

Received July 5, 2021, accepted September 14, 2021, date of publication September 16, 2021, date of current version September 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3113323

A Modified Manta Ray Foraging Optimization for Global Optimization Problems

ANDI TANG¹, HUAN ZHOU¹, TONG HAN, AND LEI XIE¹

Air Force Engineering University, Xi'an 710038, China

Corresponding author: Huan Zhou (kgy_zhouh@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62101590, in part by the Science Foundation of the Shaanxi Province, China, under Grant 2020JQ-481 and Grant 2021JM-224, and in part by the Aero Science Foundation of China under Grant 201951096002.

ABSTRACT The Manta ray foraging optimization (MRFO) is a novel swarm-based metaheuristic optimizer. It is mainly modeled by simulating three foraging behaviors of the Manta rays, which has a good performance. However, several drawbacks of MRFO have been noticed by analyzing its mathematical model. Random selection of reference points in the early iterations weakens the exploitation capability of MRFO. Chain foraging tends to lead the algorithm into local optimum. In addition, the algorithm suffers from the deficiency of decreasing population diversity in the late iteration. To address these shortcomings, a modified MRFO using three strategies, called m-MRFO, is proposed in this paper. An elite search pool (ESP) is established in this paper to enhance exploitation capability. By using adaptive control parameter strategies (ACP), we expand the range of MRFO's exploration in the early iterations and enhance the accuracy of exploitation in the later iterations, balancing exploiting and exploring capabilities. Furthermore, we use a distribution estimation strategy (DES) to adjust the evolutionary direction using the dominant population information to promote convergence. The m-MRFO performance was verified by selecting 23 classical test functions and CEC2017 test suite. The significance of the results was also verified by Friedman test, Wilcoxon test and Iman-Davenport test. Moreover, we have confirmed the potential of m-MRFO to solve real-world problems by solving three engineering design problems. The simulation results show that the improvement strategy proposed in this paper can effectively improve the performance of MRFO. m-MRFO is highly competitive.

INDEX TERMS Manta rays foraging optimizer, metaheuristic, swarm intelligence algorithm, engineering optimization problems.

I. INTRODUCTION

Global optimization problems can be found in almost every field of finance, engineering, and science. With the development of science and technology, increasingly complex optimization problems are emerging. Many realistic optimization problems are accompanied by several difficulties: expensive computational costs, complex non-linear constraints, dynamic objective functions and huge search ranges [1]. In this case, it is a challenge to efficiently find a solution that satisfies the constraint. Conventional mathematical or numerical programming methods are overwhelmed when faced with multiple types of non-integrable, non-continuous

problems [2]. In addition, they have difficulty in balancing accuracy and time cost when solving large-scale real-world multimodal problems. Metaheuristic optimization algorithms, as a class of stochastic optimization algorithms, perform better in balancing the quality of solution and time cost. It has been widely used to solve complex optimization problems in natural and engineering fields due to its simple structure and its non-reliance on the gradient information of specific problems [3]. More and more scholars are paying attention to and working on metaheuristic optimization algorithms.

In the past decades, various algorithms have been proposed one after another. In general, meta-heuristic optimization algorithms can be divided into three groups [4]: evolution-based algorithms, physics-based algorithms, and

The associate editor coordinating the review of this manuscript and approving it for publication was Jagdish Chand Bansal.

swarm-based algorithms. Genetic algorithm (GA) [5] that simulates the survival of the fittest mechanism in nature is a widely used evolutionary algorithm. In addition, other evolutionary algorithms have been proposed, including differential evolution (DE) [6], evolutionary programming (EP) [7], and evolutionary strategies (ES) [8]. The popularity of these evolutionary algorithms has also encouraged more and more researchers to study and propose other evolutionary algorithms [9]–[11]. The physics-based algorithms construct optimization models by emulating the physical laws of the universe. Simulated annealing (SA) [12] inspired by annealing phenomena in metallurgy is one of the best-known physics-based algorithms. Apart from SA, other physics-based algorithms have been proposed, such as gravity search algorithm (GSA) [13], nuclear reaction optimizer (NRO) [14], sine cosine algorithm (SCA) [15], black hole algorithm (BOA) [16], and water cycle algorithm (WCA) [17]. The swarm-based optimization algorithm performs by simulating the social behavior of the population. Particle swarm optimization (PSO) [18] and ant colony optimization (ACO) [19] are two classical swarm-based optimization algorithms. They perform by simulating a bird colony and an ant colony cooperating in foraging, respectively. There are other swarm intelligence optimization algorithms including: grey wolf optimizer (GWO) [20], whale optimization algorithm (WOA) [21], sparrow search algorithm (SSA) [22], firefly algorithm (FA) [12], artificial bee colony algorithm (ABC) [23], and so on [24]–[27].

Recently, a swarm-based algorithm called Manta ray foraging optimization (MRFO) that emulates the foraging behavior of manta rays is proposed by Zhao in 2020 [28]. As a newly proposed algorithm, MRFO is quickly applied to solve various engineering optimization problems. Abd Elaziz *et al.* [29] use fractional order calculus to enhance MRFO and applies it to multilevel thresholding image segmentation. Ghosh *et al.* [30] propose a binary version of MRFO and solves the feature selection problem. An improved version of MRFO is proposed by Xu *et al.* [31] and high-temperature proton exchange membrane fuel cell is analyzed and optimized. Hassan *et al.* [32] propose an improved MRFO with a hybrid gradient-based optimizer and uses it to solve the economic emission dispatch problem. The global maximum power point tracker based on MRFO is proposed by Fathy *et al.* [33].

However, MRFO also has the shortcomings of insufficient exploitation ability, decreasing population diversity, and easy to fall into local optimum. These deficiencies are mainly caused by the imbalance in the exploitation and exploration of the search space by the algorithm. In order to enhance the algorithm performance and balance the exploitation and exploration capabilities, in this paper, a modified MRFO (m-MRFO) with three improvement strategies is proposed. An elite search pooling strategy is proposed to improve the algorithm exploitation ability for the deficiency of too slow convergence due to random selection of reference points in the early iterations. To balance the algorithm exploitation and

exploration, an adaptive parameter control strategy is proposed. And a Gaussian probability model is used to describe the dominant population distribution and modify the evolutionary direction, thus improving the algorithm performance.

To fully verify the performance of m-MRFO, 51 functions and 3 engineering design problems are used. And the superiority of the algorithm is verified by numerical analysis, convergence analysis, stability analysis, Wilcoxon test and Friedman test. The main contributions of this paper are as follows: (1) An elite search pool is introduced to improve the exploitation of the algorithm. (2) To balance the exploitation and exploration of the algorithm, an adaptive parameter control strategy is proposed. (3) The evolutionary direction is modified using Gaussian probability model to improve the performance of m-MRFO. (4) The superiority of m-MRFO is tested on 51 test functions and 3 engineering design problems.

The remainder of this paper is organized as follows. A review of the basic MRFO is presented in Section II. Section III provides a detailed description of the proposed m-MRFO. In Section IV, the effectiveness of the proposed improvement strategy is verified using the classical test functions and CEC 2017 test suite. Furthermore, the m-MRFO is applied to solve three engineering design problems in Section IV. Finally, we summarize this work in Section V and offer directions for future research.

II. THE BASIC MRFO

In this section, the basic steps of MRFO are described. MRFO is performed by simulating three foraging strategies of manta rays, namely chain foraging, cyclone foraging and somersault foraging. Similar to other swarm-based metaheuristic algorithms, MRFO generates initial populations randomly in the search space. Then it is updated by the three strategies mentioned above. The mathematical models for these three foraging strategies are given respectively below.

A. CHAIN FORAGING

The manta rays form a foraging chain by linking their heads and tails in a line. MRFO considers that the best solution is a higher concentration of plankton, which is the target food for manta rays. While the first individual moves only towards food, the rest of the individuals move not only towards food but also towards individuals located in front of themselves in the foraging chain. The mathematical model of chain foraging is described as follows.

$$\mathbf{x}_i^{t+1} = \begin{cases} \mathbf{x}_i^t + \mathbf{r}_1 \cdot (\mathbf{x}_{best}^t - \mathbf{x}_i^t) \\ + \alpha \cdot (\mathbf{x}_{best}^t - \mathbf{x}_i^t), & i = 1 \\ \mathbf{x}_i^t + \mathbf{r}_2 \cdot (\mathbf{x}_{i-1}^t - \mathbf{x}_i^t) \\ + \alpha \cdot (\mathbf{x}_{best}^t - \mathbf{x}_i^t), & i = 2, 3, \dots, NP \end{cases} \quad (1)$$

$$\alpha = 2 \cdot \mathbf{r}_3 \cdot \sqrt{|\log(\mathbf{r}_4)|} \quad (2)$$

where \mathbf{x}_i^t is the position of the i^{th} individual at generation t . $\mathbf{r}_i \in [0, 1]$, $i = 1, 2, 3, 4$ are uniformly distributed random vectors. \mathbf{x}_{best}^t is the plankton with the highest concentration,

that is, the optimal individual. NP is the number of populations. α is a weight coefficient.

B. CYCLONE FORAGING

When manta rays find plankton in deep water, they form long foraging chains and then move toward food in a spiral. This behavior is similar to WOA, but in addition to spiraling close to food, it also follows the individuals in front of it. The mathematical model of cyclone foraging can be given by the following equation.

$$\mathbf{x}_i^{t+1} = \begin{cases} \mathbf{x}_{best}^t + r_5 \cdot (\mathbf{x}_{best}^t - \mathbf{x}_i^t) \\ + \beta \cdot (\mathbf{x}_{best}^t - \mathbf{x}_i^t), i = 1 \\ \mathbf{x}_{best}^t + r_6 \cdot (\mathbf{x}_{i-1}^t - \mathbf{x}_i^t) \\ + \beta \cdot (\mathbf{x}_{best}^t - \mathbf{x}_i^t), i = 2, 3, \dots, NP \end{cases} \quad (3)$$

$$\beta = 2 \cdot \exp(r_7 \cdot (iter_{max} - iter + 1)/iter_{max}) \cdot \sin(2\pi r_7) \quad (4)$$

where $r_i \in [0, 1], i = 5, 6$ is uniformly distributed random vectors. β is the weight coefficient. $r_7 \in [0, 1]$ is a uniformly distributed random number. $iter_{max}$ and $iter$ are the maximum number of iterations and the current number of iterations, respectively.

In Eq. (3), food is mainly used as a reference point for spiral foraging, which contributes to the full exploitation of the space near food. In addition, to expand the search range, a randomly generated location in the search space is used as a reference location for spiral foraging. This allows all individuals to search for areas far from their current best position. The random spiral foraging mechanism focuses mainly on exploration, allowing MRFO to perform a broad global search. The specific mathematical model is described as follows.

$$\mathbf{x}_{rand} = \mathbf{lb} + r_8 \cdot (\mathbf{ub} - \mathbf{lb}) \quad (5)$$

$$\mathbf{x}_i^{t+1} = \begin{cases} \mathbf{x}_{rand} + r_9 \cdot (\mathbf{x}_{best}^t - \mathbf{x}_i^t) \\ + \beta \cdot (\mathbf{x}_{best}^t - \mathbf{x}_i^t), i = 1 \\ \mathbf{x}_{rand} + r_{10} \cdot (\mathbf{x}_{i-1}^t - \mathbf{x}_i^t) \\ + \beta \cdot (\mathbf{x}_{best}^t - \mathbf{x}_i^t), i = 2, 3, \dots, NP \end{cases} \quad (6)$$

where \mathbf{x}_{rand} is a random position randomly produced in the search space. $r_i \in [0, 1], i = 8, 9, 10$ are uniformly distributed random vectors. \mathbf{ub} and \mathbf{lb} are the upper and lower bounds of the search space, respectively.

C. SOMERSAULT FORAGING

In this phase, the food location is considered as a pivot point. Each individual flip around the pivot and thus searches for a new location. The mathematical model of this phase is represented as follows.

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + S \cdot (r_{11} \cdot \mathbf{x}_{best}^t - r_{12} \cdot \mathbf{x}_i^t), \quad i = 1, 2, \dots, NP \quad (7)$$

where S is the somersault factor that decides the somersault range of manta rays and $S = 2$. r_{11} and r_{12} are two random numbers in $[0, 1]$.

MRFO regulates the exploration and exploitation behavior by controlling the change of $(iter/iter_{max})$. When $(iter/iter_{max}) < rand$, the exploration behavior is mainly performed, and food sources are randomly generated as reference points in the search space. When $(iter/iter_{max}) \geq rand$, the optimal individual is used as a reference point, which facilitates the exploitation of the algorithm. In addition, a random number is used to select chain foraging or spiral foraging. After that, Somersault foraging is performed.

III. THE MODIFIED MRFO

To overcome the shortcomings of MRFO, we use an elite search pool instead of randomly generated individuals as reference points to improve the algorithm exploitation performance. In addition, we make a good transition from exploration to exploitation with an adaptive parameter control strategy. And a balance between them is achieved. To modify the evolutionary direction, we use a distribution estimation strategy. By sampling the dominant population information, we enhance the population diversity and improve the algorithm performance. The mathematical model of m-MRFO is described in detail as follows.

A. ELITE SEARCH POOL STRATEGY (ESP)

Analysis of Eq. (5) and Eq. (6) shows that the reference location of cyclone foraging is randomly generated in the search space in the early iterative stage. While this facilitates the algorithm to search more space, the large range of random positions weakens the algorithm exploitation ability and slows down the convergence speed. In order to enhance the exploitation capability while still retaining its ability to search large spaces, an elite search pool strategy is proposed in this paper. We put the current best three individuals into a set. As shown in Eq. (8).

$$\mathbf{X}_{esp} = \{\mathbf{X}_{esp1}, \mathbf{X}_{esp2}, \mathbf{X}_{esp3}\} \quad (8)$$

where $\mathbf{X}_{esp1}, \mathbf{X}_{esp2}$ and \mathbf{X}_{esp3} are the best three individuals.

The reference point is chosen randomly from these three individuals each time. By using the ESP strategy, the position of the reference point is changed from randomly generated to one of the best three individuals. This greatly enhances the algorithm exploitation capability. Meanwhile, the three individuals are chosen randomly, which to some extent avoids the prematureness of the algorithm caused by the optimal individual falling into the local optimum. To balance the algorithm exploitation and exploration, we also add a new individual to the ESP. This new individual is randomly composed of the best three individuals. This retains the possibility to select dominant individuals and also provides the option to select positions at a longer range. Thus, the final mathematical model of the ESP strategy is described as follows:

$$\mathbf{x}_{esp} = \{\mathbf{x}_{esp1}, \mathbf{x}_{esp2}, \mathbf{x}_{esp3}, \mathbf{x}_{espr}\} \quad (9)$$

$$\mathbf{x}_{espr} = r_{13} \cdot \mathbf{x}_{esp1} + r_{14} \cdot \mathbf{x}_{esp2} + r_{15} \cdot \mathbf{x}_{esp3} \quad (10)$$

where $r_i \in [0, 1], i = 13, 14, 15$ are uniformly distributed random vectors.

B. ADAPTIVE CONTROL PARAMETER STRATEGY (ACP)

The original MRFO balances the search behavior by controlling the value of $(iter/iter_{max})$. $(iter/iter_{max})$ is a linearly increasing variable, which does not accurately reflect and accommodate the complex nonlinear search process. The nonlinear parameter control strategy is an effective measure to prevent premature. Several researchers have proposed various nonlinear parametric control strategies for balancing exploitation and exploration [34]–[36]. In this paper, we propose an adaptive control parameter strategy with a mixture of sine and cosine functions. The specific mathematical model of parameter *Coef* is as follows.

$$Coef = \sin(0.5\pi \cdot iter/iter_{max})^{2.5\cos(iter/iter_{max})^3} \quad (11)$$

As shown in Figure 1, the new strategy focuses more on exploration in the early stage to avoid the algorithm from falling into local optimum. In the later stage, it keeps the exploitation of large probability, which helps the algorithm to accelerate the convergence. Furthermore, we note that the parameter *S* is constant when the original MRFO performs somersault foraging, which is not beneficial for the algorithm to perform effectively. In the early stages of optimization, the algorithm performs more exploratory behavior, so *S* needs to be large enough to search more space. In the late iteration, the algorithm needs to be more precise for exploitation. At this point, too large *S* will lead to a weakening of the algorithm’s exploitation ability. Therefore, we need a smaller value of *S*. Therefore, we propose a linear decreasing strategy with parameter *S*. The mathematical model is as follows.

$$S = (Smin - Smax)/iter_{max} \cdot iter + Smax \quad (12)$$

where *Smin* and *Smax* are the maximum and minimum values of the parameter *S*, respectively.

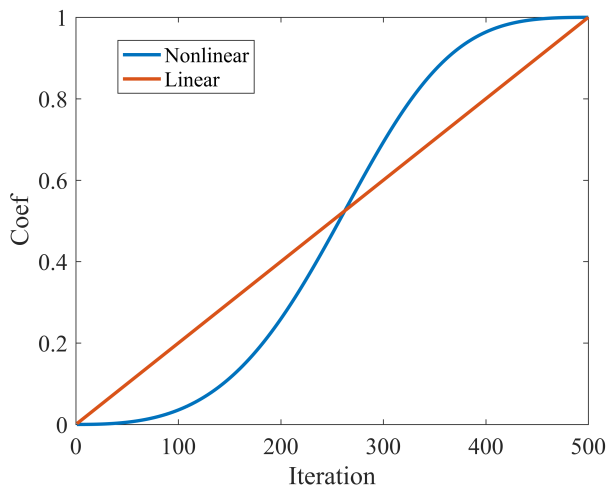


FIGURE 1. Comparison of control parameters.

C. DISTRIBUTION ESTIMATION STRATEGY (DES)

The original MRFO’s chain foraging strategy uses the optimal individual and neighboring individuals for position updating.

This leads to a premature convergence of the algorithm if the optimal individual has fallen into a local optimum, then the chain rule leads all subsequent individuals to approach the locally optimal individual. In order to enhance the algorithm performance, a distribution estimation strategy is proposed in this paper with the following mathematical model.

$$x_i^{t+1} = mean + y, y \sim N(0, Cov) \quad (13)$$

$$mean = (x_{esp} + x_{mean}^t + x_i^t)/3 \quad (14)$$

$$Cov(i) = \frac{1}{NP/2} \sum_{i=1}^{NP/2} (x_i^{t+1} - x_{mean}^t) \times (x_i^t - x_{mean}^t)^T \quad (15)$$

$$x_{mean}^t = \sum_{i=1}^{NP/2} \omega_i \times x_i^t \quad (16)$$

$$\omega_i = \frac{\ln(NP/2 + 0.5) - \ln(i)}{\sum_{i=1}^{NP/2} (\ln(NP/2 + 0.5) - \ln(i))} \quad (17)$$

where x_{mean}^t denotes the weighted position of the dominant population and ω denotes the weight coefficient in the dominant population in descending order of fitness values. *Cov* is the weighted covariance matrix of the dominant populations. In m-MRFO, the DES and the chain foraging strategy are randomly selected for execution.

The pseudocode and flow chart of the m-MRFO are shown in Algorithm 1 and Figure 2.

Algorithm 1 The Procedure of m-MRFO

Initialize search agents (Manta ray) populations $i = 1, \dots, n$
While termination criteria are not met
Calculate the fitness, construct the elite search pool based on Equation (9)
Calculate <i>mean</i> and <i>Cov</i> based on equation (14) and equation (15)
If $rand < 0.5$
If $Coef > rand$
Update Manta ray based on equation (3)
Else
Update Manta ray based on equation (6) and equation (9)
End if
Else
If $0.5 > rand$
Update Manta ray based on equation (1)
Else
Update Manta ray based on equation (13)
End if
End
Boundary control; calculate fitness of each agent
Greedy strategy is adopted to select the offspring
Calculate the <i>S</i> based on equation (12)
Update Manta ray based on equation (7)
Boundary control; calculate fitness of each agent
Greedy strategy is adopted to select the offspring
End while
Output the best solution

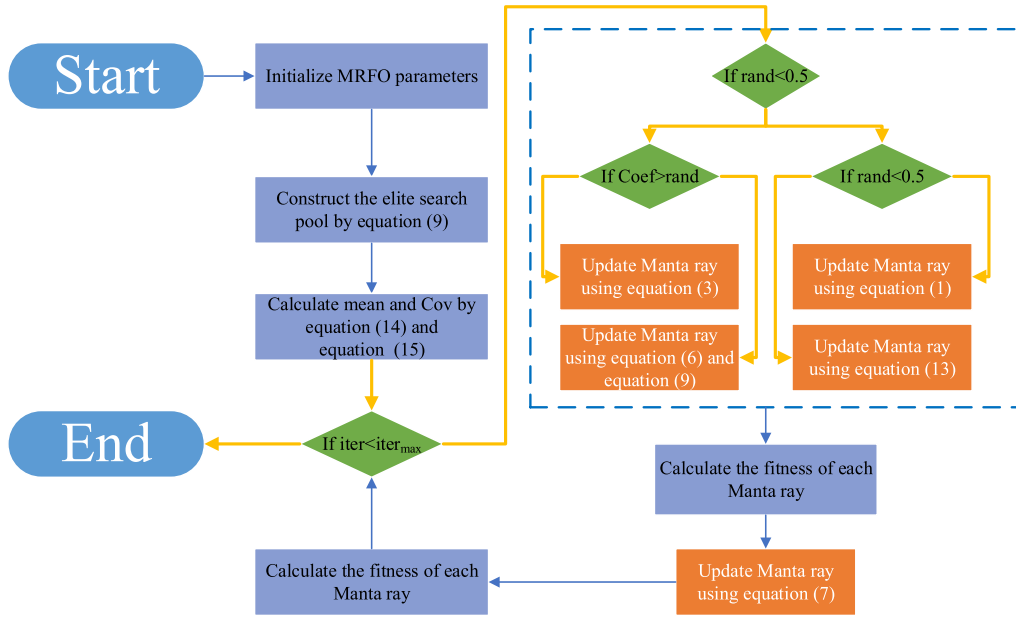


FIGURE 2. Flow chart of m-MRFO.

D. TIME COMPLEXITY OF THE MRFO

The time complexity of mrfo can be seen from the literature [28] as follows.

$$O(MRFO) = O(T(O(cyclone foraging) + chain foraging) + O(somersault foraging)) \tag{18}$$

$$O(MRFO) = O(T(NP \cdot D + NP \cdot D)) = O(T \cdot NP \cdot D) \tag{19}$$

In this paper, three improvement strategies are proposed, ESP and ACP do not change the time complexity. The time complexity of the covariance matrix of DES is $O(T(NP/2 \cdot D^2))$. Therefore, the time complexity of m-MRFO is shown below.

$$O(m - MRFO) = O(T(O(cyclone foraging) + O(chain foraging + DES)) + O(somersault foraging)) \tag{20}$$

$$O(m - MRFO) = O(T(NP \cdot D) + T(NP/2 \cdot D + NP/2 \cdot D^2) + T(NP \cdot D)) = O(T \cdot NP/2 \cdot D^2) \tag{21}$$

where T is the maximum value of the iteration, NP is the number of individuals, and D is the number of variables.

IV. EXPERIMENTS RESULTS

To verify the performance of m-MRFO, two different sets of benchmark functions are used for testing. The first group includes 23 classical test functions with the information

shown in Table 1. The second group is the CEC2017 test suite, which has the detailed information shown in Table 2. There are 51 test functions in total in the two test suites, which can be divided into unimodal test functions, multimodal test functions, low-dimensional test functions, hybrid functions and composite functions. The unimodal test functions have only one global optimal solution, so they are often applied to evaluate the exploitation capability of an algorithm. The multimodal test functions, on the other hand, have multiple global optimal solutions and are therefore used to test the exploration capability of the algorithms. The low-dimensional test functions check the algorithm’s ability to explore in low dimensions. Hybrid and composite functions are more complex and can be used to verify the overall performance of an algorithm.

There are four parts of the experiment on m-MRFO. In the first part of the experiment, we need to determine the values of S_{max} and S_{min} . The values of S_{max} and S_{min} affect the development and exploration ability of the algorithm, so we adopt 23 classical test functions to get their optimal values. Secondly, to verify the effectiveness of the three improvement strategies proposed in this paper, we employ the CEC2017 test suite for testing. In the third and fourth parts of the experiment, m-MRFO is compared with other algorithms using the classical test functions and the CEC2017 test suite, respectively. In the last part, three engineering optimization problems are used to test the proposed m-MRFO performance.

To ensure fair comparison, for the classical test functions, all algorithms adopt the same dimensions, the maximum number of iterations is set to 300, the number of populations is set to 50, and all test functions are run 30 times

TABLE 1. The classic benchmark functions (M: Multimodal, U: Unimodal, S: Separable, N: Nonseparable, Dim: Dimension, Range: Limits of search space, Optimum: Global optimal value).

Test function	Name	Type	Dim	Range	Optimum
$f_{01}(x) = \sum_{i=1}^D x_i^2$	Sphere	US	30	[-100, 100]	0
$f_{02}(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	Schwefel 2.22	UN	30	[-10, 10]	0
$f_{03}(x) = \sum_{i=1}^D (\sum_{j=1}^D x_j)^2$	Schwefel 1.2	UN	30	[-100, 100]	0
$f_{04}(x) = \max_i \{ x_i , 1 \leq i \leq D\}$	Schwefel 2.21	US	30	[-100, 100]	0
$f_{05}(x) = \sum_{i=1}^D 100(x_{i+1}^2 - x_i^2)^2 + (x_i - 1)^2$	Rosenbrock	UN	30	[-30, 30]	0
$f_{06}(x) = \sum_{i=1}^D (\lfloor x_i + 0.5 \rfloor)^2$	Step	US	30	[-100, 100]	0
$f_{07}(x) = \sum_{i=1}^D ix_i^4 + random[0,1)$	Quartic	US	30	[-1.28, 1.28]	0
$f_{08}(x) = \sum_{i=1}^D -x_i \sin(\sqrt{ x_i })$	Schwefel 2.26	MS	30	[-500, 500]	-418.9829*D
$f_{09}(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	Rastrigin	MS	30	[-5.12, 5.12]	0
$f_{10}(x) = 20 + e - 20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i))$	Ackley	MS	30	[-32, 32]	8.8818e-16
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^D (x_i^2) - (\prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}})) + 1$	Griewank	MN	30	[-600, 600]	0
$f_{12}(x) = \frac{\pi}{D} \{10 \sin^2(\pi y_i) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)\}$ $+ \sum_{i=1}^D u(x_i, 10, 100, 4) \quad y_i = 1 + \frac{x_i + 1}{4} \quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	Penalized	MN	30	[-50, 50]	0
$f_{13}(x) = 0.1 \{ \sin^2(3\pi x_i) + \sum_{i=1}^D (x_i - 1)^2 [1 + \sin^2(3\pi x_i)] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \} + \sum_{i=1}^D u(x_i, 5, 100, 4)$	Penalized2	MN	30	[-50, 50]	0
$f_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_j)^6})^{-1}$	Foxholes	MS	2	[-65.53, 65.53]	0.998004
$f_{15}(x) = \sum_{i=1}^{11} (a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4})^{-1}$	Kowalik	MS	4	[-5, 5]	0.0003075
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 + x_1x_2 - 4x_2^2 + x_2^4$	Six Hump Camel Back	MN	2	[-5, 5]	-1.03163
$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10$	Branin	MS	2	[-5, 10] x [0, 15]	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	Goldstein Price	MN	2	[-5, 5]	3
$f_{19}(x) = -\sum_{i=1}^4 (c_i \exp(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2))$	Hartman 3	MN	3	[0, 1]	-3.8628
$f_{20}(x) = -\sum_{i=1}^4 (c_i \exp(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2))$	Hartman 6	MN	6	[0, 1]	-3.32
$f_{21}(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	Langermann 5	MN	4	[0, 10]	-10.1532
$f_{22}(x) = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	Langermann 7	MN	4	[0, 10]	-10.4029
$f_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	Langermann 10	MN	4	[0, 10]	-10.5364

independently. For CEC2017, *Dim*, *iter_{max}*, and *NP* are set uniformly to 30, 600, and 500, respectively. all test functions are run independently 51 times. The experiments in this paper were conducted on a computer with an AMD R7 4800U processor and 16 GB RAM. Programming was performed using MATLAB R2016b.

A. ANALYSIS OF Smax AND Smin PARAMETER SETTINGS

The algorithm parameter settings have a great impact on its performance. In the original MRFO, *S* is the key parameter for somersault foraging. We control the variation of *S* by the parameters *Smax* and *Smin* to balance the algorithm exploitation and exploration. In this section, we need to identify the

TABLE 2. Descriptions of CEC 2017 test suite.

Type	No.	Description	Fj*
Unimodal functions	24	Shifted and Rotated Bent Cigar Function	300
	25	Shifted and Rotated Rosenbrock's Function	400
	26	Shifted and Rotated Rastrigin's Function	500
	27	Shifted and Rotated Expanded Scaffer's F6 Function	600
	28	Shifted and Rotated Lunacek Bi-Rastrigin Function	700
	29	Shifted and Rotated Non-Continuous Rastrigin's Function	800
Hybrid functions	30	Shifted and Rotated Levy Function	900
	31	Shifted and Rotated Schwefel's Function	1000
	32	Hybrid Function 1 (N = 3)	1100
	33	Hybrid Function 2 (N = 3)	1200
	34	Hybrid Function 3 (N = 3)	1300
	35	Hybrid Function 4 (N = 4)	1400
	36	Hybrid Function 5 (N = 4)	1500
	37	Hybrid Function 6 (N = 4)	1600
	38	Hybrid Function 6 (N = 5)	1700
	39	Hybrid Function 6 (N = 5)	1800
Composite functions	40	Hybrid Function 6 (N = 5)	1900
	41	Hybrid Function 6 (N = 6)	2000
	42	Composition Function 1 (N = 3)	2100
	43	Composition Function 2 (N = 3)	2200
	44	Composition Function 3 (N = 4)	2300
	45	Composition Function 4 (N = 4)	2400
	46	Composition Function 5 (N = 5)	2500
	47	Composition Function 6 (N = 5)	2600
	48	Composition Function 7 (N = 6)	2700
	49	Composition Function 8 (N = 6)	2800
	50	Composition Function 9 (N = 3)	2900
	51	Composition Function 10 (N = 3)	3000

optimal values of S_{max} and S_{min} . S_{max} determines the maximum range of the search and affects the exploration capability of the algorithm. The optimal values are chosen from $S_{max} = \{2.4, 2.2, 2.0, 1.8, 1.6\}$. S_{min} affects the exploitation ability of the algorithm. Here, we discuss six values of $S_{min} = \{1.6, 1.4, 1.2, 1.0, 0.8, 0.6\}$.

In this part, 23 classical test functions are used to compare the optimization results of m-MRFO for the above 30 (5 × 6) different parameter settings. Each test function was run 30 times independently, and a totally of 20,700 data were obtained. Due to the large amount of data, the experimental results are not compared specifically, but the differences are reflected by ranking the simulation results under different parameter settings by Friedman test.

Based on the experimental data, the Friedman test results are given for unimodal test functions and multimodal test functions, respectively. As shown in Figure 3, for the unimodal test functions, the algorithm mostly performs poorly when $S_{max} = \{2.4, 2.2\}$. And when $S_{max} = \{1.8, 1.6\}$, the algorithm is generally better. On the other hand, the algorithm performs poorly when S_{min} has a value that is too large or too small. The algorithms are generally better when $S_{min} = 1$. For the multimodal test functions, the influence of the parameter settings is exactly the opposite of the case for the unimodal test functions. In addition, to identify the optimal values of S_{max} and S_{min} , we present the results of the Friedman test considering all the tested functions. The results show that the algorithm performs best when $S_{max} = 2.4, S_{min} = 1.4$.

TABLE 3. m-MRFO variants with different improvement strategies.

Algorithm	ESP	ACP	DES
m-MRFO-1	Yes	No	No
m-MRFO-2	No	Yes	No
m-MRFO-3	No	No	Yes
m-MRFO	Yes	Yes	Yes

B. ANALYSIS OF M-MRFO IMPROVEMENT STRATEGIES

The improvement proposed in this paper for the original MRFO consists of three parts: search pool strategy, adaptive control parameter strategy and distribution estimation strategy. To evaluate the effectiveness of different improvement strategies, we propose three m-MRFO derivation algorithms with different improvement strategies, as shown in Table 3. m-MRFO-1 utilizes ESP to enhance the algorithm performance. m-MRFO-2 is used for evaluating the effectiveness of the ACP. The DES is fused into m-MRFO-3. The performance of the five algorithms is compared using the CEC2017 test suite. The five algorithms run under the same experimental parameters. Each function is run independently 51 times. The mean error results for each algorithm are listed in Table 4, and the Friedman test results for the five algorithms are given in the last row.

From the statistical results, it is clear that the m-MRFO with the complete improvement strategies performs best with a Friedman test ranking value of 1.25. The ranking values of

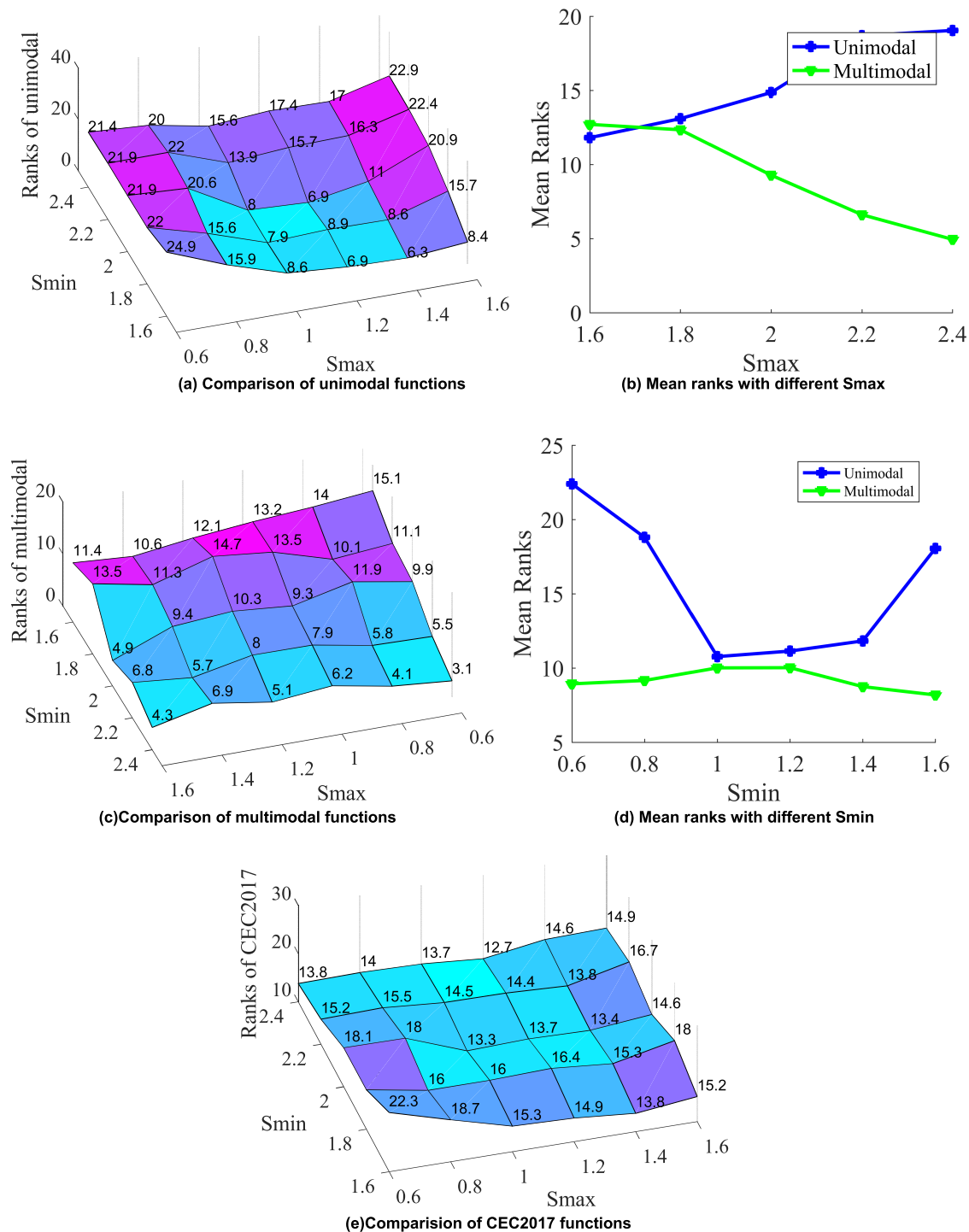


FIGURE 3. Effect of different Smax and Smin on algorithm performance.

the three derived algorithms with one improved strategy are also better than the original MRFO. The ranking values for the three derived algorithms are 3.39, 3.57 and 2.25, respectively. Thus, the impact of these three improvement strategies on the performance of the algorithm can be obtained in descending order: EDS > ESP > ACP. m-MRFO-3

performs the best among the three derived algorithms, proving that the utilization of DES can effectively enhance the performance of the algorithm. It generates offspring using the overall distribution information of the dominant population, which effectively avoids the deficiency of falling into local optimum caused by the population following only the

TABLE 4. Statistics of the results in CEC2017 30D test.

No	Index	MFRO	IMFRO-1	IMFRO-2	IMFRO-3	IMFRO
F24	Mean	6.61E+00	4.66E+00	2.18E+00	4.73E-06	3.91E-06
	Rank	5	4	3	2	1
F25	Mean	6.70E+01	5.14E+01	7.29E+01	2.18E+01	1.92E+01
	Rank	4	3	5	2	1
F26	Mean	1.46E+02	1.26E+02	1.27E+02	1.16E+02	9.82E+01
	Rank	5	3	4	2	1
F27	Mean	1.05E+01	6.95E+00	4.84E+00	1.78E+00	2.55E-01
	Rank	5	4	3	2	1
F28	Mean	2.12E+02	1.84E+02	1.96E+02	1.88E+02	1.23E+02
	Rank	5	2	4	3	1
F29	Mean	1.27E+02	1.12E+02	1.19E+02	1.08E+02	9.54E+01
	Rank	5	3	4	2	1
F30	Mean	1.77E+03	1.27E+03	1.39E+03	6.26E+01	2.17E+00
	Rank	5	3	4	2	1
F31	Mean	3.41E+03	3.44E+03	3.50E+03	3.51E+03	3.53E+03
	Rank	1	2	3	4	5
F32	Mean	9.20E+01	8.99E+01	8.72E+01	3.76E+01	4.23E+01
	Rank	5	4	3	1	2
F33	Mean	9.00E+04	7.23E+04	7.85E+04	1.11E+03	1.10E+03
	Rank	5	3	4	2	1
F34	Mean	1.36E+04	1.52E+04	1.88E+04	7.10E+01	6.93E+01
	Rank	3	4	5	2	1
F35	Mean	2.01E+03	1.60E+03	1.78E+03	6.02E+01	5.94E+01
	Rank	5	3	4	2	1
F36	Mean	8.64E+03	7.53E+03	7.37E+03	5.18E+01	4.12E+01
	Rank	5	4	3	2	1
F37	Mean	8.93E+02	7.96E+02	8.23E+02	8.13E+02	7.74E+02
	Rank	5	2	4	3	1
F38	Mean	3.36E+02	3.03E+02	2.92E+02	3.41E+02	2.50E+02
	Rank	4	3	2	5	1
F39	Mean	7.66E+04	7.16E+04	7.87E+04	3.52E+01	3.47E+01
	Rank	4	3	5	2	1
F40	Mean	8.42E+03	1.00E+04	8.89E+03	3.01E+01	2.93E+01
	Rank	3	5	4	2	1
F41	Mean	3.45E+02	3.28E+02	3.07E+02	3.38E+02	2.89E+02
	Rank	5	3	2	4	1
F42	Mean	3.09E+02	2.93E+02	2.99E+02	2.86E+02	2.76E+02
	Rank	5	3	4	2	1
F43	Mean	1.00E+02	1.65E+02	1.00E+02	1.00E+02	1.00E+02
	Rank	3	5	4	1	2
F44	Mean	4.95E+02	4.83E+02	4.77E+02	4.67E+02	4.52E+02
	Rank	5	4	3	2	1
F45	Mean	5.65E+02	5.56E+02	5.48E+02	5.55E+02	5.26E+02
	Rank	5	4	2	3	1
F46	Mean	3.89E+02	3.88E+02	3.88E+02	3.88E+02	3.87E+02
	Rank	5	3	4	2	1
F47	Mean	2.71E+03	2.14E+03	2.31E+03	2.04E+03	1.86E+03
	Rank	5	3	4	2	1
F48	Mean	5.56E+02	5.53E+02	5.55E+02	5.43E+02	5.42E+02
	Rank	5	3	4	2	1
F49	Mean	3.42E+02	3.37E+02	3.37E+02	3.20E+02	3.23E+02
	Rank	5	4	3	1	2
F50	Mean	8.27E+02	8.06E+02	7.89E+02	7.33E+02	6.81E+02
	Rank	5	4	3	2	1
F51	Mean	7.34E+03	5.96E+03	5.84E+03	2.04E+03	2.03E+03
	Rank	5	4	3	2	1
Friedman test		4.54	3.39	3.57	2.25	1.25

optimal individual. The performance of m-MRFO-1 in solving the multimodal test functions is similar to that of m-MRFO-3 with better results. This is due to the introduction of ESP, which effectively expands the search range by randomly selecting an individual in the ESP as a reference point, thus enhancing the algorithm's ability to solve multimodal problems. Figure 4 presents the ranking radar diagram of the five algorithms. It can be observed that the area enclosed by m-MRFO is the smallest, which visually indicates that m-MRFO has the best performance.

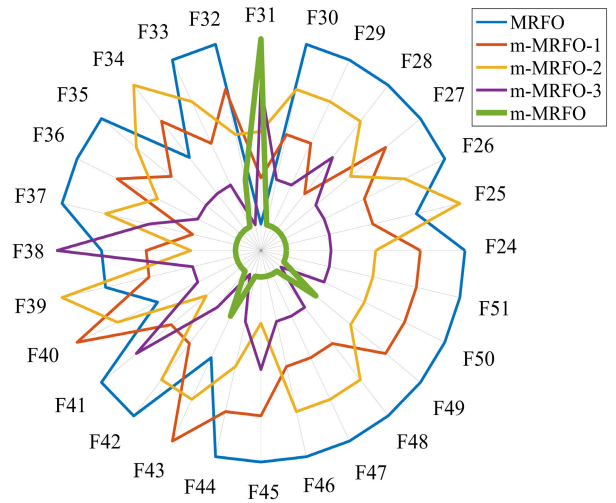


FIGURE 4. Mean error ranking result.

TABLE 5. Algorithms used for comparative analysis and their parameter settings.

Algorithm	Parameters
FPA	$P = 0.8$
BBO	$I = 1, E = 1, p = 0.03$
MFO	$a = -1$ (Linearly decreased over iterations)
MVO	$WEP_{max} = 1, WEP_{min} = 0.2, p = 6$
SCA	$a = 2$ (Linearly decreased over iterations)
SSA	$P = 0.2, C = 0.2$
PSO	$c_1 = 2, c_2 = 2, \omega_{max} = 0.9, \omega_{min} = 0.2$
WOA	$b = 1, a = 2$ (Linearly decreased over iterations)
GSA	$G_0 = 100, R_{norm} = 2, R_{power} = 1$

C. ANALYSIS OF CLASSICAL TEST FUNCTIONS TEST

In this part, the performance of m-MRFO on classical test functions will be verified. A comparison with m-MRFO is performed using other advanced algorithms. The parameter settings of each algorithm are given in Table 5. The experimental parameters and environment are consistent with Section IV A. The algorithms employed for comparison include flower pollination algorithm (FPA) [37], biogeography-based optimization (BBO) [9], moth-flame optimization (MFO) [38], multi-verse optimizer (MVO) [39], sine cosine algorithm (SCA) [15], sparrow search algorithm (SSA) [22], particle swarm optimization (PSO) [18], whale optimization algorithm (WOA) [21] and gravitational search algorithm (GSA) [13].

As shown in Table 6, m-MRFO performs best in solving five of the seven unimodal test functions (F1-F4, F7). The m-MRFO provides the second-best solution in solving F5 and F6. It is due to the replacement of randomly generated reference points using ESP, which further enhances the

TABLE 6. Statistical results of the nine comparison algorithms for classical functions.

No	Index	m-MRFO	FPA	BBO	MFO	MVO	SCA	SSA	PSO	WOA	GSA
F1	Mean	1.47E-270	3.93E+03	2.90E+00	1.11E+03	2.01E+00	9.78E+01	5.22E-145	4.01E-02	3.19E-46	4.00E+01
	Std	0.00E+00	8.24E+02	6.06E-01	3.04E+03	5.42E-01	1.38E+02	1.97E-144	1.02E-01	1.15E-45	5.19E+01
F2	Mean	2.93E-135	5.07E+01	5.35E-01	3.04E+01	1.62E+01	1.68E-01	5.15E-59	1.30E-01	2.19E-28	8.87E-02
	Std	9.31E-135	9.68E+00	6.88E-02	1.95E+01	3.56E+01	1.64E-01	2.82E-58	1.16E-01	8.74E-28	2.61E-01
F3	Mean	1.47E-263	4.36E+03	1.12E+03	2.55E+04	3.90E+02	1.17E+04	4.27E-90	7.26E+02	4.43E+04	7.89E+02
	Std	0.00E+00	1.22E+03	5.19E+02	1.22E+04	1.93E+02	4.72E+03	2.34E-89	3.53E+02	1.16E+04	2.10E+02
F4	Mean	4.51E-135	3.51E+01	1.57E+00	5.91E+01	2.40E+00	4.09E+01	4.04E-60	5.10E+00	6.63E+01	6.74E+00
	Std	1.61E-134	4.27E+00	2.07E-01	9.80E+00	9.42E-01	1.24E+01	2.21E-59	1.55E+00	1.78E+01	2.13E+00
F5	Mean	2.40E+01	1.73E+06	2.04E+02	3.87E+04	4.95E+02	2.72E+05	1.79E-04	9.99E+01	2.83E+01	1.17E+02
	Std	4.10E-01	8.51E+05	1.44E+02	3.69E+04	5.99E+02	4.75E+05	2.85E-04	5.98E+01	3.65E-01	1.02E+02
F6	Mean	3.75E-07	4.34E+03	2.85E+00	8.08E+02	2.06E+00	9.10E+01	5.29E-08	5.57E-02	1.99E+00	3.93E+01
	Std	2.53E-07	1.10E+03	5.20E-01	2.54E+03	5.43E-01	9.61E+01	8.58E-08	1.80E-01	6.68E-01	4.42E+01
F7	Mean	1.19E-04	9.19E-01	1.14E-02	3.14E+00	3.20E-02	2.37E-01	2.76E-04	2.76E-02	4.21E-03	3.95E-02
	Std	1.06E-04	4.13E-01	3.88E-03	6.78E+00	1.36E-02	2.41E-01	1.92E-04	1.05E-02	3.81E-03	1.85E-02
Mean Ranks		1.33	9.00	5.67	8.83	5.33	7.83	1.67	4.17	5.50	5.67
F8	Mean	-7.51E+03	-6.72E+03	-8.69E+03	-9.04E+03	-7.58E+03	-3.75E+03	-8.03E+03	-9.37E+03	-1.01E+83	-2.71E+03
	Std	6.62E+02	2.11E+02	5.78E+02	7.62E+02	7.49E+02	3.32E+02	6.87E+02	8.90E+02	5.47E+83	4.54E+02
F9	Mean	0.00E+00	1.97E+02	3.98E+01	1.64E+02	1.22E+02	6.20E+01	0.00E+00	5.64E+01	7.58E-15	1.75E+01
	Std	0.00E+00	1.42E+01	1.13E+01	4.06E+01	2.20E+01	4.24E+01	0.00E+00	1.67E+01	2.47E-14	5.70E+00
F10	Mean	8.88E-16	1.31E+01	6.24E-01	1.04E+01	1.95E+00	1.09E+01	8.88E-16	1.49E+00	6.34E-15	1.85E-02
	Std	0.00E+00	1.70E+00	9.18E-02	7.23E+00	5.36E-01	8.41E+00	0.00E+00	5.87E-01	2.76E-15	1.01E-01
F11	Mean	0.00E+00	3.72E+01	1.02E+00	4.86E+00	9.55E-01	1.65E+00	0.00E+00	5.16E-02	4.50E-02	1.17E+02
	Std	0.00E+00	9.10E+00	1.37E-02	1.65E+01	5.40E-02	8.59E-01	0.00E+00	4.27E-02	1.19E-01	2.28E+01
F12	Mean	6.79E-09	1.11E+05	7.81E-03	3.41E+07	1.93E+00	2.40E+05	1.07E-09	6.60E-01	1.30E-01	1.92E+00
	Std	4.71E-09	1.22E+05	2.33E-03	8.85E+07	1.30E+00	5.56E+05	2.15E-09	8.21E-01	1.25E-01	9.84E-01
F13	Mean	2.02E+00	1.50E+06	1.35E-01	1.37E+07	2.50E-01	7.10E+05	3.51E-08	7.05E-01	1.32E+00	1.60E+01
	Std	1.37E+00	1.15E+06	3.30E-02	7.49E+07	1.56E-01	1.59E+06	6.14E-08	1.05E+00	4.06E-01	6.29E+00
Mean Ranks		2.20	9.20	4.80	7.60	6.40	8.00	2.00	4.60	2.60	7.00
F14	Mean	9.98E-01	1.00E+00	4.12E+00	1.43E+00	9.98E-01	1.53E+00	6.45E+00	9.98E-01	2.31E+00	6.05E+00
	Std	1.37E-16	1.97E-02	4.59E+00	1.26E+00	4.80E-11	8.91E-01	5.60E+00	5.83E-17	2.53E+00	4.56E+00
F15	Mean	3.38E-04	8.09E-04	1.68E-03	9.20E-04	2.75E-03	1.08E-03	3.32E-04	7.38E-04	1.20E-03	5.20E-03
	Std	1.67E-04	1.50E-04	3.56E-03	2.81E-04	5.98E-03	3.95E-04	9.17E-05	4.68E-04	7.66E-04	2.77E-03
F16	Mean	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Std	6.52E-16	1.99E-07	1.53E-12	6.78E-16	6.12E-07	7.95E-05	5.68E-16	6.12E-16	1.91E-09	5.13E-16
F17	Mean	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.99E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	Std	0.00E+00	1.78E-09	1.48E-10	0.00E+00	2.63E-07	1.17E-03	0.00E+00	0.00E+00	2.17E-05	0.00E+00
F18	Mean	3.00E+00	3.00E+00	4.80E+00	3.00E+00	3.00E+00	3.00E+00	3.90E+00	3.00E+00	3.00E+00	3.00E+00
	Std	1.57E-15	1.35E-06	6.85E+00	2.06E-15	5.07E-06	1.53E-04	4.93E+00	1.33E-15	6.46E-05	2.20E-15
F19	Mean	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.85E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
	Std	2.67E-15	7.98E-07	2.17E-15	2.71E-15	2.84E-06	3.40E-03	2.39E-15	2.68E-15	3.60E-03	1.16E-03
F20	Mean	-3.28E+00	-3.29E+00	-3.28E+00	-3.23E+00	-3.25E+00	-2.95E+00	-3.28E+00	-3.28E+00	-3.27E+00	-3.32E+00
	Std	5.83E-02	1.84E-02	5.83E-02	5.13E-02	6.02E-02	2.54E-01	5.83E-02	5.70E-02	6.90E-02	1.47E-15
F21	Mean	-9.98E+00	-1.00E+01	-6.32E+00	-7.23E+00	-7.21E+00	-2.69E+00	-9.13E+00	-5.82E+00	-8.34E+00	-6.17E+00
	Std	9.31E-01	1.07E-01	3.51E+00	3.29E+00	3.10E+00	1.93E+00	2.07E+00	3.65E+00	2.80E+00	3.79E+00
F22	Mean	-1.00E+01	-9.98E+00	-6.52E+00	-8.67E+00	-7.71E+00	-4.01E+00	-9.52E+00	-7.91E+00	-7.25E+00	-1.04E+01
	Std	1.35E+00	3.26E-01	3.50E+00	2.98E+00	3.45E+00	1.53E+00	2.01E+00	3.40E+00	3.51E+00	1.09E+01
F23	Mean	-1.05E+01	-9.89E+00	-6.42E+00	-8.07E+00	-8.98E+00	-4.15E+00	-1.02E+01	-7.56E+00	-6.56E+00	-1.00E+01
	Std	1.51E-15	5.31E-01	3.69E+00	3.35E+00	3.17E+00	1.89E+00	1.37E+00	3.74E+00	3.42E+00	2.04E+00
Mean Ranks		2.08	5.08	6.25	5.00	6.50	8.92	3.25	3.50	6.83	4.92
Friedman rank		2.28	7.00	5.87	6.85	6.17	8.43	2.89	4.20	5.57	5.74

exploitation capability. SSA provides satisfactory answers on F5 and F6. From the statistics of the multimodal test functions, m-MFRO performs the best in three test functions (F9-F11). However, m-MFRO gives unsatisfactory results in solving F8 and F13. The analysis of the results of the low-dimensional test functions shows that m-MFRO achieves the best solution on 9 of the 10 test functions (4 first and 5 second). PSO, SSA and MFO are in the second, third and fourth positions. In addition, we evaluate the performance of the algorithms using the Friedman test. All algorithms are ranked according to the mean value. The results show that m-MRFO ranks first with a ranking value of 2.28. The statistical results show that m-MRFO has the best performance compared to the nine algorithms mentioned above for solving the classical test functions.

D. ANALYSIS OF THE CEC 2017 TEST

The classical test functions can verify the performance of algorithms to a certain extent. However, with the development of intelligent optimization algorithms, more and more algorithms have better performance in the classical test functions. In order to further verify the superiority of the m-MFRO proposed in this paper, more complex test functions are needed for testing. Therefore, we use the IEEE CEC2017 test suite to further validate the performance of the improved algorithms. This test set consists of more complex and difficult test functions. It has been widely used to evaluate the performance of various newly proposed and improved algorithms. In this part, eight recently proposed algorithms are evaluated for comparison with m-MRFO. These state-of-the-art algorithms include artificial ecosystem-based

TABLE 7. Algorithms used for comparative analysis and their parameter settings.

Algorithm	Parameters
AEO	No parameters
HHO	$\beta = 1.5, E_0 \in [-1, 1]$
VCS	$\lambda = 0.5, \sigma = 0.3$
AOA	$Mop_{max} = 1, Mop_{min} = 0.2, C = 1, \alpha = 5, Mu = 0.499$
SMA	$z = 0.03$
JS	$\eta = 4$
PFA	$u_1 = -1 + 2rand, u_2 = -1 + 2rand$
TSA	$x_{min} = 1, x_{max} = 4$

optimization (AEO) [40], harris hawks optimization (HHO) [41], virus colony search (VCS) [42], arithmetic optimization algorithm (AOA) [43], slime mould algorithm (SMA) [44], jellyfish search (JS) [45], pathfinder algorithm (PFA) [46] and tunicate swarm algorithm (TSA) [47]. All algorithm parameters are set the same as the original literature, as shown in Table 7. In this paper, the performance of m-MRFO is comprehensively evaluated by numerical analysis, convergence analysis, stability analysis, Wilcoxon test, Friedman test, and Iman-Davenport test.

Table 8 lists the numerical statistics for each algorithm independently solving the CEC2017 test suite 51 times. Analysis of Table 8 shows that although m-MRFO does not achieve the optimal value of 0 for the unimodal test function F24, it provides the best solution of the nine algorithms. This demonstrates the superiority of m-MRFO in solving pathological functions and once again validates that improved strategies can effectively improve the exploitation capability. In the multimodal test functions F25-F31, all algorithms perform differently. The m-MRFO performs best on F25 and F30. SMA provides the best solutions for F26, F28 and F31. JS achieves the best answers for F27 and F29. The average rank of m-MRFO in solving the multimodal test functions is better than all the comparison algorithms, which indicates the competitive exploration capability of m-MRFO. In the hybrid functions and composite functions, m-MRFO outperforms the comparison algorithm overall. Specifically, m-MRFO achieves the best results on eight of the nine hybrid functions. m-MRFO ranks in at least the top three of all 11 composite functions. The results for the hybrid and composite functions illustrate that m-MRFO achieves a good balance between exploitation and exploration behaviors and has a strong potential for solving complex real-world optimization problems.

To analyze the distribution characteristics of the solutions solved by the improved algorithms, the box diagrams are shown in Figure 5 based on the results of each algorithm solving the test function 51 times independently. For each algorithm, the center mark of each box indicates the median of the results of 51 solving functions, the bottom and top edges of the box indicate first and third-degree points, and the

symbol “+” indicates bad values that are not inside the box. As can be seen from Figure 5, for F24, F25, F30, F35, F37 and F42, there are no bad values for m-MRFO, which indicates that the distribution of solutions obtained from m-MRFO is more concentrated and m-MRFO has better stability. For other test functions with some bad values, the distribution of the m-MRFO solutions is also more concentrated compared to the comparison algorithm. In conclusion, the variance of the m-MRFO solving test functions is much lower and the stability is better than that of the comparison algorithm.

Convergence speed and convergence accuracy are important indicators of algorithm performance. Figure 6 shows the mean error convergence curves for each algorithm solving the test functions. It can be seen that m-MRFO has a faster convergence speed and better convergence accuracy. In the convergence curve of the unimodal test function F24, m-MRFO has the fastest convergence speed and the highest accuracy. This indicates that m-MRFO has better exploitation capability. The convergence curves of the multimodal functions show that m-MRFO can explore well and thus avoid the local optimum. In most hybrid and composite functions, m-MRFO can achieve a better result quickly. This demonstrates that m-MRFO transitions well from exploration to exploitation, balancing the behaviors of exploration and exploitation in the search space.

The literature [48], [49] show that analyzing algorithm performance based on mean values alone is not sufficient. To avoid coincidence in the test, we adopt a variety of statistical analyses to verify algorithm performance.

In this paper, the Wilcoxon signed-rank test is first used to verify whether m-MRFO is significantly different from the comparison algorithms in statistical sense. Table 8 presents the results of the Wilcoxon signed rank test for each algorithm and m-MRFO at the significance level $\alpha = 0.05$. In the Table 9, the symbol “+” indicates that m-MRFO outperforms the comparison algorithm. The symbol “-” indicates that m-MRFO underperforms the comparison algorithm. The symbol “=” indicates that m-MRFO performs similarly to the comparison algorithm. The symbol “R+” is a positive rank value indicating the extent to which m-MRFO is better than the comparison algorithm, and “R-” indicates the opposite result. Counting the number of “+/-” for each algorithm, it can be seen that m-MRFO has the best performance among the algorithms involved in the test. m-MRFO outperforms all the comparison algorithms on at least 15 functions, which shows that m-MRFO is statistically significantly different from the other algorithms.

In addition, to check the differences and rankings between several algorithms, another non-parametric multiple comparison method is used in this paper: the Friedman test. A lower ranking in the Friedman test means a better performance, and the Friedman test compares three aspects: mean, standard deviation and time. As shown in Table 9, the probability of significance for the three aspects of the Friedman

TABLE 8. Statistical results of the eight comparison algorithms for CEC2017.

No	Index	AEO	HHO	VCS	AOA	SMA	JS	PFA	TSA	m-MRFO
F24	Mean	9.01E-04	1.68E+03	7.32E+03	6.91E+04	4.94E-01	4.04E+04	4.66E+04	3.83E+04	3.91E-06
	Std	1.89E-03	7.95E+02	5.58E+03	1.15E+04	3.18E-01	4.32E+03	1.22E+04	1.19E+04	9.28E-07
	Rank	2	4	5	9	3	7	8	6	1
F25	Mean	6.65E+01	1.23E+02	7.39E+01	7.61E+03	8.99E+01	1.16E+02	9.80E+01	1.62E+03	1.92E+01
	Std	3.72E+01	3.33E+01	3.48E+01	2.45E+03	5.12E+00	4.57E+00	1.77E+01	1.40E+03	2.72E+01
	Rank	2	7	3	9	4	6	5	8	1
F26	Mean	1.34E+02	2.05E+02	1.09E+02	2.95E+02	8.17E+01	9.24E+01	1.14E+02	2.76E+02	9.82E+01
	Std	3.17E+01	3.62E+01	2.56E+01	3.20E+01	1.99E+01	1.59E+01	3.11E+01	4.09E+01	2.84E+01
	Rank	6	7	4	9	1	2	5	8	3
F27	Mean	1.67E+01	5.62E+01	8.72E-01	6.21E+01	7.35E-01	1.28E-02	1.47E+01	6.15E+01	2.55E-01
	Std	5.63E+00	5.92E+00	8.88E-01	6.71E+00	3.21E-01	3.34E-02	4.99E+00	1.43E+01	7.59E-01
	Rank	6	7	4	9	3	1	5	8	2
F28	Mean	2.75E+02	4.98E+02	1.79E+02	6.00E+02	1.18E+02	1.22E+02	1.34E+02	4.83E+02	1.23E+02
	Std	6.77E+01	6.57E+01	1.20E+02	5.66E+01	2.40E+01	3.57E+01	3.12E+01	7.78E+01	3.15E+01
	Rank	6	8	5	9	1	2	4	7	3
F29	Mean	1.13E+02	1.40E+02	9.19E+01	2.25E+02	9.39E+01	8.63E+01	9.97E+01	2.34E+02	9.54E+01
	Std	2.33E+01	2.13E+01	2.25E+01	2.67E+01	2.03E+01	1.54E+01	2.66E+01	3.99E+01	2.64E+01
	Rank	6	7	2	8	3	1	5	9	4
F30	Mean	2.49E+03	4.69E+03	5.22E+02	4.50E+03	9.95E+02	4.79E+00	2.28E+02	8.57E+03	2.17E+00
	Std	9.16E+02	8.28E+02	1.44E+03	7.24E+02	1.22E+03	4.35E+00	1.83E+02	3.01E+03	2.96E+00
	Rank	6	8	4	7	5	2	3	9	1
F31	Mean	3.37E+03	4.35E+03	4.00E+03	5.51E+03	3.04E+03	6.50E+03	4.98E+03	5.55E+03	3.53E+03
	Std	5.42E+02	7.25E+02	1.03E+03	5.83E+02	5.06E+02	6.85E+02	9.01E+02	6.07E+02	5.25E+02
	Rank	2	5	4	7	1	9	6	8	3
F32	Mean	1.09E+02	1.61E+02	1.25E+02	1.72E+03	1.16E+02	6.80E+01	1.91E+02	2.23E+03	4.23E+01
	Std	3.10E+01	4.86E+01	2.42E+01	9.74E+02	4.33E+01	2.66E+01	5.28E+01	1.69E+03	2.36E+01
	Rank	3	6	5	8	4	2	7	9	1
F33	Mean	3.74E+04	7.61E+06	3.51E+06	6.27E+09	1.31E+06	5.84E+05	1.88E+06	8.88E+08	1.10E+03
	Std	3.35E+04	4.21E+06	1.89E+06	2.56E+09	1.09E+06	2.34E+05	1.97E+06	1.07E+09	3.34E+02
	Rank	2	7	6	9	4	3	5	8	1
F34	Mean	1.51E+04	1.51E+05	9.00E+04	3.80E+04	2.71E+04	3.65E+03	7.54E+04	1.75E+08	6.93E+01
	Std	1.46E+04	9.05E+04	4.08E+04	1.71E+04	2.64E+04	2.41E+03	4.12E+04	4.14E+08	2.25E+01
	Rank	3	8	7	5	4	2	6	9	1
F35	Mean	1.40E+02	3.82E+04	1.49E+04	5.72E+04	4.71E+04	4.53E+03	3.00E+04	3.73E+05	5.94E+01
	Std	4.36E+01	4.25E+04	1.80E+04	4.92E+04	2.84E+04	2.23E+03	2.94E+04	6.73E+05	4.22E+00
	Rank	2	6	4	8	7	3	5	9	1
F36	Mean	4.10E+03	6.86E+04	1.59E+04	2.35E+04	1.99E+04	1.49E+03	3.35E+04	2.48E+07	4.12E+01
	Std	4.49E+03	4.86E+04	1.82E+04	1.22E+04	1.57E+04	1.34E+03	1.77E+04	7.80E+07	1.77E+01
	Rank	3	8	4	6	5	2	7	9	1
F37	Mean	1.03E+03	1.55E+03	8.28E+02	1.98E+03	8.18E+02	1.01E+03	1.00E+03	1.43E+03	7.74E+02
	Std	3.11E+02	3.56E+02	2.45E+02	5.09E+02	2.83E+02	2.83E+02	2.63E+02	2.92E+02	2.67E+02
	Rank	6	8	3	9	2	5	4	7	1
F38	Mean	4.06E+02	7.48E+02	2.56E+02	9.12E+02	4.34E+02	2.05E+02	3.77E+02	6.06E+02	2.50E+02
	Std	1.84E+02	2.19E+02	1.22E+02	2.67E+02	1.64E+02	6.80E+01	1.71E+02	2.30E+02	1.44E+02
	Rank	5	8	3	9	6	1	4	7	2
F39	Mean	1.70E+04	6.90E+05	2.30E+05	1.29E+06	3.75E+05	1.65E+05	2.75E+05	2.08E+06	3.47E+01
	Std	1.28E+04	8.77E+05	1.74E+05	1.60E+06	3.55E+05	7.51E+04	2.82E+05	4.09E+06	2.39E+00
	Rank	2	7	4	8	6	3	5	9	1
F40	Mean	1.94E+03	1.46E+05	1.49E+04	1.08E+06	3.00E+04	3.65E+03	4.45E+04	1.11E+07	2.93E+01
	Std	3.07E+03	1.42E+05	1.87E+04	1.39E+05	2.11E+04	1.94E+03	3.91E+04	3.45E+07	2.40E+00
	Rank	2	7	4	8	5	3	6	9	1
F41	Mean	3.95E+02	6.71E+02	2.22E+02	6.94E+02	3.59E+02	2.95E+02	4.61E+02	7.24E+02	2.89E+02
	Std	1.62E+02	2.01E+02	6.41E+01	1.54E+02	1.59E+02	6.47E+01	1.52E+02	2.09E+02	1.56E+02
	Rank	5	7	1	8	4	3	6	9	2
F42	Mean	3.26E+02	4.06E+02	2.78E+02	4.87E+02	2.93E+02	2.80E+02	2.90E+02	4.68E+02	2.76E+02
	Std	3.26E+01	3.51E+01	2.98E+01	5.23E+01	2.17E+01	1.23E+01	2.62E+01	4.96E+01	2.18E+01
	Rank	6	7	2	9	5	3	4	8	1
F43	Mean	1.73E+02	2.39E+03	1.07E+02	5.13E+03	2.90E+03	1.02E+02	2.08E+02	4.47E+03	1.00E+02
	Std	5.11E+02	2.37E+03	6.10E+00	1.21E+03	1.36E+03	1.29E+00	7.61E+02	2.09E+03	5.84E-01
	Rank	4	6	3	9	7	2	5	8	1
F44	Mean	5.41E+02	7.05E+02	4.62E+02	9.68E+02	4.35E+02	4.32E+02	4.85E+02	7.86E+02	4.52E+02
	Std	4.86E+01	7.35E+01	2.35E+01	9.10E+01	1.95E+01	1.70E+01	4.09E+01	8.15E+01	3.25E+01
	Rank	6	7	4	9	2	1	5	8	3
F45	Mean	6.41E+02	8.26E+02	5.32E+02	1.14E+03	5.30E+02	5.00E+02	5.24E+02	8.47E+02	5.26E+02
	Std	7.63E+01	7.42E+01	3.20E+01	1.09E+02	2.95E+01	1.99E+01	3.72E+01	8.08E+01	3.56E+01
	Rank	6	7	5	9	4	1	2	8	3
F46	Mean	3.94E+02	4.11E+02	3.78E+02	1.67E+03	3.88E+02	3.94E+02	3.97E+02	7.61E+02	3.87E+02
	Std	1.47E+01	1.87E+01	3.35E+00	4.55E+02	1.69E+00	8.26E+00	1.72E+01	3.02E+02	2.97E+00
	Rank	4	7	1	9	3	5	6	8	2
F47	Mean	2.62E+03	3.94E+03	1.55E+03	6.40E+03	1.98E+03	9.93E+02	2.12E+03	5.01E+03	1.86E+03
	Std	1.10E+03	1.10E+03	9.39E+02	7.22E+02	3.42E+02	9.39E+02	7.10E+02	8.76E+02	7.48E+02
	Rank	6	7	2	9	4	1	5	8	3
F48	Mean	5.92E+02	6.05E+02	5.00E+02	1.34E+03	5.11E+02	5.44E+02	5.46E+02	7.30E+02	5.42E+02
	Std	3.05E+01	4.00E+01	1.51E-04	2.14E+02	1.17E+01	1.18E+01	2.85E+01	9.92E+01	2.06E+01
	Rank	6	7	1	9	2	4	5	8	3
F49	Mean	3.49E+02	4.62E+02	4.96E+02	2.95E+03	4.46E+02	4.91E+02	4.31E+02	1.27E+03	3.23E+02
	Std	5.89E+01	2.60E+01	5.05E+00	6.15E+02	2.89E+01	9.81E+00	1.90E+01	4.52E+02	4.46E+01
	Rank	2	5	7	9	4	6	3	8	1
F50	Mean	9.30E+02	1.32E+03	5.69E+02	2.43E+03	7.75E+02	6.10E+02	1.03E+03	1.58E+03	6.81E+02
	Std	2.09E+02	2.56E+02	1.43E+02	5.22E+02	1.83E+02	1.12E+02	2.35E+02	4.08E+02	1.85E+02
	Rank	5	7	1	9	4	2	6	8	3
F51	Mean	9.26E+03	1.01E+06	9.38E+03	1.47E+07	1.27E+04	4.27E+03	4.09E+05	1.33E+07	2.03E+03
	Std	4.95E+03	6.08E+05	7.22E+03	1.01E+07	5.29E+03	6.66E+02	4.56E+05	1.07E+07	6.45E+01
	Rank	3	7	4	9	5	2	6	8	1

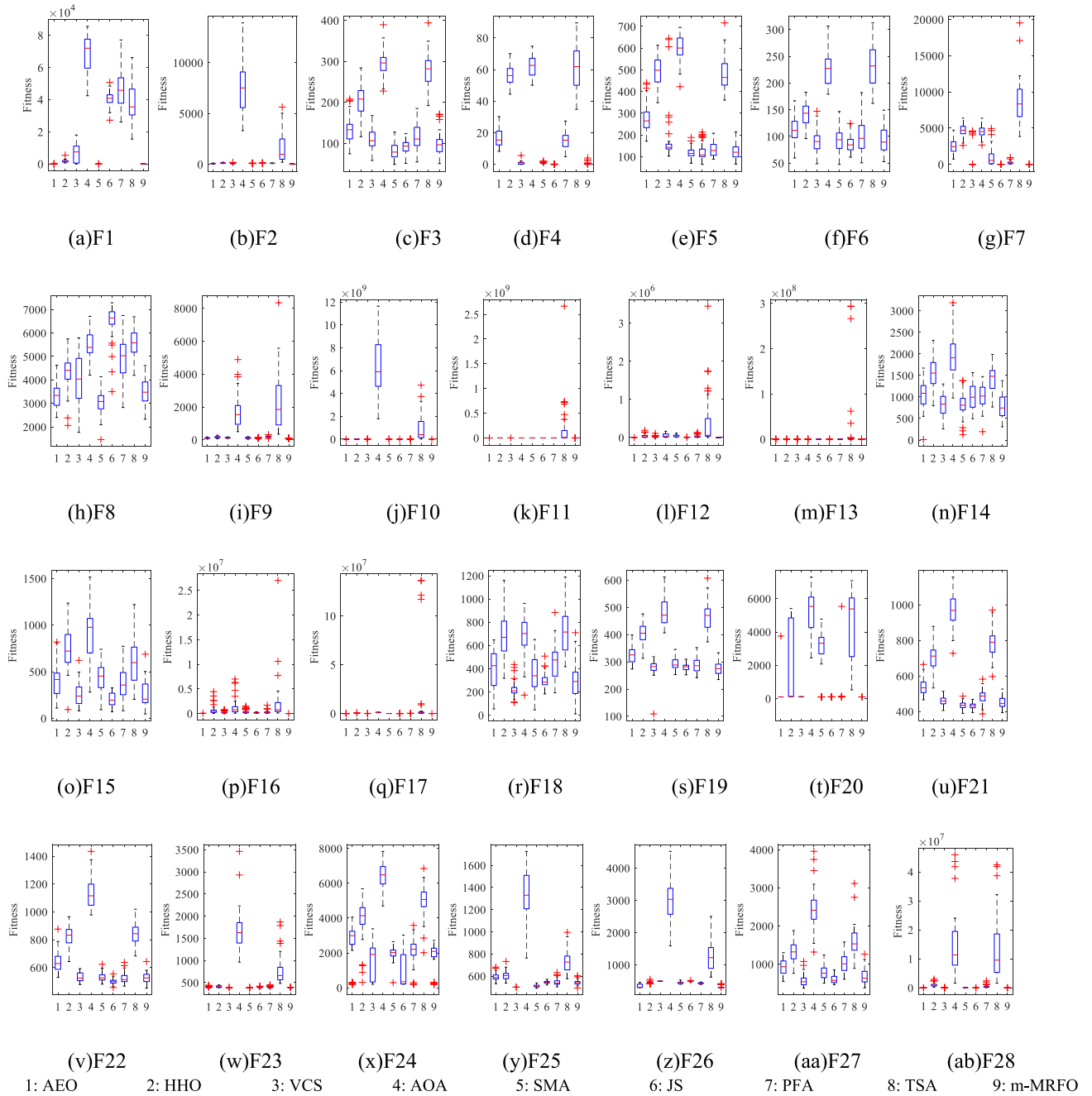


FIGURE 5. Boxes diagram of CEC2017.

test is significantly less than 0.05. Therefore, the hypothesis is rejected and the performance of the seven compared algorithms is significantly different. In terms of mean and time, m-MRFO performed best. As for standard deviation, m-MRFO ranks behind JS, but still outperforms the rest of the algorithms. To further analyze the differences between the algorithms, a post-hoc Iman-Davenport’s test was employed. The Iman-Davenport test is based on the F-distribution with $(k - 1)$ and $(k - 1)(N - 1)$ degrees of freedom. It can be seen from Table 10, the F distributions with 8 and 216 degrees

of freedom in terms of mean, standard deviation and time are: 67.05, 31.49 and 989.06. The corresponding p-values are 7.7715e-16, 9.4705e-12 and 0, respectively. To find differences in all algorithms, critical difference (CD) based on the Nemenyi test was used. The critical value q_{α} is 2.3053, so the CD is 1.6873. A post-hoc test concludes that if the difference in Friedman ranking values between the two algorithms is less than the CD value, there is no significant difference between the two algorithms; conversely, there is a significant difference between the algorithms. Figure 7 provides a visual

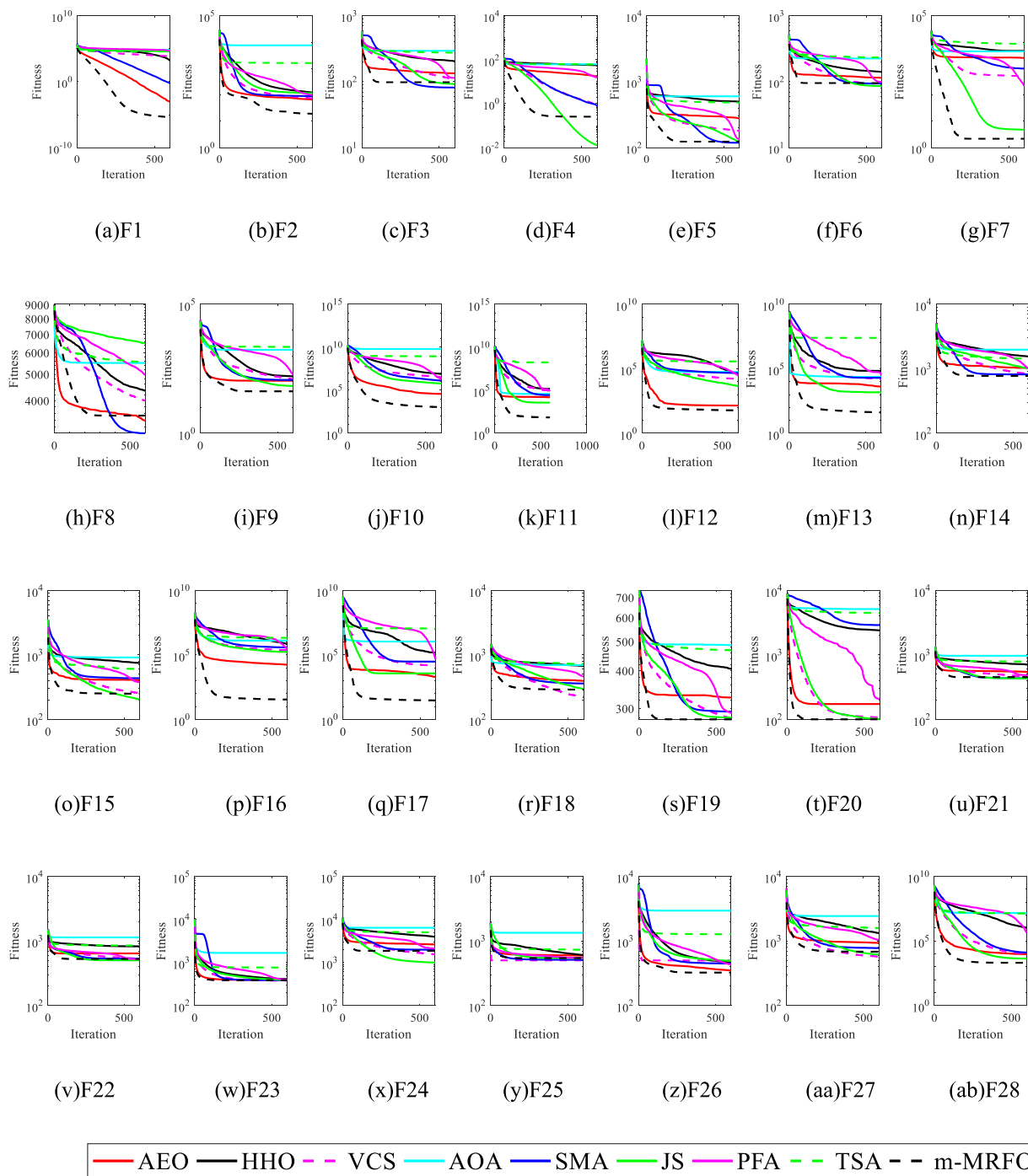


FIGURE 6. Convergence curves of CEC2017.

representation of the significant differences between the nine algorithms, with similarly performing algorithms connected by thick solid lines of length CD values. As shown in Figure 7, m-MRFO ranks first, second and first in Mean, Std and Time, respectively. The m-MRFO and JS have similar performance in terms of Mean and Std. In addition, m-MRFO and VCS have similar time costs. In conclusion, m-MRFO shows a superior performance.

E. ANALYSIS OF ENGINEERING DESIGN PROBLEMS TEST

The engineering design problem is a nonlinear optimization problem with complex geometry, many design variables, and many real engineering constraints. The performance of m-MRFO is evaluated by solving real-world engineering problems. Considering that these engineering design problems are constrained optimization problems involving inequality and equality, we use penalty functions

TABLE 9. Wilcoxon sign rank test results for CEC2017.

No	AEO				HHO				VCS				AOA				
	P-value	R+	R-	Win	P-value	R+	R-	Win	P-value	R+	R-	Win	P-value	R+	R-	Win	
F24	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F25	5.15E-08	1244	82	+	5.15E-10	1326	0	+	1.50E-08	1267	59	+	5.15E-10	1326	0	+	
F26	3.96E-07	1204	122	+	5.46E-10	1325	1	+	5.47E-02	868	458	=	5.15E-10	1326	0	+	
F27	5.15E-10	1326	0	+	5.15E-10	1326	0	+	3.65E-06	1157	169	+	5.15E-10	1326	0	+	
F28	5.15E-10	1326	0	+	5.15E-10	1326	0	+	8.92E-05	1081	245	+	5.15E-10	1326	0	+	
F29	1.92E-03	994	332	+	4.94E-09	1287	39	+	5.49E-01	599	727	=	5.15E-10	1326	0	+	
F30	5.15E-10	1326	0	+	5.15E-10	1326	0	+	4.90E-02	453	873	-	5.15E-10	1326	0	+	
F31	6.62E-02	467	859	=	4.16E-07	1203	123	+	1.20E-02	931	395	+	5.46E-10	1325	1	+	
F32	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.46E-10	1325	1	+	5.15E-10	1326	0	+	
F33	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F34	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F35	6.15E-10	1323	3	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F36	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F37	3.29E-05	1106	220	+	5.80E-10	1324	2	+	1.77E-01	807	519	=	5.80E-10	1324	2	+	
F38	5.14E-05	1095	231	+	5.80E-10	1324	2	+	8.59E-01	682	644	=	5.15E-10	1326	0	+	
F39	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F40	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F41	4.51E-03	966	360	+	1.40E-09	1309	17	+	8.92E-03	384	942	-	7.35E-10	1320	6	+	
F42	8.65E-09	1277	49	+	5.15E-10	1326	0	+	2.77E-01	779	547	=	5.15E-10	1326	0	+	
F43	4.72E-04	1036	290	+	5.15E-10	1326	0	+	5.85E-09	1284	42	+	5.15E-10	1326	0	+	
F44	5.80E-10	1324	2	+	5.15E-10	1326	0	+	7.80E-02	851	475	=	5.15E-10	1326	0	+	
F45	1.40E-09	1309	17	+	5.15E-10	1326	0	+	3.30E-01	767	559	=	5.15E-10	1326	0	+	
F46	1.23E-02	930	396	+	1.57E-09	1307	19	+	8.18E-09	48	1278	-	5.15E-10	1326	0	+	
F47	4.72E-04	1036	290	+	2.50E-09	1299	27	+	5.35E-02	457	869	=	5.15E-10	1326	0	+	
F48	7.74E-09	1279	47	+	5.15E-10	1326	0	+	5.46E-10	1	1325	-	5.15E-10	1326	0	+	
F49	1.80E-03	996	330	+	5.46E-10	1325	1	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F50	8.20E-07	1189	137	+	8.77E-10	1317	9	+	3.77E-03	354	972	-	5.15E-10	1326	0	+	
F51	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
+/-/=		27/0/1				28/0/0				15/5/8				28/0/0			
No	SMA				JS				PFA				TSA				
	P-value	R+	R-	Win	P-value	R+	R-	Win	P-value	R+	R-	Win	P-value	R+	R-	Win	
F24	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F25	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F26	5.06E-04	292	1034	-	2.65E-01	544	782	=	7.14E-03	950	376	+	5.15E-10	1326	0	+	
F27	1.83E-06	1172	154	+	6.60E-01	710	616	=	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F28	4.31E-01	579	747	=	5.12E-01	593	733	=	7.34E-02	854	472	=	5.15E-10	1326	0	+	
F29	8.96E-01	677	649	=	9.52E-02	485	841	=	4.31E-01	747	579	=	5.15E-10	1326	0	+	
F30	5.15E-10	1326	0	+	4.40E-04	1038	288	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F31	1.21E-04	253	1073	-	5.15E-10	1326	0	+	1.25E-09	1311	15	+	5.15E-10	1326	0	+	
F32	1.05E-09	1314	12	+	6.82E-06	1143	183	+	5.46E-10	1325	1	+	5.15E-10	1326	0	+	
F33	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F34	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F35	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F36	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F37	4.59E-01	742	584	=	2.57E-04	1053	273	+	5.35E-05	1094	232	+	1.67E-09	1306	20	+	
F38	1.20E-06	1181	145	+	5.01E-02	454	872	=	8.47E-04	1019	307	+	5.53E-09	1285	41	+	
F39	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F40	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F41	4.19E-02	880	446	+	6.80E-01	707	619	=	3.65E-06	1157	169	+	9.31E-10	1316	10	+	
F42	7.40E-04	1023	303	+	1.63E-01	812	514	=	6.20E-03	955	371	+	5.15E-10	1326	0	+	
F43	5.15E-10	1326	0	+	3.73E-09	1292	34	+	4.16E-07	1203	123	+	5.15E-10	1326	0	+	
F44	8.21E-03	381	945	-	7.92E-04	305	1021	+	6.52E-05	1089	237	+	5.15E-10	1326	0	+	
F45	5.67E-01	724	602	=	1.53E-04	259	1067	+	6.19E-01	610	716	=	5.15E-10	1326	0	+	
F46	3.08E-04	1048	278	+	2.94E-07	1210	116	+	3.96E-07	1204	122	+	5.15E-10	1326	0	+	
F47	6.33E-01	612	714	=	3.72E-05	223	1103	+	8.63E-02	846	480	=	5.15E-10	1326	0	+	
F48	3.52E-09	33	1293	-	3.78E-01	757	569	=	8.88E-01	678	648	=	5.15E-10	1326	0	+	
F49	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
F50	2.33E-02	905	421	+	1.86E-02	412	914	-	2.58E-08	1257	69	+	6.15E-10	1323	3	+	
F51	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	5.15E-10	1326	0	+	
+/-/=		19/4/5				19/1/8				23/0/5				28/0/0			

to transform the constrained optimization problems into unconstrained optimization problems. These engineering design problems are completely described in the following sections.

The pressure vessel design problem shown in Figure 8, presented by [50], is a typical hybrid optimization

problem where the objective is to reduce the total cost including forming cost, material cost and welding cost. There are four different variables: vessel thickness T_s (x_1), head thickness T_h (x_2), inner diameter R (x_3) and vessel cylindrical section length L (x_4). The problem can be described as Eq. (22). The comparison results are shown

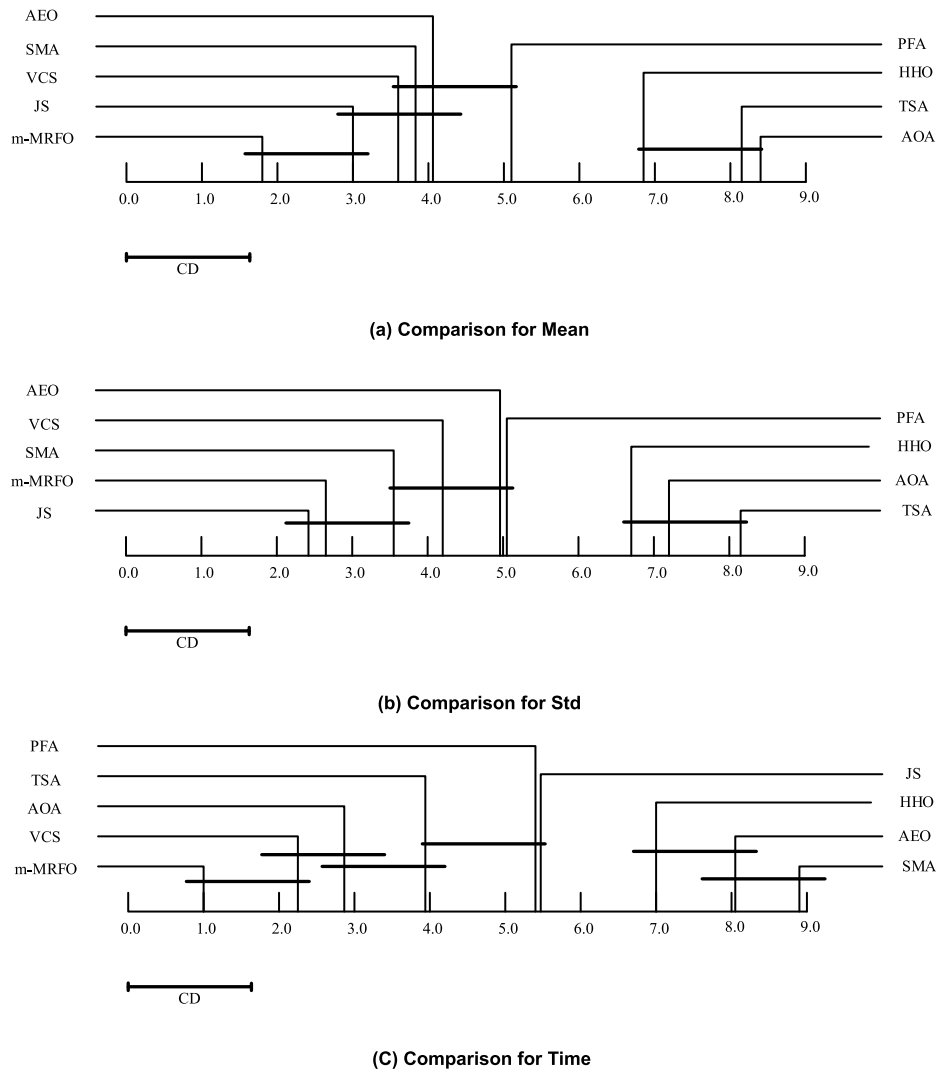


FIGURE 7. Algorithm comparison using a post-hoc test for CEC 2017.

in Table 11 and Table 12.

$$\min f(x_1, x_2, x_3, x_4) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to $g_1(X) = -x_1 + 0.0193x_3 \leq 0$
 $g_2(X) = -x_2 + 0.00954x_3 \leq 0$
 $g_3(X) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^2 + 1,296,000 \leq 0$
 $g_4(X) = x_4 - 240 \leq 0$
 Variable ranges : $1 \times 0.0625 \leq x_1,$
 $x_2 \leq 99 \times 0.0625, 10 \leq x_3, x_4 \leq 200$ (22)

The tension/compression spring design problem is a mechanical engineering design optimization problem [65], which can be used to evaluate the superiority of the algorithm. As shown in Figure 9, the goal of this problem is to reduce the weight of the spring. It includes four nonlinear inequalities and three continuous variables: wire diameter w (x_1), coil average diameter d (x_2), coil length or number L (x_3).

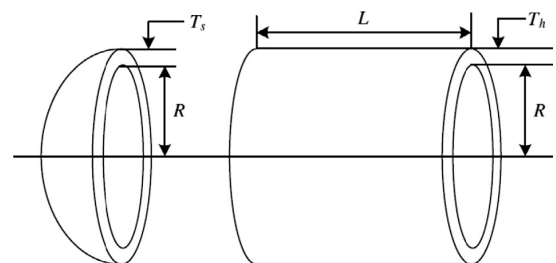


FIGURE 8. Schematic of the pressure vessel design problem.

The comparison results are shown in Table 13 and Table 14. The mathematical model of this problem can be described as Eq. (23).

$$\min f(x_1, x_2, x_3) = (x_3 + 2)x_1^2x_2$$

Subject to $g_1(X) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0$

TABLE 10. Average ranks of m-MRFO and eight algorithms for CEC 2017 test according to the Friedman test ($\alpha = 0.05$).

Test project	AEO	HHO	VCS	AOA	SMA	JS	PFA	TSA	m-MRFO
Mean Ranks ^a	4.18	6.86	3.64	8.39	3.86	3.00	5.11	8.14	1.82
Std Ranks ^b	4.96	6.71	4.21	7.21	3.54	2.46	5.07	8.14	2.68
Time Ranks ^c	8.04	7.00	2.25	2.89	8.96	5.46	5.43	3.96	1.00

- a) 1) P-value computed by the Friedman test: 1.8521e-30. Chi-square is 159.69
 2) P-value computed by the Iman-Davenport test: 7.7715e-16.
 3) F distribution with 6 x 168 degree freedom is 67.05.
- b) 1) P-value computed by the Friedman test: 2.4811e-22. Chi-square is 120.60
 2) P-value computed by the Iman-Davenport test: 9.4705e-12.
 3) F distribution with 6 x 168 degree freedom is 31.49.
- c) 1) P-value computed by the Friedman test: 9.9535e-43. Chi-square is 218.04
 2) P-value computed by the Iman-Davenport test: 0.
 3) F distribution with 6 x 168 degree freedom is 989.06.

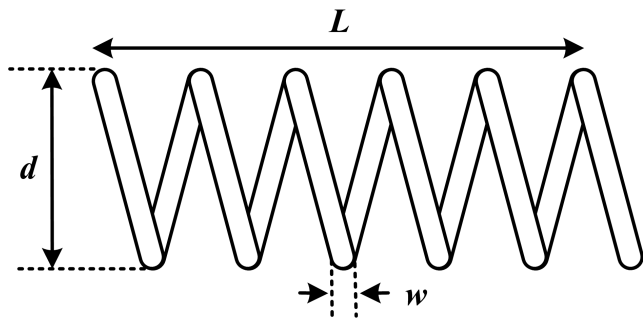


FIGURE 9. Schematic of tension/compression spring design problem.

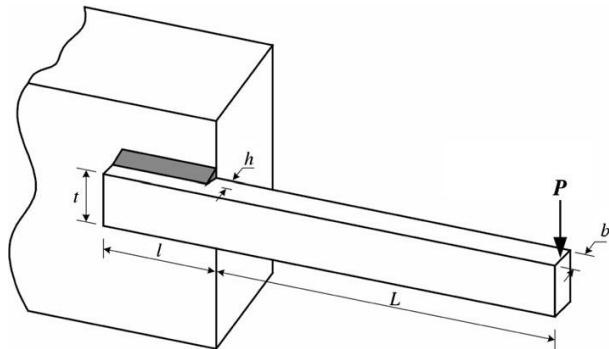


FIGURE 10. Schematic of welded beam design problem.

$$\begin{aligned}
 g_2(X) &= \frac{x_2(4x_2 - x_1)}{12566x_1^3(x_2 - x_1)} + \frac{1}{5108x_1^2} - 1 \leq 0 \\
 g_3(X) &= 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \\
 g_4(X) &= \frac{2(x_1 + x_2)}{3} - 1 \leq 0 \\
 \text{Variable range : } &0.05 \leq x_1 \leq 2, \\
 &0.25 \leq x_2 \leq 1.3, \quad 2.0 \leq x_3 \leq 15.0 \quad (23)
 \end{aligned}$$

As shown in Figure 10, the main purpose of the welded beam design problem is to reduce the manufacturing cost of the welded beam, which mainly involves four variables:

TABLE 11. Comparisons of statistical results using reported optimizers in the literature for pressure vessel design.

Algorithm	Worst	Mean	Best	Std
G-QPSO[51]	5942.3234	5887.3161	5885.3336	10
LCA[52]	6090.6114	6070.5884	6059.8553	11.3753
BA[53]	6059.71	6192.13	6318.95	137.22
MBA[54]	6392.5062	6200.6476	5889.3216	160.34
AFA[55]	6090.5261	6059.7142	6064.3360	11.28
HGA[56]	6080.4228	6073.2548	6059.7167	15.06
PSO-DE[57]	NA	6059.7140	6059.71	NA
CDE[58]	6371.0455	6085.2303	6059.7340	41.0130
AEO[40]	6820.8007	6136.3019	5994.5069	161.2901
TEO[59]	6134.1879	5942.5659	5887.5110	62.2212
IAPSO[60]	6090.5314	6068.7539	6059.1743	14.0057
m-MRFO	5885.8938	5885.5216	5885.3858	0.1291

TABLE 12. Comparisons of best solutions offered by reported optimizers for pressure vessel design.

Algorithm	Optimal values for variables				Optimal cost
	x_1	x_2	x_3	x_4	
DDSCA[61]	0.7782	0.3855	40.3198	176.6389	5888.3366
ISCA[62]	0.8125	0.4375	42.0982	176.6389	6059.7410
MBA[54]	0.7802	0.3856	40.4292	198.4964	5889.3216
VCS[42]	0.7781	0.3846	40.3196	200.0000	5885.3320
TEO[59]	0.7791	0.3852	40.3698	199.3018	5887.5110
hHHO-SCA[63]	0.9459	0.4471	46.8513	125.468	6393.0927
HPSO[64]	0.8125	0.4375	42.0984	176.6366	6059.7143
MVO[39]	0.8215	0.4375	42.0907	176.7386	6060.8066
AFA[55]	0.8125	0.4375	42.0984	176.6366	6059.7143
m-MRFO	0.7782	0.3804	40.3197	199.9999	5885.3858

the width h (x_1) and length l (x_2) of the weld zone, the depth t (x_3) and the thickness b (x_4), and subject to the constraints of bending stress, shear stress, maximum end deflection and load conditions. The comparison results are shown in Table 15 and Table 16. the mathematical model of the problem is described as Eq. (24).

$$\begin{aligned}
 \text{minf}(x_1, x_2, x_3, x_4) &= 1.10471x_1^2x_2 \\
 &\quad + 0.04811x_3x_4(14.0 + x_2) \\
 \text{subject to } g_1(X) &= \tau_d - \tau(X) \geq 0
 \end{aligned}$$

TABLE 13. Comparisons of statistical results using reported optimizers in the literature for tension/compression spring design.

Algorithm	Worst	Mean	Best	Std
GA2[65]	0.0128221	0.0127690	0.0127047	3.9390E-05
GA3[66]	0.0129730	0.0127420	0.0126810	5.9000E-05
CA[67]	0.0151156	0.0135681	0.0127210	8.4215E-04
CPSO[68]	0.0129240	0.0127300	0.0126747	5.1985E-04
HPSO[64]	0.0127191	0.0127072	0.0126652	1.5824E-05
QPSO[51]	0.0181270	0.0138540	0.0126690	1.3410E-03
UPSO[69]	0.0503651	0.0229478	0.0131200	7.2057E-03
CDE[58]	0.0127900	0.0127030	0.0126702	2.7000E-05
SSB[70]	0.0167173	0.0129227	0.0126692	5.9000E-04
m-MRFO	0.0127150	0.0126676	0.0126667	1.1635E-05

TABLE 14. Comparisons of best solutions offered by reported optimizers for tension/compression spring design.

Algorithm	Optimal values for variables			Optimal cost
	x_1	x_2	x_3	
GA3[66]	0.051989	0.363965	10.890522	0.0126810
CPSO[68]	0.051728	0.357644	11.244543	0.0126747
CDE[58]	0.051609	0.354714	11.410831	0.0126702
DDSCA[61]	0.052669	0.380673	10.0153	0.012688
GSA[13]	0.050276	0.323680	13.525410	0.0127022
hHHO-SCA[63]	0.054693	0.433378	7.891402	0.0128229
AEO[40]	0.051897	0.361751	10.879842	0.0126662
MVO[39]	0.05251	0.3762	10.33513	0.012970
m-MRFO	0.051747	0.358090	11.210570	0.012667

TABLE 15. Comparisons of statistical results using reported optimizers in the literature for welded beam design problem.

Algorithm	Worst	Mean	Best	Std
PSO-DE[57]	1.724881	1.724858	1.724853	4.10E-06
GA3[66]	1.993408	1.792654	1.728226	7.4700E-02
WCA[17]	1.744697	1.726427	1.724856	4.29E-03
CPSO[68]	1.782143	1.748831	1.728024	1.29E-04
HPSO[64]	1.8142950	1.7490400	1.7248520	4.0100E-02
IDMEA[71]	1.7248526	1.7248524	1.7248523	6.58E-08
IGPSO[72]	1.724852	1.724852	1.724852	4.76378E-09
MGWO-III[73]	1.931161	1.768040	1.725284	0.0581661
AEO[40]	1.725566	1.725005	1.724852	2.4763E-04
m-MRFO	1.7248621	1.7248529	1.724852	5.4277E-07

TABLE 16. Comparisons of best solutions offered by reported optimizers for welded beam design problem.

Algorithm	Optimal values for variables				Optimal cost
	x_1	x_2	x_3	x_4	
DDSCA[61]	0.20516	3.4759	9.0797	0.20552	1.7305
HGA[56]	0.205712	3.470391	9.039693	0.205716	1.725236
MGWO-III[73]	0.205667	3.471899	9.036679	0.205733	1.724984
IAPSO[60]	0.205729	3.470886	9.036623	0.205729	1.724852
TEO[59]	0.205681	3.472305	9.035133	0.205796	1.725284
hHHO-SCA[63]	0.190086	3.696496	9.386343	0.204157	1.779032
HPSO[64]	0.20573	3.470489	9.036624	0.20573	1.724852
CPSO[68]	0.202369	3.544214	9.048210	0.205723	1.728024
WCA[17]	0.205728	3.470522	9.036620	0.205729	1.724856
m-MRFO	0.205729	3.470488	9.036623	0.205729	1.724852

$$\begin{aligned}
 g_2(X) &= \sigma_d - \sigma(X) \geq 0 \\
 g_3(X) &= x_4 - x_1 \geq 0 \\
 g_4(X) &= P_c(X) - P \geq 0 \\
 g_5(X) &= \delta_d - \delta(X) \geq 0
 \end{aligned} \tag{24}$$

where

$$\begin{aligned}
 \tau(X) &= \sqrt{\frac{(\tau'(X))^2 + (\tau''(X))^2 + x_2 \tau'(X) \tau''(X)}{0.25(x_2^2 + (x_1 + x_3)^2)}} \\
 \sigma(X) &= 50400/x_3^2 x_4 \\
 P_c(X) &= 64746.002(1 - 0.0282346x_3)x_3 x_4^3 \\
 \delta(X) &= 2.1952/x_3^2 x_4 \\
 \tau'(X) &= 6000/(\sqrt{2} x_1 x_2) \\
 \tau''(X) &= \frac{6000(14 + 0.5x_2)\sqrt{0.25(x_2^2 + (x_1 + x_3)^2)}}{2(0.707x_1 x_2(x_2^2/12 + 0.25(x_1 + x_3)^2))}
 \end{aligned}$$

V. CONCLUSION

This paper proposes a new improved MRFO variant with ESP strategy, ACP strategy and DES strategy, namely m-MRFO. First, ESP enhances MRFO exploitation during the cyclone foraging phase by using the three optimal individuals and their random synthetic individuals as reference points. Second, the ACP strategy balances exploitation and exploration capabilities by controlling the key parameters of MRFO. Finally, the DES strategy effectively utilizes the dominant population information to guide the evolutionary direction of the population and improve MRFO performance.

Classical test functions and the CEC2017 test suite are used to verify the effectiveness of the improvement strategies and the superiority of m-MRFO. Simulation results show that the ESP strategy can effectively improve the exploitation capability. ACP achieves the balance of exploitation and exploration. DES improves the convergence speed and convergence accuracy of MRFO. To indicate whether the performance differences between the algorithms are statistically significant, the CEC2017 test results are analyzed using the Wilcoxon sign rank test, Friedman test, and post hoc Iman-Davenport test. The statistical results show that the proposed m-MRFO significantly outperforms the other algorithms. To demonstrate the performance of the proposed m-MRFO on real optimization problems three engineering design problems were employed. The results demonstrate m-MRFO can effectively solve real-world optimization problems.

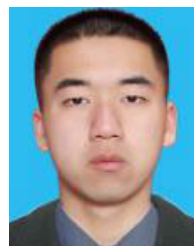
In the subsequent study, we can address the following questions. First, the DES strategy increases the time complexity of MRFO. How to reduce the complexity with guaranteed performance is what we need to study further. The number and composition of individuals stored in the ESP strategy can be further investigated. Moreover, m-MRFO can be extended to solve multi-objective optimization problems. For real-world optimization problems, we plan to use m-MRFO for solving the multi-UAV cooperative path planning problem and the multi-UAV cooperative target allocation problem.

REFERENCES

[1] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: A comprehensive survey," *Artif. Intell. Rev.*, vol. 52, no. 4, pp. 2191–2233, Dec. 2019, doi: 10.1007/s10462-017-9605-z.

- [2] G. Wu, W. Pedrycz, P. N. Suganthan, and R. Mallipeddi, "A variable reduction strategy for evolutionary algorithms handling equality constraints," *Appl. Soft Comput.*, vol. 37, pp. 774–786, Dec. 2015, doi: [10.1016/j.asoc.2015.09.007](https://doi.org/10.1016/j.asoc.2015.09.007).
- [3] S. Droste, T. Jansen, and I. Wegener, "Upper and lower bounds for randomized search heuristics in black-box optimization," *Theory Comput. Syst.*, vol. 39, no. 4, pp. 525–544, Jul. 2006, doi: [10.1007/s00224-004-1177-z](https://doi.org/10.1007/s00224-004-1177-z).
- [4] W. Hare, J. Nutini, and S. Tesfamariam, "A survey of non-gradient optimization methods in structural engineering," *Adv. Eng. Softw.*, vol. 59, pp. 19–28, May 2013, doi: [10.1016/j.advengsoft.2013.03.001](https://doi.org/10.1016/j.advengsoft.2013.03.001).
- [5] H. John, *Holland, Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, USA: Ann Arbor MI Univ. Michigan Press, 1992.
- [6] R. A. Sarker, S. M. Elsayed, and T. Ray, "Differential evolution with dynamic parameters selection for optimization problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 5, pp. 689–707, Oct. 2014, doi: [10.1109/TEVC.2013.2281528](https://doi.org/10.1109/TEVC.2013.2281528).
- [7] D. B. Fogel, "Applying evolutionary programming to selected traveling salesman problems," *Cybern. Syst.*, vol. 24, no. 1, pp. 27–36, Jan. 1993, doi: [10.1080/01969729308961697](https://doi.org/10.1080/01969729308961697).
- [8] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies—A comprehensive introduction," *Natural Comput.*, vol. 1, no. 1, pp. 3–52, 2002, doi: [10.1023/A:1015059928466](https://doi.org/10.1023/A:1015059928466).
- [9] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008, doi: [10.1109/TEVC.2008.919004](https://doi.org/10.1109/TEVC.2008.919004).
- [10] S. A. Uymaz, G. Tezel, and E. Yel, "Artificial algae algorithm (AAA) for nonlinear global optimization," *Appl. Soft Comput.*, vol. 31, pp. 153–171, Jun. 2015, doi: [10.1016/j.asoc.2015.03.003](https://doi.org/10.1016/j.asoc.2015.03.003).
- [11] P. Civicioglu, "Backtracking search optimization algorithm for numerical optimization problems," *Appl. Math. Comput.*, vol. 219, no. 15, pp. 8121–8144, Apr. 2013, doi: [10.1016/j.amc.2013.02.017](https://doi.org/10.1016/j.amc.2013.02.017).
- [12] X. Yang, *Nature-Inspired Metaheuristic Algorithms*. U.K.: Luniver Press, 2010.
- [13] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009, doi: [10.1016/j.ins.2009.03.004](https://doi.org/10.1016/j.ins.2009.03.004).
- [14] Z. Wei, C. Huang, X. Wang, T. Han, and Y. Li, "Nuclear reaction optimization: A novel and powerful physics-based algorithm for global optimization," *IEEE Access*, vol. 7, pp. 66084–66109, 2019, doi: [10.1109/ACCESS.2019.2918406](https://doi.org/10.1109/ACCESS.2019.2918406).
- [15] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016, doi: [10.1016/j.knsys.2015.12.022](https://doi.org/10.1016/j.knsys.2015.12.022).
- [16] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," *Inf. Sci.*, vol. 222, pp. 175–184, Feb. 2013, doi: [10.1016/j.ins.2012.08.023](https://doi.org/10.1016/j.ins.2012.08.023).
- [17] H. Eskandar, A. Sadollah, A. Bahreinnejad, and M. Hamdi, "Water cycle algorithm—A novel Metaheuristic optimization method for solving constrained engineering optimization problems," *Comput. Struct.*, vols. 110–111, pp. 151–166, Nov. 2012, doi: [10.1016/j.compstruc.2012.07.010](https://doi.org/10.1016/j.compstruc.2012.07.010).
- [18] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Perth, WA, Australia, 1995, doi: [10.4018/ijmfmp.2015010104](https://doi.org/10.4018/ijmfmp.2015010104).
- [19] M. Dorigo and G. Di Caro, "Ant colony optimization: A new meta-heuristic," in *Proc. Congr. Evol. Comput. (CEC)*, Jul. 1999, pp. 1470–1477, doi: [10.1109/CEC.1999.782657](https://doi.org/10.1109/CEC.1999.782657).
- [20] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: [10.1016/j.advengsoft.2013.12.007](https://doi.org/10.1016/j.advengsoft.2013.12.007).
- [21] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016, doi: [10.1016/j.advengsoft.2016.01.008](https://doi.org/10.1016/j.advengsoft.2016.01.008).
- [22] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: Sparrow search algorithm," *Syst. Sci. Control Eng.*, vol. 8, no. 1, pp. 22–34, Jan. 2020, doi: [10.1080/21642583.2019.1708830](https://doi.org/10.1080/21642583.2019.1708830).
- [23] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Oct. 2007, doi: [10.1007/s10898-007-9149-x](https://doi.org/10.1007/s10898-007-9149-x).
- [24] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017, doi: [10.1016/j.advengsoft.2017.01.004](https://doi.org/10.1016/j.advengsoft.2017.01.004).
- [25] W. Zhao, L. Wang, and Z. Zhang, "Supply-demand-based optimization: A novel economics-inspired algorithm for global optimization," *IEEE Access*, vol. 7, pp. 73182–73206, 2019, doi: [10.1109/ACCESS.2019.2918753](https://doi.org/10.1109/ACCESS.2019.2918753).
- [26] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017, doi: [10.1016/j.advengsoft.2017.07.002](https://doi.org/10.1016/j.advengsoft.2017.07.002).
- [27] E. Cuevas, M. Cienfuegos, D. Zaldivar, and M. Pérez-Cisneros, "A swarm optimization algorithm inspired in the behavior of the social-spider," *Expert Syst. Appl.*, vol. 40, no. 16, pp. 6374–6384, Nov. 2013, doi: [10.1016/j.eswa.2013.05.041](https://doi.org/10.1016/j.eswa.2013.05.041).
- [28] W. Zhao, Z. Zhang, and L. Wang, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103300, doi: [10.1016/j.engappai.2019.103300](https://doi.org/10.1016/j.engappai.2019.103300).
- [29] M. Abd Elaziz, D. Yousri, M. A. A. Al-qaness, A. M. Abdelaty, A. G. Radwan, and A. A. Ewees, "A Grunwald-Letnikov based manta ray foraging optimizer for global optimization and image segmentation," *Eng. Appl. Artif. Intell.*, vol. 98, Feb. 2021, Art. no. 104105, doi: [10.1016/j.engappai.2020.104105](https://doi.org/10.1016/j.engappai.2020.104105).
- [30] K. K. Ghosh, R. Guha, S. K. Bera, N. Kumar, and R. Sarkar, "S-shaped versus V-shaped transfer functions for binary manta ray foraging optimization in feature selection problem," *Neural Comput. Appl.*, vol. 33, no. 17, pp. 11027–11041, Jan. 2021, doi: [10.1007/s00521-020-05560-9](https://doi.org/10.1007/s00521-020-05560-9).
- [31] H. Xu, H. Song, C. Xu, X. Wu, and N. Yousefi, "Exergy analysis and optimization of a HT-PEMFC using developed manta ray foraging optimization algorithm," *Int. J. Hydrogen Energy*, vol. 45, no. 55, pp. 30932–30941, Nov. 2020, doi: [10.1016/j.ijhydene.2020.08.053](https://doi.org/10.1016/j.ijhydene.2020.08.053).
- [32] M. H. Hassan, E. H. Houssein, M. A. Mahdy, and S. Kamel, "An improved manta ray foraging optimizer for cost-effective emission dispatch problems," *Eng. Appl. Artif. Intell.*, vol. 100, Apr. 2021, Art. no. 104155, doi: [10.1016/j.engappai.2021.104155](https://doi.org/10.1016/j.engappai.2021.104155).
- [33] A. Fathy, H. Rezk, and D. Yousri, "A robust global MPPT to mitigate partial shading of triple-junction solar cell-based system using manta ray foraging optimization algorithm," *Sol. Energy*, vol. 207, pp. 305–316, Sep. 2020, doi: [10.1016/j.solener.2020.06.108](https://doi.org/10.1016/j.solener.2020.06.108).
- [34] Y. Sun, X. Wang, Y. Chen, and Z. Liu, "A modified whale optimization algorithm for large-scale global optimization problems," *Expert Syst. Appl.*, vol. 114, pp. 563–577, Dec. 2018, doi: [10.1016/j.eswa.2018.08.027](https://doi.org/10.1016/j.eswa.2018.08.027).
- [35] F. B. Ozsoydan and A. Baykasoglu, "A swarm intelligence-based algorithm for the set-union knapsack problem," *Future Gener. Comput. Syst.*, vol. 93, pp. 560–569, Apr. 2019, doi: [10.1016/j.future.2018.08.002](https://doi.org/10.1016/j.future.2018.08.002).
- [36] F. B. Ozsoydan and A. Baykasoglu, "Analysing the effects of various switching probability characteristics in flower pollination algorithm for solving unconstrained function minimization problems," *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7805–7819, Nov. 2019, doi: [10.1007/s00521-018-3602-2](https://doi.org/10.1007/s00521-018-3602-2).
- [37] X. S. Yang, "Flower pollination algorithm for global optimization," in *Proc. 11th Int. Conf. Unconventional Comput. Natural Comput. (UCNC)*, Orléans, France, 2012, doi: [10.1007/978-3-642-32894-7_27](https://doi.org/10.1007/978-3-642-32894-7_27).
- [38] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015, doi: [10.1016/j.knsys.2015.07.006](https://doi.org/10.1016/j.knsys.2015.07.006).
- [39] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Feb. 2016, doi: [10.1007/s00521-015-1870-7](https://doi.org/10.1007/s00521-015-1870-7).
- [40] W. Zhao, L. Wang, and Z. Zhang, "Artificial ecosystem-based optimization: A novel nature-inspired meta-heuristic algorithm," *Neural Comput. Appl.*, vol. 32, no. 13, pp. 9383–9425, Jul. 2020, doi: [10.1007/s00521-019-04452-x](https://doi.org/10.1007/s00521-019-04452-x).
- [41] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019, doi: [10.1016/j.future.2019.02.028](https://doi.org/10.1016/j.future.2019.02.028).
- [42] M. D. Li, H. Zhao, X. W. Weng, and T. Han, "A novel nature-inspired algorithm for optimization: Virus colony search," *Adv. Eng. Softw.*, vol. 92, pp. 65–88, Feb. 2016, doi: [10.1016/j.advengsoft.2015.11.004](https://doi.org/10.1016/j.advengsoft.2015.11.004).
- [43] L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Comput. Methods Appl. Mech. Eng.*, vol. 376, Apr. 2021, Art. no. 113609, doi: [10.1016/j.cma.2020.113609](https://doi.org/10.1016/j.cma.2020.113609).
- [44] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020, doi: [10.1016/j.future.2020.03.055](https://doi.org/10.1016/j.future.2020.03.055).

- [45] J.-S. Chou and D.-N. Truong, "A novel Metaheuristic optimizer inspired by behavior of jellyfish in ocean," *Appl. Math. Comput.*, vol. 389, Jan. 2021, Art. no. 125535, doi: [10.1016/j.amc.2020.125535](https://doi.org/10.1016/j.amc.2020.125535).
- [46] H. Yapiçi and N. Cetinkaya, "A new meta-heuristic optimizer: Pathfinder algorithm," *Appl. Soft Comput.*, vol. 78, pp. 545–568, May 2019, doi: [10.1016/j.asoc.2019.03.012](https://doi.org/10.1016/j.asoc.2019.03.012).
- [47] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, "Tunicate swarm algorithm: A new bio-inspired based Metaheuristic paradigm for global optimization," *Eng. Appl. Artif. Intell.*, vol. 90, Apr. 2020, Art. no. 103541, doi: [10.1016/j.engappai.2020.103541](https://doi.org/10.1016/j.engappai.2020.103541).
- [48] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011, doi: [10.1016/j.swevo.2011.02.002](https://doi.org/10.1016/j.swevo.2011.02.002).
- [49] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, pp. 617–644, Dec. 2009, doi: [10.1007/s10732-008-9080-4](https://doi.org/10.1007/s10732-008-9080-4).
- [50] B. K. Kannan and S. N. Kramer, "An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design," *J. Mech. Des.*, vol. 116, no. 2, pp. 405–411, Jun. 1994, doi: [10.1115/1.2919393](https://doi.org/10.1115/1.2919393).
- [51] L. D. S. Coelho, "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems," *Expert Syst. Appl.*, vol. 37, no. 2, pp. 1676–1683, Mar. 2010, doi: [10.1016/j.eswa.2009.06.044](https://doi.org/10.1016/j.eswa.2009.06.044).
- [52] A. H. Kashan, "An efficient algorithm for constrained global optimization and application to mechanical engineering design: League championship algorithm (LCA)," *Comput.-Aided Des.*, vol. 43, no. 12, pp. 1769–1792, Dec. 2011, doi: [10.1016/j.cad.2011.07.003](https://doi.org/10.1016/j.cad.2011.07.003).
- [53] A. H. Gandomi, X.-S. Yang, A. H. Alavi, and S. Talatahari, "Bat algorithm for constrained optimization tasks," *Neural Comput. Appl.*, vol. 22, no. 6, pp. 1239–1255, May 2013, doi: [10.1007/s00521-012-1028-9](https://doi.org/10.1007/s00521-012-1028-9).
- [54] A. Sadollah, A. Bahreinejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2592–2612, May 2013, doi: [10.1016/j.asoc.2012.11.026](https://doi.org/10.1016/j.asoc.2012.11.026).
- [55] A. Baykasoğlu and F. B. Ozsoydan, "Adaptive firefly algorithm with chaos for mechanical design optimization problems," *Appl. Soft Comput.*, vol. 36, pp. 152–164, Nov. 2015, doi: [10.1016/j.asoc.2015.06.056](https://doi.org/10.1016/j.asoc.2015.06.056).
- [56] X. Yan, H. Liu, Z. Zhu, and Q. Wu, "Hybrid genetic algorithm for engineering design problems," *Cluster Comput.*, vol. 20, no. 1, pp. 263–275, Mar. 2017, doi: [10.1007/s10586-016-0680-8](https://doi.org/10.1007/s10586-016-0680-8).
- [57] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Appl. Soft Comput.*, vol. 10, no. 2, pp. 629–640, Mar. 2010, doi: [10.1016/j.asoc.2009.08.031](https://doi.org/10.1016/j.asoc.2009.08.031).
- [58] F.-Z. Huang, L. Wang, and Q. He, "An effective co-evolutionary differential evolution for constrained optimization," *Appl. Math. Comput.*, vol. 186, no. 1, pp. 340–356, Mar. 2007, doi: [10.1016/j.amc.2006.07.105](https://doi.org/10.1016/j.amc.2006.07.105).
- [59] A. Kaveh and A. Dardas, "A novel meta-heuristic optimization algorithm: Thermal exchange optimization," *Adv. Eng. Softw.*, vol. 110, pp. 69–84, Aug. 2017, doi: [10.1016/j.advengsoft.2017.03.014](https://doi.org/10.1016/j.advengsoft.2017.03.014).
- [60] N. B. Guedria, "Improved accelerated PSO algorithm for mechanical engineering optimization problems," *Appl. Soft Comput.*, vol. 40, pp. 455–467, Mar. 2016, doi: [10.1016/j.asoc.2015.10.048](https://doi.org/10.1016/j.asoc.2015.10.048).
- [61] Y. Li, Y. Zhao, and J. Liu, "Dimension by dimension dynamic sine cosine algorithm for global optimization problems," *Appl. Soft Comput.*, vol. 98, Jan. 2021, Art. no. 106933, doi: [10.1016/j.asoc.2020.106933](https://doi.org/10.1016/j.asoc.2020.106933).
- [62] S. Gupta and K. Deep, "Improved sine cosine algorithm with crossover scheme for global optimization," *Knowl.-Based Syst.*, vol. 165, pp. 374–406, Feb. 2019, doi: [10.1016/j.knsys.2018.12.008](https://doi.org/10.1016/j.knsys.2018.12.008).
- [63] V. K. Kamboj, A. Nandi, A. Bhadoria, and S. Sehgal, "An intensify Harris hawks optimizer for numerical and engineering optimization problems," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106018, doi: [10.1016/j.asoc.2019.106018](https://doi.org/10.1016/j.asoc.2019.106018).
- [64] Q. He and L. Wang, "A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization," *Appl. Math. Comput.*, vol. 186, no. 2, pp. 1407–1422, 2007, doi: [10.1016/j.amc.2006.07.134](https://doi.org/10.1016/j.amc.2006.07.134).
- [65] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Comput. Ind.*, vol. 41, no. 2, pp. 113–127, 2000, doi: [10.1016/S0166-3615\(99\)00046-9](https://doi.org/10.1016/S0166-3615(99)00046-9).
- [66] C. A. C. Coello and E. M. Montes, "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection," *Adv. Eng. Informat.*, vol. 16, no. 3, pp. 193–203, 2002, doi: [10.1016/S1474-0346\(02\)00011-3](https://doi.org/10.1016/S1474-0346(02)00011-3).
- [67] C. A. C. Coello and R. L. Becerra, "Efficient evolutionary optimization through the use of a cultural algorithm," *Eng. Optim.*, vol. 36, no. 2, pp. 219–236, Apr. 2004, doi: [10.1080/03052150410001647966](https://doi.org/10.1080/03052150410001647966).
- [68] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Eng. Appl. Artif. Intell.*, vol. 20, no. 1, pp. 89–99, Feb. 2007, doi: [10.1016/j.engappai.2006.03.003](https://doi.org/10.1016/j.engappai.2006.03.003).
- [69] K. E. Parsopoulos and M. N. Vrahatis, "Unified particle swarm optimization for solving constrained engineering optimization problems," in *Proc. 1st Int. Conf. Adv. Natural Comput.*, Changsha, China, 2005, doi: [10.1007/11539902_71](https://doi.org/10.1007/11539902_71).
- [70] T. Ray and K. M. Liew, "Society and civilization: An optimization algorithm based on the simulation of social behavior," *IEEE Trans. Evol. Comput.*, vol. 7, no. 4, pp. 386–396, Aug. 2003, doi: [10.1109/TEVC.2003.814902](https://doi.org/10.1109/TEVC.2003.814902).
- [71] J. Xiao, J.-J. He, P. Chen, and Y.-Y. Niu, "An improved dynamic membrane evolutionary algorithm for constrained engineering design problems," *Natural Comput.*, vol. 15, no. 4, pp. 579–589, Dec. 2016, doi: [10.1007/s11047-016-9569-y](https://doi.org/10.1007/s11047-016-9569-y).
- [72] H.-B. Ouyang, L.-Q. Gao, S. Li, and X.-Y. Kong, "Improved global-best-guided particle swarm optimization with learning operation for global optimization problems," *Appl. Soft Comput.*, vol. 52, pp. 987–1008, Mar. 2017, doi: [10.1016/j.asoc.2016.09.030](https://doi.org/10.1016/j.asoc.2016.09.030).
- [73] V. Kumar and D. Kumar, "An astrophysics-inspired grey wolf algorithm for numerical optimization and its application to engineering design problems," *Adv. Eng. Softw.*, vol. 112, pp. 231–254, Oct. 2017, doi: [10.1016/j.advengsoft.2017.05.008](https://doi.org/10.1016/j.advengsoft.2017.05.008).



ANDI TANG was born in Chongqing, China, in 1996. He is currently pursuing the master's degree in unmanned aircraft combat system and technology with Air Force Engineering University, Xi'an.

His research interests include mission planning and optimization algorithm.



HUAN ZHOU received the Ph.D. degree in weapon science and technology from Air Force Engineering University, Xi'an, China, in 2017. He has engaged in teaching and researching on weapon systems, application engineering, and multi-objective optimization.



TONG HAN received the Ph.D. degree in weapon science and technology from Air Force Engineering University, Xi'an, China, in 2008, where he is currently an Associate Professor. He has engaged in teaching and researching on weapon systems, application engineering, and multi-objective optimization.



LEI XIE was born in Changzhou, Jiangsu, China, in 1997. He received the B.E. degree in arms engineering from Air Force Engineering University, Xi'an, Shaanxi, China, in 2019, where he is currently pursuing the master's degree.

His research interests include air combat, intelligent optimization algorithm, maneuvering decision, situation assessment, and trace line aiming of gun.