



# Moth-flame optimization algorithm based on diversity and mutation strategy

Lei Ma<sup>1,2</sup> · Chao Wang<sup>3,4</sup> · Neng-gang Xie<sup>1,2</sup> · Miao Shi<sup>2,3</sup> · Ye Ye<sup>3</sup> · Lu Wang<sup>3</sup>

Accepted: 13 November 2020 / Published online: 17 January 2021  
© Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

In this work, an improved moth-flame optimization algorithm is proposed to alleviate the problems of premature convergence and convergence to local minima. From the perspective of diversity, an inertia weight of diversity feedback control is introduced in the moth-flame optimization to balance the algorithm's exploitation and global search abilities. Furthermore, a small probability mutation after the position update stage is added to improve the optimization performance. The performance of the proposed algorithm is extensively evaluated on a suite of CEC'2014 series benchmark functions and four constrained engineering optimization problems. The results of the proposed algorithm are compared with the ones of other improved algorithms presented in literatures. It is observed that the proposed method has a superior performance to improve the convergence ability of the algorithm. In addition, the proposed algorithm assists in escaping the local minima.

**Keywords** Moth-flame optimization · Diversity · Inertia weight · Mutation

## 1 Introduction

In recent years, many applications constitute non-convex optimizations problems. These non-convex problems are very complex in nature. The traditional gradient algorithms are unable to achieve ideal results [1]. Some population-based random search algorithms inspired by nature have been proposed to solve these problems. These proposed solutions are considered effectively. Since Holland proposed the Genetic Algorithm (GA) [2], the meta-heuristic

random search algorithm has led to a different approach in solutions. At present, this meta-heuristic algorithm based on natural phenomena is mainly divided into three categories. These include Genetic Algorithm(GA) [2], Differential Evolution algorithm(DE) [3], and Cultural Algorithm(CA)[4]. The first group of solutions are a family of intelligent optimization algorithms inspired by the laws of natural evolution. The second set of solutions is inspired by the group behavior of creatures like the Particle Swarm Optimization(PSO) [5] that mimics the foraging behavior of birds, the Gray Wolf Optimization(GWO) [6] that mimics the hunting behavior of gray wolves. In addition, there are Salp Swarm Algorithm(SSA) [7], Ant Colony Optimization(ACO) [8], Bat Algorithm(BA) [9], Artificial Bee Colony algorithm (ABC) [10], and Firefly Algorithm(FA) [11]. The third is inspired by natural physical phenomena, such as Gravitational Search Algorithm(GSA) [12] based on the interaction force between particles, Water Cycle Algorithm (WCA) [13], Simulated Annealing algorithm(SA) [14], Ion Motion Optimization (IMO)[15], and Multiverse Optimization(MVO) [16]. Until now, intelligent computing algorithms created by taking inspiration from nature are still incessantly emerging. Kaveh et al developed the Water Strider Algorithm(WSA) [17] based on the social life behavior of water strider. Hayylalam et al. proposed a Black Widow Optimization algorithm(BWO) [18] inspired by the breeding behavior of black widow spiders. Faramarzi

✉ Chao Wang  
cw2013@ahut.edu.cn

✉ Lu Wang  
wanglu@ahut.edu.cn

<sup>1</sup> Department of Management Science and Engineering, Anhui University of Technology, Anhui, Ma'anshan 243002, People's Republic of China

<sup>2</sup> Anhui Province Key Laboratory of Multidisciplinary Management and Control of Complex Systems, Anhui, Ma'anshan 243002, People's Republic of China

<sup>3</sup> Department of Mechanical Engineering, Anhui University of Technology, Anhui, Ma'anshan 243002, People's Republic of China

<sup>4</sup> Department of Engineering Mechanics, Hohai University, Nanjing 211100, People's Republic of China

et al. respectively developed the Marine Predators Algorithm(MPA) [19] and the Equilibrium Optimizer(EO) [20] based on biological and physical phenomena. In addition, MPA was inspired by foraging strategies among organisms in marine ecosystems. It uses three different movement patterns to simulate the interaction between predator and prey. While the EO is a physics-based optimization algorithm, which originates from the control volume mass balance model. Due to their excellent convergence properties, there have been some research reports on their practical applications.

Metaheuristics are usually considered a promising alternative for the complex optimization problems. For example, Faris et al. [21] combines GA with Random Weight Network (RWN) and propose a hybrid spam detection system, Auto-GA-RWN. The experimental results on three email corpora show that this method achieves higher recognition and detection rate. For the Feature Selection(FS) problem, Faris et al. [22] designed an enhanced binary SSA optimizer by selecting the best transfer function. The comparison with several advanced methods proves that the proposed method has better accuracy. Taradeh et al. [23] applies GSA to solve the FS problem, and for the first time propose the use of evolutionary crossover and mutation operators to improve the exploration and exploitation capabilities of GSA. Extensive research shows that this method is significantly better than other similar methods when dealing with FS tasks.

The Moth-Flame Optimization algorithm(MFO)[24] is a new intelligent optimization algorithm proposed by Mirjalili in 2015. The algorithm is inspired by a peculiar navigation mechanism of the moth, i.e., transverse orientation. The moths are able to keep flying straight at night because they use the moon as a reference and always fly at a fixed angle to the moon. However, this mechanism is only effective when dealing with long-range reference objects. When moths are disturbed by artificial light sources, the same positioning method will only mislead them to fly around the light sources, resulting in fire-fighting behavior. Inspired by this notion, Mirjalili simulated the night flight principle of the moth and proposed this optimization method based on a mathematical model of spiral motion.

The MFO algorithm has attracted a wide attention due to its simple structure, robustness, and easy implementation[15]. This algorithm has been applied in many fields. Yildiz et al. [25] used the MFO algorithm to determine the optimal machining parameters in the manufacturing process and solved the milling optimization problem in the manufacturing process. Youstri et al. [26] used the MFO algorithm to extract the diode parameters in the polycrystalline silicon solar module. The experiment proves that MFO has a better performance when solving this type of problem. Lei et al. [27] proposed a moth-flame optimization based protein complex prediction algorithm, and applied it to solve the

problem of protein complex recognition in Protein-Protein Interaction(PPI) networks. The experimental results demonstrate that this method has certain advantages over other similar methods. However, MFO has insufficient global convergence ability in the optimization process and is prone to getting stuck in the local minima. To address the aforementioned problems, Xu et al. [28] combined MFO with a series of mutations to help the algorithm escape the local optimal solution and avoid premature convergence when dealing with high-dimensional multi-modal problems. Chen et al. [29] combined the distributed computing spark platform with the MFO. The experiments of this combined setup reveal that the algorithm effectively improves the classification performance of feature selection. Wang et al. [30] used the diploid structure of replication coding to update the moth population, thereby improving the global search ability of the algorithm. Emary et al. [31] proposed an improved moth-flame optimization algorithm based on search and development speed control by changing the path of moth spiral motion. The study shows that the improved MFO can establish a better balance between exploration and exploitation when compared with other methods. Xu et al. [32] introduced the Cultural Learning(CL) mechanism and Gaussian Mutation(GM) operator into the MFO. The testing of the benchmark functions shows that it improves the insufficient search ability of the algorithm and easily mitigates the local optimal issue, thus resulting in the improved solution quality and reliability. Huang et al. [33] guided moths to constantly seek better values by establishing a ring network between flames. This technique improves the exploration ability of the algorithm and provides good results in practical engineering applications. Sapre et al. [34] combined MFO with opposite learning, Cauchy mutation and evolutionary boundary constraint handling. The simulation results show that these methods can significantly enhance the exploration and exploitation capabilities of MFO. Li et al. [35] opined that chaotic maps are one of the best ways to improve the performance of meta-heuristic algorithms. The authors use chaotic maps to improve the MFO algorithm by proposing a chaos enhanced MFO algorithm. The presented method verifies the effectiveness of the method on the basis of benchmark function tests and practical engineering applications. Zhang et al. [36] proposed an improved MFO algorithm that combines Broyin-Fletcher-Goldfarb-Shanno method(BFGS) and Orthogonal Learning (OL). The results show that this method effectively alleviates the shortcomings of stagnant MFO evolution to improve convergence. To mitigate the stagnation of local optimization of MFO, Pelusi et al. [1] divided the optimization process of MFO into exploration, hybrid exploration and exploitation, and exploitation to ensure that MFO is able to establish a better balance between exploration and exploitation. Li et al. [37] proposed the use of differential evolution flame

generation and dynamic flame guidance to guide the population evolution and improve the global search capability of MFO. In addition, Khalilpourazari et al. [38] also introduced the water cycle algorithm into MFO, and proposed a hybrid meta-heuristic algorithm for water cycle moth flame optimization to enhance the exploitation capabilities of MFO. The results show that this method effectively assists the MFO to get rid of local minima.

The aforementioned methods tend to enhance the performance of the solution. However, sometimes the consideration of diversity is insufficient. For instance, Xu et al. [28] improved the algorithm using the notion of mutating search agents, while traditional mutation operators such as Gaussian mutation perform non-directional mutations. These mutations are unable to increase the diversity of the population. The high-frequency operation greatly increases the complexity of the algorithm. Therefore, the improvement of diversity is undoubtedly an important aspect to enhance the optimization performance of the algorithm. When the population diversity is poor and the solution has fallen into local minima, the calculation results do not exhibit any substantial improvements even if the algorithm continues. Jacques et al. [39] proposed a particle swarm optimization algorithm using diversity measures to control the population. The experiment proved that it is significantly better than other contrast algorithms in handling multimodal optimization problems. Wang et al. [40] improved the search diversity of the ant colony algorithm by using both positive and negative feedback at the same time. The experimental results show that the performance of the improved algorithm is significantly improved in dealing with combinatorial optimization problems. In order to enhance the search capability of differential evolution algorithm, Yang et al. [41] proposed an Auto-Enhanced Population Diversity (AEPD) mechanism to improve the algorithm. The simulation results demonstrate that the modified algorithm has better performance than other similar algorithms.

To improve the optimization performance of the algorithm, a diversity and mutation strategy based moth-flame optimization algorithm (DMMFO) is proposed. The main contributions of this paper are as follows:

- In the exploration phase of the moth-flame optimization, a diversity weight component is integrated to alleviate the premature convergence caused by the rapid evolution of the algorithm. This component can continuously switch the population between divergence and contraction within a limited iteration time to maintain the diversity of the population, and balance the exploration and exploitation capabilities of the algorithm.
- The dimensional mutation operator is employed to improve the algorithm's ability to update the optimal solution. It allows the algorithm to explore unknown

area in space in a reasonable way, so that the population can cover a wider range of feasible solutions, and avoid stagnation in evolution.

- The optimal control parameters of the proposed method are obtained through the sensitivity test of the control parameters.

At present, there are few studies on embedding diversity and mutation strategies in MFO algorithms. This analysis forms the basis of this work. The rest of the manuscript is organized as follows. Section 2 presents the main structure of the MFO. Section 3 discusses the proposed DMMFO which combines diversity and mutation based on MFO. The experiments and results are analyzed in detail to validate the validity in Section 4. Section 5 investigates the effectiveness of the proposed algorithm in solving four constrained engineering optimization problems. Section 6 contains the conclusions of the paper.

## 2 Moth-flame optimization

This section presents the mathematical model of the MFO as a simple and efficient optimization technique.

### 2.1 Population initialization and storage mechanism

The MFO algorithm uses the following relation to initialize the population position in the search space.

$$M = R(n, d) \times (ub - lb) + lb \quad (1)$$

where,  $ub$  and  $lb$  represent the upper and lower bounds of the searching space, respectively.  $n$  represents the population size,  $d$  represents the number of variables, i.e., the number of dimensions,  $R$  represents the random numbers that generate random numbers with a uniform distribution between (0, 1). The initial position corresponding to each moth is stored in the matrix  $M$ .

$$M = \begin{bmatrix} m_{1,1} & \cdots & m_{1,d} \\ \vdots & \ddots & \vdots \\ m_{n,1} & \cdots & m_{n,d} \end{bmatrix} \quad (2)$$

When the positions of all moths in the search space are obtained, the fitness value of each moth is calculated by using fitness function, and the calculated result is stored in fitness matrix  $OM$ .

$$OM = [om_1 \quad om_2 \quad \cdots \quad om_n]^T \quad (3)$$

It is notable that another variable is introduced here, i.e., flame. The moths and flames are actually candidate solutions. The flame can be seen as the flag of the moth in the search space. The moth is responsible for moving in the

search space and finding the best position, and the flame represents the best position that has been found so far. The position of the flames and the corresponding fitness values are stored in the matrices  $F$  and  $OF$ , respectively.

$$F = \begin{bmatrix} f_{1,1} & \cdots & f_{1,d} \\ \vdots & \ddots & \vdots \\ f_{n,1} & \cdots & f_{n,d} \end{bmatrix} \tag{4}$$

$$OF = [of_1 \ of_2 \ \cdots \ of_n]^T \tag{5}$$

where  $n$  represents the number of moths.

### 2.2 Position update mechanism

This mechanism mimics the fire-fighting behavior of moths in real life. According to the sorted flames, the moths fly around the flames in a spiral motion. This phenomenon is described as

$$M_i = S(M_i, F_j) \tag{6}$$

$$S(M_i, F_j) = D_i \times e^{bt} \times \cos(2\pi t) + F_j \tag{7}$$

$$D_i = |F_j - M_i| \tag{8}$$

where,  $M_i$  represents the updated position of the  $i$ -th moth,  $S$  represents a logarithmic spiral curve,  $F_j$  represents the  $j$ th flame,  $b$  represents a constant,  $t$  represents a random number between  $[a, 1]$ , where  $a$  decreases linearly from  $-1$  to  $-2$  as the number of iterations increases, and  $D_i$  is the distance from the  $i$ -th moth to the  $j$ -th flame.

Please note that to improve the ability to develop and explore the optimal solution and improve the convergence rate of the algorithm in the later stages, MFO adopts an adaptive reduction mechanism for the number of flames, as described by the following relation

$$Num - flame = round \left( N - T \times \frac{N - 1}{T_{max}} \right) \tag{9}$$

where,  $N$  represents the maximum number of flames,  $T$  represents the current number of iterations, and  $T_{max}$  represents the maximum number of iterations.

## 3 The proposed methods

This section elaborates on the two important components integrated in the MFO, and explains the principle of action. Finally, the computational complexity of the proposed method is analyzed.

### 3.1 Inertia weight

The poor diversity is often considered as the main reason for premature convergence of the MFO algorithm. Thus,

its richness is of great significance for the process of population optimization. When the population has a high diversity, it searches a larger range in a limited area. In other words, the high population diversity directly represents that a large area in the search space has been searched. In contrast, the rapid decrease of diversity in the evolutionary process results in a smaller search space for the algorithm.

In the MFO, the position update of moths is divided into two categories. When the number of moths is less than or equal to the number of flames, each moth flies around the corresponding flame and updates its position. Due to the presence of an adaptive flame reduction mechanism, when the number of moths is greater than the number of flames, all moths update their positions based on only one flame. In this work, the inertia weight of a diversity feedback control [42] is added to the first case by manipulating (7). The updated relation of the moths is

$$S(M_i, F_j) = W \times D_i \times e^{bt} \times \cos(2\pi t) + F_j \tag{10}$$

Please note that a smaller value of  $W$  means that all groups in the search space are more likely to converge to the optimal advantage. The relation for calculating the inertia weight is

$$W = e^{-C} \times \left( 1 - \frac{T}{T_{max}} \right) \tag{11}$$

$$C = \frac{1}{N \times L} \times \sum_{i=1}^N \times \sqrt{\sum_{j=1}^{dim} (M_{i,j}^T - \bar{M}_j^T)^2} \tag{12}$$

where,  $N$  represents the population number,  $dim$  represents the dimension, and  $M_{i,j}^T$  represents the value of the  $i$ -th moth in the  $j$  dimension at the  $T$ -th iteration, and  $\bar{M}_j^T$  represents the mean value of the moth in the  $j$  dimension at the  $T$ -th iteration.

The symbol  $L$  is defined as the maximum diagonal distance of the search space and is obtained as

$$L = \sqrt{\sum_j^{dim} (ub_j - lb_j)^2} \tag{13}$$

where,  $ub_j$  and  $lb_j$  represent the upper bound and lower bound of the  $j$ -th dimension in the search space, respectively.

The value of  $C$  is determined by the difference in positions of the moths in different dimensions. In (10), the smaller  $W$  indicates that the whole population continuously evolves toward the optimal value. At the beginning of the iteration, as the value of  $T$  is small, the value of  $C$  has a greater impact on (11). When the overall trend of convergence reaches to a certain point,  $W$  increases rapidly, making the group approach the optimal point in a stable form. The current convergence result will reduce  $W$  in the

next iteration, so that the moths conduct a more detailed search within a smaller flight radius near a better solution.

The effect of diversity on population evolution is graphically presented in Fig. 1. It is evident from Fig. 1a that in the early stage of the iterations, due to the influence of weights, the population in the search domain is relatively scattered. For some complex functions, the algorithm easily falls into the local minima trap during the evolution process. However, increasing the diversity of the population reduces this risk as presented in (Fig. 1b). Even though most solutions fall into local optima, there are still search agents that jump out of the trap and approach global optima due to the expansion of search space. In the later stages of iteration,  $W$  decreases due to the gradual approach of  $1 - \frac{T}{T_{max}}$  to zero. At this time, more moths will find the best flame mainly based on (10). Finally, the population converges when a termination criterion is met as presented in (Fig. 1c).

### 3.2 Position variation

In the MFO algorithm, the pros and cons of flame fitness are all calculated and ranked by the moth's position calculation. The sorted moth can only be updated according to the corresponding flame. If the first moth falls into the local minima, the algorithm faces a hard time to exit this minima. Therefore, at the end of each position update, (1) with a small probability is used to initialize the positions of the three moths with the best fitness values in the random dimensions. This is presented in Fig. 2.

This not only retains some of the original attributes of the moth, but also changes its position in a certain direction, causing the search starting point to shift. In other words, the flame corresponding to the first moth is not necessarily the brightest. There may be brighter flames around it, but the distance dimmed their glow. The random variation is similar to the wind in nature. It has the probability to blow from different directions to change the position of the moths, thereby helping the moths to find a better solution. Because the amplitude and frequency of the mutations are low, the moth do not lose the search for the original solution significantly. The probability calculation procedure is (*if rand* < 0.1). The random dimension selection procedure is

$$data = rand(1, dim) \quad (14)$$

$$index = find(data < 0.3) \quad (15)$$

### 3.3 Computational complexity analysis

The time computational complexity of the DMMFO mainly depends on the maximum number of iterations ( $Q$ ),

initialization, sorting, position update, inertia weight and mutation mechanism. The time complexity during the initialization process is  $O(n \times d)$ , where  $n$  is the number of moths. Since the fast method is adopted, the complexity of each iteration sort is  $O(Q \times n^2)$  in the worst case. Considering the population with  $n$  moths and  $d$ -dimensional optimization problem, the positions are updated and the diversity weight requires  $O(Q \times n \times d)$ . Since it is a probabilistic mutation and only the best three moths are selected for mutation, the worst case requires  $O(Q \times 3 \times d)$ . After the data is sorted out, the upper bound of DMMFO computational complexity is:  $O(Q(n^2 + nd))$ . The pseudocode of the proposed DMMFO is as presented below.

---

#### Improved moth-flame optimization (DMMFO).

---

Initialize the positions of all moths

**While** FEs<sub>i</sub>=Max\_FEs

    Update the Num\_flame using (9)

    OM=Fitness Function(M)

**if** iteration==1

        F=sort(M)

        OF=sort(OM)

**else**

        F=sort( $M_{T-1}$ ,  $M_T$ )

        OF=sort( $OM_{T-1}$ ,  $OM_T$ )

**end**

**for** i=1:n

**for** j=1:d

            Update a and t

            Calculate  $D_i$  using (8) with respect to the corresponding moth

            Update  $M_{(i,j)}$  using (7) and (10) with respect to the corresponding moth

**end**

**end**

    According the Section 3.2 to mutate the position of the moths

    FEs=FEs+N

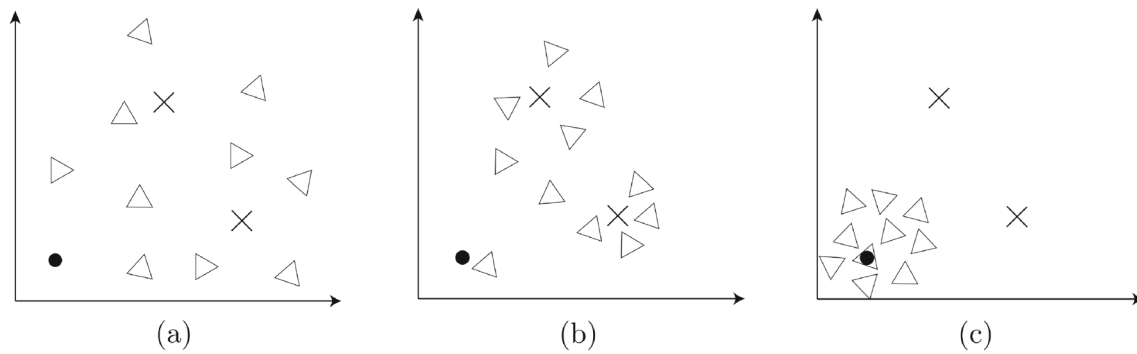
**end**

---

## 4 Simulation experiment

The CEC'2014 benchmark functions [43] were selected to test the performance of the proposed DMMFO method (see Table 1) and were compared with the basic MFO, PSO, and other improved algorithms [28]. CEC'2014 includes the four parts, namely 3 unimodal functions (1-3), 13 simple multimode functions (4-16), 6 hybrid function (17-22), and 8 composition functions (23-30).





**Fig. 1** Diversity diagram, moths( $\Delta$ ), global optimal solution( $\bullet$ ), local optimal solution( $\times$ )

#### 4.1 Discussion of control parameters

The parameters of meta-heuristic algorithms often have a greater impact on the convergence performance of the algorithm. The main control parameter of MFO is  $b$  in (7). This parameter defines the shape of the logarithmic spiral curve. To explore the evolution caused by the different values of  $b$  and analyze the effect of  $b$  on convergence, the algorithm runs in six different types of functions with the following fixed parameters: population number  $n=50$ , maximum fitness evaluation times  $MaxFes = 10000 \times dim$ . Tables 2, 3, 4 show the test results in three different dimensions under different values of  $b$ .

As presented in Table 2, the optimization results are obtained when  $b=1$  with the best results among the five functions. But the situation has changed when the dimension is 30, the best results are only achieved when  $b=1$  in the F7 and F30 functions. Except for  $b=1.5$ , the other values have the same effect. For 50 dimensions, when  $b=1$ , the best optimization effect will be achieved, followed by 0.2. Therefore, this paper sets  $b$  to 1 in order to ensure best performance.

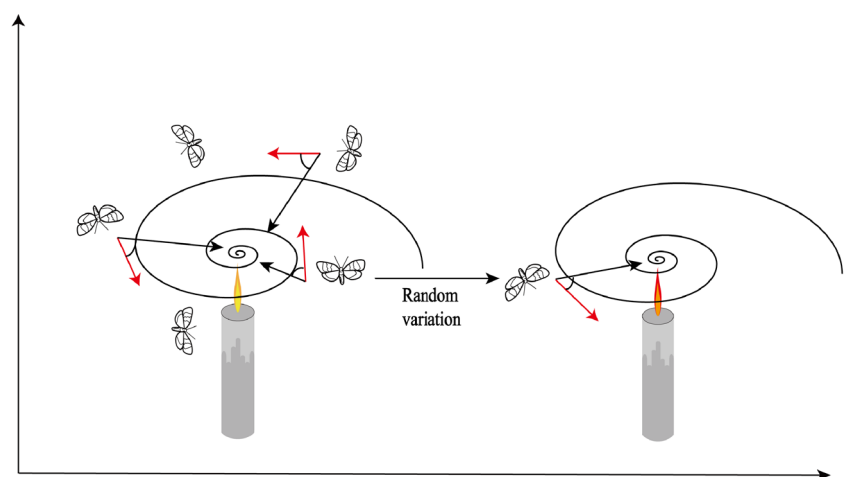
For the purpose of the fairness in comparisons, all algorithms are executed in the same environment in this

paper. The software used for executing all the algorithms is Matlab2018b. The OS used of the PC used for performing simulations is Windows10. The CPU is Intel Core i7 3.0Hz. The experimental parameters are the number of population equal to 50, and the maximum fitness evaluation times are  $MaxFes = 10000 \times dim$ . Each algorithm runs independently 50 times and the execution results of the algorithm in 10 dimensions, 30 dimensions, and 50 dimensions are collected. The collected results are then subject to statistical analysis. The control parameters of all algorithms and their variant versions are consistent with the corresponding literatures.

#### 4.2 Evaluation standard

To evaluate the performance of the algorithm, this paper introduces the commonly used evaluation indexes, namely, average convergence value and standard deviation. The mean value reflects the convergence accuracy of the algorithm, while the standard deviation shows the stability of the algorithm. So as to further compare the algorithms more comprehensively, wilcoxon symbol rank test is also used in this work [44]. This method uses a limited number of samples to evaluate the pros and cons of the algorithm. First,

**Fig. 2** Position variation diagram



**Table 1** Descriptions of the test functions

NO	Functions	Optimum
1	Rotated High Conditioned Elliptic Function	100
2	Rotated Bent Cigar Function	200
3	Rotated Discus Function	300
4	Shifted and Rotated Rosenbrock's Function	400
5	Shifted and Rotated Ackley's Function	500
6	Shifted and Rotated Weierstrass Function	600
7	Shifted and Rotated Griewank's Function	700
8	Shifted Rastrigin's Function	800
9	Shifted and Rotated Rastrigin's Function	900
10	Shifted Schwefel's Function	1000
11	Shifted and Rotated Schwefel's Function	1100
12	Shifted and Rotated Katsuura Function	1200
13	Shifted and Rotated HappyCat Function	1300
14	Shifted and Rotated HGBat Function	1400
15	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	1500
16	Shifted and Rotated Expanded Scaffer's F6 Function	1600
17	Hybrid Function 1 (N=3)	1700
18	Hybrid Function 2 (N=3)	1800
19	Hybrid Function 3 (N=4)	1900
20	Hybrid Function 4 (N=4)	2000
21	Hybrid Function 5 (N=5)	2100
22	Hybrid Function 6 (N=5)	2200
23	Composition Function 1 (N=5)	2300
24	Composition Function 2 (N=3)	2400
25	Composition Function 3 (N=3)	2500
26	Composition Function 4 (N=5)	2600
27	Composition Function 5 (N=5)	2700
28	Composition Function 5 (N=5)	2800
29	Composition Function 5 (N=5)	2900
30	Composition Function 8 (N=3)	3000

Search Range: [-100,100]

the final convergence result indicators of the two algorithms in the same test function are paired and subtracted, and the differences are accumulated and recorded as  $R+$  and  $R-$ , respectively. It is generally believed that the former algorithm is better when the number of  $R+$  is greater

than the number of  $R-$ , and vice versa. Generally, the significance level is set to 0.05 in this paper. Please note that, symbols “+”, “-” and “=” indicate that the algorithm proposed in this work is superior to, inferior to, and equal to the algorithm to which it is being compared.  $gm$  represents the total difference between the symbols “+” and “-”.

### 4.3 Unimodal test functions result analysis

Tables 5, 6, 7 summarize the performance indicators of several algorithms on the unimodal test functions in different dimensions. Except for F3, the proposed DMMFO is dominant in all evaluation indicators. These functions are unimodal and non-separable plate shape problems. So even though there is only a single minimum, it is difficult for the algorithm to converge on this minimum. Although the DMMFO is less accurate than LGCMFO in the F3 of each dimension, it is better than CMFO. The convergence graphs of average best-so-far solutions are shown in Figs. 3, 4, 5, 6, 7 and 8. Note that, the proposed DMMFO does not converge prematurely as compared to MFO and CMFO in the test function F1 and nor does the echelon form appear. In addition, the convergence speed of the proposed DMMFO is faster than LGCMFO. This indicates that DMMFO has the ability to continuously converge when dealing with this type of problem. The underlying reason is that our added diversity weight raises the search space of the population and enables the algorithm to find a better solution in a limited range. Moreover, it also guides the population to evolve towards the global optimum, thereby improving the convergence speed of the algorithm. This shows that the exploitation ability of the proposed method on the unimodal problem has been greatly improved.

### 4.4 Simple multimodal text functions result analysis

As presented in Tables 8, 9, 10, the DMMFO has a great advantage in the calculation of different dimensions. The main reason is that the diversity weight and mutation strategy in DMMFO establishes a balance between exploration and exploitation. For F5 and F12 functions, the gaps of several algorithms are very small. The DMMFO still has a slight lead after several runs. For function F7, the optimization performance of the DMMFO displays a qualitative leap as the number of dimension increases. From Figs. 9, 10, 11, 12, 13 and 14, the convergence graph of the simple multimodal test function has a step-like or cliff-like decline. This suggests that the algorithm has a relatively prominent exploration ability. Although the functions F7, F14, and F15 have experienced several local optimal solutions, they always evolve continuously. This shows that the diversity weight in this work effectively improves the

**Table 2** 10-dimension function test

Function		10dim				
		<i>b</i> =0.2	<i>b</i> =0.5	<i>b</i> =1	<i>b</i> =1.5	<i>b</i> =2
F1	Mean	1.38E+05	2.05E+05	<b>1.18E+05</b>	1.30E+05	1.33E+05
	Std	2.63E+05	3.63E+05	<b>7.04E+04</b>	8.21E+04	7.13E+04
F2	Mean	6.44E+03	6.13E+03	<b>5.76E+03</b>	5.94E+03	7.16E+03
	Std	4.12E+03	<b>4.02E+03</b>	4.52E+03	4.56E+03	4.27E+03
F7	Mean	1.64E-01	1.63E-01	<b>1.55E-01</b>	2.15E-01	1.70E-01
	Std	<b>9.74E-02</b>	1.21E-01	1.03E-01	1.20E+00	9.19E-02
F8	Mean	3.05E+00	2.93E+00	2.43E+00	1.87E+00	<b>1.45E+00</b>
	Std	2.25E+00	1.70E+00	2.19E+00	<b>1.31E+00</b>	1.52E+00
F18	Mean	9.68E+03	9.13E+03	<b>8.98E+03</b>	1.31E+04	1.33E+04
	Std	1.14E+04	1.03E+04	<b>8.76E+03</b>	1.38E+04	1.31E+04
F30	Mean	7.12E+02	6.96E+02	<b>5.72E+02</b>	5.97E+02	5.85E+02
	Std	1.99E+02	2.11E+02	<b>1.04E+02</b>	1.72E+02	1.10E+02

The bold entries indicated that the value is optimal in the same row

**Table 3** 30-dimension function test

Function		10dim				
		<i>b</i> =0.2	<i>b</i> =0.5	<i>b</i> =1	<i>b</i> =1.5	<i>b</i> =2
F1	Mean	<b>1.56E+06</b>	1.59E+06	1.68E+06	1.98E+06	2.41E+06
	Std	8.60E+05	1.12E+06	<b>8.11E+05</b>	1.08E+06	1.25E+06
F2	Mean	<b>1.23E+04</b>	1.46E+04	1.40E+04	1.58E+04	1.66E+04
	Std	<b>1.42E+04</b>	1.46E+04	1.58E+04	1.58E+04	1.62E+04
F7	Mean	1.34E-02	3.63E-02	<b>1.31E-02</b>	1.49E-02	1.54E-02
	Std	1.66E-02	1.08E-01	<b>1.62E-02</b>	1.98E-02	1.64E-02
F8	Mean	5.63E+01	4.66E+01	3.05E+01	1.82E+01	<b>1.02E+01</b>
	Std	1.31E+01	1.28E+01	8.79E+00	8.39E+00	<b>5.11E+00</b>
F18	Mean	4.44E+03	<b>4.38E+03</b>	7.58E+03	7.90E+03	1.36E+04
	Std	5.18E+03	<b>4.70E+03</b>	8.93E+03	7.70E+03	9.69E+03
F30	Mean	3.71E+03	4.48E+03	<b>2.20E+03</b>	3.88E+03	3.93E+03
	Std	1.23E+03	4.40E+03	<b>5.98E+02</b>	1.23E+03	8.96E+02

The bold entries indicated that the value is optimal in the same row

**Table 4** 50-dimension function test

Function		10dim				
		<i>b</i> =0.2	<i>b</i> =0.5	<i>b</i> =1	<i>b</i> =1.5	<i>b</i> =2
F1	Mean	2.89E+06	2.87E+06	<b>2.81E+06</b>	3.80E+06	5.01E+06
	Std	1.41E+06	<b>7.63E+05</b>	1.01E+06	1.32E+06	1.71E+06
F2	Mean	<b>8.80E+03</b>	1.40E+04	1.35E+04	1.34E+04	1.72E+04
	Std	<b>9.59E+03</b>	1.17E+04	1.19E+04	1.21E+04	1.15E+04
F7	Mean	1.15E-02	1.02E-02	<b>8.76E-03</b>	1.71E-02	1.63E-02
	Std	2.26E-02	1.42E-02	<b>1.39E-02</b>	3.81E-02	3.14E-02
F8	Mean	1.70E+02	1.38E+02	7.57E+01	4.52E+01	<b>2.99E+01</b>
	Std	3.83E+01	2.48E+01	2.37E+01	1.44E+01	<b>1.36E+01</b>
F18	Mean	<b>2.11E+03</b>	2.92E+03	3.18E+03	4.44E+03	4.61E+03
	Std	<b>1.78E+03</b>	2.13E+03	1.96E+03	2.48E+03	2.43E+03
F30	Mean	1.23E+04	1.29E+04	<b>1.17E+04</b>	1.26E+04	1.35E+04
	Std	2.05E+03	2.54E+03	<b>2.54E+03</b>	2.86E+03	2.57E+03

The bold entries indicated that the value is optimal in the same row



**Table 5** 10-dimension unimodal function test result statistics

Function		This paper	MFO	CMFO	LGCMFO	PSO
F1	Mean	<b>1.184E+05</b>	2.305E+06	1.470E+06	3.679E+05	5.295E+06
	Std	<b>7.738E+04</b>	6.994E+06	4.919E+06	5.194E+05	1.933E+07
	Wilcoxon		+	+	+	+
F2	Mean	<b>5.752E+03</b>	7.655E+03	7.049E+03	6.042E+03	2.251E+05
	Std	4.268E+03	<b>4.035E+03</b>	4.396E+03	4.571E+03	2.486E+05
	Wilcoxon		+	+	+	+
F3	Mean	7.154E+03	1.641E+04	1.435E+04	8.673E+02	<b>7.280E+02</b>
	Std	7.408E+03	1.217E+04	1.147E+04	<b>2.845E+02</b>	1.756E+03
	Wilcoxon		+	+	-	-
+/-/=gm		10/2/0/8				

The bold entries indicated that the value is optimal in the same row

**Table 6** 30-dimension unimodal function test result statistics

Function		This paper	MFO	CMFO	LGCMFO	PSO
F1	Mean	<b>1.735E+06</b>	1.021E+08	6.850E+07	4.670E+06	4.179E+07
	Std	<b>9.457E+05</b>	9.651E+07	7.050E+07	2.466E+06	4.221E+07
	Wilcoxon		+	+	+	+
F2	Mean	<b>1.406E+04</b>	9.378E+09	1.093E+10	1.746E+04	2.727E+09
	Std	1.512E+04	6.088E+09	7.041E+09	<b>1.374E+04</b>	2.261E+09
	Wilcoxon		+	+	+	+
F3	Mean	1.758E+04	8.936E+04	8.821E+04	<b>7.622E+03</b>	1.556E+04
	Std	1.467E+04	5.349E+04	4.138E+04	<b>3.550E+03</b>	1.391E+04
	Wilcoxon		+	+	-	+
+/-/=gm		11/1/0/10				

The bold entries indicated that the value is optimal in the same row

**Table 7** 50-dimension unimodal function test result statistics

Function		This paper	MFO	CMFO	LGCMFO	PSO
F1	Mean	<b>2.856E+06</b>	3.176E+08	2.836E+08	9.183E+06	1.017E+08
	Std	<b>1.199E+06</b>	2.191E+08	2.491E+08	3.862E+06	6.618E+07
	Wilcoxon		+	+	+	+
F2	Mean	<b>1.358E+04</b>	3.972E+10	4.204E+10	2.038E+04	9.584E+09
	Std	<b>1.218E+04</b>	1.607E+10	1.392E+10	6.249E+04	7.067E+09
	Wilcoxon		+	+	+	+
F3	Mean	2.724E+04	1.708E+05	1.700E+05	<b>1.769E+04</b>	2.503E+04
	Std	1.349E+04	7.729E+04	7.556E+04	<b>6.353E+03</b>	1.689E+04
	Wilcoxon		+	+	-	-
+/-/=gm		10/2/0/8				

The bold entries indicated that the value is optimal in the same row

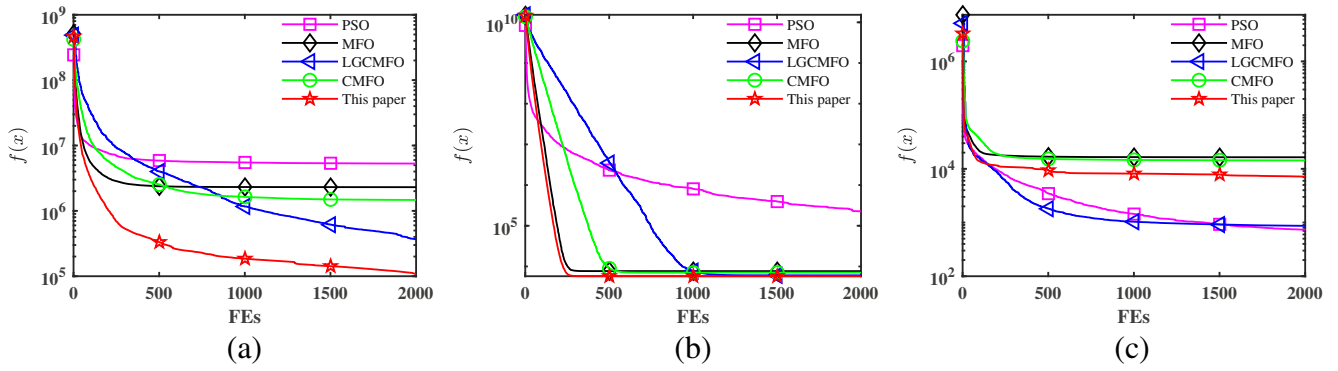


Fig. 3 10-dimensional unimodal function convergence curve

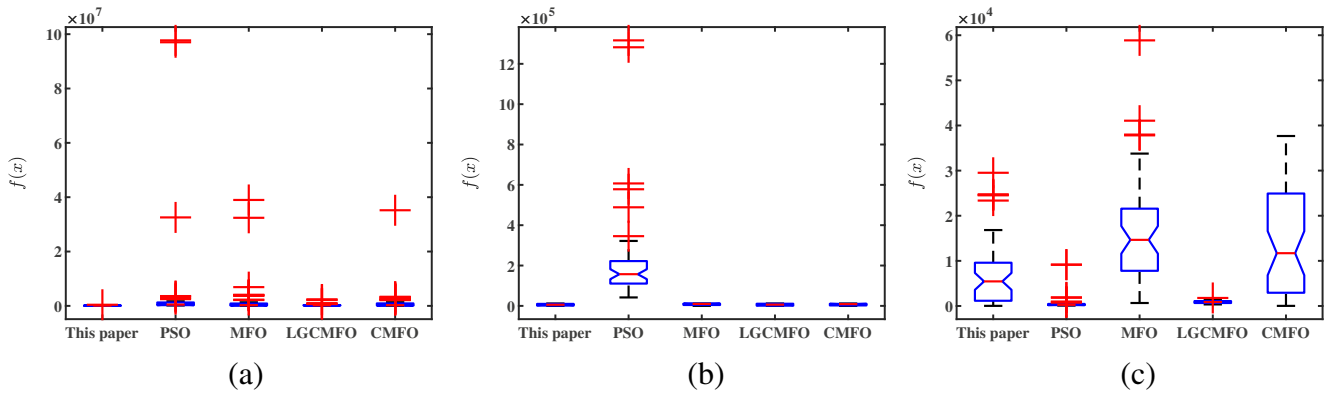


Fig. 4 10-dimensional unimodal function box plot

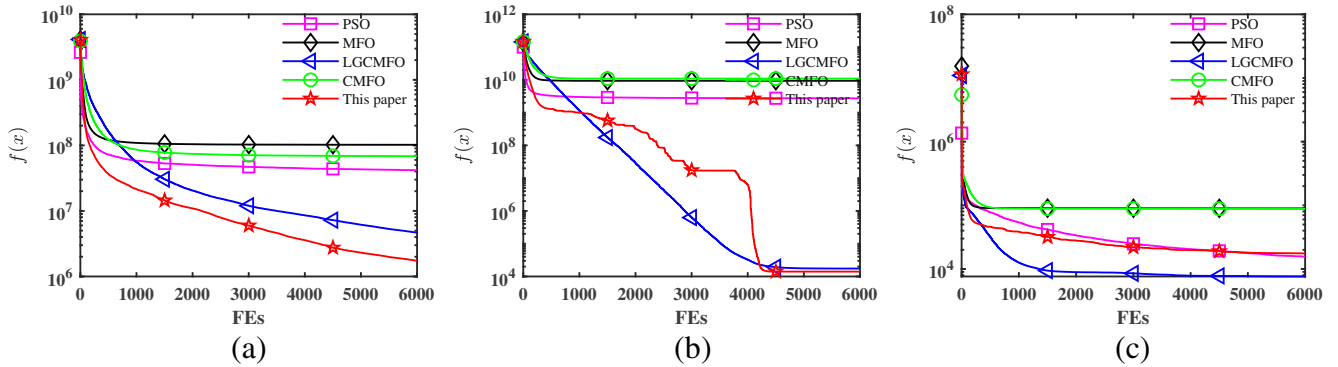


Fig. 5 30-dimensional unimodal function convergence curve

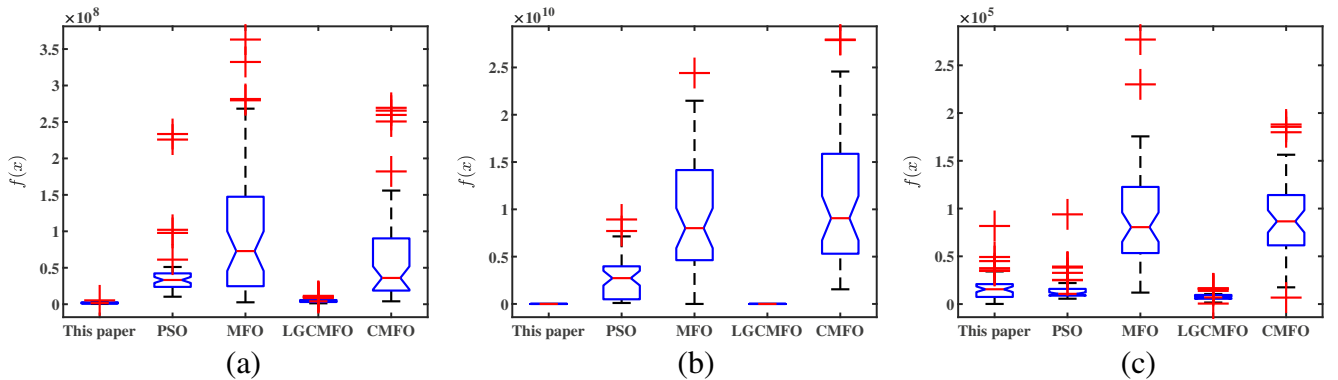


Fig. 6 30-dimensional unimodal function box plot

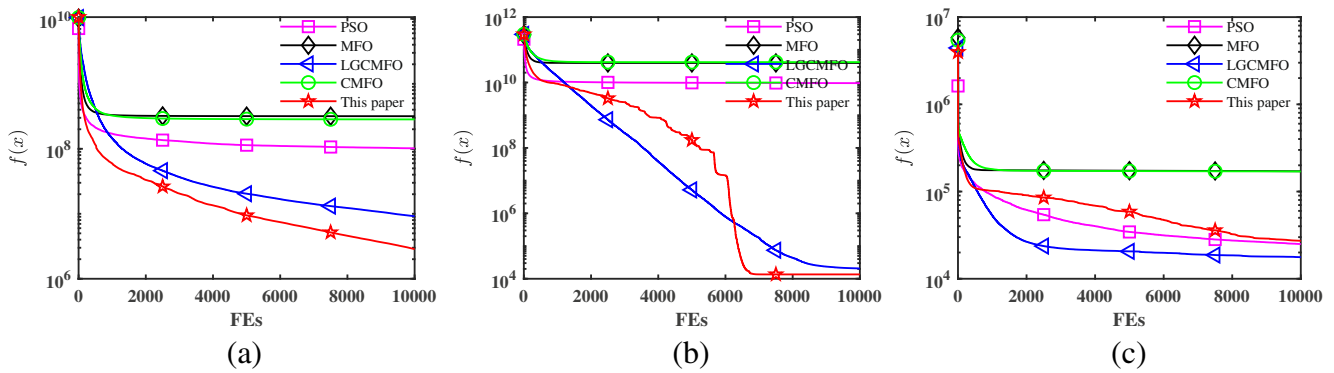


Fig. 7 50-dimension unimodal function convergence curve

diversity of the population and alleviates the clustering and trampling of the population in the multimodal problem. The mutation strategy added after the position update is completed changes the evolution direction of the population after the algorithm falls into the local optimum. This helps the proposed algorithm to jump out of the local solutions.

**4.5 Hybrid text functions result analysis**

The proposed algorithm is applied to the hybrid text functions in Tables 11, 12 and 13. Except for F20 and F23, the proposed DMMFO has the best optimization ability. For the function F20, although the final convergence result of the DMMFO is not as good as for the PSO algorithm, its performance is clearly better as compared to MFO. In the F18 test, the proposed DMMFO is significantly better than the LGCMFO and a higher latitude yields a better optimization performance. Compared with CMFO and MFO, the solution accuracy has been greatly improved. This may be the following reasons that the diversity weight embedded in the exploration stage enhance the algorithm’s global search capabilities, and the mutation strategy increases the randomness of population movement

to a certain extent. Thus, DMMFO achieves a balance between exploration and exploitation in the testing of hybrid functions of different dimensions. From the box plot presented in Fig. 18, it is observed that the results of the algorithm are stable after multiple runs. In the tests of the function F21 for different dimensions, the indicators of the DMMFO outperform other presented algorithms. Hence, although the proposed DMMFO has insufficient advantages in low dimensions, its performance is quite competitive for medium and high number of dimensions (Figs.15, 16, 17, 18, 19, and 20).

**4.6 Composition text functions result analysis**

Tables 14 15 and 16 present the comparison results for the selected composition text functions. The performance of the proposed DMMFO in comparison with MFO and CMFO on F29 and F30 demonstrates that the former has a better performance. It still maintains obvious advantages in the remaining seven test functions. Although LGCMFO’s performance is very stable on the F23 in each dimension and F24, F25, F27, and F28 in tmiddle and high number of dimensions. However, all five algorithms are unable to find the global optimum. In the 50-dimensional F26

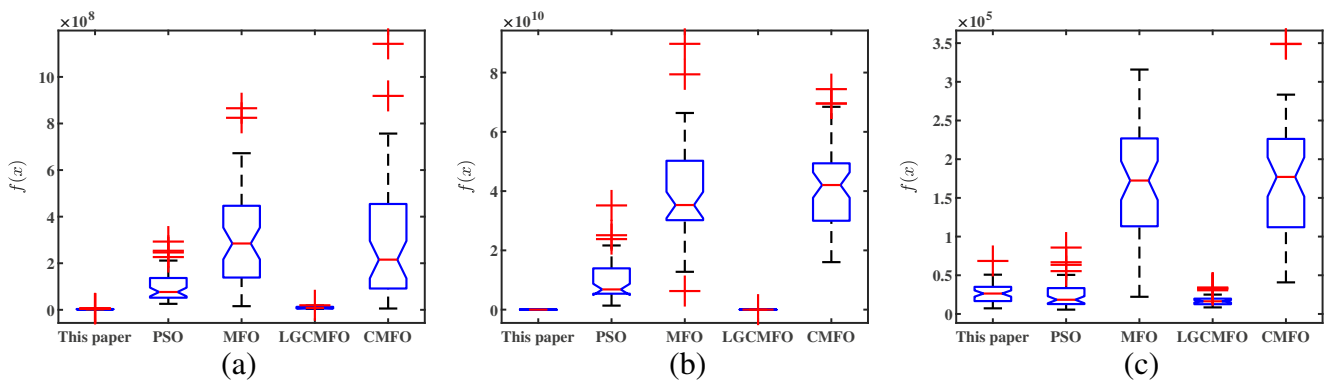


Fig. 8 50-dimension unimodal function box plot

**Table 8** 10-dimension simple multimodal function result statistics

Function		This paper	MFO	CMFO	LGCMFO	PSO
F4	Mean	<b>23.119</b>	29.807	28.633	24.777	44.484
	Std	15.594	12.891	<b>10.682</b>	15.315	61.424
	Wilcoxon		+	+	+	+
F5	Mean	<b>20.035</b>	20.122	20.158	20.180	20.350
	Std	<b>0.068</b>	0.096	0.145	0.124	0.076
	Wilcoxon		=	+	+	+
F6	Mean	<b>2.540</b>	4.430	3.714	2.603	3.516
	Std	<b>1.469</b>	1.836	1.678	1.543	1.565
	Wilcoxon		+	+	=	+
F7	Mean	<b>1.564E-01</b>	1.041E+00	5.474E-01	1.896E-01	3.692E+00
	Std	<b>1.036E-01</b>	1.839E+00	1.436E+00	1.074E-01	3.883E+00
	Wilcoxon		+	+	=	+
F8	Mean	<b>2.368E+00</b>	1.901E+01	1.808E+01	1.518E+01	1.122E+01
	Std	<b>1.533E+00</b>	8.364E+00	8.427E+00	5.822E+00	5.355E+00
	Wilcoxon		+	+	+	=
F9	Mean	<b>19.966</b>	29.145	27.796	27.029	20.253
	Std	<b>7.190</b>	12.480	12.859	11.727	8.190
	Wilcoxon		+	+	+	+
F10	Mean	<b>8.743E+01</b>	4.865E+02	4.540E+02	3.896E+02	4.627E+02
	Std	<b>6.471E+01</b>	2.270E+02	2.581E+02	2.475E+02	2.430E+02
	Wilcoxon		+	+	+	+
F11	Mean	<b>7.439E+02</b>	1.032E+03	9.387E+02	7.719E+02	8.761E+02
	Std	3.339E+02	3.139E+02	3.581E+02	<b>2.964E+02</b>	3.315E+02
	Wilcoxon		+	+	=	+
F12	Mean	<b>0.227</b>	0.328	0.351	0.504	0.949
	Std	<b>0.195</b>	0.255	0.223	0.337	0.266
	Wilcoxon		+	+	+	+
F13	Mean	<b>0.238</b>	0.321	0.316	0.258	0.324
	Std	0.083	0.108	0.097	0.106	<b>0.072</b>
	Wilcoxon		+	+	=	+
F14	Mean	<b>0.308</b>	0.481	0.469	0.321	0.356
	Std	0.143	0.286	0.310	<b>0.127</b>	0.262
	Wilcoxon		+	+	=	=
F15	Mean	<b>1.271</b>	2.515	1.817	1.468	2.399
	Std	<b>0.476</b>	6.508	0.907	0.612	0.685
	Wilcoxon		+	+	+	+
F16	Mean	3.237	3.363	3.360	2.965	<b>2.820</b>
	Std	<b>0.312</b>	0.320	0.474	0.354	0.383
	Wilcoxon		+	+	-	-
+/-/=gm		43/2/7/41				

The bold entries indicated that the value is optimal in the same row

function test, the results of the two improved algorithms are worse than MFO, but the proposed DMMFO improves the accuracy of the solution. Hence, the performance of this algorithm in the composition test functions is better than MFO, CMFO, PSO, and second to LGCMFO. The aforementioned results show that the diversity weight and mutation strategy proposed in this work significantly

enhance the optimization performance of the algorithm (Figs. 21, 22, 23, 24, 25, and 26).

#### 4.7 Population diversity comparison

This section will discuss the role of the proposed method in improving population diversity. Figures 27, 28, 29 show

**Table 9** 30-dimension simple multimodal function result statistics

Function		This paper	MFO	CMFO	LGCMFO	PSO
F4	Mean	<b>8.005E+01</b>	8.787E+02	6.358E+02	1.150E+02	2.573E+02
	Std	<b>3.500E+01</b>	8.055E+02	5.763E+02	3.600E+01	1.778E+02
	Wilcoxon		+	+	+	+
F5	Mean	<b>20.066</b>	20.292	20.260	20.554	20.925
	Std	0.067	0.170	0.160	0.267	<b>0.064</b>
	Wilcoxon		+	+	+	+
F6	Mean	18.805	21.841	23.276	<b>18.265</b>	24.482
	Std	3.167	3.292	3.207	<b>3.077</b>	3.945
	Wilcoxon		+	+	=	+
F7	Mean	<b>1.175E-02</b>	8.866E+01	8.479E+01	1.543E-02	39.974
	Std	<b>1.461E-02</b>	4.909E+01	5.422E+01	1.509E-02	41.087
	Wilcoxon		+	+	=	+
F8	Mean	<b>2.948E+01</b>	1.303E+02	1.293E+02	7.636E+01	7.626E+01
	Std	<b>8.142E+00</b>	3.723E+01	4.068E+01	1.884E+01	2.165E+01
	Wilcoxon		+	+	+	+
F9	Mean	<b>138.937</b>	193.256	199.637	158.250	152.098
	Std	34.044	43.411	52.591	<b>26.273</b>	35.519
	Wilcoxon		+	+	+	=
F10	Mean	<b>9.921E+02</b>	3.456E+03	3.302E+03	2.206E+03	3.205E+03
	Std	<b>3.113E+02</b>	8.513E+02	1.004E+03	1.165E+03	7.136E+02
	Wilcoxon		+	+	+	+
F11	Mean	<b>3.549E+03</b>	4.339E+03	4.016E+03	4.122E+03	4.521E+03
	Std	<b>5.976E+02</b>	7.773E+02	7.100E+02	8.246E+02	8.742E+02
	Wilcoxon		+	+	+	+
F12	Mean	<b>0.301</b>	0.426	0.436	0.780	2.364
	Std	<b>0.127</b>	0.196	0.186	0.601	0.291
	Wilcoxon		+	=	+	+
F13	Mean	6.441E-01	1.666E+00	1.672E+00	<b>5.231E-01</b>	1.008E+00
	Std	1.247E-01	1.206E+00	1.148E+00	<b>1.128E-01</b>	7.852E-01
	Wilcoxon		+	+	-	+
F14	Mean	9.007E-01	2.676E+01	2.627E+01	<b>4.157E-01</b>	1.114E+01
	Std	3.167E-01	2.279E+01	1.802E+01	<b>2.271E-01</b>	1.523E+01
	Wilcoxon		+	+	-	+
F15	Mean	<b>1.468E+01</b>	1.487E+05	1.256E+05	1.559E+01	3.528E+02
	Std	<b>6.265E+00</b>	2.921E+05	2.111E+05	6.271E+00	1.462E+03
	Wilcoxon		+	+	+	+
F16	Mean	12.381	12.576	12.577	12.116	<b>11.989</b>
	Std	0.473	0.598	0.542	0.510	<b>0.404</b>
	Wilcoxon		+	+	-	-
+/-/=gm		44/4/4/40				

The bold entries indicated that the value is optimal in the same row

a comparison chart of the diversity of MFO and DMMFO under six different types of functions. The average distance between all solutions during each iteration is presented in the diversity chart of the algorithm.

As presented in Fig. 27, the population of proposed DMMFO is very active in three different dimensions. For

the functions F2, F7, and F8 under 10 dimensions condition, the MFO curve becomes smoother at the initial stage of iteration due to the rapid loss of algorithm's diversity. In functional tests of 30 and 50 dimensions, it can be clearly observed that the average distance between the proposed DMMFO populations in the exploration stage is



**Table 10** 50-dimension simple multimodal function result statistics

Function		This paper	MFO	CMFO	LGCMFO	PSO
F4	Mean	<b>9.830E+01</b>	4.411E+03	4.451E+03	1.602E+02	8.013E+02
	Std	<b>2.588E+01</b>	3.313E+03	3.081E+03	5.315E+01	6.854E+02
	Wilcoxon		+	+	+	+
F5	Mean	<b>20.071</b>	20.289	20.360	20.721	21.134
	Std	0.062	0.160	0.125	0.293	<b>0.041</b>
	Wilcoxon		+	+	+	+
F6	Mean	<b>39.567</b>	44.878	45.251	39.998	47.296
	Std	<b>4.000</b>	4.469	4.396	4.065	4.257
	Wilcoxon		+	+	=	+
F7	Mean	<b>1.223E-02</b>	4.366E+02	3.960E+02	2.739E-02	6.736E+01
	Std	<b>1.643E-02</b>	1.598E+02	1.730E+02	1.852E-02	5.412E+01
	Wilcoxon		+	+	+	+
F8	Mean	<b>7.415E+01</b>	2.756E+02	2.667E+02	1.582E+02	1.993E+02
	Std	<b>1.902E+01</b>	8.135E+01	6.925E+01	2.577E+01	3.792E+01
	Wilcoxon		+	+	+	+
F9	Mean	343.583	474.315	463.644	<b>281.711</b>	340.330
	Std	61.907	94.084	70.681	<b>36.522</b>	75.583
	Wilcoxon		+	+	-	-
F10	Mean	<b>2.651E+03</b>	6.184E+03	5.898E+03	3.927E+03	6.608E+03
	Std	<b>5.638E+02</b>	1.714E+03	1.724E+03	2.197E+03	1.397E+03
	Wilcoxon		+	+	-	+
F11	Mean	<b>6.327E+03</b>	7.678E+03	7.524E+03	7.766E+03	9.139E+03
	Std	<b>8.631E+02</b>	1.091E+03	1.086E+03	1.895E+03	1.438E+03
	Wilcoxon		+	+	+	+
F12	Mean	<b>0.387</b>	0.461	0.441	1.055	3.239
	Std	<b>0.176</b>	0.193	0.227	0.602	0.303
	Wilcoxon		+	+	+	+
F13	Mean	6.945E-01	4.002E+00	4.060E+00	<b>6.696E-01</b>	1.041E+00
	Std	<b>1.083E-01</b>	1.189E+00	9.975E-01	1.126E-01	7.990E-01
	Wilcoxon		+	+	=	=
F14	Mean	1.098	1.239E+02	1.102E+02	<b>4.947E-01</b>	21.357
	Std	0.321	5.117E+01	4.413E+01	<b>2.511E-01</b>	21.977
	Wilcoxon		+	+	-	+
F15	Mean	5.805E+01	1.412E+06	1.271E+06	<b>4.222E+01</b>	4.004E+03
	Std	2.180E+01	2.056E+06	2.180E+06	<b>1.502E+01</b>	8.141E+03
	Wilcoxon		+	+	-	+
F16	Mean	21.887	21.970	22.106	<b>21.459</b>	21.467
	Std	0.6479	0.6168	0.6299	0.664	<b>0.461</b>
	Wilcoxon		+	+	=	=
+/-/=gm		42/5/5/37				

The bold entries indicated that the value is optimal in the same row

greater than that of the MFO. Moreover, the diversity curve of the algorithm quickly approaches 0 when MFO optimizes 50 dimensional functions F8, F18, and F30. From the results in Section 4.4, this indicates that the population has fallen into the trap of local minima. Due to the effect of the diversity weight, the population of proposed DMMFO is evenly distributed over the search space at the beginning

of the iterations. It balances the exploration and exploitation ability of the algorithm. In addition, this allows the algorithm to perform a global search more efficiently and jump out of the local minima if trapped in one.

The convergence of the algorithm includes two stages, divergence and contraction of particles. The divergence process of the particles of the algorithm is the premise of

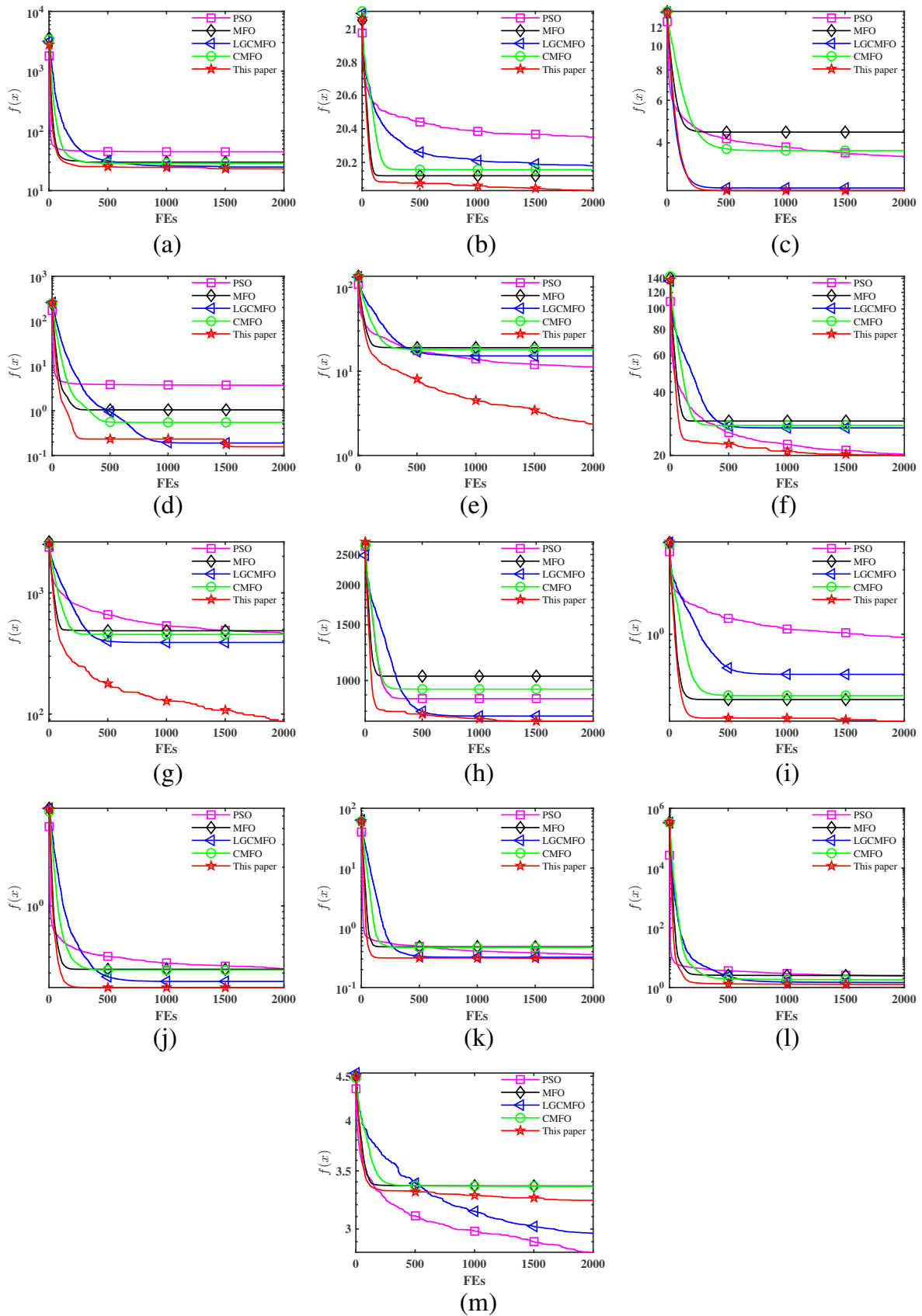


Fig. 9 10-dimension simple multimodal function convergence curve

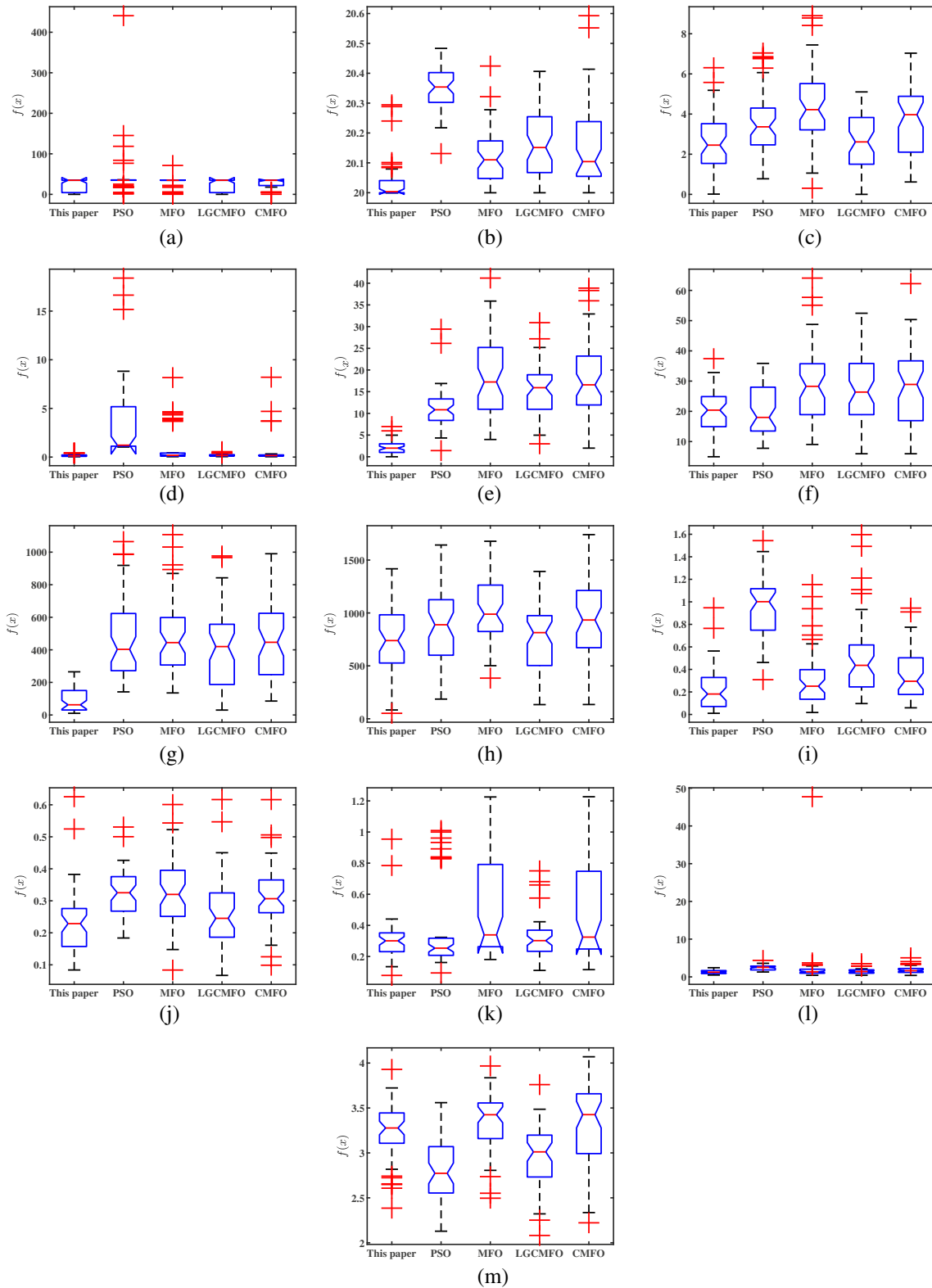


Fig. 10 10-dimension simple multimodal function box plot

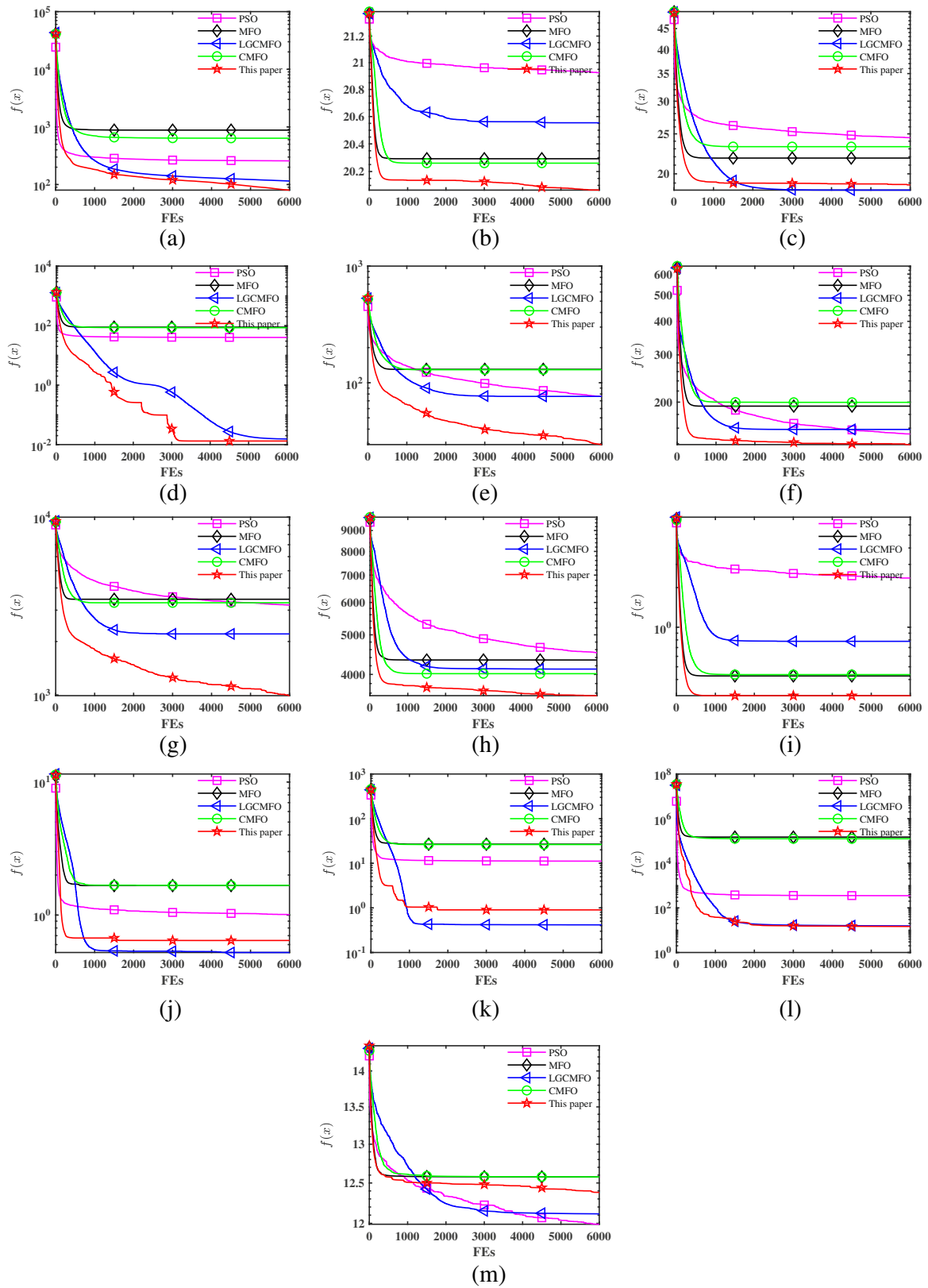


Fig. 11 30-dimension simple multimodal function convergence curve

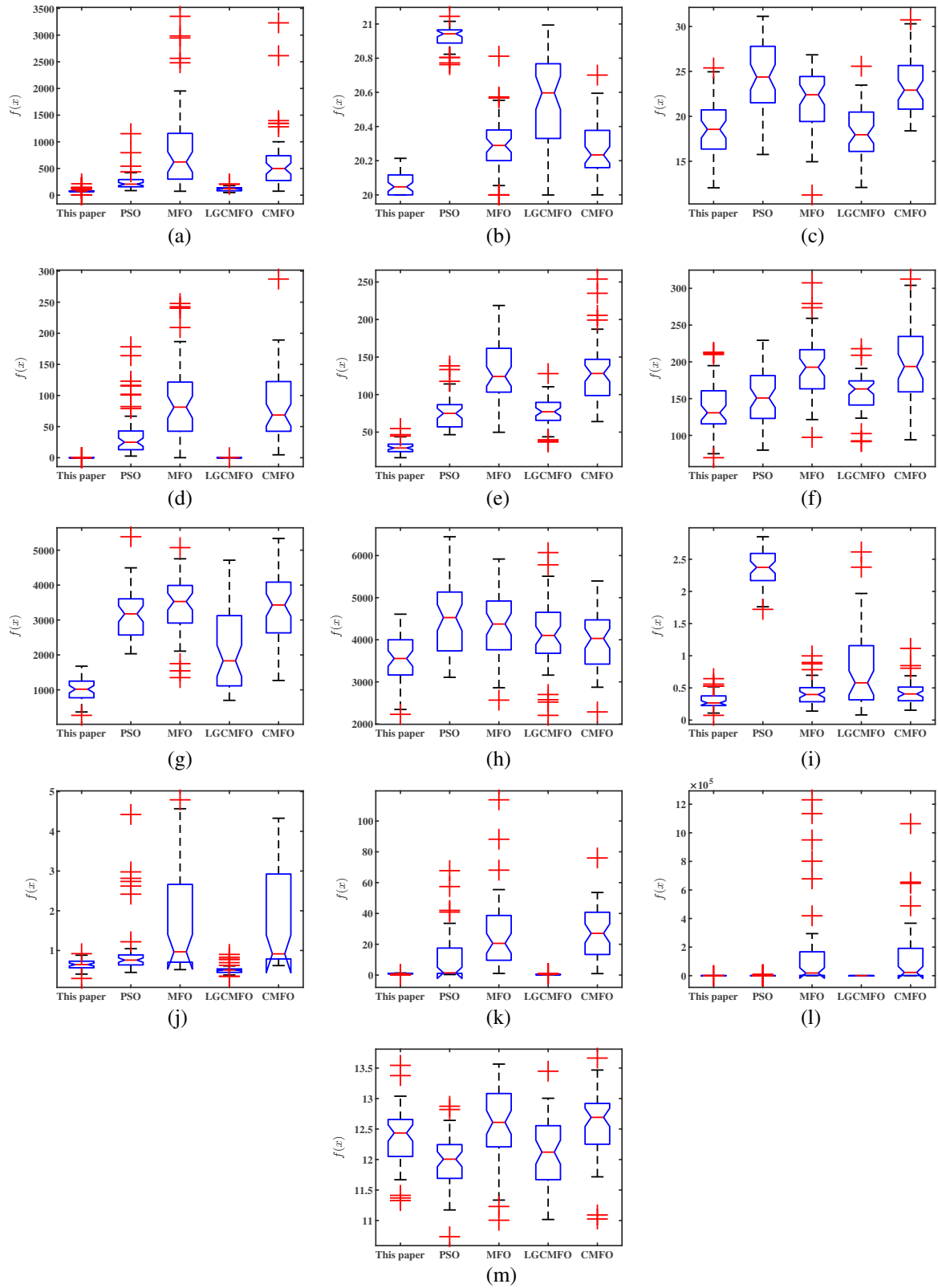


Fig. 12 30-dimension simple multimodal function box plot



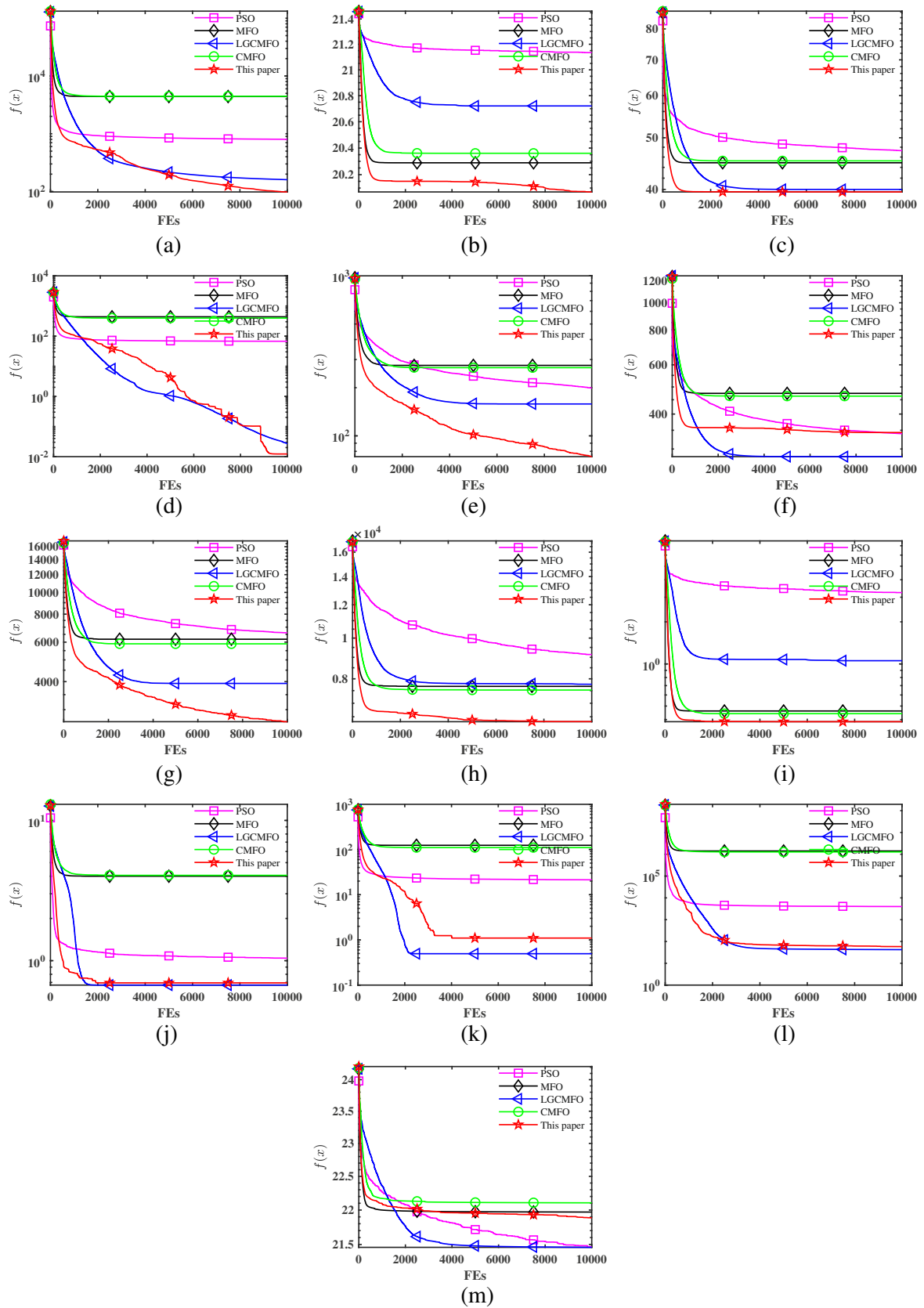


Fig. 13 50-dimension simple multimodal function convergence curve

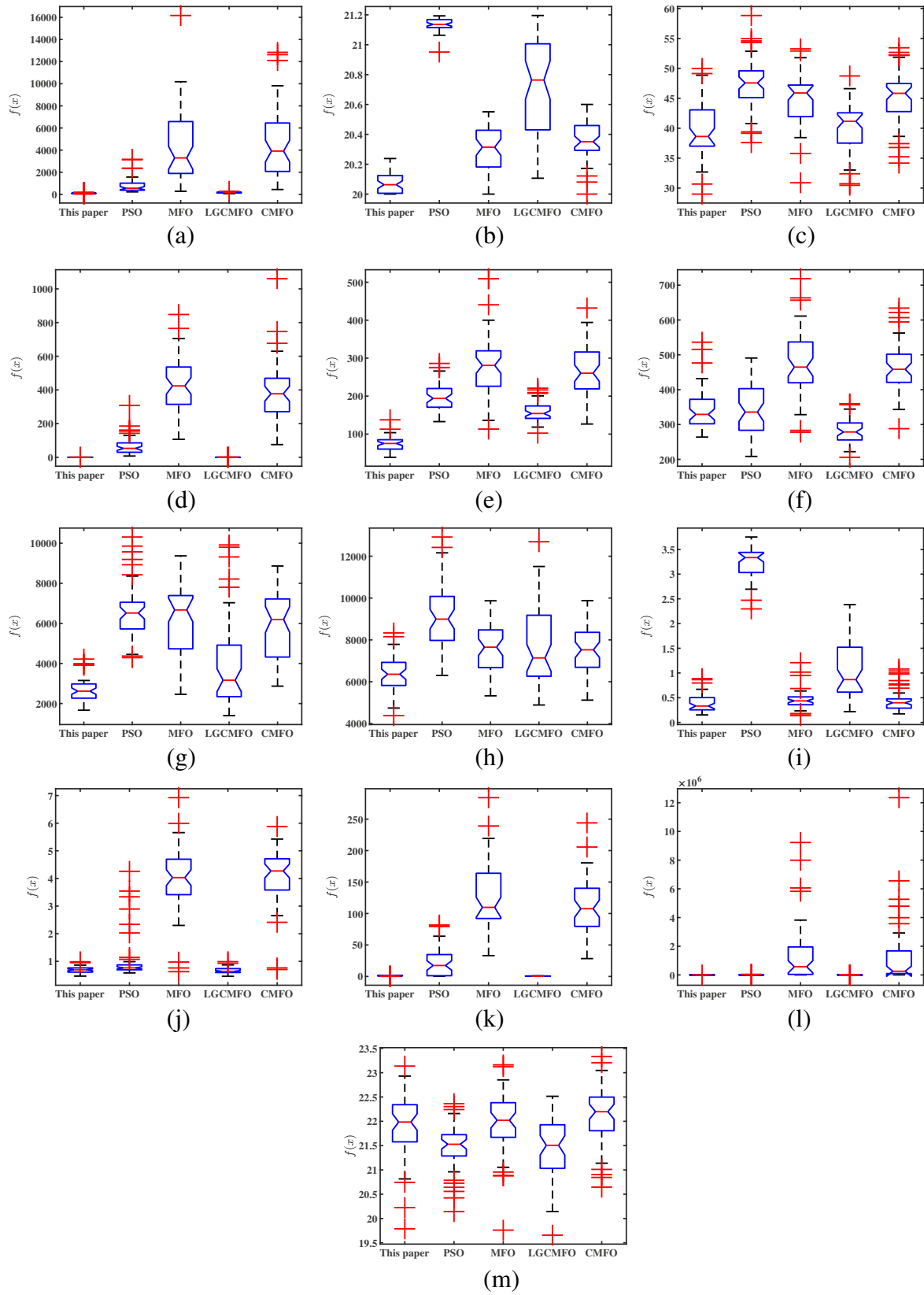


Fig. 14 50-dimension simple multimodal function box plot

**Table 11** 10-dimension hybrid function test result statistics

Function		This paper	MFO	CMFO	LGCMFO	PSO
F17	Mean	2.120E+04	3.579E+04	3.944E+04	6.647E+03	<b>5.755E+03</b>
	Std	2.677E+04	6.317E+04	6.245E+04	1.092E+04	<b>5.358E+03</b>
	Wilcoxon		+	+	-	-
F18	Mean	<b>8.906E+03</b>	1.399E+04	1.179E+04	9.387E+03	1.151E+04
	Std	9.956E+03	1.488E+04	1.419E+04	<b>3.522E+03</b>	1.054E+04
	Wilcoxon		+	+	+	+
F19	Mean	<b>1.908</b>	2.900	2.881	2.611	4.153
	Std	<b>0.783</b>	1.029	1.117	1.138	1.431
	Wilcoxon		+	+	+	+
F20	Mean	4.253E+03	1.481E+04	9.115E+03	4.802E+03	<b>3.251E+02</b>
	Std	4.803E+03	3.623E+04	9.777E+03	3.350E+03	<b>5.150E+02</b>
	Wilcoxon		+	+	+	-
F21	Mean	<b>4.801E+03</b>	8.570E+03	7.636E+03	8.020E+03	6.396E+03
	Std	<b>6.701E+03</b>	1.032E+04	8.634E+03	7.921E+03	8.259E+03
	Wilcoxon		+	+	+	+
F22	Mean	<b>35.016</b>	68.387	57.002	44.653	91.915
	Std	<b>46.612</b>	66.503	53.854	47.612	58.562
	Wilcoxon		+	+	+	+
+/-/=gm		21/3/0/18				

The bold entries indicated that the value is optimal in the same row

the shrinking process, i.e., the diversity of the population is reduced only if the diversity of population is increased first. This is the purpose of introducing diversity weight in the proposed DMMFO.

### 5 Engineering optimization examples

In this section, four typical practical complex constraints engineering optimization problems are employed to further

**Table 12** 30-dimensional hybrid function test result statistics

Function		This paper	MFO	CMFO	LGCMFO	PSO
F17	Mean	<b>2.903E+05</b>	3.530E+06	3.044E+06	7.973E+05	1.751E+06
	Std	<b>1.880E+05</b>	3.973E+06	3.911E+06	5.065E+05	2.723E+06
	Wilcoxon		+	+	+	+
F18	Mean	<b>5.321E+03</b>	1.311E+07	1.055E+07	5.169E+06	1.034E+06
	Std	<b>6.041E+03</b>	6.526E+07	5.057E+07	3.616E+07	1.364E+06
	Wilcoxon		+	+	+	+
F19	Mean	<b>10.856</b>	57.931	52.231	16.703	34.575
	Std	<b>2.222</b>	44.424	44.472	15.838	45.675
	Wilcoxon		+	+	+	+
F20	Mean	3.316E+04	7.726E+04	5.863E+04	3.398E+04	<b>5.179E+03</b>
	Std	1.922E+04	1.084E+05	3.462E+04	2.069E+04	<b>1.080E+04</b>
	Wilcoxon		+	+	+	-
F21	Mean	<b>1.619E+05</b>	1.042E+06	7.326E+05	3.139E+05	3.810E+05
	Std	<b>1.518E+05</b>	2.249E+06	1.008E+06	2.644E+05	4.762E+05
	Wilcoxon		+	+	+	+
F22	Mean	<b>539.756</b>	754.711	746.957	633.668	562.269
	Std	<b>164.665</b>	291.769	232.733	234.314	198.032
	Wilcoxon		+	+	+	+
+/-/=gm		23/1/0/22				

The bold entries indicated that the value is optimal in the same row

**Table 13** 50-dimension hybrid function test result statistics

Function		This paper	MFO	CMFO	LGCMFO	PSO
F17	Mean	<b>4.554E+05</b>	1.062E+07	1.065E+07	1.592E+06	8.284E+06
	Std	<b>2.859E+05</b>	1.175E+07	1.090E+07	2.069E+06	1.155E+07
	Wilcoxon		+	+	+	+
F18	Mean	<b>3.266E+03</b>	2.755E+08	3.096E+08	3.884E+03	3.776E+08
	Std	<b>1.904E+03</b>	4.338E+08	4.281E+08	2.029E+03	6.692E+08
	Wilcoxon		+	+	+	+
F19	Mean	<b>23.495</b>	192.789	138.281	36.627	123.501
	Std	<b>3.339</b>	132.587	72.996	17.843	92.890
	Wilcoxon		+	+	+	+
F20	Mean	4.455E+04	1.252E+05	1.201E+05	3.029E+04	<b>1.455E+04</b>
	Std	2.718E+04	1.135E+05	9.269E+04	<b>1.396E+04</b>	2.173E+04
	Wilcoxon		+	+	-	-
F21	Mean	<b>3.550E+05</b>	7.089E+06	5.178E+06	6.471E+05	1.635E+06
	Std	<b>2.231E+05</b>	1.175E+07	9.058E+06	4.353E+05	1.967E+06
	Wilcoxon		+	+	+	+
F22	Mean	<b>1.274E+03</b>	1.634E+03	1.610E+03	1.436E+03	1.338E+03
	Std	<b>3.373E+02</b>	3.703E+02	3.997E+02	4.129E+02	3.869E+02
	Wilcoxon		+	+	=	=
+/-=/gm		20/2/2/18				

The bold entries indicated that the value is optimal in the same row

investigate the performance of DMMFO and compared with other reported methods.

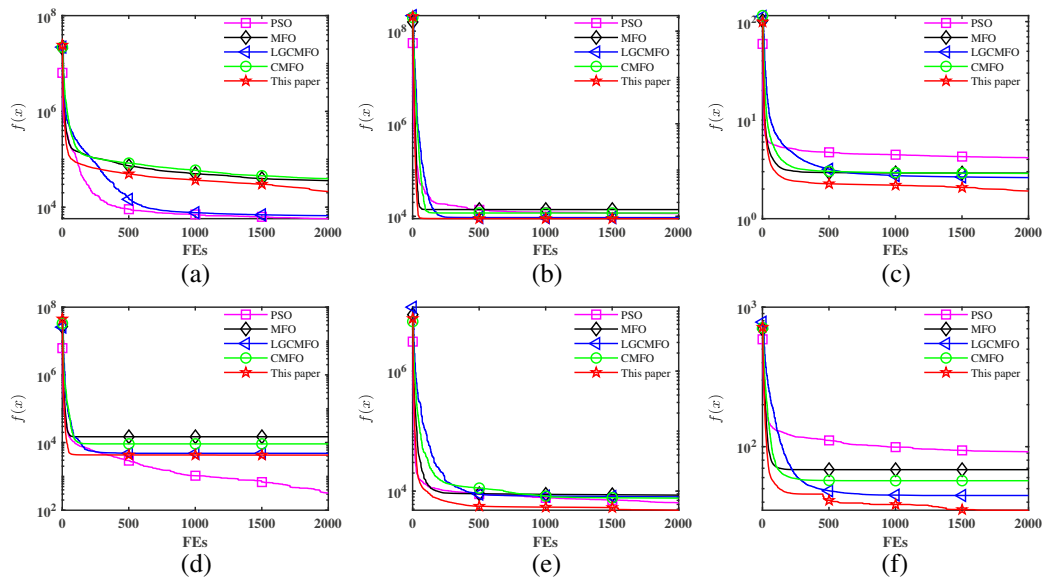
### 5.1 Problem of tension/compression spring design

This problem was raised by Belegundu (1982) [45], and it requires solving for minimizing the weight of a tension / compression spring (Fig. 30).

There are three design variables for this problem, namely, the wire diameter  $d$  ( $x_1$ ), the coil diameter  $D$  ( $x_2$ ), and the number of active coils  $P$  ( $x_3$ ). The optimization model of the problem is formulated as

Minimize:

$$f(x) = (x_3 + 2)x_2x_1^2 \tag{16}$$



**Fig. 15** 10-dimension hybrid function convergence curve

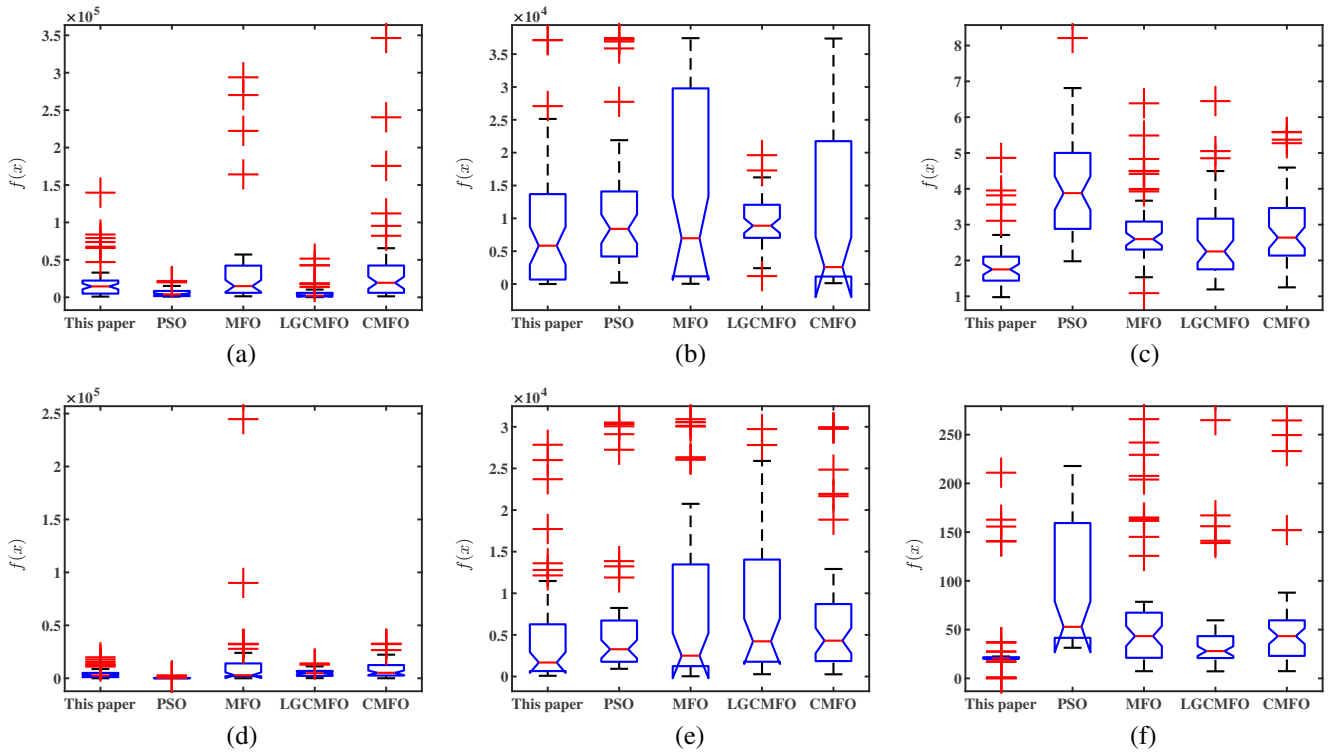


Fig. 16 10-dimension hybrid function box plot

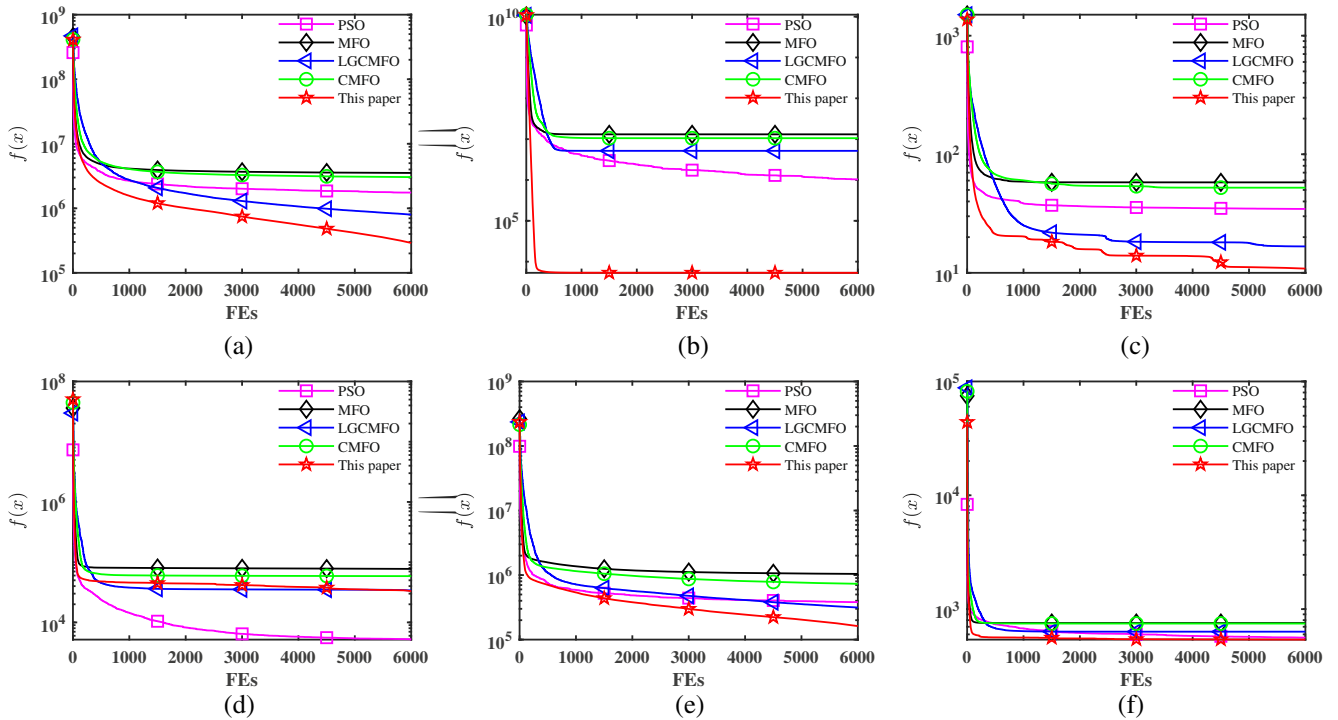


Fig. 17 30-dimension hybrid function convergence curve



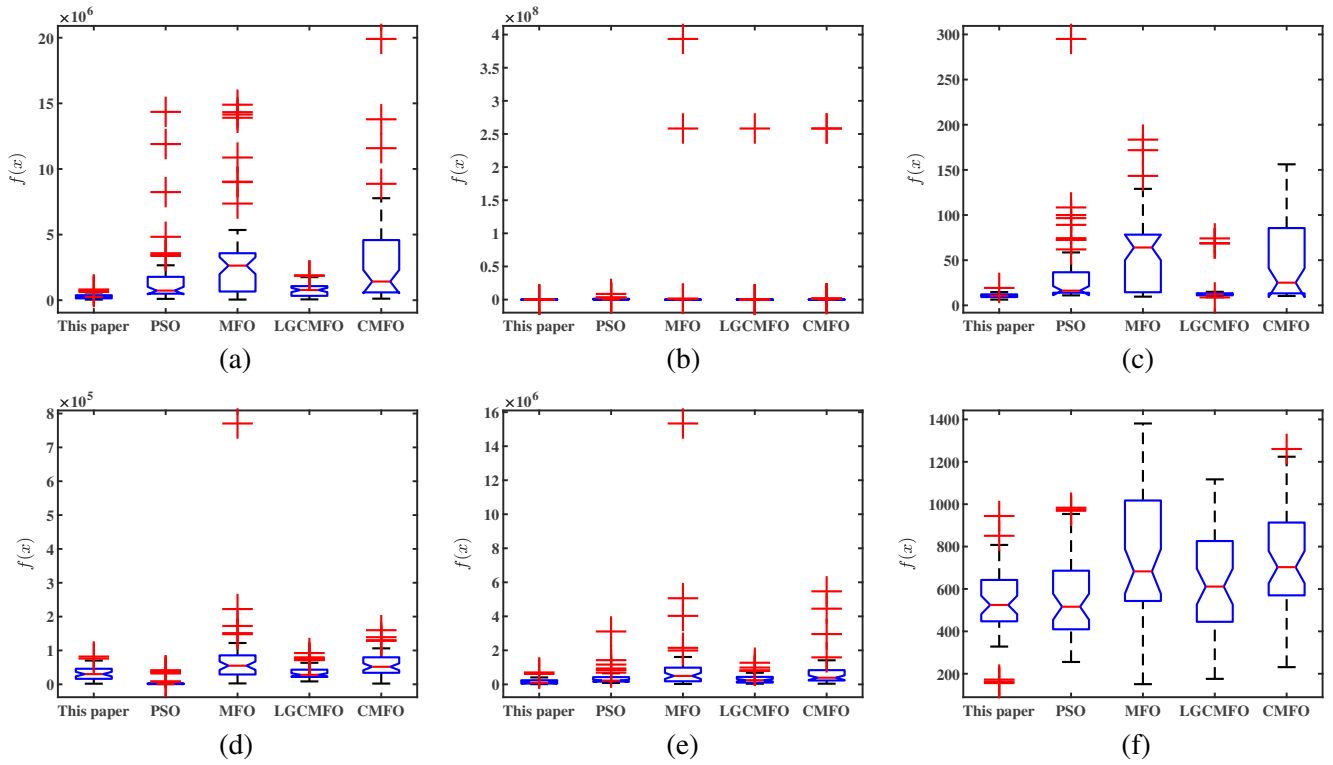


Fig. 18 30-dimension hybrid function box plot

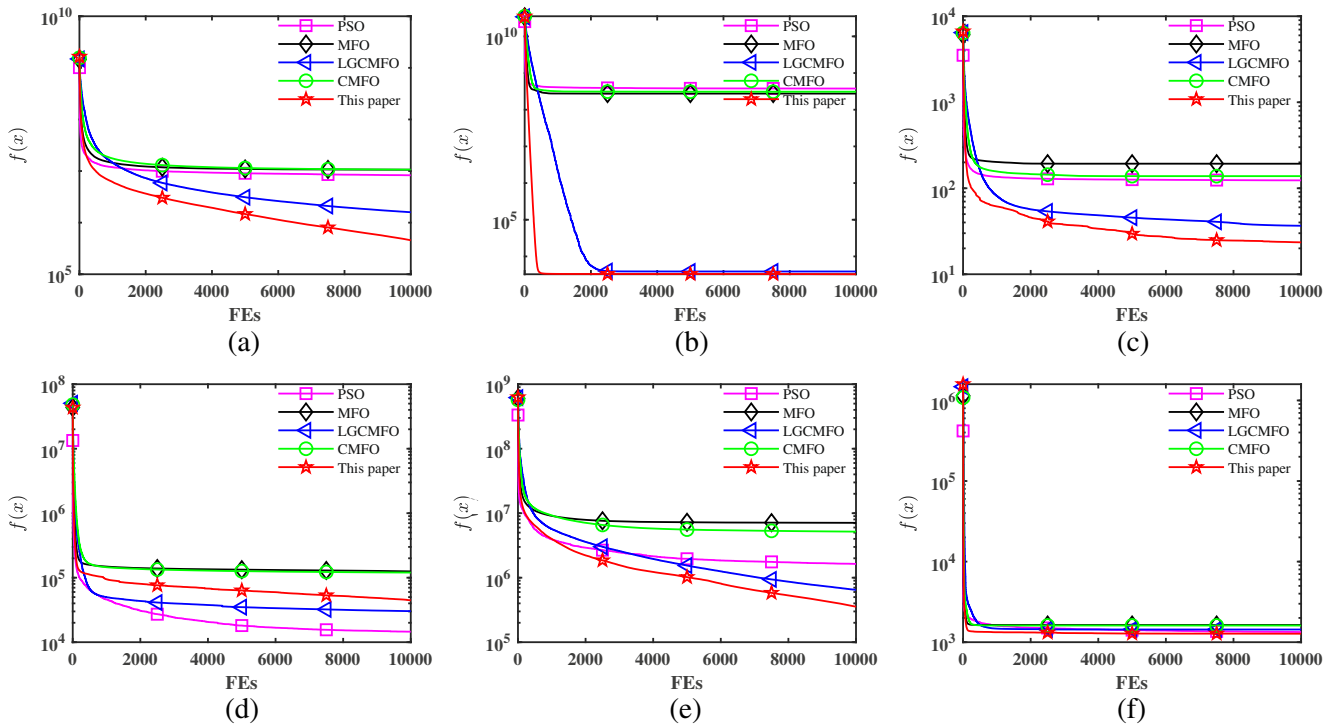


Fig. 19 50-dimension hybrid function convergence curve

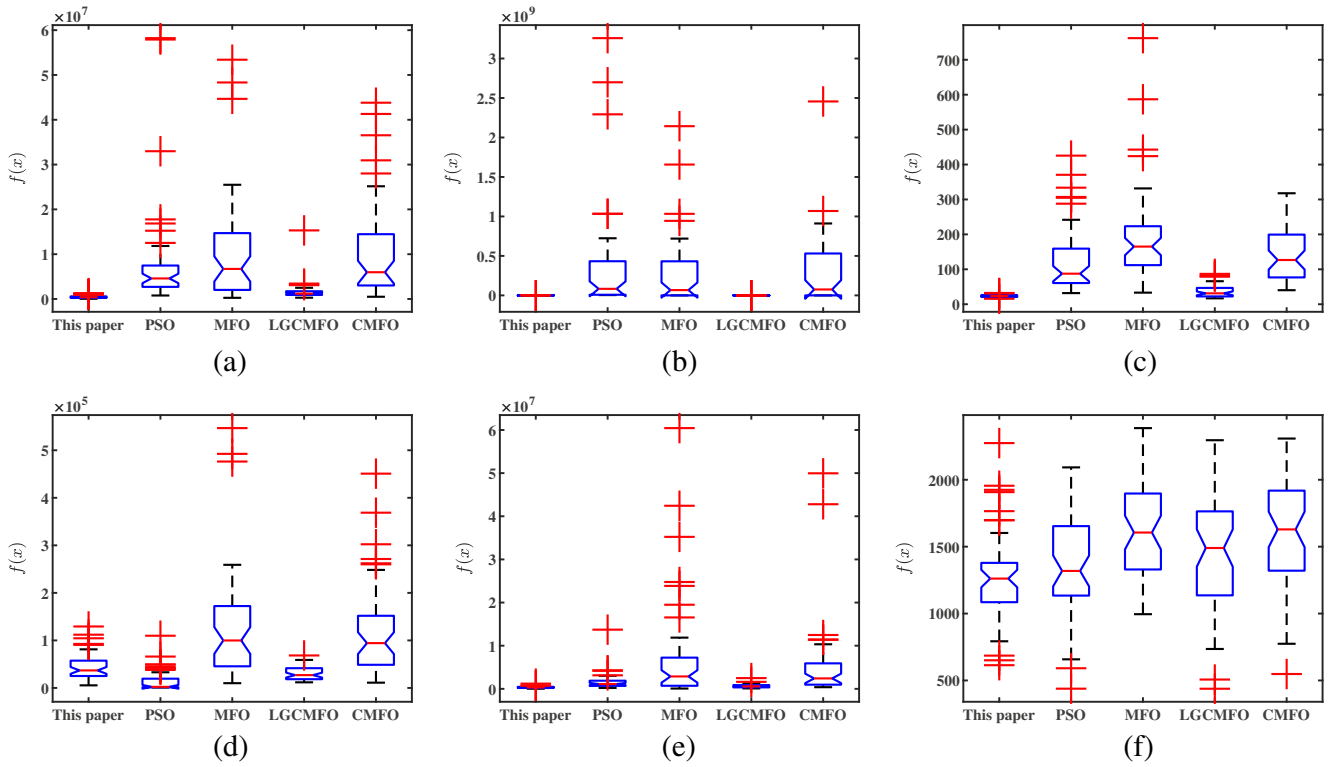


Fig. 20 50-dimension hybrid function box plot

Table 14 10-dimension composition function test result statistics

Function		This paper	MFO	CMFO	LGCMFO	PSO
F23	Mean	329.457	334.021	333.513	<b>200.000</b>	331.664
	Std	1.696E-13	7.144	6.856	<b>0.000</b>	6.900
	Wilcoxon		+	+	-	=
F24	Mean	133.530	138.355	135.737	179.166	<b>125.963</b>
	Std	10.700	12.072	9.839	30.726	<b>8.234</b>
	Wilcoxon		+	=	+	-
F25	Mean	196.232	195.072	197.570	200.000	<b>176.336</b>
	Std	14.586	18.018	14.163	<b>0.000</b>	30.835
	Wilcoxon		+	+	+	-
F26	Mean	<b>100.221</b>	100.300	100.287	100.231	100.280
	Std	<b>0.096</b>	0.155	0.147	0.121	0.113
	Wilcoxon		=	=	=	=
F27	Mean	288.967	329.284	320.465	<b>169.096</b>	325.974
	Std	169.116	158.706	160.149	<b>70.813</b>	168.842
	Wilcoxon		+	+	-	+
F28	Mean	387.255	393.140	389.708	<b>200.000</b>	544.849
	Std	26.885	28.730	23.400	<b>0.000</b>	115.939
	Wilcoxon		+	=	-	+
F29	Mean	<b>6.219E+02</b>	1.114E+03	1.009E+03	1.188E+03	6.043E+05
	Std	<b>3.474E+02</b>	6.662E+02	5.471E+02	5.867E+02	1.128E+06
	Wilcoxon		+	+	+	+

**Table 14** (continued)

Function		This paper	MFO	CMFO	LGCMFO	PSO
F30	Mean	<b>647.286</b>	882.936	741.143	731.936	1268.894
	Std	<b>180.944</b>	419.126	342.517	304.145	681.540
	Wilcoxon		+	+	+	+
+/-=/gm		20/5/7/15				

The bold entries indicated that the value is optimal in the same row

Subject to:

$$0.25 \leq x_2 \leq 1.30 \tag{22}$$

$$2.00 \leq x_3 \leq 15.0 \tag{23}$$

$$g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0 \tag{17}$$

$$g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} \leq 0 \tag{18}$$

$$g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0 \tag{19}$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \tag{20}$$

Variable range:

$$0.05 \leq x_1 \leq 2.00 \tag{21}$$

This problem has been solved by using different heuristic algorithms, for instance, PSO [46], ES[47], GA[48], DE[49]. In Table 17, the best results of this paper are compared with the other methods. It is observed that the proposed DMMFO effectively solves this problem and provides the best design solution.

### 5.2 Problem of pressure vessel design

The design problem of a pressure vessel was first proposed by Kannan et al. [50]. This problem focuses on computing the minimum total cost (including material, forming, and

**Table 15** 30-dimension composition function test result statistics

Function		This paper	MFO	CMFO	LGCMFO	PSO
F23	Mean	315.244	368.105	356.491	<b>200.000</b>	336.763
	Std	<b>0.000</b>	39.009	27.262	<b>0.000</b>	24.178
	Wilcoxon		+	+	-	+
F24	Mean	242.625	273.841	271.116	<b>200.000</b>	234.169
	Std	5.806	30.870	23.552	<b>0.000</b>	8.696
	Wilcoxon		+	+	-	=
F25	Mean	206.868	213.600	214.472	<b>200.000</b>	216.565
	Std	2.579	5.273	7.278	<b>0.000</b>	5.667
	Wilcoxon		+	+	-	+
F26	Mean	<b>100.703</b>	101.667	101.753	108.494	111.373
	Std	<b>0.477</b>	1.033	1.061	26.984	26.581
	Wilcoxon		+	+	+	+
F27	Mean	732.959	922.224	911.331	<b>200.000</b>	886.319
	Std	227.407	174.326	194.839	<b>0.000</b>	266.369
	Wilcoxon		+	+	-	+
F28	Mean	9.635E+02	1.079E+03	1.037E+03	<b>2.000E+02</b>	2.024E+03
	Std	6.421E+01	1.967E+02	9.204E+01	<b>0.000E+00</b>	5.537E+02
	Wilcoxon		+	+	-	+
F29	Mean	<b>2.664E+03</b>	2.680E+06	1.806E+06	3.524E+03	1.555E+07
	Std	<b>1.840E+03</b>	3.671E+06	3.028E+06	8.085E+03	2.170E+07
	Wilcoxon		+	+	+	+
F30	Mean	<b>4.247E+03</b>	3.410E+04	3.006E+04	6.843E+03	9.471E+04
	Std	<b>1.387E+03</b>	3.716E+04	2.051E+04	1.727E+04	1.502E+05
	Wilcoxon		+	+	+	+
+/-=/gm		26/5/1/21				

The bold entries indicated that the value is optimal in the same row

**Table 16** 50-dimension composition function test result statistics

Function		This paper	MFO	CMFO	LGCMFO	PSO
F23	Mean	344.035	521.264	506.079	<b>200.000</b>	434.891
	Std	1.172E-06	95.743	95.311	<b>0.000</b>	65.035
	Wilcoxon		+	+	-	+
F24	Mean	283.797	404.821	392.899	<b>200.000</b>	294.281
	Std	5.496	68.039	59.066	<b>0.000</b>	17.179
	Wilcoxon		+	+	-	=
F25	Mean	212.711	239.865	233.853	<b>200.000</b>	243.218
	Std	4.494	24.128	19.089	<b>0.000</b>	11.370
	Wilcoxon		+	+	-	+
F26	Mean	<b>116.634</b>	137.896	117.070	156.250	157.440
	Std	49.595	94.669	62.266	49.357	<b>47.820</b>
	Wilcoxon		+	=	+	+
F27	Mean	1.419E+03	1.597E+03	1.561E+03	<b>2.000E+02</b>	1.696E+03
	Std	9.280E+01	1.105E+02	1.226E+02	<b>0.000E+00</b>	2.165E+02
	Wilcoxon		+	+	-	+
F28	Mean	1.660E+03	1.769E+03	1.757E+03	<b>2.000E+02</b>	4.340E+03
	Std	3.651E+02	5.492E+02	4.075E+02	<b>0.000E+00</b>	8.404E+02
	Wilcoxon		+	+	-	+
F29	Mean	8.763E+06	3.384E+07	3.406E+07	<b>2.000E+02</b>	1.402E+08
	Std	1.560E+07	1.080E+07	1.105E+07	<b>0.000E+00</b>	1.020E+08
	Wilcoxon		+	+	-	+
F30	Mean	1.231E+04	1.232E+05	1.098E+05	<b>2.000E+02</b>	7.541E+05
	Std	2.587E+03	7.945E+04	7.763E+04	<b>0.000E+00</b>	8.150E+05
	Wilcoxon		+	+	-	+
+/-/=gm		23/7/2/16				

The bold entries indicated that the value is optimal in the same row

welding costs) of the pressure vessel. As presented in Fig. 31, the vessel has four design variables, namely,  $T_s$  ( $X_1$ , cylinder thickness),  $T_h$  ( $X_2$ , head cover thickness),  $R$  ( $X_3$ , inner radius of the cylinder), and  $L$  ( $X_4$ , cylinder length). Two variables of  $T_s$ ,  $T_h$  are an integer multiple of 0.0625in of the thickness of the steel plate. The optimization model of this problem is formulated as below.

Minimize:

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (24)$$

Subject to:

$$g_1(x) = -x_1 + 0.193x_3 \leq 0 \quad (25)$$

$$g_2(x) = -x_3 + 0.00954x_3 \leq 0 \quad (26)$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \quad (27)$$

$$g_4(x) = x_4 - 240 \leq 0 \quad (28)$$

Variable range:

$$0 < x_1 \leq 99 \quad (29)$$

$$0 < x_2 \leq 99 \quad (30)$$

$$10 < x_3 \leq 200 \quad (31)$$

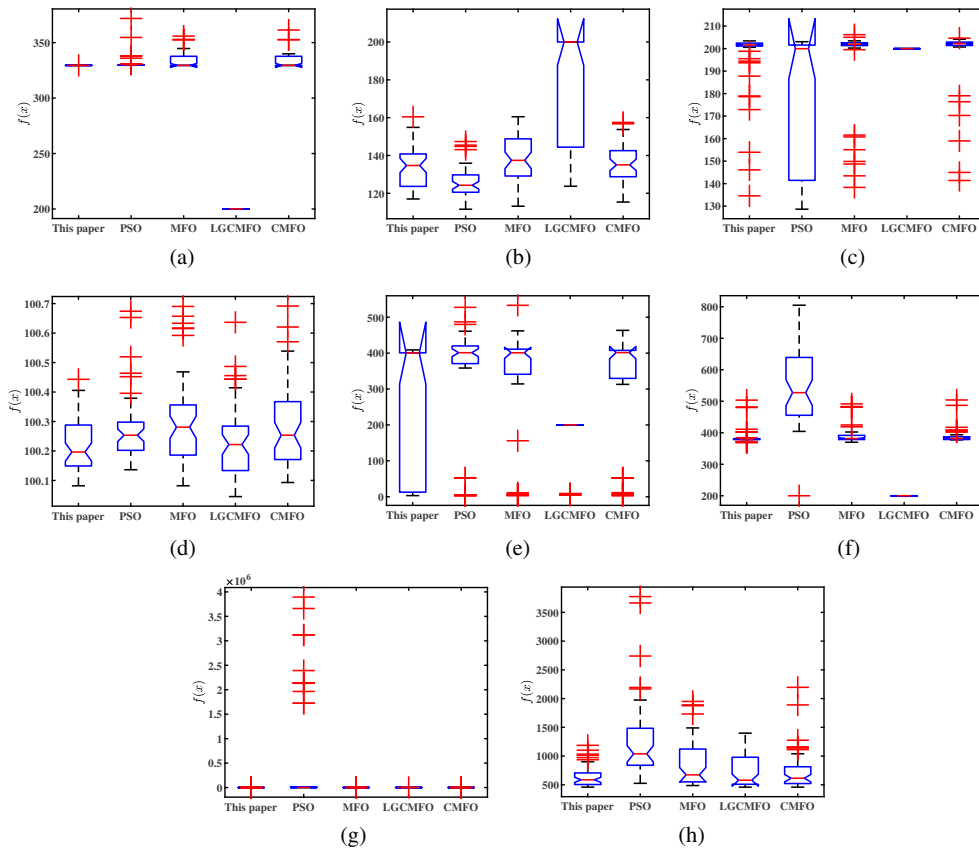
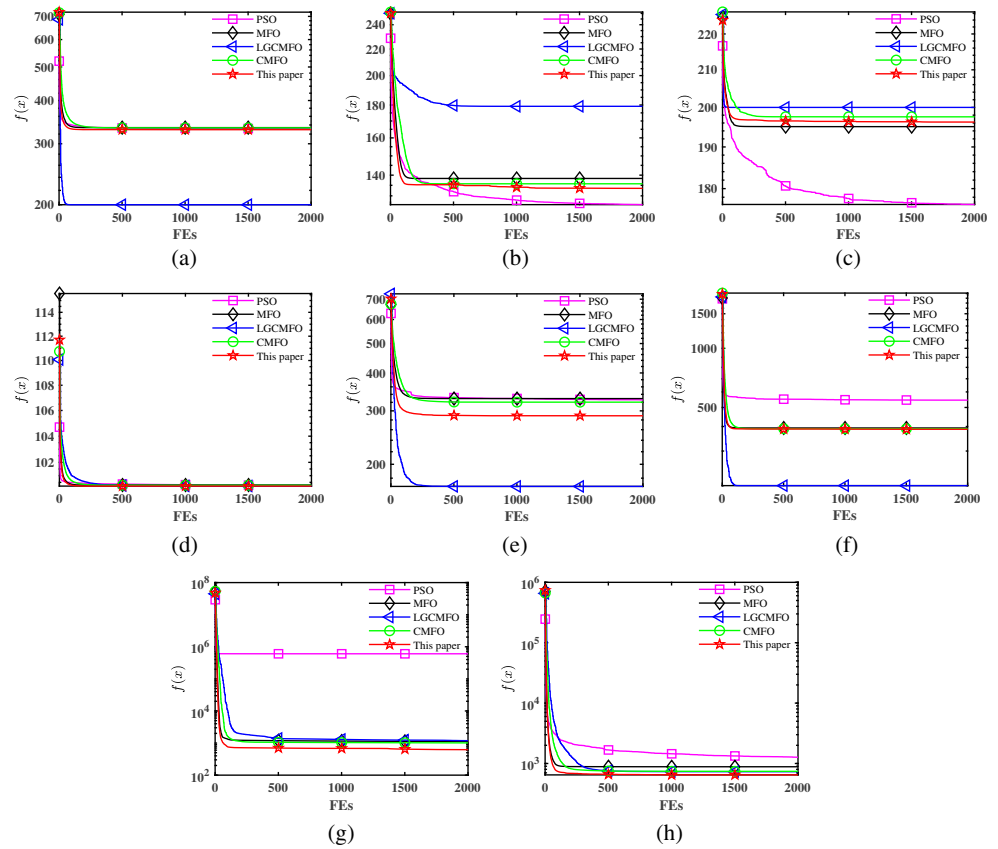
$$10 < x_4 \leq 200 \quad (32)$$

The proposed DMMFO is used to solve this problem, and its results are compared with PSO [46], GA [51], DE [49] and ACO [52] as presented in Table 18. Table 18 presents that an optimal design cost is obtained by using the algorithm proposed in this work.

### 5.3 Problem of cantilever beam design

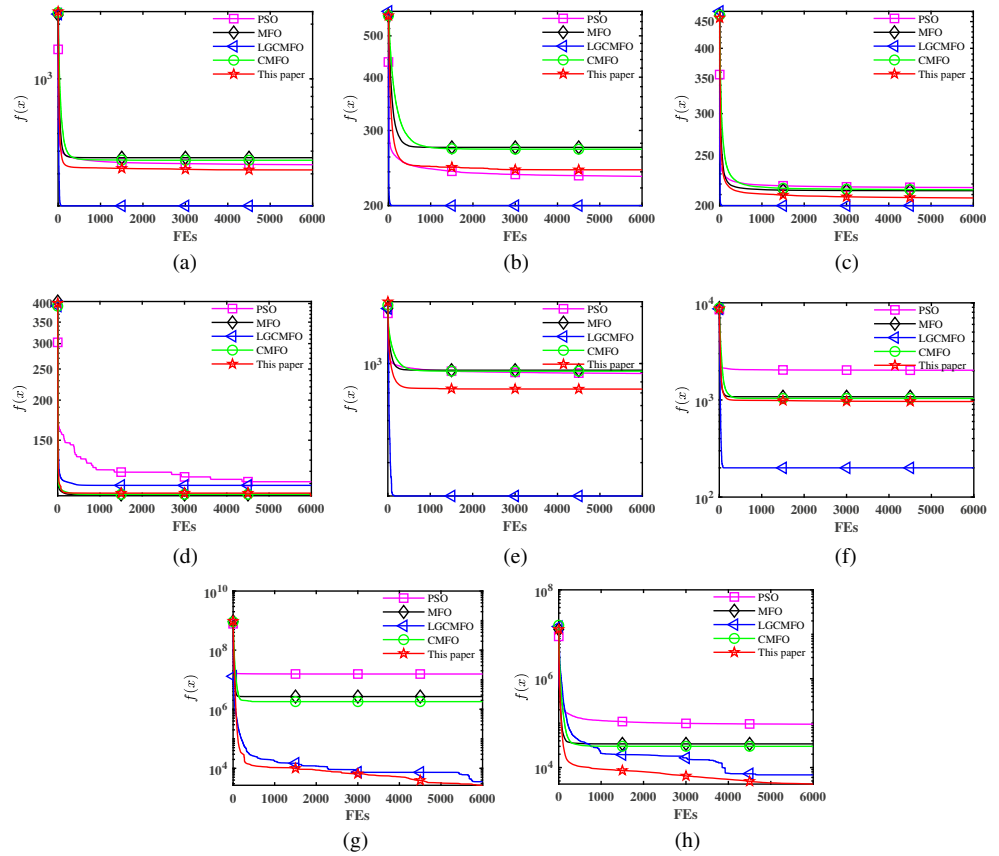
It is a weight minimizing problem for solving square cross section cantilever beam [53]. As presented in Fig. 32, the cantilever beam is composed of five hollow blocks, so the number of parameters is also five, and the design parameter  $X_i$  is the side length of the square block. The comparison results of the proposed algorithm with the

**Fig. 21** 10-dimension composition function convergence curve

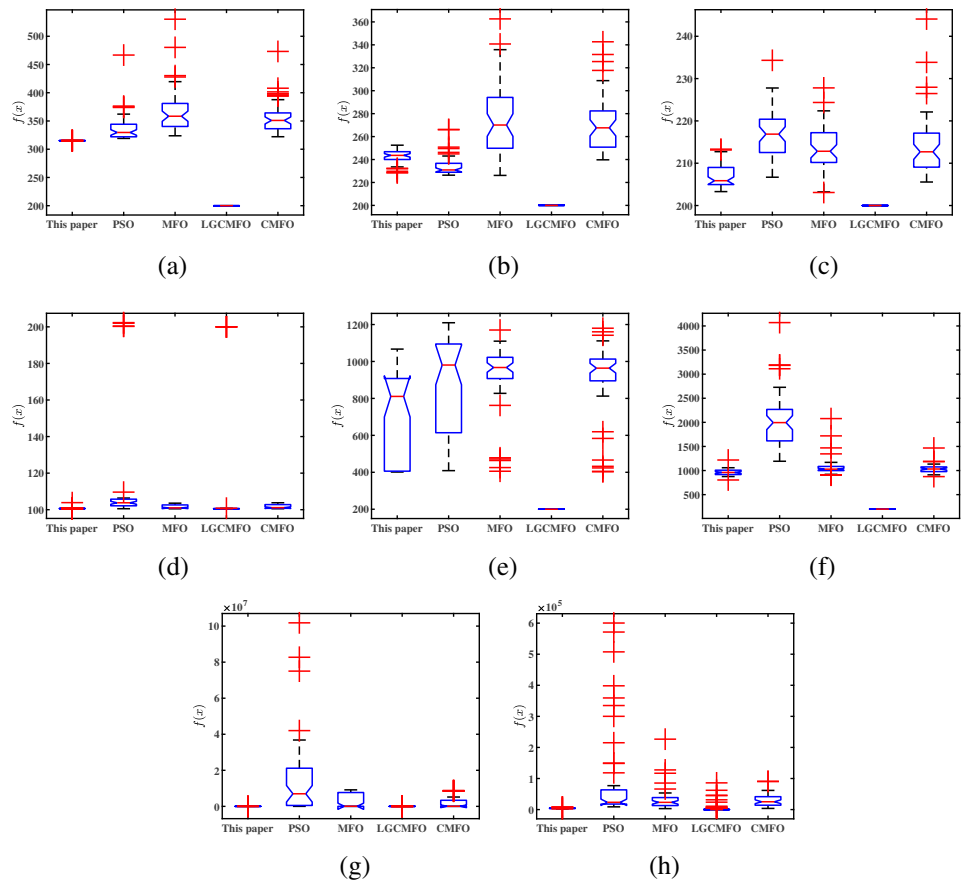


**Fig. 22** 10-dimension composition function box plot

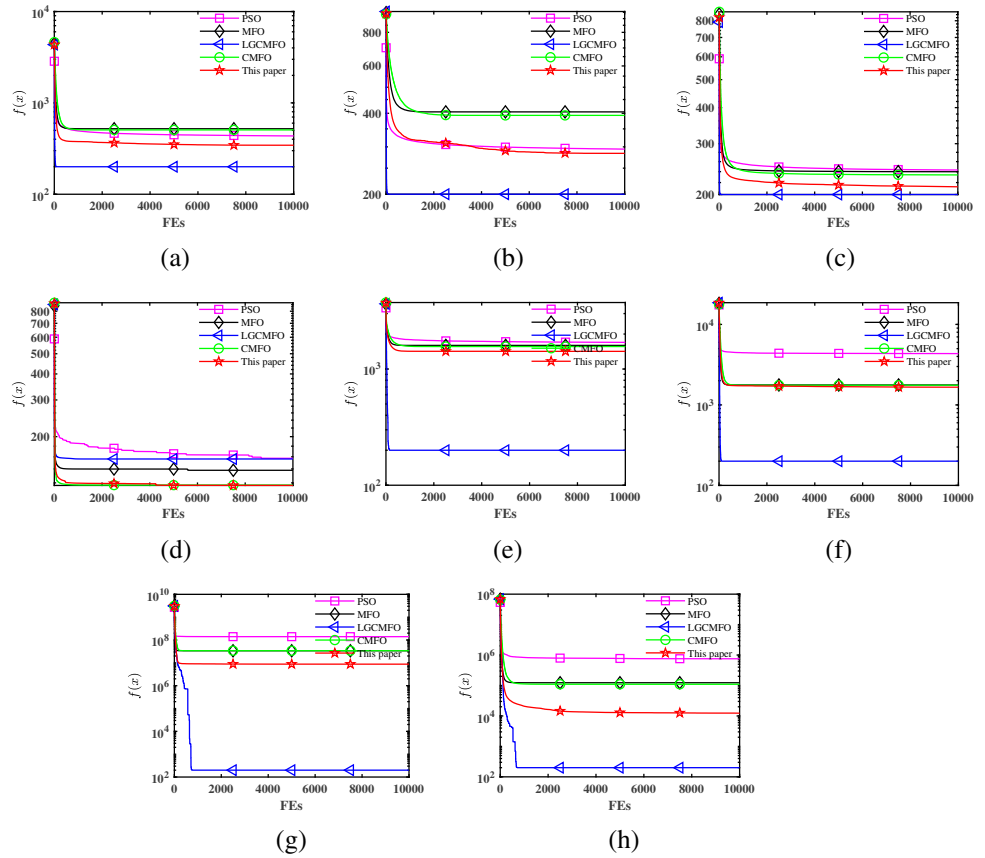
**Fig. 23** 30-dimension composition function convergence curve



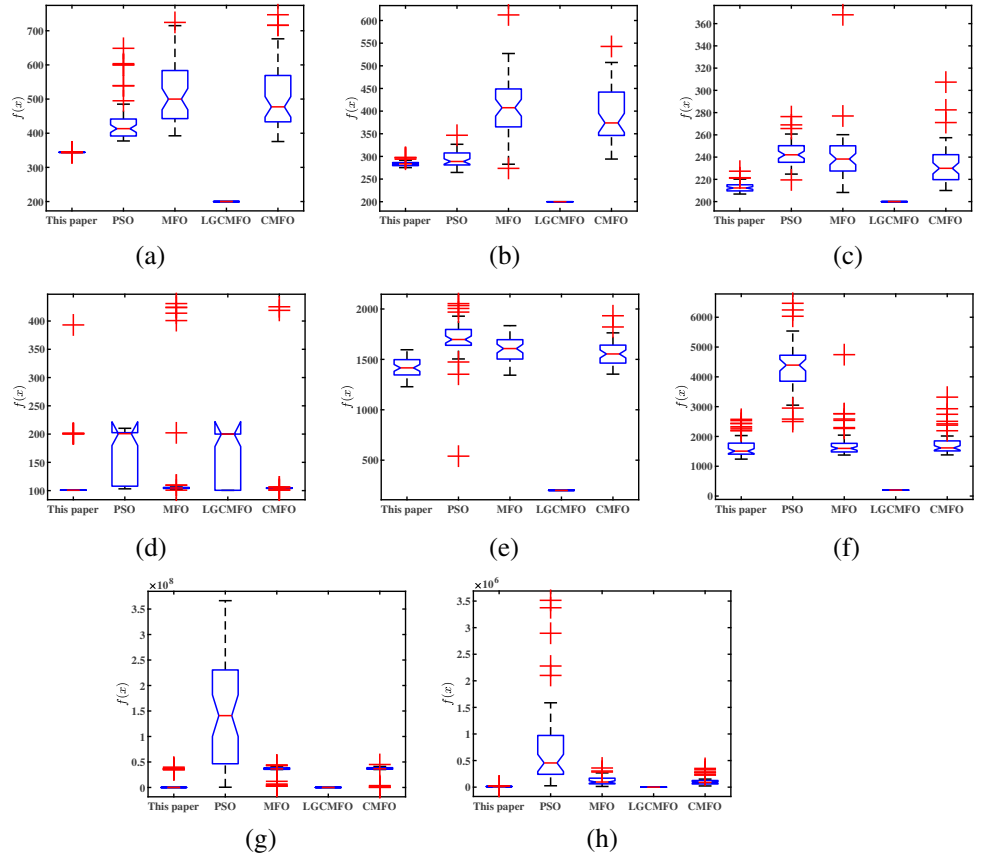
**Fig. 24** 30-dimension composition function box plot



**Fig. 25** 50-dimension composition function convergence curve



**Fig. 26** 50-dimension composition function box plot



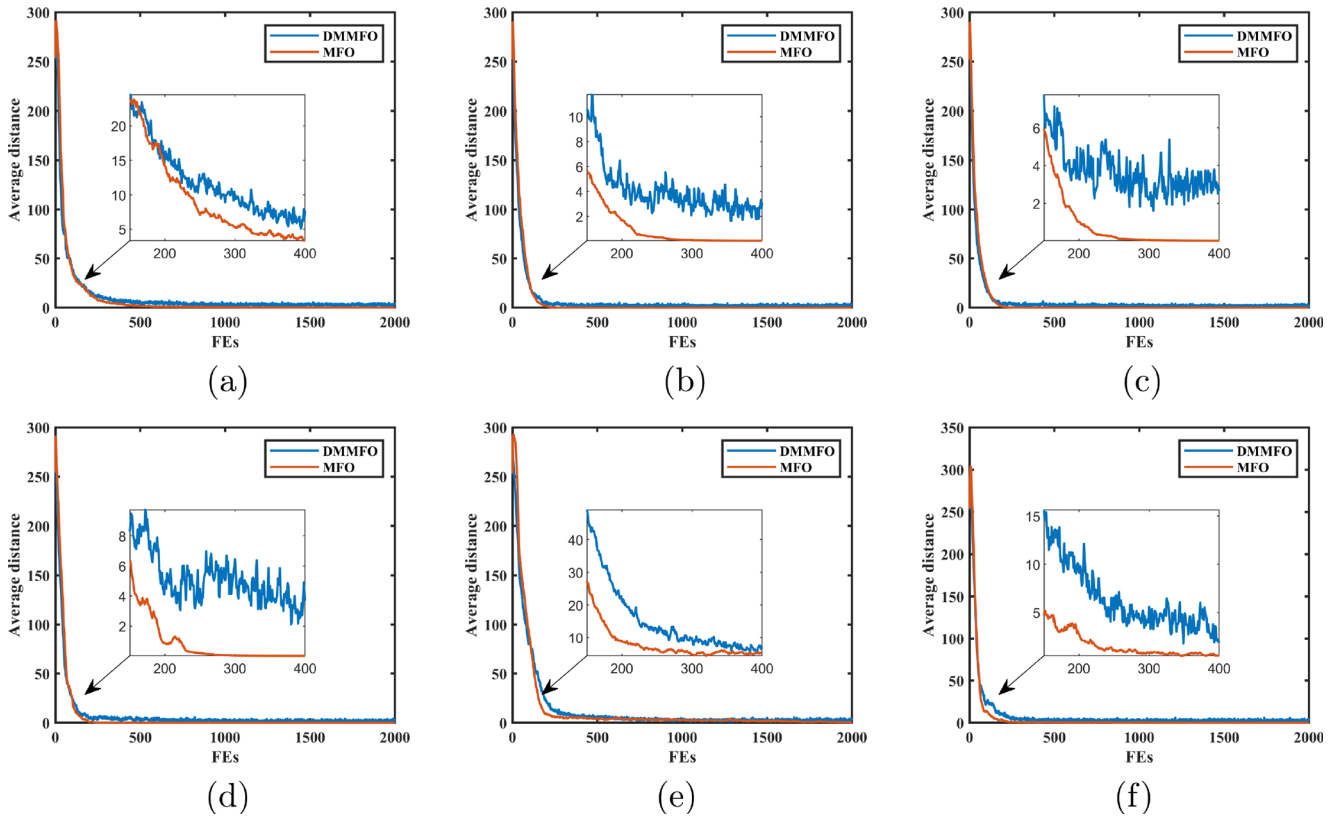


Fig. 27 10-dimension function diversity comparison

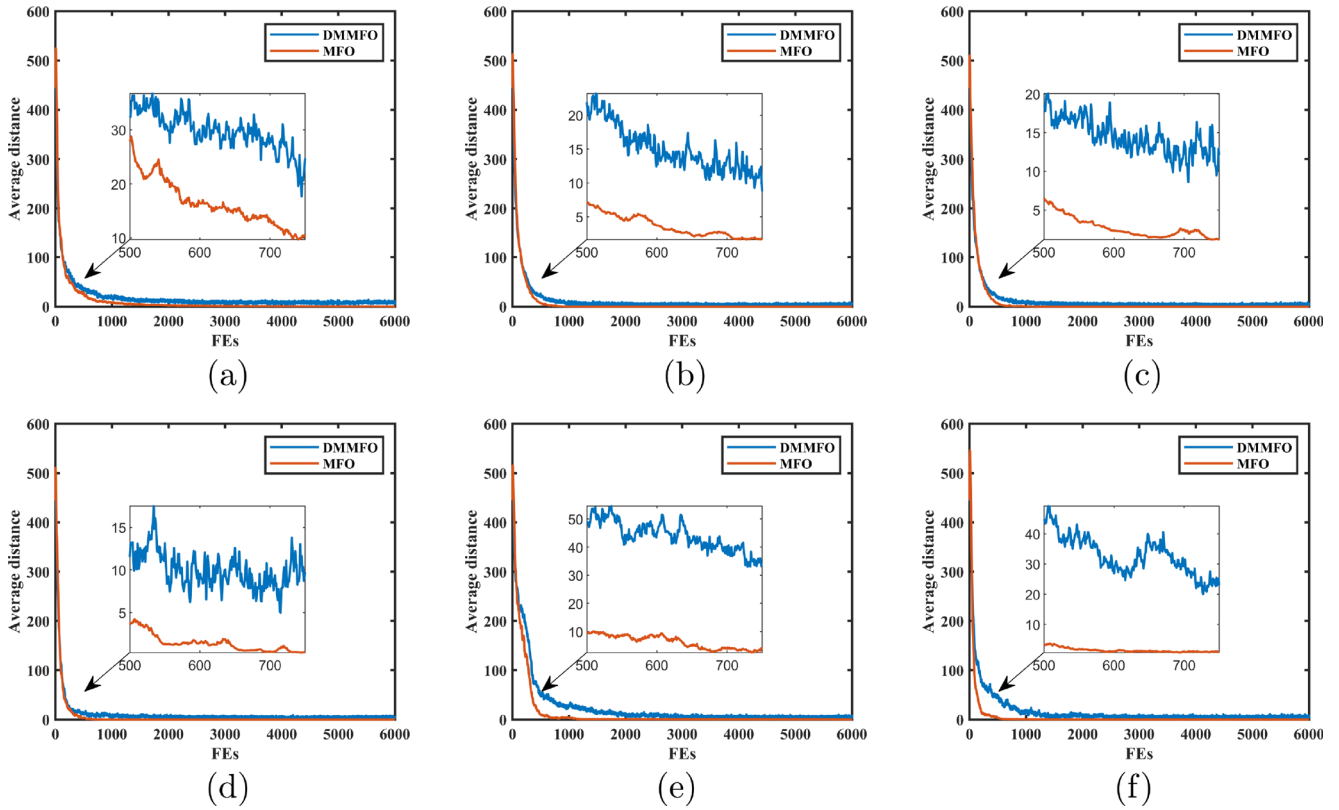


Fig. 28 30-dimension function diversity comparison



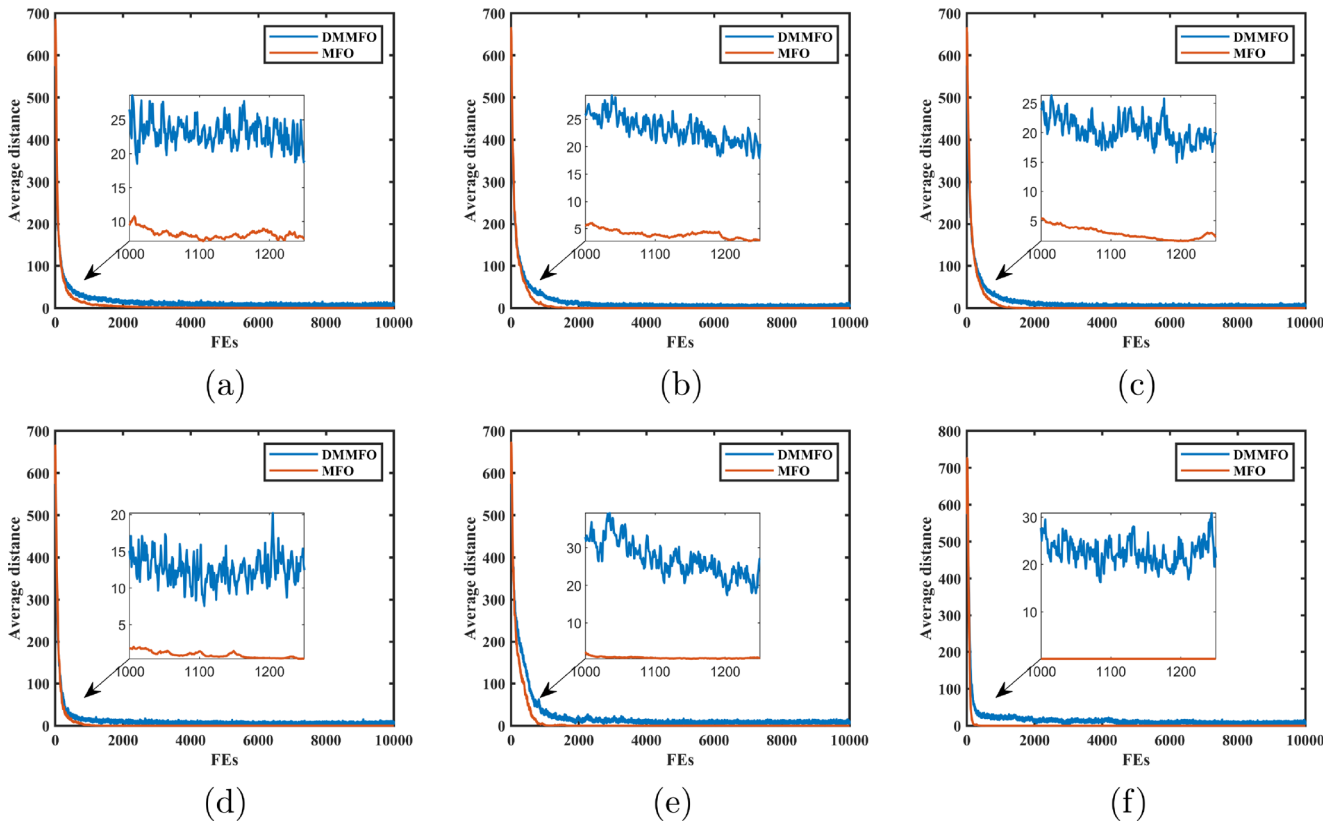


Fig. 29 50-dimension function diversity comparison

methods GOA[54], CS[55], MMA[56], and SOS[57] as presented in Table 19. The optimization model of this problem is formulated as

Minimize:

$$f(x) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5) \tag{33}$$

Subject to:

$$g(x) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \tag{34}$$

Variable range:

$$0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100 \tag{35}$$



Fig. 30 A tension-compression spring schematic design problem

Table 19 presents the comparisons of the results. It is observed that the proposed DMMFO performs significantly better than other compared algorithms.

### 5.4 Problem of three-bar truss design

The three-bar truss problem [58] (Fig. 33) is a structural optimization problem. It is focused on minimizing the weight. There are two parameters that need to be addressed. Due to its difficult constrained search space, this problem

Table 17 Comparison of results for tension/compression spring design problem

Algorithm	Optimal values for variables			Optimum cost
	<i>d</i>	<i>D</i>	<i>P</i>	
This paper	0.051679	0.356485	11.302575	<b>0.012665</b>
PSO	0.517280	0.357644	11.244543	0.012675
ES	0.519890	0.363965	10.890522	0.012681
GA	0.051480	0.351661	11.632201	0.012705
DE	0.051609	0.354714	11.410831	0.012670

The bold entries indicated that the value is optimal in the same row

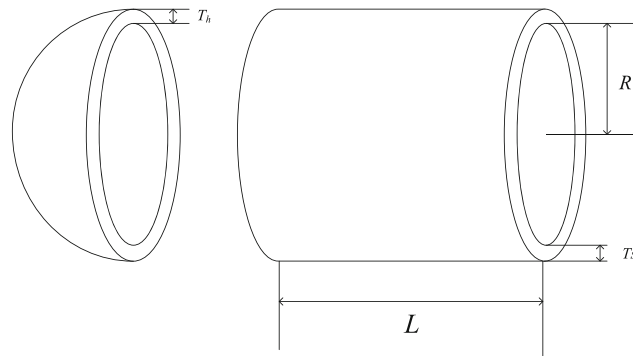


Fig. 31 Pressure vessel design problem

Table 18 Comparison of results for pressure vessel design problem

Algorithm	Optimal values for variables				Optimum cost
	$T_s$	$T_h$	$R$	$L$	
This paper	0.7430	0.3842	40.319619	200.000000	<b>6032.5484</b>
PSO	0.8125	0.4375	42.091266	176.746500	6061.0777
GA	0.9375	0.5000	48.329000	112.679000	6410.3811
DE	0.8125	0.4375	42.098411	176.637690	6059.7340
ACO	0.8125	0.4375	42.103624	176.572656	6059.0888

The bold entries indicated that the value is optimal in the same row

Fig. 32 Cantilever beam design problem

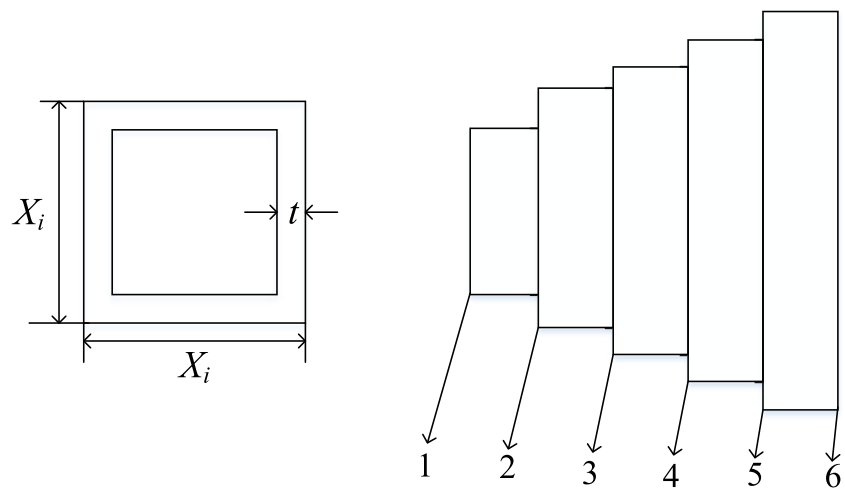


Table 19 Comparison of results for cantilever beam design problem

Algorithm	Optimal values for variables					Optimum cost
	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	
This paper	5.98769	4.86147	4.49969	3.44926	2.14218	<b>1.30667</b>
GOA	6.01167	5.31297	4.48307	3.50279	2.16333	1.33996
CS	6.00890	5.30490	4.50230	3.50770	2.15040	1.33999
MMA	6.01000	5.30000	4.49000	3.49000	2.15000	1.34000
SOS	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996

The bold entries indicated that the value is optimal in the same row

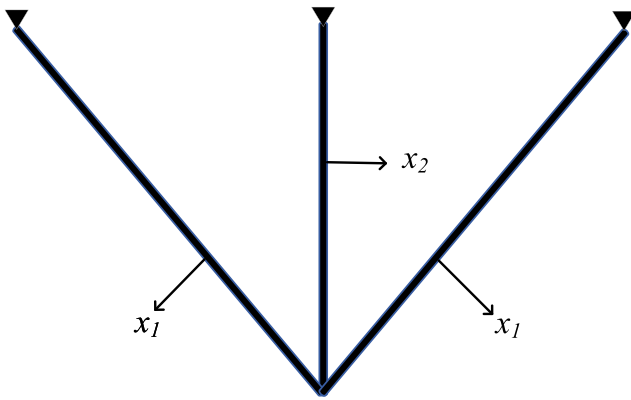


Fig. 33 Three-bar truss design problem

has a wide range of application prospects. The formulation of this problem is

Minimize:

$$f(x) = (2\sqrt{2}x_1 + x_2) \times l \tag{36}$$

Subject to:

$$g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0 \tag{37}$$

$$g_2(x) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}} P - \sigma \leq 0 \tag{38}$$

$$g_3(x) = \frac{1}{\sqrt{2x_2 + x_1}} P - \sigma \leq 0 \tag{39}$$

Variable range:

$$0 < x_1, x_2 \leq 1 \tag{40}$$

The proposed algorithm is used to solve this problem and compare the results with the methods presented in [53, 58], and [59]. As presented in Table 20, the proposed algorithm outperforms the other presented methods.

As a summary, this section investigates the performance of the proposed DMMFO on four practical engineering examples. These examples are some complex constraint problems within the unknown search space. The experimental results show that DMMFO can obtain a better solution

Table 20 Comparison of results for three-bar truss design problem

Algorithm	Optimal values for variables		Optimum cost
	$x_1$	$x_2$	
This paper	0.788687421	0.408213541	<b>263.8958435</b>
MBA [58]	0.788565	0.4085597	263.8958522
CS [53]	0.78867	0.40902	263.9716
Ray [59]	0.795	0.395	264.3

The bold entries indicated that the value is optimal in the same row

than other methods. All these provide a strong proof for the applicability of DMMFO in dealing with real-life problems.

## 6 Conclusions and future work

This paper integrates two efficient mechanisms into the MFO to improve the performance of the algorithm. The global exploration ability of the algorithm is enhanced by embedding an inertia weight of diversity feedback control in a specific position update stage. The optimal solution of the algorithm is obtained by updating and ranking according to the position of the moth, reducing the optimization ability of some moths. To mitigate this issue, the better moths are mutated with a certain probability in the improved algorithm to further ensure the diversity of the population and improve the exploration capacity of the algorithm. CEC'2014 series of functions were used to test the proposed algorithm and comprehensively evaluate it with three indicators, namely, average value, standard deviation, and Wilcoxon rank test. Furthermore, the proposed algorithm was employed to resolve four engineering problems. The experimental results show that the improved algorithm proposed in this work performs better in terms of convergence accuracy and the ability to jump out of the local optimal solutions.

Although the DMMFO showed an acceptable performance in the tested problems it has some limitations. It is still prone to fall into the trap of local minima when it is adopted to deal with the optimization of high-dimensional problems. The duty of moths in DMMFO is to perform a global search. If all the moths are trapped in a large local minima, the underlying reason is that the radius of the moth's spiral motion is not large enough to find a better solution to help the algorithm jump out of local optima. Therefore, DMMFO's exploration ability still needs to be further improved.

There are many complex optimization problems, for instance in practical design tasks, the barrel vault structure design to meet the optimal weights and the marine propeller shape optimization to achieve the least loss. These problems are highly constrained and computationally expensive. The results of this paper prove that DMMFO is a simple and effective method to solve this type of constraint problem within the unknown search space. In addition, apply it to optimize the penalty factor and *gamma* of the support vector machine to improve its adaptability, or to fit the best undetermined parameters from the photovoltaic system to improve the conversion efficiency is worth researching. Moreover, it is also an interesting problem to use DMMFO to optimize the site selection coordinates of the base station on the map to obtain higher coverage. On the other hand, the current study of MFO has focused on the single-objective optimization problem. To find the techniques to apply this

algorithm to multi-objective problems, it can be investigated in future studies.

**Acknowledgments** This work was supported by the National Natural Science Foundation of China (Grant No.11705002), and the Scientific Research Foundation of Education Department of Anhui Province, China (Grant No.KJ2019A0091, Grant No.KJ2019ZD09), the Humanities and Social Science Fund of Ministry of Education of China (Grant NO.19YJAZ H098). The authors would like to thank all the anonymous referees for their valuable comments and suggestions to further improve the quality of this work.

## Compliance with Ethical Standards

**Declaration of competing interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- Pelusi D, Mascella R, Tallini LG, Nayak J, Naik B, Deng Y (2020) An improved moth-flame optimization algorithm with hybrid search phase. *Knowl Based Syst* 191:105277
- Holland JH (1973) Genetic algorithms and the optimal allocation of trials. *SIAM J Comput* 2(2):88–105
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359
- Khodabakhshian A, Hemmati R (2013) Multi-machine power system stabilizer design by using cultural algorithms. *Int J Electr Power Energy Syst* 44(1):571–580
- Kennedy J, Eberhart RC (2002) Particle swarm optimization. *Proceedings of the 1995 IEEE International Conference on Neural Networks* 4:1942–1948
- Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
- Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm. *Adv Eng Softw* 114:163–191
- Dorigo M, Maniezzo V, Colomi A et al (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern Syst* 26(1):29–41
- Yang X, Gandomi AH (2012) Bat algorithm: a novel approach for global engineering optimization. *Eng Comput* 29(5):464–483
- Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. *Appl Math Comput* 214(1):108–132
- Yang X (2010) Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-inspired Computation* 2(2):78–84
- Rashedi E, Nezamabadi-pour H, Saryzdi S (2009) Gsa: A gravitational search algorithm. *Inform Sci* 179(13):2232–2248
- Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm - a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput Struct* 110(1):151–166
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
- Wang C, Wang B, Cen Y, Xie N (2019) Ions motion optimization algorithm based on diversity optimal guidance and opposition-based learning. *Control and Decision*: 1–13
- Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput and Applic* 27(2):495–513
- Kaveh A, Dadras Eslamlou A (2020) Water strider algorithm: A new metaheuristic and applications. *Structures* 25:520–541
- Hayyolalam V, Kazem AAP (2019) Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Eng Appl Artif Intell* 87:103249
- Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH (2020) Marine predators algorithm: A nature-inspired metaheuristic. *Expert Syst Appl* 152:113377
- Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S (2020) Equilibrium optimizer: A novel optimization algorithm. *Knowledge Based Systems* 191
- Faris H, Alzoubi AM, Heidari AA, Aljarah I, Mafarja M, Haseeb MA, Fujita H (2019) An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks. *Information Fusion* 48:67–83
- Faris H, Mafarja M, Heidari AA, Aljarah I, Alzoubi AM, Mirjalili S, Fujita H (2018) An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowl Based Syst* 154:43–67
- Taradeh M, Mafarja M, Heidari AA, Faris H, Aljarah I, Mirjalili S, Fujita H (2019) An evolutionary gravitational search-based feature selection. *Inform Sci* 497:219–239
- Mirjalili S (2015) Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl Based Syst* 89:228–249
- Yıldız BS, Yıldız AR (2017) Moth-flame optimization algorithm to determine optimal machining parameters in manufacturing processes. *Materials Testing* 59(5):425–429
- Allam D, Yousri D, Eteiba M (2016) Parameters extraction of the three diode model for the multi-crystalline solar cell/module using moth-flame optimization algorithm. *Energy Convers Manag* 123:535–548
- Lei X, Fang M, Fujita H (2019) Moth-flame optimization-based algorithm with synthetic dynamic ppi networks for discovering protein complexes. *Knowl Based Syst* 172:76–85
- Xu Y, Chen H, Luo J, Zhang Q, Jiao S, Zhang X (2019) Enhanced moth-flame optimizer with mutation strategy for global optimization. *Inform Sci* 492:181–203
- Chen H, Fu H, Cao Q, Han L, Yan L (2019) Feature selection of parallel binary moth-flame optimization algorithm based on spark. In: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE, pp 408–412
- Wang P, Zhou Y, Luo Q, Fan C, Xiang Z (2019) A complex-valued encoding moth-flame optimization algorithm for global optimization. In: *International Conference on Intelligent Computing*. Springer, pp 719–728
- Emery E, Zawbaa HM (2016) Impact of chaos functions on modern swarm optimizers. *PloS one* 11(7):e0158738
- Xu L, Li Y, Li K, Beng GH, Jiang Z, Wang C, Liu N (2018) Enhanced moth-flame optimization based on cultural learning and Gaussian mutation. *J Bionic Eng* 15(4):751–763
- Huang L, Yang B, Zhang X, Yin L, Yu T, Fang Z (2019) Optimal power tracking of doubly fed induction generator-based wind turbine using swarm moth-flame optimizer. *Trans Inst Meas Control* 41(6):1491–1503
- Sapre S, Mini S (2019) Opposition-based moth flame optimization with cauchy mutation and evolutionary boundary constraint handling for global optimization. *Soft Comput* 23(15):6023–6041

35. Hongwei LI, Liu J, Chen L, Bai J, Sun Y, Kai LU (2019) Chaos-enhanced moth-flame optimization algorithm for global optimization. *J Syst Eng Electron* 30(6):1144–1159
36. Zhang H, Li R, Cai ZN, Gu Z, Chen M (2020) Advanced orthogonal moth flame optimization with broyden-fletcher-goldfarb-shanno algorithm: Framework and real-world problems. *Expert Syst Appl* 159:113617
37. Li C, Niu Z, Song Z, Li B, Fan J, Liu PX (2018) A double evolutionary learning moth-flame optimization for real-parameter global optimization problems. *IEEE Access* 6:76700–76727
38. Khalilpourazari S, Khalilpourazary S (2019) An efficient hybrid algorithm based on water cycle and moth-flame optimization algorithms for solving numerical and constrained engineering optimization problems. *Soft Comput* 23(5):1699–1722
39. Riget J, Vesterstrøm JS (2002) A diversity-guided particle swarm optimizer-the arps, Dept. of Computer Science, University of Aarhus, Denmark 2 (2002)
40. Wang Y (2013) An improved ant colony algorithm based on the search for diversity. *Computer Digital Engineering* 41(6):896–898
41. Yang M, Li C, Cai Z, Guan J (2014) Differential evolution with auto-enhanced population diversity. *IEEE Trans Cybern* 45(2):302–315
42. Fang W, Sun J, Xu W (2008) Diversity-controlled particle swarm optimization algorithm. *Control and Decision* 23(8):863–868
43. Liang J, Qu B, Suganthan P Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory
44. Mirjalili S (2016) Sca: A sine cosine algorithm for solving optimization problems. *Knowl Based Syst* 96:120–133
45. Belegundu AD, Arora JS (1985) A study of mathematical programming methods for structural optimization. Part i: Theory. *Int J Numer Methods Eng* 21(9):1583–1599
46. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng Appl Artif Intell* 20(1):89–99
47. Mezura-Montes E, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *Int J Gen Syst* 37(4):443–473
48. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. *Comput Ind* 41(2):113–127
49. Li L, Huang Z, Liu F, Wu Q (2007) A heuristic particle swarm optimizer for optimization of pin connected structures. *Comput Struct* 85(7-8):340–349
50. Kannan B, Kramer SN (1994) An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Mechanical Design* 116(2):405–411
51. Deb K (1997) Geneas: A robust optimal design technique for mechanical component design. In: *Evolutionary algorithms in engineering applications*. Springer, pp 497–514
52. Kaveh A, Talatahari S (2010) An improved ant colony optimization for constrained engineering design problems. *Eng Comput* 27(1):155–182
53. Gandomi AH, Yang X.-S., Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng Comput* 29(1):17–35
54. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. *Adv Eng Softw* 105:30–47
55. Kaveh A, Mahdavi VR (2014) Colliding bodies optimization: A novel meta-heuristic method. *Computers & Struct* 139:18–27
56. Chickermane H, Gea H (1996) Structural optimization using a new local approximation method. *Int J Numer Methods Eng* 39(5):829–846
57. Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: A new metaheuristic optimization algorithm. *Comput Struct* 139:98–112
58. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Appl Soft Comput* 13(5):2592–2612
59. Ray T, Saini P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng Optim* 33(6):735–748

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Lei Ma** is studying for his M.S. degree in Management science and Engineering College, Anhui University of Technology. His research interests include swarm intelligence optimization algorithms and their applications.



**Chao Wang** received the M.S. degree in mechatronics from Anhui University of Technology. He also earned a PhD degree in solid mechanics from Hohai University. He is currently a Lecturer with the College of Machinery, Anhui University of Technology. His research interests include swarm intelligence optimization algorithms and their applications. He has published more than 10 SCI/EI academic articles as the first author.



**Neng-gang Xie** received the Ph.D. degree in engineering from Hohai University, in 1999. He is currently a Professor and a Doctoral Supervisor with the College of Management Science and Engineering, Anhui University of Technology.





**Miao Shi** is studying for his M.S. degree in Mechanical Engineering College, Anhui University of Technology. He received his B.S. degree in Heifei university in 2018. His research interests in machine learning, Pattern recognition of EEG signals and swarm intelligence optimization algorithm.



**Lu Wang** graduated from from Anhui University of Technology. He is currently an Professor and a Master Supervisor with the college of Mechanical Engineering, Anhui University of Technology. His research interests include modern design methods and simulations. He has published more than 20 SCI/EI academic articles as the first author.



**Ye Ye** received the M.S. degree in mechatronics from Anhui University of Technology, China, in 2008. She is currently an Associate Professor in the college of Mechanical Engineering, Anhui University of Technology. She is mainly engaged in the research of modern design methods and simulations.