Fish Migration Optimization with Dynamic Grouping Strategy for Solving Job-Shop Scheduling Problem

Qingyong Yang[1], Shu-Chuan Chu[1], Chia-Cheng Hu[2], Jimmy Ming-Tai Wu[1], Jeng-Shyang Pan[1,3*]

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, China
[2] College of Artificial Intelligence, Yango University, China
[3] Department of Information Management, Chaoyang University of Technology, Taiwan
yang_qy@sdust.edu.cn, scchu0803@gmail.com, cchu.chiachenghu@gmail.com, wmt@wmt35.idv.tw, jspan@cc.kuas.edu.tw

Abstract

Aiming at the job-shop scheduling problem (JSP), a dynamic grouping fish migration optimization (DFMO) is proposed to solve it. The DFMO algorithm adopts a multi-group structure to improve the convergence ability of the algorithm. And the opposition-based learning (OBL) strategy is applied to the group with poor overall fitness value to improve its solving environment. This paper proposes three different communication strategies to exchange information between different groups. In order to better determine the communication time between groups, a dynamic detection method based on population diversity is proposed. Compared with the static method of determining the communication time between groups, the proposed method can make the group more fully explore the current area and more hopefully find the optimal solution. The experiment in this paper is divided into two parts, one part is the numerical experiment test, the other part is the JSP problem standard library test. From the experimental results, the DFMO algorithm can obtain good results in both parts of the experiment, and has a good problem optimization ability.

Keywords: Fish migration optimization, Job-shop scheduling problem, Dynamic grouping strategy, Population diversity, Opposition-based learning

1 Introduction

Scheduling problem means that the final scheduling result can achieve the expected goal by reasonably arranging the sequence of each work. The goal can be to maximize the resource utilization of the system, or minimize the maximum completion time, etc. As the basic model of scheduling problems, the job-shop scheduling problem (JSP) has a wide range of application backgrounds, such as transportation [1-2], manufacturing, network communication [3], medical and health care, wireless sensor networks, cloud computing [4] and so on. Therefore, the research on the JSP problem has important theoretical value and practical significance [5-6].

As a complex combinatorial optimization problem, the JSP problem has been proved to be a typical NP-hard problem [7]. The algorithms for solving the JSP problem can be roughly divided into three categories: exact algorithms [8-9], heuristic algorithms and meta-heuristic algorithms [5, 10]. Although the exact algorithm can guarantee to find the optimal solution of the problem, it will take more time. With the increase of the scale of the JSP problem, the solution time of exact algorithm will become unbearable. The heuristic algorithm and the meta-heuristic algorithm can approximate the theoretical optimal solution and spend less time. At present, the main methods of solving the JSP problems are heuristic algorithms and meta-heuristic algorithms [5, 10]. The meta-heuristic algorithm is still a heuristic algorithm in essence, which adds different requirements to the search process on the basis of heuristic algorithm. And the meta-heuristic algorithm is not designed for a specific class of problems, so it has a wider application field [11-12]. The inspiration of meta-heuristic algorithms usually comes from various physical, chemical and biological phenomena in nature. By simulating certain behaviors and laws, the main body of the algorithm is constructed. Many meta-heuristics have existed so far. The representative algorithms are: simulated annealing (SA) [13], genetic algorithm (GA) [14-15], differential evolution (DE) [2, 16], tabu search (TS) [17-18], particle swarm optimization (PSO) [19-21], ant colony optimization (ACO) [22], artificial neural network (ANN) [23], grey wolf optimizer (GWO) [24], cuckoo search (CS) algorithm [25-26], phasmatodea population evolution (PPE) algorithm [27], sine cosine algorithm (SCA) [28], quasi-affine transformation evolution (QUATRE) [29], etc.

At present, more and more meta-heuristic algorithms have been successfully applied to solve the JSP problems. For example, Park et al. [30] designed a genetic algorithm to efficiently solve the JSP problem. Tasgetiren et al. [31] discretized the PSO algorithm and the DE algorithm, and successfully applied them to the solution of the JSP problem. Other meta-heuristic algorithms such as simulated annealing [32], tabu search [33], ant colony optimization [34], artificial neural network [35], and artificial bee colony [36] have also been successfully applied to solve the JSP problem. Some scholars proposed to mix the two meta-heuristic algorithms to solve the JSP problem more effectively. In [37], Ponsich and Coello mixed the DE algorithm and tabu search, and proposed the DE-TS algorithm. The hybrid DE-TS algorithm can solve the JSP problem better and has strong competitiveness. Literatures [38-39] give the algorithms design of the hybrid PSO algorithm and the hybrid ACO algorithm in solving the JSP problem respectively. With the popularity of deep learning in recent years, some scholars have also proposed to

*Corresponding Author: Jeng-Shyang Pan; E-mail: jspan@cc.kuas.edu.tw
DOI: 10.53106/160792642022112306010

apply the relevant technologies in deep learning to solve the JSP problem [40-43].

As a new meta-heuristic algorithm, the fish migration optimization (FMO) algorithm [44] simulates the swim and migration process of the graylings. The FMO algorithm has a good global search ability, but the local search ability is poor, which leads to poor optimization ability of the algorithm. At present, some studies have improved the FMO algorithm and applied it to different fields. Chai et al. [45] introduced the adaptive and self-assessment mechanism into the FMO algorithm to enhance the local search capability of the algorithm, and applied the improved algorithm to solve the localization problem in wireless sensor networks. In order to improve the optimization performance of the algorithm on complex functions, Guo et al. [46] used the theory of fractional calculus to improve the FMO algorithm. Pan et al. [47] binarized the FMO algorithm and successfully applied it to solve the unit commitment problem in the power system.

Aiming at the problems of poor solution accuracy and poor local search ability of the FMO algorithm, a dynamic grouping FMO algorithm (DFMO) is proposed in this paper. In the algorithm structure, the DFMO algorithm uses a multi-group structure. Existing studies have shown that multi-group structure can speed up the convergence speed and improve the optimization ability of the algorithm. At present, many intelligent optimization algorithms have been improved with multi-group structure, such as the PPSO algorithm [48], the PEO algorithm [49], the MFPA algorithm [50] etc. In the algorithms with multi-group structure, it is also necessary to formulate appropriate communication strategies and communication time among groups. In the DFMO algorithm, this paper formulates three communication strategies among different groups to complete the exchange of information. In the formulation of communication time among groups, previous studies often adopt a static method. Although this method can guarantee the completion of information exchange among groups, it will also lead to the destruction of group population information, which in turn leads to the deterioration of algorithm performance. Therefore, this paper proposes a dynamic detection method of communication time based on the population diversity [51]. In addition, the opposition-based learning (OBL) [52] strategy is also introduced in the DFMO algorithm to improve the global search ability. Finally, the test results show that the proposed DFMO algorithm can better solve numerical optimization problems and the JSP problem, and has a good optimization ability.

The rest of this paper is arranged as follows: the Section 2 describes the principle of the FMO algorithm, and establishes the mathematical model of the JSP problem. The improvement ideas and process design of the DFMO algorithm will be introduced in Section 3. The Section 4 is the numerical experiment, which is used to verify the optimization ability of the DFMO algorithm. The application and testing of the DFMO algorithm to the JSP problem will be covered in Section 5. The Section 6 is the summary of this paper and the improvement direction of future work.

2 Preliminaries

In this section, the mathematical model of the job-shop scheduling problem (JSP) and the optimization principle of the fish migration optimization (FMO) algorithm are introduced.

2.1 The Mathematical of Job-Shop Scheduling Problem

The JSP problem can be described as a processing factory using m machines to process n workpieces. The set of workpieces is expressed as $J = \{J_1, J_2, \dots, J_n\}$, and the set of processing machines is expressed as $M = \{M_1, M_2, \dots, M_m\}$. The machine and the corresponding processing time required for the processing of each workpiece are known and cannot be changed. The task of scheduling is to arrange a reasonable workpiece processing sequence to optimize the processing performance index under the condition of meeting the constraints.

The constraints are described as follows:

1) Each processing machine can only process one workpiece at a time and cannot be interrupted halfway;

2) Each workpiece is processed only once on each processing machine;

3) The failure of the machine is not considered;

4) Each workpiece must be processed according to the predetermined processing technology, and each workpiece has the same priority.

In this paper, the minimum machine makespan is selected as the performance index. The makespan of each machine is expressed by $T_j(j = 1,2,..,m)$. Based on the above description, the mathematical model of JSP problem is established [53].

Objective function:

$$f^* = min\{max(T_j)\}. \quad \textbf{(1)}$$

Subject to:

$$C_{ik} - p_{ik} + H(1 - a_{ihk}) \geq C_{ih}. \quad \textbf{(2)}$$

$$C_{jk} - C_{ik} + H(1 - x_{ijk}) \geq p_{jk}. \quad \textbf{(3)}$$

$$C_{ik} \geq 0. \quad \textbf{(4)}$$

$$x_{ijk} = \begin{cases} 1, & if\ condition\ 1\ is\ met \\ & 0, else \end{cases}. \quad \textbf{(5)}$$

$$a_{ihk} = \begin{cases} 1, if\ condition\ 2\ is\ met \\ \quad 0,\ \ else \end{cases}. \quad \textbf{(6)}$$

In the above formula, C_{ik} and p_{ik} respectively represent the completion time and processing time of the workpiece i on the processing machine k. Both variables are non-negative numbers. H is a large enough positive number. x_{ijk} and a_{ihk} are indicative variables and coefficients, and Equation (5) and (6) are the values of the two variables. Equation (1) is the objective function of the JSP problem, and is also the fitness function of the proposed algorithm. Equation (2) represents the processing sequence constraint of each workpiece. Equation (3) is the constraint of the machine required for each process of the workpiece. In Equation (5) and (6), condition 1 means that workpiece i is processed on machine k before workpiece j, and condition 2 means that machine h processes workpiece i before machine k.

2.2 Fish Migration Optimization

Fish Migration Optimization (FMO) is a new swarm intelligence optimization algorithm proposed by Pan et al. in 2010. The FMO algorithm is divided into two stages, which simulates the migration and swim process of the grayling respectively. Figure 1 shows the growth cycle of the grayling. "0+" to "4+" represents the 5 growth stages of the grayling from juvenile to adult.

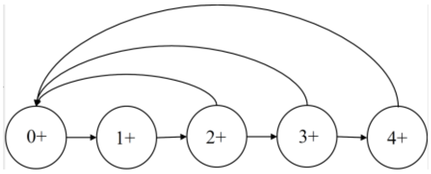


Figure 1. Life cycle graph of the grayling.

S represents the survival rate of the grayling. With the continuous growth of the grayling, the survival rate will be getting higher, that is $(S_1 < S_2 < S_3 < S_4)$. F represents the proportion of the graylings returning to their birthplace for reproduction. Similarly, with the continuous growth of the grayling, the proportion of reproduction will become higher, that is $(F_2 < F_3 < F_4)$. In the FMO algorithm, each individual is represented according to the data structure given in Equation (7).

$$X = < P, P_{pre}, Val, Eng, Phase >, \quad (7)$$

where X represents an individual of the grayling. P and P_{pre} represent the current and pervious position of the individual, respectively. Val is the fitness value calculated by P. Phase is the growth stage of the individual. Eng represents the energy. The greater the value, the stronger the migration ability. With the continuous growth of the grayling, the energy will gradually decrease. In the initialization phase, Eng is set to 2. Equation (8) gives the updated equation of Eng. i represents the i-th individual.

$$\mathrm{Eng(i)} = Eng(i) - \frac{Val(i)}{sum(Val)}. \quad (8)$$

The swim process actually simulates the process of the grayling looking for food. In this process, the individual grayling moves in the solution space through Equation (9).

$$P_i^{new} = P_i^{old} + \frac{consumption * ori\_speed}{a + b * abs(ori\_speed)^x}, \quad (9)$$

where, P_i^{old} and P_i^{new} represent the current position and the updated position of the i th individual, respectively. a, b and x are all constants. In the FMO algorithm, they are set to 5.27, 11.44 and 2.44 respectively. $consumption$ refers to the random energy consumption in the process of individual movement, and the calculation formula is shown in Equation (10). Parameter $r1$ is the random number between 0 and 1, and E_max represents the maximum energy consumption of the individual movement, which is set as a constant 2.

$$\mathrm{consumption} = \begin{cases} r1 * E_{max}, if\ Eng(i) > E\_max \\ r1 * Eng(i),\ \ else \end{cases}. \quad (10)$$

Equation (11) is the calculation formula of ori_speed, and $r2$ is the random number between 0 and 1.

$$\mathrm{ori\_speed} = \begin{cases} P - P_{pre}, if\ r2 > 0.5 \\ P_{pre} - P, else \end{cases}. \quad (11)$$

When the grayling individuals mature, some individuals will return to their birthplaces to reproduce and breed offspring. In this process, the FMO algorithm first counts the number of individuals in each stage. Since individuals with growth stages of "0+" and "1+" do not have the ability to reproduce, so these individuals only update the growth stage. For individuals with growth stages of "2+" and "3+", some individuals will return to their birthplace, and others will only update their growth stage. For individuals returning to their birthplace, the position will be updated according to Equation (12), and the $Phase$ will be updated to "0+". In the Equation (12), P_{gbest} is the global optimal solution, and $r3$ is the random number. Individuals whose growth stage is "4+" will all return to their birthplace and their $Phase$ will be updated to "0+".

$$P = P_{gbest} + r3 * (P - P_{gbest}). \quad (12)$$

Through the above process, FMO algorithm can better solve many complex optimization problems. However, the FMO algorithm still has the problem of poor global search ability and low solution accuracy. In order to further improve the optimization ability of the FMO algorithm, this paper proposes the DFMO algorithm. And in the next section, we will introduce the improvement process of the FMO algorithm in this paper.

3 Dynamic Grouping Fish Migration Optimization

To solve the problem of poor accuracy of the FMO algorithm, this paper proposes a dynamic grouping FMO algorithm (DFMO). Different from the FMO algorithm structure, the DFMO algorithm adopts multi-group (population) structure. Compared with the single population algorithm structure, the multi-group algorithm structure can greatly improve the search range of the algorithm, and is more beneficial to explore the promising areas in the space.

The DFMO algorithm consists of two groups: one named group B and one named group W. The group optimal solution is expressed by $pbest$, and the global optimal solution is expressed by $gbest$. The two groups contained the same number of individuals, but the fitness value of individuals in group B was better than that in group W. Therefore, at each regrouping, some individuals in the two groups will be exchanged. After each grouping, group B updates its $pbest$ to $gbest$. The purpose of this operation is to enable group B to search the region near $gbest$ more fully, so as to enhance the local search ability of the algorithm. After grouping, group W will execute the opposition-based learning (OBL) strategy, so as to expand the search space of the group and enhance the global search ability of the algorithm. Equation (13) is the individual update formula based on the opposition-based learning strategy.

$$X_{i,op}^{W} = \mathrm{lb} + \mathrm{ub} - X_{i}^{W}, \quad \textbf{(13)}$$

where, lb and ub are the lower and upper bounds of the current search space.

In the DFMO algorithm, this paper formulates three different communication strategies to realize information sharing between different groups (populations). Before information exchange, the optimal values of the two groups need to be compared. The winning side, we think, has the better optimization ability to guide the evolution of the poorer side. For the convenience of description, this paper assumes that the winning side is represented by X_{win}, the losing side by X_{lose}, and the optimal solution of the winning side is represented by X_{win}.

Communication Strategy 1:

In Strategy 1, the optimal individual in the winning side guides the evolution of all individuals in the losing side, as shown in Equation (14).

$$X_{lose,i} = X_{lose,i} + 2 * rand * abs\left(X_{win,best} - X_{lose,i}\right), \quad \textbf{(14)}$$

where, $rand$ is a random number between 0 and 1.

Communication Strategy 2:

Strategy 2 adds the influence of random individuals on the basis of Strategy 1. In Strategy 2, the individual in the losing side randomly selects an individual from the winning side. If the fitness value of the randomly selected individual is better than that of the losing side, the optimal individual of the winner and the randomly selected individual jointly guide the evolution of this individual. Otherwise, the method of Strategy 1 is still adopted. Specifically, the formulas are shown in Equation (15) and (16).

$$\mathrm{weight} = \begin{cases} \frac{abs(X_{win,rand} - X_{lose,i}) + abs(X_{win,best} - X_{lose,i})}{2}, & if\ Fit\left(X_{win,rand}\right) < Fit\left(X_{lose,i}\right) \\ abs\left(X_{win,best} - X_{lose,i}\right), & else \end{cases}. \quad \textbf{(15)}$$

$$X_{lose,i} = X_{lose,i} + 2 * rand * weight. \quad \textbf{(16)}$$

Communication Strategy 3:

Both Strategy 1 and 2 are used for all individuals of the losing side. In Strategy 3, only some individuals are selected to accept the evolutionary guidance of the winner. In the DFMO algorithm, these individuals are those whose fitness value is greater than the average value of the failed group. The inter-group communication equation of Strategy 3 still uses the communication equation of Strategy 2, namely Equations (15) and (16).

In the multi-group algorithms, while formulating the communication strategies among groups, it is also necessary to set the appropriate communication time. However, previous studies usually set the communication time to static, that is, it usually adopts a fixed number of iterations at a certain interval. Although this method can ensure the information sharing among groups, it may lead to the situation that the current group fails to fully explore the current region due to the introduction of information from other groups, and even lead to the deterioration of the solving quality of the group. So the communication time needs to be determined dynamically according to the state of the population. In order to capture the dynamic changes of population state, the concept of population diversity is introduced in this paper. By measuring changes in population diversity, communication strategies and individual exchanges between different groups are dynamically executed. For the measurement of population diversity, this article adopts the measurement method of literature [51]. This method is relatively simple to implement, and the calculation cost is also low. Equation (17) is the measurement formula of population diversity.

$$\mathrm{Diversity} = \mathrm{ps} * \left[\sum_{i=1}^{Dim} \sum_{j=1}^{ps} (X_{ij} - C_i)\right], \quad \textbf{(17)}$$

where, ps is the number of individuals in the population, and Dim is the dimension of the current solution space. C_i represents the center position of the i th column in the population X, that is, the mean value of the ith column.

The execution flow chart of the DFMO algorithm is shown in Figure 2. Ds in the figure is the threshold of population diversity, which is set as $1e - 10$ in the algorithm. Below this threshold, it is considered that the population has stagnated. $flag$ is the flag to judge whether to regroup. It should be noted that the DFMO algorithm is still a single population structure at the beginning, which can better accelerate the convergence speed of the algorithm in the early stage.

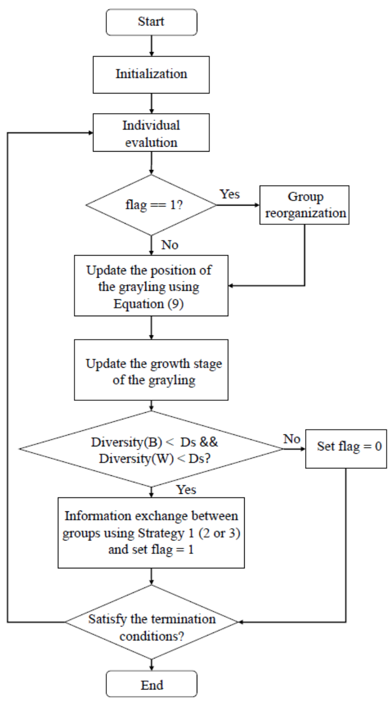


Figure 2. Flow chart of the DFMO algorithm

4 Numerical Experimental Analysis

In order to verify the effectiveness of the DFMO algorithm, this paper selects 23 benchmark functions for performance testing. Table 1 lists the details of the 23 benchmark functions. The f^* column is the optimal value of the function. The experiment of this part is divided into two parts. One part is the comparison experiment between three versions of the DFMO algorithm and the original FMO algorithm to verify the performance improvement of the proposed improvement idea on the original FMO algorithm. The other part is the comparison between the DFMO algorithm and other swarm intelligence algorithms to verify the competitiveness of the

DFMO algorithm. During the experiment, the maximum number of iterations of each algorithm is set to 1000 and the maximum number of particles is set to 30. Each algorithm has 30 consecutive tests on each function. The optimal value, mean and standard deviation of the experiment are recorded. The Matlab version is R2019b and the operating system is Windows 10.

Table 1. The details of the benchmark functions

Function	Dim	Bound	f^*
Sphere	30	[-100, 100]	0
Schwefel's function 2.21	30	[-10, 10]	0
Schwefel's function 1.2	30	[-100, 100]	0
Schwefel's function 2.22	30	[-100, 100]	0
Rosenbroke	30	[-30, 30]	0
Step	30	[-100, 100]	0
Dejong's noisy	30	[-1.28, 1.28]	0
Schwefel	30	[-500, 500]	-12569
Rastringin	30	[-5.12, 5.12]	0
Ackley	30	[-32, 32]	0
Griewank	30	[-600, 600]	0
Generalized penalized 1	30	[-50, 50]	0
Generalized penalized 2	30	[-50, 50]	0
Fifth of Dejong	2	[-65, 65]	1
Kowalik	4	[-5, 5]	0.0003
Six-hump camel back	2	[-5, 5]	-1.0316
Branins	2	[-5, 5]	0.398
Goldstein-Price	2	[-2, 2]	3
Hartman 1	3	[1, 3]	-3.86
Hartman 2	6	[0, 1]	-3.32
Shekel 1	4	[0, 10]	-10.1532
Shekel 2	4	[0, 10]	-10.4028
Shekel 3	4	[0, 10]	-10.5363

4.1 Comparison Between the DFMO Algorithm and the FMO Algorithm

Table 2 records the comparison of experimental results of the FMO algorithm and the DFMO algorithm, and the better results are bolded. DFMO_vi represents the DFMO algorithm that adopts the ith (i=1,2,3) communication strategy. From the experimental results, the number of better results obtained by the DFMO algorithm is significantly better than that of the FMO algorithm, and the optimization ability has been greatly improved. The stability has also been improved. The DFMO algorithm is only weaker than the FMO algorithm on the F21-23 functions.

Comparing the three inter-group communication strategies of the DFMO algorithm, the DFMO_v2 algorithm adopting the second communication strategy is slightly better than the other two strategies in terms of optimal value, mean and standard deviation. Although the DFMO_v1 algorithm can obtain better results on the optimal value, its stability is poor. For example, on functions F3 and F12, the optimal value obtained by the DFMO_v1 algorithm is the best among the three DFMO algorithms, but the mean and standard deviation are the worst among the three. Therefore, it is uncertain to improve the algorithm only by using the optimal individual to guide the evolution. It may bring benefits to the algorithm, and it may also lead to the deterioration of the algorithm. Strategies 2 and 3 adopt the way that some individuals accept the best individuals to guide the evolution. While achieving better results, the algorithm can also be relatively stable.

4.2 Comparison with Other Swarm Intelligence Algorithms

In order to verify the competitiveness of the proposed DFMO algorithm proposed, this paper selects the PSO algorithm, the SCA algorithm and the parallel PSO algorithm (PPSO) as the comparison objects. In the experiment, the parameters $c1$ and $c2$ of the PSO algorithm are set to 2 and ω is set to 0.9. In the SCA algorithm, parameter a is set to 2. In the PPSO algorithm, the number of groups is set to 2, $c1$ and $c2$ are set to 2, and ω is set to 0.7. Table 3 records the experimental results of the DFMO algorithm and the other three algorithms, and the better results are bolded. The DFMO algorithm version in Table 3 is the DFMO_v2 in Table 2.

From the results in the Table 3, the DFMO algorithm, the PSO algorithm and the PPSO algorithm are close in the number of optimal values, and the three algorithms have achieved theoretical optimal values on functions F9, F11 and F16. However, the DFMO algorithm is more competitive than the other three algorithms in the number of mean and standard deviation. Both the DFMO algorithm and the PPSO algorithm are multi-group structures. And from the experimental results, they are better than the SCA algorithm and the PSO algorithm with a single population structure in terms of optimal value, mean and standard deviation. But the DFMO algorithm has better optimization ability than PPSO algorithm. This is because the communication time adopted by the PPSO algorithm is static, which may make the group fail to fully explore the current region and lead to the deterioration of the optimization ability of the algorithm. The DFMO algorithm, by measuring the population diversity of each group, dynamically determines the communication time, which can ensure that each group can fully explore its own solution area, so as to improve the local search ability of the algorithm and obtain better solution results.

In order to further illustrate the comparison between DFMO algorithm and other swarm intelligence algorithms, Wilcoxon rank-sum test is also carried out in this paper. During the test, the significance level α is set to 0.05. The relevant experimental data is recorded in Table 4. The "+" symbol indicates that the optimization performance of the DFMO algorithm is better than the compared algorithm, and the "-" symbol indicates that the DFMO is worse than the compared algorithm. According to the results in the Table 4, the DFMO algorithm is superior to other three algorithms on the whole.

To verify the convergence of the DFMO algorithm, the convergence curves of four algorithms are selected as the verification method. Figure 3 shows the convergence curves of the four algorithms on functions F3, F6, F8, F9, F10, F21, F22, and F23. It can also be seen from the convergence curve that the convergence speed of the DFMO algorithm is faster than the other three algorithms, and has better convergence performance.

Table 2. The results of simulation experiments

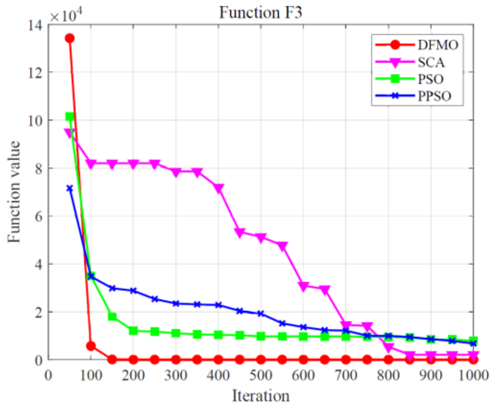
Function	FMO			DFMO v1			DFMO v2			DFMO v3		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
F1	54443.96564	68176.26235	7446.581344	7.66E-75	5.81295E-54	3.18386E-53	**6.41E-83**	**2.74866E-77**	**9.16536E-77**	5.01E-13	6.76E-10	1.99414E-09
F2	5.31222E+39	1.12862E+43	2.30512E+43	2.37E-41	8.87767E-37	3.675E-36	**7.62E-43**	**1.49502E-40**	**4.18981E-40**	0.000223	0.007822	0.012355092
F3	77623.88053	159208.3124	43910.82439	**5.8E-55**	27513.94153	26502.95627	6.88E-49	**3.15944E-37**	**1.03225E-36**	0.047175	14.13235	31.46629971
F4	79.19042006	87.76900118	3.110648609	1.42E-20	78.56443842	22.22196408	**8.38E-38**	**1.50672E-34**	**4.42173E-34**	1.49E-07	3.59E-06	4.66222E-06
F5	190671319.5	257423892.9	36572183.44	28.60294	18382574.46	45022485.04	**28.53862**	28.73828569	0.058706857	28.55216	**28.7239**	**0.043477388**
F6	40412.95169	66963.58531	8417.805175	**1.033977**	2.695765552	0.614051609	1.604405	**2.414645069**	**0.337315975**	3.050836	3.655558	0.362548824
F7	66.54516113	123.0034198	25.18404367	**6.44E-06**	12.01643293	35.39371643	1.19E-05	**0.00014107**	**0.000117854**	2.78E-05	0.000312	0.000252047
F8	-3904.351782	-2224.708821	610.0306723	**-12569.5**	-12434.26487	586.8173074	**-12569.5**	-12331.07111	1302.165135	**-12569.4**	**-12566.3**	**4.2114779**
F9	381.749036	441.2136255	22.00062585	**0**	**0**	**0**	**0**	**0**	**0**	**0**	1.06E-10	3.28125E-10
F10	20.2439856	20.66950842	0.134370795	**8.88E-16**	**8.88178E-16**	**0**	**8.88E-16**	**8.88178E-16**	**0**	1.26E-09	5.45E-08	9.28077E-08
F11	484.5055285	603.954638	64.69438001	**0**	**0**	**0**	**0**	**0**	**0**	1.64E-13	1.31E-10	2.86678E-10
F12	234360140.1	545167867.5	133502219.9	**0.024728**	192493779.6	238274552.4	0.061469	**0.149980393**	**0.064574383**	0.257427	0.470248	0.177434593
F13	633263220.3	1127598172	249587339.2	0.705971	197172531.3	310307046.9	**0.231183**	0.620708907	0.178543174	0.25118	**0.483854**	**0.145942196**
F14	6.51596933	143.4591811	146.8267019	**0.998004**	3.811517245	3.620531755	**0.998004**	2.859565842	3.468293763	**0.998004**	**1.765655**	**0.925796076**
F15	0.005930128	0.122777272	0.10225652	**0.000312**	0.002188669	0.001741932	0.000435	0.002211493	**0.001418953**	0.000322	**0.001751**	0.001493497
F16	-1.007465125	1.297567804	2.524035425	**-1.03163**	-1.026318843	0.016798319	**-1.03163**	**-1.031598089**	**0.000115563**	**-1.03163**	-1.02727	0.010453825
F17	0.45278412	2.522995405	2.341547372	**0.39789**	**0.398861612**	**0.002030387**	**0.397889**	0.399117827	0.003524538	**0.397888**	0.407212	0.024567717
F18	3.347998696	72.00285869	88.97223937	**3**	**3.00124333**	0.002184056	3.000026	3.003683459	0.008499798	3.000213	3.011605	0.018470822
F19	-3.813999232	-3.425622358	0.370877765	-3.86256	-3.741663719	0.086039668	**-3.86276**	-3.745946516	0.145654267	-3.8559	**-3.75329**	**0.078081986**
F20	-2.110525285	-1.291600351	0.412166498	-3.22115	-2.843621761	0.476386978	**-3.30813**	**-3.123959616**	**0.097760988**	-3.20711	-2.88664	0.233026077
F21	-1.900612114	-0.59088347	**0.375700945**	**-10.1532**	-6.332409196	3.880489764	-10.1531	**-9.242086476**	2.069120832	**-10.1532**	-9.1151	2.090745413
F22	-1.323445959	-0.613496763	**0.21810435**	**-10.4027**	-5.360218849	3.654133017	**-10.4028**	-7.749394958	3.10516732	**-10.4028**	**-9.78119**	1.88125415
F23	-2.399574213	-0.908403214	0.467424692	-10.5362	-6.885030407	3.883833826	**-10.5363**	**-9.59815583**	2.1250152	-10.5362	-8.56463	2.817238304
Sum	0	0	3	15	5	5	16	14	13	7	7	5

Table 3. The statistical results of the DFMO algorithm and other three swarm intelligence algorithms

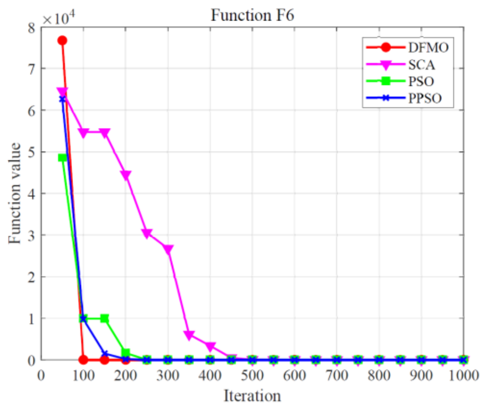
Function	DFMO			SCA			PSO			PPSO		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
F1	6.41146E-83	2.74866E-77	9.16536E-77	2.69E-06	0.0505	0.161903	**0**	1333.333333	3457.459036	**0**	**0**	**0**
F2	7.62164E-43	**1.49502E-40**	**4.18981E-40**	2.92E-07	5.23E-05	7.99E-05	**0**	110	109.3870067	**0**	2.652324	14.52737691
F3	6.87523E-49	**3.15944E-37**	**1.03225E-36**	231.382	3633.389	3529.051	**0**	15645.0882	8692.64672	**0**	490.4217	1262.276352
F4	8.38104E-38	1.50672E-34	4.42173E-34	2.004366	17.98619	11.05054	**0**	**0**	**0**	**0**	**0**	**0**
F5	28.5386165	**28.73828569**	**0.058706857**	**28.16552**	201.479	425.9967	28.77225	9287.973281	27386.12145	28.85031	125.0836	418.3718468
F6	1.604405201	**2.414645069**	**0.337315975**	3.981857	4.742449	0.708074	3.208252	1001.534089	3041.337333	**1.283955**	3.564684	0.753311503
F7	**1.18769E-05**	**0.00014107**	**0.000117854**	0.001433	0.026797	0.028499	0.000808	1.97907892	4.56601418	0.000211	0.008562	0.006334807
F8	**-12569.48561**	**-12331.07111**	1302.165135	-4622.82	-3881.29	**242.3523**	-10282.1	-8335.730938	793.6447516	-9951.52	-8168.02	1128.411939
F9	**0**	**0**	**0**	3.42E-05	20.2957	25.98259	**0**	40.24431503	34.89039772	**0**	28.18958	35.95304908
F10	**8.88178E-16**	**8.88178E-16**	**0**	0.000149	13.62697	8.701756	**8.88E-16**	3.282461862	6.061361599	**8.88E-16**	**8.88E-16**	**0**
F11	**0**	**0**	**0**	8.74E-05	0.157621	0.225942	**0**	9.045004074	27.59896339	**0**	**0**	**0**
F12	**0.061468771**	**0.149980393**	**0.064574383**	0.50269	4.185299	5.581628	0.097938	0.368471839	0.149316848	0.062356	0.199487	0.088963627
F13	**0.231183478**	**0.620708907**	**0.178543174**	2.283731	140.9698	575.6916	1.679716	2.60662873	0.327626169	2.144481	2.554139	0.223013957
F14	**0.998003838**	2.859565842	3.468293763	**0.998004**	1.329502	0.751709	**0.998004**	**0.998003838**	1.7529E-10	**0.998004**	**0.998004**	**1.577E-10**
F15	0.000434512	0.002211493	0.001418953	**0.000324**	**0.000929**	**0.000422**	0.000345	0.004712117	0.007258793	0.000515	0.00284	0.005948968
F16	**-1.031628335**	**-1.031598089**	0.000115563	**-1.03163**	**-1.03161**	1.81E-05	**-1.03163**	**-1.031620436**	1.73866E-05	**-1.03163**	**-1.03163**	**8.01686E-07**
F17	**0.397889165**	0.399117827	0.003524538	0.397954	**0.398781**	**0.001023**	**0.397888**	0.57267727	0.522247392	0.397925	0.463244	0.113514149
F18	3.000025505	3.003683459	0.008499798	**3.000001**	**3.000013**	**1.51E-05**	**3**	3.000024736	4.20318E-05	**3**	3.000017	3.24566E-05
F19	-3.862755388	-3.745946516	0.145654267	-3.86149	**-3.85478**	**0.001938**	-3.8148	-3.459847198	0.329196166	**-3.86096**	-3.71078	0.170743021
F20	**-3.308128835**	**-3.123959616**	**0.097760988**	-3.26165	-2.99381	0.179043	-2.66197	-1.371838249	0.510292952	-3.23502	-2.08805	0.70843629
F21	**-10.15310323**	**-9.242086476**	2.069120832	-5.55057	-2.03189	1.808624	-3.36983	-0.69587288	**0.593342816**	-4.84365	-2.40599	1.067293591
F22	**-10.402819**	**-7.749394958**	3.105165732	-5.48413	-2.71799	1.745256	-2.66385	-0.817496029	**0.521887683**	-4.4561	-2.07772	1.041542403
F23	**-10.53633175**	**-9.59815583**	2.1250152	-8.46635	-4.26134	1.885789	-1.9146	-0.834422478	**0.335514144**	-9.12941	-2.3765	1.459623931
Sum	14	16	11	5	5	5	11	3	4	12	6	6

Table 4. The experimental data of Wilcoxon rank-sum test (significance level α = 0.05)

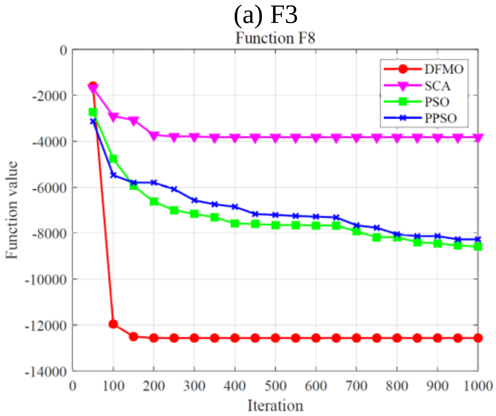
Function	FMO		SCA		PSO		PPSO	
	p	h	p	h	p	h	p	h
F1	1.51E-11	1	1.51E-11	1	1	0	1	0
F2	1.51E-11	1	1.51E-11	1	0.003802	1	1	0
F3	1.51E-11	1	1.51E-11	1	5.52E-07	1	0.999982	0
F4	1.51E-11	1	1.51E-11	1	1	0	1	0
F5	1.51E-11	1	5.33E-08	1	1.51E-11	1	7.32E-11	1
F6	1.51E-11	1	1.51E-11	1	5.47E-11	1	4.53E-08	1
F7	1.51E-11	1	1.51E-11	1	1.51E-11	1	1.51E-11	1
F8	1.51E-11	1	1.51E-11	1	1.51E-11	1	1.51E-11	1
F9	6.06E-13	1	6.06E-13	1	9.32E-10	1	6.52E-08	1
F10	6.06E-13	1	6.06E-13	1	0.080401	0	1	0
F11	6.06E-13	1	6.06E-13	1	0.040761	1	1	0
F12	1.51E-11	1	1.51E-11	1	3.06E-10	1	0.027773	1
F13	1.51E-11	1	1.51E-11	1	1.51E-11	1	1.51E-11	1
F14	1.19E-08	1	0.962586	0	1	0	1	0
F15	1.74E-10	1	0.999024	0	0.031766	1	0.945655	0
F16	6.03E-11	1	0.025939	1	0.918812	0	0.999954	0
F17	2.29E-09	1	6.27E-08	1	2.5E-09	1	1.01E-08	1
F18	1.51E-11	1	1	0	1	0	1	0
F19	0.000618	1	0.999941	0	8.49E-09	1	0.015159	1
F20	2.49E-11	1	0.001312	1	1.51E-11	1	5.78E-08	1
F21	1.51E-11	1	1.91E-10	1	1.51E-11	1	1.84E-11	1
F22	2.75E-11	1	1.02E-09	1	1.51E-11	1	3.06E-10	1
F23	1.51E-11	1	3.85E-08	1	1.51E-11	1	4.96E-11	1



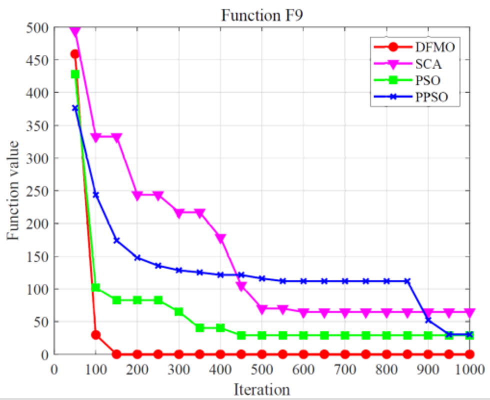
(a) F3



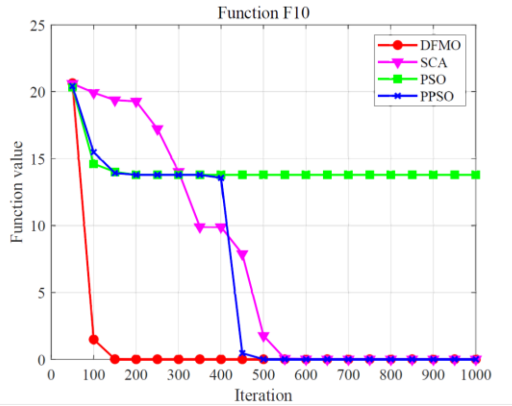
(b) F6



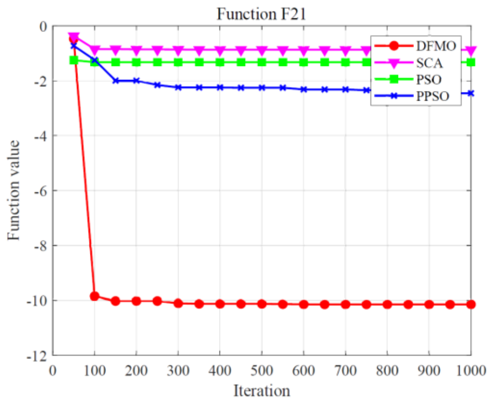
(c) F8



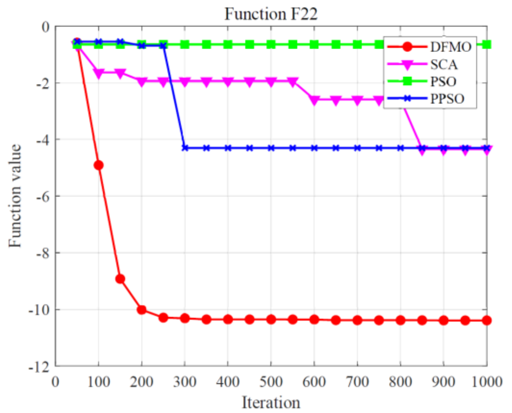
(d) F9



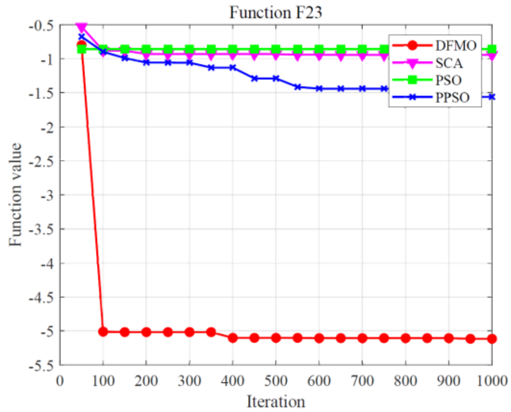
(e) F10



(f) F21



(g) F22



(h) F23

Figure 3. Convergence curve of functions F3, F6, F8, F9, F10, F21, F22, and F23

5 The Application of the DFMO Algorithm in the JSP problem

5.1 Coding Scheme

Since the JSP problem is a discrete space problem, the FMO algorithm and the DFMO algorithm proposed in this paper are both algorithms for solving continuous space problems. Therefore, it is necessary to use a suitable coding scheme to complete the mapping between the two. This paper adopts the coding rule in the literature [54] to code the individuals in the algorithm.

In the algorithm, each individual is a vector of $n \times m$, which is used to represent the arrangement of a process. For example, for a JSP problem with three workpieces and two processing machines. Assuming that the sequence of one individual is [1, 2, 3, 1, 3, 2], the corresponding processing sequence is [$J_{1,1}$, $J_{2,1}$, $J_{3,1}$, $J_{1,2}$, $J_{3,2}$ $J_{2,2}$]. $J_{i,j}$ represents the jth processing process of the ith workpiece. By matching the decoded processing sequence with the machine required for processing each workpiece determined in advance, a scheduling plan is obtained.

5.2 Analysis of Experimental Results

In order to verify the effectiveness of the proposed DFMO algorithm in solving the JSP problem, this paper selects the standard test library of the JSP problem for testing. In this paper, three examples are selected from the FT [55] and seven examples are selected from the LA [56] test libraries respectively. The results are compared with the FMO algorithm, the PSO algorithm and the SCA algorithm.

In the experiment, the number of particles of each algorithm is 50 and the maximum number of iterations is 1000. The PSO algorithm and the SCA algorithm still use the parameter settings in the previous section. Each algorithm runs independently for 20 times. Table 5 records the experimental results of the four algorithms. Among them, the Size column represents the scale of the problem to be solved. For example, "10×5" means that the number of workpieces to be processed is 10, and the number of processing machines is 5. The DFMO algorithm is overall better than the other three algorithms in terms of optimal and mean values. However, it can also be seen from the data in the table that there are certain problems in the stability of the DFMO algorithm, which leads to a large difference between the mean value and the optimal value of the algorithm, which affects the final optimization success rate. For example, in the FT10, LA07 and LA16 examples, although the DFMO algorithm obtains better mean and optimal values than the PSO and SCA algorithms, the std values are worse than the PSO and SCA algorithms.

Figure 4 shows the convergence curves of the four algorithms in solving the above some examples. As can be seen from the figure, compared with the other three algorithms, the DFMO algorithm has a faster convergence speed. For example, in the FT06 and FT20 examples, although the final solution results of the SCA algorithm and the DFMO algorithm are the same, the convergence speed of the DFMO algorithm is faster. It can also be seen from the figure that the

FMO algorithm has premature convergence. Thanks to the introduction of multi-group structure and opposition-based learning strategy, the DFMO algorithm can jump out of the local optimal solution in time, which improves the convergence ability of the original FMO algorithm.

On the whole, the DFMO algorithm can better solve the JSP problem and is an effective algorithm. However, the stability of the DFMO algorithm is poor. Therefore, how to improve the stability and the optimization ability of the DFMO algorithm will be the focus of future research work.

Table 5. The experimental results of the JSP problem examples

Example	Size	FMO			DFMO			PSO			SCA		
		Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
FT06	6x6	60	63	2.675424	**55**	56.7	1.719853	**55**	58.2	1.239694	55	**55.5**	**0.88852**
FT10	10x10	1277	1371.95	50.27972	**1056**	**1155.7**	56.91462	1127	1181.2	42.924117	1164	1206.75	**26.4871**
FT20	20x5	1582	1705.95	60.10471	**1370**	**1467.9**	42.55016	1423	1513.55	47.9731	1425	1483.35	**30.0863**
LA01	10x5	761	808.15	30.81912	**666**	**689.8**	**19.7926**	**666**	699.2	25.15552	**666**	697.85	21.18409
LA03	10x5	721	760.3	27.70541	**631**	**660.25**	17.13683	640	675.95	16.64008	651	671.25	**13.0379**
LA06	15x5	991	1057.55	45.06892	**926**	**934.2**	**12.1118**	926	941.1	18.5214	926	945.9	14.97331
LA07	15x5	1033	1104.3	41.57694	**921**	**975.75**	29.44375	952	981	**21.2924**	944	979.1	21.30456
LA09	15x5	1010	1099.45	52.79003	**951**	**955.1**	**7.35491**	**951**	981.35	19.88989	**951**	973.65	14.27281
LA11	20x5	1345	1404.35	45.59233	**1222**	**1237.5**	16.8623	**1222**	1268.8	35.11575	1238	1277.75	16.1143
LA16	10x10	1147	1263.85	56.94436	**982**	**1069.8**	46.55907	1042	1091.45	29.35172	1049	1112.45	**27.5556**

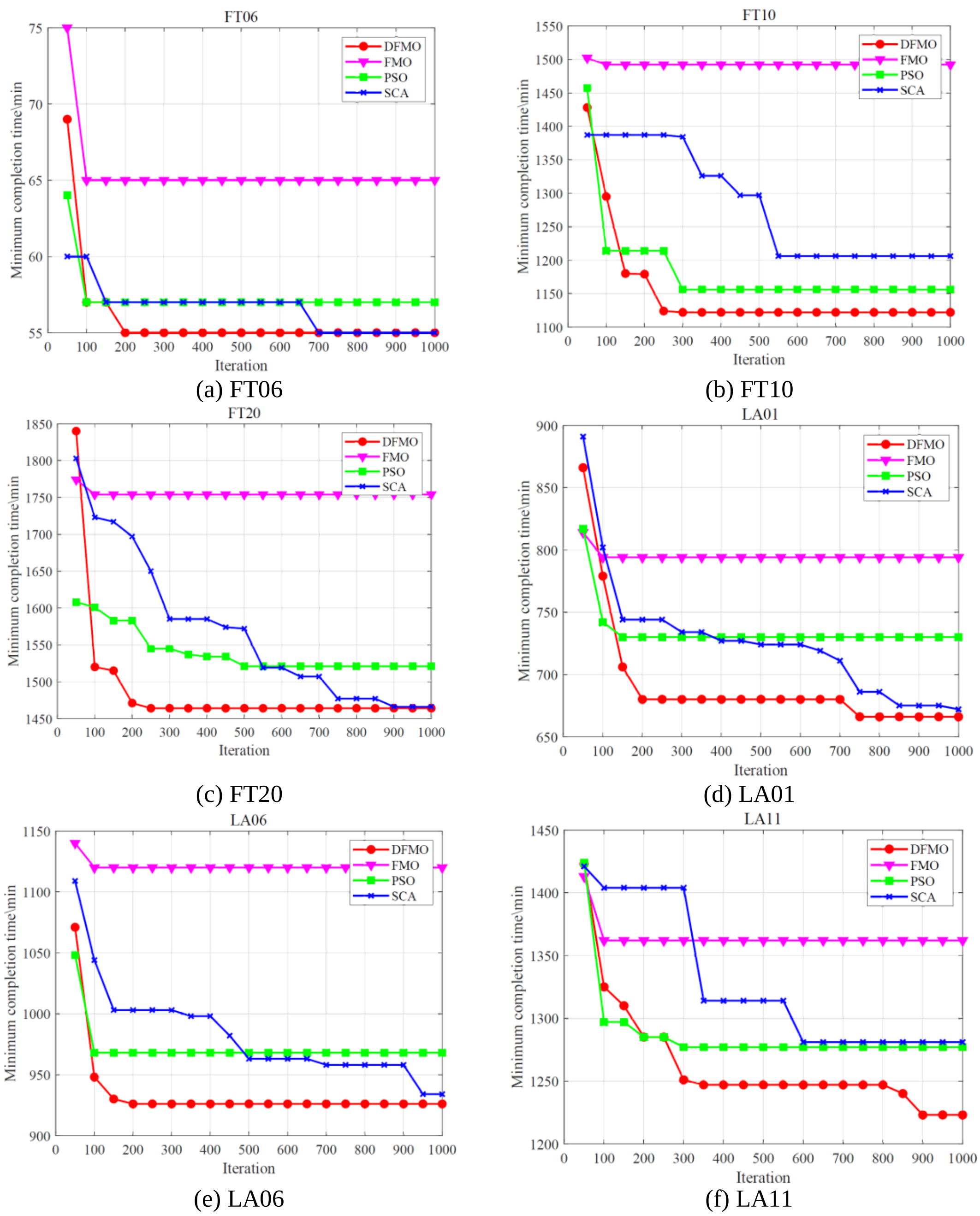


Figure 4. Convergence curve of the JSP problem examples FT06, FT10, FT20, LA01, LA06, and LA11

6 Conclusion

In order to better solve the JSP problem, this paper proposes a dynamic grouping FMO algorithm (DFMO). Firstly, the population in the FMO algorithm are grouped, which is improved from the original single population structure to multi-group structure. Each group contains the same individuals and does not affect each other before communication between groups. In this paper, three different communication strategies are developed to realize information sharing between different groups. Different from previous studies, this paper adopts a method based on population diversity to dynamically determine the communication time between groups, which can ensure that each group has fully mined the current region and improve the local search ability of the algorithm. In addition, the opposition-based learning strategy is introduced to expand the search space of the DFMO algorithm. Through the test of 23 benchmark functions, the function optimization ability and the feasibility of the improvement idea of the proposed DFMO algorithm are verified. Finally, in order to verify the effectiveness of the DFMO algorithm in solving the JSP problem, this paper selects relevant examples for testing. From the experimental results, the DFMO algorithm can solve the JSP problem better and is an effective method.

In this paper, the number of groups of the DFMO algorithm only considers the division into two groups. When there are more than two groups, how the architecture and policies proposed in this article work will also be a problem to be solved in the future. At the same time, in the future work, this paper will also focus on solving other problems of DFMO algorithm and expanding the application fields of the DFMO algorithm [57-61].

References

[1] H. E. Nouri, O. B. Driss, K. Ghédira, A classification schema for the job shop scheduling problem with transportation resources: state-of-the-art review, in: R. Silhavy, R. Senkerik, Z. Oplatkva, P. Silhavy, Z. Prokopova (Eds.), *Advances in Intelligent Systems and Computing*, Vol. 464, Springer, Cham, 2016, pp. 1-11.

[2] T. Wang, K. Wu, T. Du, X. Cheng, Adaptive weighted dynamic differential evolution algorithm for emergency material allocation and scheduling, *Computational Intelligence*, pp. 1-17, August, 2020, https://doi-org.uconn.80599.net/10.1111/coin.12389.

[3] C.-C. Lin, D.-J. Deng, Y.-L. Chih, H.-T. Chiu, Smart manufacturing scheduling with edge computing using multiclass deep Q network, *IEEE Transactions on Industrial Informatics*, Vol. 15, No. 7, pp. 4276-4284, July, 2019.

[4] L. Tang, K. Ma, Z. Li, A new scheduling algorithm based on ant colony algorithm and cloud load balancing, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 8, No. 1, pp. 191-199, January, 2017.

[5] J. Zhang, G. Ding, Y. Zou, S. Qin, J. Fu, Review of job shop scheduling research and its new perspectives under industry 4.0, *Journal of Intelligent Manufacturing*, Vol. 30, No. 4, pp. 1809-1830, April, 2019.

[6] D. Gao, G.-G. Wang, W. Pedrycz, Solving fuzzy job-shop scheduling problem using DE algorithm improved by a selection mechanism, *IEEE Transactions on Fuzzy Systems*, Vol. 28, No. 12, pp. 3265-3275, December, 2020.

[7] M. R. Garey, D. S. Johnson, R. Sethi, The complexity of flowshop and jobshop scheduling, *Mathematics of operations research*, Vol. 1, No. 2, pp. 117-129, May, 1976.

[8] D. Applegate, W. Cook, A computational study of the job-shop scheduling problem, *ORSA Journal on computing*, Vol. 3, No. 2, pp. 149-156, May, 1991.

[9] P. Brucker, B. Jurisch, B. Sievers, A branch and bound algorithm for the job-shop scheduling problem, *Discrete applied mathematics*, Vol. 49, No. 1-3, pp. 107-127, March, 1994.

[10] B. Calis, S. Bulkan, A research survey: review of AI solution strategies of job shop scheduling problem, *Journal of Intelligent Manufacturing*, Vol. 26, No. 5, pp. 961-973, October, 2015.

[11] J. Chen, J. Zhang, X. Xue, Y. Huang, Multi-view ontology alignment visualization (MOAV): A human-cognition based ontology alignment visualization technique, *Journal of Network Intelligence*, Vol. 5, No. 4, pp. 198-210, November, 2020.

[12] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, L. Liu, Application of quantum genetic optimization of LVQ neural network in smart city traffic network prediction, *IEEE Access*, Vol. 8, pp. 104555-104564, June, 2020.

[13] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, *Science*, Vol. 220, No. 4598, pp. 671-680, May, 1983.

[14] D. Whitley, A genetic algorithm tutorial, *Statistics and Computing*, Vol. 4, No. 2, pp. 65-85, June, 1994.

[15] J. Li, X. Hou, Y. Wang, J. Huo, Quantum genetic uneven clustering algorithm for wireless sensor networks, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 9, No. 3, pp. 596-605, May, 2018.

[16] S. Das, P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, *IEEE transactions on evolutionary computation*, Vol. 15, No. 1, pp. 4-31, February, 2011.

[17] F. Glover, M. Laguna, Tabu Search, in: D. Z. Du, P. M. Pardalos (Eds.), *Handbook of Combinatorial Optimization*, Springer, 1998, pp. 2093-2229.

[18] H.-C. Huang, S.-C. Chu, J.-S. Pan, C.-Y. Huang, B.-Y. Liao, Tabu search based multi-watermarks embedding algorithm with multiple description coding, *Information Sciences*, Vol. 181, No. 16, pp. 3379-3396, August, 2011.

[19] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, *Swarm intelligence*, Vol. 1, No. 1, pp. 33-57, June, 2007.

[20] K. Wang, Y. Liu, D. Liu, Q. Shi, Improved particle swarm optimization algorithm based on gaussian-grid search method, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 9, No. 4, pp. 1031-1037, July, 2018.

[21] S.-C. Chu, Z.-G. Du, Y.-J. Peng, J.-S. Pan, Fuzzy hierarchical surrogate assists probabilistic particle swarm optimization for expensive high dimensional

problem, *Knowledge-Based Systems*, Vol. 220, Article No. 106939, May, 2021.

[22] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *IEEE computational intelligence magazine*, Vol. 1, No. 4, pp. 28-39, November, 2006.

[23] S.-C. Wang, *Interdisciplinary Computing in Java Programming*, Springer, 2003.

[24] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Advances in Engineering Software*, Vol. 69, pp. 46-61, March, 2014.

[25] A. H. Gandomi, X.-S. Yang, A. H. Alavi, Cuckoo search algorithm: a meta-heuristic approach to solve structural optimization problems, *Engineering with Computers*, Vol. 29, No. 1, pp. 17-35, January, 2013.

[26] J.-S. Pan, P.-C. Song, S.-C. Chu, Y.-J. Peng, Improved compact cuckoo search algorithm applied to location of drone logistics hub, *Mathematics*, Vol. 8, No. 3, Article No. 333, March, 2020.

[27] P.-C. Song, S.-C. Chu, J.-S. Pan, H. Yang, Simplified phasmatodea population evolution algorithm for optimization, *Complex & Intelligent Systems*, June, 2021, https://doi.org/10.1007/s40747-021-00402-0.

[28] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, *Knowledge-based Systems*, Vol. 96, pp. 120-133, March, 2016.

[29] Z. Meng, J.-S. Pan, Quasi-affine transformation evolution with external archive (quatre-ear): an enhanced structure for differential evolution, *Knowledge-Based Systems*, Vol. 155, pp. 35-53, September, 2018.

[30] B. J. Park, H. R. Choi, H. S. Kim, A hybrid genetic algorithm for the job shop scheduling problems, *Computers & Industrial Engineering*, Vol. 45, No. 4, pp. 597-613, December, 2003.

[31] M. F. Tasgetiren, M. Sevkli, Y.-C. Liang, M. M. Yenisey, A particle swarm optimization and differential evolution algorithms for job shop scheduling problem, *International Journal of Operations Research*, Vol. 3, No. 2, pp. 120-135, July, 2006.

[32] P. J. Van Laarhoven, E. H. Aarts, J. K. Lenstra, Job shop scheduling by simulated annealing, *Operations Research*, Vol. 4, No. 1, pp. 113-125, January- February, 1992.

[33] M. Dell'Amico, M. Trubian, Applying tabu search to the job-shop scheduling problem, *Annals of Operations Research*, Vol. 41, No. 3, pp. 231-252, September, 1993.

[34] E. Florez, W. Gomez, L. Bautista, An ant colony optimization algorithm for job shop scheduling problem, September, 2013, https://arxiv.org/abs/1309.5110.

[35] Y. Tamura, M. Yamamoto, I. Suzuki, M. Furukawa, Acquisition of dispatching rules for job-shop scheduling problem by artificial neural networks using PSO, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 17, No. 5, pp. 731-738, September, 2013.

[36] A. Muthiah, R. Rajkumar, A comparison of artificial bee colony algorithm and genetic algorithm to minimize the makespan for job shop scheduling, *Procedia Engineering*, Vol. 97, pp. 1745-1754, 2014.

[37] A. Ponsich, C. A. C. Coello, A hybrid differential evolution- tabu search algorithm for the solution of job-shop scheduling problems, *Applied Soft Computing*, Vol. 13, No. 1, pp. 462-474, January, 2013.

[38] D. Sha, C.-Y. Hsu, A hybrid particle swarm optimization for job shop scheduling problem, *Computers & Industrial Engineering*, Vol. 51, No. 4, pp. 791-808, December, 2006.

[39] K.-L. Huang, C.-J. Liao, Ant colony optimization combined with taboo search for the job shop scheduling problem, *Computers & Operations Research*, Vol. 35, No. 4, pp. 1030-1046, April, 2008.

[40] C.-L. Liu, C.-C. Chang, C.-J. Tseng, Actor-critic deep reinforcement learning for solving job shop scheduling problems, *IEEE Access*, Vol. 8, pp. 71752-71762, April, 2020.

[41] B. Cunha, A. M. Madureira, B. Fonseca, D. Coelho, Deep reinforcement learning as a job shop scheduling solver: A literature review, *International Conference on Hybrid Intelligent Systems*, Porto, Portugal, 2018, pp. 350-359.

[42] Z. Zang, W. Wang, Y. Song, L. Lu, W. Li, Y. Wang, Y. Zhao, Hybrid deep neural network scheduler for job-shop problem based on convolution two-dimensional transformation, *Computational Intelligence and Neuroscience*, Vol. 2019, pp. 1-19, July, 2019.

[43] C. Zhang, W. Song, Z. Cao, J. Zhang, P. S. Tan, C. Xu, Learning to dispatch for job shop scheduling via deep reinforcement learning, October, 2020, https://arxiv.org/abs/2010.12367.

[44] J.-S. Pan, P.-W. Tsai, Y.-B. Liao, Fish migration optimization based on the fishy biology, *2010 Fourth International Conference on Genetic and Evolutionary Computing*, Shenzhen, China, 2010, pp. 783-786.

[45] Q.-W. Chai, S.-C. Chu, J.-S. Pan, W.-M. Zheng, Applying adaptive and self assessment fish migration optimization on localization of wireless sensor network on 3-d terrain, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 11, No. 2, pp. 90-102, June, 2020.

[46] B. Guo, Z. Zhuang, J.-S. Pan, S.-C. Chu, Optimal design and simulation for PID controller using fractional-order fish migration optimization algorithm, *IEEE Access*, Vol. 9, pp. 8808-8819, January, 2021.

[47] J.-S. Pan, P. Hu, S.-C. Chu, Binary fish migration optimization for solving unit commitment, *Energy*, Vol. 226, Article No. 120329, July, 2021.

[48] J.-F. Chang, S.-C. Chu, J. F. Roddick, J.-S. Pan, A parallel particle swarm optimization algorithm with communication strategies, *Journal of Information Science and Engineering*, Vol. 21, No. 4, pp. 809-818, 2005.

[49] Z. Fu, P. Hu, W. Li, J.-S. Pan, S. Chu, Parallel equilibrium optimizer algorithm and its application in capacitated vehicle routing problem, *Intelligent Automation and Soft Computing*, Vol. 27, No. 1, pp. 233-247, 2021.

[50] J.-S. Pan, J. Zhuang, H. Luo, S.-C. Chu, Multi-group flower pollination algorithm based on novel communication strategies, *Journal of Internet Technology*, Vol. 22, No. 2, pp. 257-269, March, 2021.

[51] R. W. Morrison, K. A. De Jong, Measurement of population diversity, *International conference on artificial evolution*, Le Creusot, France, 2001, pp. 31-41.

[52] Q. Xu, L. Wang, N. Wang, X. Hei, L. Zhao, A review of opposition-based learning from 2005 to 2012,

Engineering Applications of Artificial Intelligence, Vol. 29, pp. 1-12, March, 2014.

[53] K. R. Baker, D. Trietsch, *Principles of sequencing and scheduling*, John Wiley & Sons, 2013.

[54] J. C. Bean, Genetic algorithms and random keys for sequencing and optimization, *ORSA journal on computing*, Vol. 6, No. 2, pp. 154-160, May, 1994.

[55] H. Fisher, G. L. Thompson, Probabilistic learning combinations of local job-shop scheduling rules, *Industrial scheduling*, Prentice-Hall, Englewood Cliffs, 1963, pp. 225-251.

[56] S. Lawrence, Resource constrained project scheduling: An experimental investigation of heuristic scheduling techniques (supplement), *Graduate School of Industrial Administration*, Pittsburgh, Pennsylvania, Carnegie-Mellon University, 1984.

[57] J. Wang, X. Gu, W. Liu, A. K. Sangaiah, H.-J. Kim, An Empower Hamilton Loop based Data Collection Algorithm with Mobile Agent for WSNs, *Human-centric Computing and Information Sciences*, Vol. 9, Article No. 18, May, 2019.

[58] S. Qin, C. Sun, Y. Jin, Y. Tan, J. Fieldsend, Large-scale evolutionary multi-objective optimization assisted by directed sampling, *IEEE Transactions on Evolutionary Computation*, Vol. 25, No. 4, pp. 724-738, August, 2021.

[59] B. Zhao, T.-W. Sung, X. Zhang, A quasi-affine transformation artificial bee colony algorithm for global optimization, *Journal of Intelligent & Fuzzy Systems*, Vol. 40, No. 3, pp. 5527-5544, March, 2021.

[60] J. M.-T. Wu, Z. Li, N. Herencsar, B. Vo, J. C.-W. Lin, A graph-based CNN-LSTM stock price prediction algorithm with leading indicators, *Multimedia Systems*, February, 2021, https://doi.org/10.1007/s00530-021-00758-w.

[61] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, M.-E. Wu, A secure authenticated and key exchange scheme for fog computing, *Enterprise Information Systems*, Vol. 15, No. 9, pp. 1200-1215, 2021.

Biographies

Qingyong Yang received the B.S. degree from the Shandong University of Science and Technology, Qingdao, China, in 2020. He is currently pursuing the master's degree with the Shandong University of Science and Technology, Qingdao, China. His recent research interests include swarm intelligence, intelligent scheduling and transportation problems.

Shu-Chuan Chu received the Ph.D. degree from the School of Computer Science, Engineering and Mathematics, Flinders University, Australia, in 2004. She joined Flinders University, in December 2009, after nine years at Cheng Shiu University, Taiwan. She has been a Research Fellow with the College of Science and Engineering, Flinders University, since December 2009. She has been a Research Fellow, with PhD advisor, at the College of Computer Science and Engineering, Shandong University of Science and Technology, since September 2019. Her research interests mainly include swarm intelligence, intelligent computing, and data mining.

Chia-Cheng Hu received the M.S. degree in Engineer Science from National Cheng Kung University in 1995. He received his Ph.D. in department of Computer Science and Information Engineering, National Taiwan University, Taiwan in 2005. He is currently a Professor with the College of Artificial Intelligence, Yango University.

Jimmy Ming-Tai Wu is currently an Assistant Professor at College of Computer Science and Engineering, Shandong University of Science and Technology. He was also an Assistant Professor at School of Computer Science and Technology, Harbin Institute of Technology-Shenzhen, China. His current research work is related to Data Mining, Big Data, Cloud Computing, Artificial Intelligence, Evolutionary Computation, Machine Learning and Deep Learning. He was graduated and earned his Ph.D. degree with major in Computer Science and Engineering from National Sun Yat-sen University, Kaohsiung, Taiwan. He worked in an IC design company in Taiwan as a firmware developer and information technology manager in two years. He also was a research scholar in the Department of Computer Science and Information Engineering in National University of Kaohsiung, Kaohsiung, Taiwan and in the Department of Computer Science, College of Engineering in University of Nevada, Las Vegas.

Jeng-Shyang Pan received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 1996. He is currently the Professor of Shandong University of Science and Technology. He is the IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section and Tainan Chapter Chair of IEEE Signal Processing Society. His current research interest includes the information hiding, artificial intelligence and wireless sensor networks.