Mathematical Biosciences and Engineering

MBE, 19(10): 10275–10315.
DOI: 10.3934/mbe.2022481
Received: 01 June 2022
Revised: 11 July 2022
Accepted: 12 July 2022
Published: 21 July 2022

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

Research article

Improved intelligent clonal optimizer based on adaptive parameter strategy

Jiahao Zhang[1,2], Zhengming Gao[2,*], Suruo Li[2], Juan Zhao[3] and Wenguang Song[1]

[1] School of computer science, Yangtze University, Jingzhou 434000, China
[2] School of computer engineering, Jingchu University of Technology, Jingmen 448000, China
[3] School of electronics and information engineering, Jingchu University of Technology, Jingmen 448000, China

* **Correspondence:** Email: gaozming@jcut.edu.cn; Tel: +867242355622.

Abstract: The intelligent clonal optimizer (ICO) is a new evolutionary algorithm, which adopts a new cloning and selection mechanism. In order to improve the performance of the algorithm, quasi-opposition-based and quasi-reflection-based learning strategy is applied according to the transition information from exploration to exploitation of ICO to speed up the convergence speed of ICO and enhance the diversity of the population. Furthermore, to avoid the stagnation of the optimal value update, an adaptive parameter method is designed. When the update of the optimal value falls into stagnation, it can adjust the parameter of controlling the exploration and exploitation in ICO to enhance the convergence rate of ICO and accuracy of the solution. At last, an improved intelligent chaotic clonal optimizer (IICO) based on adaptive parameter strategy is proposed. In this paper, twenty-seven benchmark functions, eight CEC 2104 test functions and three engineering optimization problems are used to verify the numerical optimization ability of IICO. Results of the proposed IICO are compared to ten similar meta-heuristic algorithms. The obtained results confirmed that the IICO exhibits competitive performance in convergence rate and accurate convergence.

Keywords: continuous optimization problems; opposition-based learning; intelligent clonal optimizer; adaptive parameter

1. Introduction

Intelligent technology has developed rapidly in recent decades. More and more real-world optimization problems need to be solved, such as engineering design, UAV path planning and data processing. Many traditional optimization methods have been proposed before. However, their performance is poor on complex optimization problems. The emergence of meta-heuristic algorithm solves this problem to some extent. This kind of algorithm with excellent performance has been studied by more and more scholars. The most famous meta-heuristic algorithm is genetic algorithm (GA) [1] proposed by John Holland in 1975 which is based on the mechanics of the natural selection process.

Meta-heuristic algorithms can be divided into four categories: physics-based, human-based, swarm-based, and evolutionary algorithms. Physics-based optimization algorithms are inspired by physics laws in nature. One of the most famous algorithms in this class is simulated annealing [2], which is based on the annealing process of solid matter in physics. Equilibrium optimizer (EO) [3] is a physics-based algorithm proposed in recent years, which inspired by control volume mass balance models used to estimate both dynamic and equilibrium states. A class of optimization algorithms with human behavior as the main idea is called human-based algorithms. For example, Teaching-learning-based optimization (TLBO) was proposed based on the influence of a teacher on learners [4]. The algorithm obtains the optimal value through iteration of its two parts, which include the "teacher phase", meaning learning the teacher, and the "learner phase", meaning learning by the interaction between learners. Other human-based algorithms include imperialist competitive algorithm (ICA) [5] and collective decision optimization algorithm (CDOA) [6]. Swarm intelligence (SI) algorithms are the most popular kind of algorithms. They are inspired by the behavioral patterns of populations in nature. This class of algorithms is characterized by focusing on the interaction between individuals in the population. Inspired by the foraging behavior of birds, Kennedy and Eberhart proposed the famous particle swarm optimization algorithm (PSO) [7]. PSO sets a swarm of birds (particles) searching for food (best solution) within a certain range (search space). When one of the birds finds the most food, the other birds fly in the direction of the optimal individual. They fly taking into account both the position of the global best individual and their own personal best position. At the same time, many excellent SI algorithms have been proposed in recent years. For example, Tu et al. [8] proposed a new stochastic optimization algorithm named the colony predation algorithm (CPA) which is based on the corporate predation of animals in nature. The algorithm finds the optimum by communication and collaboration, dispersing prey, encircling prey, supporting the most likely successful hunter, and seeking another target. Inspired by the behavior of the moths, Wang [9] proposed a new bio-inspired meta-heuristic algorithm called moth search (MS) algorithm. The algorithm treats the best individual as the light source. Some sub-optimal individuals move in a Lévy flights, and poor individuals move towards the best individual. Other SI optimization algorithms include ant colony optimization (ACO) [10], artificial bee colony (ABC) [11], grey wolf optimization (GWO) [12], marine predators algorithm (MPA) [13], slime mould algorithm (SMA) [14], mayfly algorithm (MA) [15], sine cosine algorithm (SCA) [16], harris hawks optimization(HHO) [17] and whale optimization algorithm (WOA) [18]. Different from SI optimization algorithm, evolutionary algorithm mainly incorporates the idea of natural selection. It selects excellent individuals from the population to the next generation according to different selection operators, and can also integrate some other operations, such as crossover operation and mutation operation.

Evolutionary algorithms include genetic algorithms (GA) [19], tree growth algorithm (TGA) [20], evolution strategy (ES) [21], differential evolution (DE) [22], arithmetic optimization algorithm (AOA) [23].

Meta-heuristic algorithms have been developed for many years, and many improvements have been proposed. As an improvement proposed in 2005, opposition-based learning (OBL) [24] has been studied and applied by many scholars. Guo et al. [25] proposed a random unscented sigma point mutation strategy and combines it with opposition-based learning strategy and nonlinear convergence factor adjustment strategy to propose an improved HHO algorithm (IHHO). This strategy achieves further optimization of the algorithm by exploiting the visible region around the optimal solution. It boosts the convergence speed and the accuracy of the solution. Si et al. [26] described the basic OBL and its four variants in detail and analyzed their impact on the search space's coverage, accuracy, exploration and exploitation, and convergence of salp swarm algorithm (SSA). They combined SSA with five OBLs to from five enhanced hybrid SSA-OBL and compared them with other algorithms. The experimental results show that incorporating different OBL in SSA enhanced its exploration ability. Hussien [27] proposed an enhanced opposition-based SSA. In this algorithm, in the initialization phase, OBL is used to generate an opposite population to enhance the diversity of the initial population. In the update phase, OBL is used in each iteration to generate the opposite population of the current population. Then, the best N individuals from these two populations are selected to replace the current population. This method effectively improved the quality of the population in the iterations. Wang et al. [28] proposed the orthogonal OBL (OOBL) based on OBL and applied it to the Yin-Yang-pair optimization to overcome the problem of low quality of candidate solutions in the exploration process. OOBL incorporates the characteristics of orthogonal experimental design, which selects the value of each dimension of the individual from the positive and opposite solutions according to the orthogonal matrix. It helps to make full use of the information of different dimensions of the positive and opposite solutions to generate better individuals. In addition to OBL, scholars have also conducted a lot of research on parameter control. The main idea of adaptive parameter control is to automatically adjust parameters during the algorithm process [29]. Lei et al. [30] proposed an aggregative learning gravitational search algorithm with self-adaptive gravitational constants (ALGSA) to alleviate the issues of low search performance and premature convergence. ALGSA adaptively adjusts the gravitational constant G according to the current state of each individual to better balance the exploration and development of the algorithm. ALGSA has better performance compared to some existing variants of the gravitational search algorithm.

Intelligent clonal optimizer (ICO) [31] is a new evolutionary algorithm which was proposed by Sahargahi et al. recently. In ICO, the population is initialized using chaos mapping, so that a population with better diversity can be obtained, which is beneficial to subsequent iterations. Next, A temporary target is generated in each iteration in the algorithm, which is determined by the best individuals in the population. A set of offsets can be calculated from this target. The offspring are generated by adding the parent to this offset or randomly generated near the parent. The number of offspring generated is defined by the fitness value of its parent. ICO has good performance in both unimodal and multimodal functions. Although the algorithm performed better than many existing algorithms, it has some defects. When it deals with the unimodal functions, the global optimal value will stagnate for a long time. After this period of stagnation, the algorithm will converge rapidly to the optimum. Therefore, the convergence rate and stability of the algorithm are affected seriously. In

order to solve the problem that the optimal value update of ICO is stagnant, the adaptive parameter strategy is introduced. The adaptive parameter strategy set a threshold for the optimal number of stagnations. When the number of stagnations exceeds this threshold, the adaptive parameter strategy will accelerate the process of the algorithm from exploration to exploitation, thereby enhancing the convergence rate of the algorithm. At the same time, ICO has the problem of falling into local optimum when dealing with some functions. The introduction of the opposition-based learning can effectively solve this problem. Different from the previous methods, two variants of OBL, quasi-opposition-based learning and quasi-reflection-based learning, are used in different phases of the algorithm to improve the exploration and exploitation capabilities of the algorithm, respectively. In the exploration phase, quasi-opposition-based learning is used to conduct a larger search around the individual. Whereas in the exploitation phase, quasi-reflection-based learning is used to perform a small search around the individual to improve the accuracy of the solution. Finally, simulation experiments were performed on three engineering optimization problems, eight CEC 2104 test functions and twenty-seven benchmark functions to verify the competitive performance of IICO.

The main contribution of this paper would be:

1) The improved ICO algorithm was proposed in this paper, which combined opposition-based learning and adaptive parameter strategy.

2) The effects of opposition-based learning and adaptive parameter strategy on ICO were discussed in detail and analyzed.

3) Comparative simulation experiments were performed on unimodal, multimodal, CEC 2104 functions and three engineering optimization problems. At the same time the experimental analysis was carried out.

The rest of the paper is structured as follows. The principle of the intelligent clonal optimizer and the improvement are described in Section 2. Simulation experiments would be carried out in Section 3. In Section 4, three engineering optimization problems are solved. Finally, conclusion is presented in Section 5.

2. The ICO and proposed IICO

2.1. Intelligent clonal optimizer

The ICO algorithm initializes the population using the chaotic mapping. A series of random points in space can be obtained by chaotic mapping. The equation of the logistic mapping is as follows:

$$X_{n+1} = \mu \times X_n \times (1 - X_n), n = 0,1, \tag{1}$$

where μ is the logistic parameter. When μ is close to 4 and $X_0 \in [0, 1]$, the logistic function is in total chaos. When a group of chaotic vectors is generated, they should be transformed into variables in search space using the following equation:

$$X_n = lb + (ub - lb) \times X_n \tag{2}$$

where lb and ub represent the upper and lower bounds of variables respectively. Assume the dimension of the problem is D, the population in the i th iteration is defined as $X_1(t), X_2(t), \dots, X_N(t)$, and the ith individual is denoted as $X_i(t) = \left(x_{i1}(t), x_{i2}(t), \dots, x_{iD}(t)\right), i =$

$1,2,\ldots,N$.

2.1.1. Cloned operator

In ICO algorithm, some individuals need to be selected to generate offspring. The number of offspring produced by each parent is defined by the following equation.

$$S_i = S_{min} + (S_{max} - S_{min}) \times NF_i \tag{3}$$

$$NF_i = \frac{f_i - f_{worst}}{f_{best} - f_{worst}} \tag{4}$$

where S_i is the number of offspring created from ith parent. S_{min} and S_{max} are the minimum and maximum number of clones respectively. S_{min} is generally set to 0. Moreover, NF_i represents the normalized fitness value of each solution while f_i refers to the fitness value of the solution. Furthermore, f_{worst} and f_{best} represent the worst and best fitness values in the current population respectively. The larger the value of NF_i, the smaller the fitness value, the better the solution.

In ICO, there are two ways to generate offspring. The first is that offspring are randomly generated within a small range of the parent. The second is that offspring are generated through the temporary target. The offspring generated by these two ways are stored in list XL and XB, respectively. For the second case, the distance from each offspring to its parents is calculated by the following equation.

$$\Delta X_i = r \times \Delta X_{i-1} + A_i \tag{5}$$

where r represents a random number between 0 and 1. The initial values of ΔX can be set to zero for all solutions. To compute A_i, some elite solutions are selected from the population and stored in ES list. The equation is as follows.

$$ES = \{x_k | x_k \in X, X = argsort\, f(x), 1 \le k \le n_{iter}\} \tag{6}$$

$$y = round(N \times [98 \times (1 - {}^{iter}/_{k}) + 2]/100) \tag{7}$$

$$n_{iter} = \begin{cases} y & y \ge 1 \\ 1 & otherwise \end{cases} \tag{8}$$

where n_{iter} is the number of the best solutions selected from the population, $iter$ is the current iteration. According to the equation, the value of n_{iter} depends on y and y is reduced from initial value N. Therefore, n_{iter} is gradually reduced from N to 1 in every iteration. The algorithm will gradually shift from the exploration to the exploitation. The value of k can be calculated as follows:

$$k = 0.25 \times (\frac{MaxFEs}{S_{max} \times N} + \frac{MaxFEs}{1 \times N}) \tag{9}$$

$$= 0.25 \times \frac{MaxFEs(1 + S_{max})}{S_{max} \times N} \tag{10}$$

where $MaxFEs$ is maximum number of evaluations of the function, N is the population number. The calculation equation of A_i is given by

$$A_i = 20 \times \alpha_{iter} \times E_i \tag{11}$$

$$\alpha_{iter} = 10 \times \ln M \times Z \tag{12}$$

$$Z = \exp(-\beta \times iter/k) \tag{13}$$

$$M = (ub - lb)/2 \tag{14}$$

where M is the mean of the upper and lower bounds of the variable. β is obtained from the following equation:

$$\beta = \begin{cases} -\ln(10 \times \gamma) \times k/iter\,, & Z \le \gamma \; and \; iter > 1 \\ \beta_0, & iter = 1 \end{cases} \tag{15}$$

where β_0 and γ are constants which is 100 and 1e-19 respectively. The value of Z is reduced from 1 to 1e-18 when it reaches γ. The value of E_i is calculated through the following equation:

$$E_i = \frac{TT_i - X_i}{R + \epsilon} \tag{16}$$

$$TT_{i,d} = \frac{\sum_{j \in ES}(r_i \times NF_j).X_{j,d}}{n_{iter}} \tag{17}$$

where TT is a temporary target which represents the weighted average of the selected elite solutions, R represents the Cartesian distance between X_i and TT_i. ϵ is a positive small constant, NF_j is determined through Eq (4), r_i is a random constant between [0, 1].

The probability of using the second method to generate offspring is defined by the following equation.

$$\sigma_{iter} = \left[{(k - iter)^{Ex}} \Big/ {(k-1)^{Ex}} \right] \times \left(\sigma_{initial} - \sigma_{final}\right) + \sigma_{final} \tag{18}$$

where $\sigma_{initial}$ and σ_{final} indicates the initial and final value of standard deviation respectively. Ex is the nonlinear modulation index which is 2. Finally, XL and XB can be obtained by the above and following equations.

$$\begin{cases} XB_{ib} = X_i + \Delta X_i, & XB\Delta X_{ib} = \Delta X_i, 1 \le ib \le NB \;\; if \; r \ge \sigma_{iter} \\ XL_{il} = X_i + \alpha_{iter} \times rn, & XL\Delta X_{il} = \Delta X_i, 1 \le il \le NL \;\; if \; r < \sigma_{iter} \end{cases} \tag{19}$$

where r is a random value in the range [0, 1], rn is a random number that obeys the normal distribution.

2.1.2. Conservative selection operator

After the offspring are generated, the elite solutions in them should be selected. The equation is as follows:

$$NL = \begin{cases} N_{XL}, & N_{XL} \le \lceil \sigma_{iter} \times N \rceil \\ \lceil \sigma_{iter} \times N \rceil, & otherwise \end{cases} \tag{20}$$

$$u = N - NL \tag{21}$$

$$NB = \begin{cases} N_{XB}, & N_{XB} \le \lceil u \times 0.9 \rceil \\ \lceil u \times 0.9 \rceil, & otherwise \end{cases} \tag{22}$$

$$NE = \begin{cases} N_X, & N_X \le u - NB \\ u - NB, & otherwise \end{cases} \tag{23}$$

In these equations, N_{XL} and N_{XB} represent the number of elements in XB and XL lists respectively. N_X shows the number of the population. NL and NB are the number of offspring selected from XL and XB respectively. NE refers to the number of offspring selected from the parents. Table 1 shows the pseudocode of the ICO algorithm.

Table 1. Pseudo-code of ICO.

Pseudo-code of ICO
Set input parameters (N, D, MaxFEs)
$\sigma_{initial} = 0.5, \sigma_{final} = 0.1, S_{min} = 0, S_{max} = 40$
$\alpha_0 = 100, \gamma = 1e-19, \mu = 4, Ex = 2$
Generate initial population $[X]_{N\times D}$ using Eqs (1) and (2)
Evaluate fitness F of each individual
$iter = 1, FEs = N$
compute k, M using Eqs (10) and (14) respectively
while $FEs \le MaxFEs$ **do**
compute Z, β using Eqs (13) and (15) respectively
compute $\sigma_{iter}, \alpha_{iter}$ using Eqs (12) and (18) respectively
compute the normalized fitness vectors NF using Eq (4)
compute n_{iter} using Eqs (7) and (8)
compute ES using Eq (6)
compute E using Eqs (16) and (17)
compute A using Eq (11)
$il = 1; ib = 1$
for $i = 1{:}N$ **do**
for $j = 1{:}S_i$ **do**
$r = random(0,1)$
if $r < \sigma_{iter}$ **do**
$XL_{il} = X_i + \alpha_{iter} \times rn$
$XL\Delta X_{il} = \Delta X_i$
$il = il + 1$
else
$\Delta X_i = r \times \Delta X_{i-1} + A_i$
$XB_{ib} = X_i + \Delta X_i$
$XB\Delta X_{ib} = \Delta X_i$
$ib = ib + 1$
end if
$FEs = FEs + 1$
end for
end for
update $X, F\ and\ \Delta X$ vectors using Eqs (20)–(23)
end while
Return the best optimal solution

2.2. Proposed IICO algorithm

2.2.1. Opposition-based learning strategy

With the development of meta-heuristic algorithms, many improvement methods have been proposed. They can be applied to all kinds of algorithms. As one of them, opposition-based learning (OBL) [24] is a very widely used improvement, which can enhance the population diversity. Later, researchers proposed its two variants named quasi-opposition-based learning [32] and quasi-reflection-based learning [33]. In this paper, these two methods are combined to improve the performance of the ICO algorithm.

Assume $X = (x_1, x_2, \dots, x_D)$ is a point in D-dimensional space with $x_j \in [lb, ub], j = 1,2, \dots, D$. The opposite point of X is defined as $\breve{X} = (\breve{x}_1, \breve{x}_2, \dots, \breve{x}_D)$. The quasi-opposite point and quasi-reflected point are defined as $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_D)$, and $\bar{X} = (\bar{x}_1, \bar{x}_2, \dots \bar{x}_D)$ respectively. These points are defined as follows:

$$\breve{x}_j = lb_j + ub_j - x_j \tag{24}$$

$$\tilde{x}_j = rand[(\tfrac{lb_j + ub_j}{2}), (lb_j + ub_j - x_j)] \tag{25}$$

$$\bar{x}_j = rand[(\tfrac{lb_j + ub_j}{2}), x_j] \tag{26}$$

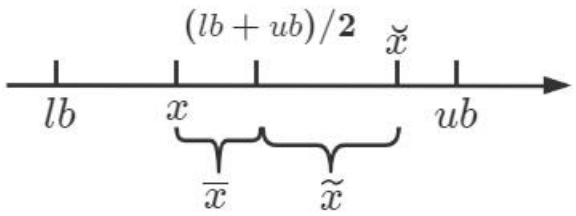


Figure 1. The value range of quasi-opposite point and quasi-reflected point.

When the dimension of the problem is one, the value ranges of the quasi-opposite and quasi-reflected points are shown in Figure 1. Figure 1 shows that the quasi-reflected point $\bar{x}$ is closer to the original point x, which is of benefit to exploiting near the original point, whereas the quasi-opposite point $\tilde{x}$ is farther from the original point, which can effectively explore around the original point. Therefore, quasi-opposition-based learning can enhance the exploration ability of ICO algorithm, and quasi-reflection-based learning can enhance the exploitation ability of ICO algorithm.

According to the updating equation of ICO algorithm, when $n_{iter} > 1$, the temporary target is the weighted average of the selected elite solutions, and the algorithm is in the exploration phase. In this phase, the quasi-opposition-based learning is applied to the algorithm when the updating Eq (19) is used and $n_{iter} > 1, r \geq \sigma_{iter}$. When the quasi-opposite solution $\tilde{X}_i(t+1)$ is calculated through Eq (25), the greedy strategy is used, and the better solution of the two enters the next iteration. The selection equation is shown in Eq (27).

$$X_i(t+1) = \begin{cases} X_i^{old}(t+1), if\ f(X_i^{old}(t+1)) < f(\tilde{X}_i(t+1)) \\ \tilde{X}_i(t+1), \qquad\qquad otherwise \end{cases} \tag{27}$$

When $n_{iter} = 1$, the temporary target is the best solution of the current population, the algorithm is in the exploitation phase. In this phase, the algorithm is quickly approaching the optimal value. Therefore, the quasi-reflection-based learning scheme is applied to IICO when the updating Eq (19) is used and $n_{iter} = 1, r \geq \sigma_{iter}$. When the quasi-reflected solution $\bar{X}_i(t+1)$ is calculated through Eq (26), the greedy selection is used again. As shown in Eq (28):

$$X_i(t+1) = \begin{cases} X_i^{old}(t+1), if\ f(X_i^{old}(t+1)) < f(\bar{X}_i(t+1)) \\ \bar{X}_i(t+1), \qquad\qquad otherwise \end{cases} \tag{28}$$

The use of the opposite-based learning strategy can effectively meliorate the performance of the algorithm.

2.2.2. Adaptive parameter strategy

According to the principle of the algorithm in the previous section, as the parameter n_{iter} changes from N to 1, the algorithm gradually transforms from exploration to exploitation. The value of n_{iter} depends on y. When $y \leq 1$, the value of n_{iter} is 1 and the ICO algorithm completely enters the exploitation phase and begins to converge rapidly. However, when the algorithm deals with the unimodal functions, this process is too long. This leads to stagnation in global optimum and decelerates the convergence rate of the algorithm as shown in Figure 2. To solve this problem, an adaptive parameter strategy is proposed, which introduces a parameter $stag$ initialized zero. When the global optimal value is not updated, the parameter $stag$ add one. When the value of $stag$ exceeds a threshold $maxStag$ and $y > 1$, the convergence of the algorithm is accelerated by subtracting y by one, and then setting $stag$ to zero. The equation is as follows:

$$y = \begin{cases} y - 1, if\ stag \geq maxStag\ and\ y > 1 \\ y, \qquad\qquad otherwise \end{cases} \tag{29}$$

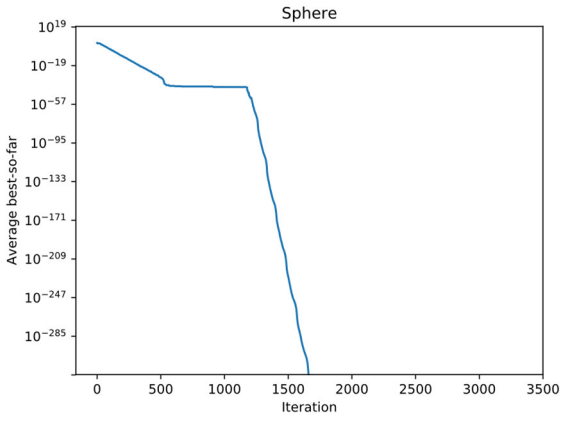


Figure 2. The convergence curve of Sphere function.

This strategy effectively prevents from the stagnation of the global optimal update and accelerates the rate of convergence on the unimodal functions. After testing, this strategy also improves the performance of the ICO algorithm on multimodal functions.

2.2.3. The flow of IICO algorithm

By combining the characteristics of the quasi-opposition-based learning and quasi-reflection-based learning, the population diversity of the original algorithm is improved and the accuracy of solutions is enhanced. At the same time, the introduction of the adaptive parameter strategy enhances the convergence speed of ICO. Figure 3 is the flow chart of the IICO algorithm. In the flowchart, the part marked in red is the improved step. When some constants in the algorithm are calculated, y is updated using Eq (29). After the population is updated, the value of the parameter $stag$ will be updated according to whether the global optimal value is updated. The pseudocode of IICO is presented in Table 2. The source code of IICO is published in https://github.com/zhangjhboy/IICO.

Table 2. Pseudo-code of IICO.

Pseudo-code of IICO
Set input parameters (N, D, MaxFEs)
$\sigma_{initial} = 0.5, \sigma_{final} = 0.1, S_{min} = 0, S_{max} = 40$
$\alpha_0 = 100, \gamma = 1e-19, \mu = 4, Ex = 2, stag = 0, maxStag = 3$
Generate initial population $[X]_{N\times D}$ using Eqs (1) and (2)
Evaluate fitness F of each individual
$iter = 1, FEs = N$
compute k, M using Eqs (10) and (14) respectively
while $FEs \leq MaxFEs$ **do**
compute Z, β using Eqs (13) and (15) respectively
compute $\sigma_{iter}, \alpha_{iter}$ using Eqs (12) and (18) respectively
compute the normalized fitness vectors NF using Eq (4)
compute y using Eq (7)
if $stag < maxStag$ and $y > 1$ **do**
$y = y - 1$
$stag = 0$
end if
compute n_{iter} using Eq (8)
compute ES using Eq (6)
compute E using Eqs (16) and (17)
compute A using Eq (11)
$il = 1; ib = 1$
for $i = 1{:}N$ **do**
for $j = 1{:}S_i$ **do**
$r = random(0,1)$
if $r < \sigma_{iter}$ **do**

Continued on next page

$XL_{il} = X_i + \alpha_{iter} \times rn$
$XL\Delta X_{il} = \Delta X_i$
$il = il + 1$
else
$\Delta X_i = r \times \Delta X_{i-1} + A_i$
$XB_{ib} = X_i + \Delta X_i$
if $n_{iter} == 1$ **do**
compute $\bar{X}$ using Eq (26)
if $fun(\bar{X}) < fun(XB_{ib})$ **do**
$XB_{ib} = \bar{X}$
end if
else
compute $\tilde{X}$ using Eq (25)
if $fun(\tilde{X}) < fun(XB_{ib})$ **do**
$XB_{ib} = \tilde{X}$
end if
$FEs = FEs + 1$
$XB\Delta X_{ib} = \Delta X_i$
$ib = ib + 1$
end if
$FEs = FEs + 1$
end for
end for
update $X, F\ and\ \Delta X$ vectors using Eqs (20)–(23)
if the global optimum is updated **do**
$stag = 0$
else
$stag = stag + 1$
end if
end while
Return the best optimal solution

2.2.4. Computational complexity of IICO

The computational complexity if IICO mainly consists of initialization, population update and fitness value calculation. Therefore, the time complexity of the ICO algorithm is $O(MaxFEs \times (D + N))$. In the opposition-based learning strategy, the time consumption of the calculation of the quasi-opposition and quasi-reflected point is $O(MaxFEs \times D)$. The complexity of adaptive parameter strategy is $O(1)$. Therefore, the whole-time complexity of the IICO algorithm is $O(MaxFEs \times (2D + N))$.

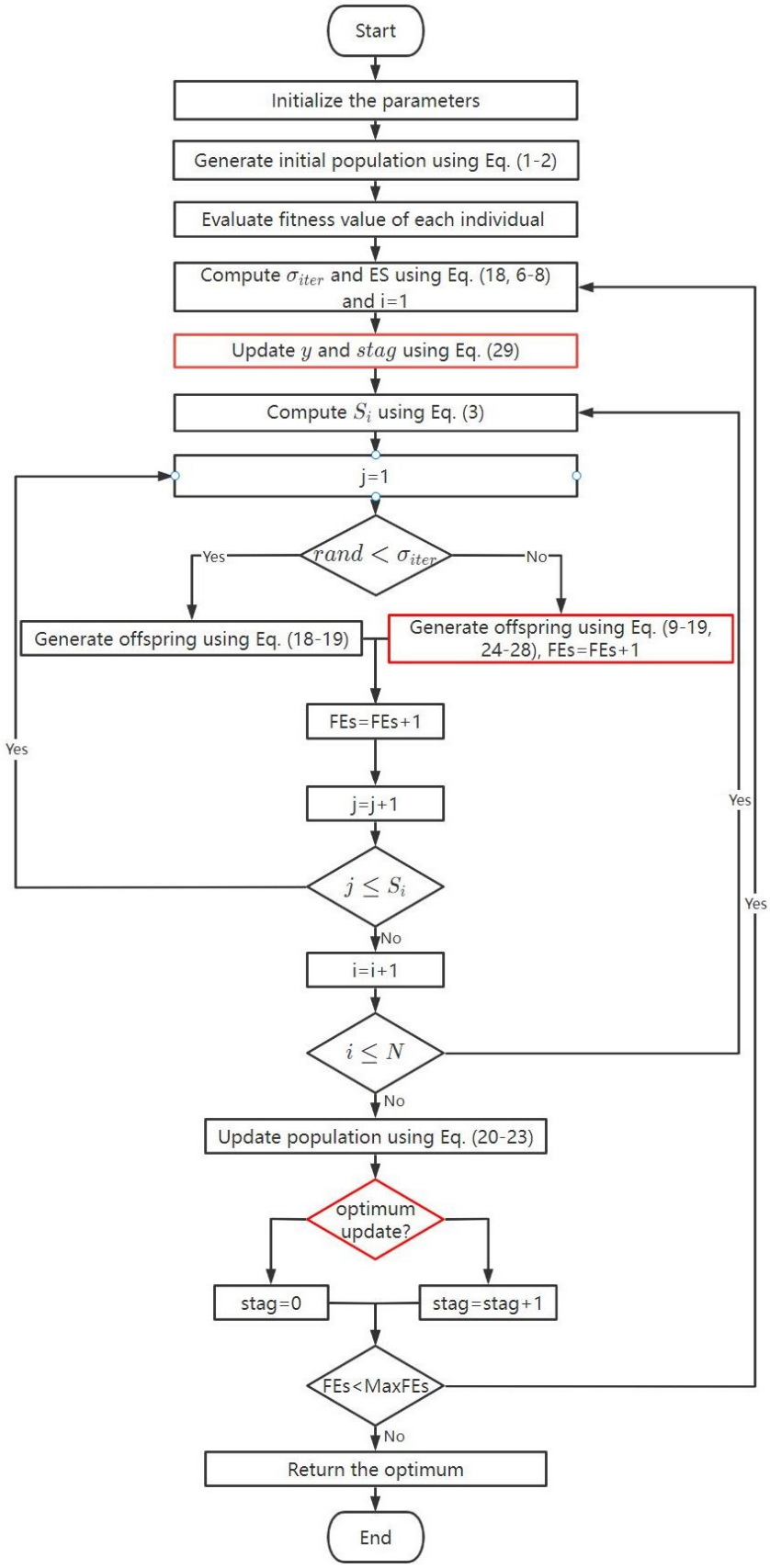


Figure 3. The flowchart of IICO.

3. Simulation experiments

3.1. Experiments setup

In order to verify the superiority of the IICO algorithm, this paper select nine unimodal, nine multimodal and nine fixed-dimension multimodal benchmark functions, as shown in Tables 3–5. Eight CEC2014 test functions are shown in Table 6. And the IICO algorithm will compare with some other algorithms. These competitive algorithms include ICO [31], EO [3], GWO [12], HHO [17], PSO [7], SCA [16], SMA [14], WOA [18], AOA [23] and MA [15]. The parameter settings of the algorithm are shown in Table 7, where D represents dimension of the functions, Range indicates the range of the search space, and f_{min} is the optimum. In the experiment, the maximum number of function evaluations is $D \times 2000$. The x-axis of the result figure of each algorithm will be displayed in the number of iterations, which can make the difference between each algorithm more obvious. The number of initial populations is set to 30 for all algorithms. In order to reduce the influence of random factors involved in each algorithm, 30 Monte Carlo simulation experiments would be carried out and the results would be averaged.

Table 3. Unimodal scalable benchmark functions.

Equations	D	Range	Min
$F_1(X) = \sum_{i=1}^{D} x_i^2$	50	[-10, 10]	0
$F_2(X) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$	50	[-100, 100]	0
$F_3(X) = \sum_{i=1}^{D} \lvert x_i \rvert^{i+1}$	50	[-1, 1]	0
$F_4(X) = (\sum_{i=1}^{D} x_i^2)^2$	50	[-100, 100]	0
$F_5(X) = \sum_{i=1}^{D} x_i^6 (2 + \sin\frac{1}{x_i})$	50	[-100, 100]	0
$F_6(X) = \sum_{i=1}^{D} i x_i^4$	50	[-10, 10]	0
$F_7(X) = \sum_{i=1}^{D} x_i^4$	50	[-100, 100]	0
$F_8(X) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	50	[-10, 10]	0
$F_9(X) = \sum_{i=1}^{D} \lvert x_i \rvert$	50	[-100, 100]	0

Table 4. Multimodal scalable benchmark functions.

Equations	D	Range	Min
$F_{10}(X) = -20e^{\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right)} - e^{\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)} + 20 + e$	50	[-32, 32]	0
$F_{11}(X) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	50	[-600, 600]	0
$F_{12}(X) =$ $-\sum_{i=1}^{D-1}\left\{e^{\left[\frac{-(x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1})}{8}\right]}\cos\left(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5x_i x_{i+1}}\right)\right\}$	50	[-5, 5]	0

Continued on next page

$F_{13}(X) = 10D + \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i)]$	50	[-5.12, 5.12]	0
$F_{14}(X) = \sum_{i=1}^{D-1}\left[0.5 + \frac{sin^2\left(\sqrt{100x_i^2+x_{i+1}^2}\right)-0.5}{1+0.001\left(x_i^2-2x_ix_{i+1}+x_{i+1}^2\right)^2}\right]$	50	[-100, 100]	0
$F_{15}(X) = \sum_{i=1}^{D-1}\left[0.5 + \frac{sin^2\left(\sqrt{x_i^2+x_{i+1}^2}\right)-0.5}{\left[1+0.001\left(x_i^2+x_{i+1}^2\right)\right]^2}\right]$	50	[-10, 10]	0
$F_{16}(X) = \sum_{i=1}^{D} ix_i^4 + rand[0,1)$	50	[-1.28, 1.28]	0
$F_{17}(X) = 0.1\sum_{i=1}^{D}(x_i-\alpha)^2 - \cos\left(k\sqrt{\sum_{i=1}^{D}(x_i-\alpha)^2}\right)$	50	[0, 10]	0
$F_{18}(X) = \sum_{i=1}^{D} uniform(0,1)\left\|x_i - \frac{1}{i}\right\|$	50	[-10, 10]	0

Table 5. Fixed-dimension multimodal benchmark functions.

Equations	D	Range	Min
$F_{19}(X) = x_1^2 + x_2^2 + 25[sin^2(x_1) + sin^2(x_2)]$	2	[-5, 5]	0
$F_{20}(X) = -\left\|\sin x_1 \cos x_2\, e^{\left\|1-\frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\|}\right\|$	2	[-10, 10]	-19.2085
$F_{21}(X) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\,[\cos(4\pi x_2)] + 0.3$	2	[-100, 100]	0
$F_{22}(X) = \frac{1}{\frac{1}{500}+\sum_j^{25}\frac{1}{j+\sum_i^2\left(x_i-a_{ij}\right)^6}}$	2	[-65.53, 65.53]	0
$F_{23}(X) = (x_1^2 + x_2^2)^{0.25}[sin^2(50(3x_1^2 + x_2^2)^{0.1}) + 1]$	2	[-100, 100]	0
$F_{24}(X) = 0.5 + \frac{sin^2(x_1^2+x_2^2)-0.5}{\left[1+0.001(x_1^2+x_2^2)\right]^2}$	2	[-10, 10]	0
$F_{25}(X) = 0.5 + \frac{sin^2(x_1^2-x_2^2)-0.5}{\left[1+0.001(x_1^2+x_2^2)\right]^2}$	2	[-100, 100]	0
$F_{26}(X) = 0.1 + sin^2(x_1) + sin^2(x_2) - 0.1e^{(-x_1^2-x_2^2)}$	2	[-10, 10]	0
$F_{27}(X) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2) + 2$	2	[-1, 1]	0

Table 6. CEC2014 test functions.

Equations	D	Range	Min
F28: Shifted and Rotated Rosenbrock's Fucntion	10	[-100, 100]	400
F29: Shifted and Rotated Ackley's Function	10	[-100, 100]	500
F30: Shifted and Rotated Weierstrass Function	10	[-100, 100]	600
F31: Shifted and Rotated Griewank's Function	10	[-100, 100]	700
F32: Shifted and Rotated Katsuura Function	10	[-100, 100]	1200

Continued on next page

F33: Shifted and Rotated HappyCat Function	10	[-100, 100]	1300
F34: Shifted and Rotated HGBat Function	10	[-100, 100]	1400
F35: Shifted and Rotated Expanded Scaffer's F6 Function	10	[-100, 100]	1600

Table 7. Parameter settings for algorithms.

Algorithm	Parameters
IICO	$Ex = 2, \mu = 4, \beta_0 = 100, \gamma = 1e-19, \sigma_{initial} = 0.5, \sigma_{final} = 0.1, maxStag = 3$
	$stag = 0, S_{min} = 0, S_{max} = \begin{cases} 2, & F \in unimodal\ function\ (F1 - F9) \\ 40, & otherwise\ (F10 - F27) \end{cases}$
ICO	$same\ as\ the\ IICO\ algorithm$
EO	$a_1 = 2, a_2 = 2, GP = 0.5$
GWO	$a = [2,0]$
HHO	$E_0 \in [-1,1]$
PSO	$c_1 = 2, c_2 = 2, weight = [0.9\ 0.4]$
SCA	$r1 \in [0,1], r2 \in [0,2\pi], r3 \in [0,2]$
SMA	$z = 0.03$
WOA	$a_1 = [2\ 0], a_2 = [-2 - 1], b = 1$
AOA	$\alpha = 5, M = 1, m = 0.2, Mu = 0.499$
MA	$g = 0.8, a1 = 1, a2 = 1.5, a3 = 1.5, \beta = 2, dance = 0.1, fl = 0.1$

The experiments were performed in Python 3.7.3 and PyCharm 2021.2.3 under the Windows 10 system with an AMD Ryzen 5 4600H CPU and 16 GB RAM.

3.2. Parameter sensitivity analysis

The sensitivity of new parameter of IICO is analyzed in this section, which is $maxStag$. This is a new parameter introduced in IICO. It affects the balance between the exploration and exploitation of the algorithm. Excessive $maxStag$ value will slow down the convergence rate of the algorithm and fail to achieve the desired effect. Too small value will lead to too short algorithm exploration phase and affect the diversity of the population. The results are as shown in Figure 4.

It is clear from the diagram that when $maxStag = 3$, the effect is best in most cases. Overall, as the value increases, the convergence rate will gradually decline.

3.3. Experimental analysis of the performance of two improvements

To verify the effect of the two improvements, the comparative experiment between IICO and two ICO variants is conducted in this section, including a combined version of ICO with OBL (ICO-OBL) and a combined version of ICO with the adaptive parameter strategy (ICO-APS). The experimental results are shown in Figure 5.

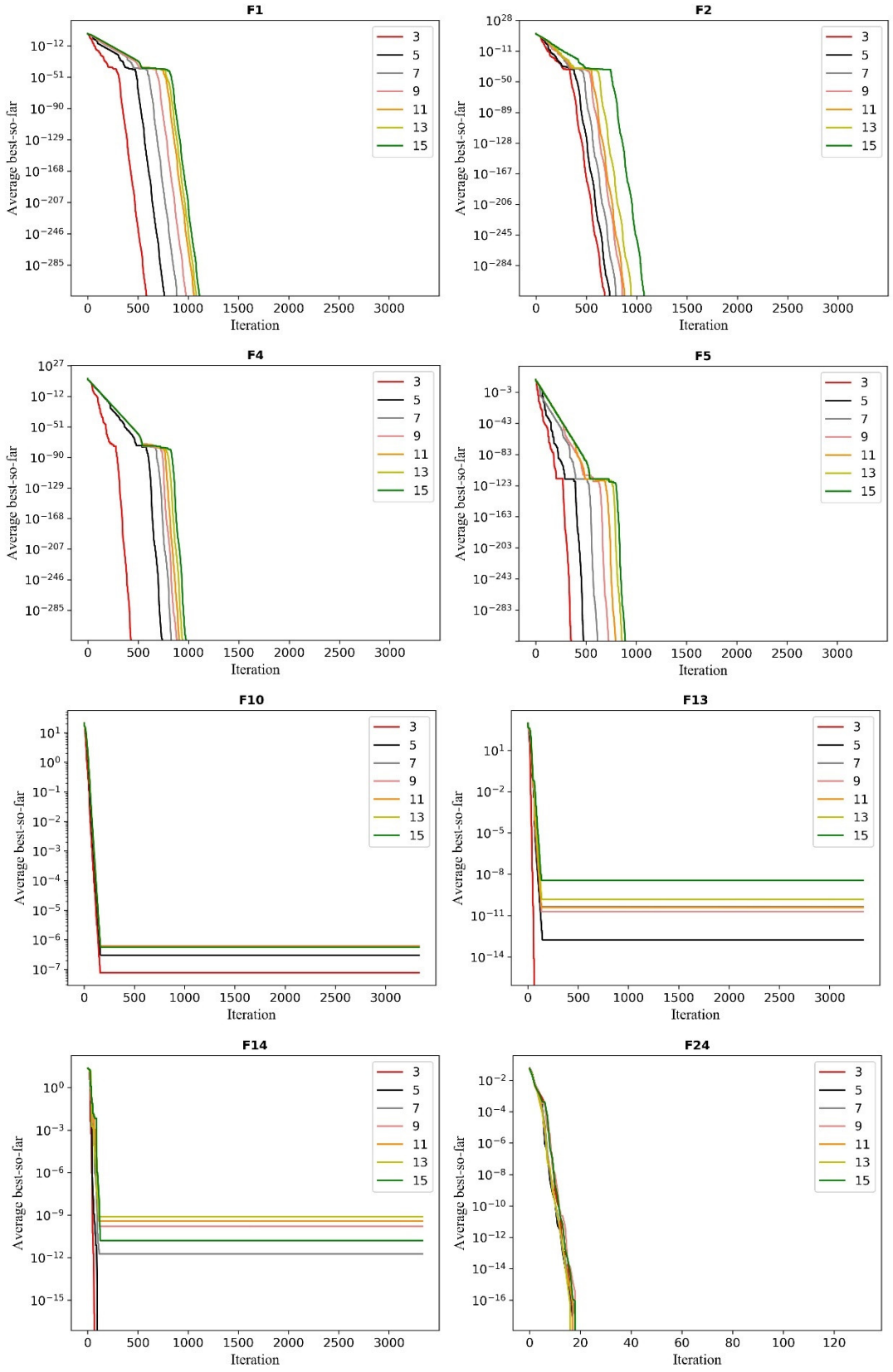
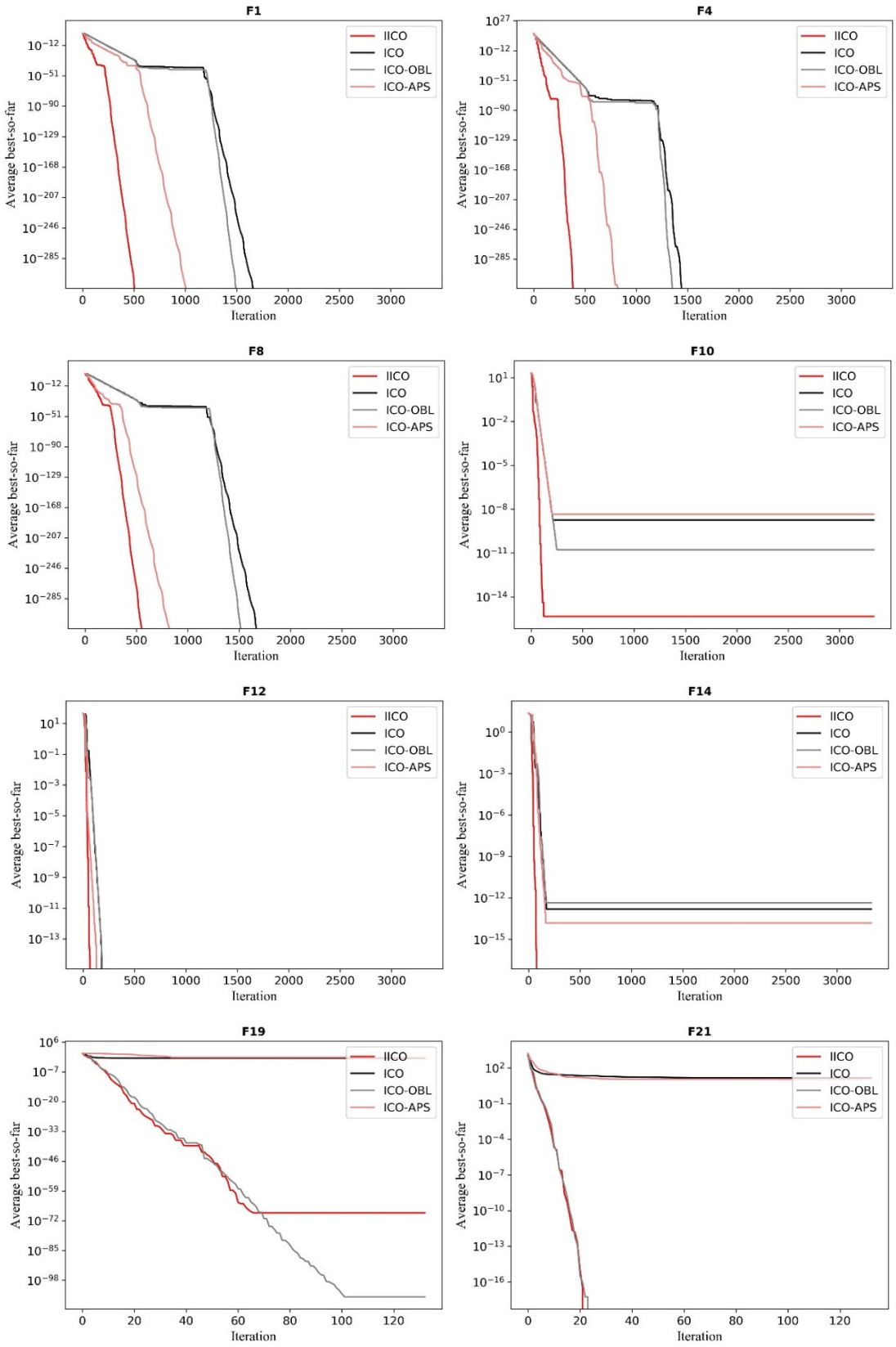


Figure 4. Sensitivity analysis results of $maxStag$.



Continued on next page

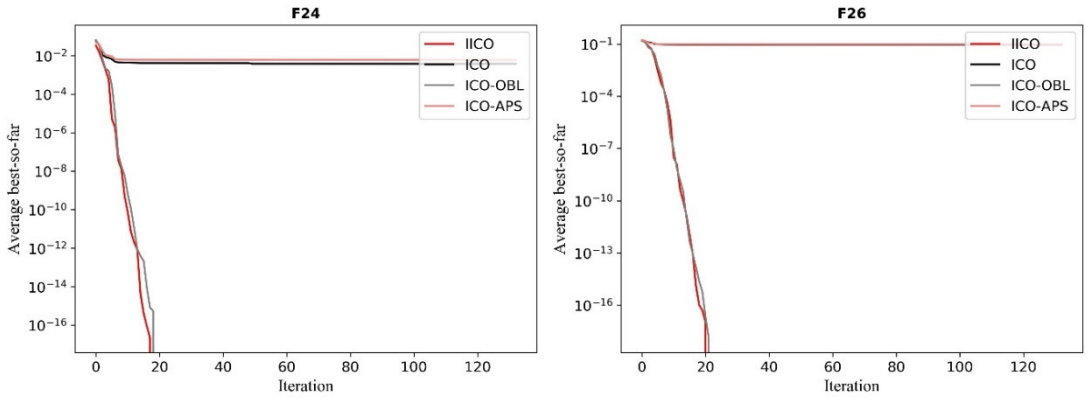


Figure 5. Experimental results of the performance of two improvements.

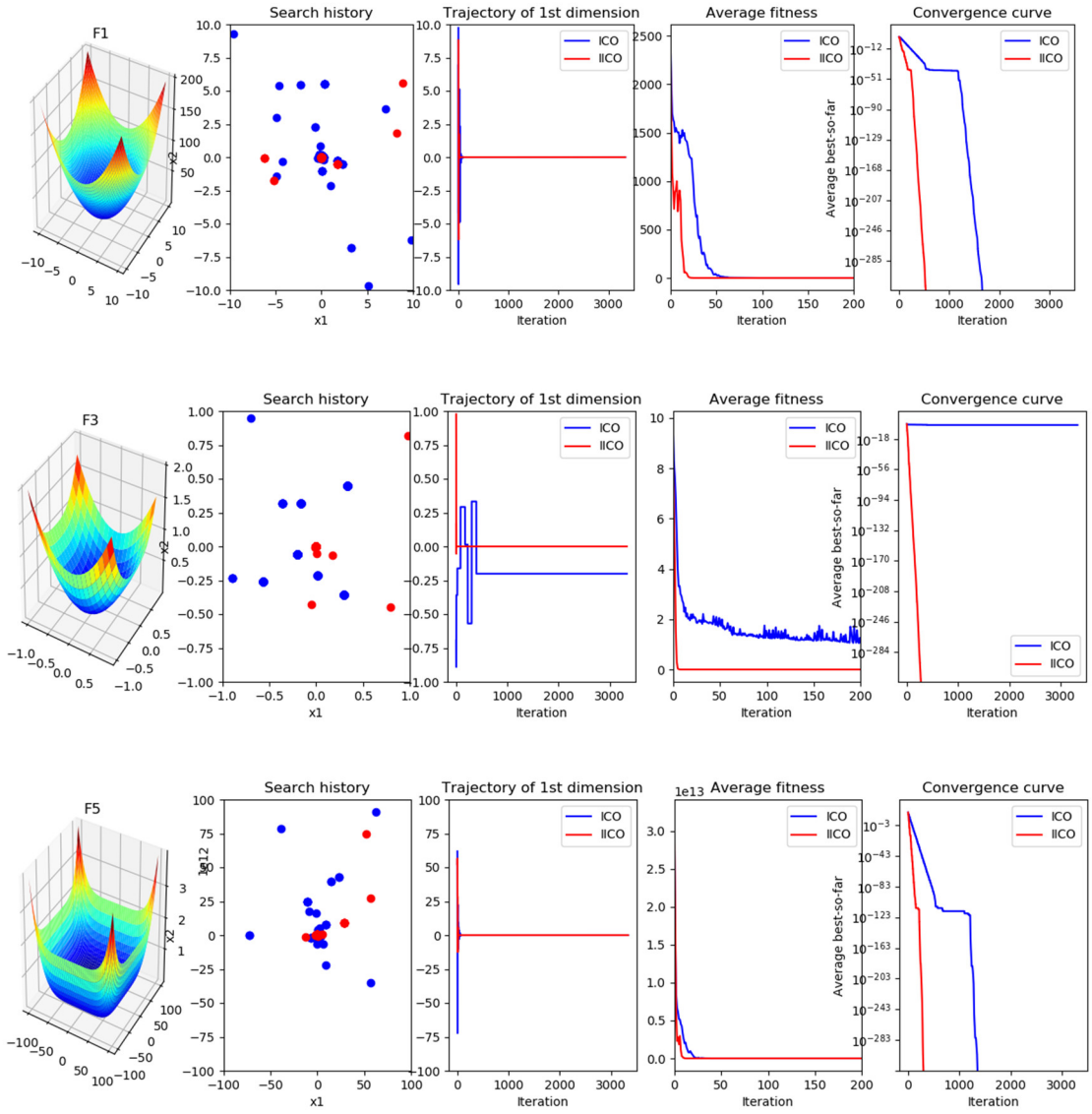
In unimodal functions F1, F4, and F8, ICO-APS has a faster convergence rate than ICO, and the reduced time is just the stagnation time of ICO. The adaptive parameter strategy effectively prevents the stagnation of the optimal value update by accelerating the process of the algorithm from exploration to exploitation and thus speeds up the convergence speed. ICO-OBL also has a faster convergence rate than ICO to a certain extent. However, it has the same stagnant phase as ICO. As a combined version of these two ICO variants, IICO has significantly better convergence speed than the other three algorithms. In multimodal functions F10, F12 and F14, ICO-APS is not much different from ICO. This is because the stagnation of ICO is mainly reflected in the unimodal function. Compared with the ICO solution, ICO-OBL has improved accuracy. When the two are combined, the convergence speed of IICO and the accuracy of the solution are significantly improved. In fixed-dimension multimodal functions F19, F21, F24 and F26, ICO-APS has the same performance as in multimodal functions. Compared with the ICO solution, the accuracy and convergence speed of ICO-OBL are obviously enhanced. In the exploration phase, quasi-opposition-based learning can search a large range around the individual to enhance the diversity of the population. During the exploitation phase, quasi-reflection-based learning can perform a fine-grained search around the individual to enhance the accuracy of the solution. The characteristics of these two make ICO-OBL achieve better results than ICO.

3.4. Qualitative analysis

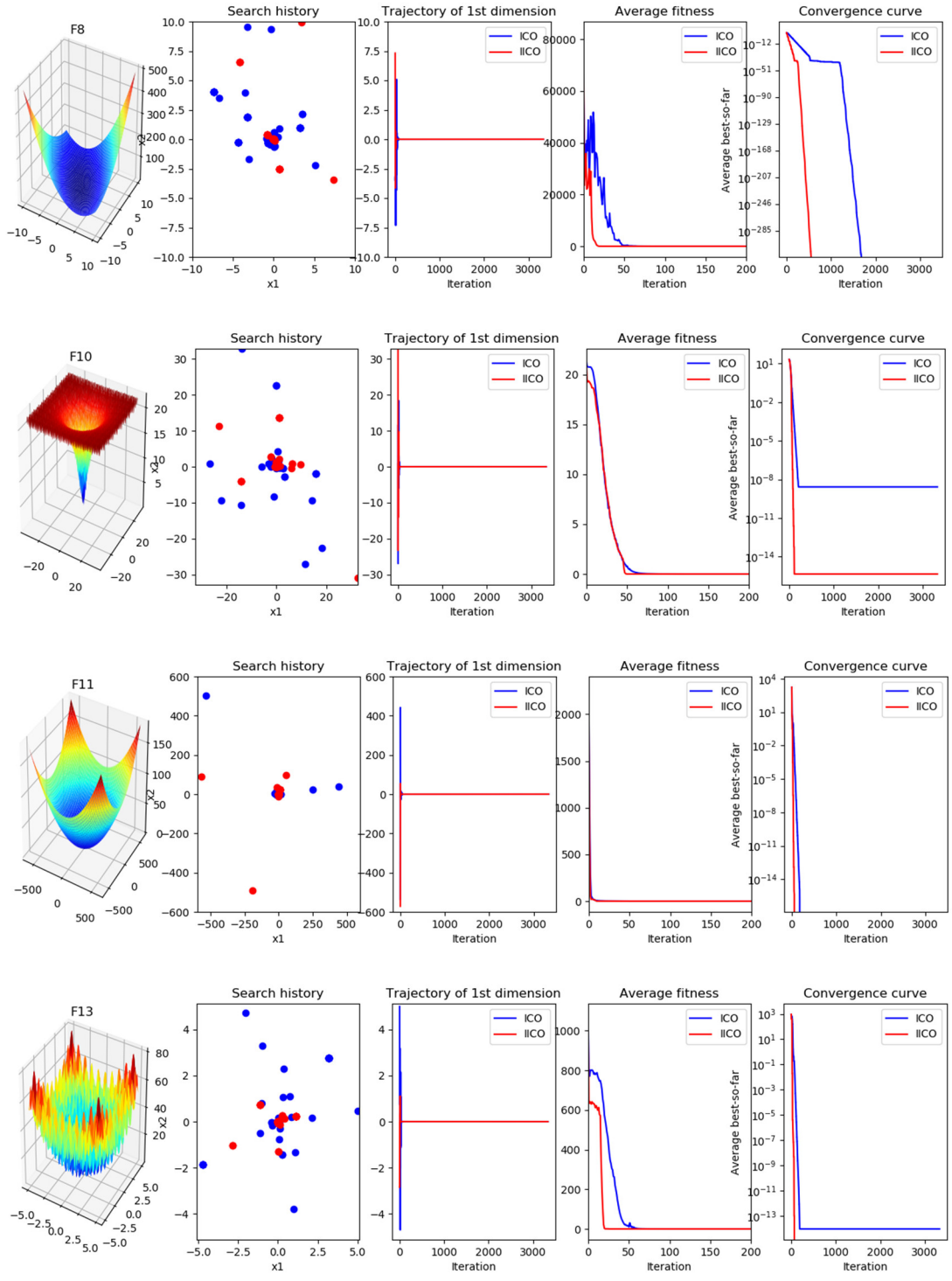
The capability of IICO is preliminarily investigated in this section. The proposed IICO algorithm would be compared to the original ICO. In Figure 6, search history, trajectory of 1st dimension, average fitness and convergence curve are demonstrated.

The first column of Figure 6 shows the 3D model of the benchmark functions. The second column of Figure 6 shows the location information of the population in the first and second dimensions. IICO and ICO are represented as red and blue points, respectively. It can be seen from the figure that red points gather more at the optimal position than blue points. This means that OBL effectively improves the quality of the population using the characteristics of quasi-opposite and quasi-reflected points, so that the individuals of IICO can reach the best position faster. The third graph of Figure 6 presents the trajectory of the first solution in the first dimension. At the beginning, it can be seen that both algorithms change dramatically. With the iteration, the curve will gradually

become stable. On function F3, IICO performs much better than ICO, and it can enter a stable state faster. However, in other functions, there is little difference between the two algorithms. The fourth and fifth columns of Figure 6 represent the average value of solutions fitness value and convergence curve respectively. The mean value of the solutions presents the overall fitness of the algorithm at each iteration. It can be seen that when the number of iterations is the same, IICO always has a better average fitness value. Moreover, the convergence curve graph records the change in the global optimal value in each iteration of the algorithm. It can be seen that IICO can reach the optimal value faster than the original algorithm. The adaptive parameter strategy speeds up convergence by reducing stagnation in optimal value updates. In the exploration phase, OBL searches in a large range around the individual to enhance the diversity of the population. In the exploitation phase, it searches within a small range of individuals to improve the accuracy of the solution and speed up the convergence rate to a certain extent. Through the above analysis, it can be concluded that the overall performance of IICO is better than the original algorithm.



Continued on next page



Continued on next page

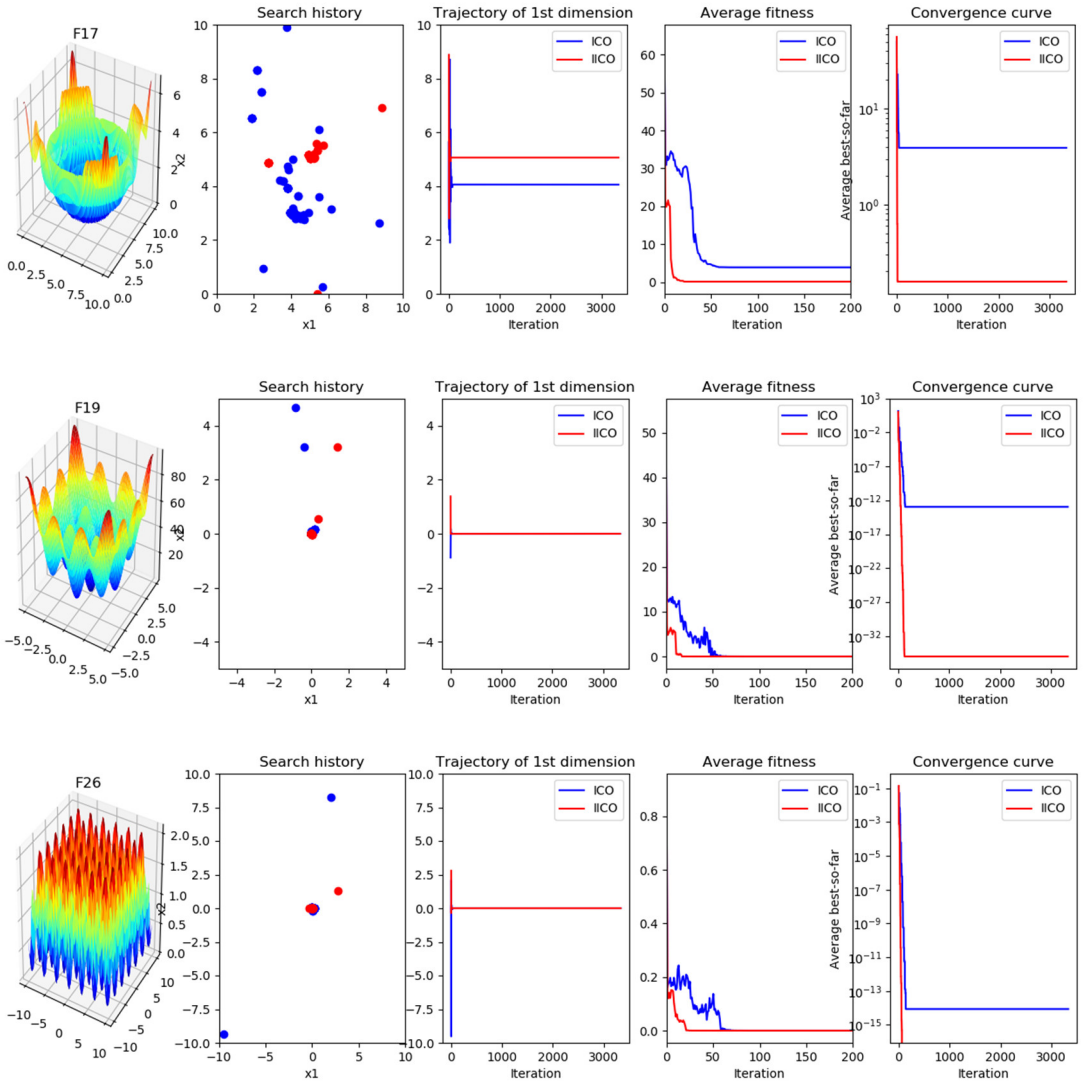


Figure 6. Qualitative results for benchmark functions involved.

3.5. Intensification capability analysis

For the unimodal benchmark functions, there is only one optimal value. All individuals in the population will aggregate towards this optimal value. The global convergence ability of the algorithm is required. Excellent algorithm can converge to the global optimal value faster. Therefore, simulation experiments on unimodal benchmark functions would be carried out to verify the intensification capability of the algorithm. The best, worst, mean and standard deviation are shown in Table 8.

Table 8. Unimodal test results.

F	IICO	ICO	EO	GWO	HHO	PSO	SCA	SMA	WOA	AOA	MA
F1											
Best	0.00E+00	0.00E+00	3.3E-161	7.44E+01	0.00E+00	1.19E-11	1.27E-46	9.85E-17	0.00E+00	5.04E-11	2.06E-24
Worst	0.00E+00	0.00E+00	2.5E-155	2.44E+02	1.1E-209	2.47E-09	4.39E-36	1.50E-08	0.00E+00	1.31E-10	2.79E-17
Mean	0.00E+00	0.00E+00	1.1E-156	1.43E+02	3.8E-211	4.19E-10	1.60E-37	6.11E-10	0.00E+00	8.01E-11	1.31E-18
Std.	0.00E+00	0.00E+00	4.6E-156	3.60E+01	0.00E+00	6.02E-10	7.88E-37	2.68E-09	0.00E+00	1.98E-11	5.15E-18
F2											
Best	0.00E+00	0.00E+00	3.9E-156	7.22E+09	3.7E-307	1.03E-03	2.18E-38	2.98E-13	0.00E+00	5.33E-03	6.33E-08
Worst	0.00E+00	0.00E+00	2.4E-149	3.13E+10	2.6E-214	1.00E+4	4.00E-28	1.49E-04	0.00E+00	1.12E-02	1.67E+02
Mean	0.00E+00	0.00E+00	9,9E-151	1.46E+10	8.7E-206	3.33E+2	2.62E-29	9.55E-06	0.00E+00	7.90E-03	8.89E+00
Std.	0.00E+00	0.00E+00	4.3E-150	5.16E+08	0.00E+00	1.79E+3	8.05E-29	2.74E-05	0.00E+00	1.80E-03	3.16E+01
F3											
Best	0.00E+00	7.93E-02	0.00E+00	5.45E-07	0.00E+00	1.81E-42	4.2E-153	9.99E-27	0.00E+00	5.18E-96	3.45E-62
Worst	0.00E+00	4.44E-01	0.00E+00	2.86E-03	4.8E-248	1.98E-34	3.8E-127	8.93E-23	0.00E+00	1.21E-55	2.95E-50
Mean	0.00E+00	2.21E-01	0.00E+00	3.16E-04	1.6E-249	6.63E-36	1.4E-128	9.73E-24	0.00E+00	4.04E-57	1.58E-51
Std.	0.00E+00	8.00E-02	0.00E+00	6.29E-04	0.00E+00	3.55E-35	6.8E-128	1.88E-23	0.00E+00	2.17E-56	5.62E-51
F4											
Best	0.00E+00	0.00E+00	0.00E+00	3.89E+07	0.00E+00	9.47E-19	7.18E-88	1.10E-27	0.00E+00	1.51E-17	2.21E-27
Worst	0.00E+00	0.00E+00	3.7E-305	5.96E+08	0.00E+00	9.37E-15	1.25E-67	2.71E-13	0.00E+00	1.29E-16	2.72E-12
Mean	0.00E+00	0.00E+00	2.2E-306	1.84E+08	0.00E+00	1.16E-15	8.30E-69	1.61E-14	0.00E+00	5.35E-17	9.10E-14
Std.	0.00E+00	0.00E+00	0.00E+00	1.17E+08	0.00E+00	1.90E-15	3.06E-68	5.99E-14	0.00E+00	2.54E-17	4.88E-13
F5											
Best	0.00E+00	0.00E+00	0.00E+00	3.29E+10	0.00E+00	1.15E-08	1.1E-119	1.19E-37	0.00E+00	1.25E-27	6.48E-07
Worst	0.00E+00	0.00E+00	1.5E-292	3.84E+11	0.00E+00	1.42E-04	8.08E-84	2.64E-21	0.00E+00	1.62E-25	5.04E-02
Mean	0.00E+00	0.00E+00	6.7E-294	1.12E+11	0.00E+00	9.08E-06	2.75E-85	9.09E-23	0.00E+00	3.19E-26	2.07E-03
Std.	0.00E+00	0.00E+00	0.00E+00	9.64E+10	0.00E+00	2.63E-05	1.45E-84	4.74E-22	0.00E+00	3.83E-26	9.11E-03
F6											
Best	0.00E+00	0.00E+00	6.0E-250	9.95E+03	0.00E+00	6.28E-13	1.05E-88	1.09E-29	0.00E+00	3.89E-21	1.79E-30
Worst	0.00E+00	0.00E+00	4.7E-233	1.61E+05	0.00E+00	1.04E-09	4.49E-64	5.90E-16	0.00E+00	3.45E-20	2.15E-22
Mean	0.00E+00	0.00E+00	1.6E-234	5.05E+04	0.00E+00	1.27E-10	1.66E-65	1.97E-17	0.00E+00	1.80E-20	1.19E-23
Std.	0.00E+00	0.00E+00	0.00E+00	3.15E+04	0.00E+00	2.13E-10	8.08E-65	1.06E-16	0.00E+00	8.32E-21	3.95E-23
F7											
Best	0.00E+00	0.00E+00	5.8E-250	2.91E+06	0.00E+00	1.24E-09	2.66E-82	2.94E-23	0.00E+00	2.92E-18	8.22E-16
Worst	0.00E+00	0.00E+00	5.2E-229	4.31E+07	0.00E+00	7.60E-07	4.50E-62	4.84E-14	0.00E+00	2.13E-17	1.27E-05
Mean	0.00E+00	0.00E+00	1.7E-230	2.04E+07	0.00E+00	8.90E-08	1.75E-63	2.17E-15	0.00E+00	8.45E-18	4.40E-07
Std.	0.00E+00	0.00E+00	0.00E+00	9.65E+06	0.00E+00	1.87E-07	8.14E-63	8.96E-15	0.00E+00	5.05E-18	2.28E-06
F8											
Best	0.00E+00	0.00E+00	1.05E-07	3.66E+02	1.1E-296	4.22E+0	2.27E-41	5.50E-15	1.04E-43	4.79E-09	1.54E-03
Worst	0.00E+00	0.00E+00	1.80E-01	1.17E+03	5.5E-218	5.72E+1	2.36E-31	8.61E-07	1.90E-01	1.71E-07	3.59E-01
Mean	0.00E+00	0.00E+00	1.00E-02	6.19E+02	3.3E-219	1.03E+1	1.77E-32	4.18E-08	6.36E-03	2.91E-08	2.75E-02
Std.	0.00E+00	0.00E+00	3.38E-02	1.65E+02	0.00E+00	9.20E+0	5.42E-32	1.55E-07	3.41E-02	3.41E-08	6.36E-02

Continued on next page

F9											
Best	0.00E+00	0.00E+00	9.23E-95	3.81E+02	6.1E-148	2.12E-06	1.86E-25	5.10E-07	0.00E+00	2.64E-04	5.36E-06
Worst	0.00E+00	0.00E+00	5.26E-91	7.50E+02	5.2E-111	1.67E-04	2.99E-20	3.55E-04	2.31E-295	4.32E-04	4.31E-02
Mean	0.00E+00	0.00E+00	3.40E-92	5.67E+02	1.7E-112	2.52E-05	1.49E-21	5.18E-05	7.70E-297	3.35E-04	4.42E-03
Std.	0.00E+00	0.00E+00	9.53E-92	9.7E+01	9.3E-112	3.68E-05	5.65E-21	6.92E-05	0.00E+00	4.30E-05	9.78E-03

It can be seen that IICO can obtain the optimal value on all selected unimodal functions, and its mean and standard deviation values are 0. This shows that the algorithm is stable. However, several other algorithms can achieve almost the same effect. For example, the results of ICO are no different from IICO except for the F3 function. Since ICO can already obtain optimal values on multiple unimodal benchmark functions, IICO cannot continue to optimize numerically. This is be because the main advantage of IICO in unimodal functions is in the speed of convergence. This table does not show the advantages of IICO very well.

3.6. Diversification capability analysis

Different from the unimodal functions, the multimodal functions have multiple local optima. The algorithm is easy to fall into local optimum and lead to premature. The strong global exploration ability is required. Like the intensification capability analysis, the relevant data are shown in Table 9 andTable **10**.

Table 9. Multimodal test results.

F	IICO	ICO	EO	GWO	HHO	PSO	SCA	SMA	WOA	AOA	MA
F10											
Best	4.44E-16	1.37E-09	7.54E-15	1.42E+01	4.44E-16	9.33E-06	4.44E-16	3.10E-08	4.44E-16	1.15E-05	1.10E+01
Worst	4.44E-16	6.26E-09	1.46E-14	1.82E+01	4.44E-16	1.82E-03	4.44E-16	5.67E-05	3.99E-15	1.93E-05	1.60E+01
Mean	4.44E-16	3.54E-09	1.07E-14	1.63E+01	4.44E-16	2.05E-04	4.44E-16	6.73E-06	6.80E-16	1.61E-05	1.40E+01
Std.	0.00E+00	1.28E-09	3.47E-15	9.30E-01	0.00E+00	3.78E-04	0.00E+00	1.13E-05	8.86E-16	1.72E-06	1.12E+00
F11											
Best	0.00E+00	0.00E+00	0.00E+00	6.53E+01	0.00E+00	4.82E-09	0.00E+00	9.75E-14	0.00E+00	4.44E-09	1.10E+01
Worst	0.00E+00	0.00E+00	7.39E-03	2.16E+02	0.00E+00	5.62E-02	0.00E+00	1.79E-07	0.00E+00	1.34E-08	6.94E+01
Mean	0.00E+00	0.00E+00	4.93E-04	1.27E+02	0.00E+00	8.02E-03	0.00E+00	1.40E-08	0.00E+00	8.81E-09	3.59E+01
Std.	0.00E+00	0.00E+00	1.84E-03	4.21E+01	0.00E+00	1.25E-02	0.00E+00	3.79E-08	0.00E+00	1.96E-09	1.51E+01
F12											
Best	0.00E+00	0.00E+00	0.00E+00	2.86E+01	0.00E+00	1.45E+01	0.00E+00	0.00E+00	0.00E+00	1.68E-10	1.43E+01
Worst	0.00E+00	0.00E+00	2.02E+01	3.68E+01	0.00E+00	2.52E+01	0.00E+00	2.08E-09	0.00E+00	4.94E-10	2.78E+01
Mean	0.00E+00	0.00E+00	5.73E+00	3.28E+01	0.00E+00	2.02E+01	0.00E+00	1.24E-10	0.00E+00	3.17E-10	1.98E+01
Std.	0.00E+00	0.00E+00	6.78E+00	1.67E+00	0.00E+00	2.77E+00	0.00E+00	3.78E-10	0.00E+00	7.35E-11	3.18E+00
F13											
Best	0.00E+00	0.00E+00	0.00E+00	1.96E+02	0.00E+00	4.07E+01	0.00E+00	0.00E+00	0.00E+00	2.32E-09	4.37E+01
Worst	0.00E+00	3.02E-14	8.65E+00	3.87E+02	0.00E+00	1.45E+02	0.00E+00	2.59E-06	0.00E+00	7.11E-09	1.26E+02

Continued on next page

Mean	0.00E+00	1.59E-15	6.29E-01	2.84E+02	0.00E+00	8.90E+01	0.00E+00	9.05E-08	0.00E+00	4.11E-09	8.87E+01
Std.	0.00E+00	5.91E-15	1.98E+00	4.40E+01	0.00E+00	2.17E+01	0.00E+00	4.65E-07	0.00E+00	1.07E-09	1.94E+01
F14											
Best	0.00E+00	2.22E-14	1.57E+01	1.86E+01	0.00E+00	1.46E+01	0.00E+00	1.73E-10	0.00E+00	4.21E-07	1.68E+01
Worst	0.00E+00	1.37E-12	1.86E+01	2.19E+01	3.73E-04	2.04E+01	0.00E+00	3.97E-05	1.51E-03	1.20E-06	2.18E+01
Mean	0.00E+00	2.25E-13	1.70E+01	2.05E+01	3.01E-05	1.78E+01	0.00E+00	3.05E-06	1.49E-04	7.74E-07	2.04E+01
Std.	0.00E+00	2.49E-13	7.73E-01	7.20E-01	8.38E-05	1.34E+00	0.00E+00	7.68E-06	3.89E-04	1.77E-07	1.6E+00
F15											
Best	0.00E+00	0.00E+00	5.48E+00	6.60E+00	0.00E+00	1.30E+00	0.00E+00	1.33E-10	0.00E+00	1.02E-10	2.03E+00
Worst	0.00E+00	0.00E+00	9.47E+00	1.03E+01	0.00E+00	4.57E+00	0.00E+00	3.66E-08	0.00E+00	2.10E-10	1.50E+00
Mean	0.00E+00	0.00E+00	7.85E+00	8.02E+00	0.00E+00	3.14E+00	0.00E+00	5.24E-09	0.00E+00	1.54E-10	9.53E+00
Std.	0.00E+00	0.00E+00	1.11E+00	1.21E+00	0.00E+00	1.09E+00	0.00E+00	1.06E-08	0.00E+00	3.41E-11	5.59E+00
F16											
Best	1.15E-07	2.07E-06	6.38E-04	9.62E+00	1.58E-07	1.20E-02	3.40E-07	2.11E-05	2.23E-06	4.83E-06	1.38E-02
Worst	2.02E-05	7.42E-05	2.08E-03	4.52E+01	1.30E-04	4.29E-02	2.41E-05	1.56E-03	6.89E-04	8.06E-05	1.48E-01
Mean	6.02E-06	2.52E-05	1.18E-03	1.99E+01	2.92E-05	2.65E-02	9.80E-06	6.33E-04	1.18E-04	1.97E-05	3.87E-02
Std.	5.43E-06	1.65E-05	4.11E-04	8.28E+00	3.38E-05	7.04E-03	6.89E-06	4.41E-04	1.66E-04	1.55E-05	2.72E-02
F17											
Best	1.56E-01	3.91E+00	2.50E+00	7.67E+00	1.56E-01	6.26E-01	2.66E+01	3.91E+00	8.13E-05	3.91E+00	3.91E+00
Worst	1.56E-01	7.67E+00	7.67E+00	1.26E+01	1.00E+01	1.40E+00	3.82E+01	7.67E+00	5.63E+00	5.64E+00	7.67E+00
Mean	1.56E-01	5.56E+00	4.05E+00	1.00E+01	4.59E+00	1.22E+00	3.27E+01	5.45E+00	3.20E-01	5.34E+00	5.13E+00
Std.	5.11E-17	1.06E+00	1.05E+00	1.71E+00	2.59E+00	3.31E-01	2.65E+00	1.14E+00	9.88E-01	6.17E-01	9.33E-01
F18											
Best	7.13E-01	7.95E-01	3.90E-01	2.84E+01	4.56E-01	3.87E+00	9.00E-01	1.01E+00	5.00E-01	9.58E-01	1.39E-01
Worst	1.05E+00	1.06E+00	7.83E-01	4.93E+01	6.52E-01	2.16E+01	1.09E+00	1.30E+00	8.69E-01	1.14E+00	3.28E+00
Mean	9.18E-01	9.23E-01	5.42E-01	3.86E+01	5.49E-01	1.07E+01	9.97E-01	1.17E+00	6.31E-01	1.05E+00	1.92E+00
Std.	8.62E-02	6.94E-02	8.81E-02	6.05E+00	4.05E-02	5.10E+00	5.00E-02	7.76E-02	9.13E-02	5.06E-02	6.05E-01

Table 10. Fixed-dimension multimodal test results.

F	IICO	ICO	EO	GWO	HHO	PSO	SCA	SMA	WOA	AOA	MA
F19											
Best	1.20E-124	2.70E-06	3.72E-65	1.27E-79	1.35E-29	4.60E-18	6.71E-16	9.19E-20	1.39E-44	0.00E+00	1.10E-22
Worst	4.01E-103	9.55E+00	8.35E-60	9.48E+00	8.40E-11	1.75E-12	2.74E-09	7.01E-14	2.10E-31	0.00E+00	4.42E-19
Mean	1.37E-104	7.99E-01	3.94E-61	3.16E-01	2.84E-12	1.79E-13	1.10E-10	3.20E-15	8.12E-33	0.00E+00	5.87E-20
Std.	7.199E-104	2.27E+00	1.51E-60	1.70E+00	1.50E-11	4.00E-13	4.92E-10	1.25E-14	3.78E-32	0.00E+00	9.29E-20
F20											
Best	-19.2085	-19.2085	-19.2085	-19.2085	-19.2045	-19.2085	-19.1978	-19.2085	-19.2085	-19.2085	-19.2085
Worst	-19.0982	-190671	-19.2085	-19.2085	-15.1402	-11.0696	-17.2418	-19.2085	-15.1402	-8.09512	-19.2085
Mean	-19.1912	-19.1909	-19.2085	-19.2085	-17.4365	-18.8976	-18.5002	-19.2085	-18.6661	-15.6753	-19.2085
Std.	3.11E-02	3.50E-02	4.62E-14	3.49E-15	1.76E+00	1.46E+00	5.39E-01	3.04E-11	1.38E+00	4.34E+00	7.08E-15

Continued on next page

F21											
Best	0.00E+00	2.46E-02	0.00E+00	0.00E+00	0.00E+00	3.89E-14	2.19E-14	0.00E+00	0.00E+00	0.00E+00	0.00E+00
Worst	0.00E+00	2.77E+02	0.00E+00	2.18E-01	4.25E-04	3.07E-10	1.61E-07	1.01E-11	2.18E-01	0.00E+00	2.18E-01
Mean	0.00E+00	2.00E+01	0.00E+00	7.27E-03	1.42E-05	4.61E-11	1.23E-08	8.45E-13	4.36E-02	0.00E+00	7.27E-03
Std.	0.00E+00	5.10E+01	0.00E+00	3.91E-02	7.63E05	7.48E-11	3.99E-08	2.53E-12	8.73E-02	0.00E+00	3.91E-02
F22											
Best	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980
Worst	7.8744	11.7187	12.6705	21.0726	19.0927	11.7187	10.7631	2.9821	15.5038	12.6705	5.9288
Mean	3.1206	5.6677	3.7015	7.1859	7.1387	3.8905	4.0405	1.2296	3.7774	9..3507	2.0953
Std.	1.5166	3.3755	4.2544	5.9973	5.0527	2.8277	2.7850	0.5545	3.7025	4.2385	1.1788
F23											
Best	2.56E-28	1.21E-01	7.90E-16	4.33E-20	1.21E-07	6.19E-04	6.68E-04	3.02E-05	2.83E-11	0.00E+00	5.24E-06
Worst	3.61E-23	2.72E+00	4.88E-14	4.17E-02	4.43E-02	1.15E-02	2.38E-02	9.67E-03	1.80E-05	0.00E+00	3.70E-04
Mean	2.17E-24	1.25E+00	1.52E-14	2.04E-03	6.25E-03	2.96E-03	6.64E-03	1.58E-03	7.75E-07	0.00E+00	9.21E-05
Std.	6.99E-24	7.33E-01	1.35E-14	7.71E-03	9.59E-03	2.13E-03	5.39E-03	2.23E-03	3.26E-06	0.00E+00	9.81E-05
F24											
Best	0.00E+00	8.11E-09	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
Worst	0.00E+00	2.87E-02	0.00E+00	1.83E-02	9.29E-03	6.66E-16	3.65E-11	3.12E-03	3.12E-03	0.00E+00	1.82E-10
Mean	0.00E+00	5.77E-03	0.00E+00	3.19E-03	6.34E-04	4.44E-17	2.42E-12	1.04E-04	1.04E-03	0.00E+00	6.08E-12
Std.	0.00E+00	6.40E-03	0.00E+00	4.97E-03	1.87E-03	1.33E-16	8.09E-12	5.61E-04	1.47E-03	0.00E+00	3.27E-11
F25											
Best	0.00E+00	7.26E-06	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.55E-15	0.00E+00	0.00E+00	0.00E+00	0.00E+00
Worst	0.00E+00	3.47E-01	0.00E+00	6.58E-02	6.24E-03	6.78E-13	3.29E-10	3.78E-07	3.08E-03	0.00E+00	6.33E-03
Mean	0.00E+00	6.88E-02	0.00E+00	4.12E-03	2.08E-04	7.06E-14	3.01E-11	1.26E-08	3.68E-04	0.00E+00	3.62E-04
Std.	0.00E+00	8.83E-02	0.00E+00	1.22E-02	1.23E-03	1.65E-13	8.36E-11	6.78E-08	8.13E-04	0.00E+00	1.37E-03
F26											
Best	0.00E+00	1.54E-05	0.00E+00	0.00E+00	0.00E+00	1.38E-17	5.89E-15	0.00E+00	0.00E+00	0.00E+00	0.00E+00
Worst	0.00E+00	1.20E-01	1.00E-01	1.00E-01	1.43E-01	1.00E-01	4.11E-02	1.00E-01	1.00E-01	0.00E+00	1.00E-01
Mean	0.00E+00	7.58E-02	3.67E-02	7.66E-02	4.63E-02	2.03E-02	1.38E-03	7.00E-02	4.33E-02	0.00E+00	2.40E-02
Std.	0.00E+00	4.08E-02	4.80E-02	4.22E-02	5.27E-02	3.98E-02	7.38E-03	4.58E-02	4.95E-02	0.00E+00	4.20E-02
F27											
Best	0.00E+00	2.00E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.22E-16	0.00E+00	0.00E+00	0.00E+00	0.00E+00
Worst	0.00E+00	3.60E-01	5.91E-09	4.84E-01	1.46E-01	1.21E-01	1.43E-07	5.11E-05	4.44E-16	0.00E+00	1.21E-01
Mean	0.00E+00	2.02E-01	1.97E-10	8.07E-02	1.29E-02	8.09E-03	5.13E-09	2.15E-16	1.48E-17	0.00E+00	1.21E-02
Std.	0.00E+00	8.03E-02	1.06E-09	1.00E-01	3.19E-02	3.02E-02	2.57E-08	9.18E-16	7.97E-17	0.00E+00	3.63E-02

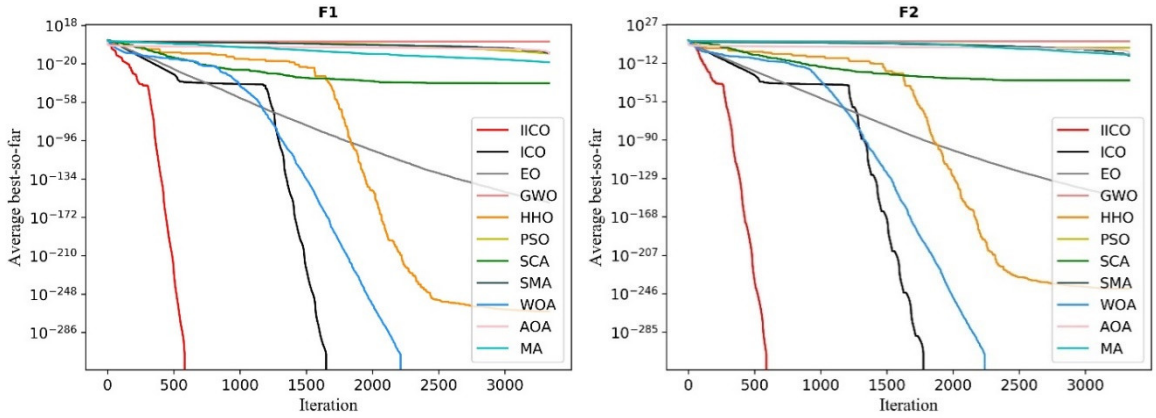
Table 9 presents the results of all algorithms in multimodal functions. According to the obtained results. The performance of IICO, SCA, WOA algorithms is significantly better than the other eight algorithms. They find better values than other algorithms in most functions. Compared with the original algorithm, IICO has better performance in functions F10, F13, F14 and F16–F18. The reason is that opposition-based learning enhances the exploration and exploitation capabilities of IICO while the adaptive parameter strategy provides more exploitation time for the algorithm. In functions F11, F12 and F15, there is no difference between the results of IICO and ICO, because ICO has obtained

the optimal data. For other algorithms, SCA and WOA have the best results. They have almost the same data as IICO.

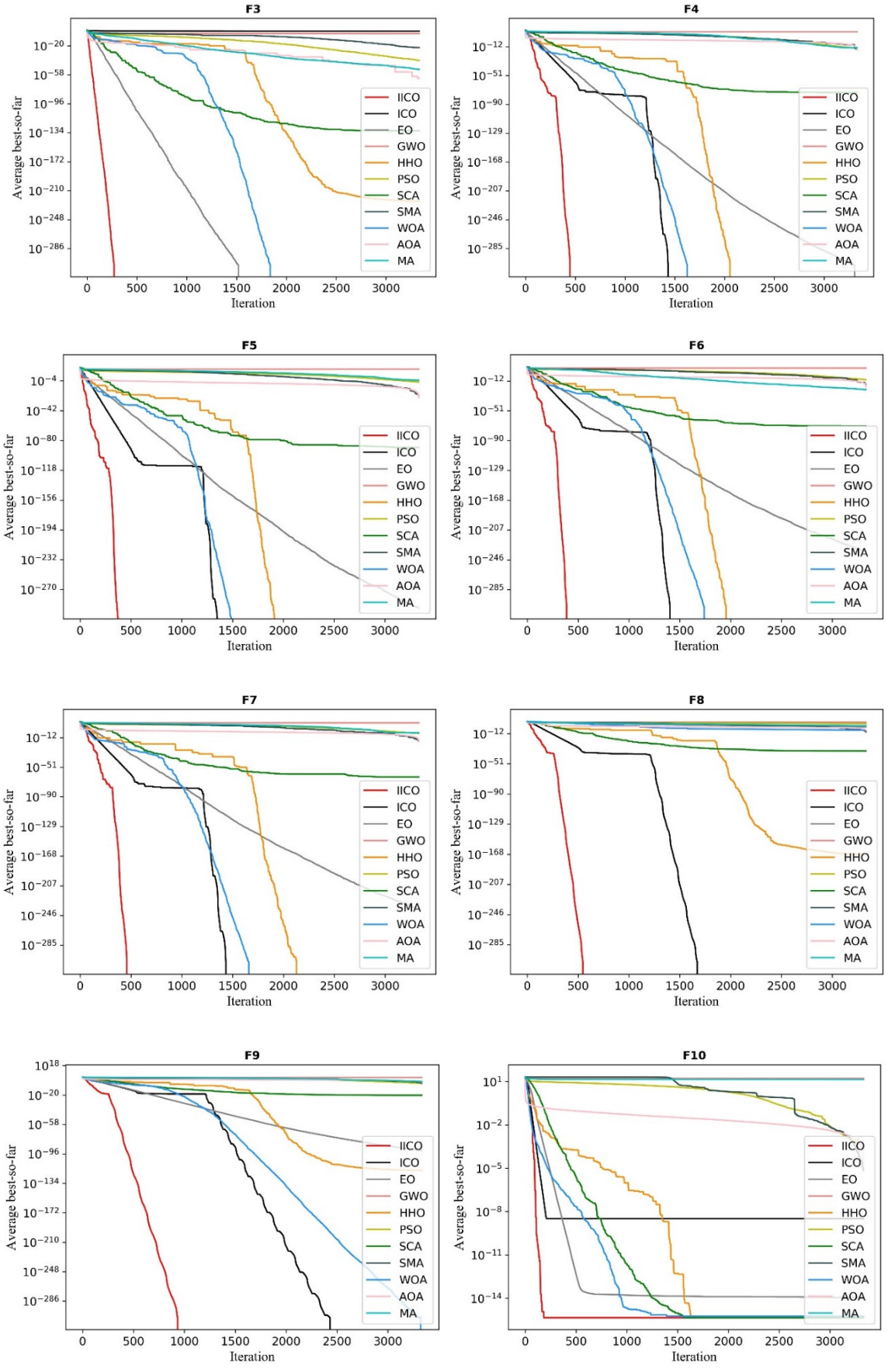
Table 10 indicates the results of all algorithms in fixed-dimension multimodal functions. Similar to the previous analysis, the results of IICO are overall better than ICO. Overall, IICO, EO and AOA have the best performance.

3.7. Acceleration convergence analysis

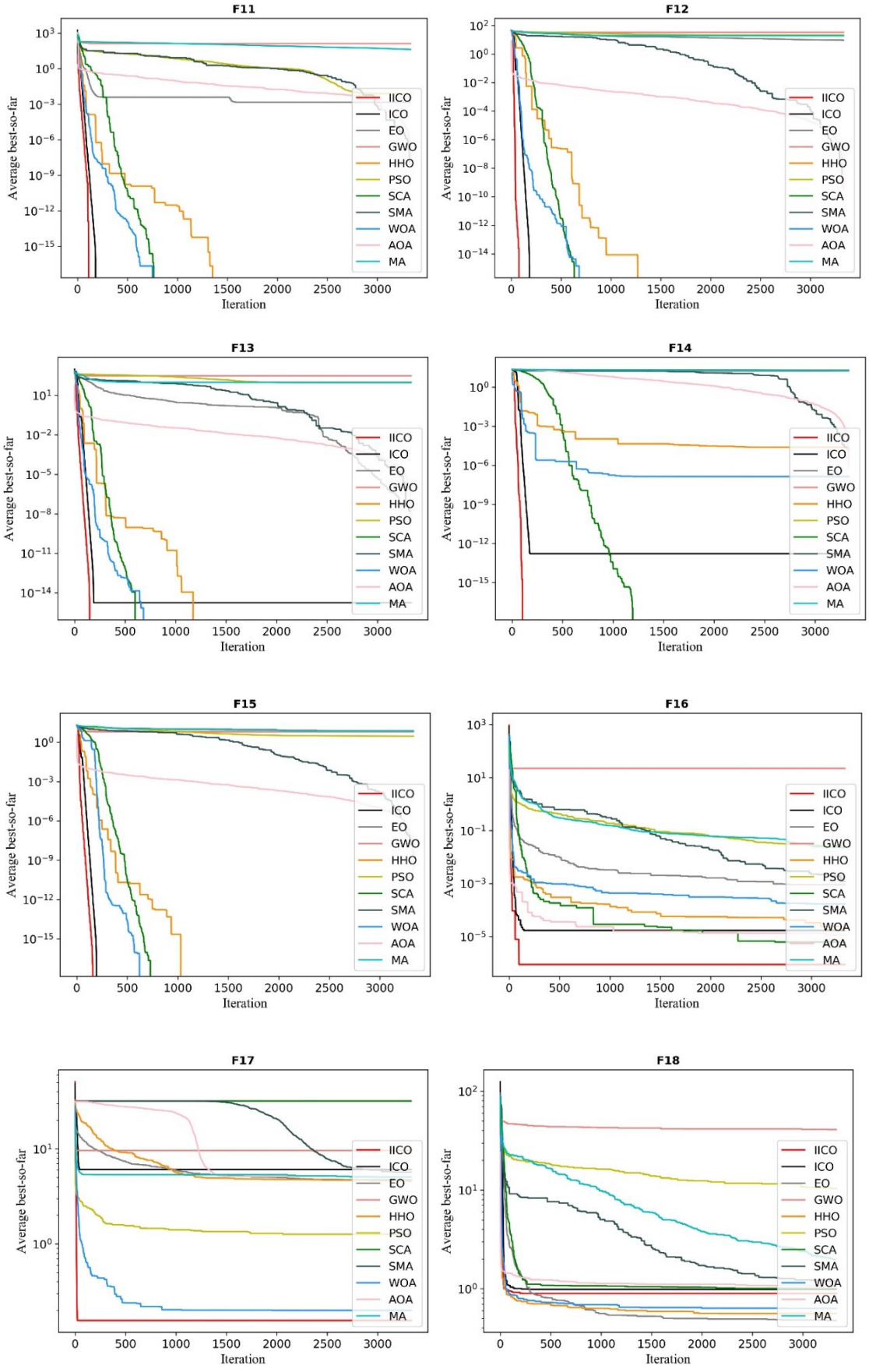
The previous part has a general understanding of the performance of IICO. In this section, IICO will be compared with other algorithms, and the comparison results are shown by the convergence curve. The results are shown in Figure 7. The figure shows the comprehensive performance of each algorithm, such as convergence speed, accuracy and so on. This can better reflect the advantages and disadvantages of each algorithm. In unimodal functions, F1–F9, IICO always reaches the optimal value fastest. This indicates that IICO has the fastest convergence rate compared to other algorithms. Furthermore, IICO also maintains the same performance in multimodal functions F10–F17, that is, it can reach the best value fastest. In addition, in functions F10, F13, F14, F16 and F17, IICO has higher solution accuracy than the original algorithm. However, in function F18, IICO does not perform very well. For fixed-dimension multimodal functions F19–F27, IICO has no best performance. In most cases, the performance of AOA is better than that of IICO. Compared with the original algorithm, the performance of IICO is still greatly improved. According to the description of NFL theory, it is impossible for an algorithm to achieve the best results in all functions, and this apples to IICO. In most benchmark functions, IICO has the best performance. In a few functions, its results are not the best.



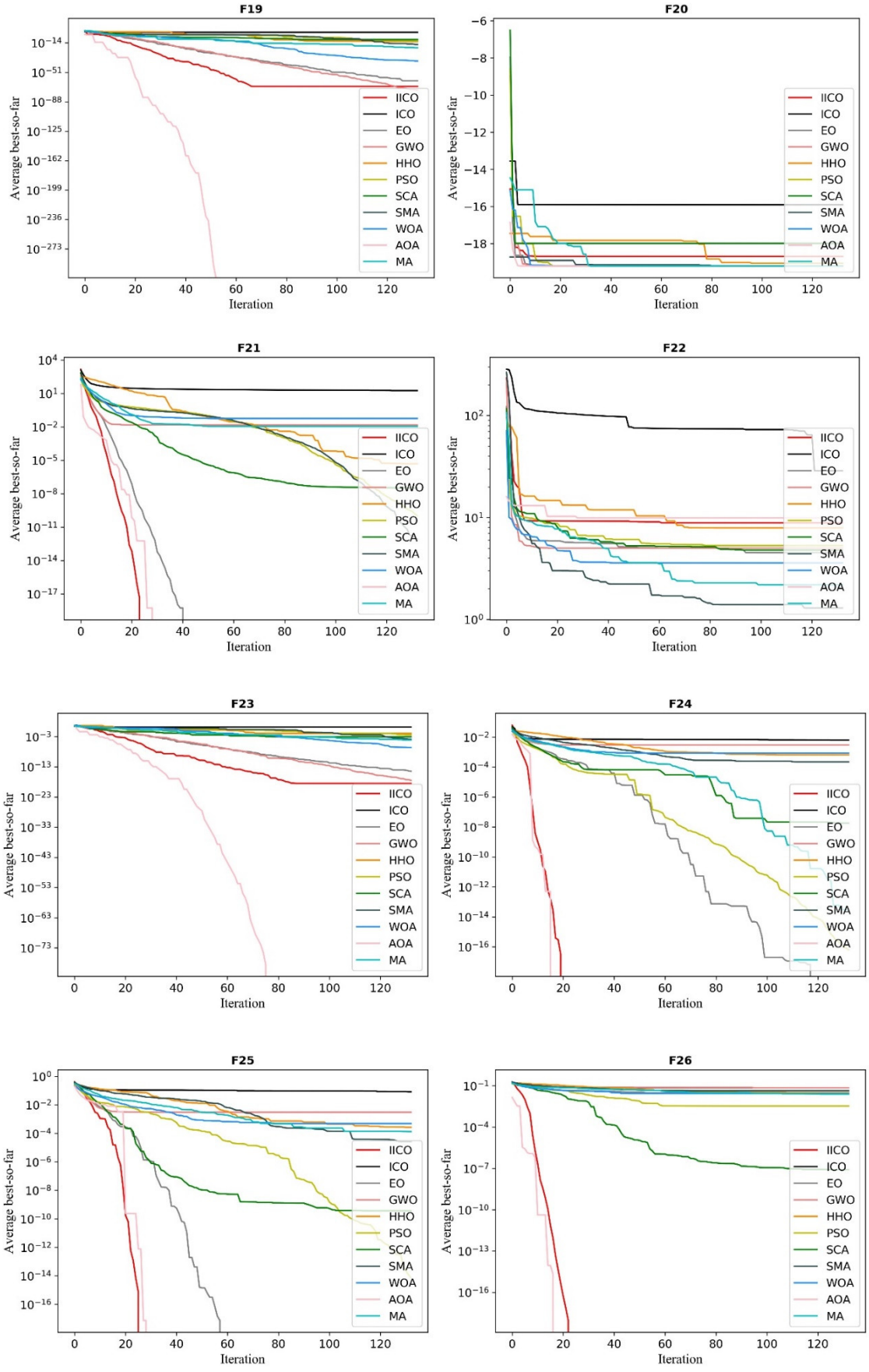
Continued on next page



Continued on next page



Continued on next page



Continued on next page

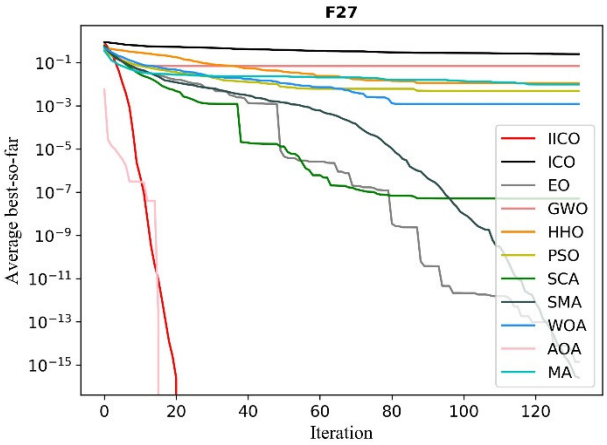
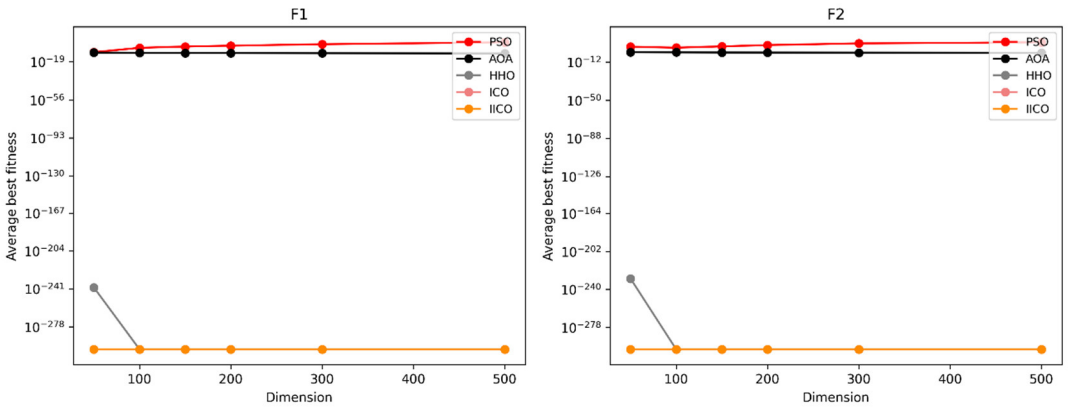


Figure 7. Convergence curves of benchmark functions.

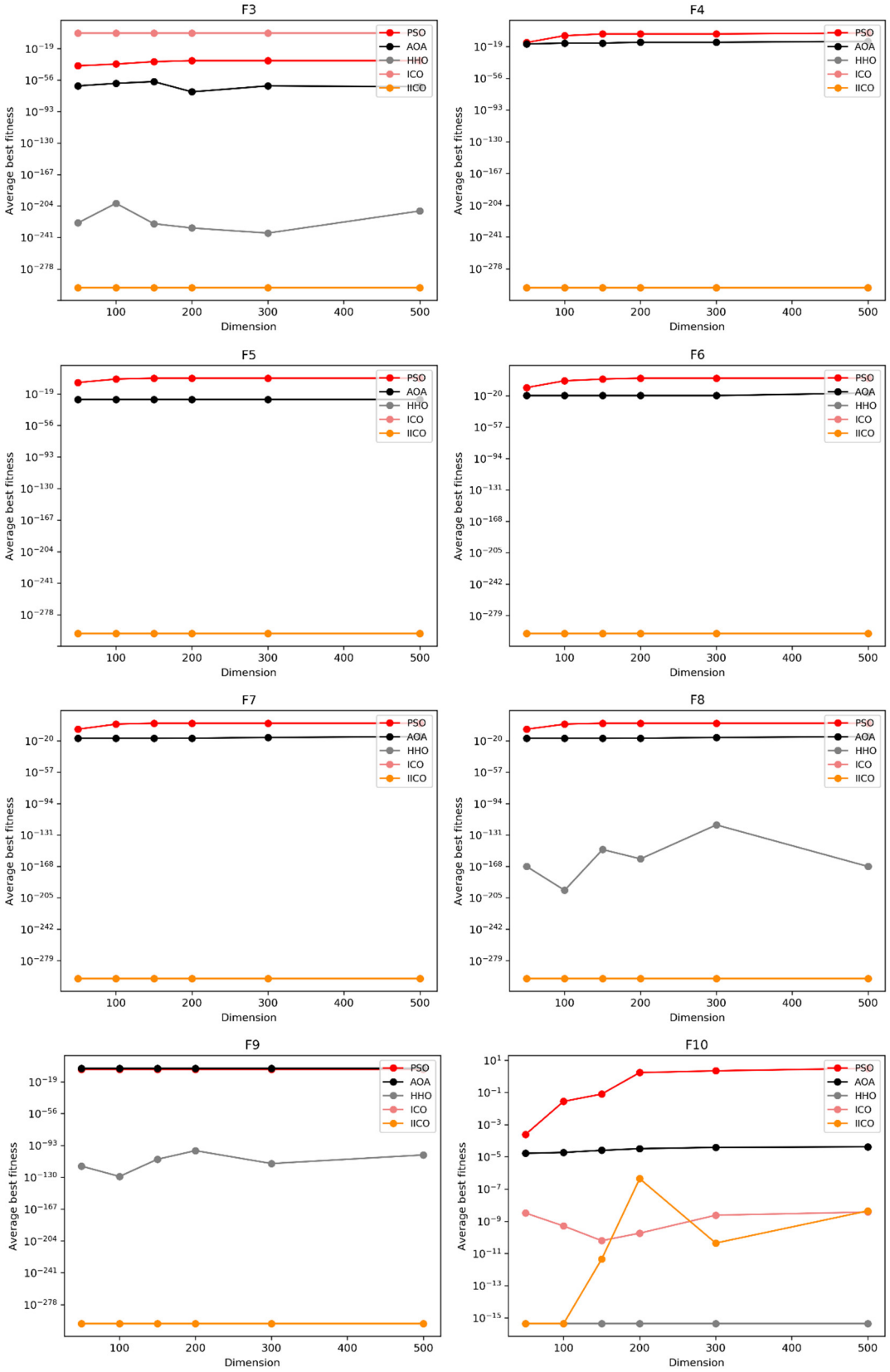
3.8. Scalability experiments

In this section, dimensional analysis will be conducted. It refers to making an algorithm run and compare results under different dimensions. In general, as the dimension of the problem increases, the accuracy of the algorithmic solution decreases. The scalability of the algorithm can be tested by dimensional analysis. The scalability means that when the dimension of the problem increases, the algorithm can still maintain the same performance except the time cost. In this experiment, the dimensions are 50, 100, 150, 200, 300 and 500 respectively. The experimental results are shown in Figure 8.

In unimodal functions F1–F9, it can be seen that the curve of IICO is mostly parallel to the x axis. This indicates that IICO has strong scalability. The increase of dimension does not change other aspects of performance except time consumption. For multimodal functions, IICO does not perform as well as in unimodal functions. In functions F12–F14 and F17, IICO still maintains good scalability. However, in other multimodal functions, IICO is unstable.



Continued on next page



Continued on next page

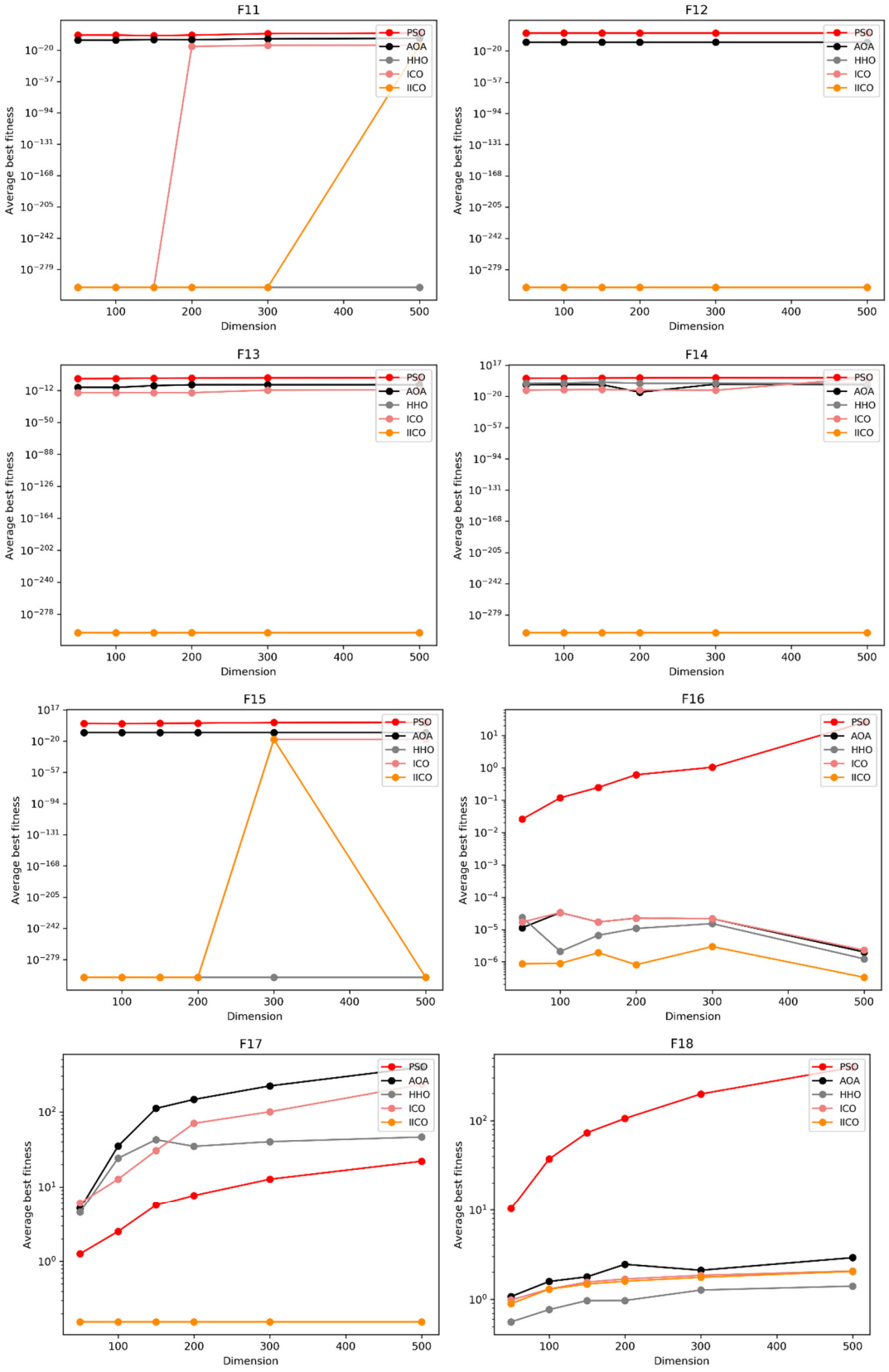


Figure 8. Scalability analysis results.

3.9. Wilcoxon rank sum test

To demonstrate the difference between IICO and other algorithms, the Wilcoxon rank sum test was performed. The optimal results obtained by IICO were compared pairwise with the results of other algorithms. The normal value p was set to be 0.05. if $p \leq 0.05$, there is a significant difference between the two algorithms, otherwise, there is no difference. The results of the Wilcoxon rank sum test are shown in Table 11.

It can be seen from the table that there is little difference between IICO and ICO in the unimodal functions. This is because both end up with the optimal value of the functions. The advantage of IICO is reflected in the fast convergence speed. In most other cases, $p < 0.05$. This shows that IICO is quite different from other algorithms.

Table 11. Results of Wilcoxon rank sum test.

Fcn	ICO	EO	GWO	HHO	PSO	SCA	SMA	WOA	AOA	MA
F1	1	6.39E-5	6.39E-5	6.39E-5	6.39E-5	6.39E-5	6.39E-5	1	6.39E-5	6.39E-5
F2	1	6.39E-5	6.39E-5	6.39E-5	6.39E-5	6.39E-5	6.39E-5	1	6.39E-5	6.39E-5
F3	6.38E-5	6.39E-5	1.83E-4	1.83E-4	1.83E-4	1.83E-4	1.83E-4	6.39E-5	1.83E-4	1.83E-4
F4	1	6.39E-5	6.39E-5	1	6.39E-5	6.39E-5	6.39E-5	1	6.39E-5	6.39E-5
F5	1	6.39E-5	6.39E-5	1	6.39E-5	6.39E-5	6.39E-5	1	6.39E-5	6.39E-5
F6	1	6.39E-5	6.39E-5	1	6.39E-5	6.39E-5	6.39E-5	1	6.39E-5	6.39E-5
F7	1	6.39E-5	6.39E-5	1	6.39E-5	6.39E-5	6.39E-5	1	6.39E-5	6.39E-5
F8	1	6.39E-5	6.39E-5	6.39E-5	6.39E-5	6.39E-5	6.39E-5	6.39E-5	6.39E-5	6.39E-5
F9	1	6.39E-5	6.39E-5	6.39E-5	6.39E-5	6.39E-5	6.39E-5	6.39E-5	6.39E-5	6.39E-5
F10	6.39E-5	1.41E-4	1.83E-4	6.39E-5	1.83E-4	6.39E-5	1.83E-4	8.74E-5	1.83E-4	1.83E-4
F11	1	1	6.39E-5	1	6.39E-5	1	6.39E-5	1	6.39E-5	6.39E-5
F12	1	7.51E-4	6.39E-5	1	6.39E-5	1	2.31E-4	1	6.39E-5	6.39E-5
F13	0.36812	0.04497	8.74E-5	0.36812	8.74E-5	0.36812	3.80E-4	0.36812	8.74E-5	8.74E-5
F14	6.39E-5	1.83E-4	1.83E-4	1.83E-4	1.83E-4	6.39E-5	1.83E-4	0.16197	1.83E-4	1.83E-4
F15	1	6.39E-5	6.39E-5	1	6.39E-5	1	6.39E-5	0.36812	6.39E-5	6.39E-5
F16	0.52052	1.83E-4	1.83E-4	0.96985	1.83E-4	0.16197	1.83E-4	0.12122	0.85010	1.83E-4
F17	1.04E-4	0.02525	0.00151	0.28718	1.25E-4	1.73E-4	0.03990	0.00106	0.96974	0.12960
F18	0.27303	1.83E-4	1.83E-4	1.83E-4	1.83E-4	0.14047	3.29E-4	1.82E-4	0.00579	0.00131
F19	1.83E-4	1.83E-4	1.83E-4	2.46E-4	1.83E-4	1.83E-4	1.83E-4	1.83E-4	6.39E-5	1.83E-4
F20	0.1858	1.83E-4	1.31E-4	0.18421	1.83E-4	0.12122	1.83E-4	0.00283	0.14046	8.74E-5
F21	6.39E-5	6.39E-5	6.39E-5	1.83E-4	1.83E-4	1.83E-4	1.82E-4	1.11E-4	6.39E-5	1.49E-4
F22	0.02574	0.00218	0.34452	0.38467	0.03756	0.02574	1.82E-4	0.00100	0.52052	5.66E-4
F23	1.83E-4	1.83E-4	1.83E-4	1.83E-4	1.83E-4	1.83E-4	1.83E-4	1.83E-4	6.39E-5	1.83E-4
F24	6.39E-5	6.39E-5	0.00610	1.11E-4	8.74E-5	1.81E-4	6.39E-5	3.11E-4	6.39E-5	6.39E-5
F25	6.39E-5	6.39E-5	0.00294	1.63E-4	1.83E-4	1.83E-4	1.10E-4	3.97E-4	6.39E-5	2.03E-4
F26	6.39E-5	7.80E-4	0.02963	0.03734	0.01725	1.83E-4	0.00211	0.00406	6.39E-5	9.18E-4
F27	6.39E-5	6.39E-5	0.00141	1.32E-4	1.83E-4	1.83E-4	6.39E-5	6.39E-5	6.39E-5	6.39E-5

3.10. Evaluation of IICO on CEC2014 function

CEC functions have complex mathematical structures. They are more difficult to find the optimal value than the benchmark functions. In this section, the comparison of the algorithms on the CEC2014 test functions is performed. These algorithms include two state-of-the-art algorithms, LSHADE [34] and LSHADE-SPACMA [35]. The results are shown in Table 12.

Table 12. CEC2104 test results.

	IICO	ICO	HHO	LSHADE	LSHADE-SPACMA
F28					
Best	458.0157	456.6196	400.0376	400	400
Worst	895.7601	1.0645e+03	496.1323	434.7803	434.7803
Mean	576.1347	588.2630	431.6983	428.1711	428.0843
Std.	111.8281	131.3821	23.2622	13.3871	13.5576
F29					
Best	517.7725	518.4564	520.0037	501.7420	500.5866
Worst	520.5204	520.1594	520.3373	520.1743	520.1129
Mean	520.1231	519.9954	520.1203	518.4621	518.3436
Std.	0.3776	0.2897	0.0929	4.8766	5.1409
F30					
Best	604.2333	604.3164	603.8481	600.0001	600
Worst	607.6096	608.5105	610.3561	602.4061	601.6936
Mean	606.4195	606.5850	607.2116	600.4229	600.3684
Std.	0.6766	0.9208	1.4511	0.6116	0.5003
F31					
Best	701.5993	703.7027	700.3113	700	700.0002
Worst	796.0747	807.5509	701.5392	700.1106	700.0861
Mean	734.9033	737.1079	700.7617	700.0393	700.0282
Std.	18.7059	19.9049	0.2443	0.0261	0.0203
F32					
Best	1.2001e+03	1.2001e+03	1.2003e+03	1.2000e+03	1.2001e+03
Worst	1.2007e+03	1.2004e+03	1.2015e+03	1.2002e+03	1.2002e+03
Mean	1.2003e+03	1.2002e+03	1.2006e+03	1.2001e+03	1.2002e+03
Std.	0.1288	0.0682	0.2549	0.0391	0.0401
F33					
Best	1.3003e+03	1.3003e+03	1.3002e+03	1.3000e+03	1.3000e+03
Worst	1.3034e+03	1.3034e+03	1.3010e+03	1.3002e+03	1.3002e+03
Mean	1.3014e+03	1.3014e+03	1.3006e+03	1.3001e+03	1.3001e+03
Std.	0.9532	0.9575	0.2074	0.0422	0.0430
F34					
Best	1.4003e+03	1.4003e+03	1.4001e+03	1.4001e+03	1.4001e+03
Worst	1.4211e+03	1.4205e+03	1.4013e+03	1.4003e+03	1.4004e+03

Continued on next page

Mean	1.4095e+03	1.4076e+03	1.4003e+03	1.4002e+03	1.4002e+03
Std.	5.0227	4.5588	0.2629	0.0722	0.0975
F35					
Best	1.6026e+03	1.6026e+03	1.6024e+03	1.6013e+03	1.6009e+03
Worst	1.6037e+03	1.6036e+03	1.6040e+03	1.6034e+03	1.6029e+03
Mean	1.6032e+03	1.6032e+03	1.6033e+03	1.6022e+03	1.6019e+03
Std.	0.2732	0.2747	0.3063	0.4347	0.4306

It can be seen that LSHADE and LSHADE-SPACAM, as the winners of the CEC competition, have superior performance on CEC2014 test functions. The results of IICO are similar to those of these two algorithms except for F28 and F31. From the results, IICO does not have much improvement compared to ICO.

4. Solving engineering design problems

IICO was applied to three famous engineering design problems to test its accuracy and efficiency, such as gear train design and welded beam design. To handle constraints, a static penalty function method is adopted. In the penalty function method, the fitness function is defined as the sum of the objective function and a penalty term which depends on the constraint violation [36]. The penalty coefficient is a constant for all constraints, which can penalize the solution that violate constraints.

The following subsections present the results of using the algorithms in each problem. Moreover, each algorithm runs 30 times independently. The number of function evaluations is $D \times 2000$.

4.1. Gear train design

The objective of this problem is to find the optimal number of tooth for four gears of a train to minimize the gear ratio [37]. There are four variables in this problem. Since the values of each variable are integers, all results are rounded to integers. This problem is defined as follows:

$$Min \quad f(x) = \left(\frac{1}{6.931} - \frac{x_1 x_3}{x_2 x_4}\right)^2$$

$$s.t.\ 12 \le x_1, x_2, x_3, x_4 \le 60 \tag{30}$$

The results obtained by IICO and other algorithms are presented in Table 13. The table shows that IICO has better fitness value in solving the problem compared with the original ICO algorithm. IICO is also better than other algorithms.

Table 13. Results on gear train design problem.

Algorithm	Optimal solution				f_{min}
	x_1	x_2	x_3	x_4	
IICO	49	19	21	58	1.1787E-25
ICO	54	13	26	47	8.1821E-16
AOA	48	13	13	25	1.0977E-10
EO	41	13	20	46	7.7582E-12
HHO	31	14	12	40	2.5704E-14
PSO	58	29	12	42	1.5514E-19
SCA	13	17	53	12	7.3005E-10

4.2. Welded beam design

The welded beam is a common engineering optimization problem with an objective to find an optimal set of the dimensions $h = x_1,\ l = x_2,\ t = x_3$, and $b = x_4$ such that the fabrication cost of the beam is minimized. It's a continuous optimization problem. The cost of the welded beam is formulated as follows:

$$\begin{gathered} Min \quad f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \\ s.t. \quad g_1(x) = \tau(x) - \tau_{max} \le 0 \\ g_2(x) = \sigma(x) - \sigma_{max} \le 0 \\ g_3(x) = x_1 - x_4 \le 0 \\ g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \le 0 \\ g_5(x) = 0.125 - x_1 \le 0 \\ g_6(x) = \delta(x) - \delta_{max} \le 0 \\ g_7(x) = P - P_c(x) \le 0 \end{gathered} \tag{31}$$

The related variables and constants are expressed as follows:

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

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

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

$$J = 2\left[\sqrt{2}x_1x_2\left(\frac{x_2^2}{12} + \frac{(x_1 + x_3)^2}{4}\right)\right],$$

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

$$\delta(x) = \frac{4PL^3}{E x_3^3 x_4},$$

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

$$P = 6000, L = 14, E = 30 \times 10^6,$$

$$G = 12 \times 10^6,$$

$$\tau_{max} = 13000, \sigma_{max} = 30000, \delta_{max} = 0.25$$

The range of the variables is as follows:

$$0.1 \le x_1 \le 2, 0.1 \le x_2 \le 10, 0.1 \le x_3 \le 10, 0.1 \le x_4 \le 2$$

There have been many meta-heuristic algorithms applied to this problem, such as GAS, EO, GA, HS. The results are shown in Table 14. Apparently, IICO is not the best, while it is a big improvement over ICO.

Table 14. Best solutions for the welded beam design problem.

Algorithm	Optimal solution				f_{min}
	x_1	x_2	x_3	x_4	
IICO	0.1601	4.5519	9.0114	0.2068	1.7928
ICO [31]	0.204	7.1773	9.0367	0.20573	2.2241
GSA [18]	0.1821	3.85697	10	0.20237	1.879952
EO [3]	0.2057	3.4705	9.0366	0.2057	1.7249
GA [38]	N/A	N/A	N/A	N/A	1.8245
HS [39]	0.2442	6.2231	8.2915	0.2433	2.3807
WOA [18]	0.20539	3.48429	9.03742	0.20627	1.730499

4.3. Pressure Vessel design

The pressure vessel design is an engineering optimization problem with the objective to minimize the total cost of material, forming and welding. There are four variables in the problem containing thickness of shell $T_s = x_1$, thickness of head $T_h = x_2$, inner radius $R = x_3$, and length of shell $L = x_4$. T_s and T_h are integer multiples of 0.0625 inch, which are the available thickness of rolled steel plates, and R and L are continuous. The structure of the problem is described below.

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

subject to four constraints

$$g_1 = -x_1 + 0.0193x_3 \leq 0,$$
$$g_2 = -x_2 + 0.00954x_3 \leq 0,$$
$$g_3 = -\pi x_3^2 x_4 - \frac{3}{4}\pi x_3^3 + 1296000 \leq 0,$$
$$g_4 = x_4 - 240 \leq 0 \tag{32}$$

where $0 \leq x_1, x_2 \leq 99$ and $10 \leq x_3, x_4 \leq 200$.

Table 15. Best solutions for the pressure vessel design problem.

Algorithm	Optimal solution				f_{min}
	x_1	x_2	x_3	x_4	
IICO	0.7857	0.38847	40.7103	194.6561	5899.637
ICO [31]	0.78168	0.38639	40.5017	197.4812	5891.383
GSA [18]	1.125	0.625	55.9886	84.4542	8538.836
EO [3]	0.8125	0.4375	42.0984	176.6365	6059.7143
GA [38]	0.8125	0.4345	40.3239	200	6288.745
PSO [40]	0.8125	0.4375	42.0912	176.7465	6061.078
WOA [18]	0.8125	0.4375	42.0982	176.6389	6059.741

Meta-heuristic algorithms have been widely used to solve this problem. Table 15 shows the results of the comparison. In the existing literature, the minimum values obtained by ICO, GSA, EO, GA, PSO, and WOA are 5891.383, 8538.836, 6059.7143, 6288.745, 6061.078 and 6059.741 respectively. Of these, ICO has the best results. IICO obtains the minimum cost of 5899.637. It is slightly worse than the results of ICO, while better than the other five algorithms. The adaptive parameter strategy speeds up the process of an algorithm from exploration to exploitation, which means that the exploration time of the algorithm is reduced. Although OBL has the effect of enhancing population diversity in the exploration phase, it may obtain relatively poor results when dealing with some complex problems. Therefore, the results of IICO are slightly worse than ICO.

5. Discussion and conclusions

In this paper, an improved intelligent chaotic clonal optimizer based on adaptive parameter strategy (IICO) is proposed. The improvement introduces the opposition-based learning (OBL) strategy to enhance the solution accuracy and the diversity of the population. The adaptive parameter strategy can effectively prevent from the stagnation of the global optimal update and accelerate the convergence speed of the algorithm. The experiment results of twenty-seven benchmark functions and three engineering optimization problems show that IICO has better comprehensive performance. And it's superior to other algorithms in terms of solution accuracy and convergence speed. Although the adaptive parameter strategy prevents the stagnation of the optimal value update, it also reduces the time of exploration. This is not a problem when dealing with benchmark functions. However, when solving some complex problems, the results are not desired. For example, in the above experiments, IICO is no better than ICO when addressing CEC test functions and the third engineering optimization problem. Solving this problem will be a future research direction. At the same time, meta-heuristic algorithms have a wide range of applications in the field of UAVs. Some

research work is also underway. In the future, the application of IICO to the UAV path planning problem will be considered.

Acknowledgements

The authors would like to thank the supports of the following projects: 1) The scientific research team project of Jingchu University of technology with grant number TD202001. 2) The Outstanding Youth Science and Technology Innovation Team Project of Colleges and Universities in Hubei Province with grants T201923. 3) The special research fund for joint training of graduate students of Jingchu University of Technology with grant number YJS202204.

Conflict of interest

All authors declare no conflicts of interest in this paper.

Reference

1. J. H. Holland, *Genetic Algorithms*, *Sci. Am.*, **267** (1992), 66–73. https://doi.org/10.1038/scientificamerican0792-66
2. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing, *Science*, **220** (1983), 671–680. https://doi.org/10.1126/science.220.4598.671
3. A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: a novel optimization algorithm, *Knowledge Based Syst.*, **191** (2020), 105190. https://doi.org/10.1016/j.knosys.2019.105190
4. R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, *Comput.-Aided Des.*, **43** (2011), 303–315. https://doi.org/10.1016/j.cad.2010.12.015
5. E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in *2007 IEEE Congress on Evolutionary Computation*, (2007), 4661–4667. https://doi.org/10.1109/CEC.2007.4425083
6. Q. Zhang, R. Wang, J. Yang, K. Ding, Y. Li, J. Hu, Collective decision optimization algorithm: a new heuristic optimization method, *Neurocomputing*, **221** (2017), 123–137. https://doi.org/10.1016/j.neucom.2016.09.068
7. 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
8. 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
9. G. G. Wang, Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, *Memetic Comput.*, **10** (2018), 151–164. https://doi.org/10.1007/s12293-016-0212-3
10. M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *IEEE Comput. Intell. Mag.*, **1** (2006), 28–39. https://doi.org/10.1109/MCI.2006.329691

11. D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *J. Global Optim.*, **39** (2007), 459–471. https://doi.org/10.1007/s10898-007-9149-x
12. 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
13. 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
14. 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
15. K. Zervoudakis, S. Tsafarakis, A mayfly optimization algorithm, *Comput. Ind. Eng.*, **145** (2020), 106559. https://doi.org/10.1016/j.cie.2020.106559
16. S. Mirjalili, SCA: a Sine Cosine Algorithm for solving optimization problems, *Knowledge Based Syst.*, **96** (2016), 120–133. https://doi.org/10.1016/j.knosys.2015.12.022
17. 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
18. 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
19. D. Whitley, A genetic algorithm tutorial, *Stat. Comput.*, **4** (1994), 65–85. https://doi.org/10.1007/BF00175354
20. A. Cheraghalipour, M. Hajiaghaei-Keshteli, M. M. Paydar, Tree Growth Algorithm (TGA): a novel approach for solving optimization problems, *Eng. Appl. Artif. Intell.*, **72** (2018), 393–414. https://doi.org/10.1016/j.engappai.2018.04.021
21. I. Rechenberg, Evolution strategy: nature's way of optimization, in *Optimization: Methods and Applications, Possibilities and Limitations*, (1989), 106–126. https://doi.org/10.1007/978-3-642-83814-9_6
22. R. Storn, K. Price, Differential evolution – A simple and efficient heuristic for global Optimization over continuous spaces, *J. Global Optim.*, **11** (1997), 341–359. https://doi.org/10.1023/A:1008202821328
23. 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
24. H. R. Tizhoosh, Opposition-based learning: a new scheme for machine intelligence, in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, (2005), 695–701. https://doi.org/10.1109/CIMCA.2005.1631345
25. W. Guo, P. Xu, F. Dai, F. Zhao, M. Wu, Improved Harris hawks optimization algorithm based on random unscented sigma point mutation strategy, *Appl. Soft Comput.*, **113** (2021), 108012. https://doi.org/10.1016/j.asoc.2021.108012
26. T. Si, P. B. C. Miranda, D. Bhattacharya, Novel enhanced Salp Swarm Algorithms using opposition-based learning schemes for global optimization problems, *Expert Syst. Appl.*, **207** (2022), 117961. https://doi.org/10.1016/j.eswa.2022.117961

27. A. G. Hussien, An enhanced opposition-based Salp Swarm Algorithm for global optimization and engineering problems, *J. Ambient Intell. Hum. Comput.*, **13** (2022), 129–150. https://doi.org/10.1007/s12652-021-02892-9
28. W. Wang, L. Xu, K. Chau, Y. Zhao, D. Xu, An orthogonal opposition-based-learning Yin–Yang-pair optimization algorithm for engineering optimization, *Eng. Comput.*, **38** (2022), 1149–1183. https://doi.org/10.1007/s00366-020-01248-9
29. A. Aleti, I. Moser, A systematic literature review of adaptive parameter control methods for evolutionary algorithms, *ACM Comput. Surv.*, **49** (2017), 1–35. https://doi.org/10.1145/2996355
30. Z. Lei, S. Gao, S. Gupta, J. Chen, G. Yang, An aggregative learning gravitational search algorithm with self-adaptive gravitational constants, *Expert Syst. Appl.*, **152** (2020), 113396. https://doi.org/10.1016/j.eswa.2020.113396
31. V. Sahargahi, V. Majidnezhad, S. T. Afshord, Y. Jafari, An intelligent chaotic clonal optimizer, *Appl. Soft Comput.*, **115** (2022), 108126. https://doi.org/10.1016/j.asoc.2021.108126
32. S. Rahnamayan, H. R. Tizhoosh, M. M. A. Salama, Quasi-oppositional differential evolution, in *2007 IEEE Congress on Evolutionary Computation*, (2007), 2229–2236. https://doi.org/10.1109/CEC.2007.4424748
33. A. A. Ewees, M. A. Elaziz, E. H. Houssein, Improved grasshopper optimization algorithm using opposition-based learning, *Expert Syst. Appl.*, **112** (2018), 156–172. https://doi.org/10.1016/j.eswa.2018.06.023
34. R. Tanabe, A. S. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in *2014 IEEE Congress on Evolutionary Computation (CEC)*, (2014), 1658–1665. https://doi.org/10.1109/CEC.2014.6900380
35. A. W. Mohamed, A. A. Hadi, A. M. Fattouh, K. M. Jambi, LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC2017 benchmark problems, in *2017 IEEE Congress on Evolutionary Computation (CEC)*, (2017), 145–152. https://doi.org/10.1109/CEC.2017.7969307
36. K. Deb, An efficient constraint handling method for genetic algorithms, *Comput. Methods Appl. Mech. Eng.*, **186** (2000), 311–338. https://doi.org/10.1016/S0045-7825(99)00389-8
37. S. Das, P. N. Suganthan, Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems, 2010. Available from: https://al-roomi.org/multimedia/CEC_Database/CEC2011/CEC2011_TechnicalReport.pdf.
38. C. A. C. 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
39. K. S. Lee, Z. W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, *Comput. Methods Appl. Mech. Eng.*, **194** (2005), 3902–3933. https://doi.org/10.1016/j.cma.2004.09.007
40. 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

AIMS Press