Mathematical Biosciences and Engineering

MBE, 19(1): 473–512.
DOI: 10.3934/mbe.2022023
Received: 24 September 2021
Accepted: 09 November 2021
Published: 16 November 2021

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

Research article

An improved arithmetic optimization algorithm with forced switching mechanism for global optimization problems

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

[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 Computer Science and Technology, Hainan University, Haikou 570228, China

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

Abstract: Arithmetic optimization algorithm (AOA) is a newly proposed meta-heuristic method which is inspired by the arithmetic operators in mathematics. However, the AOA has the weaknesses of insufficient exploration capability and is likely to fall into local optima. To improve the searching quality of original AOA, this paper presents an improved AOA (IAOA) integrated with proposed forced switching mechanism (FSM). The enhanced algorithm uses the random math optimizer probability (*RMOP*) to increase the population diversity for better global search. And then the forced switching mechanism is introduced into the AOA to help the search agents jump out of the local optima. When the search agents cannot find better positions within a certain number of iterations, the proposed FSM will make them conduct the exploratory behavior. Thus the cases of being trapped into local optima can be avoided effectively. The proposed IAOA is extensively tested by twenty-three classical benchmark functions and ten CEC2020 test functions and compared with the AOA and other well-known optimization algorithms. The experimental results show that the proposed algorithm is superior to other comparative algorithms on most of the test functions. Furthermore, the test results of two training problems of multi-layer perceptron (MLP) and three classical engineering design problems also indicate that the proposed IAOA is highly effective when dealing with real-world problems.

Keywords: arithmetic optimization algorithm; meta-heuristic algorithm; global optimization; exploration and exploitation; high-dimensional optimization problems

1. Introduction

Optimization problems have become more and more complex with the rapid development in various application fields. When dealing with these new optimization problems, traditional optimization methods require overmuch time and costs. In most cases, it is known that strict and exact solutions are not necessary. That's to say, estimate optimal solutions can be acceptable in practice because of the significantly fewer time and costs. Therefore, many optimization algorithms have been proposed in recent decades to solve these non-convex, non-linear restrictions and complex optimization problems [1], and proved to be highly effective for these practical issues.

It is worth mentioning that meta-heuristic algorithms (MAs) have attracted much attention in recent decades [2]. These MAs normally are inspired by creatures or principles in nature, human behaviors, etc. For instance, inspired by the biological evolution in nature, genetic algorithm (GA) [3], genetic programming (GP) [4] and differential evolution (DE) [5] were proposed by scholars. There are also many MAs inspired by the collective or social intelligence of natural biology. Some examples are the particle swarm optimization (PSO) [6], grey wolf optimization (GWO) [7], ant colony optimization (ACO) [8], artificial bee colony (ABC) [9], whale optimization algorithm (WOA) [10], slime mould algorithm (SMA) [11], marine predators algorithm (MPA) [12], ROA [13], etc. Moreover, the physical or chemical principles and human intelligence also can be utilized to form new optimization algorithms. Some typical representatives are the simulated annealing (SA) [14], gravitational search algorithm (GSA) [15], multi-verse optimizer (MVO) [16], heat transfer relation-based optimization algorithm (HTOA) [17], artificial chemical reaction optimization algorithm (ACROA) [18], curved space optimization (CSO) [19], harmony search (HS) [20], teaching learning-based optimization (TLBO) [21,22], social-based algorithm (SBA) [23], etc. With the rapid development of MAs, lots of improved MAs which are embed some mechanisms or strategies have also been proposed. Some representative improved algorithms are the JSWOA [24], EHHO [25], DSCA [26], WQSMA [27], and MMPA [28]. Though there are various optimization algorithms proposed for global optimization problems, the No-Free-Lunch (NFL) theory [29] says that none of the algorithms can solve all the optimization problems. This motivates us to continue our researches. Generally speaking, the performance of original MAs can be further enhanced by adding some effective strategies or mechanisms. Then the improved MAs can solve the optimization problems more effectively.

Despite different origins, it can be concluded that there are five general essential elements for each meta-heuristic algorithm: 1) Generate a set of random initial feasible solutions in a given search area; 2) Preset an evaluation function; 3) Generate new candidate solution according to the position updating formula; 4) Judge whether to accept the new solution; and last 5) Judge whether to terminate the search. As mentioned above, the primary distinction of different MAs is the third step. During the iterations, search agents are supposed to explore the whole solution space as much as possible in the early phase and exploit the area near the optimal location found so far in the later stage. In this way, the local minima can be avoided, and the precision of the solution can be sufficient. Therefore, the balance between exploration and exploitation search is a critical indicator for the performance of MAs.

The arithmetic optimization algorithm (AOA) is a new meta-heuristic algorithm proposed by Abualigah et al. in 2021 [30], which is inspired by four basic arithmetic operators (i.e., Multiplication (M), Division (D), Addition (A), and Subtraction (S)). The AOA has a straightforward framework with few parameters and shows better performance compared to the PSO [6], GWO [7], GSA [15], MFO [31], etc. Also, the AOA is able to deal with some real-world problems. Recently, Manoharan et al. proposed the multi-objective arithmetic optimization Algorithm (MOAOA) for solving the constrained multi-objective optimization problems [32]. In this paper, we focus on the single-objective optimization problems. For more information about the multi-objective optimization problems also can refer to [33]. Abualigah et al. combined the AOA with the differential evolution technique to improve the local search capability [34]. Khatir et al. proposed the improved artificial neural network using arithmetic optimization algorithm (IANN-AOA) to deal with the damage quantification problem in functionally graded material (FGM) plate structures [35].

As a matter of fact, from the position updating formulas of AOA, it can be seen that AOA still has limitations in exploration capability and insufficient balance between exploration and exploitation search due to the simple structure of the math optimizer probability (*MOP*) and math optimizer accelerated (*MOA*). Thus the AOA can be trapped into local optima at times. Therefore, this paper proposes an improved AOA (IAOA) with a forced switching mechanism (FSM) for the global optimization problems. The *MOP* in AOA is modified into *RMOP* to increase population diversity. And the *MOA* in AOA is replaced by a parameter p for a better balance between global and local search. Thus the IAOA still has a relatively simple structure but better performance compared with the original AOA. The proposed IAOA has been tested using twenty-three classical benchmark functions and ten CEC2020 single objective test functions. The performance of IAOA has been evaluated from multiple indexes, like the convergence accuracy, convergence speed, statistical estimation, and runtime result. Furthermore, several representative real-world problems are applied to assess the validity of the proposed method in practice.

The rest of this paper is structured as follows: Section 2 introduces a brief overview of the original AOA. Section 3 presents the forced switching mechanism (PSM) and the proposed IAOA. In Section 4, two sets of standard benchmark functions are utilized to test the performance of IAOA. The experimental results are analyzed and discussed from multiple aspects, including accuracy, stability, convergence speed, and running time. Section 5 shows the capabilities of IAOA in solving real-world problems, including the training of multi-layer perceptron (MLP) and engineering design problems. At last, Section 6 concludes this paper and gives some future research directions.

2. Arithmetic optimization algorithm

The AOA is a new meta-heuristic method proposed in 2021 [30]. As its name implies, four traditional arithmetic operators (i.e., Multiplication operator (M), Division operator (D), Addition operator (A), and Subtraction operator (S)) are modeled into the position updating equations for searching the global optimization solution. According to the different effects of these four arithmetic operators, the Multiplication (M) and Division (D) are used for the exploration search, producing large step in the search space. And the Addition (A), and Subtraction (S)) are applied to execute the exploitation search, which is able to generate small step sizes in the search space. The detailed optimization mechanism of the AOA is presented in Figure 1.

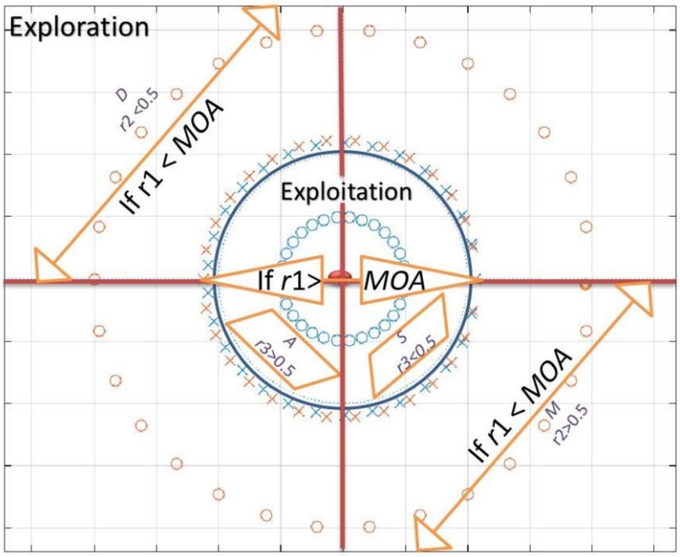

Figure 1. Position updating mechanism of search agents in AOA and effects of *MOA* on it.

According to the Figure 1, the mathematical equations of exploration and exploitation behaviors are expressed by following formulas:

$$X_i(t+1)=\begin{cases} X_b(t)\div(MOP+eps)\times((UB-LB)\times\mu+LB), & rand<0.5 \\ X_b(t)\times MOP\times((UB-LB)\times\mu+LB), & rand\geq 0.5 \end{cases} \tag{1}$$

$$X_i(t+1)=\begin{cases} X_b(t)-MOP\times((UB-LB)\times\mu+LB), & rand<0.5 \\ X_b(t)+MOP\times((UB-LB)\times\mu+LB), & rand\geq 0.5 \end{cases} \tag{2}$$

where $X_i(t+1)$ denotes the newly generated position. And $X_b(t)$ is the best position found by search agents in the tth iteration. *eps* is a very small positive number to ensure the dividend is positive. μ is a constant coefficient. *UB* and *LB* are the upper boundary and lower boundary, respectively. *rand* is a random number uniformly distributed between 0 and 1.

The math optimizer probability (*MOP*) is a vital coefficient that is non-linearly decreased from 1 to 0 along with the iterations. And the calculation expression of *MOP* is as follows:

$$MOP=1-(\frac{t}{T})^{1/\alpha} \tag{3}$$

where T is the maximum number of iterations. α is a constant value, which is set as 5 in AOA.

In addition, another important parameter in AOA is the math optimizer accelerated (*MOA*), which is utilized to balance exploration and exploitation. The *MOA* is calculated by:

$$MOA(t)=Min+t\times(\frac{Max-Min}{T}) \tag{4}$$

where *Min* and *Max* are the minimum and maximum values of *MOA*.

When the AOA starts working, if a random number (between 0 and 1) is bigger than MOA, then the exploration search will be selected and performed, i.e., Multiplication (M) or Division (D). Otherwise the exploitation search will be conducted, i.e., Addition (A) or Subtraction (S). As the number of iterations increases, the search agents will be more likely to perform local searches. The pseudo-code of AOA is shown in Algorithm 1.

Algorithm 1 The pseudo-code of original arithmetic optimization algorithm

```
Initialization
  Initialize the population size (N) and the number of iterations (T)
  Initialize the positions of all search agents X_i (I = 1, 2, 3, …, N)
  Evaluate the fitness of search agents and find the current best position and bestFitness, X_b
  Set the parameters α, μ, Min and Max
  Main loop{
  While (t ≤ T)
    Calculate the MOP by Eq (3)
    Calculate the MOA by Eq (4)
    For each search agent
      If rand > MOA
        Update position by Eq (1)
      Else
        Update position by Eq (2)
      End if
    Calculate the fitness of search agent
    Update current best position and bestFitness, X_b
    End for
    t = t + 1
  End While}
Return bestFitness, X_b
```

3. Proposed IAOA

As described previously, AOA has a straightforward framework with only a few parameters. However, the formulas of position updating lack randomness for the search agents. Hence AOA has limited exploration capability and is easy to fall into local minima. To overcome this problem, this paper modifies the original AOA from two aspects, which are introduced in the following sections.

3.1. Random math optimizer probability (RMOP)

The coefficient MOP in AOA is gradually decreased with the iterations, which lacks randomness. Thus, to increase the range of newly generated positions, the parameter α is calculated by following:

$$\alpha = 10 \times rand - 1 \tag{5}$$

According to the Eq (5), the range of α is [−1, 9]. Then the new *MOP* still can be calculated by Eq (3), which is named as random math optimizer probability (*RMOP*) here. As shown in Figure 2, the *RMOP* presents a relatively random change nearby the original *MOP*. Note that, the *RMOP* could be negative when α is negative, which is not displayed in Figure 2. Hence the diversity of the generated search agents can be effectively increased by using the *RMOP*. To some extent, the capability of local optima avoidance is enhanced.

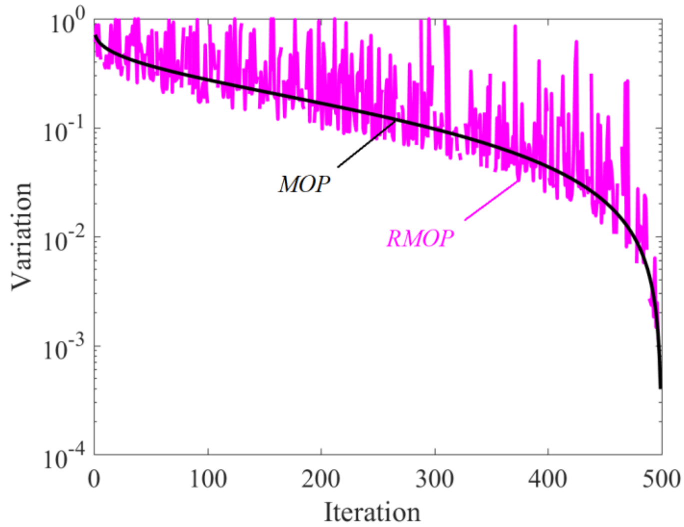

Figure 2. Trend comparison between *RMOP* and *MOP*.

3.2. *Forced switching mechanism (FSM)*

It can be seen from AOA that the *MOA* plays a vital role on the balance of exploration and exploitation. There are more chances for the search agents to execute Division (*D*) operator or Multiplication (*M*) operator in the preliminary stage. That is to say; the whole search space can be detected. At the later stage, Subtraction (*S*) or Addition (*A*) is expected to conduct a local search near the best location found so far. According to the extensive research of scholars in this field, the transformation method between exploration and exploitation is worth studying and beneficial to improving the algorithm's performance. Therefore, this work proposes a kind of forced switching mechanism (FSM) to enhance the AOA's optimization capability effectively.

Firstly, a probability parameter p is substituted for the *MOA*, which is defined as follows:

$$p = \tanh\left|rand \times \left(\frac{F(i) - bF}{F(i) + bF}\right)\right| \tag{6}$$

where $F(i)$ is the fitness of ith search agent, and bF is the best fitness found by so far.

From Eq (6), it can be known that the range of p is [0, 0.7616], which is related to the fitness of the current search agent, best fitness at present, and a random value (between 0 and 1). Using the new parameter p, if a random number (between 0 and 1) is smaller than the current p, the current search agent may be far from the best position. Thus D operator or M operator will be carried out for global search. Otherwise, the operator S or operator A should be performed for local search.

Secondly, the best position could be a local optima, and then the search agents may not be able to jump out of this point with the local search operator, which can only generate small steps. Hence a forced switchover is introduced here to make the search agent conduct the exploration behavior, i.e., D or M. To be specific, this switching process is realized by using a counter. Each search agent owns a counter. This counter will increase by one when the search agent cannot find a better position in one iteration. Otherwise, it will be reset to 0. When the counter exceeds a limited value, the parameter p will be set to 1 for this search agent, and the counter corresponding to the search agent is reset to 0. In summary, the proposed FSM is realized by using the probability parameter p and counter, which help the AOA avoid the local optima and promote the balance between global and local search. At last, the flowchart and pseudo-code of the proposed IAOA are depicted in Algorithm 2 and Figure 3, respectively.

3.3. The computational complexity of IAOA

Algorithm 2 The pseudo-code of the improved arithmetic optimization algorithm

```
Initialization
    Initialize the population size (N) and the number of iterations (T)
    Initialize the positions of all search agents X_i (i = 1, 2, 3, …, N)
    Evaluate the fitness of search agents and find the current best position and bestFitness, X_b
    Set the parameters μ
    Main loop{
    While (t ≤ T)
        Calculate RMOP by Eq (3) and (5)
        Calculate p by Eq (6)
        If trial(i) > Limit
            trial(i) = 0
            p = 1
        End if
        For each search agent
            If rand < p
                Update position by Eq (1)
            Else
                Update position by Eq (2)
            End if
        Calculate the fitness of the search agent
        Update current best position and bestFitness, X_b
        End for
        t = t + 1
    End While}
Return bestFitness, X_b
```

The complexity of the optimization algorithm is also an essential aspect concerning the performance. Typically, low complexity means that the computer can produce results faster and saving time. In the AOA, the computational complexity of initializing search agents is $O(N \times D)$, where N is the population size and D is the dimension of the problem. Then during the iterations, the computational complexity of calculating MOA and MOP is $O(2T)$. The computational complexity of updating new positions is $O(N \times D \times T)$, where T is the maximum number of iterations. Therefore, the computational complexity of AOA is $O(N \times (D \times T + D) + 2T)$.

In the same way, the computational complexity of IAOA in the initialization phase is $O(N \times D)$. In the iterations, the computational complexity of calculating p and $RMOP$ is $O(2N \times T)$. The computational complexity of updating new positions is $O(N \times D \times T)$. Thus the computational complexity of IAOA is $O(N \times (D \times T + D + 2T))$. It is noted that the computational complexity of IAOA is only slightly higher than that of the original AOA. In other words, the computational complexity of the AOA and IAOA can be considered at a similar level.

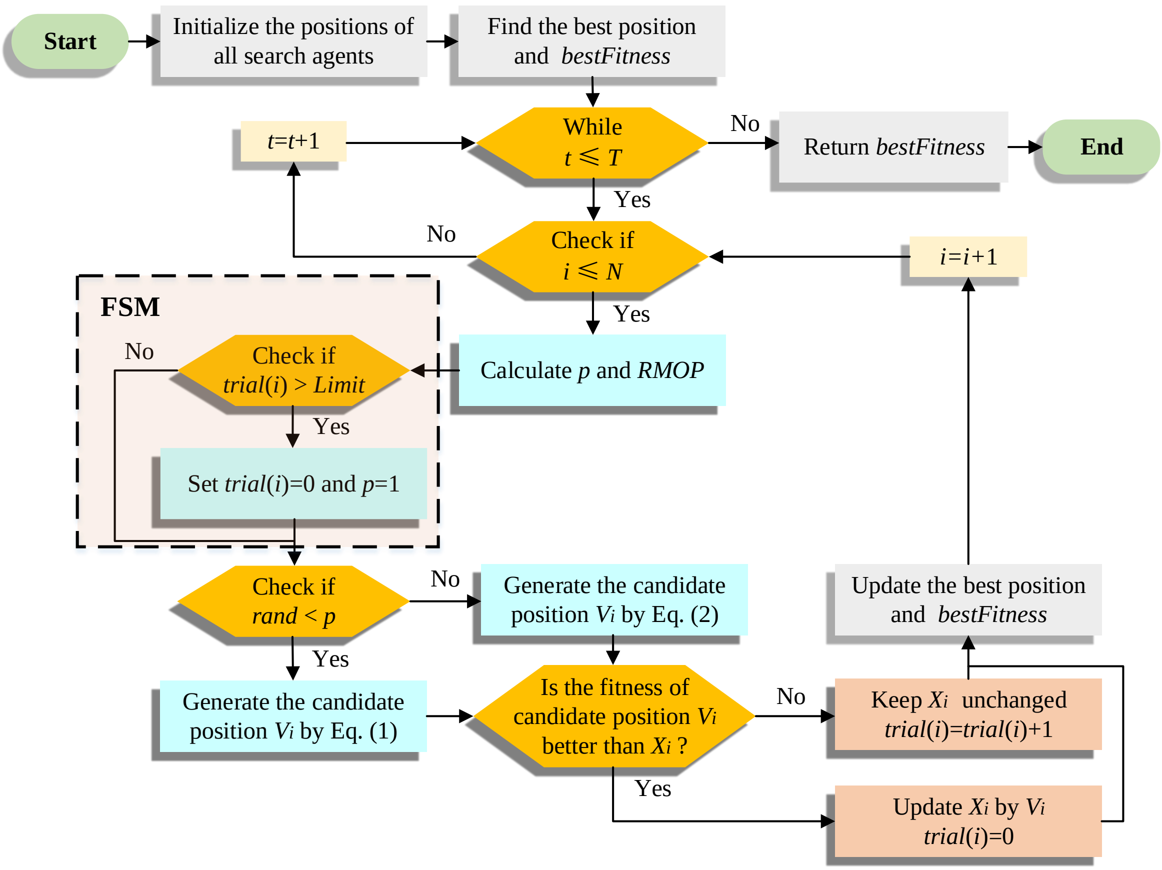

Figure 3. The flowchart of the proposed IAOA.

4. Experimental tests and analysis

In this section, the performance of proposed IAOA is evaluated by using twenty-three classical benchmark functions and ten CEC2020 test functions [36,37]. The detailed information of test functions is introduced at first. Then seven meta-heuristic algorithms and six modified algorithms are employed to illustrate the outstanding performance of the proposed improved algorithm. The convergence precision of algorithms is analyzed from three aspects, including the best value, mean value, and standard deviation. The best value is the best solution obtained in a certain number of tests, which is set as 30 in this paper. The mean value is the mean solution obtained from these tests. And the standard deviation (Std) is used to reflect the degree of dispersion of optimal solutions. In addition, statistical methods like Wilcoxon signed-rank test [38] and Friedman ranking test [39] are used to confirm the significant differences between the IAOA and other algorithms. The convergence curves are used for the visual description of the optimization effect. Furthermore, the running time of these comparative algorithms on high-dimensional functions has been analyzed and compared.

4.1. Experimental settings

The detailed information of twenty-three classical benchmark functions and ten CEC2020 test functions are listed in Table 1. For the classical standard benchmark functions, F1−F7 belong to the unimodal test functions, which means there is only one extreme optimization point within the given space. Thus, they can usually be used to evaluate the exploitation precision of algorithms. For the F8−F13, multiple local extreme points exist in the given space. Hence, these functions can be used to reveal the exploration capability of given algorithms and see whether the algorithms are able to jump out of the local point and find the global optima. On the F14−F23, similarly, there are many local optima points which may trap the optimization algorithm. However, the dimension of these functions is determinate. Therefore, they can be used to test the stability of the algorithm. It is worth mentioning that the dimension for the first 13 functions can be set as required. Thus, to see the performance of the proposed algorithm on high-dimensional functions, the F1−F13 on high dimensions (200/500/1000) are also considered for the tests.

For the CEC 2020 test functions, the first four functions are called as shifted and rotated bent cigar function, shifted and rotated schwefel’s function, shifted and rotated lunacek bi-rastrigin function, and expanded rosenbrock’s plus griewangk’s function, respectively. Functions CEC_05−CEC_07 belong to the hybrid functions. Functions CEC_08−CEC_10 are the composition functions.

Moreover, to fully illustrate the performance of proposed algorithm, seven meta-heuristic algorithms and six modified algorithms are employed for the comparisons. These meta-heuristic methods are particle swarm optimization (PSO) [6], sine cosine algorithm (SCA) [40], grey wolf optimizer (GWO) [7], whale optimization algorithm (WOA) [10], salp swarm algorithm (SSA) [41], multi-verse optimizer (MVO) [16] and the original arithmetic optimization algorithm (AOA) [30]. The parameter settings of these meta-heuristic algorithms and proposed IAOA are given in Table 2. Modified algorithms include four improved algorithms (DSCA [26], MALO [42], ROL-GWO [43], RL-WOA [44]) and two hybrid algorithms (DESMAOA [45] and HSMSSA [46]). The parameters of these enhanced algorithms are kept the same as those in the original papers.

For a proper comparison, the number of iterations and population size are set to 500 and 30, respectively. Each algorithm was independently performed 30 times to obtain reliable statistical results.

Table 1. Properties of test functions (D indicates the dimension).

Function type	Function	Dimensions	Range	Theoretical optimization value
Unimodal test functions	F1	30/200/500/1000	[−100, 100]	0
	F2	30/200/500/1000	[−10, 10]	0
	F3	30/200/500/1000	[−100, 100]	0
	F4	30/200/500/1000	[−100, 100]	0
	F5	30/200/500/1000	[−30, 30]	0
	F6	30/200/500/1000	[−100, 100]	0
	F7	30/200/500/1000	[−1.28, 1.28]	0
Multimodal test functions	F8	30/200/500/1000	[−500, 500]	$-418.9829 \times D$
	F9	30/200/500/1000	[−5.12, 5.12]	0
	F10	30/200/500/1000	[−32, 32]	0
	F11	30/200/500/1000	[−600, 600]	0
	F12	30/200/500/1000	[−50, 50]	0
	F13	30/200/500/1000	[−50, 50]	0
Fixed-dimension multimodal test functions	F14	2	[−65, 65]	1
	F15	4	[−5, 5]	0.00030
	F16	2	[−5, 5]	−1.0316
	F17	2	[−5, 5]	0.398
	F18	2	[−2, 2]	3
	F19	3	[−1, 2]	−3.86
	F20	6	[0, 1]	−3.32
	F21	4	[0, 10]	−10.1532
	F22	4	[0, 10]	−10.4028
	F23	4	[0, 10]	−10.5363
CEC2020 single objective test functions	CEC_01	10	[−100, 100]	100
	CEC_02	10	[−100, 100]	1100
	CEC_03	10	[−100, 100]	700
	CEC_04	10	[−100, 100]	1900
	CEC_05	10	[−100, 100]	1700
	CEC_06	10	[−100, 100]	1600
	CEC_07	10	[−100, 100]	2100
	CEC_08	10	[−100, 100]	2200
	CEC_09	10	[−100, 100]	2400
	CEC_10	10	[−100, 100]	2500

Table 2. Parameter values for the IAOA and comparative algorithms.

Algorithm	Parameters
PSO [6]	$c_1 = 2$; $c_2 = 2$; $W \in [0.2, 0.9]$; $vMax = 6$
SCA [29]	$a = 2$
GWO [7]	$a = [2, 0]$
WOA [10]	$a_1 = [2, 0]$; $a_2 = [-2, -1]$; $b = 1$
SSA [30]	$c_1 \in [0, 1]$; $c_2 \in [0,1]$
MVO [15]	$WEP \in [0.2, 1]$; $TDR \in [0, 1]$; $r_1, r_2, r_3 \in [0, 1]$
AOA [24]	$\alpha = 5$; $\mu = 0.499$; $Min = 0.2$; $Max = 0.9$
IAOA	$\alpha \in [-1, 9]$; $\mu = 0.499$; $Limit = 4$

4.2. *Comparison between IAOA and AOA*

Table 3. Comparison results of IAOA and AOA for low-dimensional benchmark functions (F1−F23).

Function	Algorithm	Best	Mean	Std
F1	AOA	2.5004E−06	4.60E−06	2.56E−06
	IAOA	**0**	**0**	**0**
F2	AOA	0.00022713	0.0016222	0.0018213
	IAOA	**0**	**0**	**0**
F3	AOA	0.0001484	0.00088135	0.00066246
	IAOA	**0**	**0**	**0**
F4	AOA	0.007372	0.02121	0.012793
	IAOA	**0**	**0**	**0**
F5	AOA	27.5561	28.0037	0.17927
	IAOA	**26.4703**	**27.9405**	**0.20653**
F6	AOA	2.7821	3.0021	0.23491
	IAOA	**0.00015408**	**0.00067796**	**1.94E−04**
F7	AOA	4.1456E−05	8.64E−05	6.48E−05
	IAOA	**1.6121E−05**	**0.000072876**	**7.61E−05**
F8	AOA	−6626.6943	−5522.0895	360.536
	IAOA	**−9410.4345**	**−7439.9702**	**781.4275**
F9	AOA	7.5374E−11	1.6679E−06	1.33E−06
	IAOA	**0**	**0**	**0**
F10	AOA	2.7335E−05	0.0004501	0.00016232
	IAOA	**8.8818E−16**	**8.8818E−16**	**0**
F11	AOA	**1.9763E−05**	**0.00084498**	**3.39E−03**
	IAOA	7.5898E−05	0.012704	0.013443
F12	AOA	0.71309	0.73513	0.032107
	IAOA	**6.1974E−06**	**0.000017862**	**3.87E−06**
F13	AOA	2.9097	2.9547	3.16E−02
	IAOA	**0.001474**	**0.069295**	**0.092459**
F14	AOA	2.9821	11.3532	2.8202
	IAOA	**0.998**	**2.1227**	**8.92E−01**
F15	AOA	0.00031509	0.0085249	0.017144
	IAOA	**0.00031542**	**0.00067023**	**0.00073262**
F16	AOA	**−1.0316**	**−1.0316**	**1.94E−11**
	IAOA	**−1.0316**	**−1.0316**	**6.91E−11**
F17	AOA	0.39789	0.43067	0.11645
	IAOA	**0.39789**	**0.39789**	**1.69E−11**
F18	AOA	3	9.3	16.9037
	IAOA	**3**	**3**	**3.66E−10**

Continued on next page

Function	Algorithm	Best	Mean	Std
F19	AOA	−3.8628	−3.7416	0.53653
	IAOA	**−3.8628**	**−3.8627**	**2.36E−04**
F20	AOA	−3.322	−3.2769	0.060402
	IAOA	**−3.322**	**−3.2863**	**0.055431**
F21	AOA	−10.1523	−7.6411	3.0153
	IAOA	**−10.153**	**−10.1527**	**3.65E−04**
F22	AOA	−10.4026	−8.2302	2.9844
	IAOA	**−10.4028**	**−10.4025**	**0.00035433**
F23	AOA	−10.5362	−7.7591	3.5771
	IAOA	**−10.5362**	**−10.5359**	**0.00030993**

Table 4. Comparison results of IAOA and AOA for high-dimensional benchmark functions (F1−F13) with D = 200.

Function	Algorithm	Best	Mean	Std
F1	AOA	0.030732	0.050867	0.013644
	IAOA	**0**	**0**	**0**
F2	AOA	0.060429	0.074017	0.017986
	IAOA	**0**	**0**	**0**
F3	AOA	0.65955	0.7687	0.15925
	IAOA	**0**	**0**	**0**
F4	AOA	0.080176	0.089516	0.0087835
	IAOA	**0**	**0**	**0**
F5	AOA	196.8073	198.1746	0.086362
	IAOA	**197.0503**	**197.2712**	**0.091721**
F6	AOA	34.2895	36.2072	0.90543
	IAOA	**1.0199**	**1.5483**	**0.11734**
F7	AOA	1.5615E−05	9.22E−05	5.87E−05
	IAOA	**1.1075e−05**	**8.16E−05**	**8.40E−05**
F8	AOA	−23070.4723	−21944.2025	921.0585
	IAOA	**−51673.0898**	**−42731.4775**	**2868.4316**
F9	AOA	0.0010863	0.0012932	0.00013706
	IAOA	**0**	**0**	**0**
F10	AOA	0.008908	0.010165	0.00096921
	IAOA	**8.8818E−16**	**8.8818E−16**	**0**
F11	AOA	1.1995	15.478	21.346
	IAOA	**0.073417**	**0.11825**	**0.015644**
F12	AOA	0.76327	0.82012	0.039459
	IAOA	**0.0030856**	**0.0038509**	**3.94E−04**
F13	AOA	19.5125	19.6503	0.12166
	IAOA	**11.156**	**19.1919**	**1.5575**

In this section, the performances of the proposed IAOA and original AOA are analyzed. The experimental results, including the best value, the mean value, and the standard deviation, are shown in Tables 3−6. The better optimal results are in bold in Table 3, which presents the low-dimensional cases (D < 100), it is easy to observe that IAOA outperforms AOA on almost all test functions. Mainly, IAOA has obtained the theoretical optimal value (0) on F1−F4, F9 and F18. And on F10, F15−17, F19−F23, the results obtained by IAOA are very close to the theoretical optimal values. These results demonstrate that the IAOA has sufficient exploration and exploitation capabilities on the stand benchmark functions. Also, from the standard deviation results, IAOA has better solution stability than AOA. Moreover, From Tables 4−6, it can be seen that the IAOA also has better performance than AOA on most of the high-dimensional test functions (D = 200/500/1000). In particular, the IAOA can still obtain the theoretical optimal value on F1−F4 and F9. Though the IAOA and AOA have roughly the same results on F5, F7 and F13, the IAOA wins the other test functions with significant advantages.

Table 5. Comparison results of IAOA and AOA for high-dimensional benchmark functions (F1−F13) with D = 500.

Function	Algorithm	Best	Mean	Std
F1	AOA	0.44645	5.36E−01	3.13E−02
	IAOA	**0**	**0**	**0**
F2	AOA	0.38441	0.51533	0.10741
	IAOA	**0**	**0**	**0**
F3	AOA	4.7158	6.7191	1.1378
	IAOA	**0**	**0**	**0**
F4	AOA	0.11225	0.12264	0.0064185
	IAOA	**0**	**0**	**0**
F5	AOA	499.3258	499.4464	0.2153
	IAOA	**495.8477**	**496.9015**	**0.15112**
F6	AOA	110.72	111.9802	2.0405
	IAOA	**16.6868**	**19.3039**	**0.8841**
F7	AOA	**9.588E−06**	**7.79E−05**	**5.94E−05**
	IAOA	9.0697E−06	1.17E−04	1.06E−04
F8	AOA	−38280.3287	−37592.7969	1696.3911
	IAOA	**−122145.3363**	**−107095.6718**	**6479.4553**
F9	AOA	0.0097635	1.09E−02	7.55E−04
	IAOA	**0**	**0**	**0**
F10	AOA	0.02492	0.026755	0.00079171
	IAOA	**8.8818E−16**	**8.8818E−16**	**0**
F11	AOA	956.8539	1360.0908	332.7481
	IAOA	**1.8905**	**2.5489**	**0.3317**
F12	AOA	0.89778	0.93086	0.021968
	IAOA	**0.032073**	**0.032479**	**0.002024**
F13	AOA	48.6264	49.6914	0.11261
	IAOA	**48.789**	**49.2815**	**0.27971**

Table 6. Comparison results of IAOA and AOA for high-dimensional benchmark functions (F1−F13) with D = 1000.

Function	Algorithm	Best	Mean	Std
F1	AOA	1.4035	1.4851	0.050689
	IAOA	**0**	**0**	**0**
F2	AOA	1.4705	1.6004	0.088484
	IAOA	**0**	**0**	**0**
F3	AOA	28.2473	31.21	6.0176
	IAOA	**0**	**0**	**0**
F4	AOA	0.14234	0.15659	0.012054
	IAOA	**0**	**0**	**0**
F5	AOA	1002.1444	1002.5848	0.23481
	IAOA	**996.1367**	**997.1556**	**0.16621**
F6	AOA	241.1288	242.1669	1.2738
	IAOA	**74.6488**	**85.9905**	**2.8304**
F7	AOA	1.1702E−05	0.0001096	1.29E−04
	IAOA	**3.5362E−07**	**7.98E−05**	**8.94E−05**
F8	AOA	−55338.4958	−54939.0341	1846.1281
	IAOA	**−215430.9145**	**−203575.8261**	**8913.3037**
F9	AOA	0.035031	0.038004	0.001746
	IAOA	**0**	**0**	**0**
F10	AOA	0.031972	0.033101	0.00068333
	IAOA	**8.8818E−16**	**8.8818E−16**	**0**
F11	AOA	11375.8334	14711.0439	2582.4096
	IAOA	**90.9442**	**139.4703**	**15.4775**
F12	AOA	1.01	1.0307	0.016899
	IAOA	**0.090605**	**0.091028**	**0.0029325**
F13	AOA	100.0347	100.2574	0.3491
	IAOA	**99.6654**	**99.7497**	**0.070909**

Furthermore, the results of statistical analysis between IAOA and AOA are shown in Tables 7 and 8. For the Wilcoxon signed-rank test, when the *p-value* is lower than 0.05, it is believed that there are significant differences between the two algorithms. The term "+ / = / –" indicates that the IAOA performs better, similar and worse than the AOA, respectively. From Table 7, it can be seen that the IAOA has better statistical results than AOA. The overall result (+ / = / –) is 20/2/1. For the high-dimensional results in Table 8, the overall results are 11/2/0 (D = 200), 12/1/0 (D = 500), 13/0/0 (D = 1000), respectively. Therefore, the IAOA also has obvious advantages over AOA on high-dimensional conditions.

Table 9 shows the solutions of CEC2020 obtained by the IAOA and AOA. It can be seen that the proposed IAOA outperforms AOA for all test functions except CEC_04. Both algorithms have achieved the theoretical optimal value (1900) with zero standard deviation on CEC_04. Therefore, the performance of AOA is also enhanced by the proposed method when solving the CEC2020 functions.

Table 7. Result of Wilcoxon signed-rank test between IAOA and AOA on low-dimensional benchmark functions (F1−F23).

Function	*p-value*	+/=/–	Function	*p-value*	+/=/–
F1	6.10E−05	+	F13	6.10E−05	+
F2	6.10E−05	+	F14	6.10E−05	+
F3	6.10E−05	+	F15	6.10E−05	+
F4	6.10E−05	+	F16	0.97797	=
F5	0.00018311	+	F17	6.10E−05	+
F6	6.10E−05	+	F18	0.047913	+
F7	0.072998	=	F19	0.04126	+
F8	6.10E−05	+	F20	0.015076	+
F9	6.10E−05	+	F21	6.10E−05	+
F10	6.10E−05	+	F22	6.10E−05	+
F11	0.0042725	–	F23	6.10E−05	+
F12	6.10E−05	+	**Overall (+/=/–)**		**20/2/1**

Table 8. Result of Wilcoxon signed-rank test between IAOA and AOA on high-dimensional benchmark functions (F1−F13).

Function	*p-value* (D = 200)	+/=/–	*p-value* (D = 500)	+/=/–	*p-value* (D = 1000)	+/=/–
F1	6.10E−05	+	6.10E−05	+	6.10E−05	+
F2	6.10E−05	+	6.10E−05	+	6.10E−05	+
F3	6.10E−05	+	6.10E−05	+	6.10E−05	+
F4	6.10E−05	+	6.10E−05	+	6.10E−05	+
F5	6.10E−05	+	6.10E−05	+	6.10E−05	+
F6	6.10E−05	+	6.10E−05	+	6.10E−05	+
F7	0.97797	=	0.71973	=	6.10E−05	+
F8	6.10E−05	+	6.10E−05	+	6.10E−05	+
F9	6.10E−05	+	6.10E−05	+	6.10E−05	+
F10	6.10E−05	+	6.10E−05	+	6.10E−05	+
F11	6.10E−05	+	6.10E−05	+	6.10E−05	+
F12	6.10E−05	+	6.10E−05	+	6.10E−05	+
F13	0.63867	=	0.000122	+	6.10E−05	+
Overall (+/=/–)		**11/2/0**		**12/1/0**		**13/0/0**

Table 9. Comparison results of IAOA and AOA for CEC2020 test functions (CEC_01−CEC_10).

Function	Algorithm	Best	Mean	Std
CEC_01	AOA	1.05E+10	1.52E+10	5.75E+09
	IAOA	**101.1321**	**1837.7653**	**2097.8393**
CEC_02	AOA	1.91E+03	2429.0929	236.9499
	IAOA	**1.88E+03**	**2130.5016**	**294.519**
CEC_03	AOA	785.2938	806.9479	17.8589
	IAOA	**779.1130**	**800.8397**	**9.6216**
CEC_04	AOA	**1900**	**1900**	**0**
	IAOA	**1900**	**1900**	**0**
CEC_05	AOA	3.99E+05	4.53E+05	1.15E+05
	IAOA	**1.36E+04**	**17729.2735**	**7896.0719**
CEC_06	AOA	1.94E+03	2122.9517	189.2305
	IAOA	**1.60E+03**	**1951.6078**	**178.7839**
CEC_07	AOA	7.51E+03	2.37E+06	3.24E+06
	IAOA	**2.74E+03**	**10484.477**	**6273.1236**
CEC_08	AOA	3.18E+03	3529.2538	353.3169
	IAOA	**2.21E+03**	**2325.4096**	**15.6135**
CEC_09	AOA	2.75E+03	2923.6263	118.2572
	IAOA	**2.50E+03**	**2706.7551**	**161.7538**
CEC_10	AOA	3.33E+03	3680.1652	363.6474
	IAOA	**2.89E+03**	**2942.7115**	**22.7046**

4.3. Comparison with meta-heuristic algorithms

4.3.1. Numerical analysis

In this section, the IAOA is compared to six well-known algorithms (PSO, SCA, GWO, WOA, SSA, and MVO) to illustrate its superiority in solving optimization problems. In these experiments, the results of test functions (F1–F13) in different dimensions (D = 30/200/500/1000) are presented in Tables 10 and 11 (Inf means infinity, and NaN means not a number), and the results of fixed-dimensional test functions (F14–F23) are listed in Table 12. The optimal results obtained by these algorithms are ranked according to the Friedman ranking test for statistical analysis. At the end of each table, each algorithm's average rank and overall rank are given for comparison.

In Table 10, the proposed IAOA has shown outstanding exploitation capability and is ranked the first in these four dimensions. According to the results shown in Table 11, the IAOA is ranked the first in 30 dimensions and the second in 200, 500, and 1000 dimensions. The WOA is found to be good at cases in high dimensions. However, it is worth noting that the gap between the IAOA and WOA is very small. Thus, the IAOA still shows equivalent or better performance than these algorithms, demonstrating the superiority of the IAOA in exploration search. Moreover, for the fixed-dimension conditions listed in Table 12, the IAOA is ranked the first again, indicating it has the most stable performance in solving optimization problems that contain multiple local best points.

Table 10. Results of the IAOA and competitor algorithms on unimodal benchmark functions (F1−F7) in different dimensions.

Function	*D*	Metric	PSO	SCA	GWO	WOA	SSA	MVO	IAOA
F1	30	Mean	1.77E−04	12.0582	6.00E−28	7.61E−73	1.64E−07	1.1527	0
		Std	2.46E−04	18.9524	6.17E−28	4.06E−72	1.61E−07	0.40056	0
	200	Mean	3.20E+02	5.72E+04	9.44E−08	6.24E−67	1.78E+04	2.84E+03	0
		Std	4.17E+01	2.55E+04	5.87E−08	3.41E−66	3.34E+03	2.78E+02	0
	500	Mean	5.95E+03	2.14E+05	1.71E−03	3.29E−70	9.64E+04	1.20E+05	0
		Std	4.06E+02	7.95E+04	6.51E−04	1.18E−69	5.80E+03	8.34E+03	0
	1000	Mean	4.14E+04	4.98E+05	2.42E−01	2.65E−70	2.37E+05	8.06E+05	0
		Std	1.89E+03	1.60E+05	3.62E−02	1.17E−69	1.17E+04	2.91E+04	0
F2	30	Mean	8.7029	0.028685	1.01E−16	4.51E−51	2.878	5.7703	0
		Std	9.7312	0.077316	5.29E−17	1.67E−50	1.8304	23.301	0
	200	Mean	4.82E+02	4.00E+01	3.04E−05	1.06E−48	1.56E+02	4.68E+75	0
		Std	5.43E+01	1.83E+01	6.96E−06	3.39E−48	1.02E+01	2.54E+76	0
	500	Mean	1.21E+114	9.77E+01	1.08E−02	6.84E−50	5.40E+02	3.78E+218	0
		Std	6.61E+114	4.11E+01	2.13E−03	2.64E−49	1.58E+01	Inf	0
	1000	Mean	1.41E+03	Inf	7.29E−01	2.97E−48	1.19E+03	1.07E+279	0
		Std	5.62E+01	NaN	5.00E−01	9.51E−48	2.34E+01	Inf	0
F3	30	Mean	81.660	8746.812	1.11E−05	47394.26	1615.589	203.770	0
		Std	26.584	6887.170	1.81E−05	12471.12	957.4707	87.693	0
	200	Mean	9.33E+04	9.89E+05	2.15E+04	4.71E+06	2.29E+05	3.28E+05	0
		Std	2.31E+04	1.85E+05	1.29E+04	1.90E+06	1.09E+05	2.39E+04	0
	500	Mean	5.55E+05	7.32E+06	3.17E+05	3.12E+07	1.43E+06	2.06E+06	0
		Std	1.38E+05	2.10E+06	8.93E+04	1.06E+07	6.20E+05	1.45E+05	0
	1000	Mean	2.43E+06	2.72E+07	1.50E+06	1.24E+08	5.08E+06	8.18E+06	0
		Std	5.98E+05	5.20E+06	2.34E+05	4.58E+07	2.12E+06	7.95E+05	0

Continued on next page

Function	*D*	Metric	PSO	SCA	GWO	WOA	SSA	MVO	IAOA
F4	30	Mean	1.0826	37.477	1.12E−06	45.6252	10.9918	1.9625	0
		Std	0.20039	13.4806	1.59E−06	27.8305	3.8408	0.74426	0
	200	Mean	1.99E+01	9.66E+01	2.75E+01	8.14E+01	3.39E+01	8.23E+01	0
		Std	1.72E+00	1.15E+00	7.84E+00	2.02E+01	3.76E+00	4.13E+00	0
	500	Mean	2.75E+01	9.90E+01	6.36E+01	7.81E+01	4.03E+01	9.44E+01	0
		Std	1.78E+00	3.31E−01	5.24E+00	2.27E+01	2.85E+00	1.02E+00	0
	1000	Mean	3.28E+01	9.95E+01	7.91E+01	7.37E+01	4.57E+01	9.80E+01	0
		Std	1.61E+00	1.20E−01	3.82E+00	2.03E+01	3.03E+00	5.17E−01	0
F5	30	Mean	90.6001	34130.1646	27.178	27.8903	257.0443	415.2657	27.9405
		Std	78.9547	68213.466	0.76439	0.45169	366.6366	403.9315	0.20653
	200	Mean	6.26E+05	5.28E+08	1.98E+02	1.98E+02	3.92E+06	4.08E+05	1.97E+02
		Std	1.61E+05	1.50E+08	4.37E−01	1.73E−01	8.56E+05	1.07E+05	9.17E−02
	500	Mean	2.95E+07	1.88E+09	4.98E+02	4.96E+02	3.81E+07	1.66E+08	4.97E+02
		Std	3.65E+06	4.64E+08	2.43E−01	4.45E−01	5.93E+06	2.68E+07	1.51E−01
	1000	Mean	2.83E+08	4.18E+09	1.05E+03	9.94E+02	1.17E+08	2.31E+09	9.97E+02
		Std	2.76E+07	8.37E+08	3.62E+01	9.65E−01	1.13E+07	1.94E+08	1.66E−01
F6	30	Mean	2.11E−04	18.0959	7.39E−01	3.86E−01	6.35E−07	1.4574	6.78E−04
		Std	2.22E−04	24.935	2.82E−01	2.48E−01	1.57E−06	0.54375	1.94E−04
	200	Mean	3.37E+02	5.57E+04	2.86E+01	1.03E+01	1.69E+04	2.73E+03	1.55E+00
		Std	4.09E+01	2.41E+04	1.40E+00	2.59E+00	2.20E+03	2.35E+02	1.17E−01
	500	Mean	6.07E+03	2.04E+05	9.19E+01	3.32E+01	9.49E+04	1.17E+05	1.93E+01
		Std	4.78E+02	6.01E+04	2.30E+00	7.46E+00	5.42E+03	9.32E+03	8.84E−01
	1000	Mean	4.14E+04	4.84E+05	2.03E+02	6.16E+01	2.34E+05	7.96E+05	8.60E+01
		Std	2.09E+03	1.62E+05	2.72E+00	1.92E+01	1.01E+04	3.38E+04	2.83E+00

Continued on next page

Function	*D*	Metric	PSO	SCA	GWO	WOA	SSA	MVO	IAOA
F7	30	Mean	4.3234	0.11836	1.83E−03	0.003127	0.17734	0.034628	8.64E−05
		Std	5.9292	0.14894	1.15E−03	0.002478	0.083019	0.011627	7.61E−05
	200	Mean	2.88E+03	1.56E+03	1.72E−02	4.05E−03	1.87E+01	5.79E+00	8.16E−05
		Std	5.54E+02	4.61E+02	7.50E−03	3.94E−03	3.83E+00	1.10E+00	8.40E−05
	500	Mean	4.65E+04	1.51E+04	4.77E−02	5.22E−03	2.82E+02	1.18E+03	1.17E−04
		Std	7.47E+03	3.55E+03	1.46E−02	6.55E−03	4.47E+01	1.55E+02	1.06E−04
	1000	Mean	2.44E+05	6.82E+04	1.54E−01	3.75E−03	1.75E+03	2.91E+04	1.10E−04
		Std	5.21E+03	1.42E+04	3.09E−02	3.50E−03	1.75E+02	2.92E+03	1.29E−04
Rank	30	Mean	4.43	6.00	2.57	3.86	4.43	5.14	**1.57**
		Overall	4.5	7	2	3	4.5	6	**1**
	200	Mean	4.43	6.29	2.79	3.21	5.14	5.14	**1.00**
		Overall	4	6	2	3	5.5	5.5	**1**
	500	Mean	4.29	6.29	3.00	3.00	4.43	5.86	**1.14**
		Overall	4	7	2.5	2.5	5	6	**1**
	1000	Mean	4.43	6.43	3.14	2.71	4.00	6.00	**1.29**
		Overall	5	7	3	2	4	6	**1**

Table 11. Results of the IAOA and competitor algorithms on multimodal benchmark functions (F8−F13) in different dimensions.

Function	*D*	Metric	PSO	SCA	GWO	WOA	SSA	MVO	IAOA
F8	30	Mean	−4766.131	−3706.60	−5970.25	−10352	−7329.73	−7626.415	−7439.97
		Std	1212.293	257.870	490.4865	1719.311	704.6803	891.2681	781.4275
	200	Mean	−1.54E+04	−9.82E+03	−2.80E+04	−6.88E+04	−3.49E+04	−4.02E+04	−4.27E+04
		Std	5.54E+03	7.04E+02	5.79E+03	1.20E+04	3.36E+03	1.69E+03	2.87E+03
	500	Mean	−2.37E+04	−1.54E+04	−5.59E+04	−1.79E+05	−6.01E+04	−7.44E+04	−1.07E+05
		Std	1.06E+04	1.11E+03	1.23E+04	2.78E+04	4.35E+03	3.42E+03	6.48E+03
	1000	Mean	−3.33E+04	−2.19E+04	−8.97E+04	−3.75E+05	−8.71E+04	−1.10E+05	−2.04E+05
		Std	1.26E+04	1.88E+03	1.45E+04	5.13E+04	6.18E+03	4.92E+03	8.91E+03

Continued on next page

Function	D	Metric	PSO	SCA	GWO	WOA	SSA	MVO	IAOA
F9	30	Mean	106.223	37.346	3.2501	3.79E−15	53.396	123.4134	0
		Std	27.5388	32.867	5.3909	1.44E−14	19.5082	26.1493	0
	200	Mean	2.00E+03	5.56E+02	2.72E+01	0	8.18E+02	1.91E+03	0
		Std	1.09E+02	2.07E+02	1.33E+01	0	7.05E+01	1.08E+02	0
	500	Mean	6.32E+03	1.38E+03	8.19E+01	1.21E−13	3.16E+03	6.40E+03	0
		Std	2.32E+02	6.15E+02	2.95E+01	4.61E−13	1.33E+02	1.57E+02	0
	1000	Mean	1.42E+04	2.27E+03	2.11E+02	0	7.64E+03	1.46E+04	0
		Std	2.76E+02	1.11E+03	5.78E+01	0	1.64E+02	2.48E+02	0
F10	30	Mean	0.23486	15.758	1.07E−13	4.44E−15	2.4873	2.3346	8.88E−16
		Std	0.41349	7.3301	1.6E−14	2.29E−15	0.64328	3.3287	0
	200	Mean	6.47E+00	1.79E+01	2.44E−05	4.56E−15	1.32E+01	2.01E+01	8.88E−16
		Std	2.67E−01	4.85E+00	7.68E−06	2.72E−15	5.77E−01	1.13E+00	0
	500	Mean	1.20E+01	1.87E+01	1.96E−03	5.27E−15	1.43E+01	2.08E+01	8.88E−16
		Std	4.18E−01	3.99E+00	2.80E−04	2.59E−15	2.26E−01	4.26E−02	0
	1000	Mean	1.61E+01	1.96E+01	1.83E−02	3.85E−15	1.45E+01	2.10E+01	8.88E−16
		Std	3.30E−01	3.27E+00	3.12E−03	2.65E−15	1.99E−01	2.08E−02	0
F11	30	Mean	0.0077965	1.1825	0.004534	0.0116	0.014301	0.85111	0.012704
		Std	0.011867	0.96313	0.008074	0.04538	0.011089	0.081176	0.013443
	200	Mean	1.48E+00	5.08E+02	8.19E−03	7.40E−18	1.54E+02	2.65E+01	1.18E−01
		Std	9.69E−01	2.25E+02	1.56E−02	2.82E−17	1.95E+01	3.19E+00	1.56E−02
	500	Mean	8.01E+01	1.92E+03	1.55E−02	0	8.58E+02	1.08E+03	2.55E+00
		Std	1.21E+01	5.60E+02	3.50E−02	0	6.03E+01	6.44E+01	3.32E−01
	1000	Mean	2.75E+02	4.19E+03	7.86E−02	0	2.12E+03	7.23E+03	1.39E+02
		Std	1.80E+01	1.34E+03	9.48E−02	0	1.01E+02	2.70E+02	1.55E+01

Continued on next page

Function	*D*	Metric	PSO	SCA	GWO	WOA	SSA	MVO	IAOA
F12	30	Mean	0.010369	499751.82	0.037732	0.029178	7.9944	2.3492	1.79E−05
		Std	0.031632	1964831.1	0.019255	0.027961	5.752	1.8399	3.87E−06
	200	Mean	3.37E+01	1.53E+09	5.38E−01	5.58E−02	1.09E+04	2.03E+03	3.85E−03
		Std	1.66E+01	3.74E+08	6.00E−02	2.20E−02	1.67E+04	1.94E+03	3.94E−04
	500	Mean	2.72E+05	5.60E+09	7.67E−01	8.27E−02	1.49E+06	1.70E+08	3.25E−02
		Std	1.47E+05	1.29E+09	6.10E−02	4.34E−02	1.02E+06	4.40E+07	2.02E−03
	1000	Mean	9.23E+06	1.34E+10	1.24E+00	1.05E−01	1.07E+07	4.38E+09	9.10E−02
		Std	2.33E+06	2.31E+09	3.95E−01	4.46E−02	2.42E+06	5.47E+08	2.93E−03
F13	30	Mean	0.0058413	59947.169	0.67424	0.54857	16.0953	0.18537	0.069295
		Std	0.0067476	238303.2	0.22145	0.28627	12.8362	0.10839	0.092459
	200	Mean	5.41E+03	2.66E+09	1.69E+01	6.40E+00	1.69E+06	1.18E+05	1.92E+01
		Std	3.46E+03	6.77E+08	4.84E−01	1.85E+00	6.99E+05	7.47E+04	1.56E+00
	500	Mean	3.89E+06	9.41E+09	5.08E+01	1.87E+01	3.77E+07	4.86E+08	4.97E+01
		Std	9.91E+05	2.51E+09	1.55E+00	6.68E+00	1.12E+07	1.01E+08	1.13E−01
	1000	Mean	8.52E+07	2.11E+10	1.21E+02	3.90E+01	1.48E+08	9.25E+09	9.97E+01
		Std	1.10E+07	3.86E+09	7.20E+00	1.41E+01	2.81E+07	1.02E+09	7.09E−02
Rank	30	Mean	3.50	6.50	3.50	2.50	5.33	4.67	**2.00**
		Overall	3.5	7	3.5	2	6	5	**1**
	200	Mean	4.67	6.33	3.33	**1.42**	5.33	5.17	1.75
		Overall	4	7	3	**1**	6	5	2
	500	Mean	4.67	6.33	3.17	**1.50**	4.83	5.83	1.67
		Overall	4	7	3	**1**	5	6	2
	1000	Mean	4.83	6.17	3.00	**1.42**	4.83	6.00	1.75
		Overall	4.5	7	3	**1**	4.5	6	2

Table 12. Results of the IAOA and competitor algorithms on fixed-dimension multimodal benchmark functions (F14−F23).

Function	Metric	PSO	SCA	GWO	WOA	SSA	MVO	IAOA
F14	Mean	3.2971	2.054	4.1971	2.2112	1.1635	0.998	2.1227
	Std	2.6924	1.9047	4.1247	2.4847	0.45784	3.60E−11	0.89234
F15	Mean	0.005002	0.0010075	0.003131	0.000936	0.002149	0.0080634	0.00067
	Std	0.0080173	0.0003976	0.00688	0.000608	0.004958	0.013454	0.000733
F16	Mean	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
	Std	6.32E−16	5.92E−05	2.49E−08	4.44E−09	3.25E−14	2.55E−07	6.91E−11
F17	Mean	0.39789	0.39913	0.39789	0.39789	0.39789	0.39789	0.39789
	Std	0	0.0010657	1.27E−06	1.57E−05	1.36E−14	1.46E−07	1.69E−11
F18	Mean	5.7	3.0001	3	3	3	5.7	3
	Std	14.7885	9.24E−05	5.29E−05	6.02E−05	1.92E−13	14.7885	3.66E−10
F19	Mean	−3.8628	−3.8512	−3.8602	−3.8336	−3.8628	−3.8628	−3.8627
	Std	2.67E−15	0.0076497	3.96E−03	1.41E−01	8.13E−12	2.12E−06	2.36E−04
F20	Mean	−3.1929	−2.8874	−3.2775	−3.2633	−3.2197	−3.2694	−3.2863
	Std	0.13959	0.32874	0.066099	0.076027	0.059376	0.061248	0.055431
F21	Mean	−8.0558	−1.7248	−8.6368	−8.6106	−7.5665	−6.9692	−10.1527
	Std	2.8803	1.5226	2.6132	2.6127	3.5062	3.3512	0.000365
F22	Mean	−9.079	−2.813	−9.9704	−6.8597	−8.314	−8.3245	−10.4025
	Std	2.7639	1.8216	1.6688	3.2997	3.067	3.0645	0.000354
F23	Mean	−10.0968	−3.9225	−9.9938	−7.4358	−8.6014	−8.7548	−10.5359
	Std	1.6938	1.5503	2.0583	3.2114	3.3155	2.8283	0.00031
Rank	Mean	4.3	5.6	3.6	4.3	3.8	4.1	**2.3**
	Overall	5.5	7	2	5.5	3	4	**1**

4.3.2. Convergence analysis

Figures 4−8 present the convergence speed of the IAOA and other well-known algorithms on the classical test functions over the course of the iterations. In each figure, six representative functions were presented. From Figures 4−7, the proposed IAOA displays obviously faster convergence performance than other algorithms on the test function F1, even in the cases of high dimensions. IAOA obtains the theoretical optimal value within less than 100 iterations. Also, from the F6 and F12, it can be seen that the IAOA is able to converge throughout the iteration continuously. This indicates that the RMOP and FSM proposed in this paper enhance the exploration space of the original AOA. Although sometimes the IAOA is beat by SSA, WOA and other algorithms, overall the IAOA is more likely to find a better solution. In Figure 8, the convergence curves of F14, F15 and F17 show the IAOA can find the optimal positions in a quick time in the case of multiple local points. The results of F20, F22, and F23 show that the IAOA has experienced several local optimal positions. However, with the help of the proposed mechanism (i.e., FSM), the IAOA can jump out of the local points within limited iterations and then find better solutions later. In particular, the IAOA also has achieved better or comparative accuracy of the solution compared with other optimization algorithms.

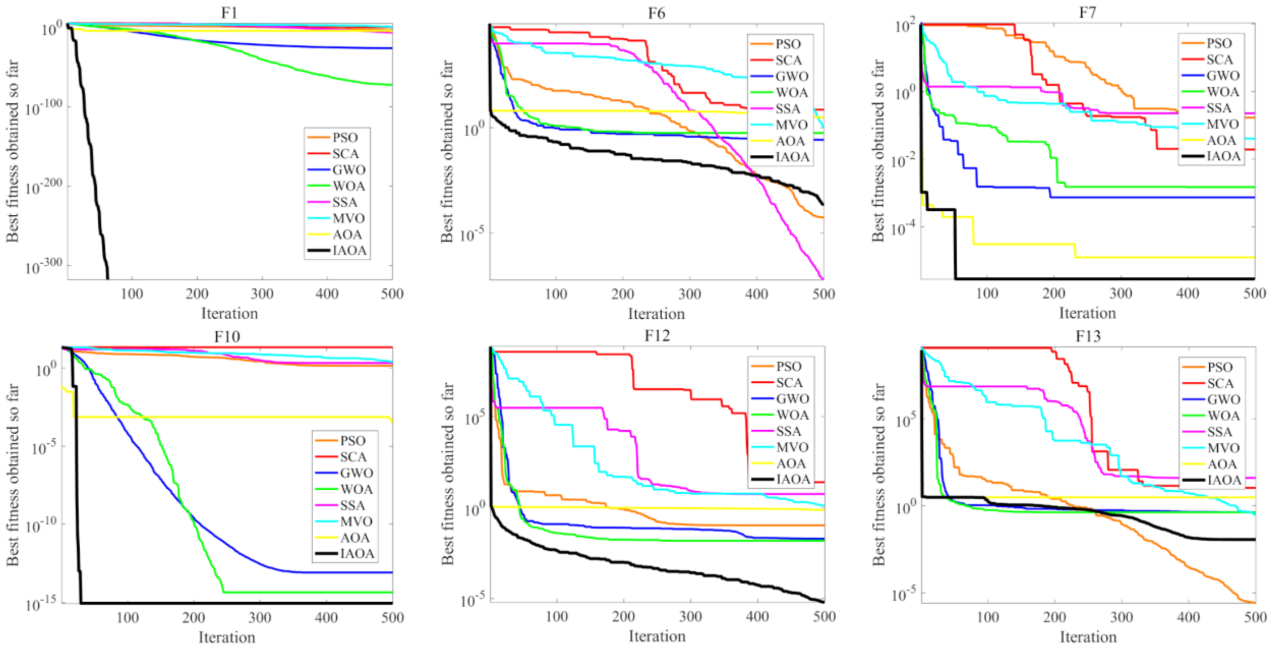

Figure 4. The convergence curves for the optimization algorithms on test functions (F1, F6, F7, F10, F12, F13) with $D = 30$.

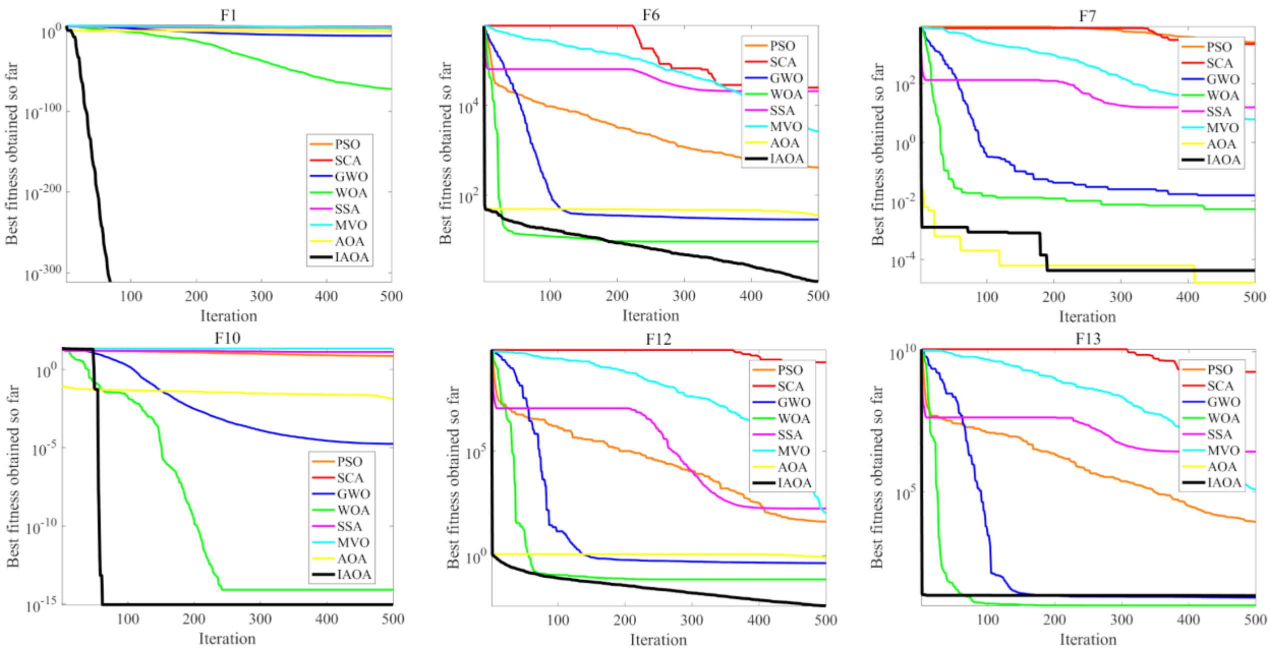

Figure 5. The convergence curves for the optimization algorithms on test functions (F1, F6, F7, F10, F12, F13) with $D = 200$.

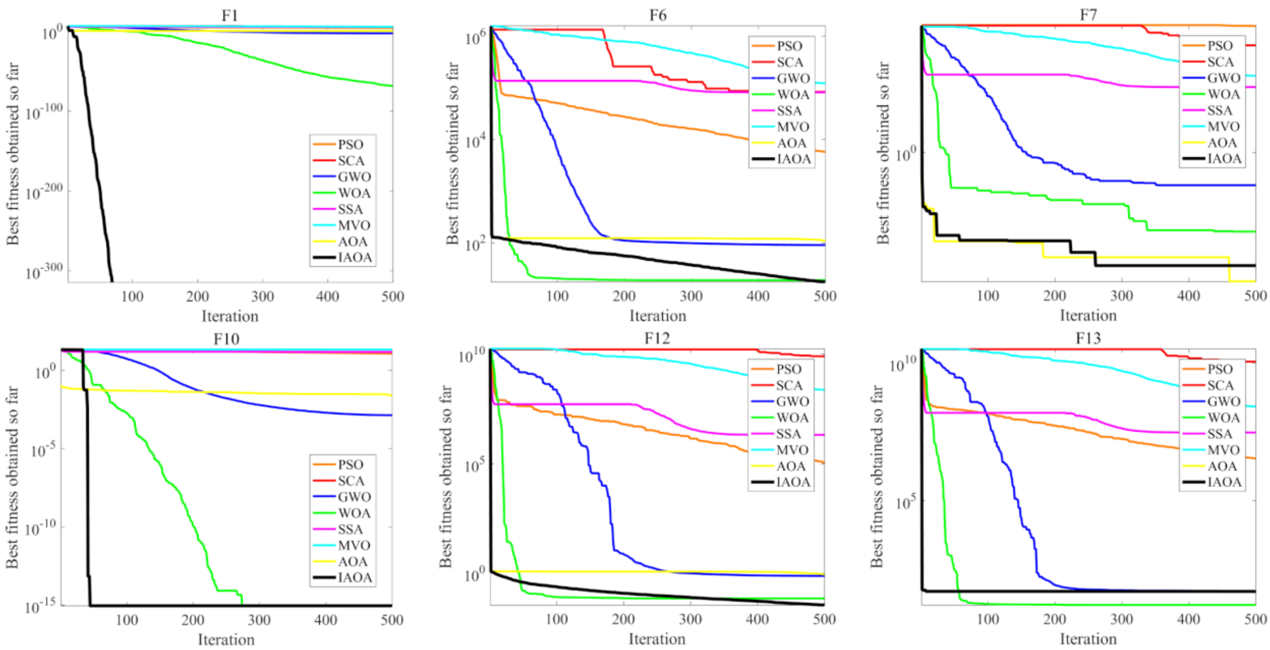

Figure 6. The convergence curves for the optimization algorithms on test functions (F1, F6, F7, F10, F12, F13) with $D = 500$.

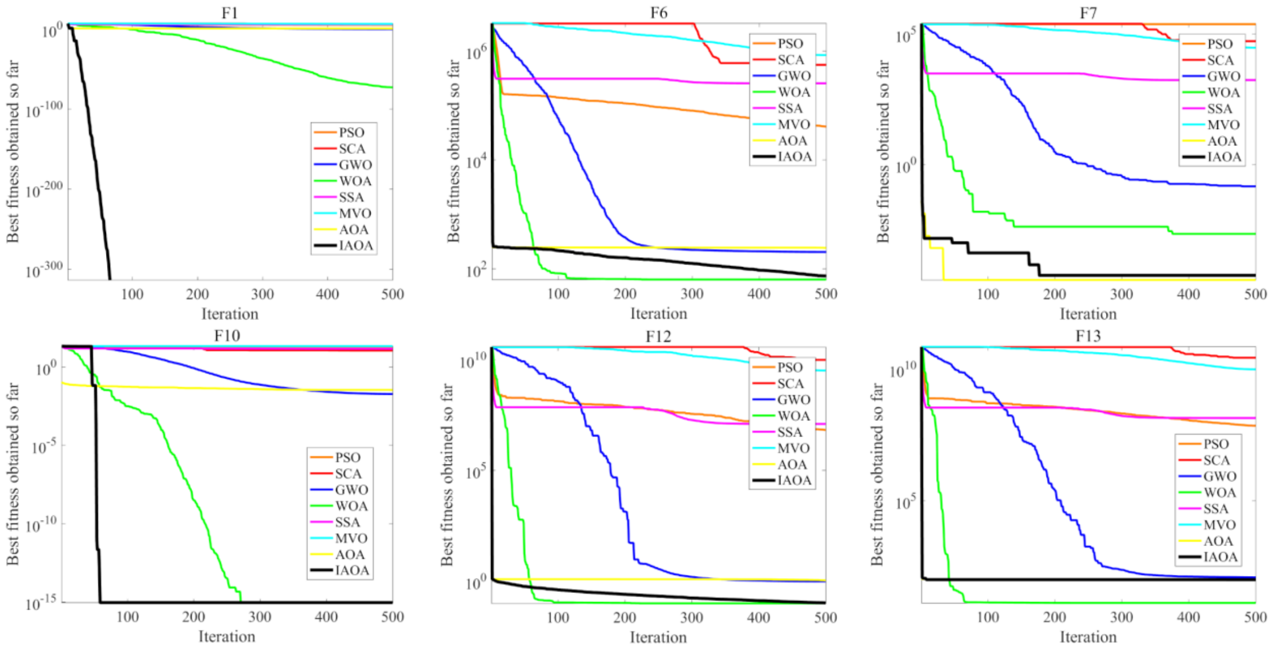

Figure 7. The convergence curves for the optimization algorithms on test functions (F1, F6, F7, F10, F12, F13) with $D = 1000$.

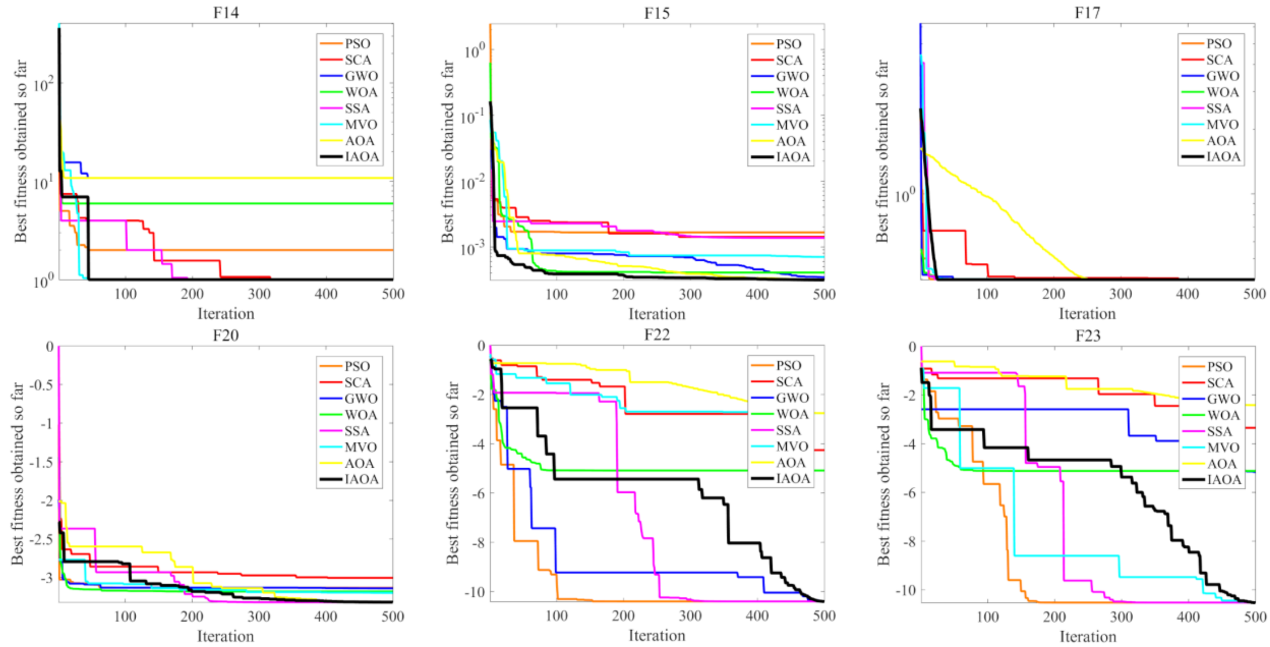

Figure 8. The convergence curves for the optimization algorithms on fixed-dimension test functions (F14, F15, F17, F20, F22, F23).

4.3.3. Analysis of running time

In this section, the time-consuming experiments of optimization algorithms are performed on 13 high-dimensional test functions. The dimension is set to 1000. Thus the MAXFEs is 1.5×10^7. All participants run ten times on each test function independently. The results of the average running time of IAOA and other comparative algorithms are listed in Table 13. The ranking-based Friedman test is employed to investigate the overall level of IAOA compared to others. Table 13 shows that the IAOA is ranked the third place, which is behind the AOA and SSA. Although the computational complexities of AOA and IAOA are almost the same, the original AOA still is obviously better overall. However, it can be seen that the average run time of IAOA is very close to that of AOA on most of the functions. Thus the effect of the proposed forced switching mechanism on running time can be ignored to some extent.

Table 13. Results of average running time (seconds) over 30 independent runs for F1–F13 with D = 1000.

Function	Metric	PSO	SCA	GWO	WOA	SSA	MVO	AOA	IAOA
F1	Ave	1.73	2.14	2.74	2.10	1.49	4.06	1.34	1.47
	Rank	4	6	7	5	3	8	1	2
F2	Ave	1.68	2.14	2.77	2.14	1.53	1.75	1.35	1.48
	Rank	4	6.5	8	6.5	3	5	1	2
F3	Ave	27.35	27.26	28.78	27.33	26.82	29.66	27.37	27.55
	Rank	4	2	7	3	2	8	5	6
F4	Ave	1.71	2.19	2.82	2.24	1.54	4.25	1.37	1.53
	Rank	4	5	7	6	3	8	1	2
F5	Ave	1.83	2.29	2.93	2.48	1.68	4.49	1.50	1.61
	Rank	4	5	7	6	3	8	1	2
F6	Ave	1.85	2.29	2.92	2.29	1.65	4.39	1.42	1.57
	Rank	4	5.5	7	5.5	3	8	1	2
F7	Ave	2.87	3.73	4.06	3.45	2.75	5.52	2.80	3.08
	Rank	3	6	7	5	1	8	2	4
F8	Ave	2.54	2.91	3.63	2.90	2.26	2.56	2.26	2.52
	Rank	4	7	8	6	1.5	5	1.5	3
F9	Ave	2.21	2.62	3.25	2.55	2.06	4.80	1.96	2.13
	Rank	4	6	7	5	2	8	1	3
F10	Ave	2.31	2.85	3.28	2.59	2.13	4.87	2.06	2.21
	Rank	4	6	7	5	2	8	1	3
F11	Ave	2.45	2.87	3.43	2.74	2.22	5.02	2.40	2.48
	Rank	3	6	7	5	1	8	2	4
F12	Ave	4.48	5.00	5.66	4.99	4.34	7.08	4.31	4.53
	Rank	3	6	7	5	2	8	1	4
F13	Ave	4.47	4.97	5.52	4.98	4.35	7.04	4.18	4.45
	Rank	4	5	7	6	2	8	1	3
Mean rank		3.77	5.54	7.15	5.31	2.19	7.54	**1.50**	3.08
Final rank		4	6	7	5	2	8	**1**	3

4.4. *Comparison with modified algorithms*

In this section, six modified algorithms (i.e., DSCA, MALO, ROL-GWO, RL-WOA, DESMAOA and HSMSSA.) are employed to further investigate the performance of proposed IAOA for solving the optimization problems. As shown in Table 14, eight test functions (i.e., F1−F4 and F7−F10) are selected from the twenty-three classical test functions for the comparison. As we can know that the higher the dimension, the more difficult for the algorithm to find the optimal solution. Hence the dimension of test functions is set to 1000 for better comparison. From the Table 14, it can be observed that the IAOA is able to obtain the best solutions on test functions F1−F4, F9 and F10 and comparative solutions on test functions F7 and F8. Based on the ranking-based Friedman test, the IAOA achieves the third place, which is behind the DESMAOA and HSMSSA. However, it should be noted that the DESMAOA and HSMSSA belong to the hybrid algorithms which have higher computational complexity. To sum up, the proposed IAOA still has very comparative performance compared to these modified algorithms in solving high-dimensional optimization problems.

5. Results of real-world problems

The performance of IAOA in solving practical problems is evaluated on two training problems of multi-layer perceptron (MLP) and three classical engineering design problems. The optimization problem in MLP can be regarded as a large-scale global optimization problem [26]. When optimizing the MLP, the objective function is the mean square error (MSE) [47]. In this works, the XOR dataset and Cancer dataset are selected. The population sizes of the algorithms for the XOR dataset and Cancer dataset are set as 50 and 200, respectively. And the maximum number of iterations is set to 250 for these algorithms. To achieve a credible result, ten times independent runs are conducted. And then the statistical results can be obtained and analyzed. When solving the engineering design problems, the population size is set as 30 and the maximum number of iterations is 500. Three engineering design problems, namely, the three-bar truss design problem, pressure vessel design problem and tension/compression spring design problem, are chosen to clarify the effectiveness of the proposed algorithm. The detailed results are described as follows.

5.1. Training of MLP

5.1.1. XOR classification problem

The XOR dataset is a simple dataset containing eight training/test samples, three attributes and two classes [47]. As shown in Table 15, the quality of training MLP is evaluated using six indexes, i.e., the best value, worst value, mean value, standard deviation, classification rate and rank. The statistical results of PSO, SCA, GWO, WOA, SSA, MVO, AOA, and IAOA are also presented in Table 15. It can be seen that PSO, MVO, and IAOA obtain 100 percent accuracy for this dataset and rank the first. It is also worth noting that AOA ranks sixth with only 23.75 percent accuracy. Thus the AOA is significantly enhanced with the strategies proposed in this paper.

In addition, the convergence and the ANOVA graphs of these algorithms are shown in Figures 9 and 10, severally. From Figure 9, the IAOA can converge continuously and finally achieve the relatively low value of MSE, which contributes to high classification rate. According to Figure 10, IAOA, PSO and MVO have better results of variance values.

Table 14. Comparison results of IAOA and other modified algorithms on test functions (F1−F4, F7−F10) with D = 1000.

Function	Metric	DSCA	MALO	ROL-GWO	RL-WOA	DESMAOA	HSMSSA	IAOA
F1	Mean	6.83E−278	6.05E+05	2.47E−323	8.54E−116	0	0	0
	Std	0	1.11E+05	0	3.91E−115	0	0	0
F2	Mean	Inf	Inf	7.87E−165	5.81E−72	0	9.24E+00	0
	Std	NaN	NaN	0	2.83E−71	0	1.65E+01	0
F3	Mean	3.18E−110	8.48E+06	2.48E−317	8.92E−20	0	0	0
	Std	1.74E−109	2.72E+06	0	4.89E−19	0	0	0
F4	Mean	2.87E+01	5.39E+01	9.35E−85	4.94E−45	0	0	0
	Std	3.13E+01	5.13E+00	3.56E−84	2.71E−44	0	0	0
F7	Mean	4.26E−04	1.20E−04	9.74E−05	1.50E−04	5.36E−05	6.46E−05	1.10E−04
	Std	5.57E−04	8.83E−05	1.09E−04	2.64E−04	4.99E−05	6.23E−05	1.29E−04
F8	Mean	−2.27E+04	−3.82E+05	−5.29E+04	−4.09E+05	−4.19E+05	−4.18E+05	−2.04E+05
	Std	2.25E+03	4.60E+04	3.49E+04	1.42E+04	8.01E+01	8.93E+02	8.91E+03
F9	Mean	0	8.84E+03	0	0	0	0	0
	Std	0	4.46E+02	0	0	0	0	0
F10	Mean	9.28E−02	1.66E+01	3.97E−15	2.66E−15	8.88E−16	8.88E−16	8.88E−16
	Std	5.09E−01	4.61E−01	1.23E−15	1.81E−15	0	0	0
Mean rank		5.75	6.31	4.06	4.69	**1.88**	2.56	2.75
Final rank		6	7	4	5	**1**	2	3

Table 15. The experimental results of XOR classification problem.

Algorithms	Best	Worst	Mean	Std	Classification rate	Rank
PSO	9.33E−159	1.64E−14	1.64E−15	5.18294E−15	**100%**	**1**
SCA	0.017693	0.103151	5.84E−02	0.030791059	52.5%	4
GWO	2.31E−05	0.016054	2.29E−03	0.005167888	81.25%	3
WOA	0.034807	0.168967	1.20E−01	0.039071866	35%	5
SSA	6.84E−09	0.062788	1.26E−02	0.026433834	95%	2
MVO	2.26E−09	5.41E−05	1.18E−05	2.0205E−05	**100%**	**1**
AOA	0.105632	0.25	1.86E−01	0.05094305	23.75%	6
IAOA	2.77E−05	0.001048	2.78E−04	0.00029707	**100%**	**1**

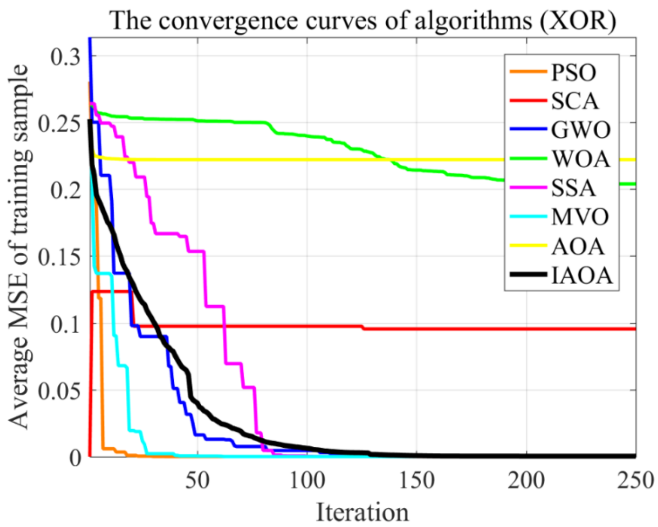

Figure 9. The convergence curves of XOR classification problem.

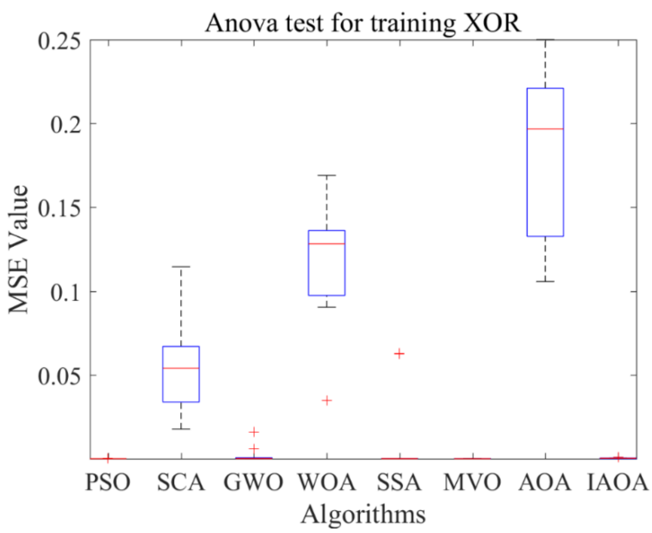

Figure 10. The variance diagram of XOR classification problem.

5.1.2. Cancer classification problem

The Cancer dataset is more complex than the XOR dataset, which has nine attributes, 599 training samples, 100 test samples, and two classes [47]. Thus the number of variables in this dataset is 209. Table 16 lists the results of IAOA and the other seven algorithms. The mean value of MSE obtained by IAOA is the lowest in this dataset. Meanwhile, the accuracy of IAOA is the highest, which is 99.1 percent. Thus the IAOA can rank first among these algorithms. As shown in Figure 11, the IAOA has the fastest convergence speed and best convergence accuracy compared with others. Also, Figure 12 exhibits that IAOA has smaller and more stable ANOVA, which demonstrates the superiority of IAOA in solving this MLP problem.

Table 16. The experimental results of cancer classification problem.

Algorithms	Best	Worst	Mean	Std	Classification rate	Rank
PSO	9.93E−04	2.27E−03	1.55E−03	4.14E−04	97.2%	4
SCA	3.95E−03	1.77E−02	1.10E−02	4.62E−03	64.8%	6
GWO	1.21E−03	1.46E−03	1.31E−03	9.69E−05	98.4%	2
WOA	1.26E−03	1.77E−03	1.54E−03	1.81E−04	98.4%	2
SSA	1.35E−03	1.72E−03	1.51E−03	1.15E−04	98.4%	2
MVO	1.31E−03	1.67E−03	1.49E−03	1.22E−04	97.6%	3
AOA	2.38E−03	6.33E−03	3.42E−03	1.26E−03	95.2%	5
IAOA	1.09E−03	1.42E−03	1.21E−03	1.21E−04	**99.1%**	**1**

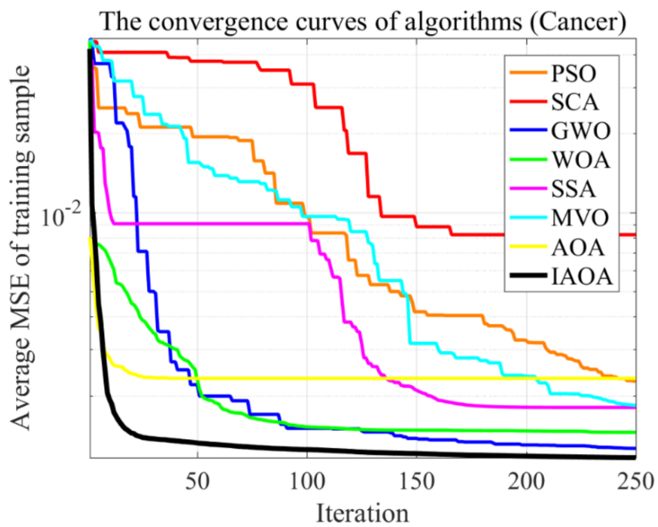

Figure 11. The convergence curves of cancer classification problem.

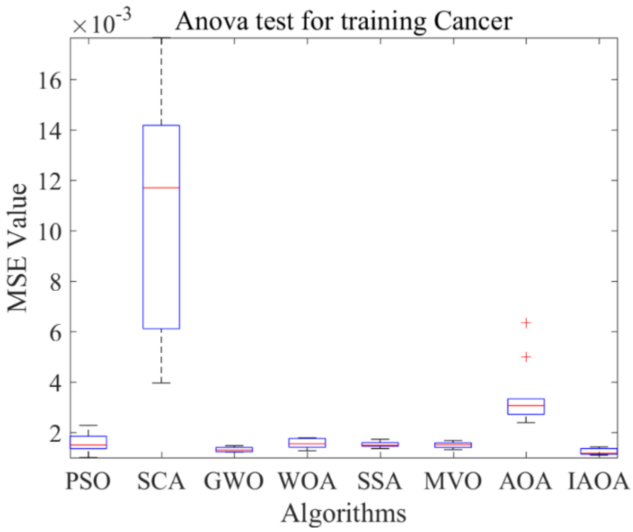

Figure 12. The variance diagram of cancer classification problem.

5.2. Engineering design problems

5.2.1. Three-bar truss design problem

The goal of designing a three-bar truss is to minimize the weights of the bar structures [48]. As shown in Figure 13, the cross-sectional area of two bars (A_1 and A_2) are the variables that need to be optimized.

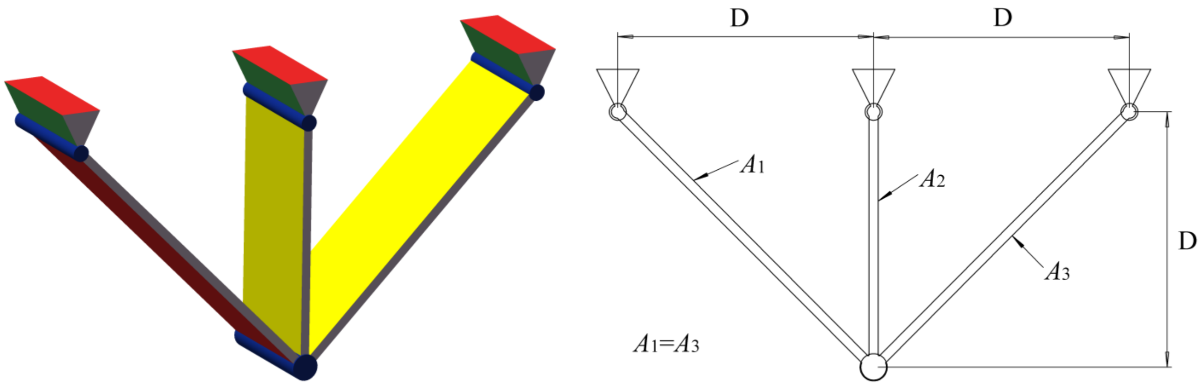

Figure 13. Three-bar truss design problem: model diagram (left) and structure parameters (right).

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

Consider

$$\vec{x} = [x_1,\ x_2] = [A_1,\ A_2]$$

Minimize

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

Subject to

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

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

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

Variable range

$$0 \le x_1,\ x_2 \le 1$$

The optimal solutions of IAOA and other comparative algorithms are given in Table 17. The results show that the IAOA can achieve the best solution among these algorithms [A_1, A_2] = [0.789676528, 0.404502112]. And the corresponding optimal weight is 263.8537231. Thus IAOA has the merits in solving this engineering design problem.

Table 17. Optimal results for comparative algorithms on the three-bar truss design problem.

Algorithm	Optimal values for variables		Optimal weight
	A_1	A_2	
IAOA	**0.789676528**	**0.404502112**	**263.8537231**
AOA [30]	0.79369	0.39426	263.9154
MFO [31]	0.788244771	0.409466906	263.8959797
SSA [41]	0.788665414	0.408275784	263.8958434
CS [49]	0.78867	0.40902	263.9716
MBA [50]	0.7885650	0.4085597	263.8958522
GOA [51]	0.7888975	0.4076195	263.8958814
PSO-DE [52]	0.7886751	0.4082482	263.8958433
HSCAHS [53]	0.7885721	0.4084012	263.881992

5.2.2. Pressure vessel design problem

The design of the pressure vessel is to obtain the lowest manufacturing cost with three constraints, i.e., the material, welding and forming [54]. As shown in Figure 14, four design variables need to be considered. They are the inner radius of the vessel (R), the thickness of the shell (T_s), the thickness of the head (T_h), and the length of the cylindrical shape (L).

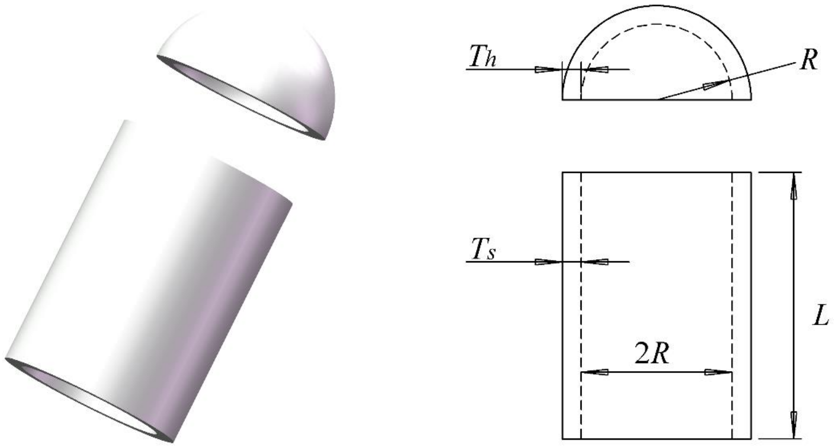

Figure 14. Pressure vessel design problem: model diagram (left) and structure parameters (right).

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

$$\vec{x} = [x_1,\ x_2,\ x_3,\ x_4] = [Ts,\ Th,\ R,\ L]$$

Minimize

$$f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to

$$\begin{aligned}
&g_1(\vec{x}) = -x_1 + 0.0193x_3 \le 0, \\
&g_2(\vec{x}) = -x_2 + 0.00954x_3 \le 0, \\
&g_3(\vec{x}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0, \\
&g_4(\vec{x}) = x_4 - 240 \le 0,
\end{aligned}$$

Variable range

$$\begin{aligned}
0 \le x_1 \le 99 \\
0 \le x_2 \le 99 \\
10 \le x_3 \le 200 \\
10 \le x_4 \le 200
\end{aligned}$$

Table 18 shows the optimal solutions for the pressure vessel design problem, obtained by nine different algorithms. It can be seen that the proposed IAOA in this paper can find the minimum cost, which is 5813.5505. This result is much lower than that of other algorithms, which indicates that the IAOA has superior performance in solving this problem.

Table 18. Optimal results for comparative algorithms on the pressure vessel design problem.

Algorithm	Optimal values for variables				Optimal cost
	T_s	T_h	R	L	
IAOA	**0.7637214**	**0.3705464**	**41.5666**	**184.1352**	**5813.5505**
AOA [30]	0.8303737	0.4162057	42.75127	169.3454	6048.7844
SMA [11]	0.7931	0.3932	40.6711	196.2178	5994.1857
MVO [16]	0.8125	0.4375	42.090738	176.73869	6060.8066
WOA [10]	0.812500	0.437500	42.098209	176.638998	6059.7410
MMPA [28]	0.77816843	0.38464899	40.31962895	199.9998973	5885.332599
MOSCA [55]	0.7781909	0.3830476	40.3207539	199.9841994	5880.71150
LWOA [56]	0.778858	0.385321	40.32609	200	5893.339
IMFO [57]	0.7781948	0.3846621	40.32097	199.9812	5885.3778

5.2.3. Tension/compression spring design problem

The objective of designing the tension/compression spring is to obtain the minimum optimal weight [58]. Three constraints (i.e., the shear stress, surge frequency, and deflection) are needed to be considered. As shown in Figure 15, there are three variables in the design of a tension/compression spring, i.e., the wire diameter (d), mean coil diameter (D), and last the number of active coils (N).

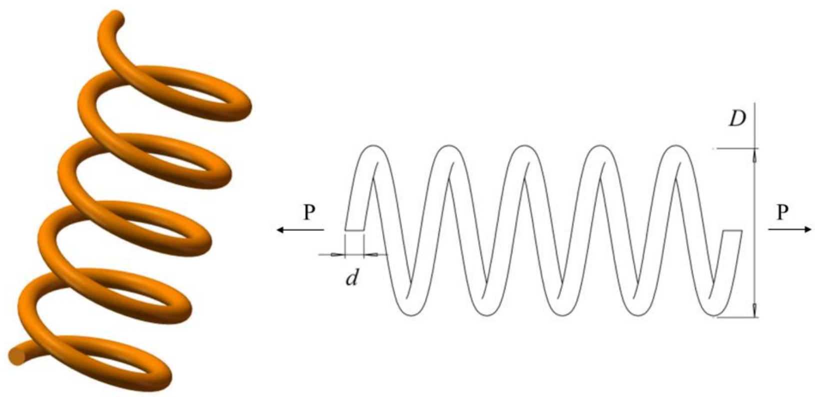

Figure 15. Tension/compression spring design problem: model diagram (left) and structure parameters (right).

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

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

Minimize

$$f(\vec{x}) = (x_3 + 2)x_2 x_1^2$$

Subject to

$$g_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0$$
$$g_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} \le 0$$
$$g_3(\vec{x}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0$$
$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \le 0$$

Variable range

$$0.05 \le x_1 \le 2.00$$
$$0.25 \le x_2 \le 1.30$$
$$2.00 \le x_3 \le 15.00$$

The results of IAOA and other well-known algorithms are listed in Table 19. The proposed IAOA achieves the best outcome for this problem, which is 0.012018312. Thus, the IAOA also can solve this problem very well.

Table 19. Optimal results for comparative algorithms on the tension/compression spring design problem.

Algorithm	Optimal values for variables			Optimal weight
	d	*D*	*P*	
IAOA	**0.05008247**	**0.363061398**	**11.19750818**	**0.012018312**
AOA [30]	0.0500	0.349809	11.8637	0.012124
MVO [16]	0.05251	0.37602	10.33513	0.012790
WOA [10]	0.051207	0.345215	12.004032	0.0126763
SSA [41]	0.051207	0.345215	12.004032	0.0126763
GWO [7]	0.05169	0.356737	11.28885	0.012666
GSA [15]	0.050276	0.323680	13.525410	0.0127022
PSO [6]	0.051728	0.357644	11.244543	0.0126747
WSA [58]	0.05168626	0.35665047	11.29291654	0.01267061

6. Conclusion and future works

The meta-heuristic algorithms are very suitable for solving optimization problems, saving time and costs. In recent years, many strategies and mechanisms have been proposed to enhance the capability of solving optimization problems for MAs. This paper presents an improved arithmetic optimization algorithm (IAOA) with a forced switching mechanism (FSM) to strengthen the exploration capability and better balance between exploration and exploitation search. The FSM will enforce the search agents to execute exploration behavior when they cannot find a better position within several iterations. Besides, the math optimizer probability used in AOA is modified by the random math optimizer probability to increase the diversity of the population. The performance of proposed IAOA is extensively evaluated by using 23 classical benchmark functions and ten CEC2020 test functions. The results of benchmark functions indicate that the IAOA is superior to the original AOA and other comparative algorithms on most of the functions, while the computational complexity of IAOA is not significantly increased. The proposed IAOA also has a stable performance on high-dimensional cases (D = 200/500/1000). In addition, two training problems of multi-layer perceptron (MLP) (XOR and Cancer classification problems) and three classical engineering design problems (three-bar truss, pressure vessel, and tension/compression spring design problems) are also employed to test the applicability of IAOA in practice. The results of real-world problems reveal that the IAOA can obtain very comparative solutions compared to the competitor algorithms.

The forced switching mechanism used in IAOA may be applicable for other MAs or improved MAs in future research. The IAOA also can be implemented on more complex real-world application problems, such as the feature selection, multilevel thresholding segmentation, 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), 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. L. Abualigah, Multi-verse optimizer algorithm: A comprehensive survey of its results, variants, and applications, *Neural Comput. Appl.*, **32** (2020), 12381–12401. doi: 10.1007/s00521-020-04839-1.
2. K. Hussain, M. N. Mohd Salleh, S. Cheng, Y. Shi, Metaheuristic research: a comprehensive survey, *Artif. Intell. Rev.*, **52** (2019), 2191–2233. doi: 10.1007/s10462-017-9605-z.

3. L. B. Booker, D. E. Goldberg, J. H. Holland, Classifier systems and genetic algorithms, *Artif. Intell.*, **40** (1989), 235−282. doi: 10.1016/0004-3702(89)90050-7.
4. J. R. Koza, J. P. Rice, Automatic programming of robots using genetic programming, in *Proceedings Tenth National Conference on Artificial Intelligence*, (1992), 194–201.
5. S. Das, P. N. Suganthan, Differential evolution: a survey of the state-of-the-art, *IEEE Trans. Evol. Comput.*, **15** (2011), 4–31. doi: 10.1109/TEVC.2010.2059031.
6. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95-International Conference on Neural Networks,* **4** (1995), 1942–1948. doi: 10.1109/ICNN.1995.488968.
7. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey Wolf Optimizer, *Adv. Eng. Softw.*, **69** (2014), 46–61. doi: 10.1016/j.advengsoft.2013.12.007.
8. D. Zhao, L. Liu, F. H. Yu, A. A. Heidari, M. J. Wang, G. X. Liang, et al., Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy, *Knowl. Based Syst.*, **216** (2020), 106510. doi: 10.1016/j.knosys.2020.106510.
9. D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, *Appl. Soft. Comput.*, **8** (2008), 687–697. doi: 10.1016/j.asoc.2007.05.007.
10. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.*, **95** (2016), 51–67. doi: 10.1016/j.advengsoft.2016.01.008.
11. S. M. Li, H. L. Chen, M. J. Wang, A. A. Heidari, S. Mirjalili, Slime mould algorithm: a new method for stochastic optimization, *Futur. Gener. Comput. Syst.*, **111** (2020), 300–323. doi: 10.1016/j.future.2020.03.055.
12. A. Faramarzi, M. Heidarinejad, S. Mirjalili, A. H. Gandomi, Marine predators algorithm: a nature-inspired metaheuristic, *Expert Syst. Appl.*, **152** (2020), 113377. doi: 10.1016/j.eswa.2020.113377.
13. H. M. Jia, X. X. Peng, C. B. Lang, Remora optimization algorithm, *Expert Syst. Appl.*, **185** (2021), 115665. doi: 10.1016/j.eswa.2021.115665.
14. C. R. Hwang, Simulated annealing: Theory and applications, *Acta. Appl. Math.*, **12** (1988), 108–111. doi: 10.1016/0378-4754(88)90023-7.
15. E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, *Inf. Sci.*, (*Ny*) **179** (2009), 2232–2248. doi: 10.1016/j.ins.2009.03.004.
16. S. Mirjalili, S. M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, *Neural Comput. Appl.*, **27** (2015), 495–513. doi: 10.1007/s00521-015-1870-7.
17. F. Asef, V. Majidnezhad, M. R. Feizi-Derakhshi, S. Parsa, Heat transfer relation-based optimization algorithm (HTOA), *Soft. Comput.*, **25** (2021), 8129–8158. doi: 10.1007/s00500-021-05734-0.
18. B. Alatas, ACROA: Artificial Chemical Reaction Optimization Algorithm for global optimization, *Expert Syst. Appl*., **38** (2011), 13170–13180. doi: 10.1016/j.eswa.2011.04.126.
19. F. F. Moghaddam, R. F. Moghaddam, M. Cheriet, Curved Space Optimization: A Random Search based on General Relativity Theory, preprint, arXiv:1208.2214.
20. Z. W. Geem, J. H. Kim, G. Loganathan, A new heuristic optimization algorithm: harmony search, *Simulation*, **76** (2001), 60–68. doi: 10.1177/003754970107600201.
21. R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching-Learning-Based optimization: an optimization method for continuous non-linear large scale problems, *Inf. Sci.*, **183** (2012), 1–15. doi: 10.1016/j.ins.2011.08.006.
22. R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching-Learning-Based Optimization: A novel method for constrained mechanical design optimization problems, *Computer-Aided Des.*, **43** (2011), 303–15. doi: 10.1016/j.cad.2010.12.015.

23. F. Ramezani, S. Lotfi, Social-Based Algorithm (SBA), *Appl. Soft. Comput.*, **13** (2013), 2837–2856. doi: 10.1016/j.asoc.2012.05.018.
24. Q. Fan, Z. J. Chen, Z. Li, Z. H. Xia, J. Y. Yu, D. Z. Wang, A new improved whale optimization algorithm with joint search mechanisms for high-dimensional global optimization problems, *Eng. Comput.*, **37** (2021), 1851–1878. doi: 10.1007/s00366-019-00917-8.
25. A. Abbasi, B. Firouzi, P. Sendur, A. A. Heidari, H. L. Chen, R. Tiwari, Multi-strategy Gaussian Harris hawks optimization for fatigue life of tapered roller bearings, *Eng. Comput.*, **2021** (2021). doi: 10.1007/s00366-021-01442-3.
26. Y. Li, Y. Zhao, J. Liu, Dynamic sine cosine algorithm for large-scale global optimization problems, *Expert Syst. Appl.*, **173** (2021), 114950. doi: 10.1016/j.eswa.2021.114950.
27. C. Y. Yu, A. A. Heidari, X. Xue, L. J. Zhang, H. L. Chen, W. B. Chen, Boosting Quantum Rotation Gate Embedded Slime Mould Algorithm, *Expert Syst. Appl.*, **181** (2021), 115082. doi: 10.1016/j.eswa.2021.115082.
28. Q. S. Fan, H. S. Huang, Q. P. Chen, L. G. Yao, D. Huang, A modified self-adaptive marine predators algorithm: framework and engineering applications, *Eng. Comput.*, **2021** (2021). doi: 10.1007/s00366-021-01319-5.
29. D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.*, **1** (1997), 67–82. doi: 10.1109/4235.585893
30. L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz, A. H. Gandomi, The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Eng.*, **376** (2021), 113609. doi: 10.1016/j.cma.2020.113609.
31. S. Mirjalili, Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, *Knowl.-Based Syst.*, **89** (2015), 228–249. doi: 10.1016/j.knosys.2015.07.006.
32. P. Manoharan, P. Jangir, D. S. Kumar, S. Ravichandran, S. Mirjalili, A new arithmetic optimization algorithm for solving real-world multiobjective CEC-2021 constrained optimization problems: diversity analysis and validations, *IEEE Access*, **9** (2021), 84263–84295. doi: 10.1109/ACCESS.2021.3085529.
33. A. Žilinskas, J. Calvin, Bi-objective decision making in global optimization based on statistical models, *J. Glob. Optim.*, **74** (2018), 599–609. doi: 10.1007/s10898-018-0622-5.
34. L. Abualigah, A. Diabat, P. Sumari, A. H. Gandomi, A novel evolutionary arithmetic optimization algorithm for multilevel thresholding degmentation of COVID-19 CT images, *Processes*, **9** (2021), 1155. doi: 10.3390/pr9071155.
35. S. Khatir, S. Tiachacht，C. L. Thanh，E. Ghandourah，M. A. Wahab, An improved artificial neural network using arithmetic optimization algorithm for damage assessment in FGM composite plates, *Compos. Struct.*, **273** (2021), 114287. doi: 10.1016/j.compstruct.2021.114287.
36. J. G. Digalakis, K. G. Margaritis, On benchmarking functions for genetic algorithms, *Int. J. Comput. Math.*, **77** (2001), 481–506. doi: 10.1080/00207160108805080.
37. C. T. Yue, K. V. Price, P. N. Suganthan, J. J. Liang, M. Z. Ali, B. Y. Qu, et al., *Problem definitions and evaluation criteria for the CEC 2020 special session and competition on single objective bound constrained numerical optimization*, (2020).
38. 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.* (*Ny*), **180** (2010), 2044–2064. doi: 10.1016/j.ins.2009.12.010
39. 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. doi: 10.1016/0010-4825(87)90003-5.

40. S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, *Knowl. Based Syst.*, **96** (2016), 120–133. doi: 10.1016/j.knosys.2015.12.022.
41. 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. Softw.*, **114** (2017), 163–191. doi: 10.1016/j.advengsoft.2017.07.002.
42. S. K. Wang, K. J. Sun, W. Y. Zhang, H. M. Jia, Multilevel thresholding using a modified ant lion optimizer with opposition-based learning for color image segmentation, *Math. Biosci. Eng.*, **18** (2021), 3092–3143. doi: 10.3934/mbe.2021155.
43. W. Long, J. J. Jiao, X. M. Liang, S. H. Cai, M. Xu, A Random Opposition-Based Learning Grey Wolf Optimizer, *IEEE Access*, **7** (2019), 113810–113825. doi: 10.1109/ACCESS.2019.2934994.
44. A. Seyyedabbasi, R. Aliyev, F. Kiani, M. U. Gulle, H. Basyildiz, M. A. Shah, Hybrid algorithms based on combining reinforcement learning and metaheuristic methods to solve global optimization problems, *Knowl. Based Syst.*, **223** (2021), 107044. doi: 10.1016/j.knosys.2021.107044.
45. R. Zheng, H. M. Jia, L. Abualigah; Q. X. Liu, S. Wang, Deep Ensemble of Slime Mold Algorithm and Arithmetic Optimization Algorithm for Global Optimization, *Processes*, **9** (2021), 1774. doi: 10.3390/pr9101774.
46. S. Wang, Q. X. Liu, Y. X. Liu, H. M. Jia, L. Abualigah, R. Zheng, et al., A Hybrid SSA and SMA with mutation opposition-based learning for constrained engineering problems, *Comput. Intel. Neurosc.*, **2021** (2021), 6379469. doi: 10.1155/2021/6379469.
47. S. Mirjalili, How effective is the grey wolf optimizer in training multi-layer perceptrons, *Appl. Intell.*, **43** (2015), 150–161. doi: 10.1007/s10489-014-0645-7.
48. T. Ray, P. Saini, Engineering design optimization using a swarm with an intelligent information sharing among individuals, *Eng. Optim.*, **33** (2001), 735–748. doi: 10.1080/03052150108940941.
49. 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. doi: 10.1007/s00366-011-0241-y.
50. A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems, *Appl. Soft. Comput.*, **13** (2013), 2592–612. doi: 10.1016/j.asoc.2012.11.026.
51. S. Saremi, S. Mirjalili, A. Lewis, Grasshopper Optimization Algorithm: Theory and application, *Adv. Eng. Softw*., **105** (2017), 30–47. doi: 10.1016/j.advengsoft.2017.01.004.
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. doi: 10.1016/j.asoc.2009.08.031.
53. N. Singh, J. Kaur, Hybridizing sine-cosine algorithm with harmony search strategy for optimization design problems, *Soft. Comput.*, **25** (2021), 11053–11075. doi: 10.1007/s00500-021-05841-y.
54. B. K. Kannan, S. N. Kramer, An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, *J. Mech. Des.*, **116** (1994), 405–411. doi: 10.1115/1.2919393.
55. R. M. Rizk-Allah, Hybridizing sine cosine algorithm with multi-orthogonal search strategy for engineering design problems, *J. Comput. Des. Eng.*, **5** (2018), 249–273. doi: 10.1016/j.jcde.2017.08.002.
56. Y. Ling, Y. Q. Zhou, Q. F. Luo, Lévy flight trajectory-based whale optimization algorithm for global optimization, *IEEE Acess*, **5** (2017), 6168–6186. doi: 10.1109/ACCESS.2017.2695498.

57. D. Pelusi, R. Mascella, L. Tallini, J. Nayak, B. Naik, Y. Deng, An improved moth-flame optimization algorithm with hybrid search phase, *Knowl. Based Syst.*, **191** (2020), 105277. doi: 10.1016/j.knosys.2019.105277.
58. 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. doi: 10.1016/j.asoc.2015.08.052.