithm for a bi-objective job shop scheduling problem," *Expert Systems with Applications*, vol. 38, no. 9, pp. 10812–10821, 2011.

[6] A. Udomsakdigool and V. Khachitvichyanukul, "Ant colony algorithm for multi-criteria job shop scheduling to minimize makespan, mean flow time and mean tardiness," *International Journal of Management Science and Engineering Management*, vol. 6, no. 2, pp. 116–122, 2011.

[7] 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.

[8] 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.

[9] 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.

[10] D. Tang, M. Dai, M. A. Salido, and A. Giret, "Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization," *Computers in Industry*, vol. 81, pp. 82–95, 2016.

[11] 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.

[12] J. Y. Ding, S. Song, and C. Wu, "Carbon-efficient scheduling of flow shops by multi-objective optimization," *European Journal of Operational Research*, vol. 248, no. 3, pp. 758–771, 2016.

[13] Y. Z. Ai and D. M. Lei, "Novel neighborhood search for low carbon scheduling of hybrid flow shop with carbon emission," *Information and Control*, vol. 46, no. 3, pp. 311–317.

[14] 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, pp. 1304–1314, 2016.

[15] R. Zhang and R. Chiong, "Solving the energy-efficient job shop scheduling problem: A multi-objective genetic algorithm with

enhanced local search for minimizing the total weighted tardiness and total energy consumption," *Journal of Cleaner Production*, vol. 112, pp. 3361–3375, 2016.

[16] G. May, B. Stahl, M. Taisch, and V. Prabhu, "Multi-objective genetic algorithm for energy-efficient job shop scheduling," *International Journal of Production Research*, vol. 53, no. 23, pp. 7071–7089, 2015.

[17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[18] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, 2016.

[19] S. Mohanty, B. Subudhi, and P. K. Ray, "A new MPPT design using grey Wolf optimization technique for photovoltaic system under partial shading conditions," *IEEE Transactions on Sustainable Energy*, vol. 7, no. 1, pp. 181–188, 2016.

[20] P. Yao and L. Wang H, "Three-dimensional path planning for UAV based on improved interfered fluid dynamical system and grey wolf optimizer," *Control and Desicion*, vol. 31, no. 4, pp. 701–708, 2016.

[21] A. A. Heidari and P. Pahlavani, "An efficient modified grey wolf optimizer with Lévy flight for optimization tasks," *Applied Soft Computing*, vol. 60, pp. 115–134, 2017.

[22] R. A. Ibrahim, M. A. Elaziz, and S. Lu, "Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization," *Expert Systems with Applications*, vol. 108, pp. 1–27, 2018.

[23] M. Naz, Z. Iqbal, N. Javaid et al., "Efficient Power Scheduling in Smart Homes Using Hybrid Grey Wolf Differential Evolution Optimization Technique with Real Time and Critical Peak Pricing Schemes," *Energies*, vol. 11, no. 2, p. 384, 2018.

[24] C. Lu, S. Xiao, X. Li, and L. Gao, "An effective multi-objective discrete grey Wolf optimizer for a real-world scheduling problem in welding production," *Advances in Engineering Software*, vol. 99, pp. 161–176, 2016.

[25] C. Lu, L. Gao, X. Li, and S. Xiao, "A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry," *Engineering Applications of Artificial Intelligence*, vol. 57, pp. 61–79, 2017.

[26] G. M. Komaki and V. Kayvanfar, "Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time," *Journal of Computational Science*, vol. 8, pp. 109–120, 2015.

[27] D. Maharana and P. Kotecha, "Optimization of Job Shop Scheduling Problem with Grey Wolf Optimizer and JAYA Algorithm," in *Smart Innovations in Communication and Computational Sciences*, vol. 669 of *Advances in Intelligent Systems and Computing*, pp. 47–58, Springer, Singapore, 2019.

[28] S. Zhao and S. Fang, "Operation-based encoding and neighborhood search genetic algorithm for job shop scheduling optimization," *Jixie Gongcheng Xuebao/Journal of Mechanical Engineering*, vol. 49, no. 16, pp. 160–169, 2013.

[29] T. H. Jiang and C. Zhang, "Application of Grey Wolf Optimization for Solving Combinatorial Problems: Job Shop and Flexible Job Shop Scheduling Cases," *IEEE Access*, vol. 6, pp. 26231–26240, 2018.

[30] Y. Demir and S. K. Işleyen, "An effective genetic algorithm for flexible job-shop scheduling with overlapping in operations," *International Journal of Production Research*, vol. 52, no. 13, pp. 3905–3921, 2014.

[31] H. Wang, W. Wang, H. Sun, Z. Cui, S. Rahnamayan, and S. Zeng, "A new cuckoo search algorithm with hybrid strategies for flow shop scheduling problems," *Soft Computing*, vol. 21, no. 15, pp. 4297–4307, 2017.

[32] E. Demirkol, S. Mehta, and R. Uzsoy, "Benchmarks for shop scheduling problems," *European Journal of Operational Research*, vol. 109, no. 1, pp. 137–141, 1998.

[33] T. H. Jiang, "Flexible job shop scheduling problem with hybrid grey wolf optimization algorithm," *Control and Desicion*, vol. 33, no. 3, pp. 503–508, 2018.

[34] J. E. Beasley, "OR-Library: distributing test problems by electronic mail," *Journal of the Operational Research Society*, vol. 41, no. 11, pp. 1069–1072, 1990.