

Research Article

Energy Saving in Flow-Shop Scheduling Management: An Improved Multiobjective Model Based on Grey Wolf Optimization Algorithm

Lvjiang Yin ¹, Meier Zhuang ², Jing Jia ¹, and Huan Wang ¹

¹School of Economics and Management, Hubei University of Automotive Technology, Shiyan 442002, China

²School of Business, Guangdong University of Foreign Studies, Guangzhou 510006, China

Correspondence should be addressed to Meier Zhuang; 990883130@qq.com

Received 16 June 2020; Accepted 21 August 2020; Published 14 October 2020

Guest Editor: Weilin Xiao

Copyright © 2020 Lvjiang Yin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Currently, energy saving is increasingly important. During the production procedure, energy saving can be achieved if the operational method and machine infrastructure are improved, but it also increases the complexity of flow-shop scheduling. Actually, as one of the data mining technologies, Grey Wolf Optimization Algorithm is widely applied to various mathematical problems in engineering. However, due to the immaturity of this algorithm, it still has some defects. Therefore, we propose an improved multiobjective model based on Grey Wolf Optimization Algorithm related to Kalman filter and reinforcement learning operator, where Kalman filter is introduced to make the solution set closer to the Pareto optimal front end. By means of reinforcement learning operator, the convergence speed and solving ability of the algorithm can be improved. After testing six benchmark functions, the results show that it is better than that of the original algorithm and other comparison algorithms in terms of search accuracy and solution set diversity. The improved multiobjective model based on Grey Wolf Optimization Algorithm proposed in this paper is conducive to solving energy saving problems in flow-shop scheduling problem, and it is of great practical value in engineering and management.

1. Introduction

Many mathematical problems in scientific research and practical engineering essentially belong to multiobjective optimization problem. The analysis of constrained multiobjective optimization algorithm has become a research hotspot in recent years.

Different theories exist in the literature regarding optimization algorithm such as the Improved Multiobjective Grey Wolf Optimizer (IMOGWO) that hybridize with the fast nondominated sorting strategy [1]. In fact, significant progress has been made in solving constrained multiobjective optimization problems at home and abroad due to its efficiency and simplicity [2], but there is still much room for improvement in terms of the diversity and convergence of solution sets.

In previous research, some scholars proposed a differential evolution algorithm based on two-population search mechanism, which randomly deletes one of the two individuals with the smallest Euclidean distance [3]. In this way, the boundary solution may be lost, and the diversity of solution set may be affected. At the same time, when updating the infeasible solution set, the individuals with a small degree of constraint violation are preferred, but the individual objective function value selected in this way may be poor, thus affecting the convergence speed of the algorithm.

Several lines of evidence suggest that a number of penalty terms were applied to modify the value of individual objective function. In the process of evolution, feasible non-dominant solutions were retained, and infeasible solutions with a small degree of constraint violation were also retained

[4]. This algorithm also has the condition of losing boundary solution, which indicates that it has some defects in the diversity maintenance of solution set. When updating feasible solution sets and infeasible solution sets, individuals located in the sparse region are given priority. However, when updating the infeasible solution sets, individuals with a large degree of constraint violation will be retained, thus reducing the convergence speed of the algorithm [5].

As for the improved elite selection strategy, it can make the solution set more widely distributed by setting preference points and expand the application of constrained multiobjective optimization algorithm to high-dimensional problems by combining with Deb criterion [6], but it still has some drawbacks.

Up to now, plenty of differential evolution algorithms have been proposed, which minimize the value of the objective function for the feasible solution and minimize the degree of constraint violation for the infeasible solution [7]. However, the information interaction between the feasible solution set and the infeasible solution set is insufficient, and the population diversity needs to be improved. On the other hand, some scholars proposed a constrained multiobjective optimization algorithm based on adaptive e truncation strategy, which can improve the diversity of solution sets and give consideration to the convergence [8]. But the selection of e parameters is difficult, which needs to be adjusted according to different problems [9]. In brief, it is difficult for most algorithms to achieve enough balance for the key performance indexes in the constrained multiobjective optimization problems in terms of diversity and convergence [10–12].

Therefore, we propose an improved Multiobjective Grey Wolf Optimizer related to Kalman filtering and reinforcement learning (MKGWO) in this paper. The main innovation of the algorithm is that Kalman filter facilitates the convergence from solution set to Pareto optimal front end introduced into the static multiobjective algorithm. It combines the characteristics of Kalman filter with the robustness, reliability, and high efficiency of the reinforcement learning system when solving problems [13]. In the process of evolution, the algorithm uses an elite population to store feasible nondominant solutions and retains the nondominant solutions generated by historical iteration. The problem of scheduling research is to allocate scarce resources to different tasks within a certain period of time [14, 15]. As the production with the continuous expansion of scale, the importance of scheduling and decision-making to enterprise management and production is increasingly prominent [16].

The scheduling problem is an interdisciplinary field of research, which involves operations research, computer science, control theory, industrial engineering, and many other disciplines [17]. A good scheduling scheme can greatly improve the production level of enterprises, making rational use of resources and enhancing the market competitiveness [18–21].

From a different perspective, combining the data mining technology and mathematical logic, we establish an improved multiobjective operation model based on Grey Wolf

Optimization Algorithm to give consideration to energy-saving problems in engineering. The results show that the algorithm can solve the Pareto front end problem in flow-shop scheduling successfully, and it is of great practical value in engineering and management.

2. Multiobjective Grey Wolf Optimizer

2.1. Multiobjective Problem. As briefly mentioned in the introduction, multiobjective optimization refers to the optimization of a problem with more than one objective function [22, 23]. Without loss of generality, it can be formulated as a maximization problem as follows:

$$\begin{aligned} \text{maximize:} \quad & F(x) = f_1(x), f_2(x), \dots, f_o(x), \\ \text{subject to:} \quad & g_i(x) \geq 0, i = 1, 2, \dots, m, \\ & h_i(x) = 0, \quad i = 1, 2, \dots, p, \\ & L_i \leq X_i \leq U_i, \quad i = 1, 2, \dots, n, \end{aligned} \quad (1)$$

where n is the number of variables, o is the number of objective functions, m is the number of inequality constraints, p is the number of equality constraints, g_i is the i th inequality constraints, h_i indicates the i th equality constraints, and $[L_i, U_i]$ are the boundaries of i th variable [24–26].

In single-objective optimization, solutions can be compared easily due to the unary objective function. For maximization problems, solution X is better than Y if and only if $X > Y$. However, the solutions in a multiobjective space cannot be compared by the relational operators due to multicriterion comparison metrics. In this case, a solution is better than (dominates) another solution if and only if it shows better or equal objective value on all of the objectives and provides a better value in at least one of the objective functions [27]. The concepts of comparison of two solutions in multiobjective problems were first proposed by Khamis et al. [28] and then extended by Khamis et al. [29]. Without loss of generality, the mathematical definition of Pareto dominance for a maximization problem is as follows [30].

Definition 1 (Pareto dominance).

Suppose that there are two vectors such as $x = (x_1, x_2, \dots, x_k)$ and $y = (y_1, y_2, \dots, y_k)$.

Vector x dominates vector y (denoted as $x > y$) if

$$[f(x_i) \geq f(y_i)] \wedge [\exists i \in \{1, 2, \dots, k\}: f(x_i) > f(y_i)], \quad \forall i \in \{1, 2, \dots, k\}. \quad (2)$$

The definition of Pareto optimality is as follows.

Definition 2 (Pareto optimality).

A solution $x \in X$ is called Pareto-optimal if

$$\exists y \in X | F(y) > F(x). \quad (3)$$

A set including all the nondominated solutions of a problem is called Pareto-optimal set, and it is defined as follows.

Definition 3 (Pareto optimality set).

The set of all Pareto-optimal solutions is called Pareto set as follows:

$$P_s := \{x, y \in X \mid \exists F(y) \succ F(x)\}. \quad (4)$$

A set containing the corresponding objective values of Pareto-optimal solution in Pareto-optimal set is called Pareto-optimal front [31]. The definition of the Pareto-optimal front is as follows.

Definition 4 (Pareto-optimal front).

A set containing the values of objective functions for Pareto solutions set is

$$P_f := \{F(x) \mid x \in P_s\}. \quad (5)$$

2.2. MOGWO. MOGWO algorithm was proposed by Holland [32]. The social leadership and hunting technique of grey wolves were the main inspiration of this algorithm. In order to mathematically model the social hierarchy of wolves when designing MOGWO, the fittest solution is considered as the alpha (α) wolf. Consequently, the second and third best solutions are named beta (β) and delta (δ) wolves, respectively. The rest of the candidate solutions are assumed to be omega (ω) wolves. In the GWO algorithm, the hunting (optimization) is guided by α , β , and δ . The ω wolves follow these three wolves in the search for the global optimum [33–35]:

$$D = |C * X_p(t) - X(t)|, \quad (6)$$

$$X(t+1) = X_p(t) - A * D, \quad (7)$$

where t indicates the current iteration, A and C are coefficient vectors, X_p is the position vector of the prey, and X indicates the position vector of a grey wolf [36].

The vectors A and C are calculated as follows:

$$A = 2a * r_1 - a, \quad (8)$$

$$C = 2r_2, \quad (9)$$

where elements of a linear decrease from 2 to 0 over the course of iterations and r_1, r_2 are random vectors in $[0, 1]$. Position updating mechanism of search agents and effects A is indicated in Figure 1 [37]. In this figure, we can see that the three top wolves (namely, the fittest solutions) guide the directions of other wolves (namely, the candidate solutions).

The following formulas are run constantly for each search agent during optimization in order to simulate the hunting and find promising regions of search space:

$$D_\alpha = |C_1 * X_\alpha - X|, \quad (10)$$

$$D_\beta = |C_2 * X_\beta - X|, \quad (11)$$

$$D_\delta = |C_3 * X_\delta - X|, \quad (12)$$

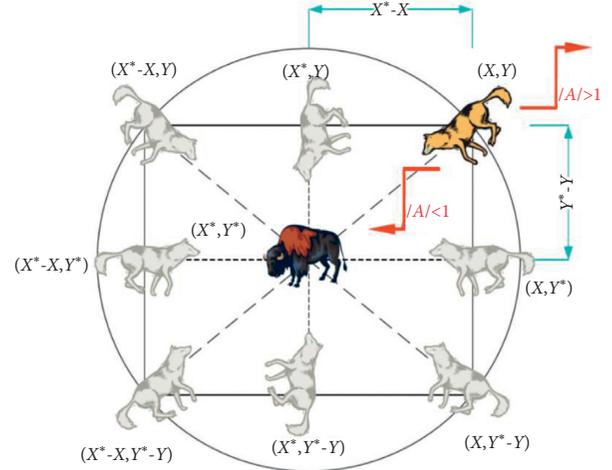


FIGURE 1: Position updating mechanism of search agents and effects.

$$X_1 = X_\alpha - A_1 * D_\alpha, \quad (13)$$

$$X_2 = X_\beta - A_2 * D_\beta, \quad (14)$$

$$X_3 = X_\delta - A_3 * D_\delta, \quad (15)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3}. \quad (16)$$

C is a random value that is generated in $[0, 2]$. The storage mechanism of the nondominant solution is grid in MOGWO. When the archive is full, each hypercube is taken out by roulette according to the probability:

$$P_i = \frac{c}{N_i}, \quad (17)$$

where c is a constant number greater than one and N is the number of obtained Pareto-optimal solutions in the segment [38].

2.3. Defect in MOGWO. Traditional Multiobjective Grey Wolf Optimizer is a grey wolf group predation was inspired by multiobjective optimization algorithm, using a fixed external file to store nondominated solution, simple multiobjective grey wolves optimizer in solving static multiobjective problem, because without a good promotion strategy, lead to being not close to the Pareto-optimal front end, and the diversity of solution set is not high [39]. To solve the above problems, an improved algorithm KMGWO is proposed in the next section.

3. An Improved MOGWO Based on Kalman Filter and Reinforcement Learning

3.1. Kalman Filter. In 1960, R. E. Kalman published a paper describing a method which can process a time series of measurements and predict unknown variables more precisely than that based on a single measurement alone. This is

referred to as the Kalman filter. Kalman filter maintains state vectors, which describe the system state, along with its uncertainties. The equations for the Kalman filter fall into two groups, time update and measurement update equations, which are performed recursively for the Kalman filter to make prediction. Here, the Kalman filter is used to directly predict for future generations in the decision space, and the two major steps are described below [40].

3.1.1. Measurement Update. The measurement update equations are responsible for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate [41]. The individual solutions just before the change occurs are taken as the actual measurements of the previous predictions. This information is used to update the Kalman filter prediction model [42].

3.1.2. Time Update. The time update equations are responsible for projecting forward the current state and error estimate covariance estimates to obtain the a priori estimates for the next step. New solutions are predicted based on the corrected Kalman filter associated with each individual in the decision space [43]. These are a priority estimates of the future.

Pareto-optimal solutions will then be used to update the reference points and subproblems. The specific equations for the two steps are presented in the following [44]:

Time update step:

$$\bar{x}_k^- = A x_{k-1} + b u_{k-1}, \quad (18)$$

$$P_k^- = A P_{k-1} A^T + Q. \quad (19)$$

Measurement update step:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}, \quad (20)$$

$$\hat{x}_k = \bar{x}_k^- + K_k (z_k - H \bar{x}_k^-), \quad (21)$$

$$P_k = (I - K_k H) P_k^-, \quad (22)$$

where x is the state vector to be estimated by the Kalman filter, A denotes the state transition model, u is the optional control input to the state x , B is the control input matrix, and P is the error covariance estimate [45]. Z denotes the measurement of the state vector, H is the observation matrix, and the process and measurement noise covariance matrices are Q and R , respectively. K is the Kalman filter gain.

Here is an example, so we can understand the Kalman filter more intuitively.

As shown in Figure 2, the state vector of an object at the period t obeys the magenta normal distribution, according to which the position of the object at period $t+1$ can be predicted, and the predicted value is the blue normal

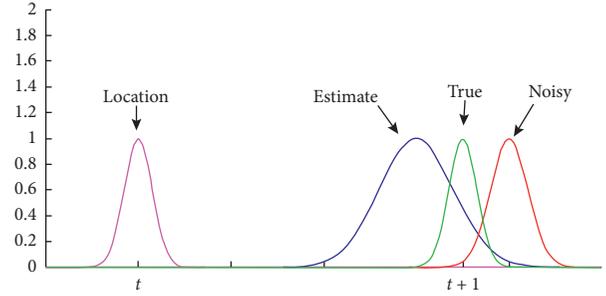


FIGURE 2: Kalman filter principle explanation.

distribution. This blue normal distribution is getting fatter, because a layer of noise to the recursion is added, so the uncertainty is getting bigger. In order to avoid the deviation caused by pure estimation, we made a measurement on the position of the object at the period of $t+1$, and the measurement results are subject to the red normal distribution. Through the five equations of Kalman filter, the real state vector in the car at the time of $t+1$ can be obtained. The state vector follows the green normal distribution, which means that the prediction of Kalman filter can be carried out iteratively.

Kalman filter has been widely applied to the dynamic multiobjective algorithm to ensure that the dynamic multiobjective algorithm can converge to the Pareto-optimal front end in time when the problem changes. It can be said that, at present, Kalman filter is one of the most effective methods to make the population converge to the Pareto-optimal set and the solution set converge to the Pareto-optimal front end. Therefore, this paper reversely applies it to the static multiobjective algorithm to promote the static multiobjective algorithm to converge to the Pareto-optimal front end faster.

In order to overcome the defects of MOGWO described above, this paper proposes a multiobjective Grey Wolf Algorithm based on Kalman filter transformation, which is hereinafter referred to as MKGWO.

After each iteration, a new grey wolf population is generated by the newly generated grey wolf population and the previous-generation grey wolf population using Kalman filter through update probability Pu , which promotes the convergence of solution set to the Pareto-optimal front end. This strategy is called Kalman filter operator, and the formula is described as follows:

$$\left\{ x_{t+1}^i \mid i \in pu * n \right\}, \quad (23)$$

$$x_{t+1} = \text{Kalman filter}(x_{t+1}^-, x_t). \quad (24)$$

3.2. Reinforcement Learning. In the field of big data and machine learning, data mining and learning technology can be divided into supervised learning, unsupervised learning, and reinforcement learning. Reinforcement learning grew out from animal learning and parameter perturbation adaptive control theory, referring to the mapping from

environmental state to the action. It is a machine learning method that can adapt to and interact with the environment. This method is different from supervised learning through positive cases and counterexamples to advise the agent of what action to take, but by trial and error to find the optimal behavioral strategy.

As is shown in Figure 3, the basic principle of reinforcement learning is test evaluation process, and the agent chooses an action for the environment, after which the action state change is accepted at the same time producing a reinforcement signal. Then, the agent selects next action according to the reinforcement signal and environmental current state, and the selection principle is increased by the probability of positive reinforcement. The selected action not only affects the immediate reinforcement value, but also the state of the environment at the next moment and the final reinforcement value.

In MOGWO improved based on Kalman filter operator, we found that Pu (update probability) could not be fixed. For different problems, the optimal vaccination probability was different. For the same problem, the optimal vaccination probability is also different at different periods of iteration, so we choose the reinforcement learning method here to dynamically determine the vaccination probability. In this paper, this improved method is collectively called reinforcement learning operator, as shown in Figure 4.

The reinforcement learning method used here is based on snap-drift neural network. It switches between snap mode and drift mode. In this operator, agent (MOGWO) accepts the state (snap or drift) and reward value (Pm) at time t then takes an action (increase or decrease Pu) to convert to a new state:

$$Pm = \frac{Se}{N}, \quad (25)$$

$$Pu = \begin{cases} \max(0, Pu - \omega), & \mu = \text{snap}, \\ \min(1, Pu + \omega), & \mu = \text{drif}. \end{cases} \quad (26)$$

Se is the number of nondominated solutions generated in this iteration; Pm is the conversion probability, that is determined by the proportion of update individuals in population this iteration. Snap mode is used for less than 50%, and drift mode is used for more than or equal to 50%. ω is the step size of Pu with each change.

3.3. MKGWO Flow. MKGWO's algorithm flow framework is as follows (Algorithm 1).

4. Simulation Experiments

To test the performance of MKGWO, MKGWO and MOGWO, MOPSO, NSGA2, MOEA/D, and PESA2 simulation experiment, the benchmark functions and correlation index are analyzed in this section.

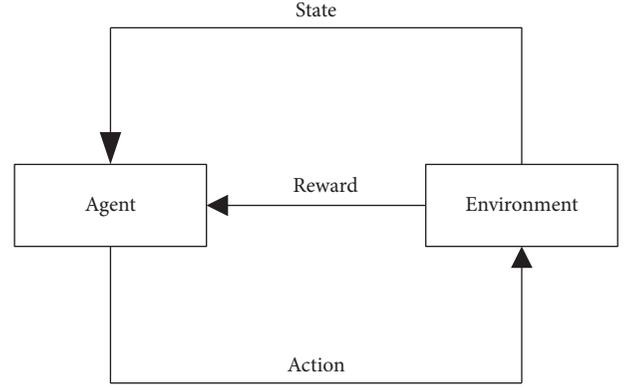


FIGURE 3: Principle of reinforcement learning.

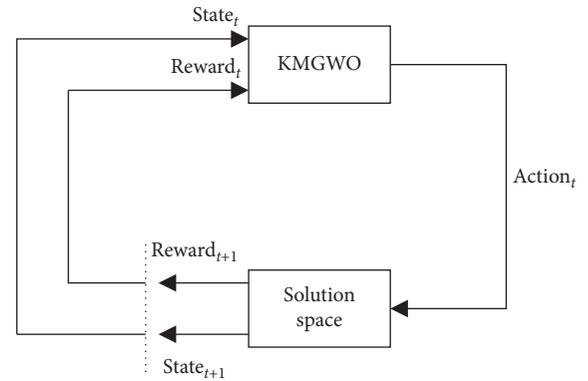


FIGURE 4: Reinforcement learning operator operation process.

4.1. Experimental Environment and Benchmark Function

4.1.1. Experimental Environment. The operating environment of the simulation experiment is as follows: the machine is Dawning 5000A supercomputer. Xeon X5620 CPU (4 cores) *2, 24 GB memory, 300 GB SAS hard disk. Equipped with RHEL5.6 operating system. The programming tool is MATLAB 2012a (for Linux).

4.1.2. Benchmark Function. In this paper, six benchmark functions are selected to evaluate the performance of the algorithm. This group of benchmark functions is widely used in the test of multiobjective optimization algorithm. The function names, dimensions, ranges, and expressions are shown in Table 1. These six benchmark functions can be divided into two categories: Kursawe, Schaffer, ZDT1, and ZDT6 are two-dimensional test functions used to investigate the search ability of the algorithm at low latitude; Viennet2 and Viennet3 are 3D test functions, adding more Pareto points and increasing the difficulty of searching, for further detecting the overall performance of the algorithm. These test problems are considered as the most challenging test problems in the literature that provide different multiobjective search spaces with different Pareto-optimal fronts: convex, nonconvex, discontinuous, and multimodal.

```

Initialize the grey wolf population  $X_i (i = 1, 2, \dots, n)$ 
Initialize  $a$ ,  $A$ , and  $C$ 
Calculate the objective values for each search agent
Find the nondominated solutions and initialize the archive with them
 $X_\alpha = \text{Select Leader (archive)}$ 
Exclude alpha from the archive temporarily to avoid selecting the same leader
 $X_\beta = \text{Select Leader (archive)}$ 
Exclude beta from the archive temporarily to avoid selecting the same leader
 $X_\delta = \text{Select Leader (archive)}$ 
Add back alpha and beta to the archive
 $t = 1$ 
while ( $t < \text{Max number of iterations}$ ).
for each search agent
    Update the position of the current search agent by equations (6)–(16)
end for
Update  $a$ ,  $A$ , and  $C$ 
Invoke Kalman filter by equations (18)–(24)
Invoke reinforcement learning operator by equations (25) and (26)
Calculate the objective values of all search agents
Find the nondominated solutions
Update the archive with respect to the obtained nondomination solutions
If the archive is full
    Run the grid mechanism to omit one of the current archive solutions
    Add the new solution to the archives
End if
 $X_\alpha = \text{Select Leader (archive)}$ 
Exclude alpha from the archive temporarily to avoid selecting the same leader
 $X_\beta = \text{Select Leader (archive)}$ 
Exclude beta from the archive temporarily to avoid selecting the same leader
Add back alpha and beta to the archive

```

ALGORITHM 1 : MKGWO flow framework.

TABLE 1: Benchmark function equation.

Function name	Equation	Search domain	Search boundary
Kursawe	Minimize = $\begin{cases} f_1(x) = \sum_{i=1}^2 [-10 \exp(-0.2 \sqrt{x_i^2 + x_{i+1}^2})], \\ f_2(x) = \sum_{i=1}^3 [x_i ^{0.8} + 5 \sin(x_i^3)] \end{cases}$	3	$-5 \leq x_i \leq 5$
Schaffer	Minimize = $\begin{cases} f_1(x) = x^2, \\ f_2(x) = (x-2)^2 \end{cases}$	1	$-5 \leq x_i \leq 5$
Viennet2	Minimize = $\begin{cases} f_1(x) = 1/2(x_1 - 2)^2 + 1/13(x_2 + 1)^2 + 3, \\ f_2(x) = 1/36(x_1 + x_2)^2 + 1/8(x_2 - x_1 + 2)^2 - 17, \\ f_3(x) = 1/175(x_1 + 2x_2 - 1)^2 + 1/17(2x_2 - x_1)^2 - 13 \end{cases}$	2	$-4 \leq x_i \leq 4$
Viennet3	Minimize = $\begin{cases} f_1(x) = 1/2(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2), \\ f_2(x) = 1/8(3x_1 - 2x_2 + 4)^2 + 1/27(x_1 - x_2 + 1)^2 + 15, \\ f_3(x) = 1/x_1^2 + x_2^2 + 1 - 1.1 \exp(-(x_1^2 + x_2^2)) \end{cases}$	2	$-3 \leq x_1 \leq 10, -10 \leq x_2 \leq 3$
ZDT1	Minimize = $\begin{cases} f_1(x) = x_1, \\ f_2(x) = g(x)h(f_1(x), g(x)), \\ g(x) = 1 + 9/29 \sum_{i=2}^{30} x_i, \\ h(f_1(x), g(x)) = 1 - \sqrt{f_1(x)/g(x)} \end{cases}$	30	$0 \leq x_i \leq 1$
ZDT6	Minimize = $\begin{cases} f_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1), \\ f_2(x) = g(x)h(f_1(x), g(x)), \\ g(x) = 1 + 9 \left[\sum_{i=2}^{10} x_i / 9 \right]^{0.25}, \\ h(f_1(x), g(x)) = 1 - (f_1(x)/g(x))^2 \end{cases}$	10	$0 \leq x_i \leq 1$

4.2. *Contrast Indicators and Algorithm Parameters.* For the performance metric, we have used Inverted Generational Distance (IGD) for measuring convergence. The Spacing (SP) is employed to quantify and measure the coverage. The mathematical formulation of IGD is similar to that of Generational Distance (GD). This modified measure formula is as follows:

$$\text{IGD} = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n}, \quad (27)$$

where n is the number of true Pareto-optimal solutions and d indicates the Euclidean distance between the i th true Pareto-optimal solution and the closest obtained Pareto-optimal solutions in the reference set. The Euclidean distance between obtained solutions and reference set is different here. In IGD, the Euclidean distance is calculated for every true solution with respect to its nearest obtained Pareto-optimal solutions in the objective space.

The mathematical formulation of the SP and MS measures is as follows:

$$\text{SP} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2}. \quad (28)$$

In the simulation experiment, the population number of each algorithm is 200, the number of archives is 200, and the number of iterations is 500. Each algorithm ran independently for 30 generations, and its minimum value, maximum value, average value, and variance were taken as the results. The remaining parameters are shown in Table 2.

4.3. *Comparative Analysis of Experimental Results.* The Pareto diagram in Figures 5–10 shows that the Pareto nondominant solution generated by KMGWO is basically consistent with the real Pareto front end and the solution set distribution is relatively uniform. Then, the simulation experiment results are analyzed by data comparison. In the Kursawe function test results, the indicators of KMGWO and MOEA/D are the best, around 0.12, slightly larger than PESA2. The worst is NSGA2 algorithm. It shows that in addition to KMGWO, MOEA/D is also suitable for solving low-dimensional multiobjective problems. The test results of Schaffer's function show that KMGWO has the best effect. MOGWO and MOEA/D are slightly worse than KMGWO, and the standard deviation of MOGWO is smaller. In both Viennet2 and Viennet3, the results of the three-target benchmark test functions, PESA2 and KMGWO, have the best effect of two algorithms, respectively. In Viennet2, KMGWO is only slightly less than PESA2 algorithm. Taken together, KMGWO is very suitable for solving three-target optimization problem; this may be associated with the algorithm search ability being stronger. The next step research direction can test the effect of this algorithm in the target problem; it is very exciting, and in the two test functions, KMGWO's standard deviation is also very good. It shows that the stability of the algorithm in this kind of test function is excellent. In this paper, two of the functions of ZDT system

TABLE 2: Algorithms' parameters.

Algorithms	Parameters
KMGWO	Alpha = 0.1; beta = 4; gamma = 2; $Pu = 0.2$; $w = 0.1$
MOGWO	Alpha = 0.1; beta = 4; gamma = 2
MOPSO	$w = 0.4$; $c1 = 2$; $c2 = 2$
NSGA2	$pc = 0.9$; $pm = 0.5$
MOEA/D	Gamma = 0.5
PESA2	$p = 0.5$; $h = 0.3$; gamma = 0.15

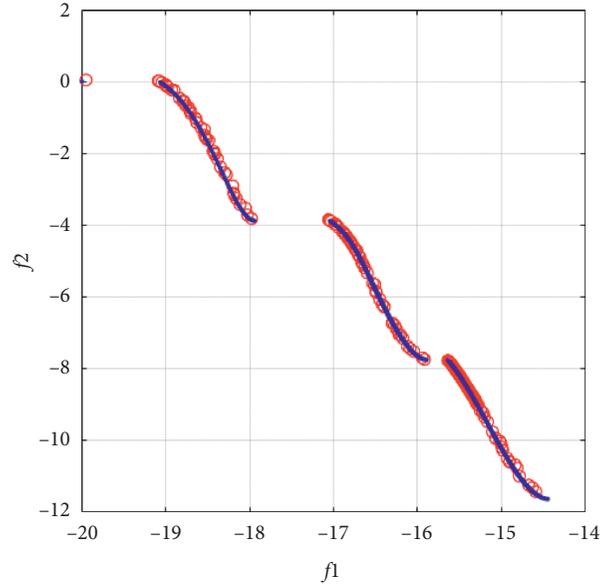


FIGURE 5: Kursawe's test result.

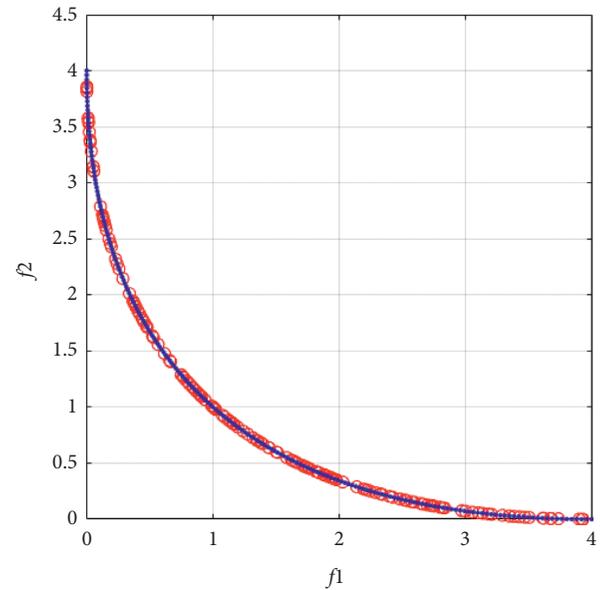


FIGURE 6: Schaffer's test result.

proposed by Deb are selected: ZDT1 and ZDT6. The test results of ZDT1 function showed that the best algorithm was MOGWO, MOPSO was slightly inferior, and KMGWO was excellent and ranked third. The test results of ZDT6 function

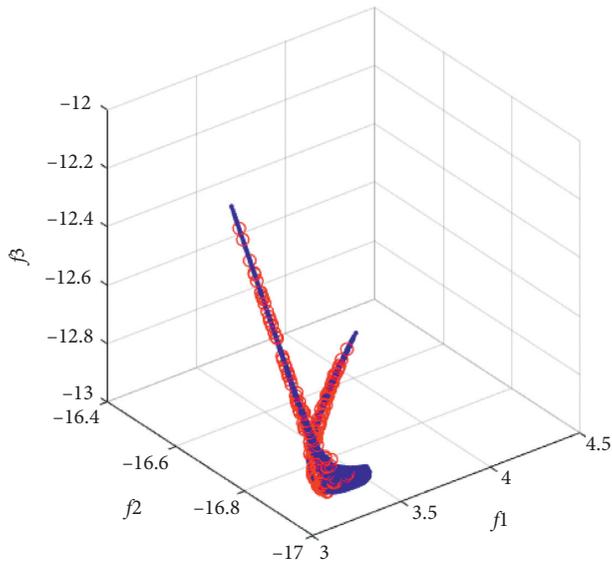


FIGURE 7: Viennet2 test result.

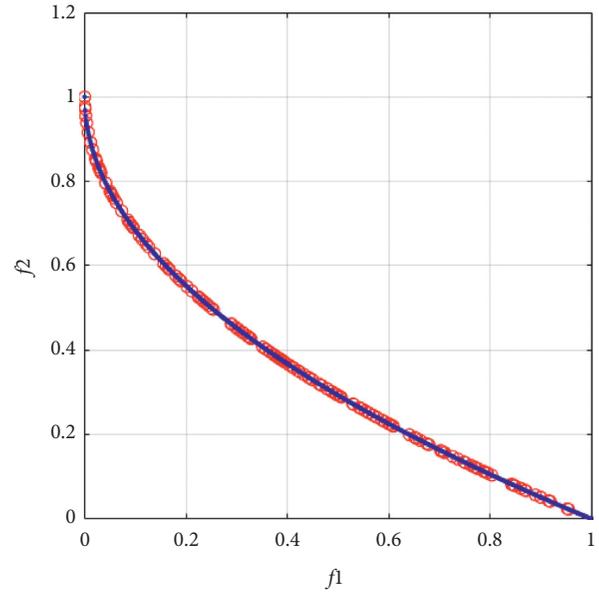


FIGURE 9: ZDT1 test result.

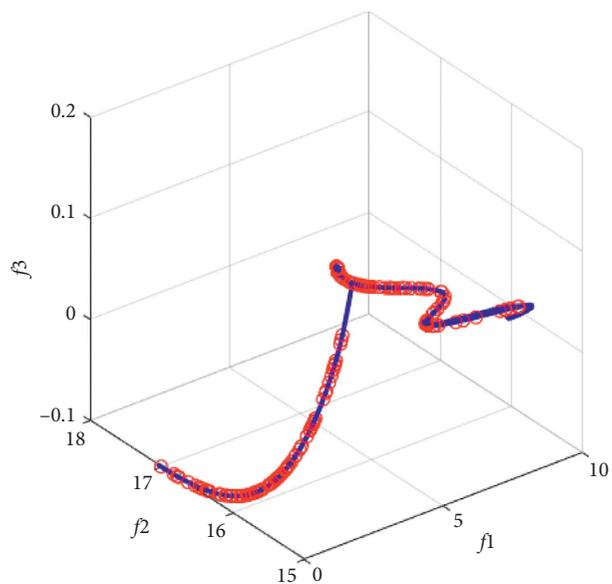


FIGURE 8: Viennet3 test result.

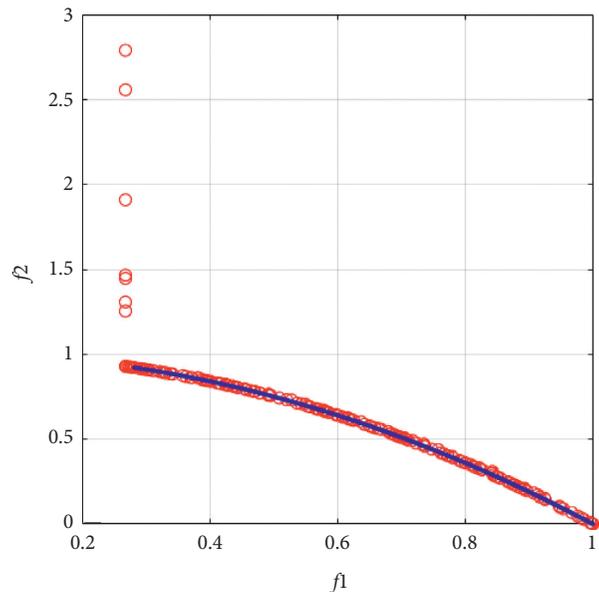


FIGURE 10: ZDT6 test result.

show that KMGWO is the best algorithm with excellent stability. In general, KMGWO's ability to approach the real Pareto front end is very strong, especially when the number of targets is high, so the algorithm will be widely used in production.

Concerning IGD metric, the merit is clear that the KMGWO significantly dominates over the KMGWO on almost the problems. As shown in Table 3, the KMGWO is the best performing method in our comparison experiment. This strongly demonstrates that the reinforcement learning operator can effectively improve the overall performance of the algorithm. The reason for this superiority of MKGWO lies in reinforcement learning operator as follows: in reinforcement learning operator, the optimal vaccination probability is also different at different periods of iteration;

population is divided into many subpopulations and each solution has its own neighbors.

Table 4 is the statistics of SP. SP value represents the degree of uniformity between Pareto solutions. The smaller this value is, the more homogeneous the Pareto solutions obtained by the algorithm will be, and the smaller the distance difference between them will be. Kursawe's function test results show that NSGA2 SP value is the smallest of these six algorithms, 1.4, and MOGWO, MOEA/D, and PESA2 with poor results, suggesting that NSGA2 simple double objective test functions can generate the most homogeneous solution set. It is worth noting that KMGWO effect is better than MOGWO; Kalman filtering and reinforcement

TABLE 3: Comparison of algorithm running results' IGD.

SP		Kursawe	Schaffer	Viennet2	Viennet3	ZDT1	ZDT6
KMGWO	Min	0.001268	0.000548	0.000195	0.00013	4.15E-05	0.000336
	Max	0.001467	0.000606	0.000221	0.000663	5.97E-05	0.000473
	Mean	0.001367	0.000577	0.000208	0.000397	5.06E-05	0.000404
	Std	0.00014	4.1E-05	1.84E-05	0.000377	1.28E-05	9.7E-05
MOGWO	Min	0.001429	0.000626	0.000207	0.000222	2.96E-05	0.002448
	Max	0.00208	0.000689	0.000218	0.000265	3.66E-05	0.01636
	Mean	0.001755	0.000657	0.000213	0.000244	3.31E-05	0.009404
	Std	0.000461	4.47E-05	7.86E-06	2.99E-05	4.93E-06	0.009838
MOPSO	Min	0.002178	0.000669	0.00036	0.0002	0.000267	0.026323
	Max	0.002426	0.000705	0.00039	0.000221	0.000296	0.030315
	Mean	0.002302	0.000687	0.000375	0.00021	0.000281	0.028319
	Std	0.000175	2.61E-05	2.08E-05	1.44E-05	2.11E-05	0.002823
NSGA2	Min	0.402172	0.09739	1.538041	0.85775	0.092674	0.134332
	Max	0.410001	0.097398	1.592757	0.870869	0.109724	0.185229
	Mean	0.406086	0.097394	1.565399	0.864309	0.101199	0.15978
	Std	0.005536	5.83E-06	0.03869	0.009277	0.012056	0.035989
MOEA/D	Min	0.001268	0.000605	2.33E-05	0.000105	3.44E-05	0.000864
	Max	0.004135	0.000613	4.94E-05	0.000208	4.66E-05	0.018473
	Mean	0.002701	0.000609	3.64E-05	0.000156	4.05E-05	0.009668
	Std	0.002027	5.98E-06	1.85E-05	7.28E-05	8.6E-06	0.012451
PESA2	Min	0.001274	0.000628	0.000183	0.000195	0.002272	0.021336
	Max	0.001549	0.000644	0.000347	0.000323	0.002847	0.052005
	Mean	0.001412	0.000636	0.000265	0.000259	0.00256	0.036671
	Std	0.000194	1.15E-05	0.000116	9.05E-05	0.000407	0.021687

TABLE 4: Comparison of algorithm running results' SP.

SP		Kursawe	Schaffer	Viennet2	Viennet3	ZDT1	ZDT6
KMGWO	Min	1.869705	0.464979	0.229326	1.681636	0.057698	0.170041
	Max	2.140332	0.628268	0.354561	2.394785	0.067846	0.255117
	Mean	2.005019	0.546624	0.291944	2.038211	0.062772	0.212579
	Std	0.191362	0.115463	0.088554	0.504272	0.007176	0.060158
MOGWO	Min	2.071943	0.544736	0.338835	1.691569	0.058589	0.082575
	Max	2.255649	0.588034	0.395661	2.20073	0.061941	0.180822
	Mean	2.163796	0.566385	0.367248	1.946149	0.060265	0.131699
	Std	0.129899	0.030616	0.040182	0.360031	0.00237	0.069471
MOPSO	Min	1.812654	0.597622	0.406334	2.205442	0.074537	0.309447
	Max	1.948818	0.601261	0.41442	2.221322	0.076896	0.435343
	Mean	1.880736	0.599441	0.410377	2.213382	0.075716	0.372395
	Std	0.096282	0.002573	0.005718	0.011228	0.001668	0.089021
NSGA2	Min	1.442097	1.198419	0.225471	2.55751	0.342942	0.245284
	Max	1.490551	1.228526	0.2299	2.645984	0.343657	0.372529
	Mean	1.466324	1.213472	0.227685	2.601747	0.343299	0.308907
	Std	0.034262	0.021289	0.003132	0.06256	0.000505	0.089976
MOEA/D	Min	1.645038	0.234028	0.238373	0.097528	0.054964	0.072255
	Max	1.655435	0.295418	0.312842	0.145355	0.063264	0.15472
	Mean	1.650237	0.264723	0.275607	0.121441	0.059114	0.113488
	Std	0.007352	0.04341	0.052658	0.033819	0.005869	0.058311
PESA2	Min	2.091793	0.599484	0.276225	2.362664	0.069884	0.22178
	Max	2.160544	0.638922	0.34284	2.398225	0.076021	0.814949
	Mean	2.126169	0.619203	0.309533	2.380444	0.072953	0.518364
	Std	0.048614	0.027887	0.047104	0.025145	0.00434	0.419434

learning operator under the condition of synergy, multi-objective algorithm, can improve the static solution set of uniformity. The test results of Schaffer's function show that

the values of KMGWO and MOEA/D are optimal, both around 0.2, but the standard deviation of KMGWO is better. In this test function, the performance of KMGWO is the

best. The test results of the Viennet2 function show that the SP values of KMGWO, MOEA/D, PESA2, and NSGA2 are roughly the same, all around 0.2. From the image, we can know that the test function is a three-objective test function, and its image is a three-dimensional image, but not complicated, indicating that most algorithms can find a relatively uniform solution set on it. The test results of Viennet3 show that the SP value of the MOEA/D algorithm is optimal, reaching 0.9. The results of KMGWO and MOGWO come next, reaching 1.6. The other test algorithms are far inferior to the three algorithms. In the test results of ZDT1, the opposite was found. Except for NSGA2, all the algorithms reached the optimal value of around 0.05, and the uniformity of NSGA2 was the worst, which was only 0.3. This situation may be caused by the cross search ability of NSGA2. In the test results of ZDT6 function, the SP value of MOEA/D, PESA2, and MOGWO was the smallest, around 0.8, followed by KMGWO, with SP value of 0.17, and then MOPSO, with 0.3. To sum up, KMGWO is quite a competitive algorithm in terms of the uniformity of the generated Pareto solution set, but other algorithms, such as MOEA/D, cannot be ignored, and their performance in this index is relatively excellent.

5. Application to Energy Saving considering Flow-Shop Scheduling

5.1. Energy Consumption considering Flow-Shop Scheduling. The low-carbon scheduling problem in the flow shop studied in this section can be described as follows: n jobs need to go through m stages in the same flow direction, and each job has only one working procedure in each stage. Considering the preparation time of the machine, the preparation time is related to the ordering of the two adjacent jobs. Therefore, the preparation time is based on the sequence correlation of the artifacts, and the machine startup is linked to the processing time of the artifacts.

At different stages, the machine has different speed gears for production and can be adjusted. From the point of view of energy consumption, the machine has four different states: (machine) on the machining processing status, start state in preparation for a new jobs (machine), standby (machine is in idle), and turned off (machine is turned off). Under normal circumstances, when the machine is working at a higher rate, the processing time will be shortened, but the corresponding energy consumption will increase. Therefore, this problem aims to maximize the completion time and energy consumption index. Due to the characteristics of the problem studied in this chapter, this problem is much more complicated than the traditional flow-shop

scheduling problem. In this problem, other settings are as follows.

The job is processed continuously in the workshop. In other words, the process cannot be interrupted. Machines are allowed free time and have unlimited buffers between phases. When there is a first job processing, the machine boots. When all the jobs are finished, the machine is shut down. The machine speed cannot be adjusted in the course of a job processing.

In order to present the mathematical model of the problem, we first defined the following related mathematical symbols according to the above description of the problem.

Symbol definition are below:

I : Set of jobs, $I = \{1, \dots, i, \dots, n\}$.

J : Set of processes, $J = \{1, \dots, j, \dots, m\}$.

V_j : The speed set of the process j machine, $V_j = \{1, \dots, v, \dots, |V_j|\}$.

$p_{i,j,v}$: The processing time in process j speed v job i .

$s_{i',i,j}$: The start time in process j job i' transform to i .

$i' = i, s_{i,i,j}$ indicates the job i is the first job allotted this machine.

$pp_{j,v}$: Process j the power consumed by the machine running at speed v .

sp_j : Process j power consumed by machine start-up.

ip_j : The power consumed by the machine in idle running in process j .

Decision variables are as below:

$b_{i,j}$: The initial processing time of job I at stage j .

$e_{i,j}$: The end processing time of job I at stage j .

$x_{i,j,v}$: The two-dimensional decision variable, when the job from I to j is processed at a rate of k , is 1; otherwise the value is 0.

$z_{i',i,j}$: Two-dimensional decision variable, when the job I' is processed onstage j , the value is 1; otherwise the value is 0.

T_j^{on} : Auxiliary decision variable, indicating the start-up time of stage j machine, decision by $b_{i,j}$ and

$x_{i,j,k,v}$.

T_j^{off} : Auxiliary decision variable, indicating the halt time of stage j machine, decision by $b_{i,j}$ and $x_{i,j,k,v}$.

Based on these mathematical symbols, the mixed-integer programming model of the flow-shop low-carbon scheduling problem is presented as follows:

Objective function:

$$\text{Min} \cdot C_{\max} = \max_{i \in I} e_{i,m}, \quad (29)$$

$$\text{Min TEC} = E_1 + E_2 + E_3, \quad (30)$$

$$E_1 = \sum_{j \in J} \sum_{i \in I} \sum_{v \in V_j} x_{i,j,v} \cdot p_{i,j,v} \cdot pp_{j,v}, \quad (31)$$

$$E_2 = \sum_{j \in J} \sum_{i \in I} \sum_{i' \in I} x_{i,j,v} \cdot z_{i',i,j} \cdot s_{i',i} \cdot sp_j, \quad (32)$$

$$E_3 = \sum_{j \in J} \left(T_j^{\text{off}} - T_j^{\text{on}} - \sum_{i \in I} \sum_{v \in V_j} x_{i,j,k,v} \cdot \left(p_{i,j,v} + \sum_{i' \in I} z_{i',i,j} \cdot s_{i',i} \right) \right) \cdot ip_j. \quad (33)$$

Constraint condition:

$$\sum_{v \in V_j} x_{i,j,v} = 1, \quad \forall i \in I, j \in J, \quad (34)$$

$$e_{i,j} - b_{i,j} = \sum_{v \in V_j} p_{i,j,v} \cdot x_{i,j,v}, \quad \forall i \in I, j \in J, \quad (35)$$

$$b_{i,j} - e_{i,j-1} \geq 0, \quad \forall i \in I, j \in \{2, \dots, m\}, \quad (36)$$

$$z_{i,i',j} + z_{i',i,j} \leq 1, \quad \forall i, i' \in I, j \in J, \quad (37)$$

$$b_{i,j} - \sum_{i' \in I} e_{i',j} \cdot z_{i',i,j} - \sum_{i' \in I} s_{i',i,j} \cdot z_{i',i,j} \geq 0, \quad \forall i \in I, j \in J, \quad (38)$$

$$e_{i,j} - \sum_{v \in V_j} p_{i,j,v} \cdot x_{i,j,v} - \sum_{i' \in I} s_{i',i,j} \cdot z_{i',i,j} = 0, \quad \forall i \in I, j \in J, \quad (39)$$

$$T_j^{\text{on}} = \min \left\{ b_{i,j} \cdot x_{i,j,v} - \sum_{i' \in I} s_{i',i,j} \cdot z_{i',i,j} \right\}, \quad \forall i \in I, j \in J, v \in V_j, \quad (40)$$

$$T_j^{\text{off}} = \max \{ e_{i,j} \cdot x_{i,j,v} \}, \quad \forall i \in I, j \in J, v \in V_j, \quad (41)$$

$$x_{i,j,v} \in \{0, 1\}, \quad \forall i \in I, j \in J, v \in V_j, \quad (42)$$

$$z_{i,i',j} \in \{0, 1\}, \quad \forall i, i' \in I, j \in J, \quad (43)$$

$$e_{i,j} > 0, b_{i,j} \geq 0, \quad \forall i \in I, j \in J. \quad (44)$$

Formula (29) stands for the minimum completion time. Formula (30) represents the minimum energy consumption. Formula (31) represents the total energy consumption when the machine is in the processing state, while formula (32) represents the total energy consumption when the machine is in the starting state, and formula (33) represents the total energy consumption when the machine is in the standby state. Formula (34) means that each job traverses all stages, and in a specific stage, each job is assigned to a machine and processed at a speed level. Formula (35) means that interruption is not allowed during processing. Formula (36) ensures that the job operation can only be started after the operation of the previous stage is completely processed. Formulas (37) and (38) guarantee the machine capacity limit, which means the job can be processed only after the previous work is completed. Formula (39) means that the

machine starting to process immediately after installation is complete. Formula (40) represents the calculation of auxiliary variables, which is equal to the minimum value between the start time and the set time of the job assigned to the corresponding machine. Formula (41) represents the calculation of auxiliary variables, which is equal to the maximum value of the end time of the job assigned to the corresponding machine. Formulas (42)–(44), respectively, define the feasible range of decision variables.

This section gives a simple example of three jobs and three stages, each with three different processing speeds. Table 5 shows the processing time and corresponding power consumption of the machine at each stage; that is, the element in the table represents $(p_{i,j}, pp_{i,j,v})$.

Table 6 shows the sequence-dependent start times; the element in the table represents $s_{i',i,j}$. Set $sp_j = 3; ip_j = 1$.

TABLE 5: Processing time and corresponding power.

Process	Job $i=1$			Job $i=2$			Job $i=3$		
	$\nu=1$	$\nu=2$	$\nu=3$	$\nu=1$	$\nu=2$	$\nu=3$	$\nu=1$	$\nu=2$	$\nu=3$
$j=1$	(28, 4)	(22, 7)	(15, 11)	(22, 4)	(20, 7)	(18, 11)	(20, 4)	(18, 7)	(17, 11)
$j=2$	(20, 5)	(16, 8)	(12, 12)	(21, 5)	(18, 8)	(15, 12)	(23, 5)	(20, 8)	(17, 12)
$j=3$	(19, 5)	(15, 7)	(11, 10)	(20, 5)	(16, 7)	(12, 10)	(23, 5)	(20, 7)	(16, 10)

TABLE 6: Sequence-dependent start-up time.

Job	Job $i=1$			Job $i=2$			Job $i=3$		
	$j=1$	$j=2$	$j=3$	$j=1$	$j=2$	$j=3$	$j=1$	$j=2$	$j=3$
$i'=1$	5	8	4	4	6	8	6	4	4
$i'=2$	6	4	6	2	3	2	4	6	2
$i'=3$	2	4	8	2	4	2	6	4	4

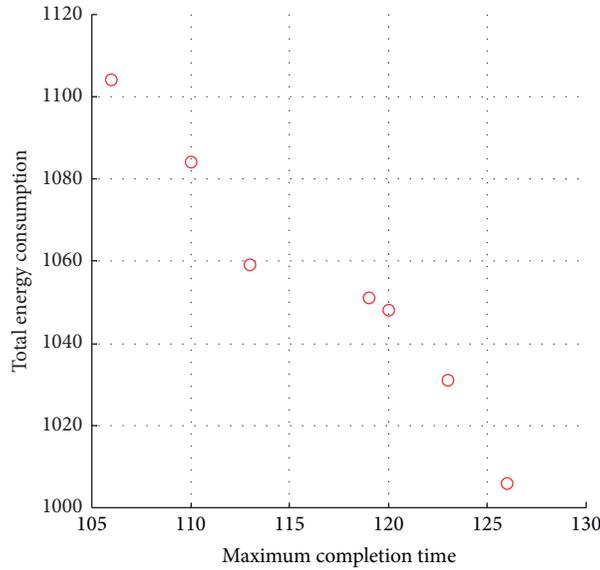


FIGURE 11: Experimental results of Pareto chart.

TABLE 7: Flow-shop scheduling.

	Job 2				Job 3				Job 1			
	Begin	Start	Work	End	Begin	Start	Work	End	Begin	Start	Work	End
Process 1	0	2	15	17	17	2	22	41	41	6	20	67
Process 2	14	3	20	37	37	4	21	62	63	4	23	90
Process 3	35	2	19	56	60	2	20	82	86	4	23	113

5.2. *Experimental Result.* To solve the above pipeline scheduling problem, this paper sets the parameters of KMGWO as follows: population number 20, warehouse number 20, 100 iterations. Figure 11 shows the Pareto front end obtained by KMGWO algorithm. It can be clearly observed that this problem is not complicated for KMGWO, and the Pareto-optimal solution set is easily obtained. Because there was not enough prior

knowledge, we chose the scheme calculated at the Pareto point with the longest time of 113 and energy consumption of 1059.

Table 7 is the Pareto point calculated plan, showing the start time of the job in each process, warm-up time, working time, and end time. We, through these data, drew a Gantt chart, as shown in Figure 12; this is a time and energy balance options assembly line scheduling Gantt chart. In

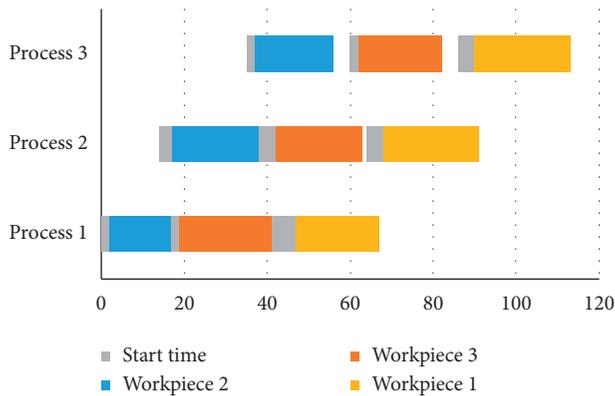


FIGURE 12: Experimental results Gantt chart.

practice, we can choose according to the actual demand of enterprise decision point countless times and can deal with complex situation; this is the power of KMGWO algorithm.

6. Conclusions

In this paper, an improved multiobjective operational model based on Grey Wolf Optimization Algorithm related to Kalman filtering and reinforcement learning (KMGWO) is proposed, which is the combination of data mining technology and mathematical logic. With Kalman filter, the algorithm promotes the understanding set to the real Pareto front end. The reinforcement learning operator is applied to enhance the utilization of the dominant position of the group, and adaptive parameters are used instead of human intervention. The results of six benchmark functions show that the algorithm performs better than the comparison algorithm in terms of approximating the real Pareto optimal solution set and keeping the solution set uniform. Considering the energy saving of the assembly line scheduling solution, KMGWO performance is excellent and accordingly suitable for solving the practical optimization problems. This operational model has the formidable superiority in the field of mathematical optimization, which can be applied to machine learning, engineering optimization design, and other important areas, thus enhancing the performance of energy saving in production management.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

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

Acknowledgments

This work was supported by the National Social Science Foundation of China (NSSFC) under Grant no. 17BGL238.

References

- [1] Y. Yao, Y. Chun-Ming, and Y. Feng, "Solving bi-objective reentrant hybrid flow shop scheduling problems by a hybrid discrete grey wolf optimizer," *Operations Research and Management Science*, vol. 28, no. 8, pp. 190–199, 2019.
- [2] X. Zhang and X. Wang, "Comprehensive review of grey wolf optimization algorithm," *Computer Science*, vol. 46, no. 3, pp. 30–38, 2019.
- [3] H. Meng, X. zhang, and L. San-Yang, "Two-group method for constrained multi-objective optimization problems," *Chinese Journal of Computers*, vol. 31, no. 2, pp. 229–235, 2008.
- [4] Y. G. Woldeesenbet, G. G. Yen, and B. G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 514–525, 2009.
- [5] Y. Zhang, "Concise multi-objective particle swarm optimization algorithm for constrained optimization," *Electronic Journals*, vol. 39, no. 6, pp. 1437–1440, 2011.
- [6] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part II: handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.
- [7] W. F. Gao, G. G. Yen, and S. Y. Liu, "A dual-population differential evolution with coevolution for constrained optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 1094–1107, 2015.
- [8] B. Xiaojun and L. Zhang, "Constrained multi-objective optimization algorithm based on adaptive e-truncation strategy," *Journal of Electronics and Information Technology*, vol. 38, no. 8, pp. 2047–2053, 2016.
- [9] J. Branke and K. Deb, *Integrating User Preferences into Evolutionary Multi-Objective Optimization Knowledge Incorporation in Evolutionary Computation*, pp. 461–477, Springer, Berlin, Germany, 2005.
- [10] J. Branke, T. Kaußler, and H. Schmeck, "Guidance in evolutionary multi-objective optimization," *Advances in Engineering Software*, vol. 32, no. 6, pp. 499–507, 2001.
- [11] C. A. C. Coello, "Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored," *Frontiers of Computer Science in China*, vol. 3, no. 1, pp. 18–30, 2009.
- [12] D. Lei, M. Li, and L. Wang, "A two-phase meta-heuristic for multiobjective flexible job shop scheduling problem with total energy consumption threshold," *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 1097–1109, 2019.
- [13] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuisen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, Berlin, Germany, 2007.
- [14] C. A. C. 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.
- [15] X. Wu, X. Shen, and Q. Cui, "Multi-objective flexible flow shop scheduling problem considering variable processing time due to renewable energy," *Sustainability*, vol. 10, no. 3, p. 841, 2018.
- [16] C. A. Coello Coello and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Paper presented at the Evolutionary Computation*, Honolulu, HI, USA, May 2002.
- [17] I. Das and J. E. Dennis, "Normal-boundary intersection: a new method for generating the pareto surface in nonlinear

- multicriteria optimization problems,” *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.
- [18] K. Deb, *Advances in Evolutionary Multi-Objective Optimization Search Based Software Engineering*, Springer, Berlin, Germany, 2012.
- [19] L. Yin, X. Li, L. Gao, C. Lu, and Z. Zhang, “A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem,” *Sustainable Computing: Informatics and Systems*, vol. 13, no. 3, pp. 15–30, 2017.
- [20] F. Y. Edgeworth, *Mathematical Physics*, Routledge, London, UK, 1881.
- [21] L. Yin, X. Li, C. Lu, and L. Gao, “Energy-efficient scheduling problem using an effective hybrid multi-objective evolutionary algorithm,” *Sustainability*, vol. 8, no. 12, pp. 1268–1301, 2016.
- [22] P. Mahat, Z. Chen, and B. Bak-Jensen, “Review of islanding detection methods for distributed generation,” in *Proceedings of the 2008 Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies*, Nanjing, China, April 2008.
- [23] Z. Zhang, L. Wu, T. Peng, and S. Jia, “An improved scheduling approach for minimizing total energy consumption and makespan in a flexible job shop environment,” *Sustainability*, vol. 11, p. 179, 2019.
- [24] A. R. Malekpour, A. R. Seifi, M. R. Hesamzadeh, and N. Hosseinzadeh, “An optimal load shedding approach for distribution networks with DGs considering capacity deficiency modelling of bulked power supply,” in *Proceedings of the Australasian Universities Power Engineering Conference (AUPEC’08)*, Sydney, Australia, December 2008.
- [25] W. Liao and T. Wang, “Promoting green and sustainability: a multi-objective optimization method for the job-shop scheduling problem,” *Sustainability*, vol. 10, no. 11, p. 4205, 2018.
- [26] M. H. Haque, “A linear static voltage stability margin for radial distribution system,” in *Proceedings of the 2006 IEEE Power Engineering Society General Meeting*, pp. 1–6, Montreal, Canada, June 2006.
- [27] S. Banerjee, D. Maity, and C. Chanda, “Teaching learning based optimization for economic load dispatch problem considering valve point leading effect,” *Electrical Power and Energy Systems*, vol. 73, 2015.
- [28] A. Khamis, H. Shareef, A. Mohamed, and Z. Dong, “A load shedding scheme for DG integrated islanded power system utilizing backtracking search algorithm,” *Ain Shams Engineering Journal*, vol. 9, no. 1, 2015.
- [29] A. Khamis, H. Shareef, A. Mohamed, and E. Bizkevelci, “An optimal load shedding methodology for radial power distribution systems to improve static voltage stability margin using gravity search,” *Jurnal Teknologi*, vol. 68, no. 3, pp. 71–76, 2014.
- [30] Z. Sun and X. Gu, “Hybrid algorithm based on an estimation of distribution algorithm and cuckoo search for the no idle permutation flow shop scheduling problem with the total tardiness criterion minimization,” *Sustainability*, vol. 9, p. 953, 2017.
- [31] X. Li, C. Lu, L. Gao, S. Xiao, and L. Wen, “An effective multiobjective algorithm for energy-efficient scheduling in a real-life welding shop,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 5400–5409, 2018.
- [32] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI, USA, 1975.
- [33] 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, no. 99, pp. 161–176, 2016.
- [34] X. Zhu and N. Wang, “Cuckoo search algorithm with membrane communication mechanism for modeling overhead crane systems using RBF neural networks,” *Applied Soft Computing*, vol. 56, no. 56, pp. 458–471, 2017.
- [35] 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, no. 57, pp. 61–79, 2017.
- [36] L. Zhang and N. Wang, “Application of coRNA-GA based RBF-NN to model proton exchange membrane fuel cells,” *International Journal of Hydrogen Energy*, vol. 43, no. 1, pp. 329–340, 2018.
- [37] K. Wang and N. Wang, “A novel RNA genetic algorithm for parameter estimation of dynamic systems,” *Chemical Engineering Research and Design*, vol. 88, no. 11, pp. 1485–1493, 2010.
- [38] L. Zhang and N. Wang, “An adaptive RNA genetic algorithm for modeling of proton exchange membrane fuel cells,” *International Journal of Hydrogen Energy*, vol. 38, no. 1, pp. 219–228, 2013.
- [39] Q. Zhu, N. Wang, and L. Zhang, “Circular genetic operators based RNA genetic algorithm for modeling proton exchange membrane fuel cells,” *International Journal of Hydrogen Energy*, vol. 39, no. 31, pp. 17779–17790, 2014.
- [40] K. Wang and N. Wang, “A protein inspired RNA genetic algorithm for parameter estimation in hydrocracking of heavy oil,” *Chemical Engineering Journal*, vol. 167, no. 1, pp. 228–239, 2011.
- [41] C. Neuhauser and S. M. Krone, “The genealogy of samples in models with selection,” *Genetics*, vol. 145, no. 2, pp. 519–534, 1997.
- [42] F. Wang, Y. Rao, C. Zhang, Q. Tang, and L. Zhang, “Estimation of distribution algorithm for energy-efficient scheduling in turning processes,” *Sustainability*, vol. 8, no. 8, p. 762, 2016.
- [43] S. M. Werner and A. J. Ruthenburg, “Nuclear fractionation reveals thousands of chromatin-tethered noncoding RNAs adjacent to active,” *Cell Reports*, vol. 12, pp. 1089–1098, 2015.
- [44] J. M. Engreitz, “Long non-coding RNAs: spatial amplifiers that control nuclear structure and gene expression,” *Nature Reviews Molecular Cell Biology*, vol. 17, pp. 756–770, 2016.
- [45] M. D. Simon, C. I. Wang, P. V. Kharchenko et al., “The genomic binding sites of a noncoding RNA,” *Proceedings of the National Academy of Sciences*, vol. 108, no. 51, pp. 20497–20502, 2011.