Applied Intelligence (2020) 50:4434–4458
https://doi.org/10.1007/s10489-020-01793-2

An improved moth-flame optimization algorithm with orthogonal opposition-based learning and modified position updating mechanism of moths for global optimization problems

Xiaodong Zhao[1] · Yiming Fang[2] · Le Liu[1] · Jianxiong Li[1] · Miao Xu[1]

Published online: 20 July 2020

Abstract

Moth-Flame Optimization (MFO) algorithm is a new population-based meta-heuristic algorithm for solving global optimization problems. Flames generation and spiral search are two key components that affect the performance of MFO. To improve the diversity of flames and the searching ability of moths, an improved Moth-Flame Optimization (IMFO) algorithm is proposed. The main features of the IMFO are: the flames are generated by orthogonal opposition-based learning (OOBL); the modified position updating mechanism of moths with linear search and mutation operator. To evaluate the performance of IMFO, the IMFO algorithm is compared with other 20 algorithms on 23 benchmark functions and IEEE (Institute of Electrical and Electronics Engineers) CEC (Congress on Evolutionary Computation) 2014 benchmark test set. The comparative results show that the IMFO is effective and has good performance in terms of jumping out of local optimum, balancing exploitation ability and exploration ability. Moreover, the IMFO is also used to solve three engineering optimization problems, and it is compared with other well-known algorithms. The comparison results show that the IMFO algorithm can improve the global search ability of MFO and effectively solve the practical engineering optimization problems.

Keywords Moth-Flame optimization algorithm · Orthogonal experiment design · Opposition-based learning · Mutation operator · Engineering optimization problems

1 Introduction

Global optimization is important in various scientific problems and engineering applications. In some literatures, traditional mathematical methods are used to solve global optimization problems in various fields [1–3]. But they meet great challenges in solving complex optimization problems, such as multimodal, discontinuous and non-convex problems. Therefore, the meta-heuristic algorithms are proposed by imitating biological evolution and insect/bird behaviors, and applied to solve complex optimization problems.

In recent decades, the population-based meta-heuristic algorithms have been widely concerned by scholars, which are simple and easy to apply in various fields. Some of the population-based algorithms are proposed, for example, Particle Swarm Optimization (PSO) [4], Bees Algorithm (BA) [5], Artificial Bee Colony (ABC) [6] algorithm, Differential Evolution (DE) [7], Krill Herd (KH) [8], Coyote Optimization Algorithm (COA) [9], Bat Algorithm (BA) [10],Bacterial Foraging Optimization (BFO) [11] algorithm, Fruit Fly Optimization Algorithm (FOA) [12], Sine Cosine Algorithm (SCA) [13], Grey Wolf Optimization (GWO) [14], Flower Pollination Algorithm (FPA) [15], Spotted Hyena Optimizer (SHO) [16], Barnacles Mating Optimizer (BMO) [17], Poor and rich optimization algorithm(PRO) [18], Multi-Verse Optimizer (MVO) [19], Whale Optimization Algorithm (WOA) [20] and Moth-Flame Optimization (MFO) [21] algorithm.

Among the above algorithms, Moth-Flame Optimization algorithm has received special attention because MFO algorithm has good search ability, which is proposed

✉ Yiming Fang
fyming@ysu.edu.cn

[1] Key Lab of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China

[2] Research Center of the Ministry of Education for Intelligent Control System and Intelligent Equipment, Yanshan University, Qinhuangdao 066004, China

by Seyedali Mirjalili based on flight behavior of moths in nature [21]. The MFO algorithm is implemented by updating the position of moths and generating flames. MFO has been widely used because of its simple and easy to use software. For example, in [22], MFO was used to optimize the parameters of least squares support vector machine (LSSVM), and the MFO-LSSVM forecasting model was established. In [23], MFO was utilized to determine the control parameters of blade pitch controllers. In [24], MFO was used to solve the vehicular ad hoc networks clustering problem. In [25], MFO was used to design the antenna array. In [26], MFO was utilized to realize the optimal link cost in wireless router network. In order to improve the control ability of multiarea hybrid interconnected power system, MFO was used to optimize its controller [27]. Puja Singh and Shashi Prakash have used MFO for multiple optical network units placement [28]. In [29], MFO was utilized to address the optimal reactive power dispatch (ORPD) problem.

To solve specific problems, researchers have developed some improved MFO algorithms. For example, in [30], to improve the global searching ability and convergence speed of MFO, the differential evolution was used to generate flames and the mechanism of flames guided moths was modified. To increase the diversity of the population, the levy flight strategy was integrated into MFO [31]. In [32], an Enhanced Moth-Flame Optimization was proposed to improve the performance of MFO in balancing exploitation and exploration. In [33], Opposition-based Moth Flame Optimization (OMFO) was proposed to solve the global optimization problem. In [34], MFO was hybridized with mutation strategy to solve the complex optimization tasks. In [35], a chaotic local search and Gaussian mutation have been introduced into MFO to optimize KELM. In [36], an improved Moth Flame Optimization (IMFO) algorithm was proposed to solve real-world optimization problems. In [37], two spiral search mechanisms were proposed to improve the search ability of MFO.

Although improved MFO algorithms proposed in the above literature enhance the global convergence of MFO algorithm to a certain extent, these enhanced MFO algorithms still face the problem of falling into local optimum in their results analysis. To further alleviate the above problems and to improve MFO's performance, in this paper, an improved MFO algorithm (IMFO) is proposed. The main contributions of this paper are summarized as follows:

(1) The OOBL strategy is used to update the best and worst flames in the iterative process, which can generate effective flames to guide moths, so as to enhance exploration ability of the MFO.

(2) A modified position updating mechanism of moths is proposed by introducing linear search and mutation operator. Linear search and spiral search are integrated to improve convergence efficiency. Meanwhile, Euclidean distance is used to select the search strategy and mutation operators in the iterative process. This mechanism can guarantee the diversity of population under the condition of accelerating convergence speed.

(3) The performance of the IMFO algorithm is evaluated on 23 benchmark functions and IEEE CEC 2014 benchmark test set, which is compared with MFO, LMFO, MFO3, CMFO, IMFO2020, COA, CMAES, SSA, SCA, GSA, ABC, PSO, GWO, WOA, PSOGSA, HCLPSO, MVO, HSCA, EWOA, IDA, SHO. Meanwhile, the IMFO is also used to solve three engineering optimization problems and the results are compared with other well-known algorithms(such as SCA, MMA and GCA).

The rest of the paper is organized as follows: Section 2, introduces Moth-Flame Optimization algorithm, Opposition-Based Learning (OBL) and Orthogonal Experiment Design (OED). In Section 3, the improved Moth-Flame Optimization algorithm is discussed in detail. Section 4 uses the classical benchmark function and IEEE CEC 2014 to estimate the performance of IMFO, and compares with other algorithms. In Section 5, IMFO is used to solve three engineering optimization problems. Section 6 presents the conclusions of this study.

2 Related works

2.1 Moth-flame optimization algorithm

MFO is a population-based optimization algorithm, which is proposed by Seyedali Mirjalili based on flight behavior of moths in nature [21]. In the search mechanism of MFO, flames and moths can exchange information quickly, and balance exploitation and exploration performance. The following is a brief review of MFO.

The moths population is expressed as:

$$M = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_n \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & \ldots & \ldots & m_{1d} \\ m_{21} & m_{22} & \ldots & \ldots & m_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{n1} & m_{n2} & \ldots & \ldots & m_{nd} \end{bmatrix} \tag{1}$$

where the number of moths/flames is n, the variables(dimension) number of per moth individual is d.

At the same time, the fitness values of moths is expressed as:

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ \vdots \\ OM_n \end{bmatrix} \tag{2}$$

where OM_i means the fitness function value of the corresponding moth $M_i = [m_{i1}, m_{i2}, \cdots, m_{id}], i = 1, 2, \cdots, n$, the fitness function is determined according to the actual situation. MFO initializes the population as follows:

$$M_{ij} = lb_i + u_j\,(ub_i - lb_i) \tag{3}$$

where $j = 1, 2, \cdots, d$; u_j is a random constant between 0 and 1; ub_i and lb_i indicate the upper and lower bounds of the i-th variables as follows:

$$ub = [ub_1, ub_2, \ldots, ub_n] \tag{4}$$

$$lb = [lb_1, lb_2, \ldots, lb_n] \tag{5}$$

where the values of ub and lb are determined by the actual situation.

Correspondingly, the flames and their fitness values are expressed as

$$F = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & \cdots\cdots & f_{1d} \\ f_{21} & f_{22} & \cdots\cdots & f_{2d} \\ \vdots & \vdots & \vdots \quad \vdots & \vdots \\ f_{n1} & f_{n2} & \cdots\cdots & f_{nd} \end{bmatrix} \tag{6}$$

$$OF = \begin{bmatrix} OF_1 \\ OF_2 \\ \vdots \\ OF_n \end{bmatrix} \tag{7}$$

The logarithmic spiral is the main form of moths position update in MFO, which is defined as follows:

$$M_i(l+1) = \begin{cases} D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_i(l), & i \le f_{no} \\ D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_{f_{no}}(l), & i > f_{no} \end{cases} \tag{8}$$

where l represents the current number of iterations; $M_i(l+1)$ means the ith moth in $l+1$ iteration; b is a constant, and the value here is 1; the value range of t is set to [r, 1], in which $r = -1 + l * (-1/L)$, the moth can converge to any point around the flame by changing t; L is the maximum number of iterations; D_i represents the distance between the i-th moth and the flame corresponding to the moth. D_i can be calculated as follows:

$$D_i = \begin{cases} |F_i - M_i|, & i \le f_{no} \\ |F_{f_{no}} - M_i|, & i > f_{no} \end{cases} \tag{9}$$

where f_{no} denotes the number of adaptive flames and is calculated:

$$f_{no} = round(n - \frac{l}{L}(n-1)) \tag{10}$$

where $round$ denotes rounding in the nearest direction. This mechanism of adaptive flame number, which can reduce the number of flames adaptively during iteration, can balance the exploration ability and development ability of the MFO. The pseudo code of the MFO are given in Algorithm 1.

Algorithm 1 Moth-Flame Optimization (MFO) Algorithm

```
1: Randomly initialize each individual in moths using (3);
2: Initialize the iteration count $l = 1$ ;
3: while $l < L + 1$
4:     Update $f_{no}$ using (10);
5:     $OM = FitnessFunction(M)$ ;
6:     if $l == 1$
7:         $F = sort(M)$; $OF = sort(OM)$;
8:     else
9:         $F = sort(M(l-1), M(l))$; $OF = sort(OM(l-1), OM(l))$;
10:    end if
11:    for $i = 1 : n$
12:        for $j = j : d$
13:            Update $r$ and $t$ ;calculate $D$ using (9);
14:            Update $M_{ij}$ using (8);
15:        end for
16:    end for
17:    $l = l + 1$;
18: end while
```

2.2 Opposition-based learning

OBL is to evaluate the current point and its reverse point at the same time, and then select the best to use. See Definition 1 for the basic definition of OBL.

Definition 1 (Opposite number) [38] Let $x \in [a, b]$ be a real number. Its opposite, $\breve{x}$, is defined as follow:

$$\breve{x} = a + b - x \tag{11}$$

Extend Definition 1 to high dimensional space.

Definition 2 (Opposite point in the d space) [38] Let $x = (x_1, \ldots, x_d)$ be a point in d-dimensional space and $x_i \in [a_i, b_i], i = 1, 2, \ldots, d$. The opposite of x is defined by $\breve{x} = (\breve{x}_1, \ldots, \breve{x}_d)$ as follow:

$$\breve{x}_i = a_i + b_i - x_i \tag{12}$$

2.3 Orthogonal experiment design

OED is to select some representative points from the comprehensive test for the test. It is widely used in optimization algorithms to improve their performance, such as GA [39], because of its characteristics of uniform dispersion and strong comparability.

An orthogonal table $L_M(Q^K)$ for K factors at Q levels, the number of combinations is M [39]. Table 1 is an example of $L_8(2^7)$.

3 Improved moth-flame optimization algorithm

In this section, the improved Moth-Flame Optimization (IMFO) algorithm is described. The detailed introduction and discussion of the IMFO are as follows:

3.1 Motivation of improving MFO algorithm

Although, the MFO is effective in solving the problem with unknown constrained search space, it sometimes faces the lack of diversity. For MFO, the flames of the next iteration are generated by selecting the best individuals from the current iterative moths and flames. This method can realize the rapid exchange of information among the flames and the moths, but it may also lose the diversity of flames. It is difficult for the moths to jump out of the local optimal state if the flames fall into the local optimal state and are far from the global optimal state. At the same time, the flames adaptive guiding moths position updating mechanism can adaptively reduce the numbers of flames, and improve local search ability to a certain extent. Furthermore, the exploration ability and exploitation ability cannot be well balanced in the early and late stages of the iteration process.

Therefore, in order to improve the above problems in MFO algorithm, IMFO algorithm is proposed. The strategies in the IMFO are indicated in the following two sections.

Table 1 An orthogonal table $L_8(2^7)$

M	K						
	1	2	3	4	5	6	7
	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

3.2 Flames generation by OOBL strategy

In order to overcome the dimension degradation of the reverse solution, the OOBL is proposed, which is made up of OBL and OED. Assuming that Table 1 is used for orthogonal design, the d-dimensional space defines two levels for flame individual $F_i = [f_{i1}, f_{i2}, \cdots, f_{id}]$ and its reverse individual $\breve{F}_i = \left[\breve{f}_{i1}, \breve{f}_{i2}, \cdots, \breve{f}_{id}\right]$. Owing to the fact that individual dimension d is generally larger than factor K, orthogonal tables cannot be used directly. According to literature [39], the individual dimension is divided into K subvectors.

$$\begin{cases} F_{i1} = \left[f_{i1}, f_{i2}, \cdots, f_{i\,r_1}\right] \\ F_{i2} = \left[f_{i\,r_1+1}, f_{i\,r_1+2}, \cdots, f_{i\,r_2}\right] \\ \cdots \\ F_{i\,K} = \left[f_{i\,r_{K-1}}, f_{i\,r_K}, \cdots, f_{i\,d}\right] \end{cases} \tag{13}$$

$$F_i = \left[\ F_{i1}\ \ F_{i2}\ \ldots\ F_{iK}\ \right] \tag{14}$$

where $1 \leq r_1 < r_2 < \cdots < r_K - 1 \leq d$.

To improve the diversity of the population and enhance the exploring ability of the MFO algorithm, the optimal flame and the worst flame are selected to generate the OOBL flames. Among them, the best flame is selected to improve the local search ability, and the worst flame is selected to improve the ability to jump out of the local optimum. The OOBL flames FM is as follows:

$$\begin{aligned} F_{bestnew} &= lb_{best} \cdot ones(1, d) \\ &\quad + (ub_{best} \cdot ones(1, d) - F_{best}) \end{aligned} \tag{15}$$

$$\begin{aligned} F_{worstnew} &= lb_{worst} \cdot ones(1, d) + rand \\ &\quad \cdot (ub_{worst} \cdot ones(1, d) - F_{worst}) \end{aligned} \tag{16}$$

$$FM = [F_{bestnew}; F_{worstnew}] \tag{17}$$

where F_{best} and F_{worst} represent the optimal flame and the worst flame; $F_{bestnew}$ and $F_{worstnew}$ represent OOBL solutions of the optimal flame and the worst flame; $ones(1, d)$ represents a vector of $1 \times d$ and its elements are all 1; $rand$ is a random number in [0,1], which is chosen to increase randomness and the possibility of jumping out of local optimum; ub_{best}, lb_{best}, ub_{worst} and lb_{worst} represent the upper and lower bounds of the optimal flame and the worst flame, respectively.

3.3 Modified position updating mechanism of moths

To further improve the convergence speed and the global search ability of the MFO algorithm, modified position updating mechanism of moths based on hybrid search strategy and mutation operator is proposed. The modified position updating mechanism of moths, which is introduced into IMFO, is divided into two parts through Euclidean distance. When the moth is near the best flame, the spiral and linear search mechanism is used to enhance the local search ability; when the moth is far away from the best flame, the mutation operator is used to enhance the global search ability. The maximum Euclidean distance in search space is as follows:

$$D_E = \sqrt{\sum_{j=1}^{d}(ub_j - lb_j)^2} \tag{18}$$

The Euclidean distance between i-th moth and optimal flame in current iteration is as follows:

$$d_i = \sqrt{\sum_{j=1}^{d}(M_{ij} - F_{bestj})^2} \tag{19}$$

The modified position updating mechanism of moths is elaborated as follows:

(1) In the iterative search process, when $d_i \leq w \cdot D_E$, linear search mechanism is introduced. The position of moths is updated as follows:

$$M_i(l+1) = \begin{cases} F_i - A \cdot D_i', & i \leq f_{no} \\ D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_{f_{no}}(l), & i > f_{no} \end{cases} \tag{20}$$

$$\begin{aligned} D_i' &= |C \cdot F_i - M_i| \\ D_i &= \left|F_{f_{no}} - M_i\right| \end{aligned} \tag{21}$$

where w is the weight coefficient and the value is selected as 0.1 in this paper; $A = 2a \cdot R - a$; $C = 2 \cdot R$; $a = -1 + l * (-1/L)$; R is a random constant in [0,1]. Different places around the flame can be reached with respect to the current position by adjusting the value of A and C.

(2) In the iterative search process, when $d_i > w \cdot D_E$, mutation operator is used to update moth position to improve the diversity of the population. The mutation operator is defined as [40]

$$M_i(l+1) = M_{g_1}(l) + R_m \otimes \left(M_{g_2(l)} - M_{g_3(l)}\right) \tag{22}$$

where M_{g_1}, M_{g_2} and M_{g_3} are randomly selected from M, g_1, g_2 and g_3 are random value that is not equal to i in [1,n]. $R_m = [q_1, q_2, \cdots, q_d]$ is a randomly position vector, $q_j\,(j = 1, 2, \cdots, d)$ is a uniformly distributed random number in [0,1]; $\otimes$ represents hadamard product.

Based on the above explanation, the pseudo code of IMFO algorithm is shown in Algorithm 2 and the steps of IMFO are illustrated as follows:

Step 1 Parameter initialization

Initialize the number of moths/flames n, the maximum number of iterations L; the variables (dimension) number of per moth d and parameters a, t, A, C, w.

Step 2 Initialize the positions of moths using (3).

Step 3 Calculate the number of flames using (10) and the fitness function values of moths.

Step 4 Update flames

The moths of the current iteration, the flames of the previous iteration, and the flames generated by orthogonal opposition-based learning are sorted, and the best n individuals are selected to form the flame population of the current iteration.

Step 5 Update moths

Update the positions of moths using (20)–(22).

Step 6 Orthogonal opposition-based learning

In the search process, the best flame and the worst flame of each iteration are chosen as the base flame, and the orthogonal reverse flames are formed using (15)–(17).

Algorithm 2 Improved Moth-Flame Optimization Algorithm (IMFO)

```
Randomly initialize each individual in moths using (3);
Initialize the iteration count $l = 1$;
while $l < L + 1$
    Update $f_{no}$ using (10);
    $OM = FitnessFunction(M)$;
    If $l == 1$
        $F = sort(M)$; $OF = sort(OM)$;
select the optimum solution $F_{best}$ and $F_{worst}$;
    else
        $F = sort(M(l-1), M(l), FM)$;
$OF = sort(OM(l-1), OM(l), OFM)$;
select the optimum solution $F_{best}$ and $F_{worst}$;
    end If
select the best $n$ individuals to form the flame population;
    for $i = 1 : n$
        If $d_i \leq w \cdot D_E$
            for $j = j : d$
                Update $r, t, A$ and $C$;
calculate $D$ and $D'$ using (21);
                Update $M_{ij}$ using (20);
            end for
        end If
        If $d_i > w \cdot D_E$
                Update $M_{ij}$ using (22);
        end If
    end for
    $FM$ is obtained by OOBL of $F_{best}$ and $F_{worst}$;
    $OFM = FitnessFunction(FM)$;
    Update the best solution $F_{best}$;
    $l = l + 1$;
end while
```

Step 7 Stop or continue the iterative process.

Repeat step 3 to step 6 until the number of iterations l has reached the L or the desired fitness function value is reached.

3.4 Computational complexity of IMFO

The computational complexity of the algorithm represents the resources consumed by the algorithm. The computational complexity of the MFO algorithm depends on

Table 2 Classical benchmark functions

Function	Dim	Range	Global optima
Unimodal benchmark problems			
$\boldsymbol{F1(x)} = \sum_{i=1}^{d} x_i^2$	30	[–100,100]	0
$\boldsymbol{F2(x)} = \sum_{i=1}^{d} \lvert x_i\rvert + \prod_{i=1}^{d} \lvert x_i\rvert$	30	[–10,10]	0
$\boldsymbol{F3(x)} = \sum_{i=1}^{d}(\sum_{j=1}^{i} x_j)^2$	30	[–100,100]	0
$\boldsymbol{F4(x)} = max_i\{\lvert x_i\rvert, 1 \le i \le d\}$	30	[–100,100]	0
$\boldsymbol{F5(x)} = \sum_{i=1}^{d-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[–30,30]	0
$\boldsymbol{F6(x)} = \sum_{i=1}^{d}([x_i + 5])^2$	30	[–100,100]	0
$\boldsymbol{F7(x)} = \sum_{i=1}^{d} i \cdot x_i^4 + rand[0, 1)$	30	[–1.28,1.28]	0
Multi-modal benchmark problems			
$\boldsymbol{F8(x)} = \sum_{i=1}^{d} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	[–500,500]	$-418.9829 \times$ d
$\boldsymbol{F9(x)} = \sum_{i=1}^{d}[x_i^2 - 10\cos(2x_i\pi) + 10]$	30	[–5.12,5.12]	0
$\boldsymbol{F10(x)} = -20exp(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}) - exp[\frac{1}{d}\sum_{i=1}^{d}\cos(2x_i\pi)] + 20 + e$	30	[–32,32]	0
$\boldsymbol{F11(x)} = \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d}\cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[–600,600]	0
$\boldsymbol{F12(x)} = \frac{\pi}{d}\{10\sin(\pi y_1) + \sum_{i=1}^{d-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1} + 1)] + (y_d - 1)^2\} + \sum_{i=1}^{d} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i+1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[–50,50]	0
$\boldsymbol{F13(x)} = 0.1 \times$ $\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{d}(x_i - 1)^2\left[1 + \sin^2(1 + 3\pi x_i)\right]\right.$ $\left.+ (x_d - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right\} +$ $\sum_{i=1}^{d} u(x_i, 5, 100, 4)$	30	[–50,50]	0
Fixed-dimension multimodal benchmark problems			
$\boldsymbol{F14(x)} = \left[0.002 + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i - x_{ij})^6}\right]^{-1}$	2	[–65,65]	1
$\boldsymbol{F15(x)} = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]^2$	4	[–5,5]	0.0003
$\boldsymbol{F16(x)} = 4x_1^2 - 2.1x_1^4 + 0.33x_1^6 10 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	-1.0316
$\boldsymbol{F17(x)} = 10 + 10 \times \left(1 - \frac{0.125}{\pi}\right)\cos(x_1) + \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2$	2	[–5,5]	0.398
$\boldsymbol{F18(x)} =$ $\left[1 + (1 + x_1 + x_2)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right]$ $\times\left[30 + (2x_1 - 3x_2)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$	2	[–2,2]	3
$\boldsymbol{F19(x)} = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	3	[1,3]	-3.86
$\mathbf{F20(x)} = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	6	[0,1]	-3.32
$\boldsymbol{F21(x)} = -\sum_{i=1}^{5}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	[0,10]	-10.1532
$\boldsymbol{F22(x)} = -\sum_{i=1}^{7}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	[0,10]	-10.4028
$\boldsymbol{F23(x)} = -\sum_{i=1}^{10}\left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	4	[0,10]	-10.5363

population initialization, position updating, population individual position evaluation and sorting mechanism. The computational complexity of initial population is $O(n \times D)$; the computational complexity of population position updating is $O(L \times n \times D)$; the complexity of position evaluation is $O(L \times n \times D)$; and the sorting mechanism adopts the fast sorting method, considering the worst case in the sorting process, the computational complexity is $O(L \times n^2)$. The computational complexity of MFO is $O(n \times D) + O(L \times (n^2 + 2n \times D))$. The orthogonal opposition-based learning strategy is used to update the optimal and worst flames, so the complexity of IMFO position updating is $O(L \times (n + 14) \times D)$, and the complexity of positions evaluation is $O(L \times (n + 14) \times D)$. Therefore, the final computational complexity of IMFO is $O(n \times D) + O(L \times (n^2 + 2(n + 14) \times D))$. The computational complexity of IMFO is the same as that of MFO and all of them are $O(n^2)$.

4 Simulation study and discussion

Compared with MFO algorithm, the IMFO algorithm proposed in this paper is more effective in exploring global optimal solutions and exploiting the promising regions in search space. In this section, to verify the performance and advantages of IMFO algorithm, the 23 benchmark functions and the CEC 2014 benchmark set have been taken. All simulation experiments are conducted on a Windows10 Professional system with Intel(R) Core(TM) i3-6100 CPU(3.70GHz), 4.0 GB of RAM and language is MATLAB R2014a.

4.1 Experimental comparisons on classical benchmark set

The 23 classical benchmark problems, which are used to test the performance of the IMFO, are divided to three parts: unimodal problems(F1-F7), multi-modal problems(F8-F13), fixed dimension multi-modal problems(F14-F23). The details described of the classical benchmark set is given in Table 2. In addition, the performance of IMFO algorithm is compared with other algorithms: MFO [21], LMFO [31], MFO3 [37], CMFO [41], IMFO2020 [42], COA [9], CMAES [43], SSA [40], SCA [13], GSA [44], ABC [6], PSO [4], GWO [14], PSOGSA [45], HCLPSO [46], WOA [20], HSCA [47], EWOA [48], IDA [49], SHO [16]. The parameter settings of the 20 optimization algorithms are given in Table 3 and each algorithm runs 30 times independently on each test problem. The results of the 23 benchmark problems are given in Tables 4, 5, 6, 7, 8, 9, 10 and 11.

Table 3 Parameter settings

Method	Year	Population size(n)	Max Iteration(L)	Other
IMFO	-	30	500	$t \in [-1, 1], b = 1, R = rand$
MFO [21]	2015	30	500	$t \in [-1, 1], b = 1$
LMFO [31]	2016	30	500	$t \in [-1, 1], b = 1, \beta = 3/2$
MFO3 [37]	2016	30	500	$t \in [-1, 1], b = 1$
CMFO [41]	2017	30	500	$\mu = 4, t \in [-1, 1], b = 1$
IMFO2020 [42]	2020	30	500	$w_{max} = 0.8, w_{min} = 0.3, t \in [-1, 1], b = 1$
COA [9]	2018	30	500	$N_p = 6, N_c = 5$
CMAES [43]	2001	30	500	$\sigma = 1$
SSA [40]	2017	30	500	$c_2 = rand, c_3 = rand$
SCA [13]	2016	30	500	a=2, $r_2 = 2 \cdot \pi \cdot rand, r_3 = 2 \cdot rand, r_4 = rand$
GSA [44]	2009	30	500	$elitistcheck = 1, rpower = 1$
ABC [6]	2007	30	500	$limit = 100$
PSO [4]	2002	30	500	$c_1 = 2, c_2 = 2, V_{max} = 0.5, V_{min} = -0.5$
GWO [14]	2014	30	500	$r_1 = rand, r_2 = rand$
WOA [20]	2016	30	500	$r_1 = rand, r_2 = rand, p = rand$
MVO [19]	2016	30	500	$r_1 = rand, r_2 = rand, r_3 = rand, r_4 = rand, max = 1, min = 0.2$
PSOGSA [45]	2010	30	500	$elitistcheck = 1; rpower = 1; w = 0.3; c_1 = 0.5; c_2 = 1.5$
HCLPSO [46]	2015	30	500	$w = 0.99 - 0.2; c_1 = 2.5 - 0.5; c_2 = 0.5 - 2.5$
HSCA [47]	2019	30	500	a=2, $r_2 = 2 \cdot \pi \cdot rand, r_3 = 2 \cdot rand, r_4 = rand$
EWOA [48]	2020	30	500	$r_1 = rand, r_2 = rand, p = rand$
IDA [49]	2020	30	500	$r_1 = rand, r_2 = rand, beta = 1.5$
SHO [16]	2017	30	500	$r_1 = rand, r_2 = rand$

Table 4 Comparison results of various mechanisms on the classic benchmark functions

	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Function	IMFO		MFO		MLMFO		OOBLMFO	
F1	0.000E+00	0.000E+00	2.672E+03	7.072E+03	1.376E-69	2.918E-68	1.211E-05	5.553E-05
F2	0.000E+00	0.000E+00	3.878E+01	3.531E+01	2.717E-38	3.092E-37	2.740E-05	3.239E-05
F3	7.022E-24	5.477E-23	2.509E+04	2.271E+04	1.587E-24	2.755E-23	9.097E+03	1.583E+04
F4	0.000E+00	0.000E+00	7.266E+01	1.763E+01	3.295E-33	6.882E-32	1.641E+00	2.273E+00
F5	2.602E+01	4.736E-01	2.690E+06	5.662E+07	2.635E+01	1.203E+00	1.131E+03	5.342E+03
F6	6.052E-05	1.326E-04	3.360E+03	1.414E+04	6.785E-03	1.339E-01	5.071E-02	1.335E-01
F7	8.629E-05	1.494E-04	6.942E+00	2.865E+01	3.067E-04	8.300E-04	5.181E-02	4.017E-02
F8	-1.086E+04	1.829E+03	-8.244E+03	1.478E+03	-9.847E+03	2.513E+03	-9.209E+03	2.188E+03
F9	0.000E+00	0.000E+00	1.909E+02	8.186E+01	0.000E+00	0.000E+00	6.121E+01	5.828E+01
F10	8.882E-16	0.000E+00	1.951E+01	1.118E+00	2.191E-15	2.512E-15	4.329E-04	1.115E-03
F11	0.000E+00	0.000E+00	4.004E+01	1.272E+02	0.000E+00	0.000E+00	7.105E-02	8.989E-02
F12	1.084E-06	1.452E-06	1.150E+02	2.199E+03	2.775E-04	4.073E-03	4.107E-01	2.712E+00
F13	3.911E-02	8.368E-02	1.786E+05	3.789E+06	4.105E-02	1.076E-01	6.322E-01	4.583E+00
F14	1.395E+00	1.403E+00	2.976E+00	3.487E+00	4.107E+00	7.551E+00	2.117E+00	4.176E+00
F15	4.612E-04	2.650E-04	1.474E-03	5.339E-03	6.457E-04	2.903E-04	1.139E-03	6.448E-04
F16	-1.032E+00	5.282E-12	-1.032E+00	0.000E+00	-1.032E+00	4.786E-10	-1.032E+00	0.000E+00
F17	3.979E-01	2.759E-14	3.979E-01	0.000E+00	3.979E-01	1.423E-08	3.979E-01	0.000E+00
F18	3.000E+00	7.229E-15	3.000E+00	6.908E-15	3.000E+00	1.739E-04	3.000E+00	6.908E-15
F19	-3.863E+00	0.000E+00	-3.863E+00	0.000E+00	-3.861E+00	2.588E-02	-3.863E+00	0.000E+00
F20	-3.294E+00	8.407E-02	-3.202E+00	1.304E-01	-3.251E+00	1.683E-01	-3.232E+00	1.664E-01
F21	-1.015E+01	0.000E+00	-4.880E+00	5.319E+00	-1.015E+01	7.851E-03	-5.055E+00	0.000E+00
F22	-1.040E+01	0.000E+00	-6.608E+00	6.057E+00	-9.926E+00	5.400E+00	-5.042E+00	9.640E-01
F23	-1.054E+01	0.000E+00	-6.011E+00	6.136E+00	-9.634E+00	5.738E+00	-5.128E+00	0.000E+00
Function	LMFO		MMFO		OOBLLMFO		OOBLMMFO	
F1	2.674E+03	1.414E+04	2.672E+03	7.072E+03	1.116E-05	4.432E-05	1.594E-08	5.315E-08
F2	4.418E+01	4.239E+01	3.878E+01	3.531E+01	4.094E-05	1.048E-04	1.568E-07	4.973E-07
F3	2.791E+04	2.437E+04	2.509E+04	2.271E+04	1.351E+04	2.186E+04	4.623E+02	2.370E+03
F4	7.156E+01	1.718E+01	7.266E+01	1.763E+01	1.401E+00	2.595E+00	1.730E-01	4.853E-01
F5	5.365E+06	5.659E+07	2.690E+06	5.662E+07	1.100E+03	3.332E+03	1.397E+02	3.320E+02
F6	3.351E+03	1.428E+04	3.360E+03	1.414E+04	3.610E-02	7.343E-02	3.971E-04	2.447E-03
F7	6.330E+00	3.037E+01	6.942E+00	2.865E+01	5.182E-02	6.131E-02	2.985E-02	4.871E-02
F8	-8.186E+03	2.525E+03	-8.244E+03	1.478E+03	-9.043E+03	2.633E+03	-8.424E+03	2.402E+03
F9	1.777E+02	7.054E+01	1.909E+02	8.186E+01	5.727E+01	7.437E+01	3.188E-11	3.546E-10
F10	1.837E+01	5.583E+00	1.951E+01	1.118E+00	3.598E-04	5.024E-04	7.866E-06	3.810E-05
F11	4.304E+01	1.909E+02	4.004E+01	1.272E+02	5.478E-02	1.315E-01	4.541E-02	1.398E-01
F12	6.022E+03	1.174E+05	1.150E+02	2.199E+03	2.170E-01	1.126E+00	1.385E-02	7.331E-02
F13	1.042E+02	1.476E+03	1.786E+05	3.789E+06	7.717E-01	5.297E+00	7.725E-02	6.431E-01
F14	2.905E+00	7.581E+00	2.976E+00	3.487E+00	2.647E+00	3.487E+00	1.988E+00	3.487E+00
F15	2.124E-03	1.385E-02	1.474E-03	5.339E-03	1.104E-03	6.338E-04	1.261E-03	5.379E-03
F16	-1.032E+00	0.000E+00	-1.032E+00	0.000E+00	-1.032E+00	0.000E+00	-1.032E+00	0.000E+00
F17	3.979E-01	0.000E+00	3.979E-01	0.000E+00	3.979E-01	0.000E+00	3.979E-01	0.000E+00
F18	3.000E+00	0.000E+00	3.000E+00	6.908E-15	3.000E+00	6.908E-15	3.000E+00	6.908E-15
F19	-3.863E+00	0.000E+00	-3.863E+00	0.000E+00	-3.863E+00	0.000E+00	-3.863E+00	0.000E+00
F20	-3.204E+00	1.304E-01	-3.202E+00	1.304E-01	-3.186E+00	3.618E-01	-3.219E+00	8.811E-02
F21	-5.887E+00	5.319E+00	-4.880E+00	5.319E+00	-5.055E+00	0.000E+00	-6.245E+00	3.605E+00
F22	-6.769E+00	5.410E+00	-6.608E+00	6.057E+00	-4.951E+00	9.640E-01	-4.906E+00	9.640E-01
F23	-6.865E+00	6.265E+00	-6.011E+00	6.136E+00	-5.128E+00	0.000E+00	-6.759E+00	4.738E+00

Table 5 Test results of eight variant algorithms

	IMFO	MFO	MLMFO	OOBLMFO
Rank	1	8	2	4
1/0/-1	~	19/4/0	11/10/2	18/5/0
Ave	1.56522	6.73913	3.30435	4.26087
	LMFO	MMFO	OOBLLMFO	OOBLMMFO
Rank	6	7	5	3
1/0/-1	19/4/0	19/4/0	19/4/0	17/6/0
Ave	5.43478	5.73913	4.95652	4.00000

4.1.1 The influence of improving mechanism on MFO

In this section, the different mechanisms of the proposed method are simulated and analyzed. Among them, MLMFO means to improve MFO by mutation operator and linear update mechanism; OOBLMFO means to improve MFO by orthogonal opposition-based learning; LMFO means to improve MFO by linear update mechanism; MMFO means to improve MFO by mutation operator; OOBLLMFO means to improve MFO by orthogonal opposition-based learning and linear update mechanism; OOBLMMFO represents the improvement of MFO with orthogonal opposition-based learning and mutation operator. In the simulation of this section, the number of population, dimension and number of iterations are 30, 30, 500 respectively, and each algorithm runs 30 times independently on each test function. The simulation results are given in Table 4.

As can be seen from Table 4, MFO variants are superior to MFO algorithm on most test functions in Mean and Std indexes. In order to intuitively analyze the convergence performance of the algorithm, Fig. 1 shows the convergence graphs of eight algorithms on six test functions. It can be seen from the figure that the IMFO algorithm has advantages in convergence accuracy and convergence speed. In conclusion, the IMFO selected in this paper is the best of the above variant algorithms.

In order to further analyze the performance of IMFO, Wilson sign rank test [50] is used to detect IMFO and other variants. The test results are given in Table 5 and are indicated by ‘1/0 /-1’. It can be seen from Table 5 that, in 23 test functions, IMFO is superior to MFO in 19 functions and has the same performance with MFO in 4 functions. Accordingly, IMFO has good results compared with other variants. At the same time, Friedman test [34] is used to sort the above eight algorithms. It can be seen in Table 5 that the average ranking (Ave) of IMFO is the best.

To sum up, IMFO shows the best performance among all variants.

4.1.2 Analysis on classical benchmark set

To verify the performance of IMFO, the IMFO algorithm and MFO algorithm are simulated and analyzed on 23 classical test problems in this subsection. The simulation results show that in Tables 6–9, the Best, Worst, Mean and Stds of each algorithm running independently for 30 times are given. On F1-F13, this section selects three dimensions (10, 30, and 50) to evaluate the performance of IMFO and MFO. Other parameters in the simulation are selected as shown in Table 3.

In the unimodal problems (F1-F7), there is only one global optimal solution and no local optimal solution. Consequently, F1-F7 are used to estimate the convergence rate and exploitation ability of the IMFO. The simulation

Table 6 Comparisons of MFO and IMFO algorithms on unimodal and multi-modal problems with dimension 10

dim=10	MFO				IMFO			
	Best	Mean	Worst	Std	Best	Mean	Worst	Std
F1	1.066E-14	2.629E-13	1.512E-12	3.619E-13	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F2	2.753E-10	5.950E-09	3.402E-08	8.598E-09	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F3	6.850E-04	6.667E+02	1.000E+04	2.171E+03	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F4	3.366E-03	1.517E+00	1.129E+01	2.573E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F5	6.386E+00	6.911E+00	7.687E+00	2.693E-01	5.423E+00	5.836E+00	6.832E+00	2.707E-01
F6	4.171E-15	4.407E-13	3.634E-12	7.887E-13	7.470E-18	2.114E-14	1.850E-13	4.839E-14
F7	1.418E-03	9.346E-03	2.019E-02	5.601E-03	3.050E-05	2.951E-04	1.779E-03	3.349E-04
F8	-4.071E+03	-3.253E+03	-2.545E+03	3.529E+02	-4.190E+03	-3.962E+03	-2.768E+03	3.544E+02
F9	5.970E+00	2.568E+01	5.771E+01	1.581E+01	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F10	1.901E-08	1.409E-01	2.580E+00	5.499E-01	8.880E-16	8.880E-16	8.880E-16	1.004E-31
F11	9.857E-03	1.812E-01	6.028E-01	1.323E-01	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F12	1.438E-16	1.037E-01	2.179E+00	4.032E-01	1.680E-18	2.347E-11	4.920E-10	9.332E-11
F13	3.421E-16	3.662E-03	1.099E-02	5.268E-03	1.200E-17	5.635E-08	1.690E-06	3.138E-07

Table 7 Comparisons of MFO and IMFO algorithms on unimodal and multi-modal problems with dimension 30

dim=30	MFO				IMFO			
	Best	Mean	Worst	Std	Best	Mean	Worst	Std
F1	5.175E-01	1.339E+03	2.000E+04	4.341E+03	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F2	1.008E+01	3.873E+01	8.001E+01	1.941E+01	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F3	3.157E+03	2.067E+04	4.581E+04	1.076E+04	0.000E+00	3.076E-19	8.980E-18	1.638E-18
F4	4.525E+01	6.940E+01	8.429E+01	9.495E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F5	2.118E+02	2.680E+06	7.995E+07	1.459E+07	2.576E+01	2.624E+01	2.691E+01	2.981E-01
F6	8.678E-01	2.678E+03	3.014E+04	6.420E+03	9.000E-06	1.778E-04	2.434E-03	4.371E-04
F7	6.659E-02	3.466E+00	2.443E+01	6.173E+00	4.020E-06	2.552E-04	1.232E-03	2.925E-04
F8	-1.013E+04	-8.671E+03	-6.713E+03	7.944E+02	-1.235E+04	-1.063E+04	-7.984E+03	1.123E+03
F9	1.005E+02	1.579E+02	2.592E+02	3.897E+01	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F10	1.332E+00	1.481E+01	1.996E+01	7.111E+00	8.880E-16	8.880E-16	8.880E-16	1.003E-31
F11	5.942E-01	1.896E+01	9.107E+01	3.662E+01	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F12	1.739E+00	6.400E+00	1.887E+01	4.383E+00	1.540E-07	1.248E-05	3.209E-04	5.828E-05
F13	1.887E+00	1.765E+03	5.029E+04	9.168E+03	3.600E-05	9.970E-02	6.017E-01	1.166E-01

Table 8 Comparisons of MFO and IMFO algorithms on unimodal and multi-modal problems with dimension 50

dim=50	MFO				IMFO			
	Best	Mean	Worst	Std	Best	Mean	Worst	Std
F1	2.538E+02	7.151E+03	3.367E+04	9.178E+03	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F2	2.594E+01	7.523E+01	1.136E+02	2.754E+01	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F3	3.562E+04	7.029E+04	1.175E+05	2.315E+04	0.000E+00	7.505E-17	1.430E-15	2.910E-16
F4	7.266E+01	8.361E+01	9.307E+01	4.660E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F5	1.682E+05	2.017E+07	8.270E+07	3.413E+07	4.590E+01	4.657E+01	4.853E+01	7.558E-01
F6	3.379E+02	7.594E+03	3.072E+04	8.368E+03	4.050E-04	2.595E-03	1.178E-02	2.451E-03
F7	7.151E-01	2.553E+01	7.529E+01	2.369E+01	1.120E-05	2.807E-04	1.713E-03	3.302E-04
F8	-1.584E+04	-1.346E+04	-1.137E+04	1.267E+03	-2.073E+04	-1.723E+04	-1.071E+04	2.240E+03
F9	2.361E+02	3.199E+02	4.421E+02	4.903E+01	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F10	1.648E+01	1.950E+01	1.997E+01	7.492E-01	8.880E-16	1.125E-15	4.440E-15	9.012E-16
F11	5.545E+00	1.003E+02	2.764E+02	8.185E+01	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F12	7.920E+03	9.503E+06	2.560E+08	4.660E+07	7.150E-06	4.485E-05	9.330E-05	2.449E-05
F13	8.895E+04	4.453E+07	4.102E+08	1.241E+08	2.169E-02	4.414E-01	1.304E+00	3.030E-01

Table 9 Comparisons of MFO and IMFO algorithms on fixed dimension multi-modal problems

	MFO				IMFO			
	Best	Mean	Worst	Std	Best	Mean	Worst	Std
F14	9.9800E-01	1.7572E+00	5.9288E+00	1.2894E+00	9.9800E-01	9.9802E-01	9.9830E-01	7.6115E-05
F15	6.9871E-04	1.6941E-03	2.0363E-02	3.5521E-03	3.0754E-04	5.8037E-04	1.2274E-03	2.0887E-04
F16	-1.03163	-1.03163	-1.03163	0	-1.03163	-1.03163	-1.03163	0
F17	0.39789	0.39789	0.39789	1.1292E-16	0.39789	0.39789	0.39789	1.12987E-16
F18	3.00000	3.00000	3.00000	0	3.00000	3.00000	3.00000	0
F19	-3.86278	-3.86278	-3.86278	0	-3.86278	-3.85971	-3.80033	0
F20	-3.32200	-3.23611	-3.13270	0.05858	-3.32200	-3.27041	-3.20130	0.06025
F21	-10.15320	-7.71967	-2.63047	3.12652	-10.15320	-10.15320	-10.15320	0
F22	-10.40294	-6.65437	-2.75193	3.41028	-10.40294	-10.40294	-10.40293	0
F23	-10.53641	-8.19712	-2.42173	3.41381	-10.53641	-10.53641	-10.53639	0

Table 10 Comparisons of IMFO and other algorithms on classical benchmark functions

	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Function	IMFO		MFO		LMFO		MFO3	
F1	0.000E+00	0.000E+00	1.676E+03	3.787E+03	4.210E-92	2.222E-91	1.350E+03	3.452E+03
F2	0.000E+00	0.000E+00	2.579E+01	1.983E+01	1.972E-52	1.002E-51	3.314E+01	2.402E+01
F3	3.075E-19	1.638E-18	2.335E+04	1.307E+04	4.807E-90	1.949E-89	1.920E+04	1.051E+04
F4	0.000E+00	0.000E+00	6.888E+01	7.449E+00	1.338E-47	7.311E-47	6.726E+01	8.206E+00
F5	2.624E+01	2.981E-01	5.362E+06	2.030E+07	2.749E+01	3.493E-01	2.690E+06	1.459E+07
F6	1.778E-04	4.371E-04	1.027E+03	3.055E+03	2.388E+00	3.257E-01	2.664E+03	4.486E+03
F7	2.552E-04	2.925E-04	4.466E+00	6.597E+00	1.680E-04	1.510E-04	3.693E+00	4.838E+00
F8	-1.063E+04	1.123E+03	-8.602E+03	8.913E+02	-6.538E+03	8.959E+02	-8.702E+03	8.799E+02
F9	0.000E+00	0.000E+00	1.465E+02	2.923E+01	0.000E+00	0.000E+00	1.578E+02	4.111E+01
F10	8.882E-16	1.003E-31	1.669E+01	5.610E+00	8.882E-16	1.003E-31	1.681E+01	5.066E+00
F11	0.000E+00	0.000E+00	1.895E+01	4.350E+01	0.000E+00	0.000E+00	9.992E+00	2.744E+01
F12	1.248E-05	5.828E-05	8.534E+06	4.674E+07	1.864E-01	4.638E-02	1.011E+01	7.656E+00
F13	9.970E-02	1.166E-01	4.637E+01	1.035E+02	2.256E+00	3.144E-01	2.734E+07	1.040E+08
F14	9.980E-01	7.611E-05	1.558E+00	1.412E+00	4.723E+00	4.246E+00	1.985E+00	1.894E+00
F15	5.804E-04	2.116E-04	1.163E-03	5.207E-04	5.156E-04	3.030E-04	1.623E-03	3.558E-03
F16	-1.032E+00	4.791E-11	-1.032E+00	0.000E+00	-1.032E+00	2.049E-05	-1.032E+00	0.000E+00
F17	3.979E-01	6.861E-13	3.979E-01	1.129E-16	3.985E-01	5.548E-04	3.979E-01	1.129E-16
F18	3.000E+00	4.930E-15	3.000E+00	4.474E-15	3.000E+00	3.974E-04	3.000E+00	3.979E-15
F19	-3.860E+00	1.184E-02	-3.863E+00	2.710E-15	-3.858E+00	3.040E-03	-3.863E+00	1.439E-03
F20	-3.270E+00	6.000E-02	-3.208E+00	8.488E-02	-3.181E+00	4.887E-02	-3.222E+00	9.079E-02
F21	-1.015E+01	4.966E-12	-6.474E+00	3.398E+00	-4.945E+00	6.340E-02	-8.312E+00	2.938E+00
F22	-1.040E+01	1.271E-06	-8.143E+00	3.523E+00	-4.993E+00	4.701E-02	-6.833E+00	3.471E+00
F23	-1.027E+01	1.482E+00	-7.642E+00	3.683E+00	-5.075E+00	2.238E-01	-6.590E+00	3.825E+00
Function	CMFO		IMFO2020		COA		CMAES	
F1	7.262E+03	1.393E+04	1.221E-125	1.501E-152	3.344E+01	1.426E+01	5.821E-18	4.237E-18
F2	5.079E+01	1.413E+01	1.961E-66	6.083E-71	8.279E-01	4.312E-02	6.744E+00	6.906E-08
F3	3.503E+04	1.670E+04	2.652E-126	9.651E-147	7.469E+03	4.024E+03	5.133E+03	6.130E+02
F4	8.143E+01	5.593E+00	2.039E-61	1.769E-76	2.676E+01	3.540E+00	2.018E-04	4.400E-05
F5	2.200E+07	5.659E+07	2.841E+01	2.693E-02	4.932E+03	6.903E+02	5.881E+01	1.182E+02
F6	5.242E+03	7.080E+00	4.653E+00	1.950E-01	3.028E+01	2.944E+00	4.599E-18	1.525E-18
F7	1.126E+01	1.519E+01	5.436E-04	4.453E-05	1.387E-01	1.652E-02	1.385E-01	6.126E-02
F8	-8.612E+03	8.136E+02	-3.656E+03	3.476E+02	-1.210E+04	5.838E+01	-2.316E+03	1.294E+01
F9	2.268E+02	6.788E+01	0.000E+00	0.000E+00	2.420E+01	6.559E+00	9.263E+01	9.850E+00
F10	1.990E+01	2.554E-02	8.882E-16	0.000E+00	7.379E+00	8.154E-01	1.799E+01	3.470E-02
F11	5.500E+01	6.375E+01	0.000E+00	0.000E+00	1.336E+00	1.520E-01	4.315E-04	1.054E-08
F12	3.200E+07	1.165E+00	5.047E-01	2.184E-02	3.890E+00	3.221E+00	1.096E-04	1.272E-04
F13	2.051E+07	3.608E+02	2.596E+00	1.726E-01	2.901E+01	7.239E+00	1.210E-03	5.807E-05
F14	2.329E+00	2.092E+00	2.466E+00	2.054E-03	1.121E+00	0.000E+00	1.333E+01	2.656E+00
F15	2.999E-03	3.115E-04	4.068E-04	5.963E-05	1.031E-03	6.475E-04	5.587E-03	1.403E-02
F16	-1.032E+00	0.000E+00	-1.032E+00	3.093E-05	-1.032E+00	0.000E+00	-1.032E+00	0.000E+00
F17	3.979E-01	0.000E+00	3.992E-01	3.202E-04	3.979E-01	0.000E+00	3.979E-01	0.000E+00
F18	3.000E+00	0.000E+00	3.000E+00	7.159E-05	3.000E+00	0.000E+00	5.700E+00	6.908E-15
F19	-3.863E+00	0.000E+00	-3.858E+00	2.824E-03	-3.863E+00	0.000E+00	-3.863E+00	0.000E+00
F20	-3.209E+00	4.629E-02	-3.160E+00	6.776E-02	-3.316E+00	3.005E-12	-3.322E+00	0.000E+00
F21	-5.545E+00	1.715E+00	-5.714E+00	3.469E-01	-9.713E+00	7.161E-14	-4.795E+00	5.282E+00
F22	-5.896E+00	5.400E+00	-5.552E+00	2.930E-01	-1.040E+01	0.000E+00	-9.162E+00	0.000E+00
F23	-6.405E+00	4.562E-02	-5.785E+00	1.675E+00	-9.829E+00	0.000E+00	-9.355E+00	0.000E+00

Table 10 (continued)

	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Function	SSA		GSA		ABC		PSO	
F1	1.514E-07	1.535E-07	6.572E-04	3.539E-03	7.045E-05	8.830E-05	5.684E-01	2.376E-01
F2	2.339E+00	1.551E+00	3.874E-02	1.322E-01	4.945E-03	2.254E-03	2.188E+00	9.217E-01
F3	1.192E+03	6.212E+02	9.716E+02	4.107E+02	1.956E+04	3.548E+03	1.729E+03	1.696E+03
F4	1.167E+01	4.122E+00	7.555E+00	2.252E+00	5.180E+01	6.728E+00	3.828E+00	1.299E+00
F5	1.659E+02	2.447E+02	6.928E+01	6.703E+01	5.876E+01	6.893E+01	1.364E+02	9.869E+01
F6	2.512E-07	4.885E-07	2.063E-02	1.111E-01	7.639E-05	1.143E-04	6.509E-01	3.331E-01
F7	1.854E-01	7.908E-02	8.577E-02	5.091E-02	3.349E-01	7.994E-02	6.563E-01	9.707E-02
F8	-7.732E+03	6.304E+02	-2.591E+03	4.473E+02	-1.145E+04	2.078E+02	-3.670E+02	2.302E+02
F9	5.161E+01	1.704E+01	3.071E+01	5.873E+00	6.659E+00	2.254E+00	4.234E+01	1.299E+01
F10	2.439E+00	5.718E-01	6.429E-02	2.401E-01	2.296E-01	1.954E-01	5.812E+00	1.195E+00
F11	1.463E-02	1.328E-02	3.009E+01	5.607E+00	2.462E-02	2.940E-02	2.837E+02	1.999E+01
F12	7.226E+00	4.081E+00	1.792E+00	1.145E+00	2.640E-05	6.900E-05	2.286E+00	1.176E+00
F13	1.813E+01	1.499E+01	9.444E+00	5.659E+00	3.600E-04	7.206E-04	1.574E+01	8.365E+00
F14	1.097E+00	3.033E-01	4.749E+00	3.372E+00	9.980E-01	3.390E-16	4.304E+02	3.455E+01
F15	1.541E-03	3.567E-03	4.980E-03	3.917E-03	9.025E-04	3.291E-04	5.740E-02	2.091E-01
F16	-1.032E+00	3.640E-14	-1.032E+00	0.000E+00	-1.032E+00	0.000E+00	-1.024E+00	1.856E-02
F17	3.979E-01	2.023E-14	3.979E-01	1.130E-16	3.979E-01	1.130E-16	3.862E+00	3.574E+00
F18	3.000E+00	1.313E-13	3.000E+00	5.342E-15	3.002E+00	8.765E-03	3.000E+00	4.856E-12
F19	-3.863E+00	1.258E-11	-3.863E+00	2.712E-15	-3.863E+00	3.173E-15	-3.861E+00	2.857E-03
F20	-3.214E+00	5.007E-02	-3.322E+00	1.356E-15	-3.322E+00	2.332E-15	-2.598E+00	9.581E-01
F21	-7.219E+00	3.301E+00	-5.904E+00	3.665E+00	-1.015E+01	7.423E-03	-5.048E+00	3.095E+00
F22	-8.664E+00	2.983E+00	-9.570E+00	1.911E+00	-1.037E+01	1.852E-01	-6.391E+00	3.524E+00
F23	-8.478E+00	3.480E+00	-1.054E+01	9.039E-15	-1.048E+01	2.286E-01	-4.714E+00	3.133E+00
Function	GWO		SCA		PSOGSA		HCLPSO	
F1	2.500E-27	6.265E-27	1.661E+01	3.372E+01	1.543E+03	7.562E+01	3.613E-02	5.768E-02
F2	9.922E-17	1.107E-16	1.222E-02	1.381E-02	2.122E+01	6.017E+00	4.727E-02	3.362E-02
F3	7.854E-06	1.537E-05	9.968E+03	5.242E+03	1.652E+04	3.695E+03	1.005E+03	7.876E+02
F4	7.444E-07	7.013E-07	3.572E+01	1.255E+01	6.355E+01	2.815E+01	4.903E+00	1.757E+00
F5	2.682E+01	6.338E-01	4.463E+04	1.004E+05	4.002E+06	3.810E+02	1.029E+02	1.122E+02
F6	8.010E-01	3.202E-01	1.972E+01	2.138E+01	2.270E+03	2.410E+01	3.767E-02	3.806E-02
F7	1.756E-03	1.058E-03	8.266E-02	1.111E-01	1.782E-01	1.071E-02	3.510E-02	1.682E-02
F8	-5.904E+03	8.329E+02	-3.764E+03	2.196E+02	-7.323E+03	1.315E+03	-8.420E+03	4.836E+02
F9	2.274E+00	3.055E+00	3.103E+01	3.016E+01	1.460E+02	6.121E+01	3.120E+01	2.007E+00
F10	9.644E-14	1.604E-14	1.325E+01	8.911E+00	1.625E+01	4.304E+00	3.178E-01	9.229E-01
F11	4.220E-03	7.450E-03	1.380E+00	1.797E+00	3.529E+01	9.347E-02	7.769E-02	5.345E-02
F12	5.137E-02	2.518E-02	2.608E+04	8.763E+04	1.784E+01	3.806E+00	1.505E-01	2.802E-03
F13	6.838E-01	2.987E-01	1.177E+05	4.653E+05	1.025E+07	5.426E+01	3.550E-01	2.023E-01
F14	3.503E+00	3.495E+00	1.926E+00	1.004E+00	6.035E+00	0.000E+00	9.980E-01	0.000E+00
F15	5.910E-03	9.080E-03	1.080E-03	3.565E-04	5.532E-03	1.386E-02	5.740E-04	3.751E-05
F16	-1.032E+00	2.533E-08	-1.032E+00	7.243E-05	-1.032E+00	0.000E+00	-1.032E+00	0.000E+00
F17	3.979E-01	5.415E-07	3.992E-01	1.149E-03	3.979E-01	0.000E+00	3.979E-01	0.000E+00
F18	3.000E+00	9.036E-05	3.000E+00	1.594E-04	5.025E+00	0.000E+00	3.000E+00	0.000E+00
F19	-3.862E+00	1.620E-03	-3.854E+00	2.640E-03	-3.863E+00	0.000E+00	-3.863E+00	0.000E+00
F20	-3.261E+00	8.066E-02	-2.928E+00	2.926E-01	-3.268E+00	8.407E-02	-3.322E+00	3.282E-08
F21	-9.728E+00	1.608E+00	-2.504E+00	1.840E+00	-6.832E+00	3.573E+00	-1.015E+01	0.000E+00
F22	-9.771E+00	1.922E+00	-3.199E+00	1.727E+00	-5.337E+00	9.931E-01	-1.040E+01	0.000E+00
F23	-1.053E+01	9.259E-04	-4.268E+00	1.217E+00	-4.788E+00	5.466E+00	-1.054E+01	0.000E+00

Table 10 (continued)

	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Function	WOA		HSCA		EWOA		IDA	
F1	1.620E-74	4.452E-76	1.129E-23	4.309E-26	3.401E-211	0.000E+00	4.986E-03	2.353E-05
F2	2.387E-49	6.612E-48	2.684E-15	2.613E-16	2.333E-143	8.601E-150	7.446E-02	1.006E-02
F3	4.291E+04	2.251E+04	2.097E+00	3.579E-02	1.006E+05	3.279E+04	1.428E+00	2.082E-01
F4	4.191E+01	4.264E+01	2.915E-01	2.837E-03	2.523E+01	2.766E+00	1.241E+01	4.429E+00
F5	2.785E+01	2.686E-01	2.713E+01	2.061E-02	4.547E-01	2.434E-01	4.364E+04	5.476E+04
F6	3.688E-01	2.569E-01	1.862E+00	9.990E-01	5.911E-02	1.418E-02	3.158E+02	1.806E+02
F7	3.671E-03	4.757E-04	3.339E-03	2.139E-03	1.616E-03	2.301E-03	2.951E-01	1.017E-01
F8	-1.018E+04	1.916E+03	-6.420E+03	3.042E+01	-1.257E+04	1.685E-02	-8.605E+03	1.953E+03
F9	0.000E+00	0.000E+00	7.260E+00	4.920E+00	2.842E-15	4.019E-14	5.583E+01	5.301E+01
F10	4.352E-15	5.024E-15	1.467E+01	1.430E+01	1.243E-15	0.000E+00	6.248E+00	5.422E-01
F11	5.715E-03	0.000E+00	5.856E-03	0.000E+00	2.539E-02	0.000E+00	3.366E+00	4.465E-01
F12	2.799E-02	2.469E-02	1.388E-01	5.282E-02	1.017E-02	4.521E-03	3.569E+01	1.550E+01
F13	5.834E-01	1.004E-02	1.557E+00	1.552E-01	6.321E-02	1.104E-01	5.342E+03	3.964E+02
F14	3.036E+00	1.569E-08	1.546E+00	3.704E-05	2.049E+00	1.181E-04	9.980E-01	0.000E+00
F15	8.088E-04	5.168E-04	6.157E-04	1.392E-05	3.180E-03	1.040E-03	8.930E-04	3.467E-05
F16	-1.032E+00	1.248E-10	-1.032E+00	4.288E-09	-1.032E+00	6.817E-07	-1.032E+00	0.000E+00
F17	3.979E-01	3.044E-06	3.979E-01	1.233E-06	4.018E-01	2.845E-07	3.979E-01	0.000E+00
F18	3.675E+00	2.749E-05	3.000E+00	5.336E-06	9.542E+00	2.732E-06	3.000E+00	6.908E-15
F19	-3.854E+00	1.618E-02	-3.857E+00	5.570E-03	-3.688E+00	4.161E-02	-3.862E+00	1.125E-05
F20	-3.241E+00	1.019E-01	-2.979E+00	8.263E-02	-2.879E+00	7.899E-01	-3.227E+00	1.449E-01
F21	-8.026E+00	5.273E+00	-6.804E+00	3.573E+00	-9.680E+00	2.659E-01	-9.002E+00	3.603E+00
F22	-7.356E+00	3.556E+00	-8.956E+00	8.884E-04	-9.486E+00	4.417E-01	-9.464E+00	2.396E-05
F23	-7.371E+00	4.222E-03	-8.484E+00	3.822E+00	-9.959E+00	2.524E-01	-9.581E+00	3.039E-07

results of IMFO algorithm and MFO algorithm are given in Tables 6–8. In these tables, it can be observed that IMFO algorithm is better than MFO algorithm in the Best, Worst, Mean and Std. Therefore, the test results from F1 to F7 show that the development ability and convergence rate of IMFO in search space are effective.

There are many local optimum solutions for multi-modal problems (F8-F13), which are used to estimate the exploration ability of the IMFO. The comparison results for F8 to F13 are given in Tables 6–8. For F9 and F11, IMFO can search for the best solution with good stability. And for F8, F10, F12 and F13, the comparison results of IMFO is better than MFO in Mean and Std indexes. Therefore, the results show that the exploration ability of IMFO is better than that of MFO.

Table 11 Test results of the 20 algorithms

	Rank	1/0/-1	Ave		Rank	1/0/-1	Ave
IMFO	1	~	4.08696	ABC	2	15/3/5	7.04348
MFO	15	20/3/0	12.95652	PSO	19	20/3/0	14.91304
LMFO	8	16/5/2	9.52174	GWO	4	19/4/0	8.00000
MFO3	16	20/3/0	13.21739	SCA	20	21/2/0	14.91304
CMFO	17	21/2/0	14.82609	PSOGSA	18	21/2/0	14.82609
IMFO2020	9	18/4/1	9.52174	HCLPSO	3	19/4/0	7.04348
COA	5	18/4/1	8.95652	WOA	11	18/5/0	9.91304
CMAES	12	19/3/1	10.26087	HSCA	10	22/1/0	9.69565
SSA	13	18/4/1	10.43478	EWOA	7	17/4/2	9.30435
GSA	6	17/1/5	9.26087	IDA	14	20/3/0	11.30435

Fixed dimension multi-modal problems(F14-F23) have fewer dimensions and fewer local optima. They are used to detect the balance between the exploration ability and exploitation ability of the IMFO. In Table 9, for F14, F15, F21,F22, F23 IMFO is superior to MFO. For F20, the mean value of IMFO is not as good as MFO, but the Std of IMFO is small. For F16, F17, F18, F19, IMFO and MFO have similar accuracy.

Therefore, by analyzing the comparison results in Tables 6–9, it can be concluded that IMFO algorithm is effective in searching global optimal solution.

To better illustrate the convergence performance of IMFO algorithm, the convergence curve in dimension 30 is drawn as shown in Fig. 2. The Fig. 2 are convergence curves of unimodal problems, multi-modal problems, fixed dimension multi-modal problems, respectively. Among them, the ordinates in the graph represent the mean of objective function value of the algorithm running

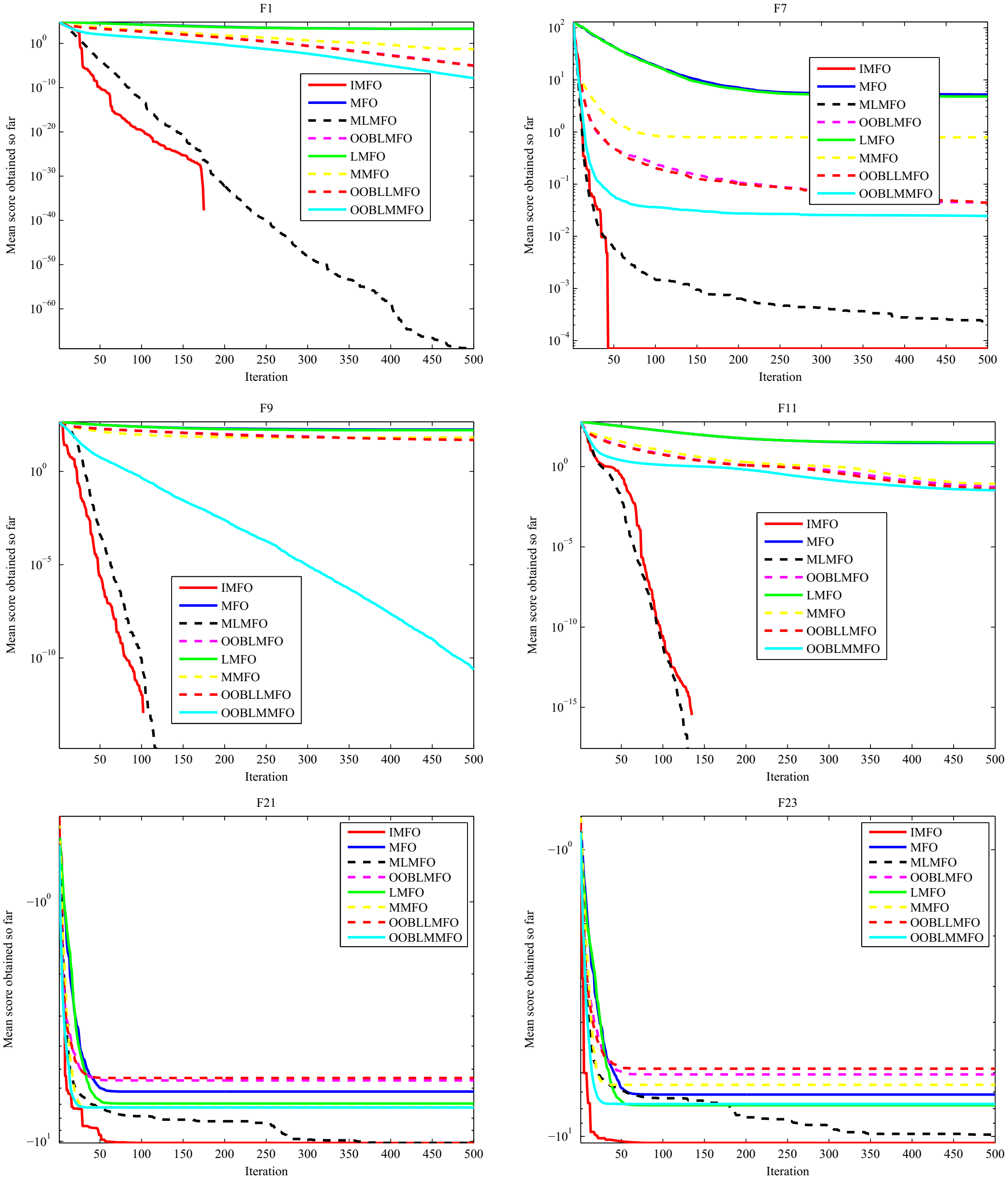

Fig. 1 Convergence corves on 6 functions

independently for 30 times, and the abscissa represents the iteration times. From Fig. 2, it can be seen that the convergence speed and accuracy of IMFO are better than those of MFO in most test functions.

4.1.3 Comparison with other algorithms

To further evaluate the performance of the IMFO, 20 optimization algorithms have been selected and compared with IMFO. The parameter settings of these algorithms used in this paper are shown in Table 3. Table 10 gives the comparative results between IMFO algorithm and MFO, LMFO, MFO3, CMFO, IMFO2020, COA, CMAES, SSA, SCA, GSA, ABC, PSO, GWO, PSOGSA, HCLPSO, WOA, HSCA, EWOA, IDA algorithms. In the table, each algorithm still runs 30 times independently in each classical test problem.

For F1, F2, F4, F9, F11, the IMFO can obtain the best solution. For F3, the IMFO can get better results than other algorithms. Among other test problems, IMFO has better

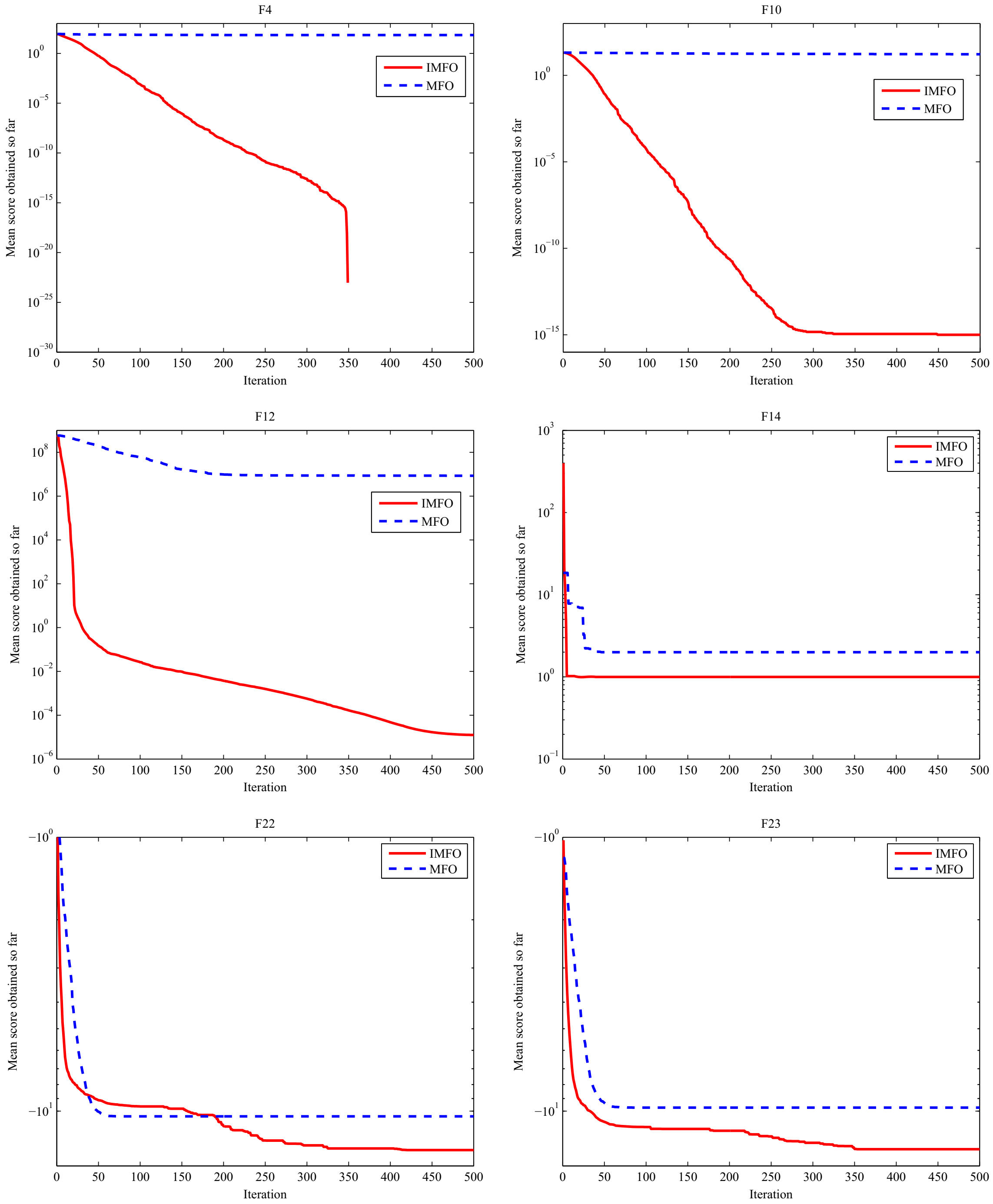

Fig. 2 Convergence corves on 6 functions

simulation results in most problems. At the same time, Fig. 3 shows the convergence curve of 20 algorithm in 6 test problems. It can be seen in this figure that IMFO has better convergence accuracy and speed on these six test problems compared with other algorithms.

Meanwhile, Wilcoxon signed rank test and Friedman test are introduced to analyze the proposed algorithm. In Table 11, the results of Wilcoxon signed rank test and Friedman test are given. In Table 11, the results of IMFO are better than that of other 19 algorithms in most test problems. It can be seen in Table 11 that the average ranking (Ave) of IMFO is the best.

Therefore, compared with other algorithms, IMFO algorithm has good competitiveness and better performance in convergence speed, search accuracy, robustness and jumping out of local optimum.

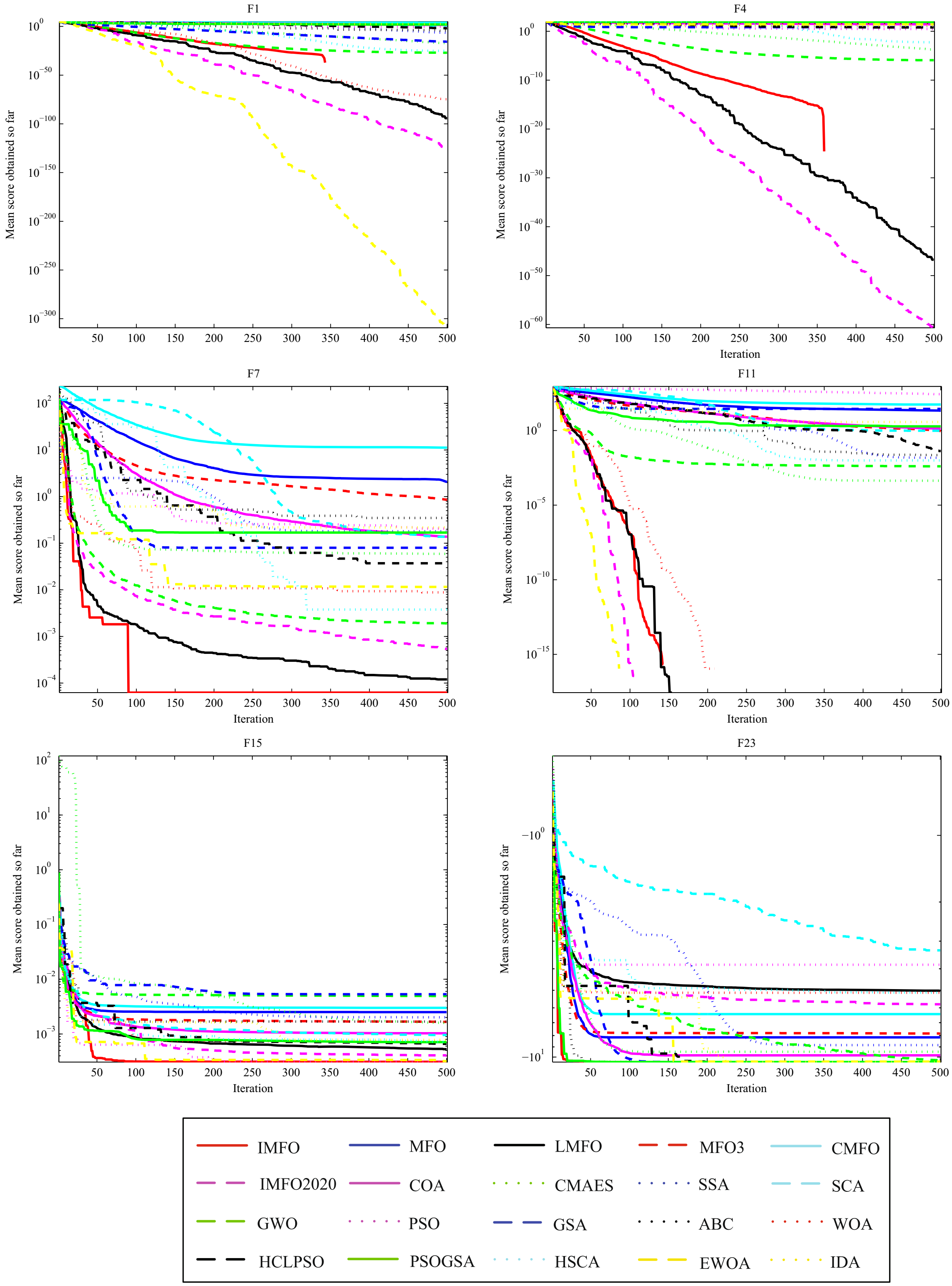

Fig. 3 Convergence corves on 6 functions

4.2 Experimental comparisons on CEC 2014

To further test the performance of IMFO algorithm, the CEC 2014 test set [51] is the second set of test problems to estimate the effectiveness of the IMFO. The CEC 2014 test problems are more difficult than the classical test problems. It consists of four groups: (1) Unimodal (f1-f3); (2) Multimodal (f4-f16); (3) hybrid

Table 12 Comparisons of MFO and IMFO algorithms on CEC 2014 test problems

	MFO				IMFO			
	Best	Mean	Worst	Std	Best	Mean	Worst	Std
f1	1.5067E+07	1.0980E+08	3.9422E+08	1.0064E+08	2.4337E+05	5.5861E+05	1.0091E+06	2.3509E+05
f2	7.4989E+09	1.9241E+10	3.8039E+10	8.5826E+09	3.1231E+02	6.6115E+03	8.9588E+03	1.9861E+03
f3	5.4870E+04	1.2820E+05	2.6121E+05	4.5480E+04	3.3057E+02	4.4013E+03	7.3543E+03	2.1651E+03
f4	9.0317E+02	1.8381E+03	4.3985E+03	9.1619E+02	4.0020E+02	4.4811E+02	4.7482E+02	3.3039E+01
f5	5.2016E+02	5.2036E+02	5.2064E+02	1.1357E-01	5.2000E+02	5.2000E+02	5.2002E+02	3.7384E-03
f6	6.2529E+02	6.3301E+02	6.3941E+02	3.7467E+00	6.2119E+02	6.2472E+02	6.3463E+02	3.0036E+00
f7	7.5678E+02	8.0996E+02	8.9579E+02	3.9328E+01	7.0000E+02	7.0003E+02	7.0009E+02	2.7976E-02
f8	9.2214E+02	9.5556E+02	1.0192E+03	2.7433E+01	8.8457E+02	9.1140E+02	9.4725E+02	1.7507E+01
f9	1.0936E+03	1.1473E+03	1.2558E+03	4.2945E+01	9.7507E+02	9.9344E+02	1.0701E+03	2.4103E+01
f10	3.9215E+03	4.9171E+03	6.0614E+03	6.0284E+02	2.9613E+03	3.9729E+03	4.8383E+03	5.1280E+02
f11	4.7729E+03	5.6681E+03	7.2360E+03	6.5860E+02	4.3274E+03	5.0562E+03	5.7532E+03	3.9282E+02
f12	1.2005E+03	1.2011E+03	1.2016E+03	3.3339E-01	1.2003E+03	1.2006E+03	1.2019E+03	3.2778E-01
f13	1.3007E+03	1.3021E+03	1.3044E+03	1.2051E+00	1.3003E+03	1.3005E+03	1.3006E+03	9.4485E-02
f14	1.4171E+03	1.4406E+03	1.4933E+03	2.1108E+01	1.4002E+03	1.4003E+03	1.4003E+03	4.3289E-02
f15	1.9934E+03	1.9957E+05	1.2431E+06	3.3070E+05	1.5256E+03	1.6332E+03	1.7892E+03	7.9868E+01
f16	1.6125E+03	1.6129E+03	1.6136E+03	2.7606E-01	1.6107E+03	1.6120E+03	1.6127E+03	4.2765E-01
f17	2.0648E+05	3.4252E+06	3.5141E+07	6.4536E+06	4.6121E+04	1.9435E+05	3.9038E+05	1.1436E+05
f18	5.2410E+03	2.5432E+07	5.0308E+08	1.0178E+08	2.0865E+03	3.5419E+03	5.8966E+03	1.2476E+03
f19	1.9166E+03	1.9960E+03	2.1961E+03	6.3791E+01	1.9098E+03	1.9136E+03	1.9169E+03	1.7814E+00
f20	3.1616E+04	7.8926E+04	1.9873E+05	4.9304E+04	4.6226E+03	1.5865E+04	2.0344E+04	3.7983E+03
f21	1.8796E+05	9.4611E+05	4.6657E+06	1.1088E+06	4.2262E+03	4.7363E+04	8.2040E+04	1.9815E+04
f22	2.8483E+03	3.1294E+03	3.8289E+03	2.6155E+02	2.5484E+03	2.8823E+03	3.0287E+03	1.3239E+02
f23	2.6505E+03	2.6798E+03	2.7697E+03	3.0043E+01	2.5000E+03	2.5000E+03	2.5000E+03	0.0000E+00
f24	2.6501E+03	2.6889E+03	2.7554E+03	2.9527E+01	2.6000E+03	2.6000E+03	2.6000E+03	9.2183E-05
f25	2.7092E+03	2.7184E+03	2.7389E+03	7.6355E+00	2.7000E+03	2.7000E+03	2.7000E+03	0.0000E+00
f26	2.7009E+03	2.7029E+03	2.7046E+03	9.1726E-01	2.7003E+03	2.7006E+03	2.7009E+03	1.3497E-01
f27	3.6157E+03	3.7254E+03	3.8674E+03	7.3522E+01	2.9000E+03	2.9000E+03	2.9000E+03	0.0000E+00
f28	3.8190E+03	3.9908E+03	4.5422E+03	1.8623E+02	3.0000E+03	3.0000E+03	3.0000E+03	0.0000E+00
f29	4.5568E+04	4.1996E+06	9.6170E+06	3.8194E+06	3.1000E+03	3.8930E+03	7.4882E+03	1.3820E+03
f30	2.7692E+04	6.8250E+04	1.6619E+05	3.9901E+04	4.7931E+03	6.7879E+03	1.1019E+04	1.4692E+03

(f17-f22); (4) composite (f23-30) problems. In this experiment, the individual dimension is 30, the population size is 30 and the maximum iteration times is 10000. The results of these test questions have been given in Tables 12, 13 and 14.

4.2.1 Analysis on CEC 2014

f1-f3 are the unimodal problems. Table 12 shows that IMFO can effectively exploiting search space. In Table 12, the performance of IMFO is better than that of MFO in four indicators. Therefore, the improved position updating strategy and orthogonal reverse flame generation of moths have been proved to be effective in the search space of development problems.

f4 to f16 are multimodal problems. As can be seen in Table 12, the other indicators except Std of f16 are that IMFO strategy is superior to MFO. For f16, although the Std of IMFO is larger than that of MFO, the mean, best and worst indices of IMFO are better than that of MFO. This shows that IMFO has better exploration ability than MFO. The results show that the mutation operator and the Euclidean distance are used to modify the moth position update mechanism, which is helpful to improve the exploration ability of the MFO. Therefore, IMFO is more effective than MFO in exploration capability.

f17-f22 and f23-30 are hybrid and composite test problems in CEC 2014 test set, respectively. As can be seen from Table 12 that IMFO performs better than MFO on test problems f17 to f30, and has better global search ability.

Table 13 Comparisons of IMFO and other algorithms on CEC 2014 problems

	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Function	IMFO		MFO		LMFO		MFO3	
f1	5.5861E+05	2.3509E+05	1.0980E+08	1.0064E+08	6.0974E+07	1.2754E+07	6.9525E+06	9.0832E+06
f2	6.6115E+03	1.9861E+03	1.9241E+10	8.5826E+09	4.3695E+09	1.3449E+09	2.5212E+08	7.7979E+08
f3	4.4013E+03	2.1651E+03	1.2820E+05	4.5480E+04	2.7460E+04	5.7809E+03	3.6198E+04	1.3590E+04
f4	4.4811E+02	3.3039E+01	1.8381E+03	9.1619E+02	7.7425E+02	9.4777E+01	4.9847E+02	4.5535E+01
f5	5.2000E+02	3.7384E-03	5.2036E+02	1.1357E-01	5.2084E+02	8.4502E-02	5.2016E+02	7.8812E-02
f6	6.2472E+02	3.0036E+00	6.3301E+02	3.7467E+00	6.2267E+02	1.3509E+00	6.3082E+02	3.0440E+00
f7	7.0003E+02	2.7976E-02	8.0996E+02	3.9328E+01	7.3564E+02	7.1195E+00	7.0278E+02	1.1770E+01
f8	9.1140E+02	1.7507E+01	9.5556E+02	2.7433E+01	9.6840E+02	1.2336E+01	9.4828E+02	3.0425E+01
f9	9.9344E+02	2.4103E+01	1.1473E+03	4.2945E+01	1.1143E+03	8.3712E+00	1.0989E+03	3.3131E+01
f10	3.9729E+03	5.1280E+02	4.9171E+03	6.0284E+02	5.0314E+03	6.0916E+02	4.8651E+03	4.1786E+02
f11	5.0562E+03	3.9282E+02	5.6681E+03	6.5860E+02	5.5995E+03	5.5375E+02	5.7109E+03	3.6679E+02
f12	1.2006E+03	3.2778E-01	1.2011E+03	3.3339E-01	1.2012E+03	4.1271E-01	1.2012E+03	4.5938E-01
f13	1.3005E+03	9.4485E-02	1.3021E+03	1.2051E+00	1.3009E+03	7.2505E-02	1.3006E+03	8.4829E-02
f14	1.4003E+03	4.3289E-02	1.4406E+03	2.1108E+01	1.4093E+03	1.6406E+00	1.4005E+03	2.1016E-01
f15	1.6332E+03	7.9868E+01	1.9957E+05	3.3070E+05	1.8792E+03	7.0097E+02	1.9130E+03	5.0830E+02
f16	1.6120E+03	4.2765E-01	1.6129E+03	2.7606E-01	1.6120E+03	1.6636E-01	1.6134E+03	1.8661E-01
f17	1.9435E+05	1.1436E+05	3.4252E+06	6.4536E+06	2.4542E+06	1.0267E+06	7.4843E+05	8.3969E+05
f18	3.5419E+03	1.2476E+03	2.5432E+07	1.0178E+08	1.6263E+07	7.0135E+06	9.6445E+03	5.5681E+03
f19	1.9136E+03	1.7814E+00	1.9960E+03	6.3791E+01	1.9322E+03	8.9198E+00	1.9276E+03	2.2283E+01
f20	1.5865E+04	3.7983E+03	7.8926E+04	4.9304E+04	1.5266E+04	2.4152E+03	5.9965E+04	2.3786E+04
f21	4.7363E+04	1.9815E+04	9.4611E+05	1.1088E+06	6.4969E+05	2.7175E+05	3.8579E+05	1.9877E+05
f22	2.8823E+03	1.3239E+02	3.1294E+03	2.6155E+02	2.6685E+03	9.6169E+01	2.9680E+03	1.1449E+02
f23	2.5000E+03	0.0000E+00	2.6798E+03	3.0043E+01	2.5000E+03	0.0000E+00	2.6187E+03	4.1835E+00
f24	2.6000E+03	9.2183E-05	2.6889E+03	2.9527E+01	2.6000E+03	0.0000E+00	2.6588E+03	7.7370E+00
f25	2.7000E+03	0.0000E+00	2.7184E+03	7.6355E+00	2.7000E+03	0.0000E+00	2.7181E+03	5.4167E+00
f26	2.7006E+03	1.3497E-01	2.7029E+03	9.1726E-01	2.7240E+03	4.2653E+01	2.7309E+03	4.6219E+01
f27	2.9000E+03	0.0000E+00	3.7254E+03	7.3522E+01	2.9000E+03	0.0000E+00	3.6169E+03	2.2919E+02
f28	3.0000E+03	0.0000E+00	3.9908E+03	1.8623E+02	3.0000E+03	0.0000E+00	4.9162E+03	7.0138E+02
f29	3.8930E+03	1.3820E+03	4.1996E+06	3.8194E+06	1.4581E+06	3.1601E+06	1.3143E+06	3.4898E+06
f30	6.7879E+03	1.4692E+03	6.8250E+04	3.9901E+04	4.3881E+04	2.3079E+04	1.2639E+04	9.0285E+03
Function	CMFO		IMFO2020		GSA		SSA	
f1	1.437E+08	1.380E+08	3.175E+08	9.985E+07	1.2021E+06	2.3792E+06	2.2004E+06	7.5460E+05
f2	1.615E+10	7.327E+09	2.314E+10	5.062E+09	1.1679E+04	4.0310E+03	1.4029E+04	7.9611E+03
f3	1.244E+05	5.822E+04	4.866E+04	5.297E+03	8.5415E+03	2.4114E+03	1.3322E+03	3.3597E+02
f4	1.927E+03	1.236E+03	1.965E+03	1.092E+03	5.4984E+02	2.7969E+01	5.0032E+02	2.6310E+01
f5	5.203E+02	1.536E-01	5.209E+02	6.759E-02	5.2000E+02	1.4767E-04	5.2007E+02	9.6167E-02
f6	6.256E+02	4.178E+00	6.309E+02	2.184E+00	6.1768E+02	1.7952E+00	6.2009E+02	2.9839E+00
f7	8.456E+02	6.967E+01	8.914E+02	4.354E+01	7.0000E+02	2.7842E-03	7.0002E+02	1.7433E-02
f8	9.641E+02	4.506E+01	1.039E+03	2.704E+01	9.4894E+02	1.2065E+01	9.1111E+02	2.7107E+01
f9	1.127E+03	5.320E+01	1.161E+03	2.363E+01	1.0654E+03	1.7459E+01	1.0351E+03	2.1580E+01
f10	4.604E+03	7.953E+02	6.468E+03	5.397E+02	4.3639E+03	4.1568E+02	4.9038E+03	5.7651E+02
f11	5.613E+03	7.306E+02	7.426E+03	5.167E+02	4.8174E+03	4.2711E+02	5.0507E+03	6.3586E+02
f12	1.201E+03	2.702E-01	1.202E+03	2.935E-01	1.2000E+03	4.4034E-04	1.2005E+03	1.6415E-01
f13	1.302E+03	1.270E+00	1.304E+03	5.216E-01	1.3002E+03	3.0053E-02	1.3005E+03	7.9522E-02
f14	1.443E+03	2.466E+01	1.471E+03	1.863E+01	1.4003E+03	4.7936E-02	1.4004E+03	1.4711E-01
f15	4.203E+05	5.282E+05	1.151E+04	5.878E+03	1.5036E+03	5.8540E-01	1.5095E+03	2.4018E+00

Table 13 (continued)

	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f16	1.613E+03	4.765E-01	1.612E+03	2.756E-01	1.6136E+03	2.3397E-01	1.6119E+03	3.8188E-01
f17	4.568E+06	5.133E+06	7.071E+06	3.835E+06	3.2095E+05	2.3608E+05	1.3561E+05	8.8414E+04
f18	6.785E+07	1.742E+08	1.056E+08	5.608E+07	2.5678E+03	5.7725E+02	9.6699E+03	5.6538E+03
f19	1.992E+03	7.392E+01	2.024E+03	4.320E+01	1.9515E+03	3.2172E+01	1.9155E+03	1.9617E+00
f20	8.164E+04	8.066E+04	2.325E+04	6.406E+03	2.2409E+04	4.7204E+03	2.3833E+03	6.2337E+01
f21	1.739E+06	3.181E+06	1.019E+06	7.237E+05	1.2878E+05	4.9968E+04	7.2403E+04	2.5391E+04
f22	3.076E+03	2.602E+02	2.994E+03	1.735E+02	3.2691E+03	1.6955E+02	2.6443E+03	1.3557E+02
f23	2.691E+03	4.935E+01	2.500E+03	0.000E+00	2.6178E+03	1.0648E+00	2.6153E+03	1.8096E-01
f24	2.696E+03	4.167E+01	2.600E+03	0.000E+00	2.6001E+03	1.4679E-02	2.6447E+03	3.6932E+00
f25	2.723E+03	1.075E+01	2.700E+03	0.000E+00	2.7000E+03	0.0000E+00	2.7130E+03	3.7650E+00
f26	2.702E+03	1.133E+00	2.719E+03	3.675E+01	2.8000E+03	3.7393E-03	2.7005E+03	8.5209E-02
f27	3.678E+03	1.582E+02	2.900E+03	0.000E+00	4.7288E+03	4.2150E+02	3.5099E+03	1.0085E+02
f28	3.958E+03	2.145E+02	3.000E+03	0.000E+00	5.9772E+03	6.3727E+02	3.9394E+03	2.4160E+02
f29	3.085E+06	3.770E+06	3.100E+03	0.000E+00	7.1615E+03	1.5910E+03	1.8515E+06	4.1764E+06
f30	5.042E+04	3.477E+04	2.926E+04	8.504E+04	9.8951E+03	2.7384E+03	1.1919E+04	1.9668E+03
Function	SCA		PSO		WOA		HSCA	
f1	2.8284E+08	6.7309E+07	4.1353E+07	3.3572E+07	2.7775E+07	3.2799E+06	3.1039E+07	2.5065E+07
f2	1.8412E+10	3.1818E+09	5.2319E+06	1.6583E+06	2.4454E+06	1.8632E+06	5.3302E+08	4.8038E+08
f3	3.9022E+04	4.7567E+03	3.1115E+04	1.1299E+04	2.8794E+04	1.9345E+04	7.5748E+03	4.0012E+03
f4	1.4800E+03	2.4624E+02	7.9329E+02	2.8944E+02	5.8018E+02	7.7683E+01	5.9642E+02	4.9540E+01
f5	5.2096E+02	2.9478E-02	5.2098E+02	3.8827E-02	5.2030E+02	2.0806E-02	5.2049E+02	1.3208E-01
f6	6.3472E+02	2.1775E+00	6.3782E+02	2.3761E+00	6.3585E+02	5.4828E+00	6.1858E+02	4.1905E+00
f7	8.4352E+02	2.4674E+01	7.0105E+02	1.3982E-02	7.0101E+02	1.1200E-01	7.0650E+02	3.8582E+00
f8	1.0461E+03	1.3183E+01	9.4325E+02	1.4649E+01	9.9051E+02	2.1792E+01	8.9068E+02	2.0586E+01
f9	1.1764E+03	1.4293E+01	1.0782E+03	2.2708E+01	1.1224E+03	4.6574E+01	1.0380E+03	3.6786E+01
f10	7.1357E+03	2.6681E+02	7.5270E+03	1.2559E+03	5.1219E+03	2.2790E+01	3.4095E+03	8.4734E+02
f11	8.3300E+03	1.6061E+02	8.8380E+03	5.0389E+02	5.8681E+03	1.6428E+03	4.5303E+03	6.7681E+02
f12	1.2026E+03	2.0717E-01	1.2032E+03	3.0465E-01	1.2018E+03	9.5782E-02	1.2006E+03	2.9481E-01
f13	1.3031E+03	2.6130E-01	1.3006E+03	8.9005E-02	1.3005E+03	1.1310E-01	1.3005E+03	1.1587E-01
f14	1.4479E+03	8.1046E+00	1.4003E+03	1.1860E-01	1.4003E+03	9.1875E-02	1.4005E+03	3.1495E-01
f15	5.4145E+03	3.2957E+03	1.6915E+03	7.2513E+01	1.5672E+03	8.0693E+00	1.5247E+03	3.2506E+01
f16	1.6130E+03	1.8527E-01	1.6129E+03	4.2092E-01	1.6126E+03	4.1530E-01	1.6120E+03	5.8618E-01
f17	6.9164E+06	3.5261E+06	1.4750E+06	4.0048E+06	5.1179E+06	1.3307E+06	1.0856E+06	7.9557E+05
f18	1.9853E+08	7.7924E+07	1.9634E+04	7.7708E+04	6.3151E+03	2.2734E+03	1.5651E+06	5.1783E+06
f19	1.9965E+03	2.2031E+01	1.9979E+03	3.5560E+01	1.9422E+03	4.2937E+00	1.9172E+03	5.0802E+00
f20	1.8043E+04	4.9540E+03	1.4499E+04	9.7187E+03	2.7384E+04	1.2683E+04	6.3540E+03	3.8955E+03
f21	1.3459E+06	3.1384E+05	1.2763E+06	6.5240E+06	7.8318E+05	7.6305E+04	2.8778E+05	2.9848E+05
f22	3.0456E+03	8.9609E+01	3.4116E+03	2.0596E+02	2.9057E+03	4.1526E+02	2.5273E+03	1.5830E+02
f23	2.6703E+03	1.0325E+01	2.6424E+03	1.5077E+01	2.6198E+03	1.4846E+00	2.6255E+03	5.2738E+00
f24	2.6001E+03	4.9113E-02	2.6399E+03	7.3006E+00	2.6262E+03	3.8080E-01	2.6000E+03	3.2118E-03
f25	2.7288E+03	4.7129E+00	2.7227E+03	5.6961E+00	2.7127E+03	1.1895E+01	2.7095E+03	4.4499E+00
f26	2.7026E+03	4.4742E-01	2.8001E+03	5.6336E-02	2.7004E+03	2.8676E-01	2.7006E+03	1.4557E-01
f27	3.4951E+03	3.0023E+02	4.4678E+03	2.2797E+02	3.9089E+03	1.5138E+02	3.4272E+03	1.3715E+02
f28	4.9332E+03	2.6794E+02	8.9952E+03	1.0766E+03	5.2256E+03	3.5458E+02	3.9118E+03	2.9316E+02
f29	1.4900E+07	7.4107E+06	2.3164E+08	1.4793E+08	7.3232E+06	1.6205E+04	1.2226E+06	3.4904E+06
f30	2.9879E+05	9.6167E+04	4.7430E+05	6.8853E+05	8.9582E+04	5.6783E+04	3.6233E+04	2.1150E+04

Table 13 (continued)

	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Function	EWOA		IDA		MVO		SHO	
f1	9.1958E+08	3.3378E+08	6.2882E+06	5.2442E+06	3.4197E+06	6.7290E+05	1.990E+09	3.076E+08
f2	6.2010E+10	1.1244E+10	3.4497E+06	6.2230E+06	1.4400E+04	1.2876E+04	8.569E+10	6.888E+09
f3	1.5068E+05	7.7436E+04	1.2413E+04	6.5291E+03	4.1182E+02	6.7267E+01	1.371E+06	2.690E+06
f4	9.9483E+03	3.4294E+03	5.2632E+02	4.2566E+01	4.8248E+02	4.2008E+01	1.845E+04	2.526E+03
f5	5.2087E+02	1.0677E-01	5.2017E+02	7.2483E-02	5.2009E+02	8.7464E-02	5.209E+02	6.624E-02
f6	6.3972E+02	2.4648E+00	6.1984E+02	2.8138E+00	6.0911E+02	4.5909E+00	6.460E+02	2.398E+00
f7	1.1583E+03	1.2290E+02	7.0098E+02	2.2100E-01	7.0004E+02	8.6228E-03	1.542E+03	6.840E+01
f8	1.1035E+03	2.7821E+01	9.0952E+02	2.6313E+01	8.6696E+02	1.5002E+01	1.224E+03	3.628E+01
f9	1.2428E+03	3.4773E+01	1.0227E+03	3.1109E+01	9.9714E+02	5.8191E+00	1.321E+03	1.682E+01
f10	7.6958E+03	7.8090E+02	3.7352E+03	5.0836E+02	3.9657E+03	7.0330E+02	1.009E+04	4.509E+02
f11	8.2111E+03	5.3770E+02	4.7353E+03	9.0648E+02	4.2089E+03	5.9622E+02	1.057E+04	4.563E+02
f12	1.2019E+03	4.8304E-01	1.2005E+03	2.8112E-01	1.2002E+03	9.6251E-02	1.204E+03	1.509E+00
f13	1.3071E+03	8.9991E-01	1.3006E+03	1.9036E-01	1.3004E+03	7.2324E-03	1.308E+03	1.230E+00
f14	1.6062E+03	3.6382E+01	1.4007E+03	4.0338E-01	1.4004E+03	2.2684E-02	1.720E+03	3.453E+01
f15	1.1738E+05	6.8768E+04	1.5137E+03	4.1319E+00	1.5066E+03	1.0717E+00	5.749E+05	1.317E+05
f16	1.6127E+03	4.5894E-01	1.6119E+03	4.0294E-01	1.6115E+03	3.6013E-01	1.613E+03	3.925E-01
f17	1.2016E+08	9.4373E+07	6.2348E+05	4.9981E+05	1.6688E+05	8.8799E+04	2.904E+08	1.347E+08
f18	1.6962E+09	1.1595E+09	3.5465E+03	1.4767E+03	1.2174E+04	2.0961E+03	7.128E+09	2.297E+09
f19	2.2738E+03	1.1894E+02	1.9138E+03	2.9832E+00	1.9102E+03	2.7574E+00	2.518E+03	8.531E+01
f20	3.0864E+05	3.8692E+05	4.0470E+03	2.0948E+03	2.2881E+03	1.1085E+01	1.348E+07	1.360E+07
f21	3.6120E+07	2.6392E+07	1.8146E+05	1.6146E+05	6.5215E+04	2.8580E+04	1.532E+08	9.628E+07
f22	4.4285E+03	3.0962E+03	2.7483E+03	1.5564E+02	2.5792E+03	3.0021E+01	4.017E+04	9.505E+04
f23	2.7045E+03	2.3214E+02	2.5945E+03	4.7343E+01	2.6155E+03	5.8440E-02	2.500E+03	0.000E+00
f24	2.6054E+03	5.3345E+00	2.6203E+03	5.2065E+00	2.6302E+03	2.2937E+00	2.600E+03	0.000E+00
f25	2.7064E+03	1.9973E+01	2.7041E+03	3.2100E+00	2.7056E+03	5.3442E-01	2.700E+03	0.000E+00
f26	2.7814E+03	8.8124E+01	2.7006E+03	1.4882E-01	2.7424E+03	7.9179E+01	2.783E+03	3.418E+01
f27	4.0896E+03	3.3355E+02	3.1060E+03	6.9888E+00	3.2687E+03	2.6202E+01	2.900E+03	0.000E+00
f28	6.2221E+03	1.4502E+03	4.8119E+03	5.8441E+02	3.8304E+03	4.0095E+01	3.000E+03	0.000E+00
f29	9.1768E+07	1.1120E+08	9.0500E+06	8.1599E+06	1.4800E+06	7.3183E+06	3.100E+03	0.000E+00
f30	2.1048E+06	1.9569E+06	8.3891E+03	3.0898E+03	7.6525E+03	1.3848E+03	3.200E+03	0.000E+00

4.2.2 Comparison with other algorithms

In this subsection, IMFO algorithm has been compared with MFO, LMFO, MFO3, CMFO, IMFO2020, SSA, SCA, PSO, WOA, HSCA, EWOA, IDA, MVO, SHO algorithms on the CEC2014 test problems. Table 13 shows the results of 16 algorithms on the test functions, with 30 dimensions and 10000 iterations. For Table 13, it can be observed that the IMFO has a strong competitiveness compared with other algorithms in Mean and Std indexes. The results show

Table 14 Test results of the 16 algorithms

	Rank	1/0/-1	Ave		Rank	1/0/-1	Ave
IMFO	1	~	2.53333	SCA	14	29/1/0	12.16667
MFO	13	29/0/1	11.20000	PSO	12	28/1/1	11.13333
LMFO	7	21/4/5	7.50000	WOA	9	28/2/0	8.73333
MFO3	8	29/0/1	8.46667	HSCA	5	22/3/5	5.96667
CMFO	11	28/0/2	11.13333	EWOA	16	29/1/0	13.96667
IMFO2020	10	22/5/3	10.13333	IDA	4	25/2/3	5.33333
GSA	6	22/4/4	6.00000	MVO	2	22/5/3	3.86667
SSA	3	22/4/4	5.03333	SHO	15	24/4/2	12.83333

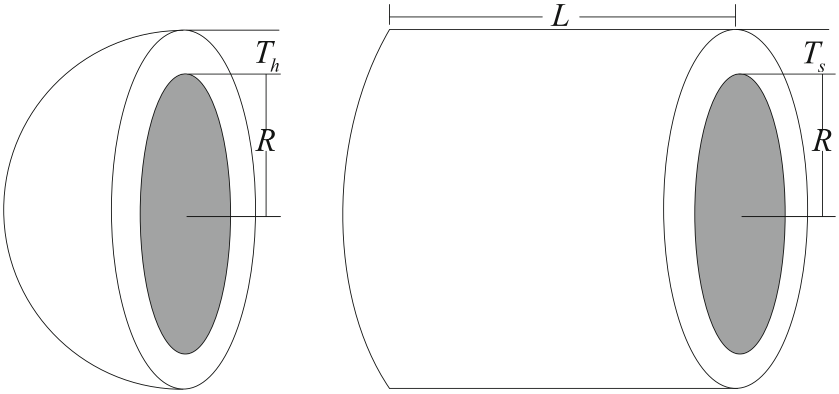

Fig. 4 Pressure vessel design problem

that the IMFO can get competitive solution by introducing OOBL and modified moth position update strategy.

To ensure the effectiveness of IMFO improvement, Wilcoxon signed rank test and Friedman test are also introduced, in which the parameter setting is consistent with the classical benchmark problem set. The statistical conclusions are shown in Table 14 and the symbols are the same as those described in Section 4.1.1. In Table 14, the results of IMFO are better than that of other algorithms in most test problems. It can be seen in Table 14 that the average ranking (Ave) of IMFO is the best.

5 Engineering optimization problems

In the above part, the performance of IMFO is simulated on classical test set and standard CEC2014 test set. In this section, to verify the ability of the IMFO to solve practical problems, the IMFO is used to optimize three practical engineering optimization problems.

5.1 Pressure vessel design problem

This problem is given in Fig. 4 [21]. Its purpose is to obtain the lowest cost cylindrical pressure vessel as much as possible. For this problem, there are four parameters to be optimized in Fig. 4: thickness of head T_h, thickness of shell T_s, length of cylindrical shell L and inner radius R. The mathematical expressions of this problem are as follows [21, 52, 53]:

$$\begin{aligned}\textbf{Min } F_1(x) &= 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 19.84x_1^2x_3 \\ &\quad +3.1661x_1^2x_4\end{aligned} \tag{23}$$

$$x = (x_1, x_2, x_3, x_4) = (T_s, T_h, R, L) \tag{24}$$

$$\begin{aligned}\textbf{s.t. } & g_1(x) = 0.0193x_3 - x_1 \le 0 \\ & g_2(x) = 0.00954x_3 - x_2 \le 0 \\ & g_3(x) = 1296000 - \tfrac{4}{3}\pi x_3^3 - \pi x_3^2 x_4 \le 0 \\ & g_4(x) = x_4 - 240 \le 0 \\ & 1 \times 0.0625 \le x_1, x_2 \le 99 \times 0.0625 \\ & 10 \le x_3, x_4 \le 200\end{aligned} \tag{25}$$

Table 15 shows the comparison results of the IMFO algorithm and other heuristic optimization algorithms for this problem, which includes the values of each variable and its corresponding results. From Table 15, the results show that IMFO can get the minimum cost than other optimization algorithms.

5.2 Cantilever beam design problem

This problem is shown in Fig. 5 [21], and its purpose is to get a set of values to minimize the weight of the cantilever. There are five parameters to be optimized: the

Table 15 Comparison of results on pressure vessel design problem

Algorithm	Optimal values for variables				Optimum cost
	T_s	T_h	R	L	
IMFO	0.8307	0.4106	43.041800	165.287000	5981.3416
MFO [21]	0.8125	0.4375	42.098445	176.636596	6059.7143
SCA [52]	0.8125	0.4375	42.048610	177.707800	6076.3651
ES [54]	0.8125	0.4375	42.098087	176.640518	6059.7456
PSO [55]	0.8125	0.4375	42.091266	176.746500	6061.0777
ACO [53]	0.8125	0.4375	42.103624	176.572656	6059.0888
Branch-bound [56]	1.1250	0.6250	47.700000	117.701000	8129.1036

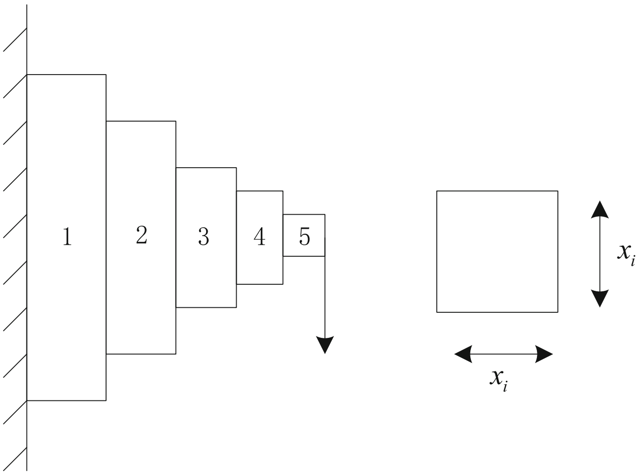

Fig. 5 Cantilever beam design problem

side length x_i $(i = 1, 2, \ldots, 5)$ of cross section (square) of the five different beams. The mathematical expressions of this problem are as follows [21, 47, 52]:

$$\textbf{Min } F_2(x) = 0.0624 \times \sum_{i=1}^{5} x_i \tag{26}$$

$$\textbf{s.t. } g(x) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \tag{27}$$

Table 16 shows the comparison results of the IMFO algorithm and other heuristic optimization algorithms for this problem, which includes the values of each variable and its corresponding results. From Table 16, the results show that IMFO can get the minimum weight design than other optimization algorithms.

5.3 Tension/compression spring design problem

This optimization problem is given in Fig. 6 [14]. Its purpose is to achieve lower manufacturing cost as much as possible. For this problem, there are three parameters to be optimized in Fig. 6: d indicates wire diameter; D indicates mean coil diameter; N number of active coils. In addition, the mathematical expressions of this problem are as follows [21, 55, 59, 60]:

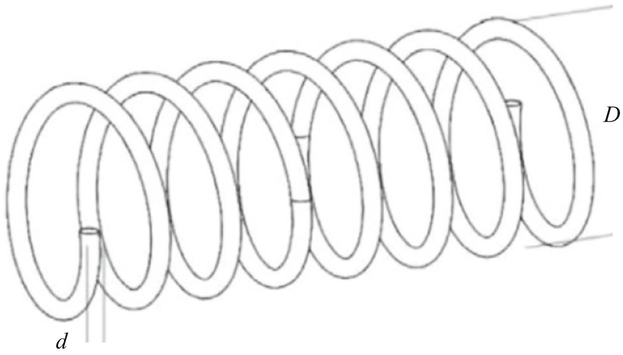

Fig. 6 Tension/compression spring design

$$\textbf{Min } F_3(x) = (x_3 + 2)\, x_2 x_1^2 \tag{28}$$

$$x = (x_1, x_2, x_3) = (d, D, N) \tag{29}$$

$$\begin{aligned} \textbf{s.t. } g_1(x) &= 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leqslant 0 \\ g_2(x) &= \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5108 x_1^2} \leqslant 0 \\ g_3(x) &= 1 - \frac{140.45 x_1}{x_2^2 x_3} \leqslant 0 \\ g_4(x) &= \frac{x_1 + x_2}{1.5} - 1 \leqslant 0 \end{aligned} \tag{30}$$

where $0.05 \leqslant x_1 \leqslant 2.00$, $0.25 \leqslant x_2 \leqslant 1.30$, $2.00 \leqslant x_3 \leqslant 15.0$.

Table 17 shows the comparison results of the IMFO algorithm and other heuristic optimization algorithms for this problem, which includes the values of each variable and its corresponding results. From Table 17, the results show that IMFO can get the minimum weight design than other optimization algorithms.

Table 16 Comparison of results on cantilever beam design problem

Algorithm	Optimal values for variables					Optimum weight
	x_1	x_2	x_3	x_4	x_5	
IMFO	5.813053964	4.886960118	5.009894999	3.028885438	2.639184503	1.330565414
MFO [21]	5.984871773	5.316726924	4.497332586	3.513616468	2.161620293	1.339988086
HSCA [47]	6.0208	5.2978	4.4913	3.5115	2.1524	1.3399
CS [57]	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
SCA [52]	6.3435	4.8755	4.1788	3.4916	2.1887	1.3400
GCA(I) [58]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
MMA [58]	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400

Table 17 Comparison of results on tension/compression spring design

Algorithm	Optimal values for variables			Optimum weight
	d	*D*	*N*	
IMFO	0.051912961	0.36212813	10.97868026	0.0126661
MFO [21]	0.051994457	0.36410932	10.86842186	0.0126669
GA [59]	0.05148	0.351661	11.632201	0.0127048
DE [60]	0.051609	0.354714	11.410831	0.0126702
PSO [55]	0.051728	0.357644	11.244543	0.0126747

6 Conclusions

In this paper, an improved Moth-Flame optimization algorithm is proposed to solve global optimization problems. The IMFO is mainly realized by flames generation strategy and modified position update mechanism of moths. The flames generation strategy is used to generate effective flames to guide moths by OOBL, which improves the performance of the MFO algorithm to jump out of local optimum and enhance exploration ability. Modified position updating mechanism of moths is proposed based on spiral search and mutation operator, which helps to increase convergence speed of the IMFO. To verify the effectiveness of IMFO, it has been compared with MFO, LMFO, MFO3, CMFO, IMFO2020, SSA, SCA, GSA, ABC, PSO, GWO, WOA, PSOGSA, HCLPSO, MVO, HSCA, EWOA, IDA, SHO in 23 benchmark functions and the CEC 2014 benchmark test functions. The comparative results show that the IMFO algorithm is effective, accurate and stable. In addition, the IMFO is also compared with other well-known algorithms (such as SCA, MMA and GCA) on three practical engineering optimization problems. The comparative results show that the IMFO algorithm can obtain competitive solutions in solving practical engineering optimization problems. Therefore, the comparative analysis results of this paper show that the IMFO algorithm overcomes some shortcomings of MFO algorithm, and it can be used to deal with the classical test problems and the practical engineering optimization.

Acknowledgments Project supported by the National Natural Foundation of China (Grant Numbers 61873226 and 61803327), Natural Science Foundation of Hebei Province (Grant Numbers F2017203304, F2019203090 and F2020203018).

References

1. Boubezoul A, Paris S (2012) Application of global optimization methods to model and feature selection. Pattern Recogn 45(10):3676–3686
2. Sebastian N, Suvrit S, Wright SJ (2011) Optimization for machine learning. The MIT Press, Cambridge
3. Pasandideh SHR, Niaki STA, Gharaei A (2015) Optimization of a multiproduct economic production quantity problem with stochastic constraints using sequential quadratic programming 84:98–107
4. Eberhart RC, Shi Y (2002) Particle swarm optimization: developments, applications and resources. In: Congress on evolutionary computation, vol 1, pp 81–86
5. Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: Proceedings of the IEEE international conference on neural networks, vol 284, pp 65–74
6. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39(3):459–471
7. Das S, Suganthan P (2011) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31
8. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17(12):4831–4845
9. Pierezan J, Coelho LDS (2018) Coyote optimization algorithm: a new metaheuristic for global optimization problems. In: 2018 IEEE Congress on evolutionary computation (CEC), pp 1–8
10. Yang X, Gandomi AH (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput 29(5):464–483
11. Sharma V, Pattnaik SS, Garg T (2012) A review of bacterial foraging optimization and its applications. Procedia - Social and Behavioral Sciences 48(1):1294–1303
12. Pan WT (2012) A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowl-Based Syst 26(2):69–74
13. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133
14. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69(3):46–61
15. Yang XS (2012) Flower pollination algorithm for global optimization. In: Unconventional computation and natural computation. Springer, pp 240–249
16. Dhiman G, Kumar V (2017) Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Adv Eng Softw 114:48–70
17. Sulaiman MH, Mustaffa Z, Saari MM, Daniyal H (2020) Barnacles mating optimizer: a new bio-inspired algorithm for solving engineering optimization problems. Eng Appl Artif Intel 87:103330
18. Moosavi SHS, Bardsiri VK (2019) Poor and rich optimization algorithm: a new human-based and multi populations algorithm. Eng Appl Artif Intel 86:165–181
19. 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
20. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
21. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249
22. Li C, Li S, Liu Y (2016) A least squares support vector machine model optimized by moth-flame optimization algorithm for annual power load forecasting. Appl Intell 45(4):1166–1178
23. Ebrahim MA, Becherif M, Abdelaziz AY (2018) Dynamic performance enhancement for wind energy conversion system using moth-flame optimization based blade pitch controller. Sustain Energy Technol Assess 27:206–212
24. Ishtiaq A, Ahmed S, Khan MF, Aadil F, Khan S (2019) Intelligent clustering using moth flame optimizer for vehicular ad hoc networks. Int J Distrib Sensor Netw 15(1):1–13

25. Das A, Mandal D, Ghoshal SP, Kar R (2018) Moth flame optimization based design of linear and circular antenna array for side lobe reduction. Int J Numer Modell: Electron Netw Devs Fields 32(7):1–15
26. Mittal N (2018) Moth flame optimization based energy efficient stable clustered routing approach for wireless sensor networks. Wireless Person Commun 1:1–18
27. Mohanty B, Acharyulu BVS, Hota PK (2018) Moth-flame optimization algorithm optimized dual-mode controller for multi-area hybrid sources agc system. Optim Control Applic Methods 39(4):720–734
28. Singh P, Prakash S (2017) Optical network unit placement in Fiber-Wireless (FiWi) access network by Moth-Flame optimization algorithm. Opt Fiber Technol 36:403–411
29. Mei RNS, Sulaiman MH, Mustaffa Z, Daniyal H (2017) Optimal reactive power dispatch solution by loss minimization using moth-flame optimization technique. Appl Soft Comput 59:210–222
30. 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
31. Li Z, Zhou Y, Zhang S, Song J (2016) Lvy-flight moth-flame algorithm for function optimization and engineering design problems. Math Probl Eng 2016:1–22
32. 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
33. 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
34. Xu Y, Chen H, Luo J, Zhang Q, Jiao S, Zhang X (2019) Enhanced moth-flameoptimizer with mutation strategy for global optimization. Inform Sci 492:181–203
35. Xu Y, Chen H, Heidari AA, Luo J, Zhang Q, Zhao X, Li C (2019) An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks. Expert Syst Applic 129:135–155
36. Taher MA, Kamel S, Jurado F, Ebeed M (2019) An improved moth-flame optimization algorithm for solving optimal power flow problem. Int Trans Electric Energy Syst 29(3):1–28
37. Tolan G, MH Khorshid M, Abou-El-Enien T (2016) Modified moth-flame optimization algorithms for terrorism prediction. Inte J Applic Innov Eng Manag (IJAIEM) 5:47–58
38. Mahdavi S, Rahnamayan S, Deb K (2018) Opposition based learning: a literature review. Swarm Evol Comput 39:1–23
39. Zhang Q, LEUNG YW (1999) An orthogonal genetic algorithm for multimedia multicast routing. IEEE Trans Evol Comput 3(1):53–62
40. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw, 1–29
41. Wang M, Chen H, Yang B, Zhao X, Hu L, Cai Z, Huang H, Tong C (2017) Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses. Neurocomputing 267(6):69–84
42. Lin GQ, Li LL, Tseng ML, Liu HM, Tan RR (2020) An improved moth-flame optimization algorithm for support vector machine prediction of photovoltaic power generation. J Clean Prod 253:119966
43. Hansen N, Ostermeier A (2001) Completely derandomized self-adaptation in evolution strategies. Evol Comput 9(2):159–195
44. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inform Sci 179:2232–2248
45. Mirjalili S, Hashim SZM (2010) A new hybrid PSOGSA algorithm for function optimization. In: 2010 International conference on computer and information application (ICCIA), pp 374–377
46. Lynn N, Suganthan PN (2015) Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. Swarm Evol Comput 24:11–24
47. Gupta S, Deep K (2019) A novel hybrid sine cosine algorithm for global optimization and its application to train multilayer perceptrons. Appl Intell, 1–34
48. Qais MH, Hasanien HM, Alghuwainem S (2020) Enhanced whale optimization algorithm for maximum power point tracking of variable-speed wind generators. Appl Soft Comput J 86:1–14
49. Li LL, Zhao X, Tseng ML, Tan RR (2020) Short-term wind power forecasting based on support vector machine with improved dragonfly algorithm. J Clean Prod 242:1–12
50. Derrac J, Garcia S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18
51. Liang J, Qu B, Suganthan P (2013) Problem Definitions and Evaluation Criteria for the CEC 2014. Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. Computational Intelligence Laboratory, Zhengzhou University Zhengzhou China and Technical Report. Nanyang Technological University, Singapore
52. Gupta S, Deep K (2018) Improved sine cosine algorithm with crossover scheme for global optimization. Knowl-Based Syst 165:374–406
53. Kaveh A, Talatahari S (2010) An improved ant colony optimization for constrained engineering design problems. Eng Comput 27(1):155–182
54. 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
55. He Q, Ling W (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intel 20(1):89–99
56. Sandgren E (1990) Nonlinear integer and discrete programming in mechanical design. J Mech Des 112(2):223–229
57. Gandomi HA, Yang X-S, Alavi HA (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29(2):245–245
58. Chickermane H, Gea H (1996) Structural optimization using a new local approximation method. Int J Numer Methods Eng 39:829–846
59. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems 41:113–127
60. Li LJ, Huang ZB, Liu F, Wu QH (2007) A heuristic particle swarm optimizer for optimization of pin connected structures. Comput Struct 85(7):340–349

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

Xiaodong Zhao received his B.E. degree in automation from Shandong University of Technology in 2014, his M.E. degree in control science and engineering from the Yanshan University in 2017. He is a Ph.D. Candidate with the control science and engineering, Yanshan University. His research interests include system modeling and swarm intelligence optimization algorithms.

Yiming Fang received his B.E. and M.E. degrees in automation from the Northeast Heavy Machinery Institute (which was renamed Yanshan University in 1997), China, in 1985 and 1988, respectively, and his Ph.D. degree in mechatronic engineering from the Yanshan University in 2003. He is currently a Professor with the College of Electrical Engineering, Yanshan University, China. His research interests include modeling and controlling of complex system, adaptive robust control theory and application, automation technology and application of metallurgical industrial process.

Le Liu received his B.E. degree in automation from the Hebei University of Science & Technology in 2008, his M.E. and Ph.D. degrees in control science and engineering from the Yanshan University, in 2011 and 2015, respectively. He is currently an associate professor with the College of Electrical Engineering, Yanshan University, China. His research interests include modeling, analysis and control of complex dynamic systems, intelligent optimization algorithm.

Jianxiong Li received the B.E. degree and Ph.D. in control science and engineering from Yanshan University in 2004 and 2012. He is currently an associate professor with the College of Electrical Engineering, Yanshan University, China. His research interests include robust adaptive control theory with applications to electro-hydraulic servo system.

Miao Xu received the B.E. degree in control science and engineering from Yanshan University in 2016. She is a Ph.D. Candidate with the control science and engineering, Yanshan University. Her current researches interesting include tracking control and coordination control of the multi agent systems.