Mathematical Biosciences and Engineering

MBE, 20(6): 10090–10134.
DOI: 10.3934/mbe.2023443
Received: 26 November 2022
Revised: 08 March 2023
Accepted: 13 March 2023
Published: 29 March 2023

http://www.aimspress.com/journal/MBE

Research article

Modified reptile search algorithm with multi-hunting coordination strategy for global optimization problems

Di Wu[1], Changsheng Wen[2], Honghua Rao[2], Heming Jia[2,*], Qingxin Liu[3] and Laith Abualigah[4]

[1] School of Education and Music, Sanming University, Sanming 365004, China
[2] School of Information Engineering, Sanming University, Sanming 365004, China
[3] School of Computer Science and Technology, Hainan University, Haikou 570228, China
[4] Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan

* **Correspondence:** Email: jiaheming@fjsmu.edu.cn.

Abstract: The reptile search algorithm (RSA) is a bionic algorithm proposed by Abualigah. et al. in 2020. RSA simulates the whole process of crocodiles encircling and catching prey. Specifically, the encircling stage includes high walking and belly walking, and the hunting stage includes hunting coordination and cooperation. However, in the middle and later stages of the iteration, most search agents will move towards the optimal solution. However, if the optimal solution falls into local optimum, the population will fall into stagnation. Therefore, RSA cannot converge when solving complex problems. To enable RSA to solve more problems, this paper proposes a multi-hunting coordination strategy by combining Lagrange interpolation and teaching-learning-based optimization (TLBO) algorithm's student stage. Multi-hunting cooperation strategy will make multiple search agents coordinate with each other. Compared with the hunting cooperation strategy in the original RSA, the multi-hunting cooperation strategy has been greatly improved RSA's global capability. Moreover, considering RSA's weak ability to jump out of the local optimum in the middle and later stages, this paper adds the Lens pposition-based learning (LOBL) and restart strategy. Based on the above strategy, a modified reptile search algorithm with a multi-hunting coordination strategy (MRSA) is proposed. To verify the above strategies' effectiveness for RSA, 23 benchmark and CEC2020 functions were used to test MRSA's performance. In addition, MRSA's solutions to six engineering problems reflected MRSA's engineering applicability. It can be seen from the experiment that MRSA has better performance in solving test functions and engineering problems.

Keywords: reptile search algorithm; Lagrange interpolation; teaching-learning-based optimization; benchmark function test; lens opposition-based learning; restart strategy

1. Introduction

Optimization is the process of finding the optimal value. The optimization problem is usually transformed into a minimum problem. The optimal global value is the minimum value within the boundary and satisfies the constraint conditions. Traditional exact algorithms require gradient information or derivative information [1–2]. Although the results are more accurate, the computational complexity of traditional exact algorithms will grow exponentially with the increasing problem dimensions. Scholars gradually favor Meta-heuristic algorithms (MAs) [3]. MAs will generate a group of random solutions in the solution space and move in the solution space according to some mathematical formulas. After a fixed number of iterations, an optimal solution is finally output. Although the accuracy of the solution obtained by MAs is insufficient compared with the traditional exact algorithms. When solving high-dimensional complex problems, MAs can obtain a relatively optimal solution. Compared with traditional exact algorithms, MAs can save much computation. Based on the characteristics of MAs, MAs have been widely used in solving modern practical problems. For instance, the combination of modified reptile optimization algorithm and deep learning can effectively complete the intrusion detection of the Internet of things and cloud environment [4]; Reference [5] uses swarm intelligence optimization to build an Internet recommendation engine; Inspired by swqrm intelligence, Forestiero et al. established a new way of Internet information reorganization and discovery [6].

MAs can be roughly divided into four categories: based on swarm intelligence, based on genetic variation, based on physical and chemical principles, and based on human behavior. For example, the particle swarm optimization (PSO) [7] algorithm imitates birds' behavior of population migration; grey wolf optimizer (GWO) algorithm [8] is inspired by the hierarchy in the gray wolf population, and GWO algorithm solves the global optimization problem by simulating the gray wolves' hunting behavior; monarch butterfly optimization (MBO) [9] algorithm simulates the migration behavior of monarch butterfly between two regions; moth search algorithm (MSA) [10] establishes a mathematical model through the phototaxis of moth and Levy flight. Hunger games search (HGS) [11] is designed according to the hunger driven activities and behaviors of animals. Colony predation algorithm (CPA)'s [12] inspiration comes from group hunting. The inspiration of genetic algorithm (GA) [13] and differential evolution (DE) algorithm [14] is derived from genetic, crossover and mutation operations; the gravitational search algorithm (GSA) [15] is inspired by Newton's law of universal gravitation and kinematics; the sine cosine algorithm (SCA) [16] solves the optimization problem through the model of sine function and cosine function; The arithmetic optimization algorithm (AOA) [17] is inspired by addition, subtraction, multiplication and division operators; The inspiration of WeIghted mean of vectors (INFO) comes from the weighted average in mathematics. [18] The principle of social learning optimization (SLO) algorithm [19] is the evolution process of human intelligence and social learning theory; The group teaching optimization algorithm (GTOA) [20] simulates the behavior of group teaching in class through mathematical formulas to solve optimization problems.

However, from the NFL theorem [21], one algorithm cannot solve all problems. To solve more problems, improving existing algorithms is also an important method. Common improvement strategies include opposition-based learning [22], local escaping operator strategy [23], mutation strategy [24], chaotic map [25], hybrid algorithm [26,27], etc. For example: Zhang et al. hybird the sine cosine algorithm and harris hawks optimizer algorithm to improve the convergence speed of the algorithm [28]. Zhao et al. uses piecewise linear mapping to increase the randomness of harris hawks optimizer algorithm's parameters [29]. In addition, the purpose of improving existing MAs is to solve practical problems. The existing MAs cannot solve all engineering problems. Therefore, to solve specific problems, scholars will modify the existing MAs. For example, in order to effectively solve the problem of multiple image threshold segmentation, Emam et al. proposed an improved RSA by combining RSA with RUNge Kutta optimizer (RUN) [30]. Then, they introduced scale factor to avoid the imbalance between exploration and exploitation. Chakraborty et al. improved WOA by changing some coefficients and variables [31]. The improved algorithm can effectively determine the severity of the disease according to the chest X-ray photos of COVID-19. Sayed et al. combined convolutional neural network and bald eagle search optimization algorithm for skin injury classification [32]. Piri et al. improved HHO and proposed a multi-objective version of HHO algorithm by using the K-nearest neighbor (KNN) method as a packaging classifier [33].

The reptile search algorithm (RSA) [34] simulates crocodiles surrounding and hunting prey. The algorithm is divided into four parts: the exploration stage is divided into high walking and belly walking, and the exploitation stage is divided into hunting coordination and cooperation. After finding the prey in the middle and late stages, crocodiles will approach the prey. Although RSA has partial optimization ability, when facing complex problems, if RSA does not find the approximate location of the optimal solution in the early and middle stages, it will be challenging to converge in the middle and late stages. To improve the performance of RSA, some scholars have modified the traditional RSA.

Almotairi et al. proposed a hybrid algorithm by integrating RSA and ROA to balance the algorithm's exploration ability and exploitation ability to solve data clustering problems [35]. Huang and others improved RSA through the Levy flight and mutation crossover strategies, enhancing RSA's overall capability [36]. Although the above improvements modified RSA's optimization ability, it is still easy to fall into local optimization when facing high-dimensional complex problems. This paper improves the Lagrangian interpolation [37] method to improve RSA's ability to solve complex problems. Then we propose a multi-hunting coordination strategy combined with the teaching-learning-based optimization (TLBO) algorithm's student stage [38] and Lagrangian interpolation. The multi-hunting coordination strategy will use the current population's random and optimal position to update the positions. This mode not only significantly enhances the algorithm's exploitation ability but also improves the algorithm's exploration ability. In addition, considering that RSA will fall into the local optimum due to a lack of exploration ability in the later period, this paper adds lens opposition-based learning (LOBL) [39] and restart strategy [40] to improve the algorithm's global performance. This paper proposes a modified reptile search algorithm based on the above improvements (MRSA). If MRSA cannot effectively find the approximate position of the global optimum in the early stage, it can jump out of the local optimum in the later stage.

The existing research on RSA only solves specific engineering problems, but few of them solve complex high-dimensional problems. The MRSA proposed in this paper not only has significant advantages in solving high-dimensional test functions, but also has good effects in classical engineering problems.

The main work of this paper is as follows:

●The Lagrange interpolation method is modified and combined with the TLBO algorithm's student stage, and a multi-hunting coordination strategy is proposed. It is used to improve the hunting coordination phase of RSA.

●Add the LOBL strategy and restart strategy to prevent the population from falling into a stagnant state and enhance the global performance of the algorithm.

●The performance of MRA was tested through 23 benchmark functions in 30/200/500 dimensions and CEC2020 functions, which reflects the advantages of MRSA.

●Let MRSA solve six classical engineering problems.

The rest of the article follows: Section 2 introduces the original RSA. Section 3 introduces the Lagrange interpolation, the TLBO algorithm's student stage, the LOBL strategy, the restart strategy and the specific details of MRSA. Section 4 introduces MRSA's solution to 23 benchmark functions and CEC2020 functions. Section 5 details MRSA's specific performance in solving practical engineering problems. Lastly, section 6 summarizes the full article.

2. The RSA

RSA is a meta-heuristic algorithm inspired by crocodiles' foraging behavior in nature. Although crocodiles appear to move slowly, they can attack quickly. As one of the top predators, crocodiles will hunt in groups. The foraging behavior of crocodiles can be divided into two stages: encircle stage (exploration) and the hunting stage (exploitation). Figure 1 shows the schematic diagram of crocodile hunting.

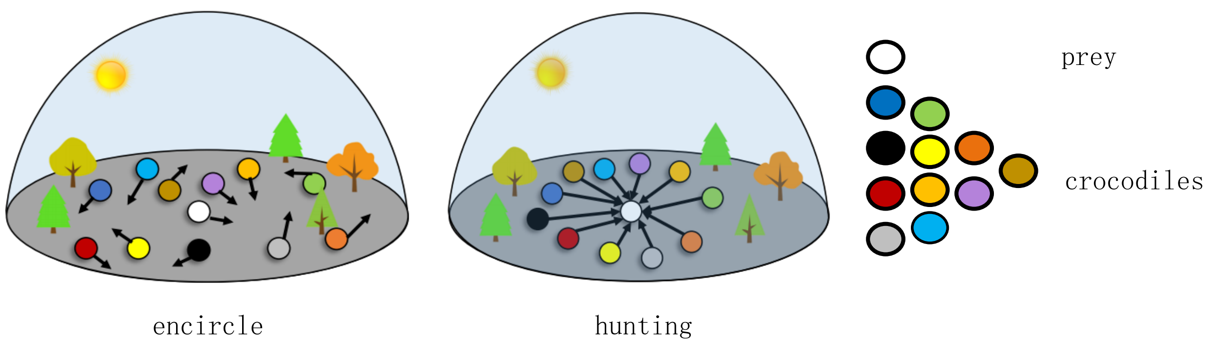


Figure 1. The schematic diagram of crocodile hunting.

2.1. Initialization

RSA will generate N candidate solutions, and the dimension size of each solution is *dim*. The ith solution is ($X_{(i,1)}$, $X_{(i,2)}$, ..., $X_{(i,j)}$, ... $X_{(i,dim)}$). The initialization Formula of the ith solution in the jth dimension is as follows:

$$X_{(i,j)} = LB_{(j)} + rand \times (UB_{(j)} - LB_{(j)}) \quad rand \in [0,1] \tag{1}$$

where $LB_{(j)}$ lower bound and $UB_{(j)}$ is upper bound. *rand* is a random number.

2.2. *Encircle stage*

Crocodiles will choose two different ways in the process of encircling prey: high walking and belly walking. Crocodiles will stretch their legs and float their bodies on the water when looking for prey. Crocodiles crawl around their prey when they find it. At this stage, crocodiles will frequently move throughout the area and will not approach their prey.

2.2.1. High walking

The calculation formula for high walking is shown in formula (2):

$$X_{(i,j)}(t+1) = Best_j(t) \times -\eta_{(i,j)}(t+1) \times \beta - R_{(i,j)}(t+1) \times rand \quad t \le \frac{T}{4} \tag{2}$$

where $X_{(i,j)}(t+1)$ is the ith individual's position in the jth dimension after updating. $Best_j(t)$ is the optimal position so far in the jth dimension. $\eta_{(i,j)}(t+1)$ represents the ith individual's hunting operator in the jth dimension, and its value is calculated by formula (3), β is the control the sensitive parameter to search capability, its value is fixed as 0.005. $R_{(i,j)}(t+1)$ is used to reduce the search area, and its size is calculated by formula (4). t represents the current number of iterations, and T represents the total number.

$$\begin{cases} \eta_{(i,j)}(t+1) = Best_j(t) \times P_{(i,j)}(t+1) \\ P_{(i,j)}(t+1) = \alpha + \dfrac{X_{(i,j)}(t) - M(X_{(i)})}{Best_j(t) \times (UB_{(j)} - LB_{(j)}) + \varepsilon} \\ M(X_{(i)}) = \dfrac{1}{dim} \sum_{j=1}^{dim} X_{(i,j)}(t) \end{cases} \tag{3}$$

where $P_{(i,j)}(t+1)$ is the percentage difference between the optimal individual and the current individual in the jth dimension. α can control the search accuracy, and its value is fixed as 0.1, $X_{(i,j)}(t)$ is the ith individual's position in the jth dimension before updating. $M(X_{(i)})$ is the average level of the ith individual in each dimensional position, and ε is a minimum that prevents the denominator from being zero.

$$R_{(i,j)}(t+1) = \frac{Best_j(t) - X_{(r1,j)}(t)}{Best_j(t) + \varepsilon} \tag{4}$$

where $X_{(r1,j)}(t)$ represents the random individual's position.

2.2.2. Belly walking

Belly walking's calculation formula is shown in formula (5):

$$X_{(i,j)}(t+1) = Best_j(t) \times X_{(r2,j)}(t) \times ES \times rand \quad t > \frac{T}{4} \,\&\&\, t \le \frac{T}{2} \tag{5}$$

where $X_{(r2,j)}(t)$ is the random individual's position. ES controls the evolution direction and randomly takes the decreasing value between 2 and –2. The value of ES is calculated as follows:

$$ES = 2 \times RAND \times (1 - \frac{t}{T}),\ RAND \in [-1,1] \tag{6}$$

2.3. *Hunting stage*

According to crocodiles' hunting behavior, there are two strategies in the hunting stage: hunting coordination and hunting cooperation. Unlike the encircle stage, the crocodiles will keep close to the prey in the hunting stage to complete the predation.

The formula for hunting coordination is as follows:

$$X_{(i,j)}(t+1) = Best_j(t) \times P_{(i,j)}(t+1) \times rand \quad t \le 3\frac{T}{4} \,\&\&\, t > \frac{T}{2} \tag{7}$$

The formula of hunting cooperation is:

$$X_{(i,j)}(t+1) = Best_j(t) - \eta_{(i,j)}(t+1) \times \varepsilon - R_{(i,j)}(t+1) \times rand \quad t > 3\frac{T}{4} \tag{8}$$

RSA is implemented by the above method, and its pseudo-code is shown in Algorithm 1:

Algorithm 1. RSA's pseudo-code

```
Initialization parameters: N, dim, T, α, β
Initialize population(X(1), X(2), ... , X(i), ... , X(N))
While t < T
    Calculate each individual's fitness value of the population
    Find the optimal position so far
    Using Formula (6) to update ES
    For each index by i
        For each dim index by j
            Using Formula (3 and 4) to update parameters η, P, and R.
            If t ≤ T/4 then
                Do high walking by Formula (2)
            Else if t > T/4 and t ≤ T/2 then
                Do belly walking by Formula (5)
            Else if t ≤ 3T/4 and t > T/2 then
                Do hunting coordination by Formula (7)
            Else
                Do hunting cooperation by Formula (8)
            End if
        End for
    End for
t = t + 1
End while
Return the best solution
```

3. Improvement strategies for RSA

The original RSA has a simple structure and excellent results when dealing with simple problems. However, when facing complex problems, RSA quickly falls into local optimum and is challenging to converge. Therefore, this paper improves RSA and the specific improvement strategies are as follows:

3.1. Multi-hunting coordination strategy

RSA will coordinate with the population in the hunting coordination stage, but it is difficult for individuals with poor positions to adjust through the whole population, and this method is challenging. Therefore, RSA is difficult to converge in the later period. In order to overcome this shortcoming, this paper uses the TLBO algorithm to improve the hunting coordination stage of RSA. In the TLBO algorithm's student stage, the current individual and the random individual will coordinate, which can be seen as the coordination between two individuals. However, the coordination between the two individuals has some limitations. It is easy to make the algorithm jump out of the local optimum by coordinating the current individual with multiple individuals. In this section, the TLBO algorithm's student stage is improved by Lagrangian interpolation.

3.1.1. Lagrange interpolation

Lagrangian interpolation can construct a polynomial similar to the objective function through some given positions. The polynomial's optimal solution is taken as the objective function's optimal solution. With the shrinking of the interval, the polynomial's optimal solution will be closer to the objective function's optimal solution. The specific formula of Lagrange interpolation is as follows:

$$P_n(x) = \sum_{i=1}^{n} y_i \Big(\prod_{j \neq i}^{1 \leq j \leq n} \frac{(x - x_j)}{(x_i - x_j)} \Big) \tag{9}$$

where n is the number of selected positions. (x_i, y_i) is the coordinate of the ith position, and j is the index value different from i.

When n = 1, the obtained polynomial is a constant function. When n = 2, the polynomial is a linear function, and its similarity with the objective function is insufficient. When $n \geq 4$, the obtained polynomial is at least a cubic function. Solving its optimal solution will consume huge costs. Therefore, this paper let n = 3, and the obtained polynomial is as follows:

$$P_3(x) = y_0 \times \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \times \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2 \times \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \tag{10}$$

We set: $a_0 = \frac{y_0}{(x_0 - x_1)(x_0 - x_2)}$, $a_1 = \frac{y_1}{(x_1 - x_0)(x_1 - x_2)}$, $a_2 = \frac{y_2}{(x_2 - x_0)(x_2 - x_1)}$. Then, it is easy to get the optimal solution of $P_3(x)$ as follows:

$$x^* = \frac{a_0(x_1 + x_2) + a_1(x_0 + x_2) + a_2(x_0 + x_1)}{2(a_0 + a_1 + a_2)} \tag{11}$$

Considering that the optimization problem is multi-dimensional, we have improved the above

Formula and obtained the following formulas:

$$X_L = \frac{a_0(X_{(r4)}(t)+Best(t))+a_1(X_{(r3)}(t)+Best(t))+a_2(X_{(r3)}(t)+X_{(r4)}(t))}{2(a_0+a_1+a_2)} \tag{12}$$

$$\begin{cases} a_0 = \dfrac{f(X_{(r3)}(t))}{(X_{(r3)}(t)-X_{(r4)}(t))(X_{(r3)}(t)-Best(t))} \\ a_1 = \dfrac{f(X_{(r4)}(t))}{(X_{(r4)}(t)-X_{(r3)}(t))(X_{(r4)}(t)-Best(t))} \\ a_2 = \dfrac{f(Best(t))}{(Best(t)-X_{(r3)}(t))(Best(t)-X_{(r4)}(t))} \end{cases} \tag{13}$$

where X_L is the new position obtained by Lagrangian interpolation. $f()$ is used to calculate the fitness value, $X_{(r3)}(t)$ and $X_{(r4)}(t)$ are two random individuals' positions, and $Best(t)$ is the position of the current optimal individual.

3.1.2. Improve the TLBO algorithm's student stage

A new position is generated by Lagrangian interpolation, and the new position is used in the TLBO algorithm's student stage:

$$X_{TLBO} = \begin{cases} X_{(i)}(t)+rand\times(X_{(i)}(t)-X_L), f(X_L)<f(X_i(t)) \\ X_{(i)}(t)+rand\times(X_L-X_{(i)}(t)), f(X_i(t))<f(X_L) \end{cases} \tag{14}$$

By comparing the fitness values of X_L and X_{TLBO}, choose the better position to change the current position.

3.2. The LOBL

The opposition-based learning (OBL) can generate an opposite position based on the current position. By comparing the current position with the opposite position, choose the better position to update the current position. However, due to the fixed distance between the opposite position and the current position, OBL's randomness is lacking. Therefore, OBL has produced many variants, such as random opposition-based learning [41], quasi-opposition-based learning [42], joint opposite selection [43] and so on. The inspiration for LOBL [39] comes from the lens imaging principle, as shown in Figure 2. The object (x, y) on one side of the lens will generate an inverted reduced real image (x', y') on the other side. The y-axis is considered a lens. The Formula of LOBL can be expressed as:

$$\frac{(LB+UB)/2-x}{x'-(LB+UB)/2} = \frac{y}{y'} \tag{15}$$

Let $h = y / y'$ to get the following formula:

$$x' = \frac{LB+UB}{2} + \frac{LB+UB}{2h} - \frac{x}{h} \tag{16}$$

Considering that the optimization problem is multi-dimensional, Formula (16) can be improved to:

$$X_{(i,j)}(t+1) = \frac{LB_{(j)} + UB_{(j)}}{2} + \frac{LB_{(j)} + UB_{(j)}}{2h} - \frac{X_{(i,j)}(t)}{h} \tag{17}$$

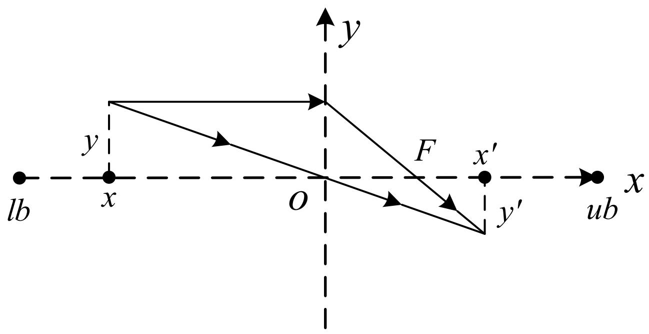


Figure 2. The schematic diagram of LOBL.

3.3. Restart strategy

The restart strategy is an optimization strategy to prevent the population from falling into a stagnant state. Its main idea is: when an individual stays in a poor position for too long, regenerate a position to replace the current individual's position. This paper refers to the idea in reference [40] and records the stagnation state of *i*th individual by $s(i)$. If $s(i)$ is greater than the *limit*, the *i*th individual will generate two new positions through formulas (19)–(21) and replace the original position with the better one. This paper's *limit* is improved, as shown in formula (18). The condition for taking a restart strategy in the later stage will be complex, and it can prevent individuals from leaving the optimal position.

$$limit = \sqrt{t} \tag{18}$$

$$New_1 = (UB - LB) \times rand + LB \tag{19}$$

$$New_2 = (UB + LB) \times rand - X_i \tag{20}$$

$$New_2 = (UB - LB) \times rand + LB \quad \text{if } New_2 > UB \,||\, New_2 < LB \tag{21}$$

3.4. The proposed MRSA

The above three strategies proposed a modified reptile search algorithm with a MRSA. The new algorithm improves the hunting coordination stage of RSA by combining the TLBO algorithm's student stage with Lagrange interpolation and then proposes a multi-hunting coordination strategy. Benefiting from the proposed strategy, the algorithm's comprehensive optimization ability is effectively improved. At the same time, the LOBL and the restart strategy are added to improve RSA's ability to jump out of the local optimum.The pseudo-code of MRSA is shown in Algorithm 2. MRSA's flow chart is in Figure 3.

Algorithm 2. MRSA's pseudo-code

```
Initialization parameters: N, dim, T, α, β
Initialize population((X(1), X(2), ... , X(i), ... , X(N))
While t < T
    Calculate each individual's fitness value of the population
    Find the optimal position so far
    Using Formula (6) to update ES
    Update population through LOBL strategy by Formula (17)
    For each index by i
        For each dim index by j
            Using Formula (3, 4) to update parameters η and R, respectively
            If t ≤ T/4 then
                Do high walking by Formula (2)
            Else if t > T/4 and t ≤ T/2 then
                Do belly walking by Formula (5)
            Else if t ≤ 3T/4 and t > T/2 then
                Use Formula (12) to generate Xnew (Lagrange interpolation)
                Use Formula (14) to generate X_TLBO
                Select the position with a better fitness value
            Else
                Do hunting cooperation by Formula (8)
            End if
        End for
        Update s(i) for each individual
        If s(i) > limit
            Generate New1 and New2 by Formulas (19–21)
            Select the position with a better fitness value
        End if
    End for
t = t + 1
End while
Return the best solution
```

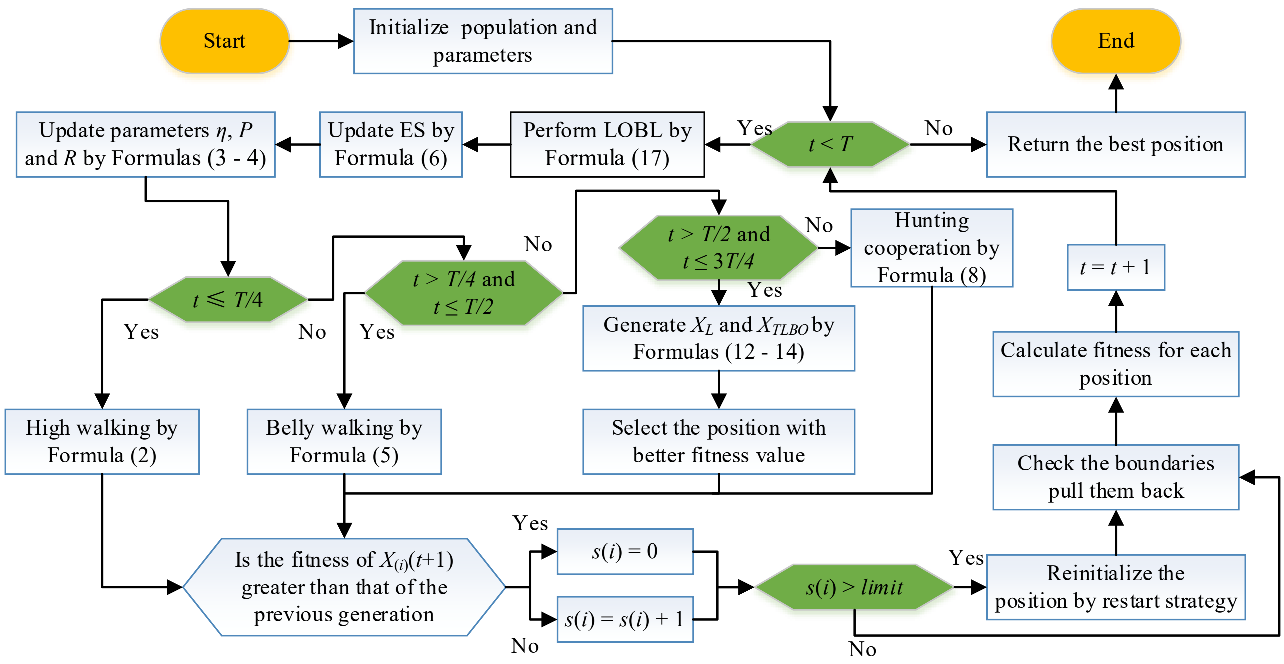


Figure 3. The flow chart of MRSA.

3.5. Complexity analysis

Computational complexity is an essential criterion for evaluating an algorithm. In MRSA, the complexity of initializing the population is $O(N \times D)$, where N is the population size and D is the dimension size. The complexity of updating positions comes from many aspects. The complexity of high walking, bell walking and hunting cooperation is $O(1/4 \times N \times D \times T)$, where T is the number of iterations. The complexity of the multi-hunting cooperation strategy is $O(1/2 \times N \times D \times T)$. The complexity of LOBL is $O(N \times D \times T)$. The complexity of the restart strategy is $O(2 \times N \times D \times T/limit)$. Through the above analysis, the complexity of MRSA is $O(N \times D \times (9/4 \times T + 2T/limit + 1))$. The complexity of traditional RSA is $O(N \times T \times (D + 1))$. Although the complexity of MRSA is improved, the performance of MRSA is better than that of traditional RSA.

4. Experimental test and analysis

All experiments in this paper are completed in MATLAB R2021a on a PC with 2.50 GHz 11th Gen Intel (R) Core (TM) i7-11700 CPU with 16 GB memory and a 64-bit Windows 11 OS.

In this section, we will use 23 benchmark functions and CEC2020 functions to verify the performance of MRSA. At the same time, to show the improvement effect of MRSA clearly, we selected reptile search algorithm (RSA) [34] and six representatives MAs for comparison. They are remora optimization algorithm (ROA) [41], bald eagle search (BES) [45], Sine cosine algorithm (SCA) [16], arithmetic optimization algorithm (AOA) [17], horse herd optimization algorithm (HOA) [46] and sand cat swarm optimization (SCSO) [47]. In addition, we also add LMRAOA (a variant of AOA) [48] as a comparison algorithm.To ensure the fairness of the experiment, we set each algorithm's population size to 30 and the number of iterations to 500. The parameter settings of MRSA and other algorithms are shown in Table 1. In addition, references are provided for comparing the algorithm's parameter settings.

Table 1. All algorithms' parameter settings.

Algorithm	Parameters Setting
MRSA	$\alpha = 0.1$; $\beta = 0.005$
RSA [34]	$\alpha = 0.1$; $\beta = 0.005$
ROA [44]	$C = 0.1$
BES [45]	$\alpha = [1.5, 2.0]$; r = [0, 1]
SCA [16]	$\alpha = 2$
AOA [17]	$MOP\_Max = 1$; $MOP\_Min = 0.2$; $A = 5$; $Mu = 0.499$
HOA [46]	$w = 1$; $phiD = 0.2$; $phi = 0.2$
SCSO [47]	$SM = 2$
LMRAOA [48]	$MOP\_Max = 1$; $MOP\_Min = 0.2$; $A = 5$; $Mu = 0.499$

4.1. Experimental results of 23 benchmark functions

This section will give the results of MRSA and compare the other algorithms on 23 benchmark functions. The 23 benchmark functions are divided into 13 variable and ten fixed-dimension functions. Where F1–F7 are unimodal functions and F8–F23 are multimodal functions. The specific information is shown in Table 2. At the same time, to reflect MRSA's ability to deal with high dimensional problems, we tested F1–F13 in different dimensions, including 30, 200 and 500 dimensions.

The statistics of MRSA and compared algorithm running 30 times in 23 benchmark functions are shown in Table 3. The bold data represents the best result. In the table, Best represents the optimal fitness value, Mean represents the average fitness value and Std represents the standard deviation. In unimodal functions F1–F7, except F6 and F7, MRSA can find the theoretical optimal value. In F6, LMRAOA gain better sotions. In F7, although MRSA did not find the theoretical optimal value, the result is still the best in all dimensions compared with other algorithms. MRSA is still superior to other algorithms in multimodal functions F8–F13 with variable dimensions. In F12 and F13, LMRAOA is superior to MRSA. However, the solution obtained by MRSA is still better than most of the comparison algorithms. In the multi-dimensional functions with fixed dimensions, MRSA only has unsatisfactory results in F18 and F20, but MRSA can still obtain the minimum Best. Only Mean and Std are not minimum.

In addition to statistical data, the convergence curve is a meaningful way to evaluate the performance of an algorithm. Figures 4–7 shows the convergence curve. MRSA has the fastest convergence rate in unimodal functions F1–F5 and F7 with different dimensions. And thanks to the multi-hunting coordination strategy, when solving F5 and F6, MRSA can converge continuously in the middle of an iteration. In F6, only LMRAOA can convergence. In the multimodal function F8–F13 of different dimensions, MRSA can constantly jump out of the local optimum and show excellent global performance. Although MRSA's convergence performance is poor in the early stage when solving some functions in the multimodal functions with fixed dimensions, it can still continue to converge in the later stage.

Table 4 shows the Wilcoxon rank-sum test results of MRSA and other algorithms. $p < 0.05$ indicates that the results obtained by the two algorithms are significantly different. Otherwise, it can be considered that the results are relatively similar. We have roughened the data with $p \geq 0.05$. It is easy to see that, in F1–F4, F9–F11, most algorithms can find the optimal value, so there is no difference between the results of MRSA and the comparison algorithm. In other functions, there is only a small amount of $p \geq 0.05$. Through the comprehensive analysis of Tables 3 and 4, MRSA has a good effect on 23 benchmark functions.

Further analysis sotuions in the 23 benchmark functions, MRSA has excellent performance compared with other MAs and improved MAs. In unimodal functions, MRSA has a faster convergence rate. For example, In F1–F5, although statistical data show that most algorithms can get the optimal solution, but in the convergence curve, MRSA converges faster. In the multimodal function, benefiting from the multi-hunting cooperation strategy, MRSA can continue to converge when it falls into the local optimum in the medium term. In the convergence curves of F12 and F13, MRSA can continue to converge in the medium term and beyond.

Table 2. Details of 23 benchmark functions.

Function	Dim	Boundary	optimal value
$F_1(x)=\sum_{i=1}^{n} x_i^2$		[−100, 100]	
$F_2(x)=\sum_{i=1}^{n} \lvert x_i \rvert + \Pi_{i=1}^{n} \lvert x_i \rvert$		[−10, 10]	
$F_3(x)=\sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$		[−100, 100]	
$F_4(x)=\max\{\lvert x_i \rvert, 1 \leq i \leq n\}$		[−100, 100]	0
$F_5(x)=\sum_{i=1}^{n-1} [100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$		[−30, 30]	
$F_6(x)=\sum_{i=1}^{n} (x_i+5)^2$		[−100, 100]	
$F_7(x)=\sum_{i=1}^{n} i \times x_i^4 + random[0,1)$		[−1.28, 1.28]	
$F_8(x)=\sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$		[−500, 500]	−418.9829 × *dim*
$F_9(x)=\sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)+10]$		[−5.12, 5.12]	
$F_{10}(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e)$	30/100/500	[−32, 32]	
$F_{11}(x)=\frac{1}{400}\sum_{i=1}^{n} x_i^2 - \Pi_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$		[−600, 600]	
$F_{12}(x)=\frac{\pi}{n}\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\} + \sum_{i=1}^{n} u(x_i,10,100,4)$, where $y_i = 1+\frac{x_i+1}{4}$, $u(x_i,a,k,m)=\begin{cases} k(x_i-a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i-a)^m & x_i < -a \end{cases}$		[−50, 50]	0
$F_{13}(x)=0.1(\sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2[1+\sin^2(3\pi x_i+1)] + (x_n-1)^2[1+\sin^2(2\pi x_n)])+\sum_{i=1}^{n} u(x_i,5,100,4)$			

Continued on next page

Function	Dim	Boundary	optimal value
$F_{14}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	2	[−65, 65]	1
$F_{15}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	4		0.00030
$F_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+x_2^4$	2	[−5, 5]	−1.0316
$F_{17}(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$			0.398
$F_{18}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3_2^2)]$ $\times[30+(2x_1-3x_2)^2\times(18-32x_2+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	5	[−2, 2]	3
$F_{19}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2)$	3	[−1, 2]	−3.86
$F_{20}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2)$	6	[0, 1]	−3.32
$F_{21}(x)=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$			−10.1532
$F_{22}(x)=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0, 10]	−10.4028
$F_{23}(x)=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$			−10.5363

Table 3. Statistics of MRSA and other algorithms in 23 benchmark functions.

Function	Dim	Statistics	MRSA	RSA	ROA	BES	SCA	AOA	HOA	SCSO	LMRAOA
F1	30	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.58×10^{-02}	8.05×10^{-155}	7.90×10^{-239}	5.70×10^{-125}	2.93×10^{-91}
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.13×10^{-313}	9.96×10^{-312}	$1.58\times10^{+01}$	6.42×10^{-66}	9.56×10^{-129}	2.79×10^{-111}	3.53×10^{-84}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$2.61\times10^{+01}$	3.52×10^{-65}	5.17×10^{-128}	1.37×10^{-110}	1.93×10^{-83}
	200	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$6.37\times10^{+03}$	1.08×10^{-01}	7.35×10^{-227}	3.97×10^{-111}	1.16×10^{-48}
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.49×10^{-315}	**$0.00\times10^{+00}$**	$4.68\times10^{+04}$	1.32×10^{-01}	7.10×10^{-140}	4.90×10^{-100}	4.46×10^{-44}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$2.41\times10^{+04}$	1.71×10^{-02}	3.89×10^{-139}	2.56×10^{-99}	1.70×10^{-43}
	500	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$4.47\times10^{+04}$	5.60×10^{-01}	3.86×10^{-225}	6.39×10^{-111}	4.51×10^{-39}
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	3.01×10^{-314}	**$0.00\times10^{+00}$**	$2.16\times10^{+05}$	6.39×10^{-01}	1.50×10^{-143}	1.31×10^{-99}	3.31×10^{-34}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$9.75\times10^{+04}$	4.64×10^{-02}	8.19×10^{-143}	5.33×10^{-99}	1.00×10^{-33}
F2	30	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	2.70×10^{-183}	9.15×10^{-229}	8.12×10^{-04}	**$0.00\times10^{+00}$**	8.44×10^{-125}	2.26×10^{-66}	9.75×10^{-229}
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.24×10^{-165}	2.53×10^{-536}	1.33×10^{-02}	**$0.00\times10^{+00}$**	6.00×10^{-76}	6.28×10^{-59}	3.78×10^{-141}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.53×10^{-02}	**$0.00\times10^{+00}$**	1.89×10^{-75}	3.31×10^{-58}	2.07×10^{-140}
	200	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.81×10^{-538}	5.55×10^{-238}	$6.08\times10^{+00}$	5.23×10^{-57}	9.12×10^{-122}	2.69×10^{-59}	8.89×10^{-20}
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	3.00×10^{-161}	2.01×10^{-533}	$3.18\times10^{+01}$	2.02×10^{-20}	1.03×10^{-84}	8.20×10^{-53}	1.31×10^{-53}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.64×10^{-160}	**$0.00\times10^{+00}$**	$1.68\times10^{+01}$	1.09×10^{-19}	5.63×10^{-84}	3.19×10^{-52}	2.34×10^{-53}
	500	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	2.17×10^{-538}	1.10×10^{-235}	$3.43\times10^{+01}$	2.77×10^{-14}	6.11×10^{-120}	1.14×10^{-57}	3.63×10^{-14}
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	7.17×10^{-161}	6.09×10^{-533}	$9.81\times10^{+01}$	1.03×10^{-03}	2.96×10^{-71}	8.15×10^{-50}	2.86×10^{-11}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	2.72×10^{-160}	**$0.00\times10^{+00}$**	$3.20\times10^{+01}$	1.31×10^{-03}	1.62×10^{-70}	3.33×10-49	6.51×10^{-11}
F3	30	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.50×10^{-323}	**$0.00\times10^{+00}$**	$6.88\times10^{+02}$	1.03×10^{-121}	1.95×10^{-25}	3.16×10^{-113}	6.94×10^{-164}
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	2.43×10^{-283}	1.07×10^{-28}	$9.99\times10^{+03}$	6.06×10^{-03}	$8.23\times10^{+01}$	1.12×10^{-97}	1.08×10^{-20}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	5.84×10^{-28}	$7.20\times10^{+03}$	1.04×10^{-02}	$2.60\times10^{+02}$	6.05×10^{-97}	4.22×10^{-20}
	200	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.14×10^{-303}	**$0.00\times10^{+00}$**	$7.65\times10^{+05}$	$1.64\times10^{+00}$	2.84×10^{-26}	6.68×10^{-98}	3.25×10^{-161}

Continued on next page

Function	Dim	Statistics	MRSA	RSA	ROA	BES	SCA	AOA	HOA	SCSO	LMRAOA
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.34×10^{-257}	8.84×10^{-145}	$1.03\times10^{+06}$	$4.13\times10^{+00}$	$6.27\times10^{+03}$	5.74×10^{-90}	5.34×10^{-06}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$4.84\times10^{-14}4$	$1.84\times10^{+05}$	$2.34\times10^{+00}$	$1.42\times10^{+04}$	1.97×10^{-89}	2.61×10^{-05}
	500	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	3.38×10^{-294}	**$0.00\times10^{+00}$**	$4.71\times10^{+06}$	$1.39\times10^{+01}$	1.67×10^{-111}	7.76×10^{-96}	1.57×10^{-160}
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.95×10^{-252}	$1.61\times10^{+03}$	$7.10\times10^{+06}$	$9.97\times10^{+03}$	$8.48\times10^{+04}$	2.18×10^{-84}	1.12×10^{-159}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$6.95\times10^{+03}$	$1.24\times10^{+06}$	$5.45\times10^{+04}$	$1.33\times10^{+05}$	1.13×10^{-83}	7.25×10^{-160}
F4	30	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	2.66×10^{-538}	3.54×10^{-245}	$1.11\times10^{+01}$	3.33×10^{-40}	1.44×10^{-96}	8.70×10^{-56}	8.89×10^{-16}
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	2.64×10^{-158}	1.02×10^{-182}	$3.64\times10^{+01}$	3.08×10^{-02}	5.42×10^{-62}	9.61×10^{-51}	3.00×10^{-12}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.45×10^{-157}	**$0.00\times10^{+00}$**	$1.18\times10^{+01}$	1.82×10^{-02}	2.96×10^{-61}	3.54×10^{-50}	9.03×10^{-12}
	200	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.13×10^{-539}	4.71×10^{-236}	$9.33\times10^{+01}$	1.12×10^{-01}	3.16×10^{-106}	2.90×10^{-51}	1.18×10^{-14}
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.86×10^{-158}	4.45×10^{-164}	$9.64\times10^{+01}$	1.33×10^{-01}	2.70×10^{-63}	1.04×10^{-44}	2.73×10^{-03}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	8.40×10^{-158}	**$0.00\times10^{+00}$**	$1.07\times10^{+00}$	1.48×10^{-02}	1.06×10^{-62}	5.62×10^{-44}	5.41×10^{-03}
	500	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.13×10^{-538}	2.59×10^{-230}	$9.82\times10^{+01}$	1.62×10^{-01}	1.52×10^{-102}	1.05×10^{-50}	4.54×10^{-07}
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	9.84×10^{-158}	1.21×10^{-534}	$9.91\times10^{+01}$	1.85×10^{-01}	7.58×10^{-62}	1.83×10^{-45}	3.34×10^{-03}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	3.93×10^{-157}	**$0.00\times10^{+00}$**	2.27×10^{-01}	2.08×10^{-02}	3.35×10^{-61}	7.23×10^{-45}	5.47×10^{-03}
F5	30	Best	**$0.00\times10^{+00}$**	2.28×10^{-25}	$2.66\times10^{+01}$	$1.44\times10^{+01}$	$8.85\times10^{+02}$	$2.80\times10^{+01}$	$2.89\times10^{+01}$	$2.72\times10^{+01}$	2.17×10^{-09}
		Mean	**6.01×10^{-01}**	$1.55\times10^{+01}$	$2.72\times10^{+01}$	$2.51\times10^{+01}$	$1.27\times10^{+05}$	$2.85\times10^{+01}$	$2.90\times10^{+01}$	$2.82\times10^{+01}$	$1.67\times10^{+01}$
		Std	$1.25\times10^{+00}$	$1.47\times10^{+01}$	6.09×10^{-01}	$9.89\times10^{+00}$	$5.09\times10^{+05}$	3.34×10^{-01}	**5.28×10^{-02}**	8.01×10^{-01}	$1.02\times10^{+01}$
	200	Best	**$0.00\times10^{+00}$**	$1.99\times10^{+02}$	$1.97\times10^{+02}$	$1.10\times10^{+00}$	$3.43\times10^{+08}$	$1.99\times10^{+02}$	$1.99\times10^{+02}$	$1.98\times10^{+02}$	7.18×10^{-02}
		Mean	**$1.26\times10^{+01}$**	$1.99\times10^{+02}$	$1.97\times10^{+02}$	$1.61\times10^{+02}$	$5.84\times10^{+08}$	$1.99\times10^{+02}$	$1.99\times10^{+02}$	$1.98\times10^{+02}$	$1.23\times10^{+02}$
		Std	$2.75\times10^{+01}$	**$0.00\times10^{+00}$**	1.70×10^{-01}	$7.57\times10^{+01}$	$2.05\times10^{+08}$	5.24×10^{-02}	2.74×10^{-02}	4.09×10^{-01}	$9.33\times10^{+01}$
	500	Best	**$0.00\times10^{+00}$**	$4.99\times10^{+02}$	$4.94\times10^{+02}$	$9.67\times10^{+00}$	$1.39\times10^{+09}$	$4.99\times10^{+02}$	$4.99\times10^{+02}$	$4.98\times10^{+02}$	9.41×10^{-02}
		Mean	**$4.46\times10^{+00}$**	$4.99\times10^{+02}$	$4.95\times10^{+02}$	$4.19\times10^{+02}$	$1.90\times10^{+09}$	$4.99\times10^{+02}$	$4.99\times10^{+02}$	$4.98\times10^{+02}$	$3.78\times10^{+02}$
		Std	$1.12\times10^{+01}$	**$0.00\times10^{+00}$**	2.90×10^{-01}	$1.77\times10^{+02}$	$5.47\times10^{+08}$	1.23×10^{-01}	2.38×10^{-02}	1.64×10^{-01}	$2.12\times10^{+02}$
F6	30	Best	**$0.00\times10^{+00}$**	$4.68\times10^{+00}$	1.81×10^{-02}	4.80×10^{-05}	$4.83\times10^{+00}$	$2.42\times10^{+00}$	$6.02\times10^{+00}$	$1.03\times10^{+00}$	**$0.00\times10^{+00}$**
		Mean	1.01×10^{-03}	$7.24\times10^{+00}$	9.90×10^{-02}	$1.07\times10^{+00}$	$1.81\times10^{+01}$	$3.22\times10^{+00}$	$6.72\times10^{+00}$	$1.96\times10^{+00}$	**$0.00\times10^{+00}$**
		Std	5.50×10^{-03}	6.04×10^{-01}	8.75×10^{-02}	$2.57\times10^{+00}$	$3.14\times10^{+01}$	3.37×10^{-01}	3.04×10^{-01}	5.00×10^{-01}	**$0.00\times10^{+00}$**
	200	Best	**$0.00\times10^{+00}$**	$5.00\times10^{+01}$	$2.09\times10^{+00}$	4.89×10^{-03}	$1.87\times10^{+04}$	$4.10\times10^{+01}$	$4.82\times10^{+01}$	$3.22\times10^{+01}$	**$0.00\times10^{+00}$**
		Mean	5.43×10^{-02}	$5.00\times10^{+01}$	$5.18\times10^{+00}$	$1.41\times10^{+01}$	$5.22\times10^{+04}$	$4.20\times10^{+01}$	$4.89\times10^{+01}$	$3.61\times10^{+01}$	**$0.00\times10^{+00}$**
		Std	1.73×10^{-01}	**$0.00\times10^{+00}$**	$2.12\times10^{+00}$	$2.21\times10^{+01}$	$2.61\times10^{+04}$	8.18×10^{-01}	4.88×10^{-01}	$2.36\times10^{+00}$	**$0.00\times10^{+00}$**
	500	Best	**$0.00\times10^{+00}$**	$1.25\times10^{+02}$	$8.06\times10^{+00}$	2.25×10^{-02}	$1.19\times10^{+05}$	$1.15\times10^{+02}$	$1.23\times10^{+02}$	$1.00\times10^{+02}$	**$0.00\times10^{+00}$**
		Mean	3.41×10^{-01}	$1.25\times10^{+02}$	$1.56\times10^{+01}$	$3.12\times10^{+01}$	$2.06\times10^{+05}$	$1.16\times10^{+02}$	$1.24\times10^{+02}$	$1.05\times10^{+02}$	**4.14×10^{-31}**
		Std	5.95×10^{-01}	**$0.00\times10^{+00}$**	$7.91\times10^{+00}$	$5.28\times10^{+01}$	$8.98\times10^{+04}$	$1.26\times10^{+00}$	5.69×10^{-01}	$4.18\times10^{+00}$	1.58×10^{-30}
F7	30	Best	**3.10×10^{-07}**	8.91×10^{-06}	4.40×10^{-06}	5.25×10^{-04}	8.75×10^{-03}	3.39×10^{-06}	1.54×10^{-02}	1.24×10^{-05}	5.17×10^{-06}
		Mean	**5.82×10^{-05}**	1.27×10^{-04}	1.91×10^{-04}	5.13×10^{-03}	1.29×10^{-01}	8.45×10^{-05}	6.74×10^{-02}	1.46×10^{-04}	9.79×10^{-05}
		Std	**4.72×10^{-05}**	1.09×10^{-04}	1.52×10^{-04}	4.13×10^{-03}	1.64×10^{-01}	6.81×10^{-05}	3.94×10^{-02}	1.72×10^{-04}	9.68×10^{-05}
	200	Best	**2.41×10^{-08}**	6.53×10^{-06}	3.45×10^{-06}	2.81×10^{-04}	$7.24\times10^{+02}$	3.51×10^{-06}	2.71×10^{-02}	1.74×10^{-05}	1.76×10^{-05}
		Mean	**6.82×10^{-05}**	1.39×10^{-04}	1.45×10^{-04}	6.29×10^{-03}	$1.53\times10^{+03}$	7.49×10^{-05}	1.54×10^{-01}	2.40×10^{-04}	2.49×10^{-04}
		Std	**6.40×10^{-05}**	1.26×10^{-04}	1.31×10^{-04}	3.51×10^{-03}	$4.22\times10^{+02}$	6.49×10^{-05}	1.08×10^{-01}	3.15×10^{-04}	2.40×10^{-04}
	500	Best	**5.52×10^{-07}**	6.78×10^{-06}	9.84×10^{-06}	7.36×10^{-04}	$7.65\times10^{+03}$	1.39×10^{-05}	3.78×10^{-02}	2.08×10^{-05}	1.61×10^{-06}
		Mean	**5.62×10^{-05}**	1.69×10^{-04}	2.55×10^{-04}	6.82×10^{-03}	$1.53\times10^{+04}$	8.39×10^{-05}	1.73×10^{-01}	1.71×10^{-04}	1.72×10^{-04}
		Std	**4.91×10^{-05}**	1.78×10^{-04}	2.49×10^{-04}	3.24×10^{-03}	$3.80\times10^{+03}$	6.87×10^{-05}	1.33×10^{-01}	2.39×10^{-04}	1.36×10^{-04}
F8	30	Best	**$-1.26\times10^{+04}$**	$-5.65\times10^{+03}$	$-1.26\times10^{+04}$	$-1.25\times10^{+04}$	$-4.57\times10^{+03}$	$-6.22\times10^{+03}$	$-5.04\times10^{+03}$	$-8.70\times10^{+03}$	$-1.08\times10^{+04}$

Continued on next page

Function	Dim	Statistics	MRSA	RSA	ROA	BES	SCA	AOA	HOA	SCSO	LMRAOA
		Mean	**$-1.26\times10^{+04}$**	$-5.27\times10^{+03}$	$-1.23\times10^{+04}$	$-9.73\times10^{+03}$	$-3.80\times10^{+03}$	$-5.18\times10^{+03}$	$-4.06\times10^{+03}$	$-6.62\times10^{+03}$	$-1.01\times10^{+04}$
		Std	**3.36×10^{-12}**	$5.22\times10^{+02}$	$4.58\times10^{+02}$	$1.71\times10^{+03}$	$3.49\times10^{+02}$	$4.40\times10^{+02}$	$5.53\times10^{+02}$	$8.50\times10^{+02}$	$4.21\times10^{+02}$
	200	Best	**$-8.38\times10^{+04}$**	$-3.17\times10^{+04}$	$-8.38\times10^{+04}$	$-8.01\times10^{+04}$	$-1.14\times10^{+04}$	$-1.66\times10^{+04}$	$-3.62\times10^{+04}$	$-3.60\times10^{+04}$	$-4.28\times10^{+04}$
		Mean	**$-8.38\times10^{+04}$**	$-2.80\times10^{+04}$	$-8.27\times10^{+04}$	$-6.11\times10^{+04}$	$-1.01\times10^{+04}$	$-1.46\times10^{+04}$	$-1.26\times10^{+04}$	$-3.22\times10^{+04}$	$-3.73\times10^{+04}$
		Std	**8.55×10^{-12}**	$2.17\times10^{+03}$	$2.09\times10^{+03}$	$1.13\times10^{+04}$	$8.52\times10^{+02}$	$1.01\times10^{+03}$	$6.70\times10^{+03}$	$2.74\times10^{+03}$	$2.90\times10^{+03}$
	500	Best	**$-2.09\times10^{+05}$**	$-7.63\times10^{+04}$	$-2.09\times10^{+05}$	$-2.09\times10^{+05}$	$-1.77\times10^{+04}$	$-2.59\times10^{+04}$	$-1.39\times10^{+05}$	$-6.63\times10^{+04}$	$-6.09\times10^{+04}$
		Mean	**$-2.09\times10^{+05}$**	$-6.45\times10^{+04}$	$-2.07\times10^{+05}$	$-1.59\times10^{+05}$	$-1.54\times10^{+04}$	$-2.25\times10^{+04}$	$-3.50\times10^{+04}$	$-6.05\times10^{+04}$	$-5.06\times10^{+04}$
		Std	**2.96×10^{-11}**	$5.96\times10^{+03}$	$7.32\times10^{+03}$	$2.72\times10^{+04}$	$9.38\times10^{+02}$	$1.56\times10^{+03}$	$2.95\times10^{+04}$	$3.43\times10^{+03}$	$5.63\times10^{+03}$
F9	30	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$1.26\times10^{+00}$	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$1.74\times10^{+01}$	$4.61\times10^{+01}$	**$0.00\times10^{+00}$**	$8.73\times10^{+01}$	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$5.39\times10^{+01}$	$2.72\times10^{+01}$	**$0.00\times10^{+00}$**	$1.06\times10^{+02}$	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
	200	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$1.82\times10^{+02}$	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$4.82\times10^{+02}$	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$2.15\times10^{+02}$	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
	500	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$4.84\times10^{+02}$	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$1.37\times10^{+03}$	5.03×10^{-06}	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$5.31\times10^{+02}$	5.28×10^{-06}	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
F10	30	Best	**8.88×10^{-16}**	**8.88×10^{-16}**	**8.88×10^{-16}**	**8.88×10^{-16}**	5.58×10^{-02}	**8.88×10^{-16}**	**8.88×10^{-16}**	**8.88×10^{-16}**	4.44×10^{-15}
		Mean	**8.88×10^{-16}**	**8.88×10^{-16}**	**8.88×10^{-16}**	**8.88×10^{-16}**	$1.46\times10^{+01}$	**8.88×10^{-16}**	5.51×10^{-15}	**8.88×10^{-16}**	4.44×10^{-15}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$8.68\times10^{+00}$	**$0.00\times10^{+00}$**	1.90×10^{-15}	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
	200	Best	**8.88×10^{-16}**	**8.88×10^{-16}**	**8.88×10^{-16}**	**8.88×10^{-16}**	$7.65\times10^{+00}$	3.02×10^{-03}	4.44×10^{-15}	**8.88×10^{-16}**	4.44×10^{-15}
		Mean	**8.88×10^{-16}**	**8.88×10^{-16}**	**8.88×10^{-16}**	9.52×10^{-02}	$1.88\times10^{+01}$	4.92×10^{-03}	5.98×10^{-15}	**8.88×10^{-16}**	4.44×10^{-15}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	5.21×10^{-01}	$4.28\times10^{+00}$	7.67×10^{-04}	1.79×10^{-15}	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
	500	Best	**8.88×10^{-16}**	**8.88×10^{-16}**	**8.88×10^{-16}**	**8.88×10^{-16}**	$1.08\times10^{+01}$	7.53×10^{-03}	**8.88×10^{-16}**	**8.88×10^{-16}**	4.44×10^{-15}
		Mean	**8.88×10^{-16}**	**8.88×10^{-16}**	**8.88×10^{-16}**	**8.88×10^{-16}**	$2.01\times10^{+01}$	8.07×10^{-03}	6.22×10^{-15}	**8.88×10^{-16}**	7.05×10^{-15}
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$2.45\times10^{+00}$	3.26×10^{-04}	2.03×10^{-15}	**$0.00\times10^{+00}$**	1.60×10^{-15}
F11	30	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	1.95×10^{-03}	1.50×10^{-02}	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	9.26×10^{-01}	1.63×10^{-01}	2.56×10^{-01}	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	4.20×10^{-01}	1.19×10^{-01}	4.00×10^{-01}	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
	200	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$1.93\times10^{+02}$	$1.98\times10^{+03}$	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$4.64\times10^{+02}$	$2.40\times10^{+03}$	3.40×10^{-02}	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$1.78\times10^{+02}$	$3.82\times10^{+02}$	1.86×10^{-01}	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
	500	Best	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$3.38\times10^{+02}$	$6.16\times10^{+03}$	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
		Mean	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$1.82\times10^{+03}$	$9.18\times10^{+03}$	1.54×10^{-02}	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
		Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$7.28\times10^{+02}$	$2.51\times10^{+03}$	8.44×10^{-02}	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
F12	30	Best	3.33×10^{-13}	6.87×10^{-01}	2.11×10^{-03}	1.88×10^{-07}	$1.13\times10^{+00}$	4.32×10^{-01}	8.29×10^{-01}	4.60×10^{-02}	**1.57×10^{-32}**
		Mean	1.10×10^{-07}	$1.58\times10^{+00}$	1.05×10^{-02}	1.82×10^{-01}	$3.29\times10^{+04}$	5.15×10^{-01}	$1.17\times10^{+00}$	9.66×10^{-02}	**1.57×10^{-32}**
		Std	4.40×10^{-07}	2.56×10^{-01}	5.24×10^{-03}	3.95×10^{-01}	$1.63\times10^{+05}$	4.86×10^{-02}	2.02×10^{-01}	3.91×10^{-02}	**5.57×10^{-48}**
	200	Best	**2.36×10^{-33}**	$1.25\times10^{+00}$	1.15×10^{-02}	1.83×10^{-05}	$9.59\times10^{+08}$	9.87×10^{-01}	$1.14\times10^{+00}$	4.57×10^{-01}	**2.36×10^{-33}**
		Mean	6.39×10^{-06}	$1.25\times10^{+00}$	3.36×10^{-02}	2.48×10^{-01}	$1.56\times10^{+09}$	$1.01\times10^{+00}$	$1.18\times10^{+00}$	5.56×10^{-01}	**2.36×10^{-33}**
		Std	1.87×10^{-05}	4.52×10^{-16}	2.00×10^{-02}	4.99×10^{-01}	$5.27\times10^{+08}$	1.62×10^{-02}	4.12×10^{-02}	7.09×10^{-02}	**6.96×10^{-49}**
	500	Best	**9.42×10^{-34}**	$1.21\times10^{+00}$	1.29×10^{-02}	8.11×10^{-06}	$4.66\times10^{+09}$	$1.07\times10^{+00}$	$1.16\times10^{+00}$	6.68×10^{-01}	**9.42×10^{-34}**

Continued on next page

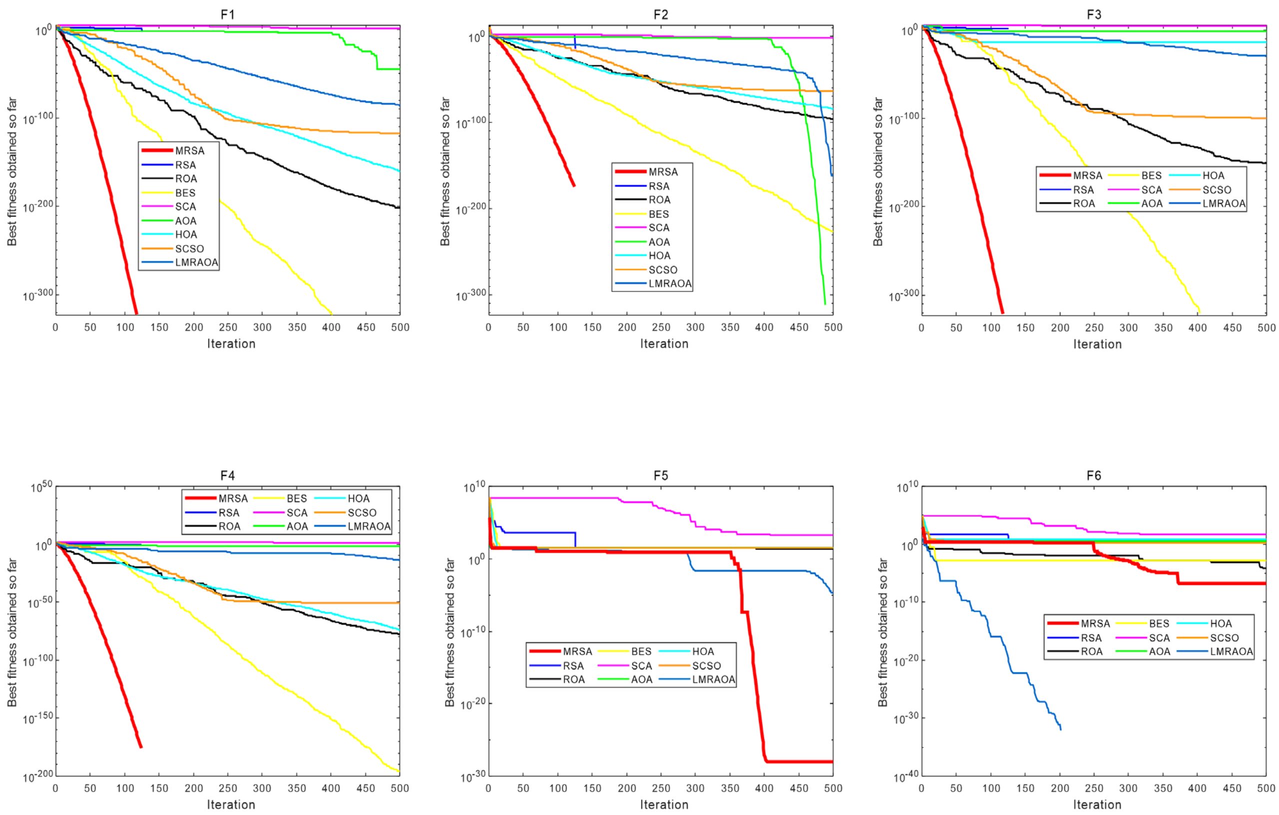
Function	Dim	Statistics	MRSA	RSA	ROA	BES	SCA	AOA	HOA	SCSO	LMRAOA
		Mean	1.62×10^{-05}	$1.21\times10^{+00}$	4.20×10^{-02}	2.03×10^{-01}	$5.90\times10^{+09}$	$1.08\times10^{+00}$	$1.18\times10^{+00}$	7.87×10^{-01}	**3.08×10^{-33}**
		Std	3.01×10^{-05}	4.52×10^{-16}	2.29×10^{-02}	4.56×10^{-01}	$1.42\times10^{+09}$	1.23×10^{-02}	1.70×10^{-02}	5.70×10^{-02}	**6.70×10^{-33}**
F13	30	Best	6.16×10^{-32}	1.89×10^{-30}	6.03×10^{-02}	6.34×10^{-04}	$3.67\times10^{+00}$	$2.61\times10^{+00}$	$2.84\times10^{+00}$	9.25×10^{-01}	**1.35×10^{-32}**
		Mean	4.07×10^{-31}	3.00×10^{-01}	2.23×10^{-01}	$1.23\times10^{+00}$	$7.89\times10^{+04}$	$2.79\times10^{+00}$	$3.08\times10^{+00}$	$2.38\times10^{+00}$	**1.35×10^{-32}**
		Std	2.06×10^{-31}	9.15×10^{-01}	1.23×10^{-01}	$1.46\times10^{+00}$	$2.11\times10^{+05}$	9.94×10^{-02}	2.30×10^{-01}	4.88×10^{-01}	**5.57×10^{-48}**
	200	Best	5.67×10^{-31}	$2.00\times10^{+01}$	$1.28\times10^{+00}$	2.39×10^{-03}	$1.64\times10^{+09}$	$2.00\times10^{+01}$	$2.00\times10^{+01}$	$1.96\times10^{+01}$	**1.35×10^{-32}**
		Mean	1.05×10^{-30}	$2.00\times10^{+01}$	$3.07\times10^{+00}$	$6.75\times10^{+00}$	$2.70\times10^{+09}$	$2.00\times10^{+01}$	$2.00\times10^{+01}$	$1.98\times10^{+01}$	**1.35×10^{-32}**
		Std	1.20×10^{-31}	**$0.00\times10^{+00}$**	$1.58\times10^{+00}$	$9.40\times10^{+00}$	$7.19\times10^{+08}$	2.20×10^{-02}	1.02×10^{-02}	1.04×10^{-01}	5.57×10^{-48}
	500	Best	1.58×10^{-30}	$5.00\times10^{+01}$	$1.85\times10^{+00}$	1.62×10^{-03}	$6.20\times10^{+09}$	$5.02\times10^{+01}$	$5.00\times10^{+01}$	$4.97\times10^{+01}$	**1.35×10^{-32}**
		Mean	2.05×10^{-30}	$5.00\times10^{+01}$	$8.65\times10^{+00}$	$1.58\times10^{+01}$	$9.78\times10^{+09}$	$5.02\times10^{+01}$	$5.00\times10^{+01}$	$4.98\times10^{+01}$	**6.89×10^{-31}**
		Std	9.47×10^{-32}	**$0.00\times10^{+00}$**	$4.38\times10^{+00}$	$2.27\times10^{+01}$	$2.55\times10^{+09}$	4.39×10^{-02}	1.77×10^{-02}	8.24×10^{-02}	1.71×10^{-30}
F14	2	Best	**9.98×10^{-01}**	$1.03\times10^{+00}$	9.98×10^{-01}	9.98×10^{-01}	9.98×10^{-01}	$1.99\times10^{+00}$	9.98×10^{-01}	9.98×10^{-01}	**9.98×10^{-01}**
		Mean	**9.98×10^{-01}**	$4.24\times10^{+00}$	$3.93\times10^{+00}$	$2.98\times10^{+00}$	$1.92\times10^{+00}$	$1.09\times10^{+01}$	$2.88\times10^{+00}$	$3.16\times10^{+00}$	$6.50\times10^{+00}$
		Std	**4.86×10^{-15}**	$3.25\times10^{+00}$	$4.68\times10^{+00}$	$1.60\times10^{+00}$	$1.91\times10^{+00}$	$3.21\times10^{+00}$	$2.34\times10^{+00}$	$3.18\times10^{+00}$	$4.75\times10^{+00}$
F15	4	Best	**3.07×10^{-04}**	9.09×10^{-04}	3.08×10^{-04}	3.27×10^{-04}	5.97×10^{-04}	3.64×10^{-04}	1.02×10^{-03}	3.08×10^{-04}	**3.07×10^{-04}**
		Mean	**4.25×10^{-04}**	3.19×10^{-03}	4.34×10^{-04}	9.48×10^{-03}	1.09×10^{-03}	1.91×10^{-02}	7.93×10^{-03}	4.49×10^{-04}	3.57×10^{-03}
		Std	**1.27×10^{-04}**	1.96×10^{-03}	1.83×10^{-04}	9.80×10^{-03}	3.89×10^{-04}	3.11×10^{-02}	8.13×10^{-03}	3.00×10^{-04}	1.04×10^{-02}
F16	2	Best	**$-1.03\times10^{+00}$**	$-1.03\times10^{+00}$	$-1.03\times10^{+00}$	$-1.03\times10^{+00}$	$-1.03\times10^{+00}$	$-1.03\times10^{+00}$	$-1.03\times10^{+00}$	$-1.03\times10^{+00}$	**$-1.03\times10^{+00}$**
		Mean	**$-1.03\times10^{+00}$**	$-1.03\times10^{+00}$	$-1.03\times10^{+00}$	-9.27×10^{-01}	$-1.03\times10^{+00}$	$-1.03\times10^{+00}$	-9.88×10^{-01}	$-1.03\times10^{+00}$	**$-1.03\times10^{+00}$**
		Std	2.00×10^{-14}	7.38×10^{-04}	4.09×10-08	2.62×10^{-01}	4.21×10^{-05}	1.24×10^{-07}	4.39×10^{-02}	6.46×10-10	**6.12×10^{-16}**
F17	2	Best	**3.98×10^{-01}**	3.98×10^{-01}	3.98×10^{-01}	3.98×10^{-01}	3.98×10^{-01}	3.98×10^{-01}	3.98×10^{-01}	3.98×10^{-01}	**3.98×10^{-01}**
		Mean	**3.98×10^{-01}**	4.24×10^{-01}	3.98×10^{-01}	6.01×10^{-01}	4.00×10^{-01}	3.98×10^{-01}	3.99×10^{-01}	3.98×10^{-01}	**3.98×10^{-01}**
		Std	5.98×10^{-14}	2.87×10^{-02}	9.11×10^{-06}	3.41×10^{-01}	1.56×10^{-03}	5.33×10-08	1.08×10^{-03}	1.66×10-08	**$0.00\times10^{+00}$**
F18	5	Best	**$3.00\times10^{+00}$**	$3.00\times10^{+00}$	$3.00\times10^{+00}$	$3.04\times10^{+00}$	$3.00\times10^{+00}$	$3.00\times10^{+00}$	$3.02\times10^{+00}$	$3.00\times10^{+00}$	**$3.00\times10^{+00}$**
		Mean	$3.90\times10^{+00}$	$1.06\times10^{+01}$	$3.00\times10^{+00}$	$5.93\times10^{+00}$	$3.00\times10^{+00}$	$1.16\times10^{+01}$	$6.81\times10^{+00}$	**$3.00\times10^{+00}$**	$1.02\times10^{+01}$
		Std	$4.93\times10^{+00}$	$1.85\times10^{+01}$	3.93×10^{-04}	$1.03\times10^{+01}$	3.29×10^{-04}	$1.98\times10^{+01}$	$1.55\times10^{+01}$	**1.63×10^{-05}**	$1.21\times10^{+01}$
F19	3	Best	**$-3.86\times10^{+00}$**	$-3.86\times10^{+00}$	$-3.86\times10^{+00}$	$-3.86\times10^{+00}$	$-3.86\times10^{+00}$	$-3.86\times10^{+00}$	$-3.86\times10^{+00}$	$-3.86\times10^{+00}$	**$-3.86\times10^{+00}$**
		Mean	**$-3.86\times10^{+00}$**	$-3.76\times10^{+00}$	$-3.86\times10^{+00}$	$-3.70\times10^{+00}$	$-3.85\times10^{+00}$	$-3.85\times10^{+00}$	$-3.86\times10^{+00}$	$-3.86\times10^{+00}$	**$-3.86\times10^{+00}$**
		Std	8.74×10^{-13}	8.82×10^{-02}	2.40×10^{-03}	2.57×10^{-01}	8.46×10^{-03}	3.82×10^{-03}	6.12×10^{-04}	4.17×10^{-03}	**2.55×10^{-15}**
F20	6	Best	**$-3.32\times10^{+00}$**	$-2.90\times10^{+00}$	$-3.32\times10^{+00}$	$-3.13\times10^{+00}$	$-3.11\times10^{+00}$	$-3.14\times10^{+00}$	$-3.31\times10^{+00}$	$-3.32\times10^{+00}$	**$-3.32\times10^{+00}$**
		Mean	$-3.21\times10^{+00}$	$-2.41\times10^{+00}$	$-3.20\times10^{+00}$	$-2.79\times10^{+00}$	$-2.73\times10^{+00}$	$-3.04\times10^{+00}$	$-3.22\times10^{+00}$	$-3.09\times10^{+00}$	**$-3.29\times10^{+00}$**
		Std	6.40×10^{-02}	5.76×10^{-01}	2.09×10^{-01}	3.84×10^{-01}	5.53×10^{-01}	1.32×10^{-01}	9.35×10^{-02}	4.04×10^{-01}	**5.54×10^{-02}**
F21	4	Best	**$-1.02\times10^{+01}$**	$-5.06\times10^{+00}$	$-1.02\times10^{+01}$	$-1.01\times10^{+01}$	$-5.76\times10^{+00}$	$-7.41\times10^{+00}$	$-1.01\times10^{+01}$	$-1.02\times10^{+01}$	**$-1.02\times10^{+01}$**
		Mean	**$-1.02\times10^{+01}$**	$-5.02\times10^{+00}$	$-1.01\times10^{+01}$	$-6.43\times10^{+00}$	$-2.74\times10^{+00}$	$-3.85\times10^{+00}$	$-9.55\times10^{+00}$	$-5.40\times10^{+00}$	**$-1.02\times10^{+01}$**
		Std	2.74×10^{-11}	1.96×10^{-01}	3.12×10^{-02}	$2.66\times10^{+00}$	$1.86\times10^{+00}$	$1.09\times10^{+00}$	9.82×10^{-01}	$1.29\times10^{+00}$	**5.56×10^{-15}**
F22	4	Best	**$-1.04\times10^{+01}$**	$-5.09\times10^{+00}$	$-1.04\times10^{+01}$	$-1.04\times10^{+01}$	$-7.90\times10^{+00}$	$-1.02\times10^{+01}$	$-1.03\times10^{+01}$	$-1.04\times10^{+01}$	$-1.04\times10^{+01}$
		Mean	**$-1.04\times10^{+01}$**	$-5.09\times10^{+00}$	$-1.04\times10^{+01}$	$-7.10\times10^{+00}$	$-3.46\times10^{+00}$	$-4.41\times10^{+00}$	$-9.26\times10^{+00}$	$-6.08\times10^{+00}$	$-1.04\times10^{+01}$
		Std	**2.14×10^{-11}**	8.50×10^{-07}	1.94×10^{-02}	$2.83\times10^{+00}$	$2.07\times10^{+00}$	$1.93\times10^{+00}$	$1.89\times10^{+00}$	$2.82\times10^{+00}$	4.46×10^{-16}
F23	4	Best	**$-1.05\times10^{+01}$**	$-5.13\times10^{+00}$	$-1.05\times10^{+01}$	$-1.05\times10^{+01}$	$-8.73\times10^{+00}$	$-7.88\times10^{+00}$	$-1.05\times10^{+01}$	$-1.05\times10^{+01}$	**$-1.05\times10^{+01}$**
		Mean	**$-1.05\times10^{+01}$**	$-5.13\times10^{+00}$	$-1.05\times10^{+01}$	$-6.04\times10^{+00}$	$-3.26\times10^{+00}$	$-4.27\times10^{+00}$	$-9.90\times10^{+00}$	$-6.31\times10^{+00}$	**$-1.05\times10^{+01}$**
		Std	3.40×10^{-11}	1.65×10^{-06}	2.32×10^{-02}	$2.91\times10^{+00}$	$1.87\times10^{+00}$	$1.60\times10^{+00}$	5.74×10^{-01}	$2.73\times10^{+00}$	**3.28×10^{-15}**

Table 4. Results of wilcoxon rank-sum test in 23 benchmark functions.

Function	Dim	MRSA VS RSA	MRSA VS ROA	MRSA VS B×10S	MRSA VS SCA	MRSA VS AOA	MRSA VS HOA	MRSA VS SCSO	MRSA VS LMRAOA
F1	30	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
	200	**$1.00\times10^{+00}$**	**2.50×10^{-01}**	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
	500	**$1.00\times10^{+00}$**	**5.00×10^{-01}**	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
F2	30	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
	200	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
	500	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
F3	30	**$1.00\times10^{+00}$**	2.56×10^{-06}	**5.00×10^{-01}**	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
	200	**$1.00\times10^{+00}$**	1.73×10^{-06}	**1.25×10^{-01}**	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
	500	**$1.00\times10^{+00}$**	1.73×10^{-06}	3.13×10^{-02}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
F4	30	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
	200	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
	500	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
F5	30	3.65×10^{-03}	2.13×10^{-06}	9.32×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	2.61×10^{-04}
	200	4.18×10^{-07}	1.73×10^{-06}	3.18×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.89×10^{-04}
	500	1.01×10^{-06}	1.73×10^{-06}	3.52×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.97×10^{-05}
F6	30	1.73×10^{-06}	2.37×10^{-05}	1.13×10^{-05}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	3.79×10^{-06}
	200	1.71×10^{-06}	3.11×10^{-05}	2.41×10^{-04}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	2.56×10^{-06}
	500	1.73×10^{-06}	2.13×10^{-06}	4.90×10^{-04}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	2.56×10^{-06}
F7	30	**6.27×10^{-02}**	6.42×10^{-03}	1.73×10^{-06}	1.73×10^{-06}	**8.61×10^{-01}**	1.73×10^{-06}	2.30×10^{-02}	6.84×10^{-03}
	200	3.11×10^{-05}	1.48×10^{-04}	1.73×10^{-06}	1.73×10^{-06}	**8.22×10^{-02}**	1.73×10^{-06}	**7.52×10^{-02}**	2.77×10^{-03}
	500	1.25×10^{-02}	6.04×10^{-03}	1.73×10^{-06}	1.73×10^{-06}	3.33×10^{-02}	1.73×10^{-06}	**1.16×10^{-01}**	1.74×10^{-04}
F8	30	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
	200	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
	500	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
F9	30	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	**5.00×10^{-01}**	1.73×10^{-06}	**$1.00\times10^{+00}$**	1.95×10^{-03}	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**
	200	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	1.73×10^{-06}	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**
	500	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	1.73×10^{-06}	4.38×10^{-04}	**2.50×10^{-01}**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**
F10	30	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	1.73×10^{-06}	**$1.00\times10^{+00}$**	9.03×10^{-07}	**$1.00\times10^{+00}$**	4.32×10^{-08}
	200	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	8.12×10^{-07}	**$1.00\times10^{+00}$**	4.32×10^{-08}
	500	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	8.21×10^{-07}	**$1.00\times10^{+00}$**	6.25×10^{-07}
F11	30	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	**6.25×10^{-02}**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**
	200	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	**5.00×10^{-01}**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**
	500	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	1.73×10^{-06}	1.73×10^{-06}	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**	**$1.00\times10^{+00}$**
F12	30	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
	200	1.73×10^{-06}	1.73×10^{-06}	1.92×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	5.61×10^{-06}
	500	1.73×10^{-06}	1.73×10^{-06}	4.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
F13	30	1.70×10^{-06}	1.73×10^{-06}	1.70×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.72×10^{-06}
	200	4.32×10^{-08}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
	500	4.32×10^{-08}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	3.59×10^{-04}

Continued on next page

Function	Dim	MRSA VS RSA	MRSA VS ROA	MRSA VS B×10S	MRSA VS SCA	MRSA VS AOA	MRSA VS HOA	MRSA VS SCSO	MRSA VS LMRAOA
F14	2	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.31×10^{-05}
F15	4	1.73×10^{-06}	$\mathbf{1.59\times10^{-01}}$	2.88×10^{-06}	1.73×10^{-06}	1.92×10^{-06}	1.73×10^{-06}	1.17×10^{-02}	6.42×10^{-03}
F16	2	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.58×10^{-06}
F17	2	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.72×10^{-06}
F18	5	2.84×10^{-05}	3.11×10^{-05}	3.11×10^{-05}	3.11×10^{-05}	2.37×10^{-05}	2.60×10^{-05}	3.11×10^{-05}	2.03×10^{-02}
F19	3	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
F20	6	1.73×10^{-06}	$\mathbf{5.19\times10^{-02}}$	2.60×10^{-06}	1.73×10^{-06}	1.24×10^{-05}	1.40×10^{-02}	$\mathbf{8.29\times10^{-01}}$	1.73×10^{-06}
F21	4	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
F22	4	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
F23	4	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}



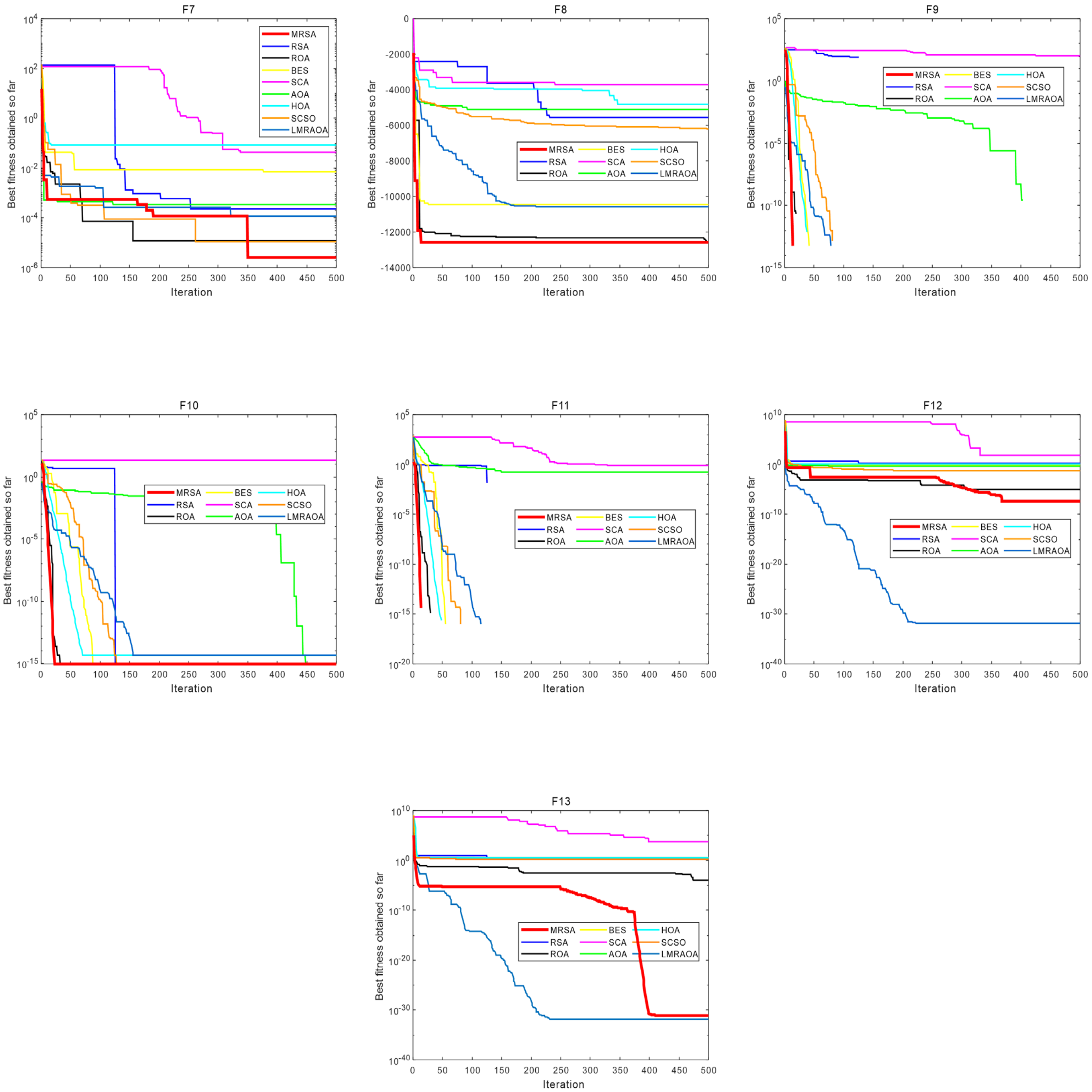


Figure 4. MRSA and other algorithms' convergence performance of in functions F1–F13 (dim = 30).

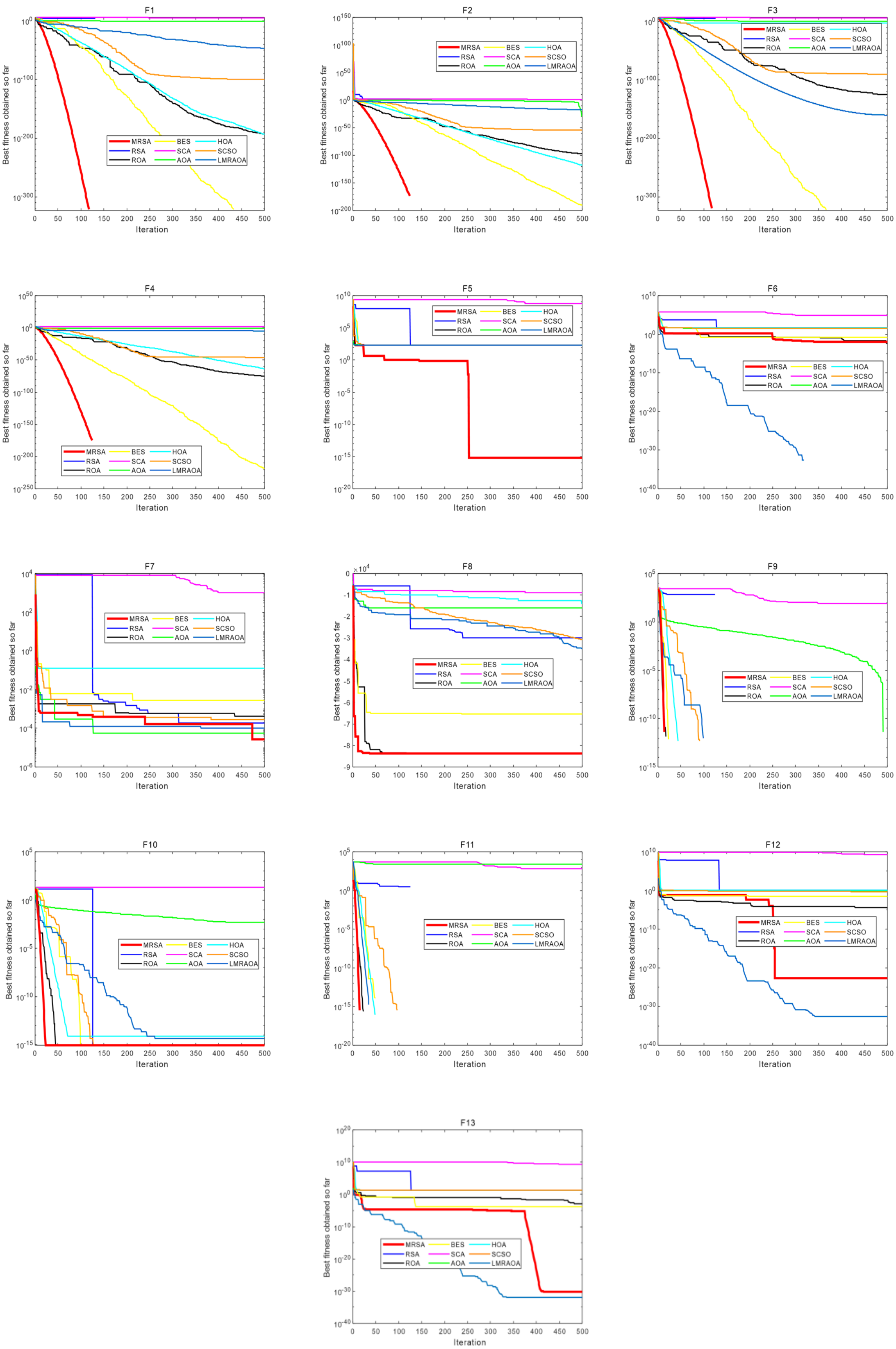


Figure 5. MRSA and other algorithms' convergence performance in functions F1–F13 (dim = 200).

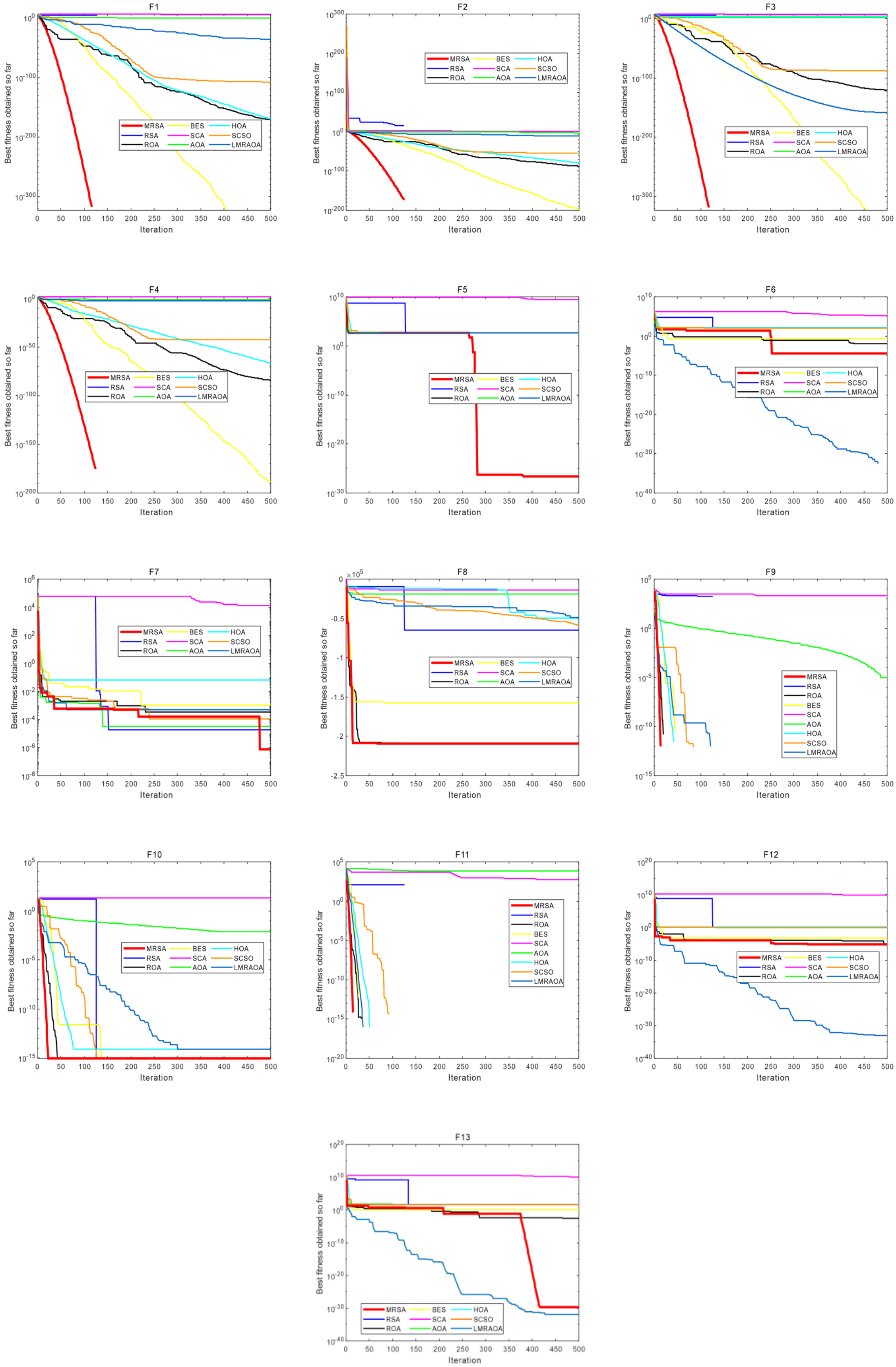


Figure 6. MRSA and other algorithms' convergence performance in functions F1–F13 (dim = 500).

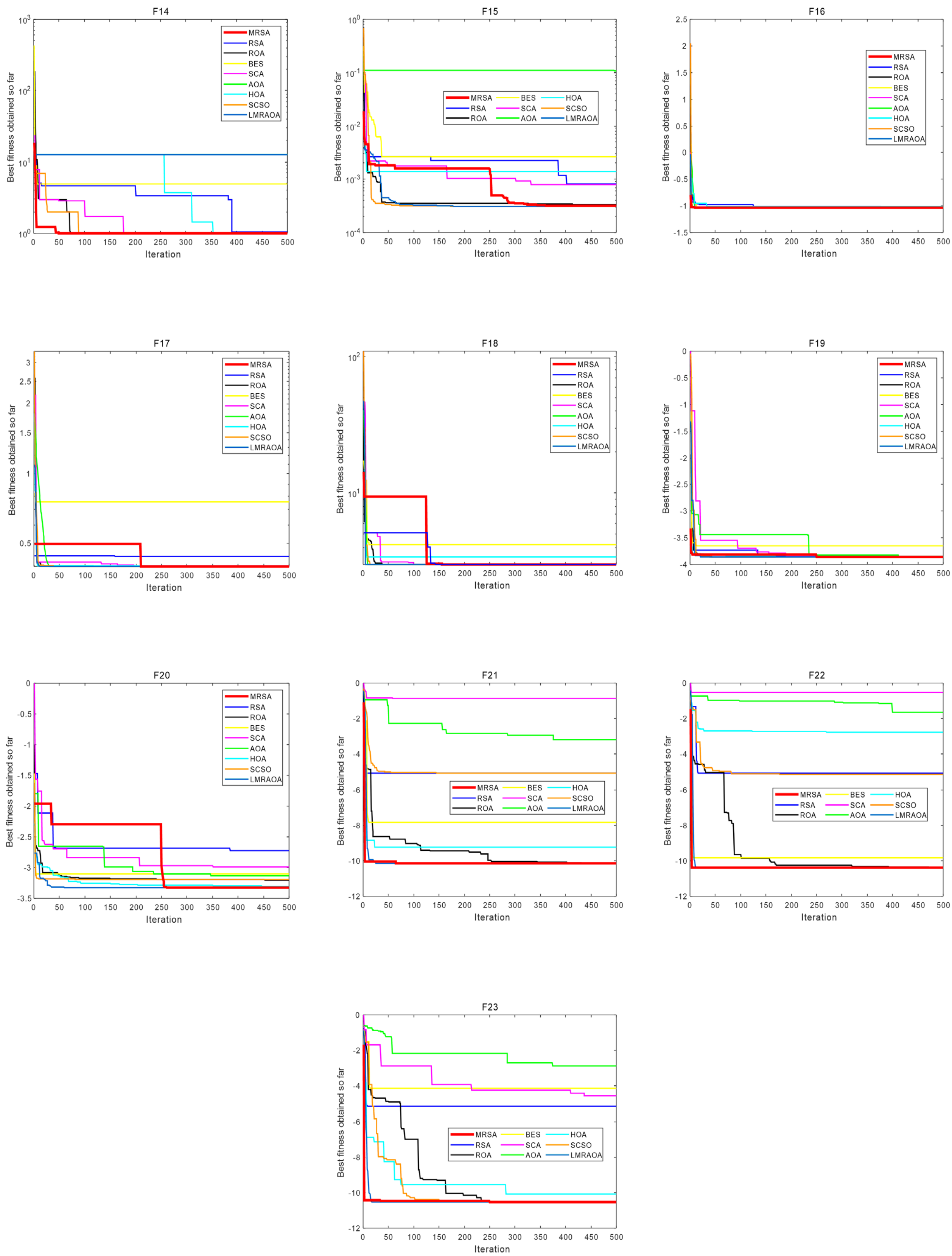


Figure 7. MRSA and other algorithms' convergence performance in functions F14–F23.

4.2. Experimental results on CEC2020 functions

Table 5. Statistics of MRSA and other algorithms in CEC2020 functions.

Function	Statistics	MRSA	RSA	ROA	BES	SCA	AOA	HOA	SCSO
	Best	**$1.00\times10^{+02}$**	$5.99\times10^{+09}$	$1.82\times10^{+07}$	$9.82\times10^{+08}$	$3.81\times10^{+08}$	$3.75\times10^{+09}$	$1.41\times10^{+08}$	$6.40\times10^{+03}$
CEC1	Mean	**$2.23\times10^{+03}$**	$1.14\times10^{+10}$	$1.38\times10^{+09}$	$4.65\times10^{+09}$	$1.03\times10^{+09}$	$9.57\times10^{+09}$	$3.45\times10^{+08}$	$7.53\times10^{+07}$
	Std	**$2.08\times10^{+03}$**	$3.90\times10^{+09}$	$1.67\times10^{+09}$	$3.98\times10^{+09}$	$3.72\times10^{+08}$	$3.75\times10^{+09}$	$1.63\times10^{+08}$	$1.92\times10^{+08}$
	Best	**$1.34\times10^{+03}$**	$2.56\times10^{+03}$	$1.75\times10^{+03}$	$2.30\times10^{+03}$	$2.34\times10^{+03}$	$1.90\times10^{+03}$	$2.40\times10^{+03}$	$1.49\times10^{+03}$
CEC2	Mean	**$1.91\times10^{+03}$**	$2.87\times10^{+03}$	$2.20\times10^{+03}$	$2.64\times10^{+03}$	$2.60\times10^{+03}$	$2.30\times10^{+03}$	$2.84\times10^{+03}$	$2.06\times10^{+03}$
	Std	**$1.45\times10^{+02}$**	$1.80\times10^{+02}$	$3.10\times10^{+02}$	$2.76\times10^{+02}$	$2.51\times10^{+02}$	$2.66\times10^{+02}$	$2.37\times10^{+02}$	$3.27\times10^{+02}$
	Best	**$7.17\times10^{+02}$**	$8.01\times10^{+02}$	$7.70\times10^{+02}$	$7.76\times10^{+02}$	$7.73\times10^{+02}$	$7.88\times10^{+02}$	$7.68\times10^{+02}$	$7.43\times10^{+02}$
CEC3	Mean	**$7.70\times10^{+02}$**	$8.15\times10^{+02}$	$7.96\times10^{+02}$	$8.13\times10^{+02}$	$7.87\times10^{+02}$	$8.05\times10^{+02}$	$7.83\times10^{+02}$	$7.76\times10^{+02}$
	Std	$2.03\times10^{+01}$	**$1.30\times10^{+01}$**	$2.30\times10^{+01}$	$2.48\times10^{+01}$	$1.52\times10^{+01}$	$1.91\times10^{+01}$	$1.55\times10^{+01}$	$2.56\times10^{+01}$
	Best	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**
CEC4	Mean	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	$1.90\times10^{+03}$	$1.90\times10^{+03}$	**$1.90\times10^{+03}$**	$1.90\times10^{+03}$	**$1.90\times10^{+03}$**
	Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	3.98×10^{-01}	$1.07\times10^{+00}$	**$0.00\times10^{+00}$**	$2.21\times10^{+00}$	**$0.00\times10^{+00}$**
	Best	**$1.80\times10^{+03}$**	$3.76\times10^{+05}$	$4.34\times10^{+03}$	$5.97\times10^{+04}$	$1.70\times10^{+04}$	$2.24\times10^{+05}$	$1.86\times10^{+04}$	$3.67\times10^{+03}$
CEC5	Mean	$1.17\times10^{+05}$	$5.24\times10^{+05}$	$1.70\times10^{+05}$	$3.01\times10^{+06}$	$1.13\times10^{+05}$	$5.39\times10^{+05}$	$4.91\times10^{+05}$	**$8.62\times10^{+04}$**
	Std	**$7.97\times10^{+04}$**	$1.55\times10^{+05}$	$2.45\times10^{+05}$	$9.72\times10^{+06}$	$2.01\times10^{+05}$	$4.93\times10^{+05}$	$2.82\times10^{+05}$	$2.09\times10^{+05}$
	Best	**$1.60\times10^{+03}$**	$2.06\times10^{+03}$	$1.75\times10^{+03}$	$1.80\times10^{+03}$	$1.77\times10^{+03}$	$1.90\times10^{+03}$	$1.92\times10^{+03}$	$1.72\times10^{+03}$
CEC6	Mean	**$1.83\times10^{+03}$**	$2.34\times10^{+03}$	$1.88\times10^{+03}$	$2.00\times10^{+03}$	$1.86\times10^{+03}$	$2.19\times10^{+03}$	$2.18\times10^{+03}$	$1.84\times10^{+03}$
	Std	$1.23\times10^{+02}$	$2.58\times10^{+02}$	$1.39\times10^{+02}$	$1.36\times10^{+02}$	**$1.02\times10^{+02}$**	$2.48\times10^{+02}$	$1.46\times10^{+02}$	$1.27\times10^{+02}$
	Best	**$2.12\times10^{+03}$**	$2.99\times10^{+04}$	$3.35\times10^{+03}$	$4.81\times10^{+03}$	$7.27\times10^{+03}$	$5.57\times10^{+03}$	$7.27\times10^{+03}$	$3.10\times10^{+03}$
CEC7	Mean	**$7.88\times10^{+03}$**	$1.92\times10^{+06}$	$1.62\times10^{+04}$	$3.70\times10^{+05}$	$2.12\times10^{+04}$	$1.54\times10^{+06}$	$1.87\times10^{+04}$	$1.45\times10^{+04}$
	Std	**$8.17\times10^{+03}$**	$3.27\times10^{+06}$	$3.45\times10^{+04}$	$7.20\times10^{+05}$	$1.99\times10^{+04}$	$2.72\times10^{+06}$	$1.95\times10^{+04}$	$3.59\times10^{+04}$
	Best	**$2.21\times10^{+03}$**	$2.87\times10^{+03}$	$2.32\times10^{+03}$	$2.49\times10^{+03}$	$2.36\times10^{+03}$	$2.74\times10^{+03}$	$2.30\times10^{+03}$	$2.30\times10^{+03}$
CEC8	Mean	**$2.31\times10^{+03}$**	$3.33\times10^{+03}$	$2.47\times10^{+03}$	$2.82\times10^{+03}$	$2.47\times10^{+03}$	$3.18\times10^{+03}$	$2.38\times10^{+03}$	$2.36\times10^{+03}$
	Std	**$1.24\times10^{+01}$**	$4.28\times10^{+02}$	$2.18\times10^{+02}$	$4.38\times10^{+02}$	$3.16\times10^{+02}$	$3.67\times10^{+02}$	$1.53\times10^{+02}$	$1.80\times10^{+02}$
	Best	**$2.42\times10^{+03}$**	$2.83\times10^{+03}$	$2.75\times10^{+03}$	$2.77\times10^{+03}$	$2.78\times10^{+03}$	$2.78\times10^{+03}$	$2.60\times10^{+03}$	$2.74\times10^{+03}$
CEC9	Mean	**$2.66\times10^{+03}$**	$2.90\times10^{+03}$	$2.76\times10^{+03}$	$2.79\times10^{+03}$	$2.80\times10^{+03}$	$2.89\times10^{+03}$	$2.79\times10^{+03}$	$2.77\times10^{+03}$
	Std	$1.46\times10^{+02}$	$8.32\times10^{+01}$	$7.55\times10^{+01}$	$5.52\times10^{+01}$	**$9.77\times10^{+00}$**	$9.35\times10^{+01}$	$1.14\times10^{+02}$	$5.43\times10^{+01}$
	Best	**$2.60\times10^{+03}$**	$3.25\times10^{+03}$	$2.95\times10^{+03}$	$3.00\times10^{+03}$	$2.96\times10^{+03}$	$3.17\times10^{+03}$	$2.94\times10^{+03}$	$2.92\times10^{+03}$
CEC10	Mean	**$2.93\times10^{+03}$**	$3.47\times10^{+03}$	$3.07\times10^{+03}$	$3.25\times10^{+03}$	$2.98\times10^{+03}$	$3.48\times10^{+03}$	$2.97\times10^{+03}$	$2.96\times10^{+03}$
	Std	**$2.25\times10^{+01}$**	$2.29\times10^{+02}$	$1.59\times10^{+02}$	$2.66\times10^{+02}$	$3.88\times10^{+01}$	$2.91\times10^{+02}$	$3.23\times10^{+01}$	$4.32\times10^{+01}$

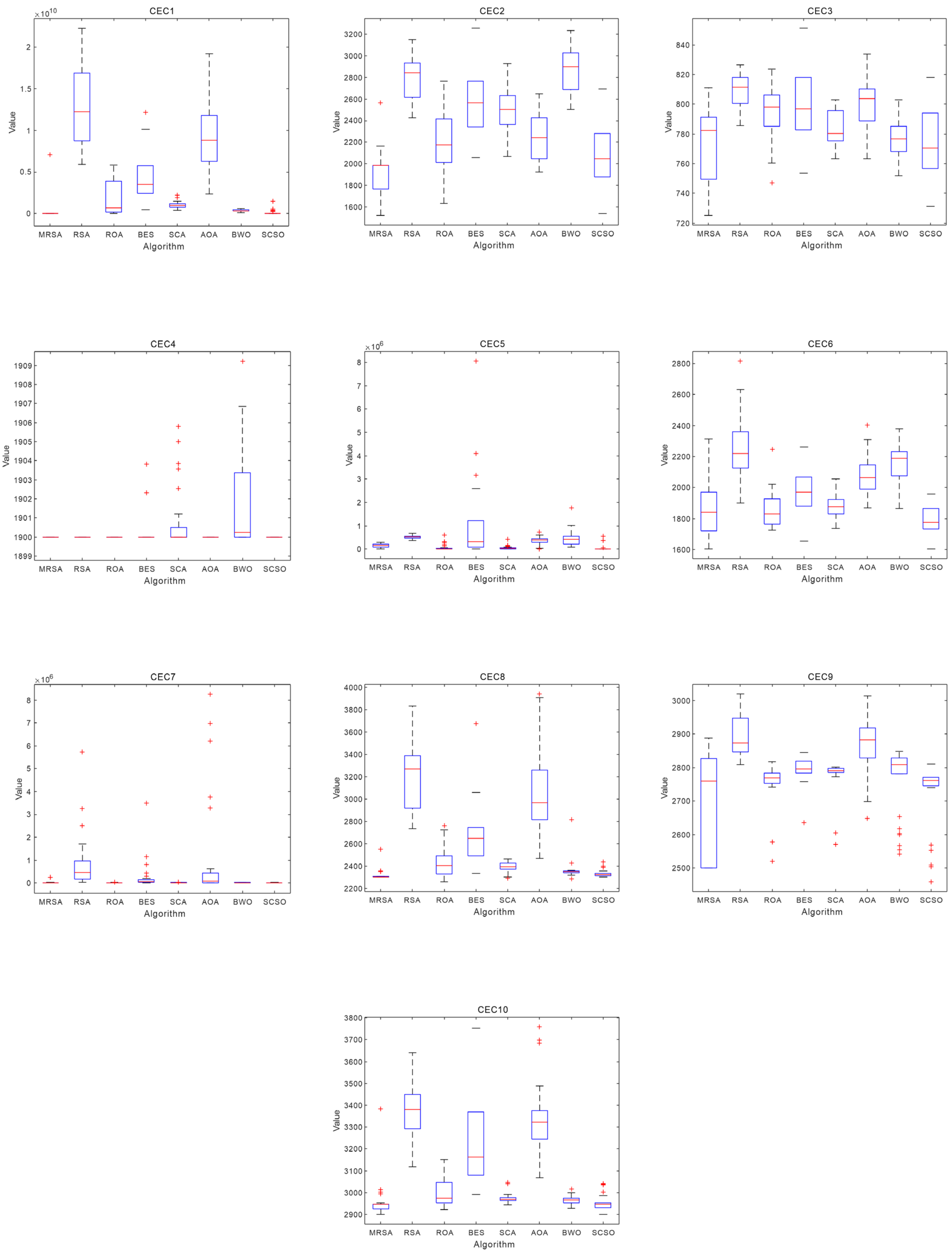


Figure 8. Box chart of statistical data in CEC2020 functions.

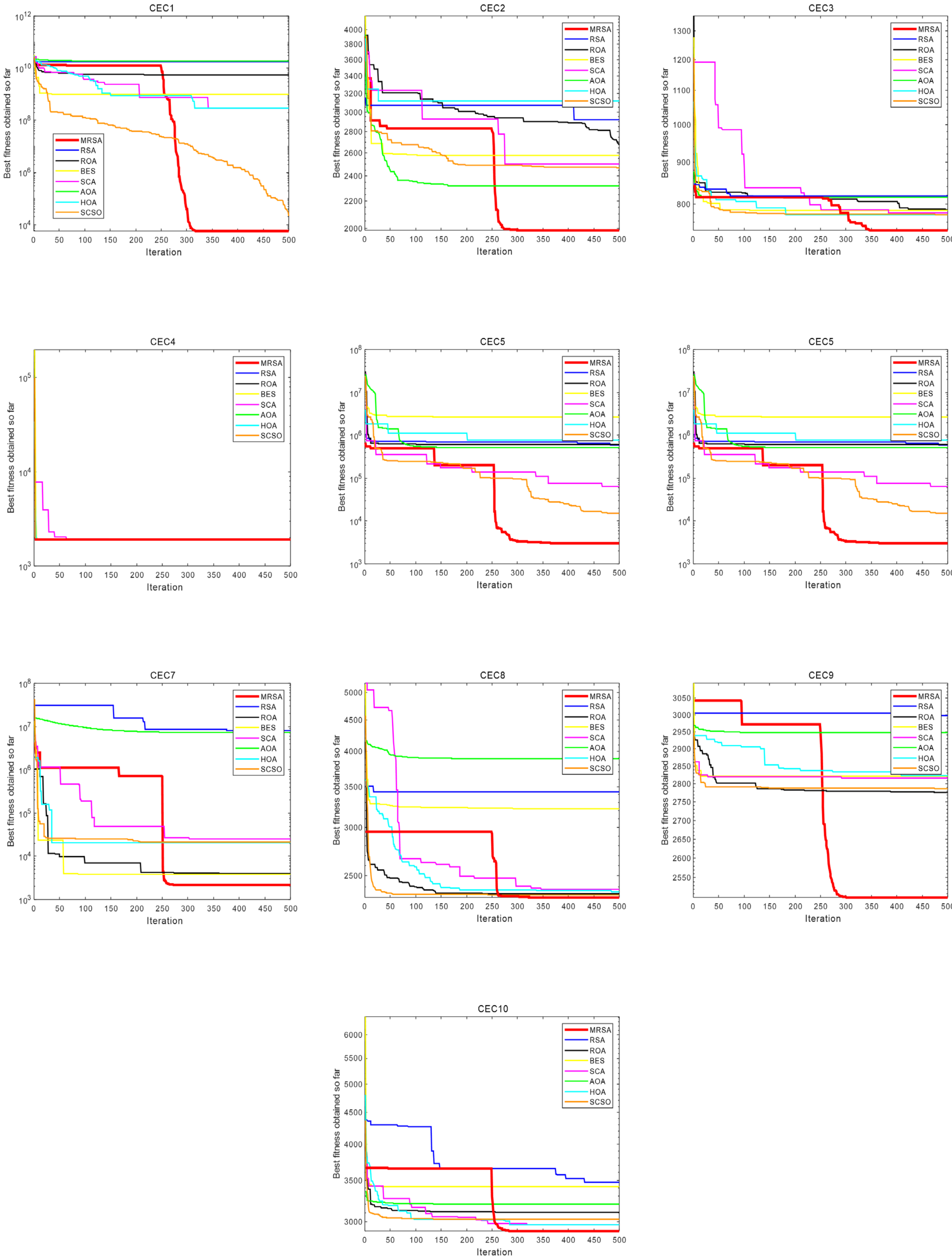


Figure 9. MRSA and other algorithms' convergence performance in CEC2020 functions.

To further verify MRSA's performance, this section gives the statistical results of MRSA running 30 times in CEC2020 functions. Table 5 gives the specific results. CEC1 is a unimodal function, CEC2–4 are essential functions, CEC5–7 are hybrid functions, and CEC8–10 are composition functions. The dimension is 10. It is not difficult to see that MRSA is very effective in most measurement functions. Especially in CEC1, MRSA's effect is better than the compared algorithm. Only in CEC5 the effect of MRSA is not very ideal. In order to show the statistical results' distribution of MRSA and compared algorithm, Figure 8 shows the box chart of statistical data. In Figure 8, the lines above and below the box represent the data set's maximum and minimum. The box's upper and lower sides represent the upper and lower quadrant, respectively. The lines in the middle of the box represent the median value, and '+' represent abnormal values. It can be seen that MRSA's fluctuation amplitude in most functions is small. The fluctuation range in CEC9 is extensive, and the reason is that MRSA can frequently jump out of the local optimum.

Figure 9 shows MRSA's convergence performance in the CEC2020 function. From the convergence curve of CEC1, 2, 5, 6, 7, 8, 9 and 10, it is not difficult to see that although MRSA's convergence ability is insufficient in the early stage, it can jump out of the local optimum in the middle and late stages.

Table 6 shows the wilcoxon rank-sum test results of MRSA and compared algorithms in CEC2020 functions. In CEC4, several algorithms' p-value is 1. As can be seen from Table 5, most algorithms can find the optimal value stably, so the difference is slight. In other functions, the data set obtained by MRSA is almost significantly different from that obtained by other algorithms. Combining Table 5 and Table 6, it is not difficult to see that MRSA can solve complex functions.

In CEC2020 funtions, from in-depth analysis of statistical results, the statistics obtained by MRSA are significantly better than other comparison algorithms. It can be seen from the convergence curve that MRSA can continue to converge in the middle and later stages thanks to the multi-hunting cooperation strategy after it stagnates in the early stage.

Table 6. Results of wilcoxon rank-sum test in CEC2020 functions.

Function	MRSA VS RSA	MRSA VS ROA	MRSA VS BES	MRSA VS SCA	MRSA VS AOA	MRSA VS HOA	MRSA VS SCSO
CEC1	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}	1.73×10^{-06}
CEC2	1.73×10^{-06}	4.90×10^{-04}	1.73×10^{-06}	1.73×10^{-06}	1.48×10^{-04}	1.73×10^{-06}	1.73×10^{-06}
CEC3	4.73×10^{-06}	$\mathbf{1.99\times10^{-01}}$	1.20×10^{-03}	$\mathbf{8.94\times10^{-01}}$	1.59×10^{-03}	$\mathbf{3.29\times10^{-01}}$	6.34×10^{-06}
CEC4	$\mathbf{1.00\times10^{+00}}$	$\mathbf{1.00\times10^{+00}}$	$\mathbf{1.00\times10^{+00}}$	5.96×10^{-05}	$\mathbf{1.00\times10^{+00}}$	6.10×10^{-05}	$\mathbf{1.00\times10^{+00}}$
CEC5	2.13×10^{-06}	8.73×10^{-03}	1.04×10^{-03}	4.07×10^{-05}	1.60×10^{-04}	8.19×10^{-05}	2.13×10^{-06}
CEC6	1.80×10^{-05}	3.87×10^{-02}	$\mathbf{1.20\times10^{-01}}$	3.87×10^{-02}	1.71×10^{-03}	1.15×10^{-04}	1.02×10^{-05}
CEC7	1.49×10^{-05}	3.68×10^{-02}	8.19×10^{-05}	1.96×10^{-03}	5.79×10^{-05}	4.68×10^{-03}	2.60×10^{-05}
CEC8	1.73×10^{-06}	1.89×10^{-04}	1.73×10^{-06}	2.37×10^{-05}	1.73×10^{-06}	4.07×10^{-05}	1.73×10^{-06}
CEC9	1.80×10^{-05}	2.30×10^{-02}	2.70×10^{-02}	1.29×10^{-03}	6.89×10^{-05}	4.39×10^{-03}	2.84×10^{-05}
CEC10	1.73×10^{-06}	1.48×10^{-04}	1.73×10^{-06}	4.45×10^{-05}	1.73×10^{-06}	4.11×10^{-03}	1.73×10^{-06}

4.3. Ablation experiments

Table 7. Results of ablation experiments.

Function	Statistics	MRSA	MutiRSA	RSALOBL	RSARS	LMRAOA	RSA
CEC1	Best	**$1.00\times10^{+02}$**	$1.01\times10^{+02}$	$4.20\times10^{+09}$	$7.87\times10^{+09}$	$2.24\times10^{+02}$	$5.99\times10^{+09}$
	Mean	**$2.23\times10^{+03}$**	$5.44\times10^{+07}$	$1.34\times10^{+10}$	$1.61\times10^{+10}$	$3.79\times10^{+03}$	$1.14\times10^{+10}$
	Std	**$2.08\times10^{+03}$**	$2.98\times10^{+08}$	$3.96\times10^{+09}$	$3.71\times10^{+09}$	$3.46\times10^{+03}$	$3.90\times10^{+09}$
CEC2	Best	**$1.34\times10^{+03}$**	$1.51\times10^{+03}$	$2.34\times10^{+03}$	$2.47\times10^{+03}$	$1.45\times10^{+03}$	$2.56\times10^{+03}$
	Mean	$1.91\times10^{+03}$	$2.31\times10^{+03}$	$2.81\times10^{+03}$	$2.80\times10^{+03}$	**$1.72\times10^{+03}$**	$2.87\times10^{+03}$
	Std	$1.45\times10^{+02}$	$4.53\times10^{+02}$	$2.09\times10^{+02}$	**$6.80\times10^{+01}$**	$2.83\times10^{+02}$	$1.80\times10^{+02}$
CEC3	Best	**$7.17\times10^{+02}$**	$7.28\times10^{+02}$	$7.97\times10^{+02}$	$7.93\times10^{+02}$	$7.49\times10^{+02}$	$8.01\times10^{+02}$
	Mean	**$7.70\times10^{+02}$**	$7.82\times10^{+02}$	$8.15\times10^{+02}$	$8.17\times10^{+02}$	$7.80\times10^{+02}$	$8.15\times10^{+02}$
	Std	$2.03\times10^{+01}$	$2.52\times10^{+01}$	**$8.57\times10^{+00}$**	$1.11\times10^{+01}$	$1.75\times10^{+01}$	$1.30\times10^{+01}$
CEC4	Best	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**
	Mean	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**	**$1.90\times10^{+03}$**
	Std	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**
CEC5	Best	**$1.80\times10^{+03}$**	$3.78\times10^{+03}$	$2.04\times10^{+05}$	$3.55\times10^{+05}$	$2.03\times10^{+03}$	$3.76\times10^{+05}$
	Mean	$1.17\times10^{+05}$	$1.96\times10^{+05}$	$4.80\times10^{+05}$	$5.24\times10^{+05}$	**$4.25\times10^{+03}$**	$5.24\times10^{+05}$
	Std	$7.97\times10^{+04}$	$1.39\times10^{+05}$	$1.03\times10^{+05}$	$6.45\times10^{+04}$	**$4.58\times10^{+03}$**	$1.55\times10^{+05}$
CEC6	Best	**$1.60\times10^{+03}$**	$1.60\times10^{+03}$	$2.02\times10^{+03}$	$1.85\times10^{+03}$	$1.60\times10^{+03}$	$2.06\times10^{+03}$
	Mean	**$1.83\times10^{+03}$**	$1.95\times10^{+03}$	$2.33\times10^{+03}$	$2.26\times10^{+03}$	$1.88\times10^{+03}$	$2.34\times10^{+03}$
	Std	**$1.23\times10^{+02}$**	$2.11\times10^{+02}$	$2.19\times10^{+02}$	$2.17\times10^{+02}$	$1.44\times10^{+02}$	$2.58\times10^{+02}$
CEC7	Best	**$2.12\times10^{+03}$**	$2.12\times10^{+03}$	$3.19\times10^{+04}$	$1.67\times10^{+04}$	$2.12\times10^{+03}$	$2.99\times10^{+04}$
	Mean	$7.88\times10^{+03}$	$5.75\times10^{+05}$	$2.66\times10^{+06}$	$1.94\times10^{+06}$	**$2.45\times10^{+03}$**	$1.92\times10^{+06}$
	Std	$8.17\times10^{+03}$	$2.87\times10^{+06}$	$2.67\times10^{+06}$	$1.84\times10^{+06}$	**$3.86\times10^{+02}$**	$3.27\times10^{+06}$
CEC8	Best	**$2.21\times10^{+03}$**	$2.23\times10^{+03}$	$2.83\times10^{+03}$	$2.75\times10^{+03}$	$2.30\times10^{+03}$	$2.87\times10^{+03}$
	Mean	$2.31\times10^{+03}$	$2.39\times10^{+03}$	$3.19\times10^{+03}$	$3.20\times10^{+03}$	**$2.30\times10^{+03}$**	$3.33\times10^{+03}$
	Std	$1.24\times10^{+01}$	$2.68\times10^{+02}$	$2.34\times10^{+02}$	$2.86\times10^{+02}$	**$1.04\times10^{+00}$**	$4.28\times10^{+02}$
CEC9	Best	**$2.42\times10^{+03}$**	$2.50\times10^{+03}$	$2.73\times10^{+03}$	$2.82\times10^{+03}$	$2.50\times10^{+03}$	$2.83\times10^{+03}$
	Mean	**$2.66\times10^{+03}$**	$2.76\times10^{+03}$	$2.87\times10^{+03}$	$2.94\times10^{+03}$	$2.73\times10^{+03}$	$2.90\times10^{+03}$
	Std	$1.46\times10^{+02}$	$1.09\times10^{+02}$	**$5.54\times10^{+01}$**	$6.07\times10^{+01}$	$9.95\times10^{+01}$	$8.32\times10^{+01}$
CEC10	Best	**$2.60\times10^{+03}$**	$2.90\times10^{+03}$	$3.22\times10^{+03}$	$3.27\times10^{+03}$	$2.90\times10^{+03}$	$3.25\times10^{+03}$
	Mean	$2.93\times10^{+03}$	$2.97\times10^{+03}$	$3.45\times10^{+03}$	$3.51\times10^{+03}$	**$2.91\times10^{+03}$**	$3.47\times10^{+03}$
	Std	**$2.25\times10^{+01}$**	$1.28\times10^{+02}$	$1.89\times10^{+02}$	$2.08\times10^{+02}$	$8.68\times10^{+01}$	$2.29\times10^{+02}$

This paper adopts three strategies to improve RSA. To reflect the impact of a single strategy on RSA, this paper uses the CEC2020 test functions to test. This section compares MRSA with MutiRSA, RSALOBL and RSALOBL. MutiRSA only adds the multi-hunting coordination strategy, RSALOBL only adds the LOBL strategy, and RSARS only adds the restart strategy. In addition, this section also introduces the variant of AOA (LMRAOA) for comparison, further reflecting MRSA's performance in solving CEC2020 test functions. The results of ablation experiment are shown in Table 7. It is not difficult to see that the three strategies have improved the RSA's performance. Moreover, the results of MRSA also have some advantages over the variant of AOA (LMRAOA).

5. Solutions to constrained engineering design problems

The main goal of improving existing MAs is to solve practical problems. To verify the engineering applicability of MRSA, six classical engineering problems are selected in this section. The problems are welding beam design, pressure vessel design, tension/compression spring design, speed reducer design, corrugated bulkhead design and multiple disc clutch brake design.

5.1. The welded beam design problem

The problem with the welded beam is finding the minimum weight to reduce the cost. It includes four variables: the length of the beam (x_1), the height of the beam (x_2), the thickness of the beam (x_3) and the thickness of the weld (x_4). At the same time, the shear stress, bending stress, beam bending load, end deviation, etc., also need to meet the constraints. The model of the welded beam design problem is shown in Figure 10.

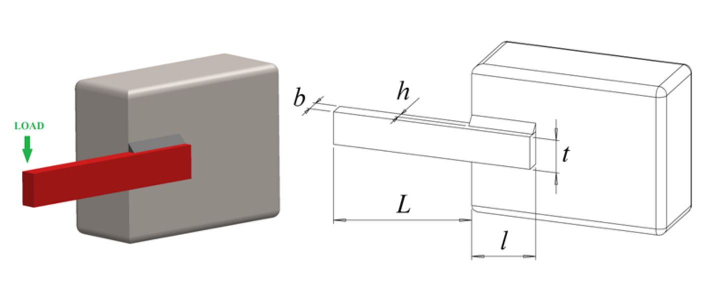


Figure 10. Model of welded beams design.

The mathematical model of welded beam design is as follows:
Construct objective function:

$$f(x) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2) \tag{22}$$

Constraint condition:

$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \le 0 \tag{23}$$

$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \le 0 \tag{24}$$

$$g_3\left(\vec{x}\right)=\delta\left(\vec{x}\right)-\delta_{\max}\leq 0 \tag{25}$$

$$g_4\left(\vec{x}\right)=x_1-x_4\leq 0 \tag{26}$$

$$g_5\left(\vec{x}\right)=P-P_c\left(\vec{x}\right)\leq 0 \tag{27}$$

$$g_6\left(\vec{x}\right)=0.125-x_1\leq 0 \tag{28}$$

$$g_7\left(\vec{x}\right)=1.10471x_1^2+0.04811x_3x_4\left(14.0+x_2\right)-0.5\leq 0 \tag{29}$$

Parameter solving:

$$\tau\left(\vec{x}\right)=\sqrt{\left(\tau^{'}\right)^2+2\tau^{'}\tau^{''}\frac{x_2}{2R}+\left(\tau^{''}\right)},\tau^{'}=\frac{P}{\sqrt{2x_1x_2}},\tau^{''}=\frac{MR}{J} \tag{30}$$

$$M=P\left(L+\frac{x_2}{2}\right),R=\sqrt{\frac{x_2^2}{4}+\left(\frac{x_1+x_3}{2}\right)^2},\sigma\left(\vec{x}\right)=\frac{6PL}{x_4x_3^2} \tag{31}$$

$$J=2\left\{\sqrt{2x_1x_2}\left[\frac{x_x^2}{4}+\left(\frac{x_1+x_3}{2}\right)^2\right]\right\},\delta\left(\vec{x}\right)=\frac{6PL^3}{Ex_4x_3^2} \tag{32}$$

$$P_c\left(\vec{x}\right)=\frac{4.013E\sqrt{\frac{x_3^2x_4^6}{0}}}{L^2},\left(1-\frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),\left(1-\frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{33}$$

$$P=6000lb,L=14\ in,\delta_{\max}=0.25\ \text{ in},E=30\times 10^6\ psi \tag{34}$$

$$\tau_{\max}=13600\ psi,and\ \ \sigma_{\max}=30000\ \ psi \tag{35}$$

Boundary constraint:

$$0.1\leq x_i\leq 2,i=1,4;0.1\leq x_i\leq 10,i=2.3 \tag{36}$$

Table 8 shows the experimental results of MRSA in solving the welded beams design problem. When x_1 = 0.205739392, x_2 = 3.252967354, x_3=9.036552395, x_4 = 0.205732954, MRSA obtained the best result (1.695257579). The solution obtained by other algorithms is obviously inferior to those obtained by MRSA.

Table 8. Different algorithms' optimal solutions for the welded beam design problem.

Algorithm	Optimal values for variables				Optimum weight
	x_1	x_2	x_3	x_4	
MRSA	0.205739392	3.252967354	9.036552395	0.205732954	1.695257579
ROA [44]	0.200077	3.365754	9.011182	0.206893	1.706447
GWO [8]	0.205676	3.478377	9.03681	0.205778	1.72624
WOA [49]	0.205396	3.484293	9.037426	0.206276	1.730499
RO [50]	0.203687	3.528467	9.004233	0.207241	1.735344
MPA [51]	0.205728	3.470509	9.036624	0.20573	1.724853
MVO [52]	0.205463	3.473193	9.044502	0.205695	1.72645
AOA [17]	0.194475	2.57092	10	0.201827	1.7164
HHO [53]	0.204039	3.531061	9.027463	0.206147	1.73199057
IHS [54]	0.20573	3.47049	9.03662	0.2057	1.7248
RSA [34]	0.203687	3.528467	9.004233	0.207241	1.735344

5.2. The pressure vessel design problem

The pressure vessel design is to meet the production demand and reduce the cost. The problem includes four variables: shell thickness (T_s), head thickness (T_h), inner radius (R) and vessel length (L). T_s and T_h are integral multiples of 0.625. R and L are continuous variables. The model of the pressure vessel design problem is shown in Figure 11.

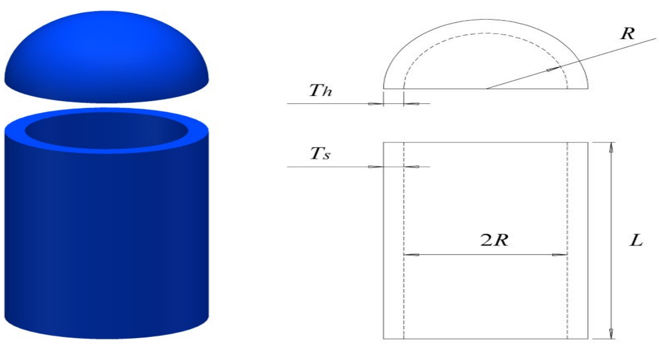


Figure 11. Model of pressure vessel design.

The mathematical model of the pressure vessel design problem is as follows:
We consider:

$$\vec{x} = [x_1\ x_2\ x_3\ x_4] = [T_s\ T_h\ R\ L] \tag{37}$$

Construct objective function:

$$f(\vec{x}) = 0.6224x_1x_2x_3 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{38}$$

Constraint condition:

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \le 0 \tag{39}$$

$$g_2(\vec{x}) = -x_3 + 0.00954x_3 \le 0 \tag{40}$$

$$g_3(\vec{x}) = -\pi x_3^2 x_4 + \frac{2}{3}\pi x_3^3 + 1296000 \le 0 \tag{41}$$

$$g_4(\vec{x}) = -x_4 - 240 \le 0 \tag{42}$$

Boundaries constraint:

$$\begin{gathered} 0 \le x_1 \le 99, 0 \le x_2 \le 99, 10 \le x_3 \le 200 \\ 10 \le x_4 \le 200 \end{gathered} \tag{42}$$

Table 9 gives the results of MRSA and other algorithms in pressure vessel design. When T_s is 0.758460965, T_h is 0.377162354, R is 41.10831839 and L is 189.3046068, MRSA obtained the lowest cost.

Table 9. Comparison of optimal solutions for the welded beam design problem.

Algorithm	Optimal values for variables				Optimum cost
	T_s	T_h	R	L	
MRSA	0.758460965	0.377162354	41.10831839	189.3046068	5765.42006
SHO [55]	0.77821	0.384889	40.31504	200	5885.5773
MPA [51]	0.77816876	0.38464966	40.31962084	199.9999935	5885.3353
SMA [56]	0.7931	0.3932	40.6711	196.2178	5994.1857
HPSO [57]	0.8125	0.4375	42.0984	176.6366	6059.7143
GWO [8]	0.8125	0.4345	42.089181	176.758731	6051.5639
DE [58]	0.8125	0.4375	42.098411	176.63769	6059.7340
COOT [59]	0.77817	0.384651	40.319618	200	5885.3487
AEO [60]	0.8374205	0.413937	43.389597	161.268592	5994.5070
CSS [61]	0.8125	0.4375	42.103624	176.572656	6059.0888

5.3. *The tension/compression spring design problem*

The tension/compression spring design is to obtain the minimum weight of the spring under four constraints. The problem has three variables: the average diameter of the spring coil (D), the diameter of the spring wire (d) and the adequate number of the spring coils (N). The specific model of the pressure spring problem is shown in Figure 12.

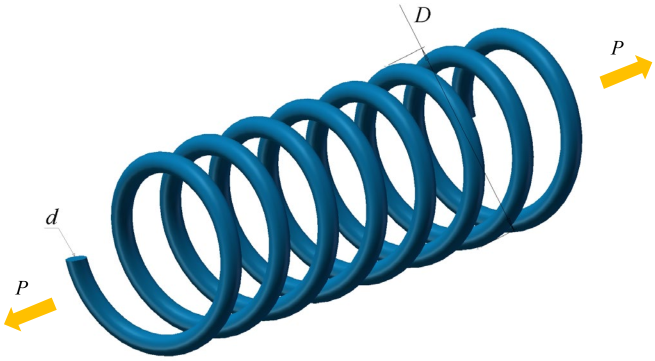


Figure 12. Model of tension/compression spring design.

The mathematical model of the tension/compression spring problem is as follows:
We consider:

$$x = [x_1\ x_2\ x_3] = [d\ D\ N] \tag{43}$$

Objective function:

$$f(x) = (x_3 + 2) \times x_2 \times x_1^2 \tag{44}$$

Subject to:

$$g_1(x) = 1 - \frac{x_3 \times x_2^3}{71785 \times x_1^4} \le 0 \tag{45}$$

$$g_2(x) = \frac{4 \times x_2^2 - x_1 \times x_2}{12566 \times x_1^4} + \frac{1}{5108 \times x_1^2} - 1 \le 0 \tag{46}$$

$$g_3(x) = 1 - \frac{140.45 \times x_1}{x_2^2 \times x_3} \le 0 \tag{47}$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \le 0 \tag{48}$$

Boundaries:

$$\begin{aligned} &0.05 \le x_1 \le 2.0; 0.25 \le x_2 \le 1.3; \\ &2.0 \le x_3 \le 15.0 \end{aligned} \tag{49}$$

Table 10 shows the results of MRSA and other algorithms for the tension/compression spring design. It can be seen that MRSA has a great effect on the tension/compression spring design. MRSA obtained the optimal solution of 0.009913786, which is obviously superior to the basic RSA.

Table 10. Comparison of optimal solutions for the tension/compression spring design problem.

Algorithm	Optimal values for variables			Optimum Value
	D	d	n	
MRSA	0.05	0.373434558	8.619033937	0.009913786
RSA [34]	0.057814	0.58478	4.0167	0.01176
MVO [52]	0.05251	0.37602	10.33513	0.01279
WOA [49]	0.051207	0.345215	12.004032	0.0126763
CSCA [52]	0.051609	0.354714	11.410831	0.0126702
AOA [17]	0.05	0.349809	11.8637	0.012124
RO [50]	0.05137	0.349096	11.76279	0.0126788
PFA [63]	0.051726	0.357629	11.235724	0.012665

5.4. The speed reducer design problem

The model of the speed reducer design problem is shown in Figure 13. This problem ensures that the speed reducer can meet the constraint conditions and achieve the minimum mass. There are seven design variables in this problem. We set the width of the tooth surface (x_1), gear module (x_2), the number of teeth on the pinion (x_3), length of the first shaft between bearings (x_4), length of the second shaft between bearings (x_5), the diameter of the first shaft (x_6) and diameter of the second shaft (x_7).

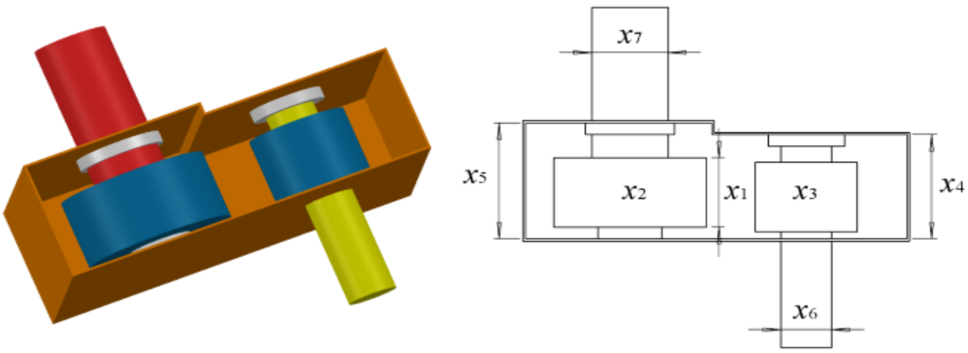


Figure 13. Model of speed reducer design.

The mathematical model and constraints of the speed reducer problem are as follows:
Objective function:

$$f(\vec{x}) = 07854 \times x_1 \times x_2^2 \times (3.3333 \times x_3^2 + 14.9334 \times x_3 - 43.0934) - 1.508 \times x_1 \times (x_6^2 + x_7^2) + 7.4777 \times x_6^3 + x_7^3 + 0.7854 \times x_4 \times x_6^2 + x_5 \times x_7^2 \tag{50}$$

Subject to:

$$g_1(\vec{x}) = \frac{27}{x_1 \times x_2^2 \times x_3} - 1 \le 0 \tag{52}$$

$$g_2(\vec{x}) = \frac{397.5}{x_1 \times x_2^2 \times x_3^2} - 1 \le 0 \tag{53}$$

$$g_3(\vec{x}) = \frac{1.93 \times x_4^3}{x_2 \times x_3 \times x_6^4} - 1 \le 0 \tag{54}$$

$$g_4(\vec{x}) = \frac{1.93 \times x_5^3}{x_2 \times x_3 \times x_7^4} - 1 \le 0 \tag{55}$$

$$g_5(\vec{x}) = \frac{1}{110 \times x_6^3} \times \sqrt{(\frac{745 \times x_4}{x_2 \times x_3})^2 + 16.9 \times 10^6} - 1 \le 0 \tag{56}$$

$$g_6(\vec{x}) = \frac{1}{85 \times x_7^3} \times \sqrt{(\frac{745 \times x_5}{x_2 \times x_3})^2 + 16.9 \times 10^6} - 1 \le 0 \tag{57}$$

$$g_7(\vec{x}) = \frac{x_2 \times x_3}{40} - 1 \le 0 \tag{58}$$

$$g_8(\vec{x}) = \frac{5 \times x_2}{x_1} - 1 \le 0 \tag{59}$$

$$g_9(\vec{x}) = \frac{x_1}{12 \times x_2} - 1 \le 0 \tag{60}$$

$$g_{10}(\vec{x}) = \frac{1.5 \times x_6 + 1.9}{x_4} - 1 \le 0 \tag{61}$$

$$g_{11}(\vec{x}) = \frac{1.1 \times x_7 + 1.9}{x_5} - 1 \le 0 \tag{62}$$

Boundaries:

$$\begin{aligned} &2.6 \le x_1 \le 3.6, 0.7 \le x_2 \le 0.8, 17 \le x_3 \le 28, 7.3 \le x_4 \le 8.3, 7.3 \le x_5 \le 8.3, \\ &2.9 \le x_6 \le 3.9, 5 \le x_7 \le 5.5 \end{aligned} \tag{63}$$

It is not difficult to see that MRSA has a good effect on speed reducer design. Table 11 shows MRSA's solution in speed reducer design. The solution obtained by MRSA is X = [3.476415091, 0.7, 17, 7.3, 7.8, 3.348630145, 5.276783057], which is obviously superior to RSA's and other comparison algorithms.

Table 11. Comparison of optimal solutions for the tension/compression spring design problem.

Algorithm	Optimal values for variables							Optimal weight
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
MRSA	3.476415091	0.7	17	7.3	7.8	3.348630145	5.276783057	2988.271359
RSA [34]	3.50279	0.7	17	7.30812	7.74715	3.35067	5.28675	2996.5157
hHHO-SCA [64]	3.506119	0.7	17	7.3	7.99141	3.452569	5.286749	3029.873076
MROA [65]	3.497571	0.7	17	7.3	7.8	3.350057265	5.28553957	2995.437447
AAO [66]	3.499	0.6999	17	7.3	7.8	3.3502	5.2872	2996.783
APSO [67]	3.501313	0.7	18	8.127814	8.042121	3.352446	5.287076	3187.630486
MFO [68]	3.497455	0.7	17	7.82775	7.712457	3.351787	5.286352	2998.94083
WSA [69]	3.5	0.7	17	7.3	7.8	3.350215	5.286683	2996.348225
CS [70]	3.5015	0.7	17	7.605	7.8181	3.352	5.2875	3000.981
PDO [71]	3.497777468	0.7	17.00002761	7.300100314	7.800675175	3.351095015	5.296455378	2993.7
DMOA [72]	3.497599093	0.7	17	7.3	7.713534977	3.350055806	5.285631197	3010.4

5.5. The corrugated bulkhead design problem

The corrugated bulkhead design is a problem of minimizing the corrugated bulkhead's mass. It includes four design variables: width (x_1), depth (x_2), length (x_3) and plate thickness (x_4). The model of this problem is shown in Figure 14.

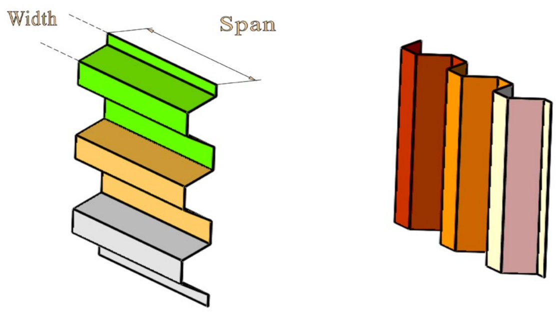


Figure 14. Model of the corrugated bulkhead design problem.

Its mathematical model and constraints are as follows:
Objective function:

$$f(x)=\frac{5.885x_4(x_1+x_3)}{x_1+\sqrt{|x_3^2-x_2^2|}} \tag{64}$$

Constraints:

$$g_1(X) = -x_4 x_2 (0.4 x_1 + \frac{x_3}{6}) + 8.94(x_1 + \sqrt{|x_3^2 - x_2^2|}) \leq 0 \tag{65}$$

$$g_2(X) = -x_4 x_2^2 (0.2 x_1 + \frac{x_3}{12}) + 2.2(8.94(x_1 + \sqrt{|x_3^2 - x_2^2|}))^{4/3} \leq 0 \tag{66}$$

$$g_3(X) = -x_4 + 0.0156 x_1 + 0.15 \leq 0 \tag{67}$$

$$g_4(X) = -x_4 + 0.0156 x_3 + 0.15 \leq 0 \tag{68}$$

$$g_5(X) = -x_4 + 1.05 \leq 0 \tag{69}$$

$$g_6(X) = -x_3 + x_2 \leq 0 \tag{70}$$

Boundaries constraints:

$$0 \leq x_1, x_2, x_3 \leq 100,\ 0 \leq x_4 \leq 5 \tag{71}$$

Table 12 shows that when X = [57.69230749, 34.14762033, 57.69230747, 1.05], MRSA obtained the optimal solution (6.842958018).

Table 12. Comparison of optimal solutions for the corrugated bulkhead design problem.

Algorithm	Optimal values for variables				Optimal cost
	x_1	x_2	x_3	x_4	
MRSA	57.69230749	34.14762033	57.69230747	1.05	6.842958018
PDO [71]	48.31191	54.78270401	61.92983	0.424913	6.9821
FA [73]	37.1179498	33.035021	37.1939476	0.7306255	7.21
LF-FA [73]	57.69231	34.14762	57.69231	1.05	6.95
LS-LF-FA [73]	57.69277	34.13296	57.55294	1.05007	6.86
AOA [72]	57.69277	34.13296	57.55294	1.05007	481.97
BBO [74]	57.69231	34.14762	57.69231	1.05	2.79×10^{12}

5.6. The multiple disc clutch brake design problem

The multiple disc clutch brake design is a problem in finding the minimum mass and meeting some constraints. It has five variables: the inner radius (x_1), the outer radius (x_2), the disc thickness (x_3), the driving force (x_4) and the number of skin friction (x_5). The specific model is shown in Figure 15.

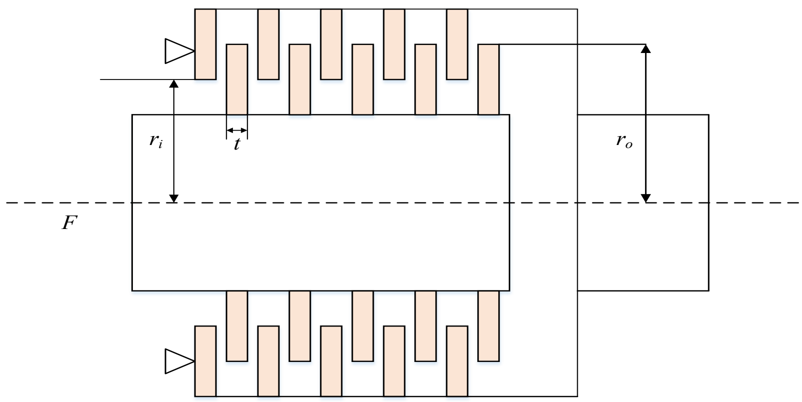


Figure 15. Model of multiple disc clutch brake design.

The mathematical model of multiple disc clutch brake design is as follows:
Objective function:

$$\vec{x} = [x_1\ x_2\ x_3\ x_4\ x_5] = [r_i\ r_o\ \ t\, F\, Z] \tag{72}$$

Objective function:

$$f(x) = \Pi\left(r_o^2 - r_i^2\right)t(Z+1)\rho \quad (\rho = 0.0000078) \tag{73}$$

Subject to:

$$g_1(x) = r_o - r_i - \Delta r \geq 0 \tag{74}$$

$$g_2(x) = l_{max} - (Z+1)(t+\delta) \geq 0 \tag{75}$$

$$g_3(x) = P_{max} - P_{rz} \geq 0 \tag{76}$$

$$g_4(x) = P_{max}\nu_{sr\ max} - P_{rz}\upsilon_{sr} \geq 0 \tag{77}$$

$$g_5(x) = \nu_{sr\ max} - \upsilon_{sr} \geq 0 \tag{78}$$

$$g_6(x) = T_{max} - T \geq 0 \tag{79}$$

$$g_7(x) = M_h - sM_s \geq 0 \tag{80}$$

$$g_8(x) = T \geq 0 \tag{81}$$

Variable range:

$$60 \le x_1 \le 80, 90 \le x_2 \le 110, 1 \le x_3 \le 3, 600 \le x_4 \le 1000, 2 \le x_5 \le 9 \tag{82}$$

Parameters:

$$M_h = \frac{2}{3}\mu FZ \frac{r_o^3 - r_i^2}{r_o^2 - r_i^3}, P_{rz} = \frac{F}{\Pi\left(r_o^2 - r_i^2\right)} \tag{83}$$

$$\upsilon_{rz} = \frac{2\Pi\left(r_o^3 - r_i^3\right)}{90\left(r_o^2 - r_i^2\right)}, T = \frac{I_z \Pi\, n}{30\left(M_h + M_f\right)} \tag{84}$$

$$\Delta r = 20mm, I_z = 55kgmm^2, P_{\max} = 1MPa, F_{\max} = 1000N \tag{85}$$

$$T_{\max} = 15s, \mu = 0.5, s = 1.5, M_s = 40Nm, M_f = 3Nm \tag{86}$$

$$n = 250rpm, \upsilon_{sr\ \max} = 10\, m/s, l_{\max} = 30mm \tag{87}$$

Table 13 shows the solutions of MRSA and other algorithms on the multiple disc clutch brake design. MRSA obtained the minimum mass of 0.235242553 when X = [69.99999072, 90, 1, 635.6851083, 2].

Table 13. Comparison of optimal solutions for the corrugated bulkhead design problem.

Algorithm	Optimal values for variables					Optimum
	x_1	x_2	x_3	x_4	x_5	weight
MRSA	69.99999072	90	1	635.6851083	2	0.235242553
TLBO [38]	70	90	1	810	3	0.313656611
WCA [75]	70	90	1	910	3	0.313656
HHO [53]	70	90	1	1000	2.3128	0.259768993
CMVO [76]	70	90	1	910	3	0.313656
QSMFO [77]	80	101.3002	3	600	9	0.2902
RSA [34]	70.0347	90.0349	1	801.7285	2.974	0.31176

6. Conclusions

This paper proposed a multi-hunting coordination strategy by combining Lagrange interpolation with TLBO's student stage. Replace the original hunting coordination stage with the proposed multi-hunting coordination strategy. It both enhanced the algorithm's exploration and exploitation. At the same time, the LOBL strategy and restart strategy are added to improve the global performance of the algorithm. Through solving the test functions of different dimensions, the statistical data shows that MRSA has great advantages in low dimensional simple problems and high dimensional complex problems compared with other original algorithms and improved algorithms. When solving engineering problems, the results obtained by MRSA are also significantly better than other algorithms.

Although the MRSA proposed in this paper has greatly improved its performance compared with RSA. However, MRSA also increases the computational complexity. In the future, we will continue to improve the performance of MRSA and reduce its complexity. We also try to enable it to solve more engineering problems such as path planning and multiple image segmentation. In addition, in order to solve more practical problems, we will try to propose the multi-objective version of MRSA in the future.

Acknowledgments

This work was funded by National Education Science Planning Key Topics of the Ministry of Education—"Research on the core quality of applied undergraduate teachers in the intelligent age" (DIA220374).

Conflict of interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. P. Toth, D. Vigo, An exact algorithm for the vehicle routing problem with backhauls, *Transport. Sci.*, **31** (1997), 372–385. https://doi.org/10.1287/trsc.31.4.372
2. J. R. Cash, A. H. Karp, A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides, *ACM T. Math. Software*, **16** (1990), 201–222. https://doi.org/10.1145/79505.79507
3. Z. Beheshti, S. M. H. Shamsuddin, A review of population-based meta-heuristic algorithms, *Int. J. Adv. Soft Comput. Appl.*, **5** (2013), 1–35.
4. A. Dahou, M. A. Elaziz, S. A. Chelloug, M. A. Awadallah, M. A. Al-Betar, M. A. A. Al-qaness, et al., Intrusion detection system for iot based on deep learning and modified reptile search algorithm, *Comput. Intell. Neurosc.*, **2022** (2022). https://doi.org/10.1155/2022/6473507
5. F. Agostino, Heuristic recommendation technique in internet of things featuring swarm intelligence approach, *Expert Syst. Appl.*, **187** (2022), 115904. https://doi.org/10.1016/j.eswa.2021.115904
6. F. Agostino, C. Mastroianni, G. Spezzano, Reorganization and discovery of grid information with epidemic tuning, *Future Gener. Comp. Syst.*, **24** (2008), 788–797. https://doi.org/10.1016/j.future.2008.04.001
7. T. Fearn, Particle swarm optimization, *NIR News*, **25** (2014), 27. https://doi.org/10.1255/nirn.1476
8. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.*, **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
9. G. G. Wang, S. Deb, Z. Cui, Monarch butterfly optimization, *Neural Comput. Appl.*, **31** (2019), 1995–2014. https://doi.org/10.1007/s00521-015-1923-y
10. G. G. Wang, Moth search algorithm: A bio–inspired metaheuristic algorithm for global optimization problems, *Memet. Comput.*, **10** (2018), 151–164. https://doi.org/10.1007/s12293-016-0212-3

11. Y. Yang, H. Chen, A. A. Heidari, A. H Gandomi, Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts, *Expert Syst. Appl.*, **177** (2021), 114864. https://doi.org/10.1016/j.eswa.2021.114864
12. J. Tu, H. Chen, M. Wang, A. H Gandomi, The colony predation algorithm, *J. Bionic Eng.*, **18** (2021), 674–710. https://doi.org/10.1007/s42235-021-0050-y
13. J. H. Holland, Genetic algorithms, *Sci. Am.*, **267** (1992), 66–73. https://doi.org/10.1038/scientificamerican0792-66
14. A. K. Qin, V. L. Huang, P. N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, *IEEE T. Evolut. Comput.*, **13** (2008), 398–417. https://doi.org/10.1109/TEVC.2008.927706
15. E. Rashedi, H. Nezamabadi–Pour, S. Saryazdi, GSA: A gravitational search algorithm, *Inf. Sci.*, **179** (2009), 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
16. S. Mirjalili, SCA: A sine cosine algorithm for solving optimization problems, *Knowl. Based Syst.*, **96** (2016), 120–133. https://doi.org/10.1016/j.knosys.2015.12.022
17. L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz, A. H. Gandomi, The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Eng.*, **376** (2021), 113609. https://doi.org/10.1016/j.cma.2020.113609
18. I. Ahmadianfar, A. A. Heidari, S. Noshadian, A. H Gandomi, INFO: An efficient optimization algorithm based on weighted mean of vectors, *Expert Syst. Appl.*, **195** (2022), 116516. https://doi.org/10.1016/j.eswa.2022.116516
19. Z. Z. Liu, D. H. Chu, C. Song, X. Xue, B. Y. Lu, Social learning optimization (SLO) algorithm paradigm and its application in QoS–aware cloud service composition, *Inform. Sci.*, **326** (2016), 315–333. https://doi.org/10.1016/j.ins.2015.08.004
20. Y. Zhang, Z. Jin, Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems, *Expert Syst. Appl.*, **148** (2020), 113246. https://doi.org/10.1016/j.eswa.2020.113246
21. D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.*, **1** (1997), 67–82. https://doi.org/10.1109/4235.585893
22. S. Gupta, K. Deep, A hybrid self-adaptive sine cosine algorithm with opposition based learning, *Expert Syst. Appl.*, **119** (2019), 210–230. https://doi.org/10.1016/j.eswa.2018.10.050
23. E. H. Houssein, B. E. Helmy, H. Rezk, A. M. Nassef, An enhanced Archimedes optimization algorithm based on local escaping operator and orthogonal learning for PEM fuel cell parameter identification, *Eng. Appl. Artif. Intell.*, **103** (2021), 104309. https://doi.org/10.1016/j.engappai.2021.104309
24. H. Dong, J. He, H. Huang, W. Hou, Evolutionary programming using a mixed mutation strategy, *Inform. Sci.*, **1** (2007), 312–327. https://doi.org/10.1016/j.ins.2006.07.014
25. Z. M. Gao, J Zhao, Y. J. Zhang, Review of chaotic mapping enabled nature-inspired algorithms, *Math. Biosci. Eng.*, **19** (2022), 8215–8258.
26. Y. J. Zhang, J. Zhao, Z. M. Gao, Hybridized improvement of the chaotic Harris Hawk optimization algorithm and Aquila optimizer, in *International Conference on Electronic Information Engineering and Computer Communication (EIECC 2021)*, **12172** (2022), 327–332. https://doi.org/10.1117/12.2634395

27. Y. J. Zhang, Y. X. Yan, J. Zhao, Z. M. Gao, AOAAO: The hybrid algorithm of arithmetic optimization algorithm with aquila optimizer, *IEEE Access*, **10** (2022), 10907–10933. https://doi.org/10.1109/ACCESS.2022.3144431
28. Y. J. Zhang, Y. X. Yan, J. Zhao, Z. M. Gao, CSCAHHO: Chaotic hybridization algorithm of the Sine Cosine with Harris Hawk optimization algorithms for solving global optimization problems, *Plos one*, **17** (2022), e0263387. https://doi.org/10.1371/journal.pone.0263387
29. J. Zhao, Z. M. Gao, Y. J. Zhang, Piecewise linear map enabled Harris Hawk optimization algorithm, *J. Phys. Conf. Ser.*, **1994** (2021), 012038. https://doi.org/10.1088/1742-6596/1994/1/012038
30. M. M. Emam, E. H. Houssein, R. M. Ghoniem, A modified reptile search algorithm for global optimization and image segmentation: Case study brain MRI images, *Comput. Biol. Med.*, **152** (2023), 106404. https://doi.org/10.1016/j.compbiomed.2022.106404
31. S. Chakraborty, A K Saha, S Nama, S. Debnath, COVID-19 X-ray image segmentation by modified whale optimization algorithm with population reduction, *Comput. Biol. Med.*, **139** (2021), 104984. https://doi.org/10.1016/j.compbiomed.2021.104984
32. G. I. Sayed, M. M. Soliman, A. E. Hassanien, A novel melanoma prediction model for imbalanced data using optimized SqueezeNet by bald eagle search optimization, *Comput. Biol. Med.*, **136** (2021), 104712. https://doi.org/10.1016/j.compbiomed.2021.104712
33. J. Piri, P. Mohapatra, An analytical study of modified multi-objective Harris Hawk Optimizer towards medical data feature selection, *Comput. Biol. Med.*, **135** (2021), 104558. https://doi.org/10.1016/j.compbiomed.2021.104558
34. L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, A. H. Gandomi, Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer, *Expert Syst. Appl.*, **191**(2021), 116158. https://doi.org/10.1016/j.eswa.2021.116158
35. K. H. Almotairi, L. Abualigah, Hybrid reptile search algorithm and remora optimization algorithm for optimization tasks and data clustering, *Symmetry*, **14** (2022), 458. https://doi.org/10.3390/sym14030458
36. L. Huang, Y. Wang, Y. Guo, G. Hu, An improved reptile search algorithm based on lévy flight and interactive crossover strategy to engineering application, *Mathematics*, **10** (2022), 2329. https://doi.org/10.3390/math10132329
37. T. Sauer, Y. Xu, On multivariate Lagrange interpolation, *Math. Comput.*, **64** (1995), 1147–1170. https://doi.org/10.1090/S0025-5718-1995-1297477-5
38. R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems, *Comput. Aided Design.*, **43** (2011), 303–315. https://doi.org/10.1016/j.cad.2010.12.015
39. Q. Liu, N. Li, H. Jia, Q. Qi, L. Abualigah, Modified remora optimization algorithm for global optimization and multilevel thresholding image segmentation, *Mathematics*, **10** (2022), 1014. https://doi.org/10.3390/math10071014
40. H. Zhang, Z. Wang, W. Chen, A. A. Heidari, M. Wang, X. Zhao, et al., Ensemble mutation-driven salp swarm algorithm with restart mechanism: Framework and fundamental analysis, *Expert Syst. Appl.*, **165** (2021), 113897. https://doi.org/10.1016/j.eswa.2020.113897
41. H. Rao, H. Jia, D. Wu, C. Wen, S. Li, Q. Liu, L. Abualigah, A modified group teaching optimization algorithm for solving constrained engineering optimization problems, *Mathematics*, **10** (2022), 3765. https://doi.org/10.3390/math10203765

42. Z. Tongyi, W. Luo, An enhanced lightning attachment procedure optimization with quasi–opposition-based learning and dimensional search strategies, *Comput. Intell. Neurosc.*, **2019** (2019). https://doi.org/10.1155/2019/1589303
43. F. Y. Arini, S. Chiewchanwattana, C. Soomlek, K. Sunat, Joint opposite selection (JOS): A premiere joint of selective leading opposition and dynamic opposite enhanced Harris' hawks optimization for solving single–objective problems, *Expert Syst. Appl.*, **188** (2022), 116001. https://doi.org/10.1016/j.eswa.2021.116001
44. H. Jia, X. Peng, C. Lang, Remora optimization algorithm, *Expert Syst. Appl.*, **185** (2021), 115665. https://doi.org/10.1016/j.eswa.2021.115665
45. H. A. Alsattar, A. A. Zaidan, B. B. Zaidan, Novel meta-heuristic bald eagle search optimisation algorithm, *Artif. Intell. Rev.*, **53** (2020), 2237–2264. https://doi.org/10.1007/s10462-019-09732-5
46. F. MiarNaeimi, G. Azizyan, M. Rashki, Horse herd optimization algorithm: A nature-inspired algorithm for high–dimensional optimization problems, *Knowl. Based Syst.*, **213** (2021), 106711. https://doi.org/10.1016/j.knosys.2020.106711
47. A. Seyyedabbasi, F. Kiani, Sand cat swarm optimization: A nature-inspired algorithm to solve global optimization problems, *Eng. Comput.*, (2022), 1–25. https://doi.org/10.1007/s00366-022-01604-x
48. Y. J. Zhang, Y. F. Wang, Y. X. Yan, J. Zhao, Z. M. Gao, LMRAOA: An improved arithmetic optimization algorithm with multi–leader and high-speed jumping based on opposition–based learning solving engineering and numerical problems, *Alex. Eng. J.*, **61** (2022), 12367–12403. https://doi.org/10.1016/j.aej.2022.06.017
49. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
50. A. Kaveh, M. Khayatazad, A new meta-heuristic method: Ray optimization, *Comput. Struct.*, **112** (2012), 283–294. https://doi.org/10.1016/j.compstruc.2012.09.003
51. A. Faramarzi, M. Heidarinejad, S. Mirjalili, A. H. Gandomi, Marine Predators Algorithm: A Nature-inspired Metaheuristic, *Expert Syst. Appl.*, **152** (2020), 113377. https://doi.org/10.1016/j.eswa.2020.113377
52. S. Mirjalili, S. M. Mirjalili, A. Hatamlou, Multi-verse optimizer: A nature-inspired algorithm for global optimization, *Neural Comput. Appl.*, **27** (2015), 495–513. https://doi.org/10.1007/s00521-015-1870-7
53. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Gener. Comp. Syst.*, **97** (2019), 849–872. https://doi.org/10.1016/j.future.2019.02.028
54. M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, *Appl. Math. Comput.*, **188** (2007), 1567–1579. https://doi.org/10.1016/j.amc.2006.11.033
55. G. Dhiman, V. Kumar, Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications, *Adv. Eng. Softw.*, **114** (2017), 48–70. https://doi.org/10.1016/j.advengsoft.2017.05.014
56. S. Li, H. Chen, M. Wangm, A. A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Future Gener. Comput. Syst.*, **111** (2020), 300–323. https://doi.org/10.1016/j.future.2020.03.055

57. Q. He, L. Wang, A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization, *Appl. Math. Comput.*, **186** (2007), 1407–1422. https://doi.org/10.1016/j.amc.2006.07.134
58. A. Kaveh, N. Farhoudi, A new optimization method: Dolphin echolocation, *Adv. Eng. Softw.*, **59** (2013), 53–70. https://doi.org/10.1016/j.advengsoft.2013.03.004
59. I. Naruei, F. Keynia, A new optimization method based on COOT bird natural life mode, *Expert Syst. Appl.*, **183** (2021), 115352. https://doi.org/10.1016/j.eswa.2021.115352
60. W. Zhao, L. Wang, Z. Zhang, Artificial ecosystem-based optimization: A novel nature-inspired meta-heuristic algorithm, *Neural Comput. Appl.*, **32** (2020), 9383–9425. https://doi.org/10.1007/s00521-019-04452-x
61. A. Kaveh, S. Talatahari, A novel heuristic optimization method: Charged system search, *Acta Mech.*, **213** (2010), 267–289. https://doi.org/10.1007/s00707-009-0270-4
62. F. Huang, L. Wang, Q. He, An effective co-evolutionary differential evolution for constrained optimization, *Appl. Math. Comput.*, **186** (2007), 340–356. https://doi.org/10.1016/j.amc.2006.07.105
63. H. Yapici, N. Cetinkaya, A new meta-heuristic optimizer: Pathfinder algorithm, *Appl. Soft Comput.*, **78** (2019), 545–568. https://doi.org/10.1016/j.asoc.2019.03.012
64. V. K. Kamboj, A. Nandi, A. Bhadoria, S. Sehgal, An intensify harris hawks optimizer for numerical and engineering optimization problems, *Appl. Soft Comput.*, **89** (2020), 106018. https://doi.org/10.1016/j.asoc.2019.106018
65. C. Wen, H. Jia, D. Wu, H. Rao, S. Li, Q. Liu, et al., Modified remora optimization algorithm with multistrategies for global optimization problem, *Mathematics*, **10** (2022), 3604. https://doi.org/10.3390/math10193604
66. J. M. Czerniak, H. Zarzycki, D. Ewald, Aao as a new strategy in modeling and simulation of constructional problems optimization, *Simul. Model. Pract. Theory*, **76** (2017), 22–33. https://doi.org/10.1016/j.simpat.2017.04.001
67. N. B. Guedria, Improved accelerated pso algorithm for mechanical engineering optimization problems, *Appl. Soft Comput.*, **40** (2016), 455–467. https://doi.org/10.1016/j.asoc.2015.10.048
68. A. G. Hussien, M. Amin, M. Aziz, A comprehensive review of moth-flame optimisation: Variants, hybrids, and applications, *J. Exp. Theor. Artif. Intell.*, **32** (2020), 705–725. https://doi.org/10.1080/0952813X.2020.1737246
69. A. Baykasoglu, S. Akpinar, Weighted superposition attraction (wsa): A swarm intelligence algorithm for optimization problems-part2: Constrained optimization, *Appl. Soft Comput.*, **37** (2015), 396–415. https://doi.org/10.1016/j.asoc.2015.08.052
70. A. H. Gandomi, X. S. Yang, A. H. Alavi, Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems, *Eng. Comput.*, **29** (2013), 17–35. https://doi.org/10.1007/s00366-011-0241-y
71. A. E. Ezugwu, J. O. Agushaka, L. Abualigah, S. Mirjalili, A. H. Gandomi, Prairie dog optimization algorithm, *Neural Comput. Appl.*, **34** (2022), 20017–20065. https://doi.org/10.1007/s00521-022-07530-9
72. J. O. Agushaka, A. E. Ezugwu, L. Abualigah, Dwarf mongoose optimization algorithm, *Comput. Methods Appl. Mech. Eng.*, **391** (2022), 114570. https://doi.org/10.1016/j.cma.2022.114570
73. M. J. Kazemzadeh-Parsi, A modified firefly algorithm for engineering design optimization problems, *IJST-T. Mech. Eng.*, **38** (2014), 403.

74. J. O. Agushaka, A. E. Ezugwu, Advanced arithmetic optimization algorithm for solving mechanical engineering design problems, *PLoS ONE*, **16** (2021), e0255703. https://doi.org/10.1371/journal.pone.0255703
75. H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm-a novel metaheuristic optimization method for solving constrained engineering optimization problems, *Comput. Struct.*, **110** (2012), 151–166. https://doi.org/10.1016/j.compstruc.2012.07.010
76. G. I. Sayed, A. Darwish, A. E. Hassanien, A new chaotic multi-verse optimization algorithm for solving engineering optimization problems, *J. Exp. Theor. Artif. Intell.*, **30** (2018), 293–317. https://doi.org/10.1080/0952813X.2018.1430858
77. C. Yu, A. A. Heidari, H. Chen, A quantum–behaved simulated annealing algorithm–based moth–flame optimization method, *Appl. Math. Model.*, **87** (2020), 1–19. https://doi.org/10.1016/j.apm.2020.04.019