Mathematical Biosciences and Engineering

MBE, 19(4): 3994−4037.
DOI: 10.3934/mbe.2022184
Received: 28 December 2021
Revised: 22 January 2022
Accepted: 25 January 2022
Published: 14 February 2022

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

*Research article*

# An improved remora optimization algorithm with autonomous foraging mechanism for global optimization problems

**Rong Zheng[1,*], Heming Jia[1,*], Laith Abualigah[2,3], Shuang Wang[1] and Di Wu[4]**

[1] School of Information Engineering, Sanming University, Sanming 365004, China
[2] Faculty of Computer Sciences and Informatics, Amman Arab University, Amman 11953, Jordan
[3] School of Computer Science, Universiti Sains Malaysia, Penang 11800, Malaysia
[4] School of Education and Music, Sanming University, Sanming 365004, China

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

**Abstract:** The remora optimization algorithm (ROA) is a newly proposed metaheuristic algorithm for solving global optimization problems. In ROA, each search agent searches new space according to the position of host, which makes the algorithm suffer from the drawbacks of slow convergence rate, poor solution accuracy, and local optima for some optimization problems. To tackle these problems, this study proposes an improved ROA (IROA) by introducing a new mechanism named autonomous foraging mechanism (AFM), which is inspired from the fact that remora can also find food on its own. In AFM, each remora has a small chance to search food randomly or according to the current food position. Thus the AFM can effectively expand the search space and improve the accuracy of the solution. To substantiate the efficacy of the proposed IROA, twenty-three classical benchmark functions and ten latest CEC 2021 test functions with various types and dimensions were employed to test the performance of IROA. Compared with seven metaheuristic and six modified algorithms, the results of test functions show that the IROA has superior performance in solving these optimization problems. Moreover, the results of five representative engineering design optimization problems also reveal that the IROA has the capability to obtain the optimal results for real-world optimization problems. To sum up, these test results confirm the effectiveness of the proposed mechanism.

**Keywords:** remora optimization algorithm; arithmetic optimization algorithm; metaheuristic algorithm; swarm intelligence; global optimization

## 1. Introduction

In recent years, metaheuristic algorithms (MAs) have aroused extensive attention from scholars of all walks of life [1]. MAs are widely recognized when applied to solve optimization problems in various fields, such as economy [2], biology [3], and engineering [4,5]. It should be pointed out that a MA usually contains several random factors, which increase the algorithm's flexibility. The random factors can help MAs with extensive global search and local optimal avoidance. The MA obtains the estimated solution by cyclic iteration. In theory, with enough iterations, the MA can always find the theoretical solution or one very close to it. For a given number of iterations, the better the algorithm, the closer the theoretical solution. For other aspects, excellent algorithms converge faster and obtain more precise solutions, thereby saving costs.

In contrast to conventional methods, metaheuristic methods, which utilize the idea of swarm intelligence, have the merits of simplicity, practicability, and ease of use. For instance, particle swarm optimization (PSO) [6] is inspired by the predation behaviors of a flock of birds. Each bird searches the food by its own experience and current best experience. The well-known whale optimization algorithm (WOA) [7] models the foraging behavior of the humpback whales, i.e., encircling the prey and using bubble-net attacking method. Unlike WOA, Grey Wolf Optimizer (GWO) [8] mimics the cooperative predation of the grey wolves, which presents an excellent performance in some engineering design problems. A recently proposed monarch butterfly optimization (MBO) [9] is inspired by the migration features of monarch butterflies. arithmetic optimization algorithm (AOA) [10] is associated with the arithmetic operators. Slime mould algorithm (SMA) [11] is developed based on the features of oscillating patterns and searching food of the slime mold. Other MAs include ant lion optimizer (ALO) [12], Salp swarm algorithm (SSA) [13], grasshopper optimization algorithm (GOA) [14], Harris hawks optimization (HHO) [15], Marine predators algorithm (MPA) [16], Aquila optimizer (AO) [17], and recently proposed artificial gorilla troops optimizer (GTO) [18], African vultures optimization algorithm (AVOA) [19], wild horse optimizer (WHO) [20], hunger games search (HGS) [21], colony predation algorithm (CPA) [22], etc. These algorithms play an important role in solving various optimization problems, and save time, cost, etc.

Due to the "No free lunch theorem" (NFL) proposed by Wolpert et al. [23], it is always vital to develop new optimization algorithms to solve new and complex optimization problems, which motivates us to propose new optimization methods continually. The improvement of existing optimization methods is a current research hot spot. For instance, Al-qaness and Ewees et al. adopted the operators in firefly algorithm (FA) to enhance the exploitation capability of WOA and SSA for solving the unrelated parallel machine scheduling problem [24,25]. Onay modified the basic hunger games search (HGS) by applying the chaotic maps [26]. And the test results show the superiority of proposed algorithm compared to other methods. Two points are worth paying attention to: on the one hand, the improvement approaches had better not increase the computational complexity or time complexity, which will not increase the cost; on the other hand, they can significantly improve the optimization capability, such as faster convergence speed, higher accuracy, and stability. In this way, the enhanced algorithm can be applied well and has more practical value. When improving a MA, it is crucially important to balance the exploration and exploitation phases. Generally speaking, the well-designed optimization algorithm will explore the entire search space in the incipient stage and then turn to local search around the best position in the later stage.

The remora optimization algorithm (ROA) is a newly proposed metaheuristic optimization algorithm in 2021 [27], which is inspired by the parasitic feature of remora. As the remora can follow the hosts, such as giant whales or swordfish, thus ROA adopts the WOA strategy and SFO strategy extracted from the WOA and SFO [7,28] to search the food. Like other MAs, such as PSO [6], WOA [7], GWO [8], AOA [10], SMA [11], and SSA [13], ROA also has a straightforward framework with comparative optimization performance. In addition, ROA solves some typical engineering problems, such as the welded beam design problem, I-beam design problem, three-bar truss design problem, pressure vessel design problem, and rolling element bearing problem.

As mentioned before, it is necessary to modify existing optimization algorithm for better search capability and solving complex optimization problems. In practice, like other MAs, the basic ROA also has the drawbacks of slow convergence speed, low solution accuracy, and local optimum for some optimization problems. The main reason for these defects is the inflexible position updating options for search agents. Therefore, an improved ROA (IROA) with a novel autonomous foraging mechanism (AFM) is introduced in this paper. The AFM can expand the search space and boost the local exploitation capability of basic ROA by providing multiple options for position updating. The main contributions of this works are as follows:

1) In the proposed IROA, each remora first has a slight chance to search the food by itself; otherwise, parasitism is selected.

2) In the process of searching food autonomously, remora can choose to explore the food position randomly or based on the current food position (i.e., current best position). In this way, each remora will have a more diverse position updating selection and quickly obtain the food.

3) The applied AFM can effectively increase population diversity and improve the global search capability and local search capability.

To test the proposed IROA, twenty-three frequently used benchmark functions and ten very new IEEE CEC 2021 standard functions were employed, and also multiple metaheuristic algorithms and modified methods were selected for performance comparison. At last, the improved algorithm was tested by five engineering problems to validate practicality.

The rest of this paper is organized as follows: Section 2 introduces the standard ROA, whereas Section 3 describes the details of the adopted autonomous foraging mechanism (AFM) and the proposed IROA. Section 4 introduces the test results of IROA and comparative optimization algorithms on various standard benchmark functions, while Section 5 shows the comparative results of IROA and other methods in solving five engineering design problems. Finally, Section 6 gives the conclusions for this paper and some future works.

## 2. Remora optimization algorithm (ROA)

The remora optimization algorithm is introduced in this section, which is a newly proposed metaheuristic method in 2021 [27]. Like other MAs, the ROA utilizes the biological characteristics of remora to complete the optimizing process, i.e., the parasitic behavior. Remora is able to attach itself to swordfish, whales, or other animals. Through the help of the host, remora can obtain food easily. Based on this, ROA adapted part of the position updating modes of WOA and SFO to conduct the global and local search. It should be noted that ROA uses an integer argument H (0 or 1) to determine to choose WOA strategy or SFO strategy. Thus to some extent the ROA will have the advantages of both optimization algorithms when solving the optimization problems. The detailed operation mechanism of the ROA is presented in Figure 1.

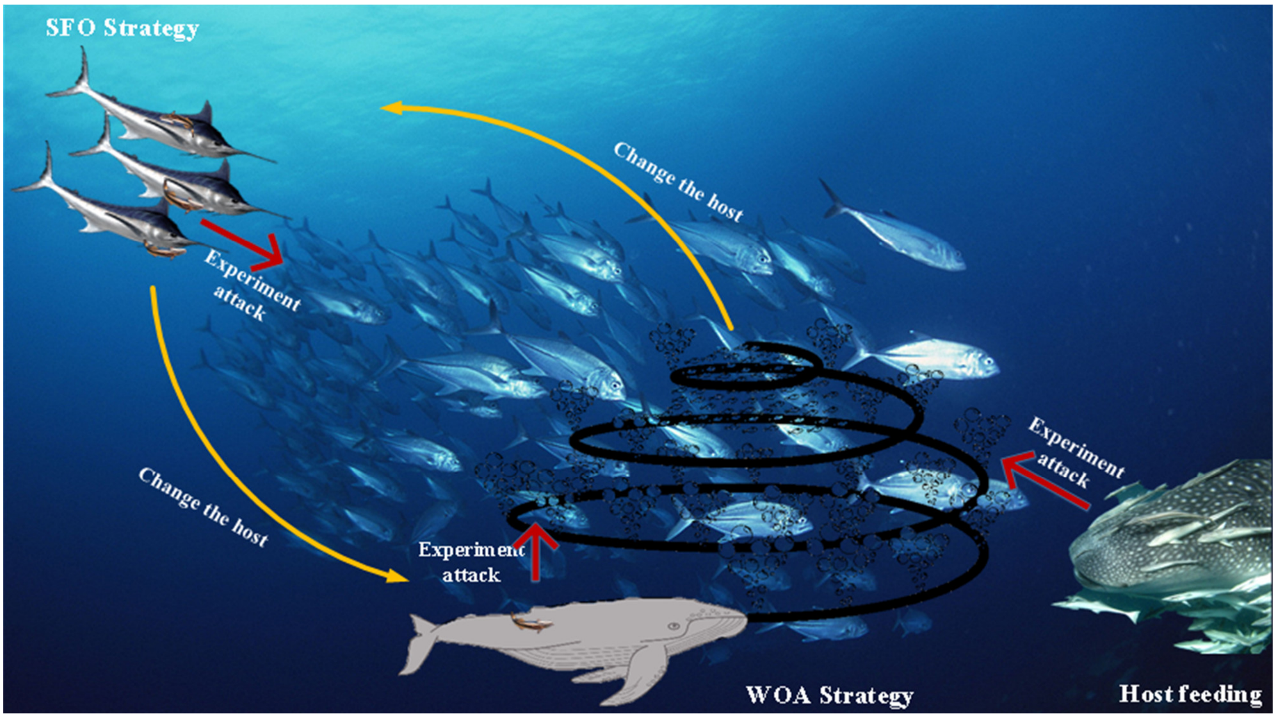


**Figure 1.** The position updating mechanism of remora in ROA.

*2.1. Free travel (Exploration)*

The ROA utilizes the SFO strategy to conduct the global search, which is based on the elite method used in the swordfish algorithm [28]. The position updating formula can be expressed as follows:

$$V_i(t+1) = X_{best}(t) - (rand \times (\frac{X_{best}(t)+X_{rand}(t)}{2}) - X_{rand}(t)) \quad (1)$$

where $V_i(t+1)$ is the candidate position of the $i$th remora. $X_{best}(t)$ is the current best position. $X_{rand}(t)$ is a random position of remora. $t$ means the current iteration number. And $rand$ is a random number between 0 and 1.

In addition, remora may change the host according to its experience. In this case, a new candidate position can be generated by:

$$V_i'(t+1) = V_i(t+1) + randn \times (V_i(t+1) - X_i(t)) \quad (2)$$

where $V_i'(t+1)$ is the candidate position of the $i$th remora. $X_i(t)$ is the previous position of the $i$th remora. And $randn$ is used to produce normally distributed random number.

*2.2. Eat thoughtfully (Exploitation)*

Remora also can attach themselves to the humpback whales for food. Hence remora will have the motion characteristics of humpback whales. The WOA strategy is employed in ROA to perform the local search [7]. To be specific, the bubble-net attacking method used in WOA is applied. The modified position updating formulas are as follows:

$$V_i(t+1) = D \times e^a \times cos(2\pi a) + X_{best}(t) \quad (3)$$

$$D = |X_{best}(t) - X_i(t)| \quad (4)$$

$$a = rand \times (b-1) + 1 \quad (5)$$

$$b = -(1 + \frac{t}{T}) \quad (6)$$

where $D$ represents the distance between remora and food. According to Eqs (5) and (6), it can be seen that $a$ is a random number within −2 and 1. And $b$ decreases linearly from −1 to −2.

**Algorithm 1** The pseudo-code of basic remora optimization algorithm

**Initialization**
  Initialize the remora population size ($N$) and maximum number of iterations ($T$)
  Initialize the positions of all search agents $X_i$ ($i$=1, 2, 3, …, $N$)
  Set the remora factor $C$
  **Main loop{**
  **While** ($t \le T$)
    Calculate the fitness of each remora
    find the best position and *bestFitness*, $X_b$
    Calculate the $a$, $b$, $A$, $B$
      **For** the $i$th remora
        **If** H(i) = 0
          Generate position $V_i$ by Eq (3)
        **Else if** H(i) = 1
          Generate position $V_i$ by Eq (1)
        **End if**
        Generate candidate position $V_i'$ by Eq (2)
        **if** $f(V_i') < f(V_i)$
          $X_i = V_i'$
          H(i) = round($rand$)
        **else**
          Update position $X_i$ by Eq (7)
        **End if**
      **End for**
      $t = t + 1$
   **End While}**
 **Return** *bestFitness*, $X_b$

Moreover, to further improve the solution quality, the remora can produce a small step by using the encircling prey mechanism in WOA, which is represented as follows:

$$X_i(t+1) = V_i(t+1) + A \times D' \quad (7)$$

$$A = 2 \times B \times rand - B \quad (8)$$

$$B = 2 \times (1 - \frac{t}{T}) \quad (9)$$

$$D' = V_i(t+1) - C \times X_{best}(t) \tag{10}$$

where $X_i(t+1)$ is the newly generated position of the $i$th remora. $C$ denotes the remora factor, which is set to 0.1 in ROA.

By conducting the above methods, ROA shows superior performance compared to WOA, SFO, HHO, EPO and other famous metaheuristic algorithms. The pseudo-code of ROA is presented in Algorithm 1.

## 3. The proposed approach

As described above, ROA is developed mainly based on the parasitic feeding on whales and swordfish. In reality, remora can also find its own food. In viewing this, a novel autonomous foraging mechanism is introduced into the basic remora optimization algorithm to enhance the search capability of remora. The improved remora optimization algorithm has a more flexible mode and strikes a good balance between exploration and exploitation. It is worth mentioning that this method will not increase the computational complexity of the original algorithm. The proposed mechanism also applies to other optimization algorithms, i.e., it has a certain universality. Details of the proposed IROA are presented below.

### *3.1. Autonomous foraging mechanism (AFM)*

We consider that remora can have two choices when finding the food by itself. The first is to find the food randomly, and the second is to obtain the food based on the current food position. Therefore, we propose the autonomous foraging mechanism to improve the basic ROA.

In the proposed AFM, two different operators are used to improve the optimization capability of ROA. First, the remora has a small chance $x$ of looking for food in other unknown locations. When $rand < x$, remora will search the whole space widely and randomly. The mathematical formula can be expressed as follows:

$$Xi(t+1) = (UB - LB) \times rand + LB \tag{11}$$

where the $UB$ and $LB$ denote the upper boundary and lower boundary of search space, respectively.

According to Eq (11), the first operator is favored to the exploration capability of the ROA, avoiding the local optimal points effectively. On the other hand, to improve the exploitation capability of ROA, the second operator is inspired by the Division ($D$) operator and Multiplication ($M$) operator in the recently proposed AOA [10]. The position updating equations are as follows:

$$Xi(t+1) = \begin{cases} Xbest(t) \div (RMOP + eps) \times ((UB - LB) \times \mu + LB) \times Levy, & rand < 0.5 \\ Xbest(t) \times RMOP \times ((UB - LB) \times \mu + LB) \times Levy, & rand \geq 0.5 \end{cases} \tag{12}$$

$$RMOP = 1 - (\frac{t}{T})^{1/\alpha} \tag{13}$$

$$\alpha = 10 \times rand - 1 \tag{14}$$

where the $RMOP$ is the random math optimizer probability, calculated using the current number of iterations, maximum number of iterations, and parameter $\alpha$. According to Eq (14), $\alpha$ is a random number between −1 and 9.

From Eq (12), the new position is generated on the basis of current best position (i.e., the position of food). And the *Levy* operator is utilized further to increase the diversity of the population [29]. Similarly, each remora will conduct the second operator when $rand < y$. Here the parameter $y$ also is a small number. The effect of both parameters $x$ and $y$ will be analyzed in Section 4.

### *3.2. The proposed IROA*

In the proposed IROA, each remora will first decide whether to find the food by itself, i.e., conduct the proposed AFM. If that is the case, after the location updating, remora also will choose a new host. If not, remora continues to engage the parasitic feeding behavior. The pseudo-code of the proposed IROA is presented in Algorithm 2. And the flowchart of the IROA is shown in Figure 2.

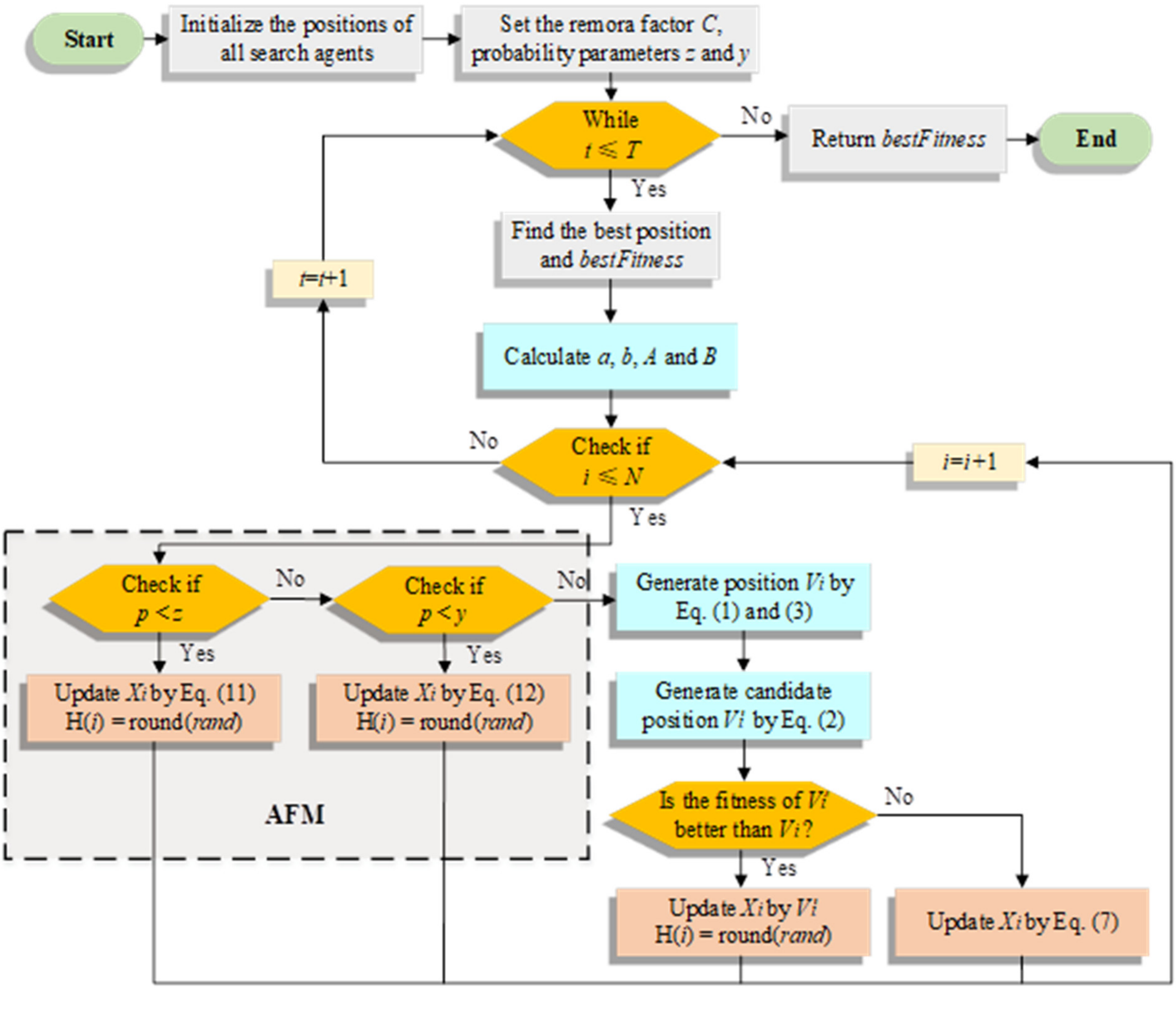


**Figure 2.** Flowchart of the proposed IROA.

**Algorithm 2** The pseudo-code of the improved remora optimization algorithm

```
Initialization
    Initialize the remora population size (N) and maximum number of iterations (T)
    Initialize the positions of all search agents X_i (i=1, 2, 3, …, N)
    Set the remora factor C, probability parameters z and y
    Main loop{
    While (t ≤ T)
        Calculate the fitness of each remora
        find the best position and bestFitness, X_b
        Calculate the a, b, A, B
          For the ith remora
            If p < z
                Update position X_i by Eq (11)
                H(i) = round(rand)
            Else if p < y
                Update position X_i by Eq (12)
                H(i) = round(rand)
            Else
                If H(i) = 0
                    Generate position V_i by Eq (3)
                Else if H(i) = 1
                    Generate position V_i by Eq (1)
                End if
                Generate candidate position V_i' by Eq (2)
                if f(V_i') < f(V_i)
                    X_i = V_i'
                    H(i) = round(rand)
                else
                    Update position X_i by Eq (7)
                End if
            End if
          End for
          t = t + 1
      End While}
  Return bestFitness, X_b
```

### *3.3. The computational complexity of IROA*

The computational complexity is an essential factor for the optimization algorithm related to the initialization, fitness evaluation, and position updating method. For the basic ROA, the computational complexity of initialization is $O(N)$. Here the parameter $N$ represents the number of search agents. And then, during the whole iterative process, the computational complexity of applying SFO strategy or WOA strategy is $O(N \times D \times T)$, where $T$ is the maximum number of iterations and $D$ is the dimensions of search space. Moreover, the experience attack's computational complexity is also $O(N \times D \times T)$.

Consider the worst-case scenario, and host feeding is needed to be performed. Thus the additional computational complexity is $O(N \times D \times T)$. Therefore, the overall computational complexity of ROA is $O(N \times (3TD + 1))$.

In the IROA, it should be noted that the proposed autonomous foraging mechanism is an optional position updating mode for each remora. When the remora looks for food randomly, the computational complexity is $O(N \times D \times T)$. When the remora decides to look for food based on the best position, the computational complexity is $O(N \times D \times T)$. Beyond that, the computational complexity of basic ROA's position updating formulas also is $O(3 \times N \times D \times T)$ in the worst condition. To sum up, the computational complexity of IROA is $O(N\times (3TD + 1))$, which is the same as that of basic ROA.

## 4. Experimental tests and analysis

### *4.1. Standard test functions and parameter settings*

To evaluate the effectiveness of the proposed IROA for solving optimization problems, two sets of test functions are utilized for the experimental tests [30,31], i.e., classical benchmark functions and IEEE CEC 2021 standard test functions. Table 1 lists the detailed information of these test functions. It is known that F1−F7 are the unimodal test functions that have only one extreme point. Oppositely, F8−F23 belongs to the multimodal test functions, containing multiple local optimal points. Hence, F1−F7 are commonly used to evaluate the exploitation capability of optimization methods, while F14−F23 are usually used to test the stability of optimization algorithms. It is noted that the dimension of test functions F1−F13 can be adjusted according to demands, which allows us to analyze the characteristics of the optimization algorithms in high-dimensional cases. Hence we investigate the performance of the optimization algorithms in four different dimensions ($D$ = 30/100/500/1000). Furthermore, the latest CEC 2021 test functions are used to evaluate the improvement effects of the optimization algorithm, which have four types of test functions, i.e., unimodal test functions, basic test functions, hybrid test functions, and composition test functions [32].

Simultaneously, the basic ROA and the other six metaheuristic optimization algorithms (including particle swarm optimization algorithm (PSO) [6], whale optimization algorithm (WOA) [7], slap swarm algorithm (SSA) [13], slime mould algorithm (SMA) [11], arithmetic optimization algorithm (AOA) [10], and marine predators algorithm (MPA) [16]) and six modified optimization algorithms (including HHOCM [33], SWGWO [34], ROLGWO [35], DSCA [36], MALO [37] and HSCAHS [38]) are employed for extensive comparison. The ROA, SMA, AOA, and MPA are the newly proposed algorithms, while SSA, WOA, and PSO are very famous optimization algorithms, which have been widely studied. The parameter details of the IROA and comparative algorithms are listed in Table 2. For the modified algorithms, the parameter settings are the same as those in the native works.

The number of iterations and population size of each algorithm are set to 500 and 30 for a proper comparison. To obtain convincing results, each algorithm was independently tested 30 times. The experimental results are analyzed mainly from two aspects. For one thing, the numerical results of these simulation experiments are analyzed according to the mean value and standard deviation. Also, two widely used statistical methods, i.e., the Wilcoxon signed-rank test [39] and Friedman ranking test [40], are employed to reveal the significant differences between the IROA and other comparative algorithms. For another, the convergence curves of these algorithms are also used to give a visual display of the optimization searching process. The experimental analyses of the IROA are presented as follows section.

**Table 1.** Feature properties of the test functions (*D* indicates the dimension).

| Function type | Function | Dimensions | Range | Theoretical optimization value |
|---|---|---|---|---|
| Unimodal test functions | F1 | 30/100/500/1000 | [−100, 100] | 0 |
| | F2 | 30/100/500/1000 | [−10, 10] | 0 |
| | F3 | 30/100/500/1000 | [−100, 100] | 0 |
| | F4 | 30/100/500/1000 | [−100, 100] | 0 |
| | F5 | 30/100/500/1000 | [−30, 30] | 0 |
| | F6 | 30/100/500/1000 | [−100, 100] | 0 |
| | F7 | 30/100/500/1000 | [−1.28, 1.28] | 0 |
| Multimodal test functions | F8 | 30/100/500/1000 | [−500, 500] | $-418.9829 \times D$ |
| | F9 | 30/100/500/1000 | [−5.12, 5.12] | 0 |
| | F10 | 30/100/500/1000 | [−32, 32] | 0 |
| | F11 | 30/100/500/1000 | [−600, 600] | 0 |
| | F12 | 30/100/500/1000 | [−50, 50] | 0 |
| | F13 | 30/100/500/1000 | [−50, 50] | 0 |
| Fixed-dimension multimodal test functions | F14 | 2 | [−65, 65] | 0.998004 |
| | F15 | 4 | [−5, 5] | 0.0003075 |
| | F16 | 2 | [−5, 5] | −1.03163 |
| | F17 | 2 | [−5, 5] | 0.398 |
| | F18 | 2 | [−2, 2] | 3 |
| | F19 | 3 | [−1, 2] | −3.8628 |
| | F20 | 6 | [0, 1] | −3.3220 |
| | F21 | 4 | [0, 10] | −10.1532 |
| | F22 | 4 | [0, 10] | −10.4028 |
| | F23 | 4 | [0, 10] | −10.5363 |
| CEC 2021 unimodal test functions | CEC_01 | 10 | [−100, 100] | 100 |
| CEC 2021 basic test functions | CEC_02 | 10 | [−100, 100] | 1100 |
| | CEC_03 | 10 | [−100, 100] | 700 |
| | CEC_04 | 10 | [−100, 100] | 1900 |
| CEC 2021 hybrid test functions | CEC_05 | 10 | [−100, 100] | 1700 |
| | CEC_06 | 10 | [−100, 100] | 1600 |
| | CEC_07 | 10 | [−100, 100] | 2100 |
| CEC 2021 composition test functions | CEC_08 | 10 | [−100, 100] | 2200 |
| | CEC_09 | 10 | [−100, 100] | 2400 |
| | CEC_10 | 10 | [−100, 100] | 2500 |

**Table 2.** Parameter values for the IROA and other comparative optimization algorithms.

| Algorithm | Parameters |
|---|---|
| IROA | $C = 0.1$; $\alpha \in [-1, 9]$; $\mu = 0.499$; $z = 0.07$; $y = 0.1$ |
| ROA [27] | $C = 0.1$ |
| SMA [11] | $z = 0.03$ |
| AOA [10] | $\alpha = 5$; $\mu = 0.499$; $Min = 0.2$; $Max = 0.9$ |
| MPA [16] | FADs = 0.2; $P = 0.5$; CF = [1, 0] |
| SSA [13] | $c_1 \in [0, 1]$; $c_2 \in [0, 1]$ |
| WOA [7] | $a_1 = [2, 0]$; $a_2 = [-2, -1]$; $b = 1$ |
| PSO [6] | $c_1 = 2$; $c_2 = 2$; $W \in [0.2, 0.9]$; $vMax = 6$ |
| HHOCM [33] | The absolute value of escaping energy decreases from 2 to 0, mutation rate decreases linearly from 1 to 0. |
| SWGWO [34] | $A = 2$; $v_{alpha} \in [-2, 0]$; $v_{beta} \in [0, 1]$; $v_{delta} \in [0, 0.5]$; |
| ROLGWO [35] | $r_3 \in [0, 1]$ |
| DSCA [36] | $w \in [0.1, 0.9]$, $\sigma = 0.1$; $a_{end} = 0$; $a_{start} = 2$ |
| MALO [37] | *Switch possibility* = 0.5 |
| HSCAHS [38] | $a = 2$; *Bandwidth* = 0.02 |

### *4.2. Sensitivity analysis of z and y on IROA*

The performance of the proposed IROA is apparently related to the parameters $z$ and $y$. Thus, obtaining the proper values for better optimization capability is necessary. Referring to remora's actual situation, the $z$ and $y$ should be comparatively small. In this paper, $z$ is considered as 0.03, 0.05 and 0.07, while $y$ can be 0.1, 0.2 and 0.3. Hence there are nine cases for the proposed IROA. To verify the effects of these two parameters, twenty-three test functions are used (the dimensions of F1−F13 are 30). The results of the mean-square error for each function are listed in Table 3. The lowest values are highlighted in bold. It can be clearly observed that the IROA with $z = 0.07$ and $y = 0.7$ outperforms other cases and achieves fourteen best results out of twenty-three test functions. In viewing this, this version of the improved algorithm will be used for further experimental analysis.

**Table 3.** Sensitivity analysis on the IROA's parameters (F1−F23).

| Function | $z$ = 0.03 $y$ = 0.1 | $z$ = 0.03 $y$ = 0.2 | $z$ = 0.03 $y$ = 0.3 | $z$ = 0.05 $y$ = 0.1 | $z$ = 0.05 $y$ = 0.2 | $z$ = 0.05 $y$ = 0.3 | $z$ = 0.07 $y$ = 0.1 | $z$ = 0.07 $y$ = 0.2 | $z$ = 0.07 $y$ = 0.3 |
|---|---|---|---|---|---|---|---|---|---|
| F1 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 |
| F2 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 |
| F3 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 |
| F4 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 |
| F5 | 2.10E−01 | 5.00E+01 | 1.01E+00 | 7.35E−02 | 4.83E−02 | 2.34E+01 | **1.28E−02** | 1.29E−02 | 1.33E−02 |
| F6 | 2.73E−05 | 7.55E−05 | 1.51E−04 | 1.29E−05 | 3.79E−05 | 4.63E−05 | **2.15E−06** | 4.90E−06 | 6.27E−05 |
| F7 | 4.82E−08 | **1.43E−08** | 1.78E−08 | 4.48E−08 | 1.92E−08 | 1.52E−08 | 2.36E−08 | 2.08E−08 | 1.99E−08 |
| F8 | 3.87E−01 | 2.14E+00 | 3.77E−01 | 3.80E−02 | 3.53E−02 | 3.47E−02 | **6.91E−03** | 8.27E−02 | 4.51E−02 |
| F9 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 |
| F10 | 7.89E−31 | 7.89E−31 | 7.89E−31 | 7.89E−31 | 7.89E−31 | 7.89E−31 | **7.89E−31** | 7.89E−31 | 7.89E−31 |
| F11 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | 0 | 0 |
| F12 | 1.53E−07 | 8.13E−08 | 1.51E−07 | 5.71E−08 | 1.96E−08 | 7.56E−08 | **1.37E−08** | 1.79E−08 | 1.73E−08 |
| F13 | 1.41E−05 | 3.12E−05 | 1.06E−04 | 1.59E−05 | 1.83E−05 | 5.99E−05 | **9.76E−07** | 3.14E−06 | 9.39E−06 |
| F14 | 2.63E−14 | 2.62E−14 | **2.56E−14** | 2.62E−14 | 2.62E−14 | 2.62E−14 | 2.62E−14 | 2.62E−14 | 2.59E−14 |
| F15 | 6.73E−08 | 5.89E−08 | 3.22E−08 | 3.93E−08 | 2.42E−08 | 3.23E−08 | 5.66E−08 | **3.17E−09** | 9.62E−09 |
| F16 | **2.66E−12** | 2.78E−12 | 3.00E−12 | 2.71E−12 | 2.95E−12 | 2.90E−12 | 2.84E−12 | 3.20E−12 | 4.06E−12 |
| F17 | 1.19E−08 | 1.18E−08 | 1.10E−08 | 1.06E−08 | **1.02E−08** | 1.08E−08 | 1.14E−08 | 1.17E−08 | 1.09E−08 |
| F18 | 9.66E−10 | 5.87E−09 | 2.15E−09 | **8.14E−11** | 7.98E−09 | 2.47E−09 | 1.15E−09 | 3.87E−09 | 2.81E−09 |
| F19 | 1.52E−06 | 9.88E−07 | 2.79E−06 | 1.34E−06 | 7.43E−07 | 1.93E−06 | **1.36E−07** | 2.70E−06 | 1.03E−06 |
| F20 | 1.26E−02 | 8.73E−03 | 1.43E−02 | 1.69E−02 | 9.40E−03 | 1.18E−02 | 9.03E−03 | **7.59E−03** | 1.04E−02 |
| F21 | 7.26E−05 | 1.71E−05 | 1.20E−04 | 9.42E−06 | 4.88E−05 | 1.02E−05 | 1.86E−05 | 1.75E−05 | **6.20E−06** |
| F22 | 7.97E−06 | 1.18E−05 | 2.98E−05 | 3.55E−06 | 9.38E−05 | 1.58E−05 | 1.80E−06 | 8.34E−06 | **5.46E−07** |
| F23 | 8.31E−05 | 9.51E−05 | 7.48E−05 | 1.42E−05 | 1.33E−06 | 3.49E−05 | **6.97E−07** | 1.73E−05 | 1.08E−06 |

### *4.3. Comparison with metaheuristic algorithms*

#### 4.3.1 Numerical analysis

To verify the effectiveness of proposed mechanism applied in the basic ROA, extensive comparative analysis between IROA and other seven metaheuristic algorithms (including ROA, SMA, AOA, MPA, SSA, WOA, PSO, etc.) has been carried out. The numerical results of thirteen test functions in different dimensions ($D$ = 30/100/500/1000) for these optimization methods are listed in Tables 4–5. And Table 6 shows the test results of ten fixed-dimension test functions. Note that the ranking results based on the Friedman ranking test are also presented at the end of each table.

As shown in Tables 4–6, the basic ROA is weak in solving some test functions (such as F2–F6, F12 and F13). However, the IROA has obtained better results on these functions, even in different dimensions. Compared to other metaheuristic algorithms, the ranking results in different dimensions indicate that the IROA is the best optimization method. Thus, the proposed IROA with an autonomous foraging mechanism has excellent exploration and exploitation capability. For the results of fixed-dimension problems listed in Table 6, the IROA is ranked second, while the MPA is the best. However, it is worth mentioning that the distance of mean rank values between IROA and MPA is very small. Therefore, the performance of IROA is still the best overall.

The p-values results of the Wilcoxon signed-rank test are reported in Table 7. The significance level between the two algorithms is set to 0.05. In Table 7, the symbol "+/=/－" means the IROA performs better, similar, and worse than the comparative algorithm, respectively. From Table 7, it can be observed that the IROA has better performance than these algorithms. In particular, IROA did not lose once compared to SMA, SSA, WOA, and PSO. IROA also presents better results than SSA and PSO on most of the test functions with both outcomes of 58/4/0, which means that IROA is 58 times better than the comparison algorithms, similar at four times, and none worse respectively. As ROA and SMA are very efficient optimization methods, the IROA obtains similar results 24 and 22 times, respectively. Overall, it can be concluded that IROA is better than other optimization approaches.

Table 8 shows the proposed IROA and ROA results on CEC 2021 test functions. In the same way, 30 times tests are performed to obtain the statistical results. Moreover, the best results of all tests for these test functions are also given in Table 8. By contrast, it is observed that IROA has won all the test functions. Although the performance gap between the original algorithm and the improved version is not significant, the listed results still demonstrate the effectiveness of the proposed approach. Therefore, the basic ROA has been intensified with the help of an autonomous foraging mechanism when solving the CEC 2021 standard functions.

#### 4.3.2 Convergence analysis

Due to the strategies of WOA and SFO, the basic ROA suffers from a slow convergence speed. Hence, the AFM is introduced into the ROA to modify this problem while also improving the algorithm's exploration capability. In the proposed AFM, the strategy derived from AOA can effectively enhance the convergence speed and solution accuracy, whereas the behavior of finding food freely can prevent the algorithm from stagnating. Figures 4−7 exhibit the convergence features of the IROA and other comparative methods on test functions F1–F5, F7, F8, F10, and F13 in different dimensions ($D$ = 30/100/500/1000), whereas Figure 8 displays the convergence features of these optimization methods on test functions F14, F15, F17–F23.

From Figures 4−7, in contrast to the basic ROA and other algorithms, the proposed IROA has achieved the fastest convergence rate and obtained the best results in the end on most of the test functions. It should be noted that IROA also presents excellent stability even though in very high dimensions. However, in any case, the performance of the IROA is still limited, and the theoretical optimal values of some test functions cannot be obtained, such as F5, F7, F13. From Figure 8, it can be seen that the IROA converges quickly and gets close to the theoretical optimal values. The advantages are obvious compared with other competitive optimizers. To sum up, the convergence characteristics in Figures 4−8 fully illustrate the effectiveness of the introduced mechanism, which makes the IROA strikes a good balance between exploration and exploitation processes.

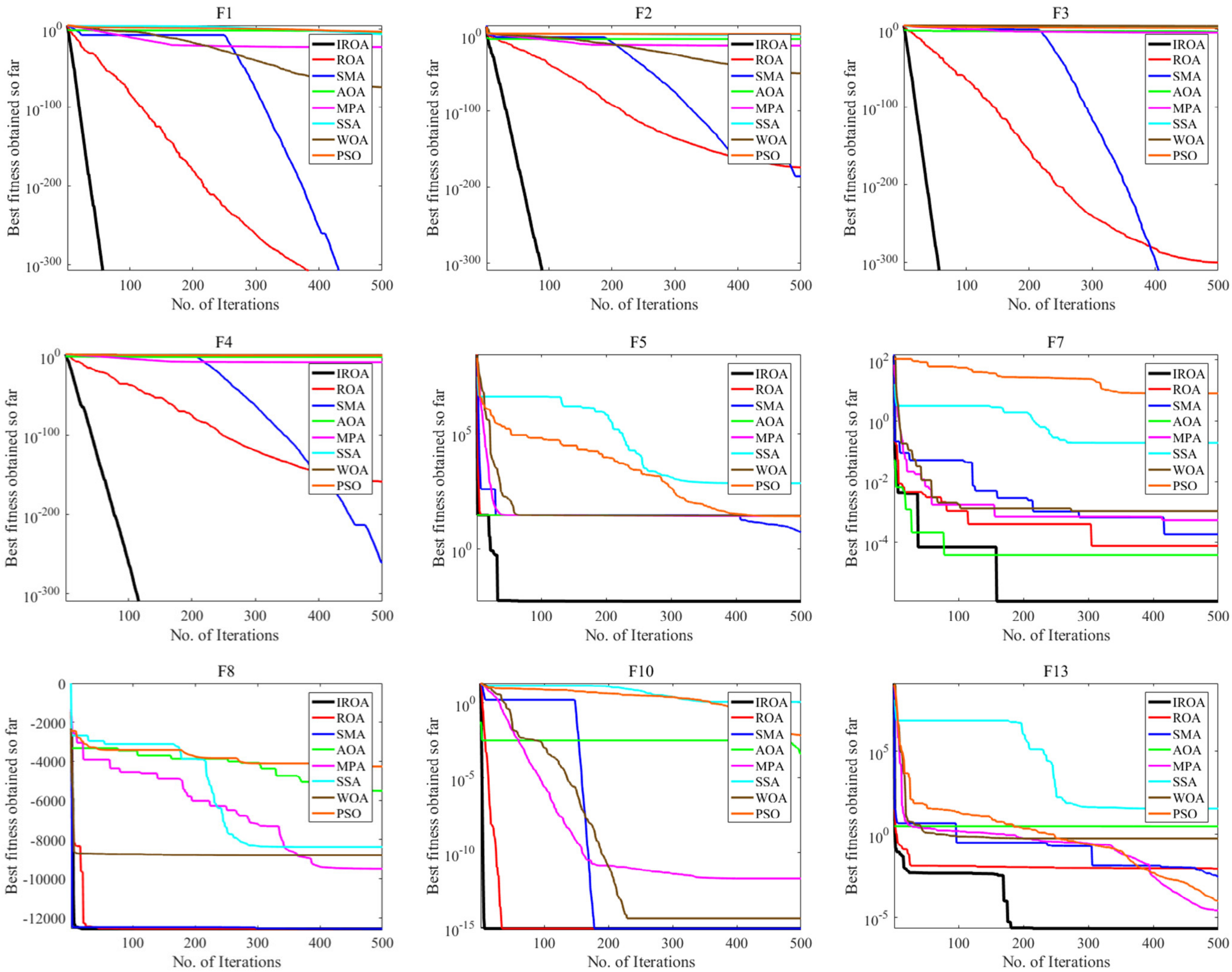


**Figure 4.** The convergence curves for the optimization algorithms on test functions (F1–F5, F7, F8, F10, F13) with $D = 30$.

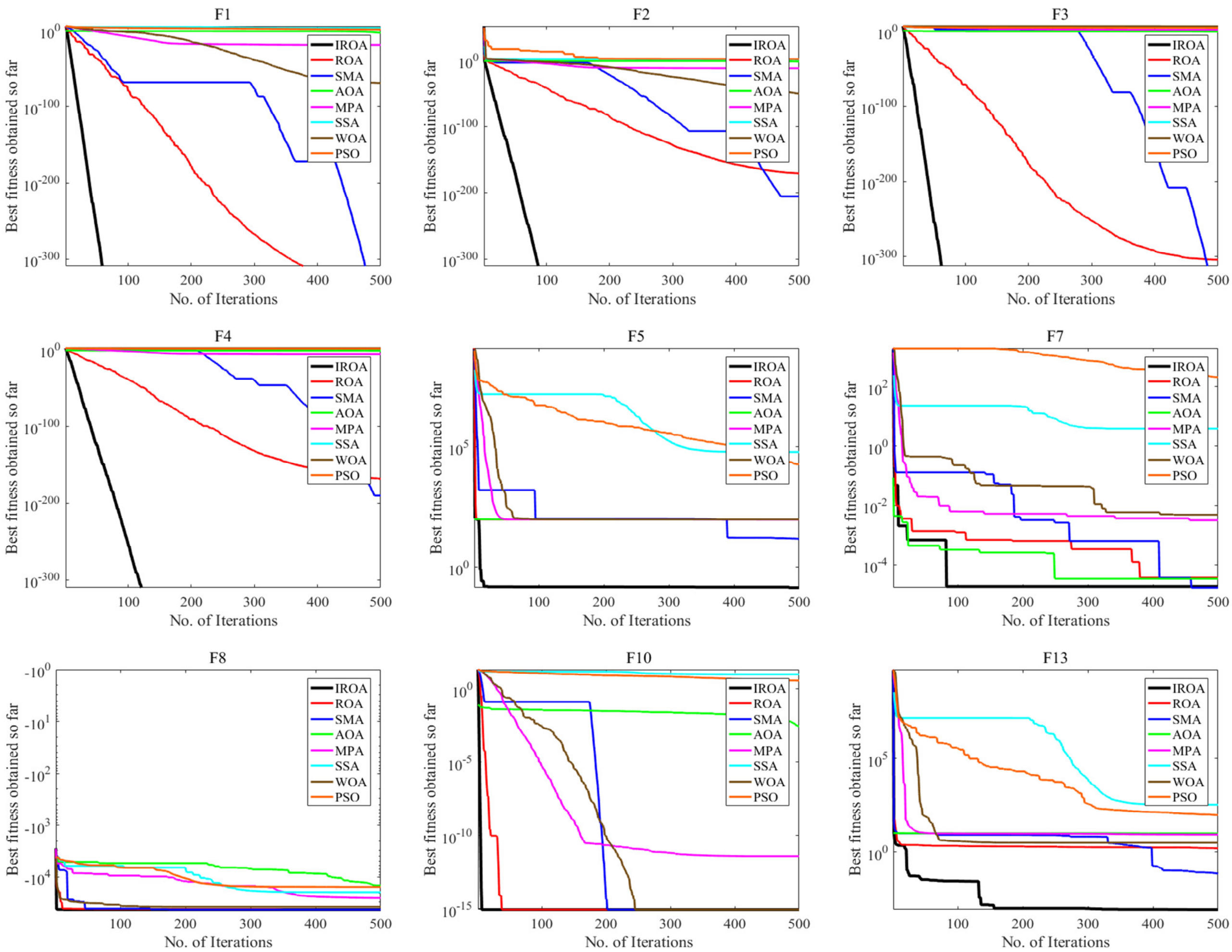


**Figure 5.** The convergence curves for the optimization algorithms on test functions (F1–F5, F7, F8, F10, F13) with $D = 100$.

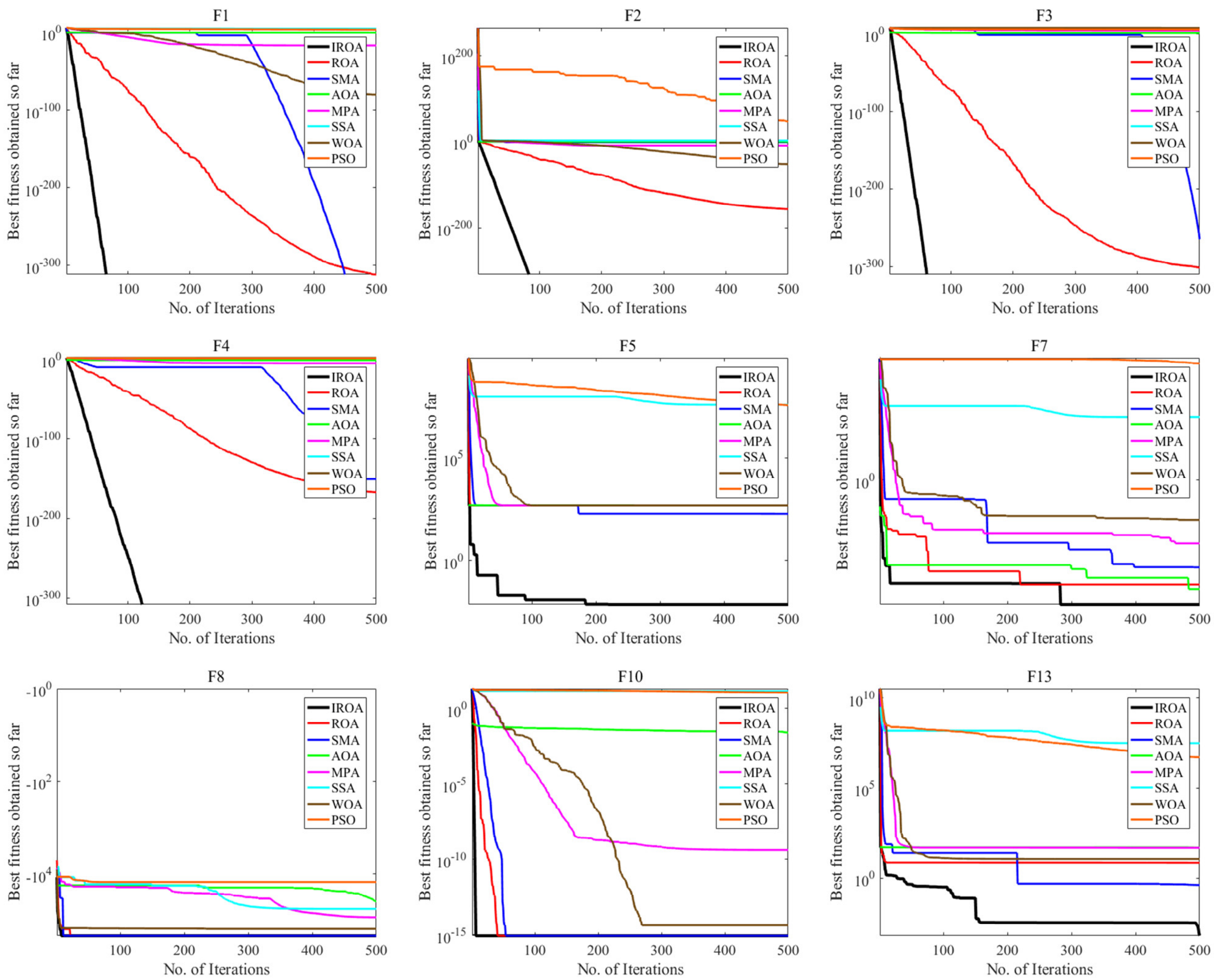


**Figure 6.** The convergence curves for the optimization algorithms on test functions (F1–F5, F7, F8, F10, F13) with $D$ = 500.

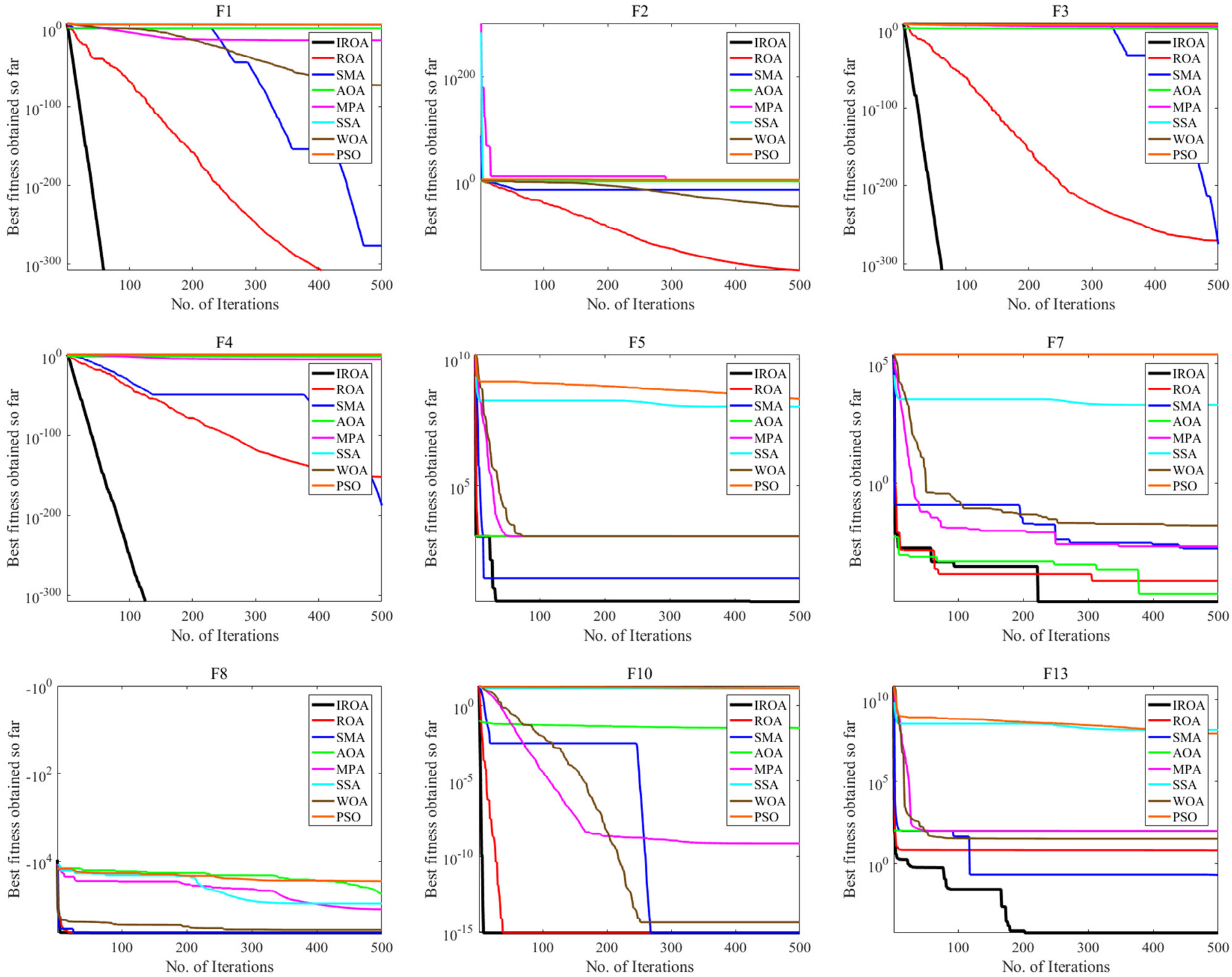


**Figure 7.** The convergence curves for the optimization algorithms on test functions (F1–F5, F7, F8, F10, F13) with $D = 1000$.

**Table 4.** Results of the IROA and other metaheuristic algorithms on unimodal test functions (F1−F7) in different dimensions.

| Function | *D* | Metric | IROA | ROA | SMA | AOA | MPA | SSA | WOA | PSO |
|---|---|---|---|---|---|---|---|---|---|---|
| F1 | 30 | Mean | 0 | 1.9159E−312 | 1.51E−283 | 4.61E−06 | 4.33E−23 | 2.18E−07 | 1.26E−70 | 4.71E−04 |
| | | Std | 0 | 0 | 0 | 2.01E−06 | 5.92E−23 | 5.13E−07 | 6.90E−70 | 1.34E−03 |
| | 100 | Mean | 0 | 4.6714E−320 | 2.95E−303 | 9.89E−04 | 2.49E−19 | 1.35E+03 | 4.80E−72 | 2.03E+01 |
| | | Std | 0 | 0 | 0 | 2.36E−04 | 2.06E−19 | 4.47E+02 | 1.76E−71 | 4.97E+00 |
| | 500 | Mean | 0 | 1.4526E−316 | 3.36E−238 | 5.38E−01 | 6.73E−17 | 9.45E+04 | 7.21E−70 | 5.85E+03 |
| | | Std | 0 | 0 | 0 | 3.03E−02 | 4.63E−17 | 7.15E+03 | 3.32E−69 | 3.98E+02 |
| | 1000 | Mean | 0 | 2.0172E−312 | 1.23E−209 | 1.49E+00 | 6.51E−16 | 2.33E+05 | 1.77E−66 | 4.12E+04 |
| | | Std | 0 | 0 | 0 | 3.42E−02 | 5.84E−16 | 1.20E+04 | 9.71E−66 | 1.63E+03 |
| F2 | 30 | Mean | 0 | 2.49E−158 | 4.71E−153 | 2.24E−03 | 3.34E−13 | 1.86E+00 | 8.27E−52 | 6.36E+00 |
| | | Std | 0 | 1.36E−157 | 2.58E−152 | 1.95E−03 | 2.65E−13 | 1.13E+00 | 2.05E−51 | 8.08E+00 |
| | 100 | Mean | 0 | 1.82E−164 | 3.24E−132 | 1.83E−02 | 1.57E−11 | 4.58E+01 | 5.77E−50 | 1.34E+02 |
| | | Std | 0 | 0 | 1.77E−131 | 2.15E−03 | 1.75E−11 | 5.90E+00 | 2.54E−49 | 2.53E+01 |
| | 500 | Mean | 0 | 2.07E−153 | 3.31E−01 | 5.58E−01 | 5.89E−10 | 5.33E+02 | 7.53E−49 | 5.15E+49 |
| | | Std | 0 | 1.13E−152 | 8.90E−01 | 7.69E−02 | 7.32E−10 | 2.20E+01 | 3.75E−48 | 2.22E+50 |
| | 1000 | Mean | 0 | 4.12E−158 | 1.23E−01 | 1.62E+00 | 1.19E+03 | 1.19E+03 | 6.66E−48 | 1.41E+03 |
| | | Std | 0 | 2.22E−157 | 3.84E−01 | 6.90E−02 | 1.87E+02 | 2.47E+01 | 2.24E−47 | 6.82E+01 |
| F3 | 30 | Mean | 0 | 8.34E−281 | 1.18E−270 | 1.03E−03 | 2.76E−04 | 1.61E+03 | 4.02E+04 | 9.54E+01 |
| | | Std | 0 | 0 | 0 | 8.34E−04 | 4.56E−04 | 1.17E+03 | 1.11E+04 | 3.97E+01 |
| | 100 | Mean | 0 | 5.64E−278 | 2.39E−219 | 1.32E−01 | 8.30E+00 | 4.89E+04 | 9.90E+05 | 1.60E+04 |
| | | Std | 0 | 0 | 0 | 3.71E−02 | 1.11E+01 | 2.07E+04 | 2.69E+05 | 3.06E+03 |
| | 500 | Mean | 0 | 6.92E−251 | 1.33E−187 | 6.85E+00 | 4.98E+03 | 1.44E+06 | 2.69E+07 | 5.37E+05 |
| | | Std | 0 | 0 | 0 | 1.30E+00 | 3.16E+03 | 7.40E+05 | 1.09E+07 | 9.51E+04 |
| | 1000 | Mean | 0 | 1.38E−251 | 7.34E−112 | 3.29E+01 | 3.44E+04 | 5.45E+06 | 1.25E+08 | 2.26E+06 |
| | | Std | 0 | 0 | 4.02E−111 | 5.83E+00 | 2.02E+04 | 2.31E+06 | 3.73E+07 | 4.98E+05 |

*Continued on next page*

| Function | *D* | Metric | IROA | ROA | SMA | AOA | MPA | SSA | WOA | PSO |
|---|---|---|---|---|---|---|---|---|---|---|
| F4 | 30 | Mean | 0 | 9.53E−158 | 5.61E−156 | 1.72E−02 | 3.48E−09 | 1.14E+01 | 4.40E+01 | 1.11E+00 |
| | | Std | 0 | 5.15E−157 | 3.07E−155 | 1.17E−02 | 2.16E−09 | 3.72E+00 | 2.72E+01 | 2.86E−01 |
| | 100 | Mean | 0 | 1.93E−155 | 1.62E−118 | 5.62E−02 | 2.27E−07 | 2.84E+01 | 7.89E+01 | 1.20E+01 |
| | | Std | 0 | 1.04E−154 | 8.89E−118 | 6.90E−03 | 1.09E−07 | 3.71E+00 | 1.84E+01 | 1.70E+00 |
| | 500 | Mean | 0 | 6.08E−156 | 3.99E−85 | 1.20E−01 | 2.19E−05 | 4.14E+01 | 8.04E+01 | 2.79E+01 |
| | | Std | 0 | 3.08E−155 | 2.19E−84 | 9.26E−03 | 1.83E−05 | 2.32E+00 | 2.11E+01 | 1.68E+00 |
| | 1000 | Mean | 0 | 4.31E−156 | 3.53E−80 | 1.53E−01 | 2.25E−04 | 4.50E+01 | 8.19E+01 | 3.34E+01 |
| | | Std | 0 | 2.36E−155 | 1.94E−79 | 9.80E−03 | 1.41E−04 | 3.80E+00 | 2.28E+01 | 1.53E+00 |
| F5 | 30 | Mean | 2.42E−02 | 2.60E+01 | 8.12E+00 | 2.80E+01 | 2.52E+01 | 2.48E+02 | 2.79E+01 | 7.21E+01 |
| | | Std | 2.88E−02 | 4.85E+00 | 1.21E+01 | 2.87E−01 | 4.51E−01 | 4.78E+02 | 4.87E−01 | 5.82E+01 |
| | 100 | Mean | 1.66E−01 | 9.76E+01 | 3.69E+01 | 9.82E+01 | 9.68E+01 | 1.45E+05 | 9.82E+01 | 1.45E+04 |
| | | Std | 2.86E−01 | 4.37E−01 | 3.88E+01 | 6.97E−02 | 7.06E−01 | 6.00E+04 | 1.83E−01 | 4.90E+03 |
| | 500 | Mean | 1.27E+00 | 4.88E+02 | 1.97E+02 | 5.00E+02 | 4.97E+02 | 3.66E+07 | 4.96E+02 | 3.02E+07 |
| | | Std | 2.06E+00 | 3.37E+01 | 2.18E+02 | 2.36E−01 | 3.35E−01 | 4.82E+06 | 5.62E−01 | 4.42E+06 |
| | 1000 | Mean | 5.21E+00 | 9.90E+02 | 4.30E+02 | 1.00E+03 | 9.97E+02 | 1.21E+08 | 9.94E+02 | 2.79E+08 |
| | | Std | 1.14E+01 | 5.38E−01 | 4.23E+02 | 2.62E−01 | 2.59E−01 | 1.38E+07 | 7.90E−01 | 2.90E+07 |
| F6 | 30 | Mean | 1.24E−04 | 9.87E−02 | 5.47E−03 | 3.08E+00 | 4.13E−08 | 2.58E−07 | 3.45E−01 | 2.06E−04 |
| | | Std | 1.06E−04 | 1.23E−01 | 2.56E−03 | 2.40E−01 | 1.96E−08 | 6.59E−07 | 1.65E−01 | 2.59E−04 |
| | 100 | Mean | 4.25E−03 | 1.82E+00 | 1.00E+00 | 1.57E+01 | 3.63E+00 | 1.34E+03 | 3.86E+00 | 1.91E+01 |
| | | Std | 7.18E−03 | 6.42E−01 | 1.23E+00 | 7.94E−01 | 7.20E−01 | 3.40E+02 | 1.35E+00 | 4.66E+00 |
| | 500 | Mean | 3.34E−02 | 1.58E+01 | 1.94E+01 | 1.13E+02 | 7.71E+01 | 9.47E+04 | 2.86E+01 | 5.82E+03 |
| | | Std | 4.48E−02 | 6.51E+00 | 3.10E+01 | 1.65E+00 | 1.83E+00 | 5.39E+03 | 7.67E+00 | 4.88E+02 |
| | 1000 | Mean | 9.02E−02 | 2.73E+01 | 5.15E+01 | 2.42E+02 | 1.89E+02 | 2.33E+05 | 7.10E+01 | 4.14E+04 |
| | | Std | 1.40E−01 | 1.30E+01 | 7.52E+01 | 1.08E+00 | 3.05E+00 | 8.90E+03 | 1.77E+01 | 1.97E+03 |
| F7 | 30 | Mean | 8.27E−05 | 1.54E−04 | 1.51E−04 | 5.07E−05 | 1.45E−03 | 1.88E−01 | 3.57E−03 | 5.70E+00 |
| | | Std | 7.27E−05 | 1.29E−04 | 1.50E−04 | 4.80E−05 | 8.20E−04 | 8.18E−02 | 4.15E−03 | 7.80E+00 |

*Continued on next page*

| Function | *D* | Metric | IROA | ROA | SMA | AOA | MPA | SSA | WOA | PSO |
|---|---|---|---|---|---|---|---|---|---|---|
| F7 | 100 | Mean | 1.46E−04 | 1.26E−04 | 3.38E−04 | 5.22E−05 | 1.75E−03 | 2.71E+00 | 3.65E−03 | 2.52E+02 |
| | | Std | 1.35E−04 | 1.28E−04 | 2.48E−04 | 4.57E−05 | 7.71E−04 | 6.06E−01 | 4.44E−03 | 1.01E+02 |
| | 500 | Mean | 1.08E−04 | 1.38E−04 | 7.19E−04 | 5.56E−05 | 2.08E−03 | 2.72E+02 | 3.65E−03 | 4.48E+04 |
| | | Std | 9.44E−05 | 1.28E−04 | 4.85E−04 | 8.56E−05 | 8.49E−04 | 5.50E+01 | 4.19E−03 | 7.28E+03 |
| | 1000 | Mean | 1.66E−04 | 2.18E−04 | 9.51E−04 | 8.67E−05 | 2.83E−03 | 1.65E+03 | 4.68E−03 | 2.42E+05 |
| | | Std | 1.77E−04 | 2.52E−04 | 8.16E−04 | 8.12E−05 | 1.61E−03 | 1.61E+02 | 4.54E−03 | 7.12E+03 |
| Rank | 30 | Mean | **1.43** | 3.14 | 3.14 | 5.43 | 3.86 | 6.29 | 6.00 | 6.71 |
| | | Overall | **1** | 2.5 | 2.5 | 5 | 4 | 7 | 6 | 8 |
| | 100 | Mean | **1.29** | 2.43 | 2.86 | 4.79 | 4.43 | 7.43 | 5.79 | 7.00 |
| | | Overall | **1** | 2 | 3 | 5 | 4 | 8 | 6 | 7 |
| | 500 | Mean | **1.14** | 2.29 | 3.29 | 4.86 | 4.71 | 7.43 | 5.29 | 7.00 |
| | | Overall | **1** | 2 | 3 | 5 | 4 | 8 | 6 | 7 |
| | 1000 | Mean | **1.14** | 2.29 | 3.14 | 4.71 | 5.07 | 7.21 | 5.29 | 7.14 |
| | | Overall | **1** | 2 | 3 | 4 | 5 | 8 | 6 | 7 |

**Table 5.** Results of the IROA and other metaheuristic algorithms on multimodal test functions (F8−F13) in different dimensions.

| Function | *D* | Metric | IROA | ROA | SMA | AOA | MPA | SSA | WOA | PSO |
|---|---|---|---|---|---|---|---|---|---|---|
| F8 | 30 | Mean | −1.26E+04 | −1.23E+04 | −1.26E+04 | −5.49E+03 | −8.83E+03 | −7.39E+03 | −1.01E+04 | −5.46E+03 |
| | | Std | 1.67E−01 | 5.82E+02 | 3.29E−01 | 4.18E+02 | 5.12E+02 | 6.88E+02 | 1.80E+03 | 1.30E+03 |
| | 100 | Mean | −4.19E+04 | −4.14E+04 | −4.19E+04 | −1.38E+04 | −2.49E+04 | −2.15E+04 | −3.53E+04 | −1.00E+04 |
| | | Std | 1.38E−01 | 1.30E+03 | 1.13E+01 | 8.36E+02 | 8.47E+02 | 2.13E+03 | 6.24E+03 | 3.56E+03 |
| | 500 | Mean | −2.09E+05 | −2.06E+05 | −2.09E+05 | −3.78E+04 | −8.26E+04 | −5.92E+04 | −1.79E+05 | −2.31E+04 |
| | | Std | 1.05E+01 | 5.34E+03 | 1.92E+02 | 1.66E+03 | 3.14E+03 | 5.25E+03 | 2.71E+04 | 9.43E+03 |
| F8 | 1000 | Mean | −4.19E+05 | −4.07E+05 | −4.19E+05 | −5.49E+04 | −1.27E+05 | −8.89E+04 | −3.46E+05 | −3.06E+04 |
| | | Std | 9.28E+00 | 2.56E+04 | 3.00E+02 | 2.36E+03 | 5.27E+03 | 6.14E+03 | 5.58E+04 | 1.27E+04 |

*Continued on next page*

| Function | *D* | Metric | IROA | ROA | SMA | AOA | MPA | SSA | WOA | PSO |
|---|---|---|---|---|---|---|---|---|---|---|
| F9 | 30 | Mean | 0 | 0 | 0 | 1.42E−06 | 0 | 6.05E+01 | 3.79E−15 | 1.03E+02 |
| | | Std | 0 | 0 | 0 | 1.49E−06 | 0 | 1.68E+01 | 1.44E−14 | 2.81E+01 |
| | 100 | Mean | 0 | 0 | 0 | 1.80E−04 | 0 | 2.40E+02 | 1.14E−14 | 7.81E+02 |
| | | Std | 0 | 0 | 0 | 4.64E−05 | 0 | 3.69E+01 | 4.58E−14 | 6.71E+01 |
| | 500 | Mean | 0 | 0 | 0 | 1.05E−02 | 0 | 3.11E+03 | 6.06E−14 | 6.27E+03 |
| | | Std | 0 | 0 | 0 | 9.58E−04 | 0 | 1.38E+02 | 3.32E−13 | 2.23E+02 |
| | 1000 | Mean | 0 | 0 | 0 | 3.75E−02 | 0 | 7.62E+03 | 6.06E−14 | 1.43E+04 |
| | | Std | 0 | 0 | 0 | 1.86E−03 | 0 | 1.67E+02 | 3.32E−13 | 3.67E+02 |
| F10 | 30 | Mean | 8.88E−16 | 8.88E−16 | 8.88E−16 | 4.58E−04 | 1.43E−12 | 2.45E+00 | 4.32E−15 | 1.15E−01 |
| | | Std | 0 | 0 | 0 | 1.87E−04 | 8.43E−13 | 5.80E−01 | 3.02E−15 | 3.10E−01 |
| | 100 | Mean | 8.88E−16 | 8.88E−16 | 8.88E−16 | 3.42E−03 | 4.17E−11 | 1.03E+01 | 4.44E−15 | 3.71E+00 |
| | | Std | 0 | 0 | 0 | 3.40E−04 | 1.97E−11 | 1.00E+00 | 2.47E−15 | 2.96E−01 |
| | 500 | Mean | 8.88E−16 | 8.88E−16 | 8.88E−16 | 2.66E−02 | 4.40E−10 | 1.43E+01 | 3.97E−15 | 1.20E+01 |
| | | Std | 0 | 0 | 0 | 1.02E−03 | 1.90E−10 | 2.77E−01 | 2.23E−15 | 4.56E−01 |
| | 1000 | Mean | 8.88E−16 | 8.88E−16 | 8.88E−16 | 3.34E−02 | 8.53E−10 | 1.45E+01 | 5.03E−15 | 1.60E+01 |
| | | Std | 0 | 0 | 0 | 5.12E−04 | 3.29E−10 | 1.64E−01 | 3.11E−15 | 2.27E−01 |
| F11 | 30 | Mean | 0 | 0 | 0 | 6.00E−04 | 0 | 1.59E−02 | 4.32E−03 | 7.69E−03 |
| | | Std | 0 | 0 | 0 | 2.21E−03 | 0 | 1.26E−02 | 2.37E−02 | 1.00E−02 |
| | 100 | Mean | 0 | 0 | 0 | 2.35E−01 | 0 | 1.36E+01 | 1.13E−02 | 4.11E−01 |
| | | Std | 0 | 0 | 0 | 2.99E−01 | 0 | 3.21E+00 | 6.18E−02 | 8.93E−02 |
| | 500 | Mean | 0 | 0 | 0 | 1.25E+03 | 0 | 8.41E+02 | 0 | 7.84E+01 |
| | | Std | 0 | 0 | 0 | 3.05E+02 | 0 | 7.28E+01 | 0 | 1.09E+01 |
| | 1000 | Mean | 0 | 0 | 0 | 1.41E+04 | 3.33E−17 | 2.10E+03 | 0 | 2.73E+02 |
| | | Std | 0 | 0 | 0 | 2.68E+03 | 5.17E−17 | 9.16E+01 | 0 | 1.60E+01 |
| F12 | 30 | Mean | 1.79E−05 | 9.82E−03 | 5.90E−03 | 7.30E−01 | 9.11E−06 | 6.41E+00 | 1.85E−02 | 2.07E−02 |
| | | Std | 1.50E−05 | 5.64E−03 | 8.01E−03 | 3.63E−02 | 4.77E−05 | 2.98E+00 | 1.17E−02 | 5.02E−02 |

*Continued on next page*

| Function | *D* | Metric | IROA | ROA | SMA | AOA | MPA | SSA | WOA | PSO |
|---|---|---|---|---|---|---|---|---|---|---|
| F12 | 100 | Mean | 2.14E−05 | 1.99E−02 | 6.94E−03 | 9.11E−01 | 4.30E−02 | 3.54E+01 | 3.98E−02 | 5.21E+00 |
| | | Std | 4.69E−05 | 1.27E−02 | 1.46E−02 | 5.46E−02 | 8.70E−03 | 1.16E+01 | 1.90E−02 | 1.49E+00 |
| | 500 | Mean | 2.74E−05 | 3.20E−02 | 6.53E−03 | 9.27E−01 | 4.17E−01 | 1.58E+06 | 9.14E−02 | 2.34E+05 |
| | | Std | 5.17E−05 | 1.85E−02 | 1.24E−02 | 2.22E−02 | 2.46E−02 | 8.85E+05 | 3.72E−02 | 1.20E+05 |
| | 1000 | Mean | 1.32E−05 | 3.80E−02 | 3.12E−03 | 1.04E+00 | 6.44E−01 | 1.16E+07 | 8.04E−02 | 9.77E+06 |
| | | Std | 3.40E−05 | 1.87E−02 | 5.69E−03 | 1.22E−02 | 2.31E−02 | 4.20E+06 | 3.59E−02 | 1.83E+06 |
| F13 | 30 | Mean | 4.82E−04 | 2.36E−01 | 8.30E−03 | 2.96E+00 | 8.78E−03 | 1.79E+01 | 5.32E−01 | 5.85E−03 |
| | | Std | 2.06E−03 | 1.53E−01 | 9.46E−03 | 2.87E−02 | 1.52E−02 | 1.59E+01 | 2.60E−01 | 5.52E−03 |
| | 100 | Mean | 1.15E−03 | 1.48E+00 | 1.46E−01 | 9.92E+00 | 8.16E+00 | 7.86E+03 | 2.66E+00 | 5.96E+01 |
| | | Std | 1.49E−03 | 7.62E−01 | 2.30E−01 | 1.27E−02 | 1.72E+00 | 1.33E+04 | 7.78E−01 | 1.44E+01 |
| | 500 | Mean | 6.05E−03 | 7.90E+00 | 1.30E+00 | 4.93E+01 | 4.88E+01 | 3.68E+07 | 1.82E+01 | 4.19E+06 |
| | | Std | 9.93E−03 | 2.90E+00 | 2.17E+00 | 3.39E−01 | 2.11E−01 | 8.43E+06 | 7.02E+00 | 9.56E+05 |
| | 1000 | Mean | 7.73E−03 | 1.85E+01 | 1.50E+00 | 1.00E+02 | 9.86E+01 | 1.53E+08 | 3.79E+01 | 8.19E+07 |
| | | Std | 1.57E−02 | 7.99E+00 | 2.33E+00 | 3.20E−01 | 2.67E−01 | 3.35E+07 | 1.09E+01 | 1.06E+07 |
| Rank | 30 | Mean | **1.92** | 3.17 | 2.42 | 6.33 | 3.33 | 7.50 | 5.00 | 6.33 |
| | | Overall | **1** | 3 | 2 | 6.5 | 4 | 8 | 5 | 6.5 |
| | 100 | Mean | **1.75** | 2.67 | 2.08 | 6.17 | 4.17 | 7.50 | 4.33 | 7.33 |
| | | Overall | **1** | 3 | 2 | 6 | 4 | 8 | 5 | 7 |
| | 500 | Mean | **1.83** | 2.75 | 2.17 | 6.50 | 4.25 | 7.33 | 4.00 | 7.17 |
| | | Overall | **1** | 3 | 2 | 6 | 5 | 8 | 4 | 7 |
| | 1000 | Mean | **1.75** | 2.67 | 2.08 | 6.50 | 4.58 | 7.17 | 3.92 | 7.33 |
| | | Overall | **1** | 3 | 2 | 6 | 5 | 7 | 4 | 8 |

**Table 6.** Results of the IROA and other metaheuristic algorithms on fixed-dimension multimodal test functions (F14−F23).

| Function | Metric | IROA | ROA | SMA | AOA | MPA | SSA | WOA | PSO |
|---|---|---|---|---|---|---|---|---|---|
| F14 | Mean | 9.98E−01 | 4.19E+00 | 9.98E−01 | 1.09E+01 | 9.98E−01 | 1.20E+00 | 2.93E+00 | 3.20E+00 |
| | Std | 2.52E−13 | 4.69E+00 | 2.19E−13 | 3.39E+00 | 1.72E−16 | 6.05E−01 | 2.70E+00 | 2.59E+00 |
| F15 | Mean | 3.08E−04 | 5.28E−04 | 6.36E−04 | 6.75E−03 | 3.07E−04 | 4.35E−03 | 5.48E−04 | 3.78E−03 |
| | Std | 4.77E−07 | 3.31E−04 | 3.08E−04 | 1.00E−02 | 3.00E−15 | 1.22E−02 | 2.18E−04 | 6.85E−03 |
| F16 | Mean | −1.03E+00 | −1.03E+00 | −1.03E+00 | −1.03E+00 | −1.03E+00 | −1.03E+00 | −1.03E+00 | −1.03E+00 |
| | Std | 5.79E−09 | 8.73E−08 | 3.52E−10 | 2.84E−11 | 4.40E−16 | 2.31E−14 | 8.01E−10 | 6.39E−16 |
| F17 | Mean | 3.98E−01 | 3.98E−01 | 3.98E−01 | 4.00E−01 | 3.98E−01 | 3.98E−01 | 3.98E−01 | 3.98E−01 |
| | Std | 1.56E−08 | 1.13E−05 | 5.95E−09 | 9.04E−03 | 0 | 1.44E−14 | 1.29E−05 | 0.00E+00 |
| F18 | Mean | 3.00E+00 | 3.00E+00 | 3.00E+00 | 1.96E+01 | 3.00E+00 | 3.00E+00 | 4.80E+00 | 3.00E+00 |
| | Std | 2.17E−09 | 1.95E−04 | 7.31E−11 | 2.83E+01 | 2.07E−15 | 2.11E−13 | 6.85E+00 | 1.82E−15 |
| F19 | Mean | −3.86E+00 | −3.86E+00 | −3.86E+00 | −3.86E+00 | −3.86E+00 | −3.86E+00 | −3.86E+00 | −3.86E+00 |
| | Std | 1.52E−04 | 2.34E−03 | 6.51E−08 | 7.52E−05 | 2.39E−15 | 9.60E−13 | 7.39E−03 | 2.64E−15 |
| F20 | Mean | −3.27E+00 | −3.23E+00 | −3.24E+00 | −3.29E+00 | −3.32E+00 | −3.20E+00 | −3.23E+00 | −3.24E+00 |
| | Std | 7.52E−02 | 1.56E−01 | 5.72E−02 | 5.12E−02 | 3.07E−12 | 4.34E−02 | 1.04E−01 | 1.25E−01 |
| F21 | Mean | −1.02E+01 | −1.01E+01 | −1.02E+01 | −7.22E+00 | −1.02E+01 | −7.81E+00 | −7.92E+00 | −6.38E+00 |
| | Std | 4.91E−04 | 1.68E−02 | 1.16E−04 | 3.10E+00 | 3.28E−11 | 3.22E+00 | 2.78E+00 | 3.28E+00 |
| F22 | Mean | −1.04E+01 | −1.04E+01 | −1.04E+01 | −6.34E+00 | −1.02E+01 | −7.57E+00 | −7.64E+00 | −8.22E+00 |
| | Std | 1.67E−03 | 2.59E−02 | 7.79E−05 | 3.08E+00 | 9.70E−01 | 3.39E+00 | 3.05E+00 | 3.24E+00 |
| F23 | Mean | −1.05E+01 | −1.05E+01 | −1.05E+01 | −8.02E+00 | −1.05E+01 | −7.97E+00 | −7.94E+00 | −8.93E+00 |
| | Std | 7.91E−04 | 9.17E−03 | 6.28E−05 | 3.43E+00 | 7.15E−11 | 3.47E+00 | 3.25E+00 | 3.02E+00 |
| Rank | Mean | 3 | 4.15 | 3.45 | 6.4 | **2.9** | 5.55 | 5.45 | 5.1 |
| | Overall | 2 | 4 | 3 | 8 | **1** | 7 | 6 | 5 |

**Table 7.** Results of the Wilcoxon signed-rank test between IROA and other metaheuristic algorithms (F1−F23).

| Function | *D* | IROA vs. ROA | IROA vs. SMA | IROA vs. AOA | IROA vs. MPA | IROA vs. SSA | IROA vs. WOA | IROA vs. PSO |
|---|---|---|---|---|---|---|---|---|
| F1 | 30 | 5.00E−01 | 1 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 100 | 1 | 2.50E−01 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 500 | 2.50E−01 | 5.00E−01 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 1000 | 2.50E−01 | 3.91E−03 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F2 | 30 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 100 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 500 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 1000 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F3 | 30 | 6.10E−05 | 1 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 100 | 6.10E−05 | 2.50E−01 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 500 | 6.10E−05 | 9.77E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 1000 | 6.10E−05 | 4.88E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F4 | 30 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 100 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 500 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 1000 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F5 | 30 | 6.10E−05 | 8.54E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 100 | 6.10E−05 | 3.05E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 500 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 1000 | 6.10E−05 | 1.16E−03 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F6 | 30 | 6.10E−05 | 3.05E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 5.37E−03 |
| | 100 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 500 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 1000 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |

*Continued on next page*

| Function | *D* | IROA vs. ROA | IROA vs. SMA | IROA vs. AOA | IROA vs. MPA | IROA vs. SSA | IROA vs. WOA | IROA vs. PSO |
|---|---|---|---|---|---|---|---|---|
| F7 | 30 | 5.61E−01 | 5.99E−01 | 4.79E−02 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 100 | 5.61E−01 | 1.16E−03 | 7.62E−01 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 500 | 5.54E−02 | 2.01E−03 | 3.02E−02 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 1000 | 1 | 8.54E−04 | 1.81E−02 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F8 | 30 | 1.83E−04 | 4.27E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 100 | 1.81E−02 | 1.22E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 500 | 3.05E−04 | 4.27E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 1000 | 1.22E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F9 | 30 | 1 | 1 | 6.10E−05 | 1 | 6.10E−05 | 1 | 6.10E−05 |
| | 100 | 1 | 1 | 6.10E−05 | 1 | 6.10E−05 | 1 | 6.10E−05 |
| | 500 | 1 | 1 | 6.10E−05 | 1 | 6.10E−05 | 1 | 6.10E−05 |
| | 1000 | 1 | 1 | 6.10E−05 | 1 | 6.10E−05 | 1 | 6.10E−05 |
| F10 | 30 | 1 | 1 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 4.88E−04 | 6.10E−05 |
| | 100 | 1 | 1 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 2.44E−04 | 6.10E−05 |
| | 500 | 1 | 1 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 9.77E−04 | 6.10E−05 |
| | 1000 | 1 | 1 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 2.44E−04 | 6.10E−05 |
| F11 | 30 | 1 | 1 | 6.10E−05 | 1 | 6.10E−05 | 1 | 6.10E−05 |
| | 100 | 1 | 1 | 6.10E−05 | 1 | 6.10E−05 | 1 | 6.10E−05 |
| | 500 | 1 | 1 | 6.10E−05 | 1 | 6.10E−05 | 1 | 6.10E−05 |
| | 1000 | 1 | 1 | 6.10E−05 | 5.00E−01 | 6.10E−05 | 5.00E−01 | 6.10E−05 |
| F12 | 30 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 1.22E−04 | 6.10E−05 | 6.10E−05 | 1.51E−02 |
| | 100 | 6.10E−05 | 1.22E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 500 | 6.10E−05 | 6.10E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 1000 | 6.10E−05 | 1.22E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F13 | 30 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 1.88E−01 | 6.10E−05 | 6.10E−05 | 2.08E−01 |
| | 100 | 6.10E−05 | 6.10E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |

*Continued on next page*

| Function | *D* | IROA vs. ROA | IROA vs. SMA | IROA vs. AOA | IROA vs. MPA | IROA vs. SSA | IROA vs. WOA | IROA vs. PSO |
|---|---|---|---|---|---|---|---|---|
| F13 | 500 | 6.10E−05 | 3.05E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| | 1000 | 6.10E−05 | 6.10E−04 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F14 | 2 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 8.33E−02 | 8.54E−04 | 2.62E−03 |
| F15 | 4 | 1.35E−01 | 1.22E−04 | 2.56E−02 | 6.10E−05 | 6.10E−05 | 3.05E−04 | 6.10E−05 |
| F16 | 2 | 3.02E−02 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F17 | 2 | 5.24E−01 | 6.10E−05 | 8.36E−03 | 6.10E−05 | 6.10E−05 | 5.61E−01 | 6.10E−05 |
| F18 | 2 | 9.34E−01 | 6.10E−05 | 7.62E−01 | 6.10E−05 | 6.10E−05 | 1.51E−01 | 6.10E−05 |
| F19 | 3 | 6.10E−05 | 6.10E−05 | 1.51E−02 | 6.10E−05 | 6.10E−05 | 1.22E−04 | 6.10E−05 |
| F20 | 6 | 1 | 4.54E−01 | 6.10E−05 | 6.10E−05 | 2.56E−02 | 3.89E−01 | 9.78E−01 |
| F21 | 4 | 1.16E−03 | 3.30E−01 | 9.46E−02 | 6.10E−05 | 7.30E−02 | 6.10E−05 | 2.62E−03 |
| F22 | 4 | 6.10E−04 | 2.77E−01 | 1.22E−04 | 6.10E−05 | 8.04E−01 | 6.10E−05 | 7.62E−01 |
| F23 | 4 | 6.10E−05 | 5.54E−02 | 3.05E−04 | 6.10E−05 | 8.04E−01 | 6.10E−05 | 3.30E−01 |
| Overall (+/=/−) | | 37/24/1 | 40/22/0 | 55/3/4 | 49/9/4 | 58/4/0 | 51/11/0 | 58/4/0 |

**Table 8.** Comparison results of IROA and ROA on CEC2021 test functions (CEC_01−CEC_10).

| Function | Algorithm | Best | Mean | Std |
|---|---|---|---|---|
| CEC_01 | IROA | 1.96E+07 | 3.25E+08 | 3.29E+08 |
| | ROA | 3.23E+07 | 1.26E+09 | 1.79E+09 |
| CEC_02 | IROA | 1.40E+03 | 2.01E+03 | 2.82E+02 |
| | ROA | 1.61E+03 | 2.08E+03 | 2.41E+02 |
| CEC_03 | IROA | 7.22E+02 | 7.67E+02 | 2.17E+01 |
| | ROA | 7.35E+02 | 7.84E+02 | 2.00E+01 |
| CEC_04 | IROA | 1.90E+03 | 1.91E+03 | 3.30E+00 |
| | ROA | 1.90E+03 | 2.27E+03 | 1.94E+03 |
| CEC_05 | IROA | 3.22E+03 | 2.31E+04 | 4.06E+04 |
| | ROA | 3.71E+03 | 7.63E+04 | 1.46E+05 |
| CEC_06 | IROA | 1.60E+03 | 1.75E+03 | 1.03E+02 |
| | ROA | 1.75E+03 | 1.86E+03 | 1.11E+02 |
| CEC_07 | IROA | 3.43E+03 | 9.22E+03 | 6.89E+03 |
| | ROA | 3.33E+03 | 1.18E+04 | 1.01E+04 |
| CEC_08 | IROA | 2.31E+03 | 2.32E+03 | 1.58E+01 |
| | ROA | 2.27E+03 | 2.40E+03 | 1.05E+02 |
| CEC_09 | IROA | 2.74E+03 | 2.76E+03 | 7.43E+01 |
| | ROA | 2.75E+03 | 2.77E+03 | 7.91E+01 |
| CEC_10 | IROA | 2.93E+03 | 2.97E+03 | 6.71E+01 |
| | ROA | 2.93E+03 | 3.01E+03 | 5.87E+01 |

### *4.4. Comparison with modified algorithms*

In this section, to further demonstrate the superior search capability of the IROA, six modified algorithms, including HHOCM [33], SWGWO [34], ROLGWO [35], DSCA [36], MALO [37], and HSCAHS [38] are employed to compare with the IROA. Table 9 shows the results of classical test functions. Note that the dimension of test functions F1−F13 is set to 1000 for better comparison.

From Table 9, the overall ranking of the IROA is still the first among these advanced algorithms. On functions F1−F7, IROA obtains the theoretical optimal values on F1−F4, whereas HHOCM wins the first on F5 and F6, and hybrid algorithm HSCAHS wins on F7. According to the optimal results on F5−F7, IROA still shows very competitive exploitation performance. For the functions F8−F13, IROA displays excellent exploration capability on high-dimensional cases. However, it is also noted that HHOCM ranks first on F12 and F13, while the IROA ranks second. Moreover, on the fixed-dimension test functions F14−F23, the IROA also presents the best optimal results with very low standard deviation, which means high stability.

Table 10 presents the Wilcoxon signed-rank test results between IROA and other modified optimization methods. The statistical results show that most p-values are lower than 0.05, which means the IROA has a significant difference in optimization capability compared to other advanced optimization algorithms. According to the overall results, IROA is able to obtain better or similar optimal solutions than others. Moreover, it is noted that IROA is obviously better than SWGWO, DSCA, MALO, and HSCAHS.

Therefore, the proposed IROA also has very comparative performance compared to other modified optimization algorithms.

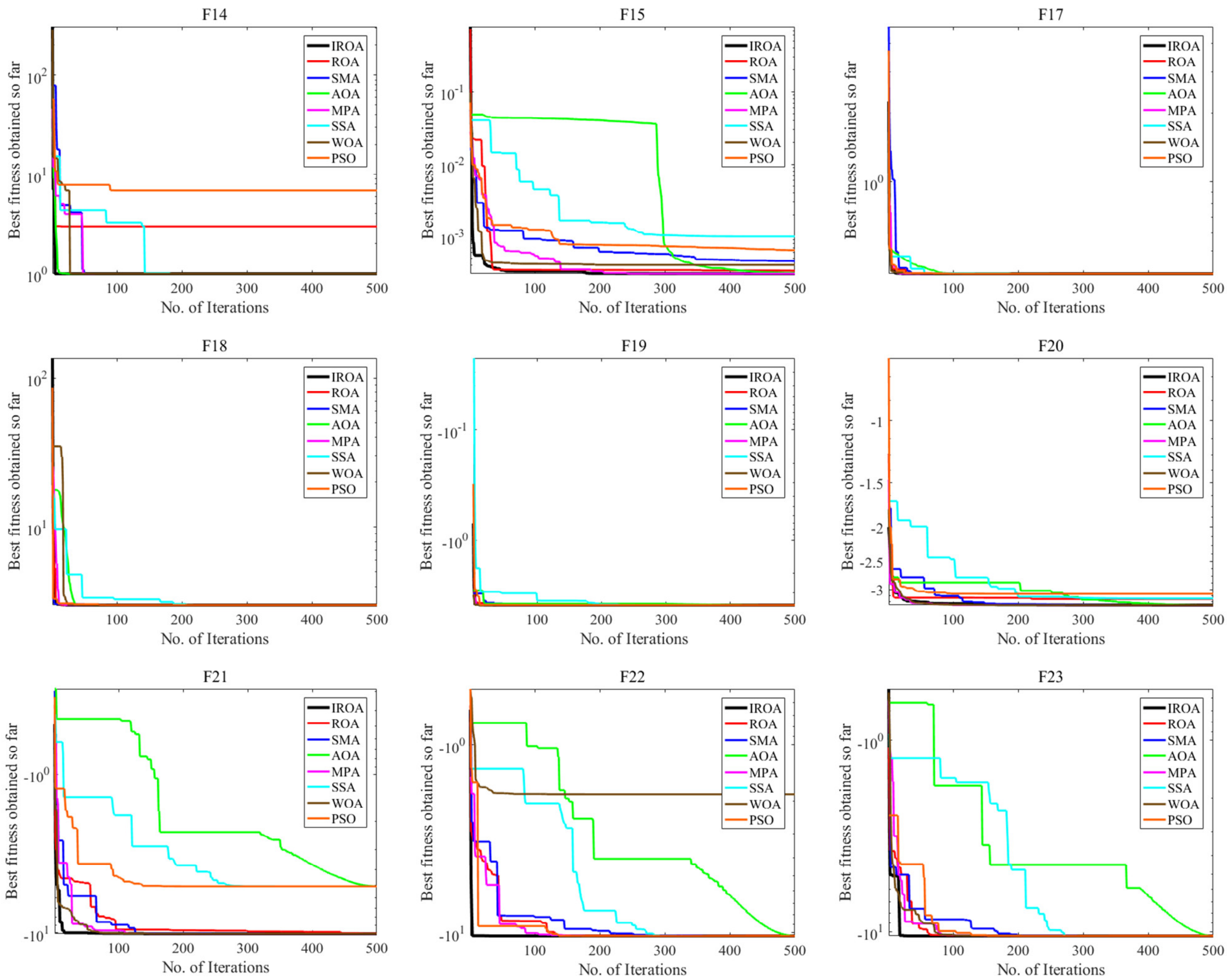


**Figure 8.** The convergence curves for the optimization algorithms on fixed-dimension test functions (F14, F15, F17–F23).

**Table 9.** Results of the IROA and other modified algorithms on test functions (F1−F23).

| Function | Metric | IROA | HHOCM | SWGWO | ROLGWO | DSCA | MALO | HSCAHS |
|---|---|---|---|---|---|---|---|---|
| F1 | Mean | 0 | 0 | 6.42E−07 | 1.9763E−323 | 6.22E−84 | 5.71E+05 | 2.20E−25 |
| | Std | 0 | 0 | 3.04E−07 | 0 | 3.36E−83 | 1.19E+05 | 5.09E−25 |
| F2 | Mean | 0 | 0 | 1.13E−03 | 1.06E−164 | Inf | Inf | 6.74E−15 |
| | Std | 0 | 0 | 1.03E−03 | 0 | NaN | NaN | 1.39E−14 |
| F3 | Mean | 0 | 0 | 7.73E+05 | 6.1755E−319 | 3.22E−35 | 7.55E+06 | 1.16E−22 |
| | Std | 0 | 0 | 3.07E+05 | 0 | 1.76E−34 | 2.61E+06 | 3.85E−22 |
| F4 | Mean | 0 | 9.82E−199 | 8.31E+01 | 4.63E−84 | 2.82E−33 | 5.48E+01 | 1.27E−07 |
| | Std | 0 | 0 | 3.67E+00 | 1.95E−83 | 1.52E−32 | 5.79E+00 | 3.56E−07 |
| F5 | Mean | 5.21E+00 | 9.42E−01 | 9.98E+02 | 9.98E+02 | 9.99E+02 | 7.18E+02 | 9.99E+02 |
| | Std | 1.14E+01 | 1.45E+00 | 8.14E−02 | 2.44E−01 | 1.54E−02 | 3.74E+02 | 9.42E−03 |
| F6 | Mean | 9.02E−02 | 3.05E−02 | 2.30E+02 | 2.13E+02 | 2.48E+02 | 1.36E+01 | 2.49E+02 |
| | Std | 1.40E−01 | 5.30E−02 | 1.27E+00 | 3.07E+00 | 2.67E−01 | 6.27E+00 | 1.72E−01 |
| F7 | Mean | 1.66E−04 | 1.52E−04 | 1.50E−02 | 7.93E−05 | 4.01E−03 | 1.06E−04 | 7.40E−05 |
| | Std | 1.77E−04 | 1.66E−04 | 6.66E−03 | 7.89E−05 | 3.17E−03 | 1.00E−04 | 6.27E−05 |
| F8 | Mean | −4.19E+05 | −4.19E+05 | −2.97E+04 | −5.51E+04 | −2.69E+04 | −3.78E+05 | −1.46E+04 |
| | Std | 9.28E+00 | 5.78E+01 | 2.18E+04 | 3.27E+04 | 2.71E+03 | 3.91E+04 | 2.41E+03 |
| F9 | Mean | 0 | 0 | 5.80E−06 | 0 | 0 | 8.81E+03 | 0 |
| | Std | 0 | 0 | 2.42E−05 | 0 | 0 | 2.95E+02 | 0 |
| F10 | Mean | 8.88E−16 | 8.88E−16 | 2.45E−05 | 3.14E−15 | 8.88E−16 | 1.61E+01 | 5.63E−15 |
| | Std | 0 | 0 | 5.45E−06 | 1.74E−15 | 0 | 1.55E+00 | 7.19E−15 |
| F11 | Mean | 0 | 0 | 5.17E−08 | 0 | 0 | 5.20E+03 | 0 |
| | Std | 0 | 0 | 3.39E−08 | 0 | 0 | 8.96E+02 | 0 |
| F12 | Mean | 1.32E−05 | 6.65E−06 | 9.96E−01 | 8.59E−01 | 1.17E+00 | 5.01E−04 | 1.18E+00 |
| | Std | 3.40E−05 | 1.26E−05 | 1.85E−02 | 2.09E−02 | 5.21E−03 | 4.52E−04 | 4.06E−03 |

*Continued on next page*

| Function | Metric | IROA | HHOCM | SWGWO | ROLGWO | DSCA | MALO | HSCAHS |
|---|---|---|---|---|---|---|---|---|
| F13 | Mean | 7.73E−03 | 2.11E−03 | 9.80E+01 | 9.99E+01 | 1.00E+02 | 1.36E+00 | 9.99E+01 |
| | Std | 1.57E−02 | 2.34E−03 | 1.16E+00 | 1.30E−02 | 3.85E−02 | 7.91E−01 | 2.65E−02 |
| F14 | Mean | 9.98E−01 | 1.20E+00 | 5.63E+00 | 5.33E+00 | 1.51E+00 | 1.59E+00 | 2.92E+00 |
| | Std | 2.52E−13 | 4.81E−01 | 4.62E+00 | 4.69E+00 | 6.48E−01 | 8.48E−01 | 2.22E−01 |
| F15 | Mean | 3.08E−04 | 3.32E−04 | 5.76E−04 | 3.50E−04 | 1.35E−03 | 7.94E−04 | 2.80E−03 |
| | Std | 4.77E−07 | 2.88E−05 | 1.62E−04 | 7.63E−05 | 3.67E−04 | 3.49E−04 | 1.94E−03 |
| F16 | Mean | −1.03E+00 | −1.03E+00 | −1.03E+00 | −1.03E+00 | −1.03E+00 | −1.03E+00 | −1.02E+00 |
| | Std | 5.79E−09 | 1.56E−09 | 4.44E−06 | 3.81E−05 | 2.89E−04 | 1.10E−13 | 7.19E−03 |
| F17 | Mean | 3.98E−01 | 3.98E−01 | 3.98E−01 | 3.98E−01 | 4.00E−01 | 3.98E−01 | 6.84E−01 |
| | Std | 1.56E−08 | 3.33E−08 | 1.08E−05 | 9.51E−07 | 3.14E−03 | 2.17E−14 | 3.11E−01 |
| F18 | Mean | 3.00E+00 | 3.00E+00 | 3.00E+00 | 3.00E+00 | 3.01E+00 | 3.00E+00 | 3.00E+00 |
| | Std | 2.17E−09 | 2.08E−08 | 5.96E−05 | 7.89E−05 | 9.13E−03 | 4.36E−13 | 5.34E−03 |
| F19 | Mean | −3.86E+00 | −3.86E+00 | −3.86E+00 | −3.86E+00 | −3.84E+00 | −3.86E+00 | −3.36E+00 |
| | Std | 1.52E−04 | 3.91E−04 | 4.62E−03 | 3.79E−04 | 1.74E−02 | 2.17E−12 | 3.42E−01 |
| F20 | Mean | −3.27E+00 | −3.27E+00 | −3.18E+00 | −3.25E+00 | −3.02E+00 | −3.24E+00 | −1.50E+00 |
| | Std | 7.52E−02 | 7.38E−02 | 1.20E−01 | 8.25E−02 | 8.13E−02 | 5.73E−02 | 5.34E−01 |
| F21 | Mean | −1.02E+01 | −5.22E+00 | −7.25E+00 | −5.45E+00 | −4.34E+00 | −7.37E+00 | −6.53E−01 |
| | Std | 4.91E−04 | 9.30E−01 | 2.57E+00 | 1.56E+00 | 7.39E−01 | 2.92E+00 | 3.40E−01 |
| F22 | Mean | −1.04E+01 | −5.26E+00 | −8.78E+00 | −6.47E+00 | −4.41E+00 | −7.02E+00 | −7.44E−01 |
| | Std | 1.67E−03 | 9.70E−01 | 2.49E+00 | 2.62E+00 | 3.88E−01 | 3.28E+00 | 2.58E−01 |
| F23 | Mean | −1.05E+01 | −5.49E+00 | −9.98E+00 | −5.97E+00 | −4.50E+00 | −8.21E+00 | −9.18E−01 |
| | Std | 7.91E−04 | 1.36E+00 | 1.65E+00 | 1.97E+00 | 7.69E−01 | 3.19E+00 | 3.25E−01 |
| Rank | Mean | **2.07** | 2.63 | 4.54 | 3.78 | 5.33 | 4.33 | 5.15 |
| | Overall | **1** | 2 | 5 | 3 | 7 | 4 | 6 |

**Table 10.** Results of the Wilcoxon signed-rank test between IROA and other modified algorithms (F1−F23).

| Function | IROA vs. HHOCM | IROA vs. SWGWO | IROA vs. ROLGWO | IROA vs. DSCA | IROA vs. MALO | IROA vs. HSCAHS |
|---|---|---|---|---|---|---|
| F1 | 1 | 6.10E−05 | 6.25E−02 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F2 | 1 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F3 | 1 | 6.10E−05 | 1.56E−02 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F4 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F5 | 6.37E−02 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F6 | 6.79E−01 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F7 | 1.88E−01 | 6.10E−05 | 3.30E−01 | 6.10E−05 | 4.21E−01 | 2.77E−01 |
| F8 | 8.36E−03 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F9 | 1 | 6.10E−05 | 1 | 1 | 6.10E−05 | 1 |
| F10 | 1 | 6.10E−05 | 4.88E−04 | 1 | 6.10E−05 | 9.77E−04 |
| F11 | 1 | 6.10E−05 | 1 | 1 | 6.10E−05 | 1 |
| F12 | 3.89E−01 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F13 | 3.59E−01 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F14 | 1.51E−01 | 1.16E−03 | 1.53E−03 | 6.10E−05 | 8.04E−01 | 6.10E−05 |
| F15 | 6.71E−03 | 6.10E−05 | 1.51E−02 | 6.10E−05 | 3.05E−04 | 6.10E−05 |
| F16 | 6.10E−05 | 3.36E−03 | 4.27E−03 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F17 | 6.10E−05 | 5.54E−02 | 1.51E−01 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F18 | 6.10E−05 | 2.56E−02 | 8.90E−01 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F19 | 2.56E−02 | 5.37E−03 | 1.51E−01 | 6.10E−05 | 6.10E−05 | 6.10E−05 |
| F20 | 5.37E−03 | 1.16E−03 | 4.21E−01 | 6.10E−05 | 4.13E−02 | 6.10E−05 |
| F21 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 2.56E−02 | 6.10E−05 |
| F22 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 2.56E−02 | 6.10E−05 |
| F23 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 6.10E−05 | 8.36E−03 | 6.10E−05 |
| Overall (+/=/−) | 11/12/0 | 22/1/0 | 15/8/0 | 20/3/0 | 21/2/0 | 20/3/0 |

## 5. Results of engineering optimization problems

Five constrained engineering design problems are selected for testing to verify the performance and efficiency of the proposed IROA in solving practical problems. These problems include the welded beam design problem, tension/compression spring design problem, three-bar truss design problem, car crashworthiness, and tubular column design problem. The maximum number of iterations is also set as 500, and the number of search agents is 30. The detailed results of the proposed IROA and other optimization methods are presented in the following subsections. Note that the best solution is highlighted in bold.

### *5.1. The welded beam design problem*

The design of a welded beam [41] requires to obtain the lowest fabrication cost with four variables, i.e., the height of the bar ($t$), the thickness of the bar ($b$), length of the welded part of the bar ($l$), and thickness of the weld ($h$), as shown in Figure 9. The four constraints of this design problem are buckling load ($p_c$), bending stress ($\theta$), shear stress ($\tau$), and deflection of the beam ($\delta$).

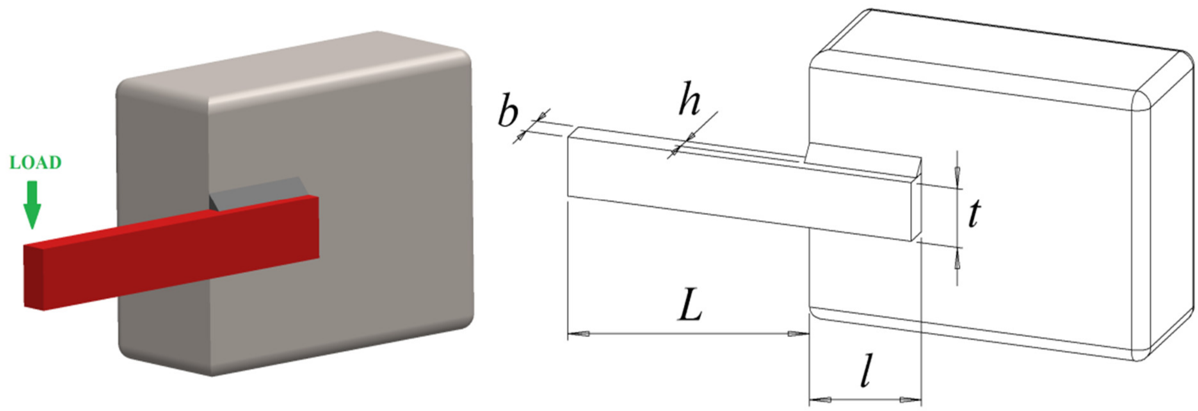


Figure 9. Welded beam design problem: three-dimensional model diagram (left), structure parameters (right).

The mathematical formulas for this problem can be expressed as follows:
Consider

$$\vec{x} = [x_1, x_2, x_3, x_4] = [h, l, t, b]$$

Minimize

$$f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$$

Subject to

$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max}$$

$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max}$$

$$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max}$$

$$g_4(\vec{x}) = x_1 - x_4 \leq 0$$

$$g_5(\vec{x}) = P - P_C(\vec{x}) \leq 0$$

$$g_6(\vec{x}) = 0.125 - x_1 \leq 0$$

$$g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5 \leq 0$$

where

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P(L + \frac{x_2}{2})$$

$$R = \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2}$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2\right]\right\}$$

$$\sigma(\vec{x}) = \frac{6PL}{x_3^2x_4}, \delta(\vec{x}) = \frac{6PL^3}{Ex_3^3x_4}$$

$$P_C(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

Variable range

$P = 6000$ lb, $L = 14$ in, $E = 30 \times 10^6$ psi, $G = 12 \times 10^6$ psi
$\tau_{\text{max}}$=13600 psi, $\sigma_{\text{max}}$=30000 psi, $\delta_{\text{max}}$=0.25 in

This problem is figured out by the proposed IROA and other eight optimization methods, like ROA [27], AOA [10], SMA [11], WOA [7], GWO [8], GSA [42], CSCA [43], and CPSO [44]. The optimal solutions are listed in Table 11. From Table 11, it can be observed that IROA gets the least cost 1.695245, and the four corresponding variables are $h$ = 0.205734, $l$ = 3.253035, $t$ = 9.036624, and $b$ = 0.205730. Thus IROA outperforms other methods when dealing with the welded beam design problem.

**Table 11.** Comparison of optimal solutions for the welded beam design problem.

| Algorithm | Optimal values for variables | | | | Optimal cost |
|---|---|---|---|---|---|
| | *h* | *l* | *t* | *b* | |
| IROA | 0.205734 | 3.253035 | 9.036624 | 0.205730 | 1.695245 |
| ROA [27] | 0.200077 | 3.365754 | 9.011182 | 0.206893 | 1.706447 |
| AOA [10] | 0.194475 | 2.57092 | 10.000 | 0.201827 | 1.7164 |
| SMA [11] | 0.2054 | 3.2589 | 9.0384 | 0.2058 | 1.69604 |
| WOA [7] | 0.205396 | 3.484293 | 9.037426 | 0.206276 | 1.730499 |
| GWO [8] | 0.205676 | 3.478377 | 9.03681 | 0.205778 | 1.72624 |
| GSA [42] | 0.182129 | 3.856979 | 10.00000 | 0.202376 | 1.879952 |
| CSCA [43] | 0.203137 | 3.542998 | 9.033498 | 0.206179 | 1.733461 |
| CPSO [44] | 0.202369 | 3.544214 | 9.04821 | 0.205723 | 1.72802 |

*5.2. The tension/compression spring design problem*

The lowest manufacturing weight is the objective of the tension/compression spring design problem [45]. In this problem, there are three constraints, i.e., the deflection, surge frequency, and shear stress. As shown in Figure 10, three design variables are needed to be considered in the optimization design process: the mean coil diameter (*D*), wire diameter (*d*), and the number of active coils (*N*).

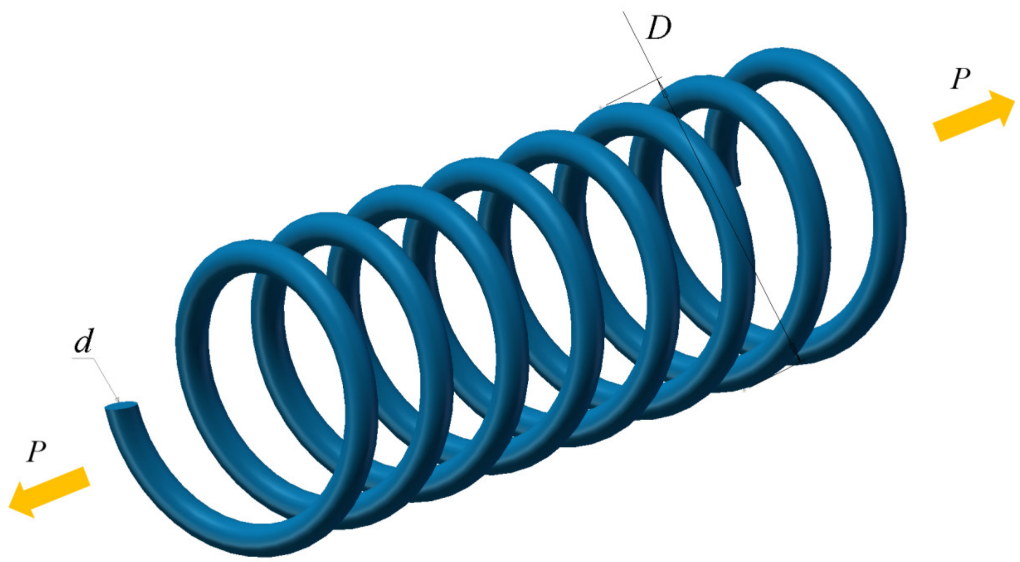


**Figure 10.** Tension/compression spring design problem (three-dimensional model diagram).

The mathematical formulas for this problem can be expressed as follows:
Consider

$$\vec{x} = [x_1, x_2, x_3, x_4] = [d, D, N]$$

Minimize

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

Subject to

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

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0$$

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

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

Variable range

$$0.05 \leq x_1 \leq 2.00$$

$$0.25 \leq x_2 \leq 1.30$$

$$2.00 \leq x_3 \leq 15.00$$

The optimal solution of IROA for this problem is also compared with those of other optimization methods, including AOA [10], MPA [16], PFA [46], WOA [7], MMPA [47], CPSO [44], ISCA [48] and IMFO [49]. The results are listed in Table 12. It can be observed that IROA obtains the minimum weight 0.01019759, and the three corresponding variables are *d* = 0.0500, *D* = 0.3674088, and *N* = 9.10217765. Thus we can conclude that IROA is better than other methods when solving this problem.

**Table 12.** Comparison of optimal solutions for the tension/compression spring design problem.

| Algorithm | Optimal values for variables | | | Optimal weight |
|---|---|---|---|---|
| | *d* | *D* | *N* | |
| IROA | 0.0500 | 0.3674088 | 9.10217765 | 0.01019759 |
| AOA [10] | 0.0500 | 0.349809 | 11.8637 | 0.012124 |
| MPA [16] | 0.051724477 | 0.35757003 | 11.2391955 | 0.012665 |
| PFA [46] | 0.051726 | 0.357629 | 11.235724 | 0.012665 |
| WOA [7] | 0.051207 | 0.345215 | 12.0043032 | 0.0126763 |
| MMPA [47] | 0.05168827 | 0.35669876 | 11.29008064 | 0.01266524 |
| CPSO [44] | 0.051728 | 0.357644 | 11.244543 | 0.012674 |
| ISCA [48] | 0.0520217 | 0.364768 | 10.832300 | 0.012667 |
| IMFO [49] | 0.05159 | 0.354337 | 11.4301 | 0.012666 |

*5.3. The three-bar truss design problem*

This problem is to minimize a three-bar truss's design weight [50]. As shown in Figure 11, two variables should be considered: the cross-sectional area of two bars ($A_1$ and $A_2$).

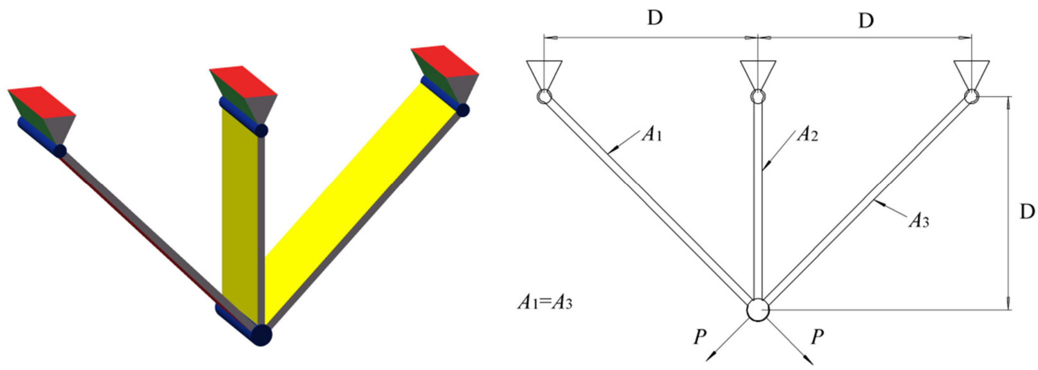


**Figure 11.** Three-bar truss design problem: three-dimensional model diagram (left), structure parameters (right).

The mathematical formulas for this problem can be expressed as follows:
Consider

$$\vec{x} = [x_1, x_2] = [A1, A2]$$

Minimize

$$f(\vec{x}) = (2\sqrt{2}x_1 + x_2)l$$

Subject to

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

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

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

Variable range

$$0 \leq x_1, x_2 \leq 1$$

For this problem, the comparative algorithms are ROA [27], SFO [28], AOA [10], HHO [15], MFO [51], SSA [13], PSO-DE [52], and HSCAHS [38]. Table 13 shows the results of this problem.

IROA outperforms other optimization techniques with lowest weight 263.8526902, and [$A_1$, $A_2$] = [0.78814380, 0.40895198]. Thus IROA is a competitive method compared with other algorithms for solving this design problem.

**Table 13.** Comparison of optimal solutions for the three-bar truss design problem.

| Algorithm | Optimal values for variables | | Optimal weight |
|---|---|---|---|
| | $A_1$ | $A_2$ | |
| IROA | 0.78814380 | 0.40895198 | 263.8526902 |
| ROA [27] | 0.79509685 | 0.38951232 | 263.8845249 |
| SFO [28] | 0.7884562 | 0.40886831 | 263.8959212 |
| AOA [10] | 0.79369 | 0.39426 | 263.9154 |
| HHO [15] | 0.788662816 | 0.40828313 | 263.8958434 |
| MFO [51] | 0.788244771 | 0.409466906 | 263.8959797 |
| SSA [13] | 0.788665414 | 0.408275784 | 263.8958434 |
| PSO-DE [52] | 0.7886751 | 0.4082482 | 263.8958433 |
| HSCAHS [38] | 0.7885721 | 0.4084012 | 263.881992 |

*5.4. The car crashworthiness design problem*

The frequently-used car crashworthiness design problem is considered firstly proposed by Gu et al. [53]. This problem also belongs to a minima problem with eleven variables, subject to ten constraints. Figure 12 shows the finite element model of this problem.

**Figure 12.** Three-dimensional model diagram of the car side crash [54].

The mathematical formulas for this problem can be expressed as follows:
Minimize

$$f(\vec{x}) = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 2.73x_7$$

Subject to

$$g_1(\vec{x}) = 1.16 - 0.3717x_2x_4 - 0.00931x_2x_{10} - 0.484x_3x_9 + 0.01343x_6x_{10} \leq 1$$

$$g_2(\vec{x}) = 0.261 - 0.0159x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 + 0.0144x_3x_5 + 0.0008757x_5x_{10} + 0.080405x_6x_9 + 0.00139x_8x_{11} + 0.00001575x_{10}x_{11} \leq 0.32$$

$$g_3(\vec{x}) = 0.214 + 0.00817x_5 - 0.131x_1x_8 - 0.0704x_1x_9 + 0.03099x_2x_6 - 0.018x_2x_7 + 0.0208x_3x_8 + 0.121x_3x_9 - 0.00364x_5x_6 + 0.0007715x_5x_{10} - 0.0005354x_6x_{10} + 0.00121x_8x_{11} \leq 0.32$$

$$g_4(\vec{x}) = 0.074 - 0.061x_2 - 0.163x_3x_8 + 0.001232x_3x_{10} - 0.166x_7x_9 + 0.227x_2^2 \leq 0.32$$

$$g_5(\vec{x}) = 28.98 + 3.818x_3 - 4.2x_1x_2 + 0.0207x_5x_{10} + 6.63x_6x_9 - 7.7x_7x_8 + 0.32x_9x_{10} \leq 32$$

$$g_6(\vec{x}) = 33.86 + 2.95x_3 + 0.1792x_{10} - 5.057x_1x_2 - 11.0x_2x_8 - 0.0215x_5x_{10} - 9.98x_7x_8 + 22.0x_8x_9 \leq 32$$

$$g_7(\vec{x}) = 46.36 - 9.9x_2 - 12.9x_1x_8 + 0.1107x_3x_{10} \leq 32$$

$$g_8(\vec{x}) = 4.72 - 0.5x_4 - 0.19x_2x_3 - 0.0122x_4x_{10} + 0.009325x_6x_{10} + 0.000191x_{11}^2 \leq 4$$

$$g_9(\vec{x}) = 10.58 - 0.674x_1x_2 - 1.95x_2x_8 + 0.02054x_3x_{10} - 0.0198x_4x_{10} + 0.028x_6x_{10} \leq 9.9$$

$$g_{10}(\vec{x}) = 16.45 - 0.489x_3x_7 - 0.843x_5x_6 + 0.0432x_9x_{10} - 0.0556x_9x_{11} - 0.000786x_{11}^2 \leq 15.7$$

Variable range

$$0.5 \leq x_1 - x_7 \leq 1.5$$

$$x_8, x_9 \in (0.192, 0.345)$$

$$-30 \leq x_{10}, x_{11} \leq 30$$

**Table 14.** Comparison of optimal solutions for the car crashworthiness design problem.

| Algorithm | IROA | ROA [27] | SMA [11] | MPA [16] | WOA [7] | HHOCM [33] | ROLGWO [35] | MALO [37] |
|---|---|---|---|---|---|---|---|---|
| $x_1$ | 0.5 | 0.5 | 0.5 | 0.5 | 0.8521 | 0.5001638 | 0.5012548 | 0.5 |
| $x_2$ | 1.23105 | 1.22942 | 1.22739 | 1.22823 | 1.2136 | 1.2486123 | 1.2455510 | 1.2281 |
| $x_3$ | 0.5 | 0.5 | 0.5 | 0.5 | 0.6604 | 0.6595579 | 0.5000457 | 0.5 |
| $x_4$ | 1.19766 | 1.21197 | 1.20428 | 1.20490 | 1.1156 | 1.0985153 | 1.1802539 | 1.2126 |
| $x_5$ | 0.5 | 0.5 | 0.50000 | 0.5 | 0.5 | 0.7579885 | 0.5000347 | 0.5 |
| $x_6$ | 1.07429 | 1.37798 | 1.04185 | 1.23930 | 1.1950 | 0.7672683 | 1.1658804 | 1.3080 |
| $x_7$ | 0.5 | 0.50005 | 0.5 | 0.5 | 0.5898 | 0.5000551 | 0.5000882 | 0.5 |
| $x_8$ | 0.34499 | 0.34489 | 0.345 | 0.34498 | 0.2711 | 0.3431048 | 0.3448952 | 0.3449 |
| $x_9$ | 0.34432 | 0.19263 | 0.34248 | 0.192 | 0.2769 | 0.1920318 | 0.2995826 | 0.2804 |
| $x_{10}$ | 0.95239 | 0.62239 | 0.29675 | 0.44035 | 4.3437 | 2.8988050 | 3.5950796 | 0.4242 |
| $x_{11}$ | 1.01140 | – | 1.15796 | 1.78504 | 2.2352 | – | 2.2901802 | 4.6565 |
| $f_{min}$ | 23.188937 | 23.235442 | 23.191021 | 23.199822 | 25.836569 | 24.483584 | 23.222427 | 23.229404 |

Table 14 shows the optimal results of IROA and other comparative algorithms, including ROA [27], SMA [11], MPA [16], WOA [7], HHOCM [33], ROLGWO [35], and MALO [37]. In Table 14, the IROA obtains the best solution ($f_{\min}$ = 23.18893698) among these optimization methods. Thus IROA has the merits in solving the car crashworthiness design problem.

*5.4. The tubular column design problem*

Designing a tubular column aims to minimize the cost with compressive load $P$ = 2,500 kgf [55], as shown in Figure 13. The yield stress ($\sigma_y$), modulus of elasticity ($E$), and density ($\rho$) of the column are 500 kgf/cm$^2$, 0.85 × 10$^6$ kgf/cm$^2$, and 0.0025 kgf/cm$^3$, respectively. And the column's length is 250 cm. In addition, the objective function is considered to be the sum of material and manufacturing costs.

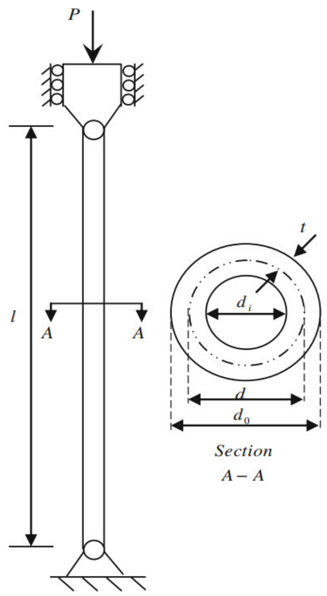


**Figure 13.** The schematic diagram of the tubular column [56].

The mathematical formulas for this problem can be expressed as follows:

Minimize

$$f(d,t) = 9.8dt + 2d$$

Subject to

$$g_1 = \frac{P}{\pi dt\sigma_y} - 1 \leq 0$$

$$g_2 = \frac{8PL^2}{\pi^3 Edt(d^2 + t^2)} - 1 \leq 0$$

$$g_3 = \frac{2.0}{d} - 1 \leq 0$$

$$g_4 = \frac{d}{14} - 1 \leq 0$$

$$g_5 = \frac{0.2}{t} - 1 \leq 0$$

$$g_6 = \frac{t}{0.8} - 1 \leq 0$$

Variable range

$$0.01 \leq d, T \leq 100$$

Table 15 lists the results of IROA and other comparative methods, including ROA [27], SMA [11], MPA [16], WOA [7], HHOCM [33], ROLGWO [35], and MALO [37]. In Table 15, it is observed that IROA obtains the lowest cost (26.531303) with [*d*, *T*] = [5.451154, 0.291965]. Thus, the IROA also can solve this problem very well.

**Table 15.** Comparison of optimal solutions for the tubular column car design problem.

| Algorithm | Optimal values for variables | | Optimal cost |
|---|---|---|---|
| | *d* | *T* | |
| IROA | 5.451154 | 0.291965 | 26.531303 |
| ROA [27] | 5.433671 | 0.294813 | 26.598146 |
| SMA [11] | 5.451212 | 0.291960 | 26.531379 |
| MPA [16] | 5.451389 | 0.291951 | 26.531737 |
| WOA [7] | 5.437032 | 0.294228 | 26.583393 |
| HHOCM [33] | 5.492022 | 0.289790 | 26.613000 |
| ROLGWO [35] | 5.452650 | 0.291894 | 26.534764 |
| MALO [37] | 5.451140 | 0.291967 | 26.531342 |

## 6. Conclusions and future works

This paper presents an improved ROA (IROA) by introducing a new search mechanism named autonomous foraging mechanism (AFM). This mechanism is based on the individual's autonomous foraging behavior. Each remora has a little chance to search the food randomly or based on food position within the determined space. Thus these two cases are mathematically modeled and utilized to improve the basic ROA. The former can enhance the global search of the algorithm, while the latter can boost the local search of the algorithm.

Experimental tests were conducted using twenty-three classical benchmark functions, ten latest CEC 2021 test functions, and five engineering design optimization problems. The results of proposed IROA on these test functions are compared to the basic ROA and other famous optimization methods. It can be found that the IROA is the best optimizer on almost all test functions and practical problems. The proposed IROA also shows very stable performance on high-dimensional test functions, which indicates the effective role of the proposed mechanism. Moreover, it should be noted that the AFM will not increase the computational complexity of the original method and keep the simplicity.

Other MAs may use the proposed mechanism applied in IROA to enhance the search capability in future work. And the IROA also can be implemented on more complex practical optimization problems, such as parameter optimization, data mining, feature selection, and large-scale global optimization problems.

## Acknowledgments

This work was supported by the Sanming University introduces high-level talents to start scientific research funding support project (21YG01, 20YG14), Sanming University National Natural Science Foundation Breeding Project (PYT2103, PYT2105), Fujian Natural Science Foundation Project (2021J011128), Guiding science and technology projects in Sanming City (2021-S-8), Educational research projects of young and middle-aged teachers in Fujian Province (JAT200618), Scientific research and development fund of Sanming University (B202009), and funded by Open Research Fund Program of Fujian Provincial Key Laboratory of Agriculture Internet of Things Application (ZD2101).

## Conflict of interest

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

## References

1. K. Hussain, M. N. Mohd Salleh, S. Cheng, Y. Shi, Metaheuristic research: a comprehensive survey, *Artif. Intell. Rev.*, **52** (2019), 2191–2233. https://doi.org/10.1007/s10462-017-9605-z
2. W. Qiao, H. Moayedi, L. K. Foong, Nature-inspired hybrid techniques of IWO, DA, ES, GA, and ICA, validated through a k-fold validation process predicting monthly natural gas consumption, *Energy Build.*, **217** (2020), 110023. https://doi.org/10.1016/j.enbuild.2020.110023
3. M. Wang, H. Chen, Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis, *Appl. Soft Comput.*, **88** (2020), 105946. https://doi.org/10.1016/j.asoc.2019.105946
4. S. Wang, J. Xiang, Y. Zhong, Y. Zhou, Convolutional neural network-based hidden Markov models for rolling element bearing fault identification, *Knowl.-Based Syst.*, **144** (2018), 65–76. https://doi.org/10.1016/j.knosys.2017.12.027
5. H. Chen, Y. Xu, M. Wang, X. Zhao, A balanced whale optimization algorithm for constrained engineering design problems, *Appl. Math. Model.*, **71** (2019), 45–59. https://doi.org/10.1016/j.apm.2019.02.004
6. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95 - International Conference on Neural Networks*, **4** (1995), 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
7. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Software*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
8. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software*, **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
9. 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. 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
11. S. Li, H. Chen, M. Wang, 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
12. S. Mirjalili, The ant lion optimizer, *Adv. Eng. Software*, **83** (2015), 80–98. https://doi.org/10.1016/j.advengsoft.2015.01.010
13. S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, S. M. Mirjalili, Salp swarm algorithm: a bio-inspired optimizer for engineering design problems, *Adv. Eng. Software*, **114** (2017), 163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002
14. S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimization algorithm: theory and application, *Adv. Eng. Software*, **105** (2017), 30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004
15. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: algorithm and applications, *Future Gener. Comput. Syst.*, **97** (2019), 849–872. https://doi.org/10.1016/j.future.2019.02.028
16. 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
17. L. Abualigah, D. Yousri, M. A. Elaziz, A. A. Ewees, M. A. A. Al-qaness, A. H. Gandomi, Aquila optimizer: a novel meta-heuristic optimization algorithm, *Comput. Ind. Eng.*, **157** (2021), 107250. https://doi.org/10.1016/j.cie.2021.107250
18. B. Abdollahzadeh, F. S. Gharehchopogh, S. Mirjalili, Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems, *Int. J. Intell. Syst.*, **36** (2021), 5887–5958. https://doi.org/10.1002/int.22535
19. B. Abdollahzadeh, F. S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems, *Comput. Ind. Eng.*, **158** (2021), 107408. https://doi.org/10.1016/j.cie.2021.107408
20. I. Naruei, F. Keynia, Wild horse optimizer: a new meta-heuristic algorithm for solving engineering optimization problems, *Eng. Comput.*, 2021. https://doi.org/10.1007/s00366-021-01438-z
21. 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
22. 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
23. 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
24. M. A. A. Al-qaness, A. A. Ewees, M. A. Elaziz, Modified whale optimization algorithm for solving unrelated parallel machine scheduling problems, *Soft Comput.*, **25** (2021), 9545–9557. https://doi.org/10.1007/s00500-021-05889-w
25. A. A. Ewees, M. A.A. Al-qaness, M. A. Elaziz, Enhanced salp swarm algorithm based on firefly algorithm for unrelated parallel machine scheduling with setup times, *Appl. Math. Model.*, **94** (2021), 285–305. https://doi.org/10.1016/j.apm.2021.01.017

26. F. K. Onay, S. B. Aydemir, Chaotic hunger games search optimization algorithm for global optimization and engineering problems, *Math. Comput. Simul.*, **192** (2021), 514–536. https://doi.org/10.1016/j.matcom.2021.09.014
27. H. Jia, X. Peng, C. Lang, Remora optimization algorithm, *Expert Syst. Appl.*, **185** (2021), 115665. https://doi.org/10.1016/j.eswa.2021.115665
28. S. Shadravan, H. R. Naji, V. K. Bardsiri, The sailfish optimizer: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems, *Eng. Appl. Artif. Intell.*, **80** (2019), 20–34. https://doi.org/10.1016/j.engappai.2019.01.001
29. A. A. Heidari, P. Pahlavani, An efficient modified grey wolf optimizer with Lévy Flight for Optimization Tasks, *Appl. Soft Comput.*, **60** (2017), 115–134. https://doi.org/10.1016/j.asoc.2017.06.044
30. M. Molga, C. Smutnicki, Test functions for optimization needs, 2005.
31. R. Zheng, H. Jia, L. Abualigah, Q. Liu, S. Wang, Deep ensemble of slime mold algorithm and arithmetic optimization algorithm for global optimization, *Process*, **9** (2021), 1774. https://doi.org/10.3390/pr9101774
32. 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** (2022), 116158. https://doi.org/10.1016/j.eswa.2021.116158
33. E. H. Houssein, N. Neggaz, M. E. Hosney, W. M. Mohamed, M. Hassaballah, Enhanced Harris hawks optimization with genetic operators for selection chemical descriptors and compounds activities, *Neural Comput. Appl.*, **33** (2021), 13601–13618. https://doi.org/10.1007/s00521-021-05991-y
34. L. Rodríguez, O. Castillo, M. García, J. Soria, A new randomness approach based on sine waves to improve performance in metaheuristic algorithms, *Soft. Comput.*, **24** (2020), 11989–12011. https://doi.org/10.1007/s00500-019-04641-9
35. W. Long, J. Jiao, X. Liang, S. Cai, M. Xu, A random opposition-based learning grey wolf optimizer, *IEEE Access*, **7** (2019), 113810–113825. https://doi.org/10.1109/ACCESS.2019.2934994
36. Y. Li, Y. Zhao, J. Liu, Dynamic sine cosine algorithm for large-scale global optimization problems, *Expert Syst. Appl.*, **173** (2021), 114950. https://doi.org/10.1016/j.eswa.2021.114950
37. S. Wang, K. Sun, W. Zhang, H. Jia, Multilevel thresholding using a modified ant lion optimizer with opposition-based learning for color image segmentation, *Math. Biosci. Eng.*, **18** (2021), 3092–3143. https://doi.org/10.3934/mbe.2021155
38. N. Singh, J. Kaur, Hybridizing sine-cosine algorithm with harmony search strategy for optimization design problems, *Soft Comput.*, **25** (2021), 11053–11075. https://doi.org/10.1007/s00500-021-05841-y
39. S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, *Inf. Sci.*, **180** (2010), 2044–2064. https://doi.org/10.1016/j.ins.2009.12.010
40. E. Theodorsson-Norheim, Friedman and Quade tests: BASIC computer program to perform nonparametric two-way analysis of variance and multiple comparisons on ranks of several related samples, *Comput. Biol. Med.*, **17** (1987), 85–99. https://doi.org/10.1016/0010-4825(87)90003-5
41. C. A. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, *Comput. Ind.*, **41** (2000), 113–127. https://doi.org/10.1016/S0166-3615(99)00046-9

42. 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
43. 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
44. Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, *Eng. Appl. Artif. Intell.*, **20** (2007), 89–99. https://doi.org/10.1016/j.engappai.2006.03.003
45. A. Baykasoğlu, 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
46. 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
47. Q. Fan, H. Huang, Q. Chen, L. Yao, K. Yang, D. Huang, A modified self-adaptive marine predators algorithm: framework and engineering applications, *Eng. Comput.*, (2021). https://doi.org/10.1007/s00366-021-01319-5
48. W. Long, T. Wu, X. Liang, S. Xu, Solving high-dimensional global optimization problems using an improved sine cosine algorithm, *Expert Syst. Appl.*, **123** (2019), 108–126. https://doi.org/10.1016/j.eswa.2018.11.032
49. Y. Li, X. Zhu, J. Liu, An improved moth-flame optimization algorithm for engineering problems, *Symmetry*, **12** (2020), 1234. https://doi.org/10.3390/sym12081234
50. T. Ray, P. Saini, Engineering design optimization using a swarm with an intelligent information sharing among individuals, *Eng. Optim.*, **33** (2001), 735–748. https://doi.org/10.1080/03052150108940941
51. S. Mirjalili, Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, *Knowl.-Based Syst.*, **89** (2015), 228–249. https://doi.org/10.1016/j.knosys.2015.07.006
52. H. Liu, Z. Cai, Y. Wang, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, *Appl. Soft Comput.*, **10** (2010), 629–640. https://doi.org/10.1016/j.asoc.2009.08.031
53. L. Gu, R. J. Yang, C. H. Tho, M. Makowskit, O. Faruquet, Y. Li, Optimisation and robustness for crashworthiness of side impact, *Int. J. Veh. Des.*, **26** (2001), 348–360. https://doi.org/10.1504/IJVD.2001.005210
54. B. S. Yildiz, N. Pholdee, S. Bureerat, A. R. Yildiz, S. M. Sait, Enhanced grasshopper optimization algorithm using elite opposition-based learning for solving real-world engineering problems, *Eng. Comput.*, 2021. https://doi.org/10.1007/s00366-021-01368-w
55. S. S. Rao, *Engineering optimization: Theory and Practice*, 3rd edn. John Wiley & Sons, Chichester, 1996.
56. 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