Applied Intelligence (2022) 52:16718–16757
https://doi.org/10.1007/s10489-021-03037-3

Chaotic arithmetic optimization algorithm

Xu-Dong Li[1] · Jie-Sheng Wang[1] · Wen-Kuo Hao[1] · Min Zhang[1] · Min Wang[1]

Accepted: 24 November 2021 / Published online: 25 March 2022

Abstract
Arithmetic Optimization Algorithm (AOA) is a meta-heuristic algorithm. Its main idea is to use the distribution behavior of the four main mathematical operators of addition(A), subtraction(S), multiplication(M) and division(D). Chaotic mapping strategy was introduced into the optimization process of AOA. Firstly, ten chaotic maps are separately embedded into two parameter Arithmetic Optimization Accelerator (MOA) and Arithmetic Optimization Probability (MOP) that affect the exploration and balance of AOA so as to enhance its ergodicity and non-repeatability, and improve its convergence speed and accuracy. Then a combination test was carried out by embedding ten chaotic maps into MOA and MOP at the same time, and their advantages and disadvantages were compared with the chaotic maps embedded separately. 26 benchmark functions in CEC-BC-2017 are used to examine the performance of the proposed chaotic arithmetic optimization algorithm (CAOA). Finally, four engineering design issues are optimized, involving three-bar truss design problem, welded beam design problem, pressure vessel design problem and spring design problem. The experimental results reveal that CAOA can obviously solve the function optimization and engineering optimization problems. AOA based on the chaotic interference factors has the merit of balancing the exploration and exploitation in the optimization process and enhances the convergence accuracy.

Keywords Arithmetic optimization algorithm · Chaotic map · Function optimization · Engineering optimization

1 Introduction

In recent decades, with the development of society, a series of problems and challenges have arisen, and people's demand for reliable optimization technology is also increasing [1]. Optimization is the process of finding the best solution among all available solutions for a specific problem. Considering the nature of optimization algorithms [2], these algorithms can be roughly divided into two kinds, namely definite algorithms and random intelligent algorithms [3]. In the case of a deterministic algorithm, if the initial value is the same when solving the same problem, the same solution will be produced. These are gradient searching methods, strictly towards the optimal solution. Contrary to deterministic algorithms, haphazard intelligent algorithms are usually non-gradient techniques, in which irregular steps are used to obtain the optimal value. In this case, the optimization process is repetitive in any situation. Random intelligent optimization algorithms can be further divided into heuristic algorithms and meta-heuristic algorithms [3]. As the name implies, heuristic algorithm is the course of finding a solution through trial and error method [4]. Meta-heuristic optimization algorithms have two vital searching tactics: exploration and exploitation [5]. Exploration is the ability to probe the searching room on a global scale. When the algorithm tries to find a promising area in the searching space as widely as possible, the exploration phase occurs [6]. This capability is related to avoiding partial optima and solving partial optima. On the contrary, the development is to explore the vicinity to improve the local optimization ability [7]. The outstanding performance of the algorithm requires an apposite balance between these two traits [8–11]. All swarm-based intelligent optimization algorithms adopt these traits, but use unequal operation operators and searching mechanisms. Many heuristic algorithms are utilized in the field of optimization. For instance, the Bat Swarm Optimization Algorithm (BSO) [5], Hill Climbing [6] and Simulated Annealing [12]. In short, the idea of

✉ Jie-Sheng Wang
wang_jiesheng@126.com

[1] School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114044, China

meta-heuristic algorithm is to use the prior knowledge of random search to solve the optimization problem randomly. This is an iterative optimization process, starting from a random initial solution and then randomly exploring and using the searching space with a clearly probability. So far, many meta-heuristic algorithms have been proposed by scholars such as Particle Swarm Optimization (PSO) [13], Genetic Algorithm (GA) [14], Rat Swarm Optimizer (RSO) [15], Archimedes Optimization Algorithm(AOA) [16], Equilibrium Optimizer (EO) [17], Ant lion optimizer (ALO) [18, 19], Gravitational Search Algorithm (GSA) [20], Harmony Search (HS) [21], Covariance Matrix Adaptive Evolution Strategy (CMA-ES) [22], Butterfly Optimization Algorithm (BOA) [23], Gray Wolf Optimization (GWO) [24] and Grasshopper Optimization Algorithm (GOA) [25], and so on.

With the continuous research and development of nonlinear theory, chaos theory has been widely used to improve exploration and exploitation [26]. As one of the characteristics of nonlinear systems, chaos is defined as randomness generated by simple real systems in the field of mathematics [27, 28]. So far, chaos theory has successfully solved the premature convergence problem of many meta-heuristic algorithms [29, 30]. Some scholars have added chaos mapping mechanism to different meta-heuristic algorithms to enhance the algorithm's ability to search for optimal solutions, enhance random diversification and strengthen the ability to achieve optimal or sub-optimal solutions in complex multi-modal situations [31]. In fact, many algorithms have been improved by using chaotic mapping. Hassan, B.A integrates the chaotic forms of the sine cosine algorithm (SCA) and the firefly algorithm (FF), and proposes the chaotic sine cosine firefly algorithm to improve the convergence speed and efficiency [17]. Juliano Pierezan et al. proposed an improved COA (MCOA) method based on the chaotic sequence generated by Tinkerbell mapping to adjust the scattering and correlation probability [32]. M.S. Sanaj et al. embedded chaotic mapping in the squirrel search algorithm, and proposed four derivative search behaviors, which significantly improved the convergence speed and accuracy [33]. In order to enhance the capability of the water cycle algorithm (WCA), A. A. Heidari proposed a method of incorporating chaotic patterns in the random process of WCA [34]. S. Gupta proposed a chaotic grey wolf optimizer (GWO) based on opposition (OCS-GWO), which improves the performance of GWO [35]. S. Saha et al. proposed a quasi-opposite chaotic ant colony optimizer (QOCALO) for solving global optimization problems [36]. Gandomi et al. introduced chaos in the variant particle swarm optimization (PSO) algorithm to enhance the global search capability [37]. Gandomi et al. used chaotic mapping to adjust the gravitational coefficient and light absorption coefficient of the Firefly algorithm (FA) [29]. Arora et al. replaced the chaotic number to control the switching probability of the global and local search capabilities, and improved the performance of the butterfly optimization algorithm [38]. Han et al. used chaos to generate noisy signals combined with genetic algorithms and proved its effectiveness in reducing different types of noise [39]. Chen et al. proposed a WOA based on the quasi-opposite learning chaos mechanism (OBCWOA). Firstly, the chaos mechanism was used to generate the initial value to improve the convergence speed of the algorithm. Then the learning method based on opposition was used to balance the exploration and exploitation capabilities of the algorithm [27]. Saha et al. simplified the basic symbiosis search algorithm (SOS) and incorporated chaos into the local searching to form a chaotic SOS (CSOS), which improved the optimization accuracy and convergence mobility of the SOS algorithm, which have successfully solved the real-world power system problems [28]. Yu et al. controlled the steps of chaotic mapping through thresholds, and used the velocity inertia weights to synchronize the speed of the agents. This improves the stability and convergence speed of the bat algorithm. Finally, it was excellently applied to three engineering optimization problems [31].You et al. used chaotic sequences to obtain dynamic parameter settings in DE, and designed a chaotic local search (CLS) operator to solve DED problem to help DE effectively avoid premature convergence [40, 41]. Pan et al. proposed a chaotic local search algorithm with a probabilistic jumping scheme, and embedded it into the proposed harmony search (HS) algorithm to enhance its local search ability [42]. Ewees et al. integrated chaotic behavior into the locust optimization algorithm (GOA) to form the chaotic locust optimization algorithm (CGOA), which was used as a trainer for learning multi-layer perceptual neural networks (MLPNN) to improve its prediction accuracy [43]. Talatahari et al. used different chaotic operators to improve the movement steps of the imperialist competition algorithm and proved its superiority [44]. Talatahari et al. introduced chaos into the charging system to increase its global search mobility to achieve better global optimization [45]. Alatas used chaotic mapping to change the Big Bang, Big Crunch stage of the Big Crunch optimization algorithm, and achieved good results [28, 46]. In the artificial bee colony (ABC) algorithm, Wu et al. used chaotic searching behavior on the candidate food positions generated to improve the convergence characteristics and prevent ABC from falling into local optimum [47]. Jordehi embeds the chaotic map into the artificial immune algorithm to alleviate its premature convergence problem [48]. Li et al. used chaotic sequences to replace the random parameters in the catfish particle swarm algorithm, which greatly improved its performance [49]. Saremi adopts chaos operators to define the probability of biogeographic optimization algorithm selection, emigration and mutation, and the algorithm produces high performance [50]. Xiao et al. mapped chaos to the gravitational constant in the gravity search algorithm (GSA), avoiding the problem of premature convergence [51]. Niknam et al. improved the shuffled frog-leapfrog algorithm (SFLA) by using chaotic mapping to solve the multi-objective OPF problem [52]. Dharmbir Prasad et al. embedded the chaotic map into two parameters of whale optimization algorithm (WOA) to solve OPF problem better [53]. Gandomi et al. proposed four different chaotic bat algorithm variants for global optimization [54]. Mukherjee et al. used the chaotic krill algorithm to solve the optimal reactive power dispatch (ORPD) problem of the power system [55]. In addition, the chaos theory has applications in cryptography [56, 57], DNA computing [58], image processing [59], nonlinear circuits [60], etc. All these attempts have proved that in different fields of engineering application, this newly constructed algorithm has better performance and higher beneficial accuracy than its standard algorithm.

Arithmetic Optimization Algorithm (AOA) is a meta-heuristic algorithm that its main idea is to use the distribution behavior of the four main mathematical operators of addition(A), subtraction(S), multiplication(M) and division(D) [1]. This paper integrates 10 chaotic maps into AOA so as to extensively study the effectiveness of chaos lore in improving the exploration and exploitation. Our main intention is to map the chaotic sequence into two key parameters of AOA. Various chaotic mapping operators have been proposed to replace linear decreasing or increasing sequences, in this way to balance the exploration and exploitation of AOA. It aims to speed up the global convergence speed and prevent the algorithm from falling into the local optimum. Firstly, 10 chaotic maps are embedded into MOA and MOP respectively, and then they are embedded in MOA and MOP at the same time. Then 26 test functions in CEC-2017 are compared and optimized. At the same time, some winning algorithms optimized for CEC2017 are selected for comparison. Finally, four engineering design issues are optimized, involving three-bar truss design problem, welded beam design problem, pressure vessel design problem and spring design problem. The simulation results show that the improved algorithm has obvious advantages compared with the original algorithm.

2 Basic principles of arithmetic optimization algorithm (AOA)

The swarm-based algorithm starts its optimization process from a set of randomly generated candidate solutions. This set of generated solutions is incrementally improved via some optimization rules, and iteratively evaluated through a specific objective function. Since the swarm-based algorithm seeks to find the optimal solution to the optimization problem randomly, it cannot guarantee that one solution will be obtained in one run. However, for a given problem, through a sufficient number of random solutions and optimization iterations, the probability of obtaining the global optimal solution increases [23]. The arithmetic optimization algorithm (AOA) is proposed through the arithmetic operators in mathematical operations, namely addition (+), subtraction (−), multiplication (x) and division (/). AOA is a swarm-based meta-heuristic algorithm that can solve optimization problems without calculating its derivatives. Arithmetic is not only a basic part of number theory, but also the cornerstone of modern mathematics, similar to algebra, geometry, analysis. Arithmetic operators (addition, subtraction, multiplication, and division) are usually used to study numbers and are a traditional and classic calculation method [27]. We use these simple operators as mathematical optimization ideas to determine the best elements that meet certain criteria from a set of candidate solutions The use of arithmetic operators in arithmetic problems is the main inspiration proposed by AOA. In the following subsections, we will discuss the specific influence of the arithmetic operators (addition, subtraction, multiplication and division) in the arithmetic optimization algorithm (AOA) on the algorithm. Figure 2 shows the search strategy and hierarchical structure of arithmetic operators from outside to inside.

(1) **Initialization phase**

The optimization process of the arithmetic optimization algorithm (AOA) starts with a set of randomly generated candidate solutions, which is shown in Eq. (1). In the optimization process, an optimal candidate solution is obtained in each iteration, and the optimal solution is regarded as obtained optimal solution or the solution close to the optimal solution so far.

$$X = \begin{pmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,n} \end{pmatrix} \tag{1}$$

Before AOA starts to work, it first enters the searching phase, that is to say the exploration or exploitation phase.

Therefore, the mathematical optimization accelerator (MOA) determines its searching phase through calculation. The calculation formula is described as follows.

$$MOA(C\_Iter) = Min + C\_Iter \times \left(\frac{Max - Min}{M\_Iter}\right) \tag{2}$$

where, $MOA(C\_Iter)$ denotes the function value at the lth iteration; C_Iter represents the current iteration, which is between 1 and the maximum number of iterations M_Iter. Min and Max denote the minimum and maximum values of the accelerated function, respectively. At the same time, the mathematical optimizer probability (MOP) coefficient shown in Eq. (3) is introduced as follows:

$$MOP(C\_Iter) = 1 - \frac{C\_Iter^{1/\partial}}{M\_Iter^{1/\partial}} \tag{3}$$

where, $MOP(C\_Iter)$ denotes the function value at the lth iteration, ∂ a is a sensitive parameter, which defines the accuracy of iterative development. Then a random number $r1$ is set, whose range is between (0, 1). When $r1 > MOA$, the algorithm is under the exploratory stage and when $r1 < MOA$, the algorithm is under the exploration stage.

(2) **Exploration Stage**

In this searching stage, according to the arithmetic operators, the division (D) operator and the multiplication (M) operator are adopted to explore the searching mechanism. At the same time, the random number is set to determine whether it is updated by multiplication or division. Its location is updated by the following strategy.

$$X_{i,j}(C\_Iter+1) = \begin{cases} best(x_j) \div (MOP + \epsilon) \times \mu + LB_j, & r2 < 0.5 \\ best(x_j) \times MOP \times ((UB_j - LB_j) \times \mu + LB_j), & otherwise \end{cases} \tag{4}$$

where,$r2$is a random number between (0,1), which is used to switch multiplication and division operators; $X_{i,j}(C\_Iter)$ indicates the j-th position of the i-th solution in the operating iteration; $\boldsymbol{best}(x_j)$ represents the j-th position in the best solution so far; $\in$ is a smaller integer number; UB_j and LB_j denote to the upper bound and lower bound values of the j-th position respectively; μ is a control parameter to adjust the searching process, which is fixed equal to 0.5.

(3) **Exploitation Stage**

In this development stage, according to the arithmetic operators, when using subtraction or addition, the result of the operation is relatively dense. Unlike other operators, addition operators and subtraction operators have the characteristics of low dispersion, which determines that they can more easily approach the search target. Using search operators (S and A) often tries to avoid falling into the local search area. This process helps to explore the searching strategy to find the optimal solution and maintain the diversity of candidate solutions. Random values are generated in each iteration, whether it is the first iteration or the last iteration, it keeps exploring. This can avoid falling into a local optimal stagnation, especially in the final iteration. Its location is updated according to Eq. (5).

$$X_{i,j}(C\_Iter+1) = \begin{cases} best(x_j) - MOP \times ((UB_j - LB_j) \times \mu + LB_j), & r3 < 0.5 \\ best(x_j) + MOP \times ((UB_j - LB_j) \times \mu + LB_j), & otherwise \end{cases} \tag{5}$$

where, $r3$ represents a random number between (0,1), which is used to switch addition and subtraction search; $X_{i,j}(C\_Iter+1)$ indicates the j-th position of the i-th solution in the next iteration; $\boldsymbol{best}(x_j)$ represents the j-th position in the best solution so far; LB_j and UB_j denote the lower bound and upper bound values of the j-th position respectively; μ is a control parameter to adjust the searching process, which is fixed equal to 0.5.

In a nutshell, the optimization process of AOA starts by generating a set of random candidate solutions. In the iterative process, A S, D, and M estimate the feasible position close to the optimal solution, and each solution updates its position from the optimal solution. In order to better distinguish between exploration and development, the parameter MOA is linearly increased from 0.2 to 0.9. When r1 > MOA, the candidate solution seeks to deviate from the near-optimal solution. When r1 < MOA, the candidate solution seeks to converge to a near-optimal solution. In the end, AOA stops due to the end-point criterion being met. The searching strategy of AOA is shown in Fig. 1.

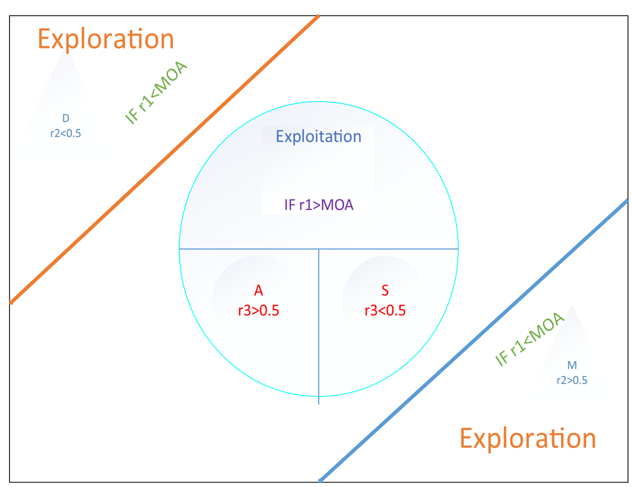


Fig. 1 Searching strategy of arithmetic optimization algorithm

3 Chaotic arithmetic optimization algorithm

3.1 Chaotic mapping

Generally speaking, chaos is a deterministic and random-like method that exists in non-linear, dynamic, aperiodic, non-convergent and bounded systems. From a mathematical point of view, chaos is a deterministic dynamic system of randomness, and chaotic systems can be considered as the source of randomness. [29]. The nature of chaos is obviously random and unpredictable, but it also contains certain laws in randomness. Chaos is based on chaotic variables, not random variables [38]. Therefore, it can perform a thorough search at a higher speed than a random search that mainly relies on probability [35]. Even for very long sequences, only a few functions (chaotic mapping) and a few parameters (initial conditions) are needed [61]. In addition, once the initial conditions are changed, many different sequences are generated. The characteristics of these sequences are reproducible and deterministic. It can be seen that it is very sensitive to parameters and initial conditions [62]. On the basis of chaos theory, various chaotic maps have been derived in the field of optimization [38, 63]. In the current research work, there are 10 chaotic maps that are widely used by scholars [64]. The visualization graphs of these ten chaotic maps are given below, as shown in Fig. 2, and their mathematical expressions are described as follows.

(1) **Chebyshev map (CH)**

The chaotic map expression is as follows:

$$x_{k+1} = cos\left(kcos^{-1}(x_k)\right) \tag{6}$$

(2) **Circle map (CI)**

The chaotic map expression is as follows:

$$x_{k+1} = \left(x_k + b - \left(\frac{a}{2\pi}\right)sin(2\pi x_k)\right)mod(1) \tag{7}$$

where, $a = 0.5$, $b = 0.2$.

(3) **Gauss map (GA)**

The chaotic map expression is as follows:

$$x_{k+1} = \begin{cases} 0 & x_{k=0} \\ \dfrac{1}{x_k mod(1)} & \textbf{\textit{otherwise}} \end{cases} \qquad \frac{1}{x_k mod(1)} = \frac{1}{x_k} - \left[\frac{1}{x_k}\right] \tag{8}$$

(4) **Iterative map (IT)**

The chaotic map expression is as follows:

$$x_{k+1} = sin\left(\frac{a\pi}{x_k}\right) \tag{9}$$

where, $a \in (0, 1)$.

(5) **Logistic map (LO)**

The chaotic map expression is as follows:

$$x_{k+1} = ax_k(1-x_k) \tag{10}$$

where, x_k is the $k-th$chaotic number, k is the number of iterations, $x \in (0, 1)$, $x_0 \notin (0, 0.25, 0.5, 0.75, 1)$, a is a real number, $a = 4$.

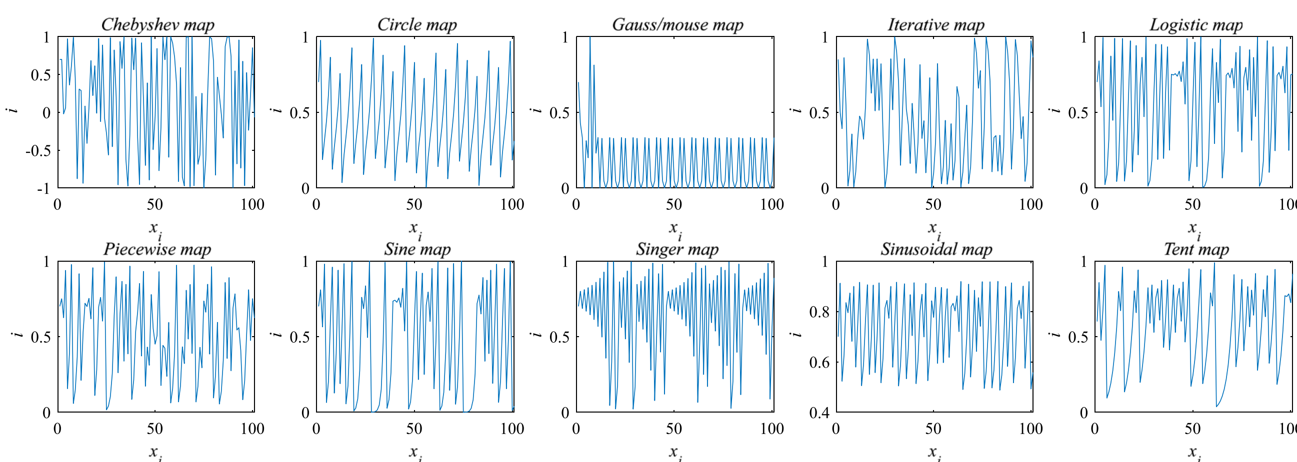


Fig. 2 Visualization graphs of ten chaotic mapping strategies

(6) **Piecewise map (PI)**

The chaotic map expression is as follows:

$$x_{k+1}=\begin{cases}\dfrac{x_k}{p} & 0\le x_k<p\\ \dfrac{x_k-p}{0.5-p} & p\le x_k<\dfrac{1}{2}\\ \dfrac{1-p-x_k}{0.5-p} & \dfrac{1}{2}\le x_k<1-p\\ \dfrac{1-x_k}{p} & 1-p\le x_k<1\end{cases} \tag{11}$$

where, $p \in (0, 1)$ is the control parameter, $x \in (0, 1)$.

(7) **Sine map (SI)**

The chaotic map expression is as follows:

$$x_{k+1}=\frac{a}{4}sin(\pi x_k) \tag{12}$$

where, $a \in (0, 4]$.

(8) **Singer map (SG)**

The chaotic map expression is as follows:

$$x_{k+1}=\mu\left(7.86x_k-23.31{x_k}^2+28.75x_k^3-13.302875x_k^4\right) \tag{13}$$

where, $\mu \in (0.9, 1.08)$.

(9) **Sinusoidal map (SO)**

The chaotic map expression is as follows:

$$x_{k+1}=ax_k^2sin(\pi x_k) \tag{14}$$

where, a=2.3.

(10) **Tent map (TE)**

The chaotic map expression is as follows:

$$X_{k+1}\begin{cases}\dfrac{x_k}{\alpha} & x_k<\alpha\\ \dfrac{(1-x_k)}{1-\alpha} & x_k\ge\alpha\end{cases} \tag{15}$$

where, $\alpha \in (0, 1)$.

3.2 Chaotic arithmetic optimization algorithm

3.2.1 MOA and MOP based on chaotic interference factors

Through the analysis of the original arithmetic optimization algorithm, two parameters (MOA and MOP) play an important role in the optimization process, and directly determine whether the algorithm is in the stage of exploration or exploitation. MOA determines which searching strategy the algorithm chooses, and MOP affects its location update. Figure 3 shows the changing trend of these two parameters. At the same time, the images of using chaotic factors to influence two parameters (MOA and MOP) respectively, which are shown in Figs. 4 and 5, respectively.

3.2.2 Flowchart of chaotic arithmetic optimization algorithm

In order to increase the randomness of searching operators, the chaotic interference factors are added in MOA so as to expand the possibility of searching for promising areas. At the same time, the chaotic interference factor is added to affect the mathematical optimizer probability (MOP) in order to affect its position update, so that it has a certain volatility. This article takes Logistic map chaotic mapping as an example and embeds it in the parameters to update the position, the proposed chaotic interference factor are calculated by Eq. (16)–(20).

$$MOA=x_{t+1}\times \boldsymbol{Value1} \tag{16}$$

$$MOP=x_{t+1}\times \boldsymbol{Value2} \tag{17}$$

$$x_{t+1}=ax_t(1-x_t) \tag{18}$$

$$\boldsymbol{Value1}=\left(\frac{C\_Iter}{M\_Iter}\right)^{\frac{1}{6}} \tag{19}$$

$$\boldsymbol{Value2}=\boldsymbol{1}-\left(\frac{C\_Iter}{M\_Iter}\right)^{\frac{1}{2}} \tag{20}$$

where t is the current iteration number, $a = 4$, $x \in (0, 1)$, $x_0 \notin (0, 0.25, 0.5, 0.75, 1)$.

This paper adopts chaotic interference factors to improve AOA so as to designs three different chaotic AOA. Firstly, the chaotic interference factors are alone used to interfere with MOA. The second is to use chaotic interference factors alone to interfere with MOP. Finally, a combination test is performed, that is to say that the chaotic interference factors are used to interfere with MOA and MOP at the same time. The AOA flowchart is shown in Fig. 6. Let's take the last idea as an example and give the pseudo code of the improved AOA.

Initialize the Arithmetic Optimization Algorithm parameters α, μ.

Initialize the solutions' positions randomly. (Solutions: i = 1, ..., N.)

```
for (i=1 to Solutions) do
   for (j=1 to Positions) do
      Generate a random values between [0, 1] (r1, r2, and r3)
      if r1 >MOA then
        Exploration phase
        if r2 >0.5 then
          (1) Apply the Division math operator (D " ÷ ").
          Update the ith solutions' positions using the first rule in Eq. (3).
        else
          (2) Apply the Multiplication math operator (M " × ").
           Update the ith solutions' positions using the second rule in Eq. (3).
         end if
       else
         Exploitation phase
         if r3 >0.5 then
          (1) Apply the Subtraction math operator (S " − ").
          Update the ith solutions' positions using the first rule in Eq. (5).
         else
          (2) Apply the Addition math operator (A " + ").
          Update the ith solutions' positions using the second rule in Eq. (5).
         end if
       end if
    end for
 end for
 C_Iter=C_Iter+1
end while
  Return the best solution (x).
```

3.2.3 Computational complexity of chaos arithmetic optimization algorithm

This section analyzes the complexity of the Chaos Arithmetic Optimization Algorithm (CAOA) proposed in this paper. Its computational complexity is mainly composed of three aspects, namely the dimension D of the problem, the population size N and the maximum number T of iterations [65]. According to the optimization process of CAOA, the number of operations at each step will be calculated. First, in the initialization phase, the time complexity of initializing N search agents in the D-dimensional search space is $O(2ND)$. Secondly, the function fitness value of N individual agents needs to be calculated, and the time complexity of the fitness function is $O(ND)$. When selecting the best fitness value, a quick sort algorithm is adopted, and its computational complexity is $O(\boldsymbol{NlogN})$ and $O(N2)$ in the best and worst cases respectively. The computational complexity of MOA and MOP are $O((N-1)^2)$. In the exploration phase of CAOA, the time complexity of all individual location updates and boundary control strategies is $O(2ND)$. In the development phase of CAOA, the time complexity of all individual location updates and boundary control strategies is $O(2ND)$. The simplified definition of overall complexity is described as follows.

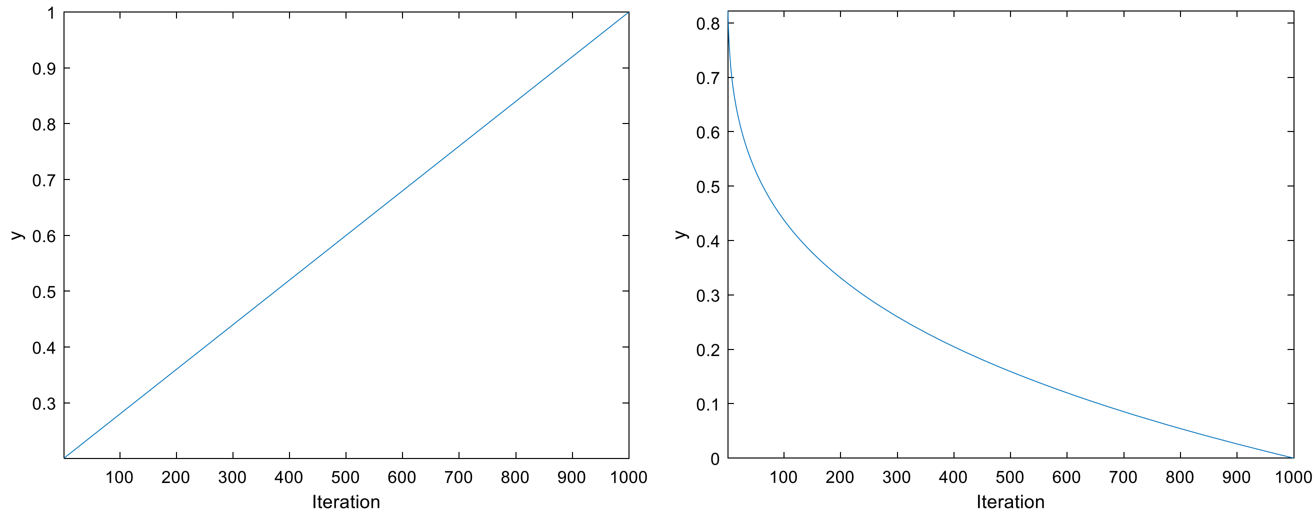


Fig. 3 MOA and MOP operators

$$O(2ND) + O(ND) + O(N2) + O(TlogN) + O\left(T(N-1)^2\right) + O(2TND) + O(2TND) \approx O(T*(N2 + 4ND)) \quad (21)$$

4 Simulation experiments and result analysis

4.1 Function optimization

This paper selects 26 test functions in CEC-BC-2017 to test the performance of chaotic arithmetic optimization algorithms. These test functions are divided into four categories: unimodal function, multimodal function, mixed function and composite function. Through the optimization results of these four testing functions, the superiority of the chaotic AOA is fully proved. In order to ensure the fairness and reliability of the test experiment, the population size of the arithmetic optimization algorithm and the arithmetic optimization algorithm based on chaotic mapping is unified to 30, and the maximum number of iterations is unified to 1000. The specific parameter settings of each algorithm are shown in Table 1.

4.1.1 CEC2017 testing functions

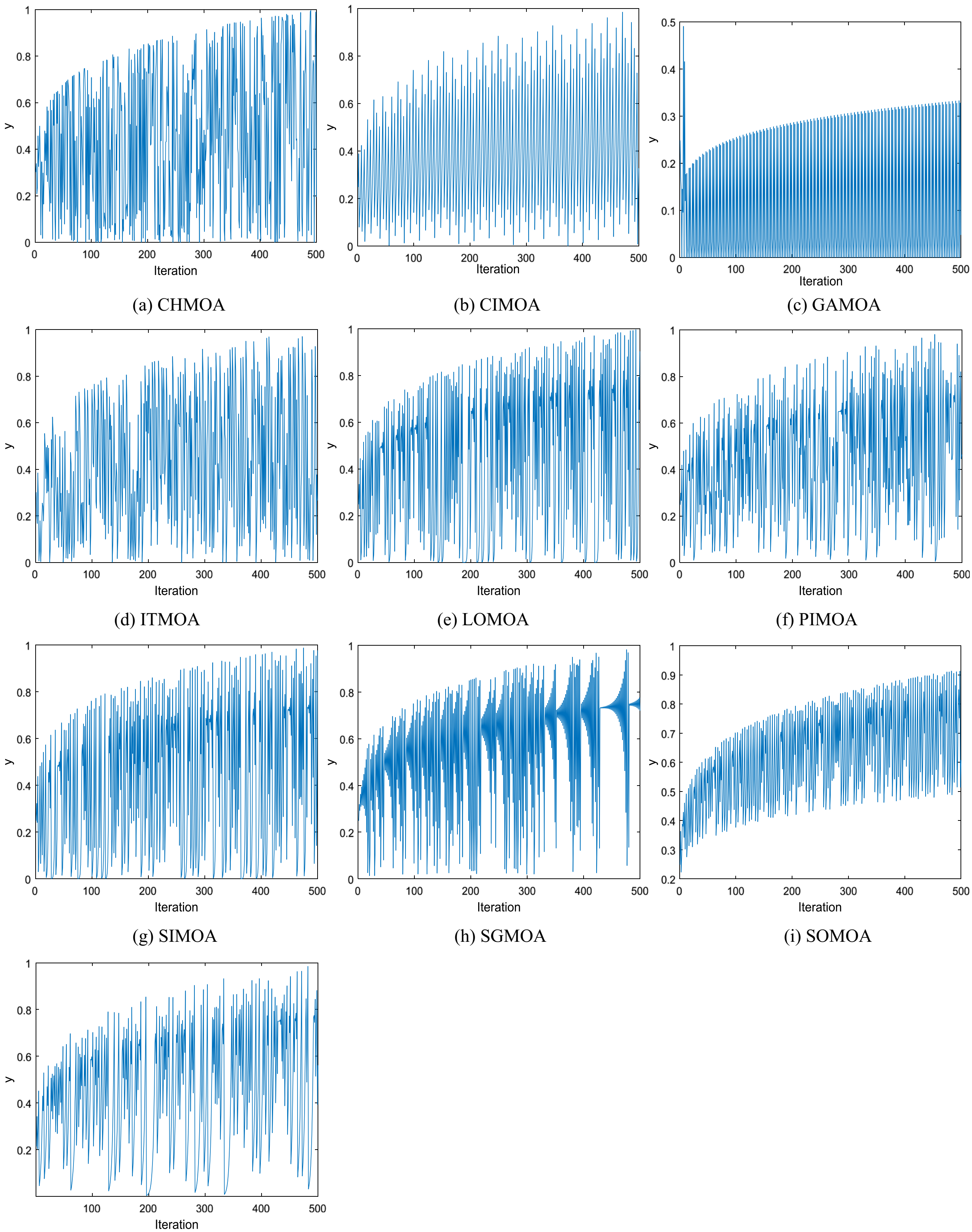
This paper uses 26 functions listed in the CEC-BC-2017 [66–68]. The dimension of these testing functions is 10. Table 2 lists the expression of the function. Range means the range of the function is [−100,100], fmin means the optimal solution of the function. These functions are divided into four categories: unimodal functions f_1-f_3, multimodal functions f_4-f_{10}, mixed functions f_{11}, f_{14}-f_{20}, and composite functions f_{21}-f_{23}, f_{25}-f_{28}.

4.1.2 Chaotic arithmetic optimization algorithm to optimize CEC-BC-2017 testing functions

For the selected 26 CEC-BC-2017 test functions, simulation experiments are carried out by adopting three proposed improved AOA. The dimension of all functions is 10, each algorithm is run 10 times and the optimal solution among ten times is recorded. This paper performs mathematical statistics on the experimental results to facilitate the comparison of the performance of the arithmetic optimization algorithm and the chaotic arithmetic optimization algorithm proposed in this paper. In order to analyze the experimental results more intuitively, this article chooses the average and variance for statistics. The obtained variance, optimal, average values are listed in Table 3, 4 and 5. The convergence curves of the experimental simulation are shown in Figs. 7, 8 and 9 respectively. It can be seen from figures and tables that the chaotic AOA has better convergence effect on most function optimizations than the standard AOA.

(1) **MOA with Chaos Factors**

Figure 7 shows the the convergence curves obtained by using AOA with MOA to optimize test functions. Table 3 is the experimental results obtained by simulation. From the results in Table 3, we can get the following conclusions. In terms of mean and minimum,the average values of GAAOA on function f_3 and f_4 are the closest to the optimal value. At the same time, the optimal and average values of function f_1, f_2, f_3, f_4, f_7, f_8, f_{14}, f_{23} and f_{28} obtained by GAAOA are the best. The



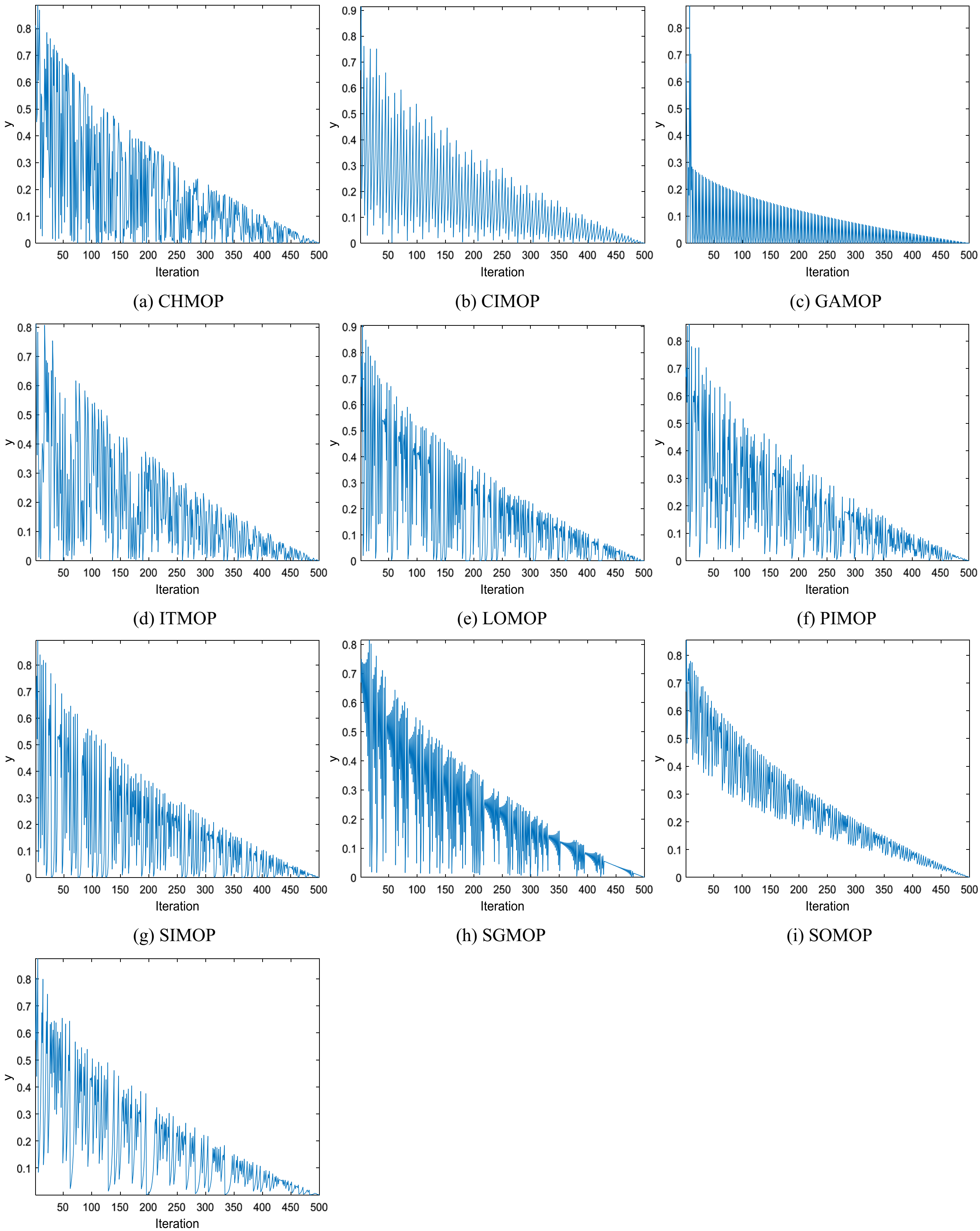
(a) CHMOA (b) CIMOA (c) GAMOA

(d) ITMOA (e) LOMOA (f) PIMOA

(g) SIMOA (h) SGMOA (i) SOMOA

(j) TEMOA

Fig. 4 MOA with chaotic factors



(a) CHMOP
(b) CIMOP
(c) GAMOP
(d) ITMOP
(e) LOMOP
(f) PIMOP
(g) SIMOP
(h) SGMOP
(i) SOMOP
(j) TEMOP

Fig. 5 MOP with chaotic factors

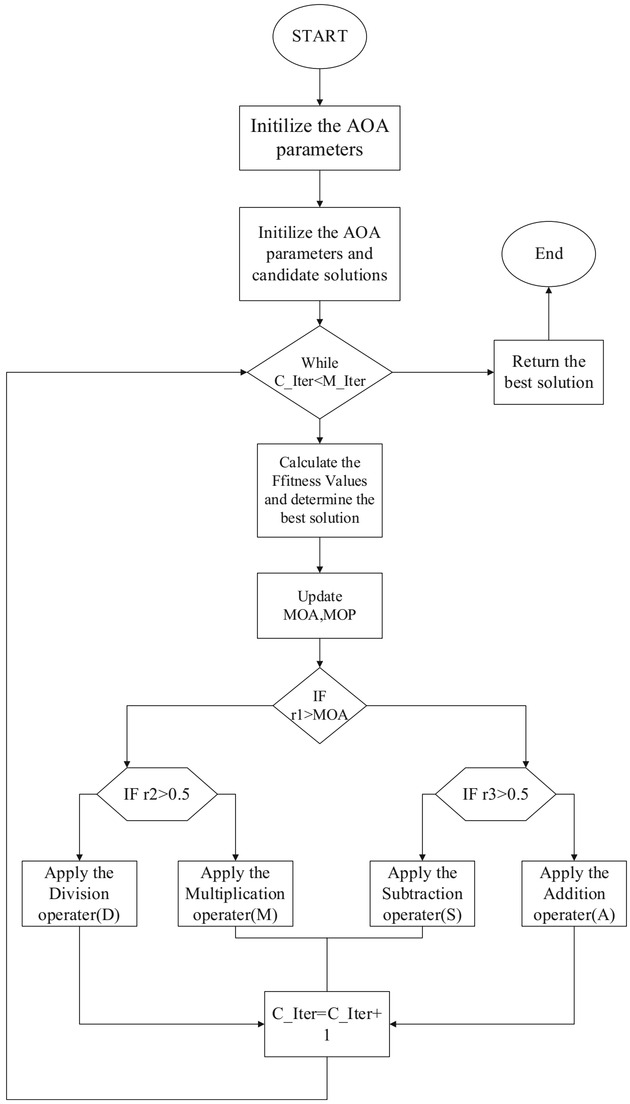


Fig. 6 Flowchart of AOA

optimal and average value of function f_{10} and f_{11} obtained by SIAOA are the smallest. The optimal and average values obtained by ITAOA to optimize the function f_6 and f_{25} are the smallest. In terms of average value, CHAOA to optimize function f_5, PIAOA to optimize function f_9, LOAOA to optimize function f_{15} and SIAOA to optimize function f_{16} obtain the smallest average value. TEAOA only has a minimum value on function f_{18}. In terms of the minimum and optimal values on function f_{15} and f_{26}, CHAOA obtains the smallest among the improved algorithms. For function f_9 and f_{16}, ITAOA has the best minimum value. TEAOA to optimize function f_{22}, SGAOA to optimize function f_5 and CIAOA to optimize function f_{18} all achieve the minimum value. GAAOA only shows slight superiority on f_6, f_9 and f_{26}. At the same time, this paper has made a ranking of the optimization performance of each test function. When considering embedding the chaotic map into the parameter MOA separately, comparing their average rankings, we can intuitively see that GAAOA has the best optimization performance and ranks the first. On the other hand, SIAOA and CHAOA rank the second and third, respectively.

(2) **MOP with Chaos Factors**

Figure 8 shows the convergence curves obtained by using AOA with the chaotic MOP to optimize test functions. Table 4 is the experimental results obtained by simulation. From the results in Table 4, we can get the following conclusions. In terms of mean and minimum, the average values of CHAOA on function f_6, ITAOA on function f_8 and SIAOA on function f_{11} are the closest to the optimal value. Among them, the average, optimal and standard deviation values obtained by PIAOA on function f_2 are the smallest. The optimal and average value of function f_6 and f_7 obtained by CHAOA are the best. At the same time, the average and optimal values of SIAOA on function f_{11}, SOAOA on function f_3, PIAOA on function f_{24} and GAAOA on function f_{27} are the smallest. Seen from the mean and standard deviation values together, the average and standard deviation values of CHAOA on

Table 1 Parameter settings of all algorithms

Algorithm	Main parameters Settings
AOA	population size: N=30, Maximum iterations: T=1000, α=5, μ=0.499, MOP_{max}=1, MOP_{min}=0.2
CHAOA	population size: N=30, Maximum iterations: T=1000, α=5, μ=0.499, MOP_{max}=1, MOP_{min}=0.2
CIAOA	population size: N=30, Maximum iterations: T=1000, α=5, μ=0.499, MOP_{max}=1, MOP_{min}=0.2
GAAOA	population size: N=30, Maximum iterations: T=1000, α=5, μ=0.499, MOP_{max}=1, MOP_{min}=0.2
ITAOA	population size: N=30, Maximum iterations: T=1000, α=5, μ=0.499, MOP_{max}=1, MOP_{min}=0.2
LOAOA	population size: N=30, Maximum iterations: T=1000, α=5, μ=0.499, MOP_{max}=1, MOP_{min}=0.2
PIAOA	population size: N=30, Maximum iterations: T=1000, α=5, μ=0.499, MOP_{max}=1, MOP_{min}=0.2
SIAOA	population size: N=30, Maximum iterations: T=1000, α=5, μ=0.499, MOP_{max}=1, MOP_{min}=0.2
SGAOA	population size: N=30, Maximum iterations: T=1000, α=5, μ=0.499, MOP_{max}=1, MOP_{min}=0.2
SOAOA	population size: N=30, Maximum iterations: T=1000, α=5, μ=0.499, MOP_{max}=1, MOP_{min}=0.2
TEAOA	population size: N=30, Maximum iterations: T=1000, α=5, μ=0.499, MOP_{max}=1, MOP_{min}=0.2

Table 2 CEC-BC-2017 test functions (Dim = 10)

Type	No.	Functions	Range	fmin
Unimodal Function	f_1	Shifted and Rotated Bent Cigar function	[−100,100]	100
	f_2	Shifted and Rotated sum of Differential Power Function	[−100,100]	200
	f_3	Shifted and Rotated Zakharov Function	[−100,100]	300
Multimodal Functions	f_4	Shifted and Rotated Rosenbrock's Function	[−100,100]	400
	f_5	Shifted and Rotated Rastrigin's Function	[−100,100]	500
	f_6	Shifted and Rotated Expanded Scaffer's Function	[−100,100]	600
	f_7	Shifted and Rotated Lunacek Bi_Rastrigin Function	[−100,100]	700
	f_8	Shifted and Rotated Non-Continuous Rastrigin's Function	[−100,100]	800
	f_9	Shifted and Rotated Levy Function	[−100,100]	900
	f_{10}	Shifted and Rotated Schwefel's Function	[−100,100]	1000
Hybrid Functions	f_{11}	Hybrid Function of Zakharov, Rosenbrock and Rastrigin's	[−100,100]	1100
	f_{14}	Hybrid Function of Bent Ciagr, Rosenbrock and Lunache Bi-Rastrigin	[−100,100]	1400
	f_{15}	Hybrid Function of Eliptic, Ackley, Schaffer and Rastrigin	[−100,100]	1500
	f_{16}	Hybrid Function of Bent Cigar, HGBat, Rastrigin and Rosenbrock	[−100,100]	1600
	f_{17}	Hybrid Function of Expanded Schaffer, HGBat, Rosenbrock and Modified Schwefel	[−100,100]	1700
	f_{18}	Hybrid Function of Katsuura, Ackley, Expanded Griewank plus Rosenbrock, Modifed Schwefel and Rastrigin	[−100,100]	1800
	f_{19}	Hybrid Function of Bent Cigar, Rastrigin, Expanded Grienwank plus Rosenbrock, Weierstrass and expanded Schaffer	[−100,100]	1900
	f_{20}	Hybrid Function of Happycat, Katsuura, Ackley, Rastrigin, Modified Schwefel and Schaffer	[−100,100]	2000
Composition Functions	f_{21}	Composition Function of Rosenbrock, High Conditioned Elliptic and Rastrigin	[−100,100]	2100
	f_{22}	Composition Function of Rastrigin's, Griewank's and Modifed Schwefel's	[−100,100]	2200
	f_{23}	Composition Function of Rosenbrock, Ackley, Modified Schwefel and Rastrigin	[−100,100]	2300
	f_{24}	Composition Function of Rastrigin, Happycat, Ackley, Discus and Rosenbrock	[−100,100]	2400
	f_{25}	Composition Function of Expanded Scaffer, Modified Schwefel, Griewank, Rosenbrock and Rastrigin	[−100,100]	2500
	f_{26}	Composition Function of HGBat, Rastrigin, Modified Schwefel, Bent-Cigar, High Conditioned Elliptic and Expanded Scaffer	[−100,100]	2600
	f_{27}	Composition Function of Ackley, Griewank, Discus, Rosenbrock, HappyCat, Expanded Scaffer	[−100,100]	2700
	f_{28}	Composition Function of shifted and rotated Rastrigin, Expanded Scaffer and Lunacek Bi_Rastrigin	[−100,100]	2800

function f_{28}, GAAOA on function f_{14}, LOAOA on function f_9 and f_{17} are the best. It means they are more stable. SOAOA only has a minimum value on function f_5, f_9 and f_{23}. From the average point of view, GAAOA is not as effective as the original algorithm on function f_1, f_2 and f_3. SOAOA only shows slight superiority on f_9, f_{17} and f_{20}. At the same time, this paper has made a ranking of the performance of each algorithm optimization test functions. When considering embedding the chaotic map into the parameter MOP separately, comparing their average rankings, we can intuitively see that SIAOA has the best optimization performance and ranks the first. LOAOA and CHAOA rank the second and third, respectively.

(3) **MOA and MOP with Chaos Factors**

Figure 9 shows the convergence curves obtained by using chaotic factors to interfere with the MOA and MOP at the same time to optimize testing functions. Table 5 is the experimental results obtained by simulation. From the results in Table 5, we can get the following conclusions. In terms of mean and minimum, the average values of ITAOA on f_3,LOAOA on f_{16},SIAOA on f_5 are the closest to the optimal value. It is worth mentioning that the average, optimal and standard deviation values obtained by ITAOA to optimize function f_1, and f_2 are the smallest. CHAOA optimizing functionf_3 and f_{14}also gets the smallest average value, optimal value and standard deviation. At the same time, the optimal and average values of function f_{11} obtained by TEAOA are the best. Looking at the mean and standard deviation values together, the average and standard deviation of CHAOA on function f_4,f_9, f_{15},f_{17},$f_{20}$and f_{26}, TEAOA on function f_{11},ITAOA on function $f_{27}$are the smallest. It shows that they are more stable. GAAOA only has a minimum value on function f_4. Meanwhile, from the average point of view, GAAOA only shows slight superiority on f_2,f_{11},f_{15} and f_{23}. At the same time, this article has made a ranking of the performance of

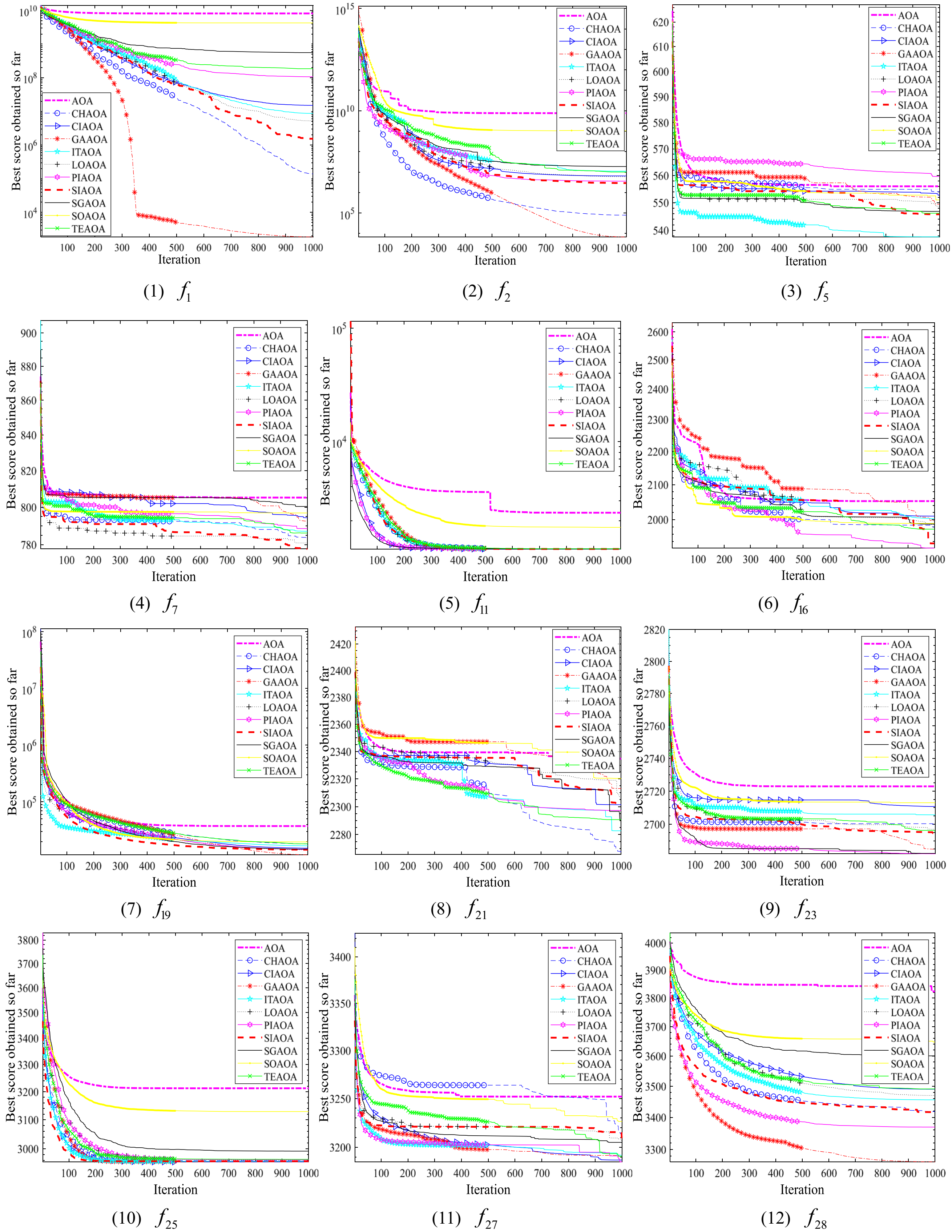


Fig. 7 AOA with chaotic MOA to optimize testing functions

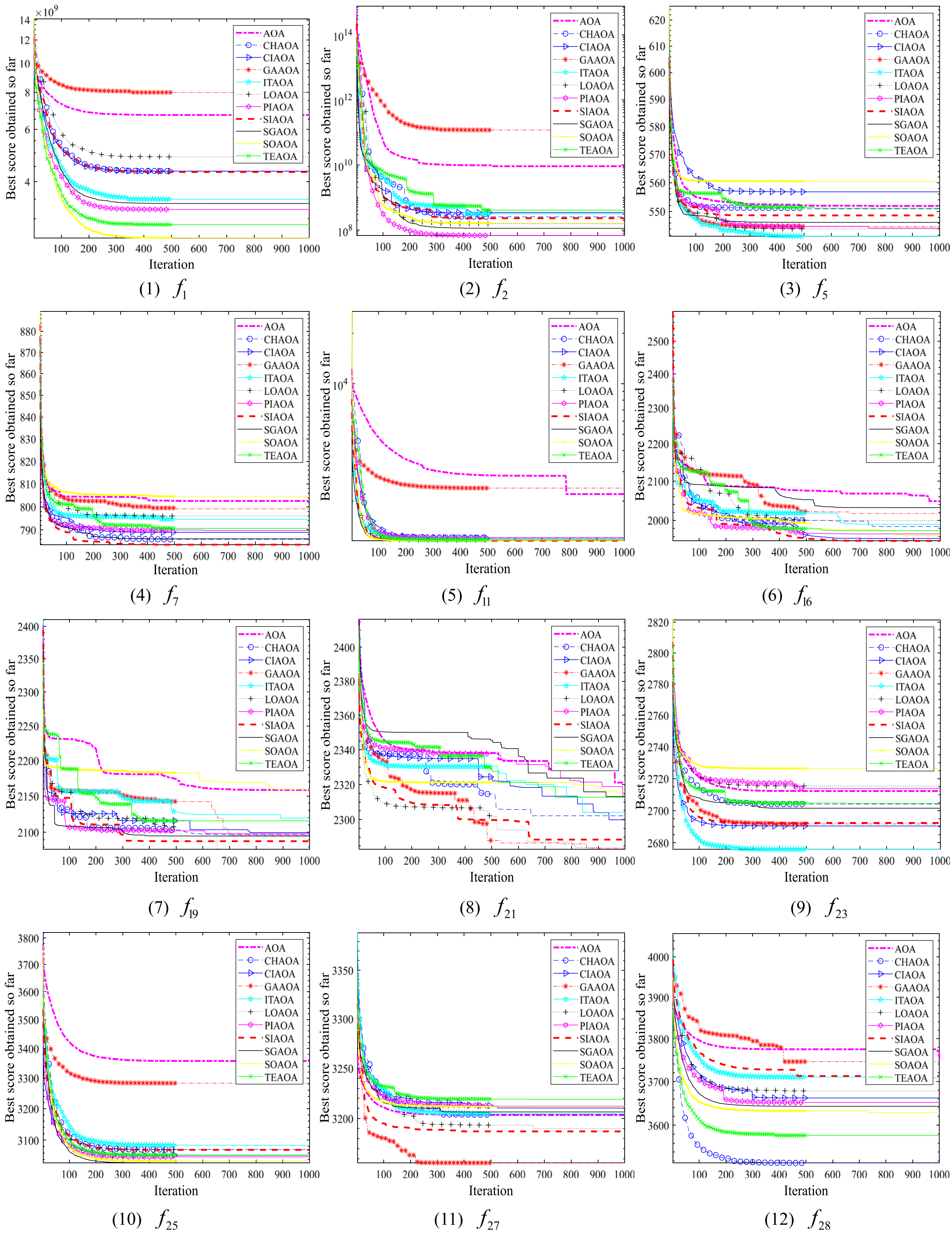


Fig. 8 AOA with Chaotic MOP to optimize testing functions

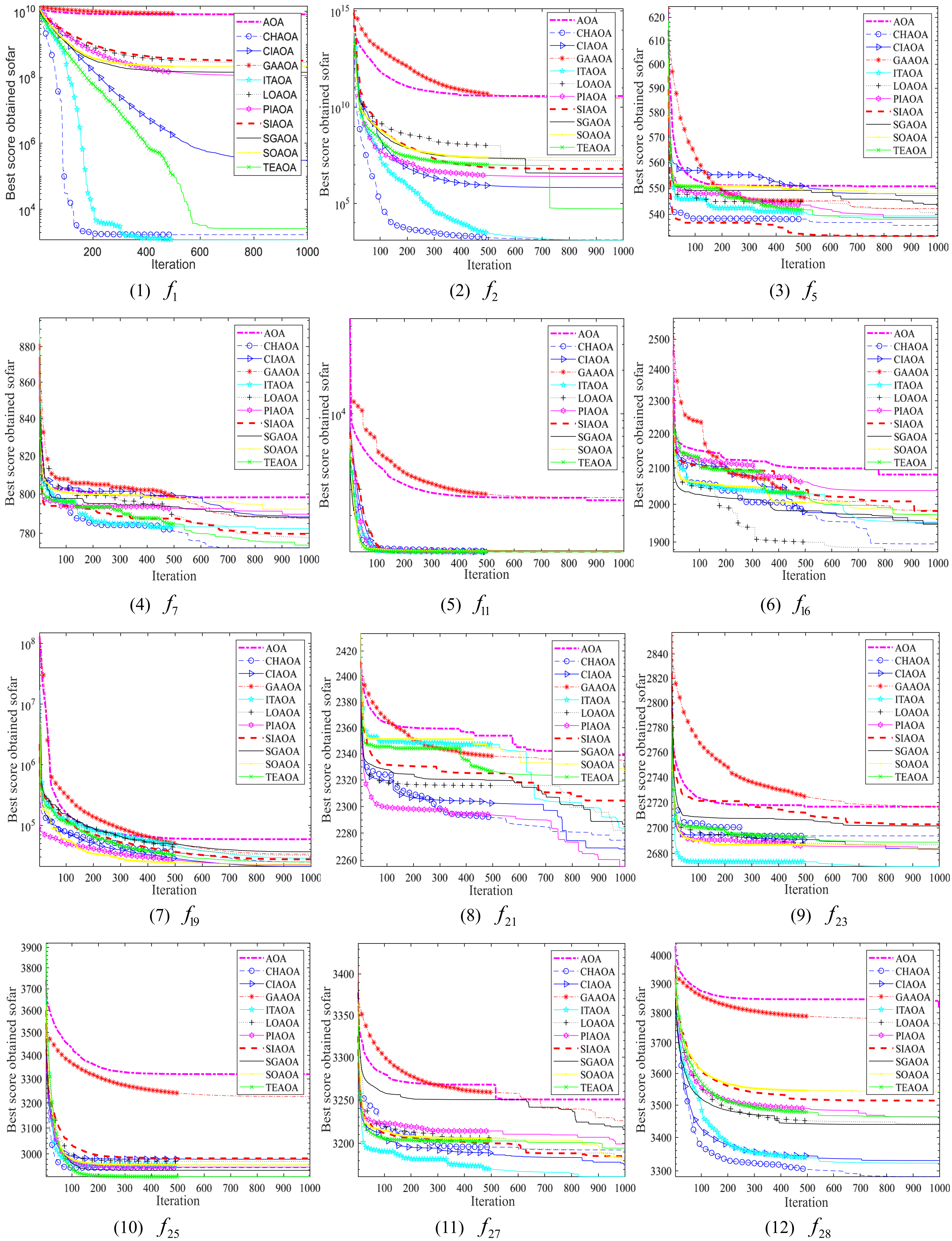


Fig. 9 AOA with Chaotic MOA and MOP to optimize testing functions

Table 3 Simulation results of AOA with chaotic MOA to optimize CEC-2017 testing functions

Function		AOA	CHAOA	CIAOA	GAAOA	ITAOA
f_1	Best	3,683,810,598	185.3004054	110.3044866	**103.3711634**	346.2030512
	Ave	8,162,957,094	139,254.5071	15,165,882.23	**1864.837899**	8,760,305.247
	Std	2,814,744,273	434,813.2414	29,301,967.38	**2900.622098**	18,058,432.83
	Rank	10	6	2	1	11
f_2	Best	389,355,242.9	209.1141742	5668.580546	**200.079654**	1077.10634
	Ave	7,633,263,181	78,822.74429	6,494,176.046	**6753.254535**	11,073,778.99
	Std	7,028,006,627	170,446.1416	15,645,821.49	**12,318.43075**	20,518,776.37
	Rank	11	2	6	1	8
f_3	Best	4601.495596	300.0000084	300.0000197	**300.0000005**	300.0000214
	Ave	9550.397919	673.0436673	800.9841229	**300.000001**	1655.719428
	Std	3039.15514	758.1256258	730.4127262	**3.21946E-07**	1498.806152
	Rank	11	2	4	1	7
f_4	Best	466.0689261	401.5929818	403.2269107	**400.0181002**	407.9195165
	Ave	955.9437181	422.8177571	445.7737222	**414.7993117**	442.1913379
	Std	407.775313	26.06338726	45.31811497	28.91640428	34.13919432
	Rank	11	3	8	1	7
f_5	Best	538.1096046	519.8991422	519.4337939	525.0936247	531.8385741
	Ave	563.4269808	**539.1200305**	548.7307259	546.1022251	544.5985368
	Std	17.11600517	12.84224582	23.52714077	18.17406124	14.24810081
	Rank	11	1	10	6	3
f_6	Best	633.6586981	627.369635	623.946677	625.3766694	**622.1882213**
	Ave	644.9554358	632.8377271	633.2933586	636.3976798	**631.3280856**
	Std	9.575686252	4.707845724	**4.255554157**	5.706305074	6.516880107
	Rank	11	4	5	7	1
f_7	Best	788.9563828	765.1593907	776.8098285	**736.1722758**	752.7569772
	Ave	804.6406986	788.5224525	797.5638614	**772.2432881**	782.0569102
	Std	**8.319044782**	11.28709371	10.05987911	15.87750461	15.08051024
	Rank	11	7	10	1	2
f_8	Best	825.9002547	814.6520562	819.5324528	**812.9344577**	813.9294172
	Ave	833.1694918	822.297889	822.3091822	**822.2142925**	825.0963825
	Std	5.958937407	6.691025283	**2.243042024**	7.166221689	9.08040025
	Rank	11	2	3	1	8
f_9	Best	1336.805718	1048.78425	1062.120726	1044.663271	**1001.960735**
	Ave	1508.054789	1243.87939	1344.026982	1255.045898	1321.209718
	Std	121.0573424	125.5338426	180.4511525	174.0799018	154.5882077
	Rank	11	2	8	3	6
f_{10}	Best	1794.437275	1448.848641	1573.09453	1683.183061	1502.257552
	Ave	2214.8135	1825.541604	1947.204643	1990.011519	1837.074005
	Std	260.7518488	258.8025168	234.2070572	170.8436274	215.3510406
	Rank	11	2	8	6	3
f_{11}	Best	1188.161209	1121.704855	1113.742666	1120.920602	1121.688566
	Ave	2638.787193	1124.128629	1126.029777	1123.995418	1204.624671
	Std	2412.359726	**1.383468965**	10.38837475	2.892759244	180.9553136

Table 3 (continued)

	Rank	11	3	6	2	10
f_{14}	Best	1683.041071873	1485.688393381	1675.465341691	**1453.504749890**	1474.473583768
	Ave	9978.857262350	7583.690714238	7852.414109425	**6764.357685453**	9100.515204553
	Std	9091.091499944	**6358.217019455**	7365.212003076	8229.003992866	8960.475923454
	Rank	8	3	5	1	6
f_{15}	Best	10,802.17719	**3054.148664**	5948.122397	6547.083485	5141.90745
	Ave	18,050.13206	15,383.69888	11,994.45575	18,465.67675	11,141.15989
	Std	3245.89836	8507.836652	5226.126588	12,308.44854	4257.754031
	Rank	9	8	4	11	3
f_{16}	Best	1895.066419	1842.087818	1853.031697	1820.733247	**1629.870477**
	Ave	2169.523881	1979.214237	2008.965702	2058.617728	2012.278822
	Std	145.2116247	86.99097775	**76.53001051**	145.2990179	145.895736
	Rank	11	6	7	10	8
f_{17}	Best	1767.696703	1750.943901	1755.513663	**1725.499099**	1751.530718
	Ave	1903.854047	1842.642307	1838.892262	1788.645572	1868.472124
	Std	107.0938891	118.7569622	67.60770897	54.10059248	115.7720567
	Rank	11	6	5	2	9
f_{18}	Best	8466.838168	7552.287454	**2290.721216**	3275.737473	3648.611451
	Ave	192,846.0981	13,668.28787	13,099.60915	16,155.16307	16,754.67402
	Std	556,763.3031	**6605.704471**	9253.448675	8851.279356	7580.386511
	Rank	11	3	2	8	10
f_{19}	Best	48,832.18067	13,714.58827	13,519.15443	9530.38496	9416.560918
	Ave	48,832.18067	13,714.58827	13,519.15443	9530.38496	9416.560918
	Std	27,891.75251	9215.168506	10,259.45094	8283.293237	9184.591328
	Rank	11	6	5	3	2
f_{20}	Best	2072.058983	2071.057425	2051.243293	2053.835309	2067.274971
	Ave	2168.550084	2116.658391	2109.265135	2105.706264	2119.352471
	Std	55.96873087	52.39872487	49.93538169	37.27336776	47.79341163
	Rank	11	8	6	5	9
f_{21}	Best	2301.487861	2226.370911	2251.092665	**2200.00002**	2206.892585
	Ave	2332.998023	2288.875538	2311.870647	2306.218465	**2284.413573**
	Std	28.72673086	41.7358049	53.31626198	58.42066094	44.44672011
	Rank	11	3	10	9	2
f_{22}	Best	2748.312064	2413.380909	2485.700769	2421.801446	2387.649431
	Ave	2929.979518	2545.758218	2643.090684	**2507.425097**	2575.779427
	Std	157.7357065	115.4070882	182.6171275	**59.36024676**	151.3052908
	Rank	11	3	8	1	5
f_{23}	Best	2695.036858	2673.489122	2644.782788	**2638.207**446	2653.872876
	Ave	2725.491937	2699.18677	2706.11923	**2678.667323**	2683.350635
	Std	33.26215806	21.95321187	30.18893021	30.98340371	**17.51389361**
	Rank	11	6	**7**	1	2
f_{24}	Best	2787.302558	2727.347649	2751.632846	2677.577643	2500.002078
	Ave	2848.181265	2830.121578	2817.725204	2827.273793	2759.233632
	Std	49.75846273	61.90999566	**39.13725495**	61.05742114	139.6032611
	Rank	10	9	7	8	3

Table 3 (continued)

f_{25}	Best	3014.224086	2899.74951	2900.148107	2899.60999	**2600.225292**
	Ave	3325.343811	2961.170933	2999.498764	2959.976115	**2923.447015**
	Std	207.942139	**27.35951915**	48.86192264	28.30592802	122.9067513
	Rank	11	6	9	4	1
f_{26}	Best	3523.923005	**2600.010504**	2600.025601	2800.001989	2800.010627
	Ave	3883.015807	3462.787075	3588.231639	3383.51141	**3209.426022**
	Std	403.1387807	406.5411552	427.3912926	385.1467326	**255.5424079**
	Rank	9	4	6	3	1
f_{27}	Best	3177.009746	3125.521984	3132.915335	3119.45974	3114.764851
	Ave	3252.052831	3198.593512	3193.845081	**3174.438202**	3186.851363
	Std	60.46487863	66.11297783	45.90216654	49.24664572	33.34203239
	Rank	11	6	5	1	4
f_{28}	Best	3646.155672	3259.621588	3100.002903	**3100.000856**	3166.010984
	Ave	3766.064595	3495.507163	3342.431882	**3259.888897**	3343.481499
	Std	**80.4147404**	163.7277079	230.250325	120.4022374	181.8053346
	Rank	11	7	2	1	3
Average Rank		11	3	7	1	5

Function	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
f_1	233.0152338	180.9009596	109.7556982	2571.646141	1,805,197,536	2783.603757
	5,245,126.213	107,244,761.8	1,542,521.301	560,312,781.1	4,268,584,950	185,881,316.5
	16,580,263.34	158,264,843.3	4,870,456.706	462,028,308.4	2,077,607,238	458,520,664.2
	9	3	7	5	8	4
f_2	257.3943218	5617.497508	242.4917007	6638.265257	64,736.26209	10,995.02519
	5,983,829.865	3,613,183.357	2,922,045.313	18,774,531.57	1,039,776,315	10,080,938.04
	9,686,273.932	8,719,477.208	6,655,722.226	36,117,717.46	2,267,219,922	11,886,163.5
	5	4	3	9	10	7
f_3	300.0000125	350.2650384	300.0000114	696.5118598	2668.377354	430.6231922
	687.0407436	1459.752221	826.0703909	3269.736707	6070.356333	1677.054532
	593.8952114	945.7032206	774.4892299	2333.682858	3138.971993	1095.083477
	3	6	5	9	10	8
f_4	400.0471235	400.1896433	406.3812811	409.3878536	464.9436178	401.7562419
	431.0910787	439.1481991	422.1923325	466.2287884	616.9759088	439.3609625
	22.96836519	31.34159876	25.66424235	61.49656806	164.4040937	31.90321134
	4	5	2	9	10	6
f_5	536.8133472	530.9223997	530.6811708	**514.9243916**	517.9115432	517.3299624
	547.2822525	545.6101295	548.2155076	544.6876383	543.0472916	546.9086146
	8.611833257	15.51776423	12.54435984	12.89315129	11.66694533	18.1047323
	8	5	9	4	2	7
f_6	622.6263871	627.7880316	624.3640835	623.6799363	623.5977691	632.817474
	631.4619557	637.2666721	631.4982666	633.7235087	637.2364078	638.9245861
	5.813890124	5.80203229	5.488509258	6.730999803	6.652327012	6.329559781
	2	9	3	6	8	10
f_7	767.2300832	767.474019	757.842942	773.7499873	775.6269144	765.9137665

Table 3 (continued)

	787.7985939	782.3054783	787.3547628	792.9327388	797.2763255	784.2509507
	13.54143352	11.94282143	16.24372633	13.36021928	13.46369591	14.04697173
	6	3	5	8	9	4
f_8	817.5904784	819.8991713	815.9193307	816.9143139	817.9109318	817.9092537
	824.5152882	825.95937	823.2769118	824.3552208	826.675798	824.178491
	5.391601884	4.707974431	4.701446254	4.069785395	4.766679302	4.117345484
	7	9	4	6	10	5
f_9	1020.483832	1041.265653	1052.175772	1020.404745	1096.237748	1238.828965
	1381.918812	**1193.979423**	1258.811904	1261.914379	1338.146359	1371.58008
	205.3179994	167.7101439	137.4613101	151.0993315	172.5277532	**93.37442956**
	10	1	4	5	7	9
f_{10}	1457.493794	1376.613385	**1355.749286**	1533.29615	1460.165483	1675.394667
	1847.25025	1916.964344	**1654.924965**	1967.194271	2010.693722	1885.40972
	292.8025268	277.1534694	294.1274118	252.4973859	358.2761598	**135.5120411**
	4	7	1	9	10	5
f_{11}	1117.844742	1120.970046	**1105.851421**	1113.16267	1148.595383	1113.824301
	1125.002814	1125.589575	**1122.185023**	1153.257918	1194.611326	1130.578368
	3.423666489	4.325850653	6.157263444	39.10488951	51.93317475	21.28415476
	4	5	1	8	9	7
f_{14}	1477.284122234	1465.972942466	2212.294434569	1486.935656388	1676.211883939	2520.016353835
	9507.816046238	11,351.54277200	12,979.38959577	7775.926826971	7537.047976700	10,868.01008742
	8389.252323883	10,731.67569045	9848.749632840	8006.997135805	6941.654297228	9559.5603494945
	7	10	11	4	2	9
f_{15}	4563.420031	4729.587339	3314.394813	9788.670913	4080.17788	9887.139606
	9297.590511	12,154.91787	10,448.6965	13,955.53394	14,614.76674	18,246.66381
	2825.990519	8539.447076	6242.391384	**2717.043135**	6696.638687	8690.860757
	1	5	2	6	7	10
f_{16}	1629.449965	1733.842864	1738.176764	1738.157447	1929.183605	1745.254491
	1973.352708	1931.375227	**1915.988334**	1918.330637	2048.638414	1956.575434
	192.2915729	131.6972995	141.5673644	131.4110728	84.05929104	98.45294021
	5	3	1	2	9	4
f_{17}	1735.771353	1746.665084	1752.395509	1739.812047	1744.007649	1766.340547
	1765.492224	1849.492125	1831.475483	1824.072869	1843.371649	1883.017107
	34.5640712	102.9155946	89.13551396	80.47883872	94.64474174	92.85990858
	1	8	4	3	7	10
f_{18}	2955.635174	2654.208326	3303.923862	2524.515928	2772.320161	3882.961535
	15,183.57597	**9680.825172**	15,293.20565	14,729.92493	16,222.83953	14,827.76399
	12,059.33115	7877.266006	9108.668485	12,862.7962	10,103.43967	9159.570062
	6	1	7	4	9	5
f_{19}	**8443.27941**	10,808.5264	16,917.40138	19,694.29885	20,879.22346	17,541.40995
	8443.27941	10,808.5264	16,917.40138	19,694.29885	20,879.22346	17,541.40995
	6677.960279	11,054.35885	10,121.05741	16,784.11398	17,524.73787	12,353.93478
	1	4	7	9	10	8
f_{20}	2050.726817	**2039.477925**	2050.762668	2077.10976	2050.196315	2050.424088
	2095.129553	2103.39953	2102.541614	2113.754822	2121.075382	**2071.490808**

Table 3 (continued)

	43.28220918	40.66911798	53.69241403	30.39619424	59.67458896	**20.72642027**
	2	4	3	7	10	1
f_{21}	2202.545145	2248.78964	2246.052119	2204.529925	2303.072665	2208.051754
	2234.862744	2290.925631	2295.910635	2296.127133	2330.226222	2292.366912
	28.81505681	35.8505314	41.90871112	52.22633005	**15.92810192**	65.10769839
	1	4	6	7	8	5
f_{22}	2445.267945	2407.323129	2410.883861	2495.742971	2468.516425	**2353.34386**
	2585.426966	2607.738875	2565.074223	2647.740669	2736.870685	2526.953429
	86.31431203	197.410025	123.836065	120.5353634	254.3419709	98.6794578
	6	7	4	9	10	2
f_{23}	2659.913106	2660.784337	2652.516179	2657.474866	2666.775711	2685.232341
	2689.002596	2711.780534	2694.503193	2706.770922	2695.982346	2714.596749
	27.75440267	28.98282605	29.0677651	31.24734558	19.75378523	19.76466077
	3	9	4	8	5	10
f_{24}	2500.002476	2615.069584	**2500.002017**	2672.335059	2583.539122	2500.008484
	2750.726835	2899.406627	**2738.266059**	2800.367549	2767.806208	2792.650443
	135.6405135	90.43097709	171.4603128	65.630169	112.8358777	114.2700701
	2	11	1	6	4	5
f_{25}	2897.934844	2897.991416	2897.89855	2897.991503	2993.704445	2899.077482
	2960.468808	2955.509833	2953.389951	2971.820116	3133.520765	2967.202184
	35.18411339	43.59345844	34.89223703	33.08735083	140.8846736	52.13062085
	5	3	2	8	10	7
f_{26}	3046.979385	3399.560671	2800.020097	2800.117951	3412.499934	3114.507525
	3619.655074	3883.719001	3352.962724	3526.7472	3973.238896	3676.527172
	357.1617082	302.5057498	520.1133126	387.4014151	343.0751334	419.5748545
	7	10	2	5	11	8
f_{27}	3129.967802	**3113.532981**	3122.123885	3169.916296	3166.564949	3178.432689
	3200.350016	3182.061969	3184.457522	3210.631173	3234.207861	3212.452801
	56.1113348	51.68114963	34.26279262	40.09007354	54.42671782	**27.2313267**
	7	2	3	8	10	9
f_{28}	3186.999804	3149.050113	3235.950251	3287.742862	3389.241599	3201.675263
	3386.748234	3387.136062	3497.530959	3537.52255	3704.20605	3440.472434
	190.3538819	170.953794	133.3449538	161.6115765	128.5983837	176.8108053
	4	5	8	9	10	6
Average Rank	4	6	2	9	10	8

Table 4 Simulation results of AOA with chaotic MOP to optimize CEC-2017 testing functions

Function		AOA	CHAOA	CIAOA	GAAOA	ITAOA	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
f_1	Best	1,754,586,286	951,770,695.9	650,666,146.2	1,747,761,872	328,475,396.3	699,529,674.3	563,189,295.7	1,014,165,990	602,729,791.9	294,719,170.9	**24,393,794.94**
	Ave	6,711,166,866	4,346,872,893	4,342,671,816	7,993,567,481	3,491,301,514	4,851,148,112	3,226,219,826	4,313,491,515	3,378,700,618	**2,589,027,953**	2,866,756,150
	Std	3,598,446,553	2,553,485,187	2,456,576,029	3,337,956,723	2,293,502,380	2,095,705,262	1,996,938,170	2,205,283,463	**1,490,064,605**	1,749,657,678	2,285,105,107
	Rank	10	8	7	11	5	9	3	6	4	1	2
f_2	Best	372,934,215.1	20,669.14228	25,382.40551	164,933,760	792,437.938	17,342.63504	**8402.041439**	62,971.04742	25,181.49952	12,995.13788	133,203.114
	Ave	9,095,141,785	252,966,183.7	341,177,999.1	1.17745E+11	314,403,413	155,705,207.7	**67,507,880.36**	229,993,279.1	112,498,102.9	162,231,776	406,770,430.7
	Std	7,876,488,626	505,205,356.8	658,337,485.6	2.0377E+11	396,626,910.8	168,225,671.2	**85,749,541.98**	306,365,066.6	168,189,360.9	482,927,474.3	1,044,242,612
	Rank	10	6	8	11	7	3	1	5	2	4	9
f_3	Best	3584.626013	5573.100498	2691.21905	8994.953191	3448.595217	4878.240288	3107.467709	4132.942062	2769.063115	**1876.175208**	2714.568082
	Ave	8816.55241	7456.478177	8518.179732	11,686.46822	7738.418602	8020.232318	7906.036349	8258.020679	7587.074682	**6929.894465**	7941.91391
	Std	3366.290691	**1846.387435**	3344.235963	2646.507192	2796.371257	2021.840119	2557.55437	2539.495647	2265.21392	3307.669535	2908.457035
	Rank	10	2	9	11	4	7	5	8	3	1	6
f_4	Best	510.82352	442.4558396	477.4213966	471.9576118	444.3715961	**419.599412**	479.6385802	428.4354806	458.529853	437.9692575	449.5706378
	Ave	969.6507934	560.9654006	609.8036038	834.9377738	661.2483081	574.1020452	589.5530387	548.7726037	**536.0522912**	582.4106073	581.1195299
	Std	354.6189048	83.50896901	126.8091634	315.211577	229.4310652	120.5186802	**60.66665391**	112.5647241	98.43644899	146.8747804	118.862437
	Rank	11	3	8	10	9	4	**7**	2	1	6	5
f_5	Best	**511.9658614**	520.0459677	529.9588087	523.3903763	516.0689774	521.3190627	525.8988202	522.943013	529.9222956	525.9013536	522.9274549
	Ave	551.9418801	550.9193782	556.8338291	544.8585094	**541.5834043**	544.0721706	544.3253471	548.6683959	546.2947423	560.4816758	551.1570876
	Std	18.03315995	17.95968641	19.08864099	12.99318828	18.08241738	16.01721933	17.52464181	21.83774333	**10.66286286**	23.16966299	20.12240176
	Rank	9	7	10	4	1	2	3	6	5	11	8
f_6	Best	631.9599785	**616.7970283**	626.1033775	623.8868312	621.0461033	623.4866058	624.172911	618.0360687	622.4248463	626.3280731	631.9154007
	Ave	644.7125366	**630.1722923**	634.977762	634.1422395	634.8481542	631.8637589	635.7902501	633.1661218	634.9723098	639.2609875	638.4427276
	Std	7.182076537	6.594646438	6.379092731	6.540644497	7.773671965	6.595835129	6.33229761	7.596332877	6.862325483	8.679446228	**4.470128977**
	Rank	11	1	7	4	5	2	8	3	6	10	9
f_7	Best	788.1811778	**746.0945853**	763.9375004	777.8796917	768.3656952	778.0592006	757.7530203	762.56859	746.489126	768.4209403	765.9389829
	Ave	802.5535017	786.0149823	789.0144586	799.1175547	794.6534885	796.0437991	789.8025304	**783.7357408**	786.2345098	804.5927508	790.734081
	Std	**8.844325355**	17.9445484	15.83338977	13.31639358	15.67573384	10.74455468	17.12149521	15.93865607	18.52583229	24.78578577	15.04032591
	Rank	10	2	4	9	7	8	5	1	3	11	6
f_8	Best	823.8932947	818.9964824	823.0866078	826.3587102	816.9459901	822.0117044	822.9000905	**814.6467626**	816.9629286	825.02981	822.9620022
	Ave	830.8269152	826.7494304	829.5268237	835.9997617	**824.8441816**	828.2620005	831.469806	829.9012368	829.2467296	832.5768098	830.5625047
	Std	4.856397346	6.11014976	**4.282068862**	6.653042177	4.616902541	5.879393073	7.510891809	8.713062449	10.81277074	7.091326957	5.637398678
	Rank	8	2	5	11	1	3	9	6	4	10	7
f_9	Best	1142.978403	1111.056189	1018.18365	1164.844676	1069.326189	1075.189324	1046.313079	**957.8355313**	1116.08177	1219.107464	1058.879854
	Ave	1411.000653	1304.617942	1275.240498	1373.297094	1298.701686	**1254.257437**	1269.39978	1298.562347	1386.77067	1403.777463	1330.045469
	Std	204.622372	138.2322081	154.6155738	180.5796165	186.7418887	**126.2755787**	148.037618	188.6046397	178.6540258	136.6786772	169.5733336

Table 4 (continued)

Function		AOA	CHAOA	CIAOA	GAAOA	ITAOA	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
	Rank	11	7	3	8	5	1	2	4	9	10	6
f_{10}	Best	1696.175619	1520.008755	1504.784934	1707.848405	1359.772018	1728.983667	**1340.816096**	1454.916581	1569.260481	1728.312267	1609.369548
	Ave	2057.142618	**1864.506496**	1924.888898	1908.27531	1944.212919	1966.491122	1877.537204	1927.465312	1937.41798	2101.407787	2023.716737
	Std	219.7920739	167.9458972	253.3860681	234.6905513	302.9041284	**133.3549679**	427.2781651	225.7662763	287.8844884	233.5710492	238.6671898
	Rank	10	1	4	3	7	8	2	5	6	11	9
f_{11}	Best	1185.579364	1139.046001	1164.889767	1189.051695	1124.377593	1127.84736	1137.40994	**1111.191135**	1127.213169	1133.373769	1123.866456
	Ave	2201.998642	1211.273858	1204.123261	2389.830219	1180.695928	1173.395399	1207.62031	**1158.90169**	1160.141069	1159.327403	1177.558627
	Std	1541.404727	56.48208215	30.9352787	939.3601122	45.26368622	37.8268407	57.83117349	35.46626479	34.75767415	**26.4586169**	50.42039773
	Rank	10	9	7	11	6	4	8	1	3	2	5
f_{14}	Best	1657.052824	1758.517289	1457.950615	2036.888737	1468.696146	1500.104599	1463.270951	**1449.085025**	1465.674932	1476.898694	1461.374567
	Ave	9043.853623	10,877.24235	11,742.12561	**5906.435497**	8426.571449	7213.806176	9292.050789	8470.408673	10,761.82926	10,714.57898	7620.159784
	Std	9909.409096	7135.24803	7978.59523	**4037.309439**	8219.470591	8649.738474	7330.076712	9070.057485	9161.928023	9339.122841	6304.744372
	Rank	6	10	11	1	4	2	7	5	9	8	3
f_{15}	Best	11,892.28985	**1948.080872**	2684.468086	2917.296854	2967.127706	4178.904072	4480.718199	3341.868234	4640.586945	5336.827303	4346.026224
	Ave	18,510.55716	10,954.14895	14,340.52657	**8951.890697**	11,267.38924	12,475.57568	20,655.74954	18,543.90075	11,823.52735	15,570.15083	16,821.43144
	Std	**3647.79785**	7557.652111	7942.74622	7550.274083	6285.341584	7624.167364	10,714.49211	14,255.31329	6655.755137	5098.412738	12,608.9752
	Rank	9	2	6	1	3	5	11	10	4	7	8
f_{16}	Best	1673.843352	1642.373257	1637.997721	1739.896935	**1630.555859**	1815.229242	1747.424105	1651.54761	1636.182397	1753.380424	1651.489211
	Ave	2048.43077	1984.598252	1955.339382	2016.99576	1990.059387	1999.592067	1966.361488	**1949.460424**	2032.055716	1962.319492	1974.390121
	Std	209.0200166	181.647751	124.3605265	161.6097415	154.7711369	**114.9298308**	153.4610132	155.0651278	177.6971835	175.5636295	179.6925661
	Rank	11	6	2	9	7	8	4	1	10	3	5
f_{17}	Best	1773.640838	1738.161289	1753.991239	1778.156842	1751.732739	1750.897054	1736.071583	1744.376631	1761.979829	1781.546689	**1735.568921**
	Ave	1877.417416	1807.40195	1812.864736	1889.484102	1805.54306	**1788.075069**	1811.923914	1808.586198	1807.236297	1895.957104	1797.518657
	Std	82.12175015	60.48567057	89.69786753	98.96291284	74.9630063	**27.53079862**	58.80863272	74.2713236	63.92670458	129.4449567	51.915473
	Rank	9	5	8	10	3	1	7	6	4	11	2
f_{18}	Best	3910.374389	3775.508662	3702.332347	4297.452196	2941.693932	2930.232884	4485.274456	**2310.35836**	2347.917592	5044.637376	4943.460792
	Ave	18,189.80661	23,000.19929	13,399.46259	16,380.64472	**10,832.44361**	14,229.77058	18,860.46356	14,951.55886	19,532.61654	21,446.80973	14,526.67611
	Std	11,186.86122	11,696.47061	8783.992224	7481.99031	8286.340083	**6350.639993**	11,990.84324	8842.63264	12,942.70152	7915.232206	8776.889925
	Rank	7	11	2	6	1	3	8	5	9	10	4
f_{19}	Best	18,930.30384	**2502.943969**	13,544.70499	12,471.93006	13,574.56875	13,089.4187	86,658.71919	6287.793152	18,584.02272	7938.855378	8540.655378
	Ave	**56,636.82022**	102,311.7499	63,339.37049	142,692.1021	107,155.4409	92,562.68061	113,776.0468	107,437.3629	106,966.6748	70,734.99754	84,670.21175
	Std	26,411.98777	69,304.62777	26,884.79763	59,713.37259	56,602.77841	58,532.39032	**15,337.56871**	49,704.90538	50,204.79423	52,796.42845	30,363.36766
	Rank	1	6	2	11	9	5	10	8	7	3	4
f_{20}	Best	2058.662089	2058.221734	2041.314939	2047.031381	2048.903806	2053.643361	2045.177381	2057.986743	**2038.310878**	2097.823775	2058.510212
	Ave	2158.555059	2096.927074	2099.721737	2096.055833	2119.565216	**2077.046331**	2098.321809	2087.967367	2094.95078	2159.419162	2115.699437

Table 4 (continued)

Function		AOA	CHAOA	CIAOA	GAAOA	ITAOA	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
	Std	52.62893332	42.37915468	54.56309535	20.60342683	49.2818086	19.71655635	45.85978735	**18.87560317**	45.13307444	45.87988351	44.1893844
	Rank	11	5	7	4	9	1	6	2	3	10	8
f_{21}	Best	2263.245638	2225.598134	2223.294976	2238.458681	2237.605058	2228.160709	2250.712382	**2214.76786**	2251.114702	2221.426463	2232.795437
	Ave	2321.095237	2302.169476	2299.908505	2283.855556	2304.102342	**2283.212833**	2312.769028	2288.663238	2313.168427	2315.740746	2312.68697
	Std	**30.63454787**	46.63363325	48.04359904	40.22728473	36.91422635	39.39748189	33.69355061	45.82122169	34.79268759	37.42750373	39.06354113
	Rank	11	5	4	2	6	1	8	3	9	10	7
f_{22}	Best	2629.704017	**2357.11856**	2518.908478	2496.78663	2423.759607	2370.93881	2477.388831	2413.041291	2373.168956	2389.074241	2421.093017
	Ave	3004.521505	2625.643373	2928.426823	3010.847461	2587.040397	2733.57001	2696.810408	2627.074539	2615.274838	2678.103958	**2575.615599**
	Std	357.7396097	282.9031214	386.6984393	288.9953338	**98.08834394**	466.1065651	390.7424999	175.5241396	192.7319487	279.1989939	156.5293722
	Rank	10	4	9	11	2	8	7	5	3	6	1
f_{23}	Best	2680.857484	2675.302026	2648.64851	2641.98098	**2634.698121**	2655.628784	2675.70212	2643.072937	2670.591776	2667.841861	2667.580767
	Ave	2712.21289	2704.072014	2690.451187	2691.710657	**2675.98632**	2715.408149	2713.912755	2692.354681	2701.505212	2726.40717	2704.495603
	Std	**14.45623403**	23.15311429	26.77642668	36.99459081	24.87167513	52.50060462	38.66813944	30.56516866	20.32440111	35.1199894	34.94637671
	Rank	8	6	2	3	1	10	9	4	5	11	7
f_{24}	Best	2804.425953	2594.391407	2657.709204	2786.051034	2697.560605	2646.626528	**2584.302913**	2711.854838	2642.551384	2791.282252	2748.172611
	Ave	2873.683789	2782.917807	2792.030818	2826.955556	2816.975931	2802.339746	**2781.465847**	2803.255024	2791.156706	2826.851846	2811.610311
	Std	46.96896728	83.75028426	69.57018769	**31.30837288**	67.5409849	74.62812458	90.54019244	57.60774344	73.35792882	36.05416106	43.50238371
	Rank	11	2	4	10	8	5	1	6	**3**	**9**	7
f_{25}	Best	3214.548777	2945.599176	2912.094171	3096.918204	**2904.129889**	2932.748281	2924.618642	2941.060756	2918.337653	2931.565272	2951.377558
	Ave	3357.277007	3070.921105	3053.388826	3283.057033	3084.271524	3070.881683	3048.256473	3069.78246	**3030.016801**	3035.789073	3053.97043
	Std	140.4227295	84.58813962	77.40776163	109.4697954	71.90200146	89.96568983	74.84130518	63.46013428	74.35962461	92.41221696	**53.71944992**
	Rank	11	8	4	10	9	7	3	6	1	2	5
f_{26}	Best	3616.272611	3106.847629	3462.050068	3537.71033	3140.311985	3286.813089	3133.93442	3182.814427	3262.736066	3110.528857	**2898.585403**
	Ave	4190.597946	3656.127965	3873.217473	3986.029878	3839.356092	3752.109436	3688.47959	3680.799133	3694.836707	**3496.702792**	3625.751101
	Std	**277.6893286**	408.5781526	298.102417	344.4584222	313.519627	289.3171553	390.9858254	320.5977836	364.620858	317.1339423	360.0212838
	Rank	11	3	9	10	8	7	**5**	4	6	1	2
f_{27}	Best	3136.648849	3139.990364	3144.123681	**3102.621946**	3144.11337	3155.157719	3162.289723	3136.683556	3158.559168	3131.701368	3137.697096
	Ave	3203.355472	3203.679924	3210.248184	**3156.287526**	3204.50178	3189.852922	3212.110484	3187.135346	3206.383335	3211.686391	3219.030247
	Std	44.69914037	49.2937113	50.47579895	32.619977	46.44337826	**22.47777365**	50.1712547	41.29131475	25.64210677	66.70397408	55.64338272
	Rank	4	5	8	1	6	3	10	2	7	9	11
f_{28}	Best	3565.535044	3322.444772	3516.328274	3469.423702	3516.477723	3333.741359	3341.08101	3432.9149	**3308.457648**	3335.910745	3285.364129
	Ave	3751.672772	**3516.063017**	3661.808437	3745.817638	3708.159164	3677.873461	3651.538042	3712.448498	3642.84793	3628.363706	3577.076118
	Std	108.5673543	**107.1505547**	119.7566369	166.1823852	112.514896	176.7737315	160.7243732	153.1186241	153.5855413	182.9694744	145.3034531
	Rank	11	1	6	10	8	7	5	9	4	3	2
Average Rank		11	3	8	10	5	2	7	1	4	9	6

Table 5 Simulation results of AOA with chaotic MOA and MOP to optimize CEC-2017 testing functions

Function		AOA	CHAOA	CIAOA	GAAOA	ITAOA	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
f_1	Best	1,883,345,401	100.2894859	104.5203741	3,378,314,656	**100.0153707**	138.8579088	1256.127094	1900.343244	716.8996174	7204.019595	184.6307383
	Ave	8,025,903,495	1707.053384	306,279.5835	8,201,821,916	**1198.331548**	277,497,749.3	112,747,360.8	318,335,825	141,255,050.1	211,801,483.7	2620.284476
	Std	4,997,942,444	1350.942284	960,112.6826	3,289,350,350	**1221.994398**	412,236,696.3	168,708,049.3	444,797,549.3	419,925,906.9	522,827,816.3	2778.347263
	Rank	11	2	9	10	1	7	4	8	5	6	3
f_2	Best	1,041,282,616	208.6626129	4097.8698	1,511,868,482	**200.0155924**	1939.412242	258.7655475	1276.800725	273.1187948	265.7818087	263.22661
	Ave	37,991,662,000	13,148,854.64	674,223.353	29,795,596,781	**1278.636138**	1661.629989	2,517,930.361	6,262,337.704	3,727,772.656	24,282,763.64	53,755.08514
	Std	98,422,838,088	2423.870072	1,494,216.389	51,486,588,494	**1629.484103**	39,830,578.68	7,868,678.955	13,925,216.28	7,423,198.657	74,703,708.91	148,458.5033
	Rank	11	8	4	10	1	2	5	7	6	9	3
f_3	Best	5365.835811	**300.0000001**	347.16162	7676.584813	300.0000009	1073.399338	534.4699473	1391.876912	369.8727306	622.414191	332.6514695
	Ave	9900.437585	**300.0073568**	1648.488892	9381.337098	301.0391722	1881.458215	1626.573145	2977.439964	2469.062175	2317.627702	1509.884128
	Std	3015.421223	**0.0000022736**	1559.762208	1209.518906	3.9125765223	599.6629075	761.7197386	893.6632263	2244.736726	1209.2416	1147.350982
	Rank	11	1	5	10	2	6	4	9	8	**7**	3
f_4	Best	516.9029036	400.1388697	400.0408537	669.7854592	**400.0249699**	406.8689898	401.4333441	408.0173689	400.0609163	400.0774788	400.5425276
	Ave	1058.139052	**401.172754**	436.0750447	1153.360943	404.2941236	484.6805615	430.9973607	459.8801403	441.0933031	419.2859652	453.831378
	Std	690.6923517	**2.064785256**	34.8341169	461.6330272	4.985942573	52.76002659	35.72124848	65.83576664	42.4807739	18.10247867	61.02584042
	Rank	10	1	5	11	2	9	4	8	6	3	7
f_5	Best	531.8558428	512.9344577	522.8840331	523.8789162	516.9142687	514.9243556	525.182839	**502.9495855**	521.3071409	527.8728353	520.89412
	Ave	550.7354531	535.7893924	547.1660138	542.1718252	539.3856815	540.6268765	538.699648	**507.8504248**	543.4150418	548.0191021	538.1158719
	Std	11.19814192	10.59238575	16.0482169	12.69569039	15.57795925	17.03052031	15.51378627	**6.48643394**	17.90922073	11.03509398	8.38708637
	Rank	11	2	9	7	5	6	4	1	8	10	3
f_6	Best	623.6837262	601.402855	611.7142868	620.3603304	613.9139447	608.8900017	619.1480522	603.7769569	619.4289604	**601.0003036**	600.6095078
	Ave	636.0606981	603.6231563	634.5800889	635.4704204	632.1048083	616.6209451	632.9215385	610.9625352	632.9780629	621.8659986	**601.3053467**
	Std	10.43083867	3.809643724	16.478795313	9.069133286	6.45128039	5.444740243	7.531261042	8.172633799	6.413702941	10.26050572	**0.68456952**
	Rank	11	2	9	10	6	4	7	3	8	5	1
f_7	Best	777.1264207	**713.0764072**	757.3948964	762.4010463	743.031827	749.0541542	770.825868	763.8903723	756.8619991	778.9980786	751.0904395
	Ave	798.3378841	**718.6369423**	788.3118488	787.7314156	782.3922722	777.7826156	789.6745282	779.7087309	787.8024891	792.2821877	774.0510973
	Std	13.56461708	8.602247933	29.78345858	15.46688553	29.64020927	18.69234848	13.58001294	11.84474686	13.57511582	**9.195372053**	19.57076844
	Rank	11	1	8	6	5	3	9	4	7	10	2
f_8	Best	821.9908083	807.6947215	**806.6394986**	819.8991458	813.9294167	816.6647021	806.91603	815.9193298	814.9243708	817.9129074	813.9294167
	Ave	831.3782382	809.3879624	823.9071076	824.8737157	824.3764541	829.5776745	**808.9964256**	826.9462493	826.2105507	828.7586435	823.3350603
	Std	7.868208623	3.103607951	8.331110952	**3.036873888**	6.800891198	6.359433397	3.228222301	7.178736111	8.002367931	6.224381048	5.465750461
	Rank	11	2	4	6	5	10	1	8	7	9	3
f_9	Best	1199.563802	903.6685584	1033.748775	1152.950009	**902.5235233**	1037.13959	1048.075633	1070.71017	1089.187849	919.517863	922.509365
	Ave	1441.492673	**913.0194852**	1285.332345	1316.352094	965.3458551	1267.210616	1334.830165	1307.558527	1339.565968	964.786821	954.6148512
	Std	191.90982	**26.32662942**	120.4734904	107.5712291	88.44613544	142.7856644	176.8640676	210.6216034	152.7838269	83.7796838	76.5473523

Table 5 (continued)

Function		AOA	CHAOA	CIAOA	GAAOA	ITAOA	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
	Rank	11	1	6	8	2	5	9	7	10	4	3
f_{10}	Best	1587.290412	**1237.334764**	1498.624513	1746.9071	1479.010215	1584.221744	1340.58008	1658.268218	1323.303764	1368.000222	1467.355501
	Ave	2144.56232	**1339.117874**	1750.840315	2022.023963	1722.922116	1831.514973	1431.354147	1887.116742	1851.104359	1895.243687	1776.402376
	Std	334.6319266	124.684267	171.829766	247.5883419	**113.0579806**	182.4095244	114.9638171	139.0185293	287.0167371	293.7399448	248.362787
	Rank	11	1	4	10	3	6	2	8	7	9	5
f_{11}	Best	1180.635351	1102.643221	1114.896763	1561.278378	1114.841823	1121.670853	1116.047583	1117.763579	1120.862707	1124.225496	**1102.478072**
	Ave	2545.799871	1104.681325	1124.921158	2658.725684	1124.535449	1125.175145	1124.000178	1129.822938	1144.594724	1137.178117	**1103.753156**
	Std	2209.846615	3.964522124	7.263217877	1371.780623	4.874004228	**1.717992169**	3.660182624	16.94849692	46.39815513	10.5032459	6.886477036
	Rank	10	2	5	11	4	6	3	**7**	9	8	1
f_{14}	Best	1739.352317	**1410.569155**	1750.369179	1490.192272	1458.686487	1482.347286	1498.001452	1481.481612	1984.672984	1472.458928	1554.592681
	Ave	9463.928036	**2466.249563**	13,031.47776	9822.804491	6959.008562	8531.255145	8468.835936	10,989.16687	9328.772643	3641.297492	7821.743272
	Std	8974.703962	**569.6515233**	8283.023993	9822.844365	8137.612477	8750.376686	9174.306126	9578.880801	9541.66434	625.933798	8238.528762
	Rank	8	1	11	9	3	6	5	10	7	2	4
f_{15}	Best	6709.982179	1900.693709	4308.450386	7786.585116	2657.192994	2200.403844	2071.035064	6387.951527	5234.067873	**1658.602632**	3528.566082
	Ave	17,540.37812	**7814.696242**	11,064.77816	21,611.66946	9925.657223	10,695.19764	11,762.53834	13,330.99538	16,975.09232	12,431.51457	14,993.30956
	Std	5981.305859	**2063.476501**	6131.032815	9831.311512	7606.808982	8035.045372	11,517.59292	5330.203945	10,895.41137	7108.354106	10,171.07758
	Rank	10	1	4	11	2	3	5	7	9	6	8
f_{16}	Best	1757.676665	1604.632055	1770.615545	1729.499862	1748.557211	**1601.046921**	1781.79659	1796.314359	1819.497034	1873.241084	1759.930499
	Ave	2081.655133	1618.580467	1949.16243	1983.415247	1951.920221	**1617.963178**	2036.946807	1981.662317	1946.047768	1961.36491	1971.796055
	Std	154.2511998	29.64108545	115.2795	134.1978294	196.0951562	**11.03769827**	159.1002848	143.4925284	103.1036282	87.48014651	134.1858939
	Rank	11	2	4	9	5	1	10	8	3	6	**7**
f_{17}	Best	1769.68982	1706.724947	1756.318575	1739.935724	**1704.704651**	1754.884294	1751.76766	1735.586306	1754.818188	1746.450545	1749.204677
	Ave	1869.464592	**1713.860495**	1831.547087	1835.329369	1828.24676	1835.046467	1828.977804	1813.94494	1819.325615	1832.941266	1798.433995
	Std	96.84942912	**13.21422047**	91.52737268	87.7970661	84.71399996	74.3277332	91.95247019	79.62778402	76.2286721	87.29569322	71.1807199
	Rank	11	1	7	10	5	9	6	3	4	8	2
f_{18}	Best	9944.48805	2407.968267	7343.530329	3362.185597	**2208.824012**	2264.488933	2817.888907	3066.513957	3581.4862	9713.151295	4814.268991
	Ave	19,476.75229	5676.68682	21,334.00795	14,177.07476	19,422.48488	**4845.17455**	17,735.83969	11,820.15549	16,521.79469	22,913.14668	22,618.9528
	Std	6615.74045	**2187.699933**	12,045.14248	8563.182294	12,655.94887	3159.03789	10,478.94237	9839.614698	9436.047017	6355.449324	10,340.75139
	Rank	8	2	9	4	7	1	6	3	5	11	10
f_{19}	Best	2683.538795	**1910.804553**	1912.446218	4759.737637	2011.758164	5083.376161	1972.146059	1940.7614	2113.48672	1918.193574	2139.023393
	Ave	59,040.20939	2016.47499	**2002.98763**	33,474.3322	8724.54284	9184.25455	2010.39379	7495.70981	36,696.20465	2014.53635	4958.98114
	Std	42,893.27085	108.1550965	**47.2528715**	12,579.32683	1432.69647	3765.74422	65.4491856	2696.23453	21,788.96037	68.0919953	1560.49621
	Rank	11	4	1	9	7	8	2	6	10	3	5
f_{20}	Best	2064.152988	2028.078053	2056.002182	2055.0525	2053.758221	2060.019623	2041.890134	2057.232	**2016.920266**	2056.277739	2058.650349
	Ave	2146.886958	**2019.00251**	2099.421691	2111.854388	2096.415434	2110.426702	2085.793309	2088.727476	2096.955545	2091.512854	2104.616671

Table 5 (continued)

Function		AOA	CHAOA	CIAOA	GAAOA	ITAOA	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
	Std	62.6174671	**22.31762162**	36.69063785	50.07935828	30.65164451	32.00724893	22.8881536	30.37515138	70.47292313	24.8796952	28.13807567
	Rank	11	1	7	10	5	9	2	3	6	4	8
f_{21}	Best	2286.394677	**2200.000003**	2200.337804	2249.558547	2238.221235	2205.21895	2201.000005	2210.588584	2205.968539	2242.258314	2257.334055
	Ave	2338.976532	2206.36794	2204.281731	2326.672369	2283.875812	2281.641476	**2203.259126**	2304.480427	2286.76621	2327.15943	2319.02376
	Std	23.41573923	8.73288871	6.97776951	29.21483326	40.81885142	51.7863343	**4.366207**	38.31013777	50.71684234	34.36027016	32.91841684
	Rank	11	3	2	9	6	4	1	7	5	10	8
f_{22}	Best	2731.457207	2282.653217	**2281.285282**	2656.477944	2312.967727	2341.506089	2315.983566	2350.130988	2347.310548	2286.405641	2326.5809
	Ave	3165.923428	2297.11069	2426.102753	2898.764469	2358.936198	2528.242742	2728.927404	2494.12741	2448.316995	**2294.634712**	2567.462039
	Std	451.3678243	19.92370460	166.9927349	288.6917819	47.69540977	399.5616399	439.8982985	249.9515446	133.9217001	**13.63054763**	367.5430286
	Rank	11	2	4	10	3	7	9	6	5	1	8
f_{23}	Best	2670.326897	2613.047066	2646.220256	2654.695049	**2605.907617**	2649.085921	2606.250756	2644.275934	2655.38292	2643.550404	2647.696473
	Ave	2716.756922	2631.644308	2683.898405	2716.865953	2669.982744	2686.75676	**2610.083608**	2702.719779	2701.601511	2684.163325	2688.899193
	Std	31.83165122	13.164526711	18.45224868	47.24707403	29.96051498	30.38134279	**9.64972472**	33.49412198	35.17010575	37.35869253	40.90662461
	Rank	10	2	4	11	3	6	1	9	8	5	7
f_{24}	Best	2730.063347	2500.000095	**2500.00007**	2680.303739	2600.99081	2775.36037	2767.158371	2500.000001	2742.258071	2501.175483	2500.000068
	Ave	2844.513296	**2709.713924**	2760.989556	2832.137423	2795.14062	2823.760417	2835.503381	2756.267078	2813.482014	2795.04398	2773.760062
	Std	79.35925564	108.1765634	143.8845757	73.76968763	101.922824	**35.4802723**	37.19583645	143.0337655	50.38248448	105.5100237	128.4189594
	Rank	11	1	3	9	6	8	10	2	**7**	5	4
f_{25}	Best	2987.277802	2699.891907	2910.517189	3059.978167	2898.005054	2936.962044	2899.617268	2913.388694	**2600.156633**	2899.757801	2600.238974
	Ave	3319.986525	2820.686125	2979.693514	3227.011998	2958.154198	2971.666929	2950.043275	2982.811229	2937.798064	2958.551056	**2814.273241**
	Std	266.8839586	**97.4247573**	41.199622	153.5732956	36.96492651	29.31804708	36.69899372	36.36796747	125.0626308	32.86204131	113.82023
	Rank	11	2	8	10	5	7	4	9	3	6	1
f_{26}	Best	3504.628343	2800.00039	3129.751784	3784.62205	**2600.004371**	2800.000015	2800.000927	2800.000008	2800.001949	3106.301293	2800.000679
	Ave	3933.747553	**2826.664792**	3573.433058	4017.919047	2939.552482	3443.578147	3544.619959	3481.509742	3631.883414	3562.691096	3547.369872
	Std	255.5431907	**34.59475695**	340.6820819	171.0530473	203.6572306	353.5614532	499.9614343	551.4718143	467.0109671	402.9725648	572.2303379
	Rank	10	1	8	11	2	3	5	4	9	7	6
f_{27}	Best	3159.319314	3059.067192	3037.206180	3167.103167	**3049.193708**	3159.613182	3203.976204	3166.179693	3182.130876	3153.892923	3179.47921
	Ave	3250.582884	3084.496288	3175.68007	3225.888254	**3061.89539**	3181.662598	3199.209877	3185.322875	3215.833647	3180.850858	3194.386213
	Std	44.09057438	10.64752694	154.7100429	50.40771583	25.04714441	43.36179968	**7.649034561**	36.53370397	69.93113033	40.1445394	38.0784391
	Rank	11	2	3	10	1	5	8	6	9	4	7
f_{28}	Best	3613.76865	3016.000127	3176.061575	3522.664855	**3100.00005**	3177.000326	3236.153011	3216.294921	3216.09151	3256.046343	3124.550147
	Ave	3823.236456	**3122.866137**	3330.592753	3764.860598	3132.804441	3442.401039	3463.030686	3513.787234	3439.391585	3543.311757	3462.748102
	Std	132.5911434	86.5835458	156.4666987	107.414211	**25.58648555**	191.0426978	131.4115727	162.8770402	157.9426633	158.4940561	139.4993559
	Rank	11	1	3	10	2	5	7	8	4	9	6
Average Rank		11	1	5	10	2	6	4	8	9	7	3

Table 6 Results of Wilcoxon test ($p \geq 0.05$)

Function	CHAOA	CIAOA	GAAOA	ITAOA	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
F1	1.8267e-04	1.8267e-04	0.00640	1.8267e-04	1.8267e-04	1.8267e-04	1.8267e-04	4.3964e-04	1.8267e-04	1.8267e-04
F2	1.8267e-04	1.8267e-04	0.0539	1.8267e-04	1.8267e-04	1.8267e-04	1.8267e-04	3.2984e-04	1.8267e-04	1.8267e-04
F3	1.8267e-04	1.8267e-04	0.0211	1.8267e-04	1.8267e-04	1.8267e-04	1.8267e-04	7.6854e-04	1.7962e-04	1.8267e-04
F4	2.4349e-04	5.7729e-04	0.0537	3.2643e-04	1.8063e-04	4.3528e-04	2.4349e-04	9.9915e-04	3.2643e-04	3.2643e-04
F5	0.0073	0.0121	0.0173	0.0113	0.0312	0.0257	0.0439	0.0376	0.0376	0.0257
F6	0.0058	0.0058	0.1405	0.0010	0.0073	0.0017	0.0058	0.0376	0.0452	0.0014
F7	0.0058	0.0064	0.2413	0.0017	0.0058	0.0013	0.0140	0.0058	1.8267e-04	0.0017
F8	0.0046	2.4673e-04	0.00257	0.00257	0.0028	0.0046	0.0046	0.0017	0.0046	0.0028
F9	7.6854e-04	0.0340	0.0640	0.0091	0.0073	0.0022	0.0017	0.0028	0.0028	0.0211
F10	0.0091	0.0340	0.2413	0.0046	0.0091	0.0257	0.0017	0.0357	0.0452	0.0058
F11	1.8267e-04	1.8267e-04	0.0036	0.0048	1.8267e-04	1.8267e-04	1.8267e-04	5.8284e-04	1.8267e-04	1.8267e-04
F14	2.4349e-04	1.8267e-04	0.0173	1.7962e-04	1.7962e-04	1.8267e-04	0.0091	1.8267e-04	1.7962e-04	1.8267e-04
F15	0.0211	0.0113	0.0640	0.0028	4.3964e-04	0.0113	0.0073	0.0028	0.0021	0.0340
F16	0.0058	0.0140	0.0376	0.0211	0.0376	0.0046	0.0028	0.0028	0.0390	0.0022
F17	0.0012	0.0376	0.1620	0.0273	0.0013	0.0257	0.0340	0.0640	0.0173	0.0067
F18	0.00257	0.0013	0.5708	0.0092	0.0018	0.0211	0.047	0.0021	0.0062	0.0140
F19	4.3964e-04	7.6854e-04	0.0140	3.2984e-04	2.4613e-04	4.3964e-04	0.0017	0.0113	4.3964e-04	0.0036
F20	0.0028	0.0312	0.0757	0.0452	0.0140	0.0113	0.0091	0.0257	0.0113	7.6854e-04
F21	0.0113	0.0340	0.9608	0.0375	1.8063e-04	0.0375	0.0287	0.0024	0.0211	0.0034
F22	1.8267e-04	0.0017	0.0640	7.6854e-04	1.8267e-04	0.0017	5.8284e-04	0.0013	1.8267e-04	1.8267e-04
F23	0.0312	0.5708	0.0173	4.3964e-04	0.0140	0.0067	0.0376	0.0113	0.0073	0.0057
F24	0.0052	0.0173	0.1859	0.0073	0.0012	0.0018	0.0012	0.0018	0.0073	0.0312
F25	1.8267e-04	5.8284e-04	0.0640	3.2984e-04	1.8267e-04	2.4613e-04	1.8267e-04	2.4613e-04	2.4613e-04	2.4613e-04
F26	0.1209	0.0034	0.6774	5.7729e-04	0.0034	0.0173	0.0450	0.012	0.00537	0.0375
F27	0.0073	0.0113	0.5705	0.0139	0.0438	0.0210	0.0091	0.0140	0.0036	0.0012
F28	5.8284e-04	7.6854e-04	0.3847	3.2984e-04	7.6854e-04	3.2984e-04	5.8284e-04	0.0010	1.8267e-04	3.2984e-04

each algorithm optimization test function. When considering embedding the chaotic map into the parameters MOA and MOP at the same time, comparing their average rankings, we can intuitively see that CHAOA has the best optimization performance and ranks the first. ITAOA and TEAOA rank the second and third, respectively.

Based on the above figures and tables, it is not difficult to see that the use of chaotic interference factors to dynamically adjust the parameters on AOA has achieved better results. By further comparing the average values of three improved schemes, for most of the testing functions, the performance of AOA with chaotic interference factor alone to affect MOP is not as effective as AOA with chaotic interference factor alone to affect MOA. When using chaotic interference factors to affect MOA and MOP at the same time, the effect is better than using chaotic interference factors to affect MOA alone. By further comparing the average ranking of the optimization performance of each algorithm, it can be found that CHAOA ranks the first in these three improved schemes. Especially in the last winning scheme, CHAOA's optimization performance leads the other algorithms, ranked first. Therefore, we have reason to believe that CHAOA is the best optimization performance in these algorithm variants.

In the above simulation experiments, this article runs the optimization process of each function independently for ten times, and determines its performance based on the minimum, average, and variance of the three indicators, but it is still impossible to determine the authenticity of each experimental result. In order to show the significance of each optimization result more fully and intuitively, a non-parametric test was performed on the results. The Wilcoxon rank sum test is used in this article. Table 6 describes the p value of the Wilcoxon rank sum test. In Table 6, based on the arithmetic optimization algorithm, the improved AOA based on the chaotic interference factor is tested.It can be found that, except for GAAOA, all p-values are less than 0.05. It shows that the improved AOA is significantly different from the original AOA.

Table 7 Results of CEC - 2017 test suite

Function		CHAOA	HHO	EO	WHO	PSO	WOA	MPA	L-SHADE	LSHADE-EpSin
f_1	Ave	1707.05	3.64E+05	2465.3	100.04	3959.6	2.38E+06	100	100.00	100
	Std	1350.94	1.53E+06	2206.2	1.71E-01	4456.6	4.20E+06	4.79E-06	0.00	0.00
f_3	Ave	300.00	301.00	300.00	300.00	300.00	1339.5	300.00	300.00	300.00
	Std	2.2E-06	1.01	2.4E-08	7.08E-14	1.91E-10	879.78	9.92E-11	0.00	0.00
f_4	Ave	401.17	409.96	404.48	401.68	405.95	442.54	400.03	400.00	400.00
	Std	2.06	16.826	0.7911	0.627	3.283	50.96	0.057	0.00	0.00
f_5	Ave	507.85	546.51	510.73	508.66	513.07	554.43	510.12	503.77	502.34
	Std	6.48	8.409	3.6707	3.58	6.543	0.87	3.955	1.006	0.875
f_6	Ave	603.62	628.56	600.00	600.03	600.24	631.16	600.00	600.00	600.00
	Std	3.81	14.09	1.5E-04	0.109	0.98	12.32	6.16E-04	2.65E-07	2.59E-07
f_7	Ave	718.63	786.43	720.93	719.14	718.98	782.05	723.32	723.91	712.23
	Std	8.60	21.53	5.7425	3.81	5.10	25.67	3.910	1.235	0.605
f_8	Ave	809.38	829.08	809.51	816.91	811.40	839.78	809.42	809.81	802.14
	Std	3.10	9.195	2.9176	3.38	5.475	14.79	3.122	1.278	1.031
f_9	Ave	913.019	1399.5	900.00	900.24	900.00	1357.7	900.00	900.00	900.00
	Std	26.32	247.32	0.0227	0.562	5.9E-14	313.48	1.63E-02	0.00	0.00
f_{10}	Ave	1339.11	1985.9	1418.7	1475.62	1473.35	2062.1	1437.42	1193.56	1070.03
	Std	228.84	275.46	261.63	179.37	214.97	274.61	141.08	84.67	56.56
f_{11}	Ave	1104.68	1153.1	1105.2	1109.00	1110.5	1183.0	1102.93	1105.83	1100.02
	Std	3.97	44.93	5.0218	15.88	6.28	47.03	1.27	1.36	0.12
f_{14}	Ave	2466.26	1553.9	1463.3	1432.7	1482.1	1732.0	1403.09	1410.85	1400.19
	Std	569.65	146.42	32.498	19.12	42.46	639.08	4.06	9.21	0.45
f_{15}	Ave	7814.63	3077.3	1585.6	1533.4	1714.3	5017.2	1500.77	1500.33	1500.33
	Std	2063.47	1286.7	48.012	45.83	282.89	3360.3	0.52	0.36	0.20
f_{16}	Ave	1618.58	1882.3	1649.0	1668.8	1860.0	1842.2	1601.82	1602.50	1600.87
	Std	29.64	120.58	50.915	93.62	127.65	131.59	0.99	2.20	0.36
f_{17}	Ave	1713.86	1779.7	1746.54	1723.3	1761.60	1789.7	1714.55	1716.39	1701.37
	Std	13.21	35.43	39.78	10.56	47.51	43.58	9.44	5.96	3.84
f_{18}	Ave	5176.68	1.46E+04	12,450	1891.8	14,599	17,733	1800.95	1809.87	1803.59
	Std	2187.69	1.21E+04	11,405	73.07	11,852.2	11,207	0.52	9.47	7.60
f_{19}	Ave	2016.47	8954.2	1951.5	2021.7	2602.8	25,420	1900.90	1900.52	1900.26
	Std	108.15	7966.2	47.108	26.77	2185.02	30,016	0.45	0.28	0.03
f_{20}	Ave	2019.00	2126.3	2020.6	2030.7	2085.03	2167.4	2015.52	2020.00	2000.23
	Std	22.31	60.59	22.283	24.71	62.25	63.45	9.67	0.01	0.43
f_{21}	Ave	2206.36	2310.4	2307.5	2249.1	2281.72	2324.6	2203.72	2282.46	2255.42
	Std	8.73	66.37	20.961	54.77	54.02	51.93	20.35	42.64	52.16
f_{22}	Ave	2297.11	2312.2	2297.40	2298.7	2314.8	2353.0	2283.76	2297.27	2300.10
	Std	19.47	15.19	18.402	12.08	66.10	225.26	38.10	16.19	0.17
f_{23}	Ave	2631.64	2659.2	2636.8	2634.6	2620.81	2649.7	2611.63	2638.08	2602.30
	Std	13.16	24.98	5.5298	6.74	9.24	17.80	3.93	1.71	1.42
f_{24}	Ave	2709.71	2791.9	2743.8	2734.2	2692.2	2740.9	2516.88	2728.92	2688.30
	Std	108.17	85.57	6.904	44.49	108.20	102.83	38.39	31.74	91.65
f_{25}	Ave	2820.65	2923.4	2934.3	2952.2	2924.0	2946.2	2897.92	2916.36	2923.77
	Std	97.42	48.50	19.76	24.16	25.02	27.07	0.49	22.94	21.30
f_{26}	Ave	2826.66	3288.3	2967.8	3809.5	2952.1	3517.9	2849.81	2909.17	2900.00
	Std	34.59	45.92	164.98	171.58	249.66	577.8	96.29	34.93	0.00
f_{27}	Ave	3084.49	3151.0	3091.3	3108.5	3116.22	3121.2	3089.37	3071.46	3089.03
	Std	10.64	42.19	2.2414	4.285	24.99	34.78	0.46	0.78	1.05

Table 7 (continued)

Function		CHAOA	HHO	EO	WHO	PSO	WOA	MPA	L-SHADE	LSHADE-EpSin
f_{28}	Ave	3122.86	3389.5	3302.7	3372.2	3315.9	3357.4	3100.10	3266.65	3154.35
	Std	86.58	128.8	133.92	140.39	121.83	148.88	6.34E-05	22.26	110.92

4.1.3 Comparison with other intelligent optimization algorithms

In this section, this article selects some intelligent optimization algorithms that perform well in optimizing the CEC-2017 test functions, and compares them with AOA based on the chaotic interference factor. These competing algorithms include MPA, L-SHADE, LSHADE-EpSin, HHO, EO, WHO, WOA and PSO. Through the previous experimental analysis, it can be seen that the CHAOA algorithm in which the chaotic interference factor is embedded in the parameters MOP and MOA at the same time has the best effect. So this improved AOA is chosen as a representative to compare with other algorithms. Table 7 quotes the data in references [17, 68, 69]. By analyzing the data listed in Table 7, it can be seen that the optimization performance of LSHADE-EpSin and MPA are the best, and CHAOA proposed in this paper ranks third. For most of the test functions, the optimized value of CHAOA is very close to MPA. The fourth and fifth places are LSHADE and EO. Based on the data analysis in the Table 7, the performance of LSHADE-EpSin, MPA and CHAOA is far superior to other methods. This fact shows that CHAOA can be classified as a high-performance optimizer.

4.2 Engineering optimization design problems

4.2.1 Three-bar truss design problem

The design problem of the three-bar truss is to minimize the weight of the truss under certain constraints. Figure 10 shows the model diagram of the three-bar truss design problem. The constraints of the truss mainly refer to buckling constraints, stress and deflection.

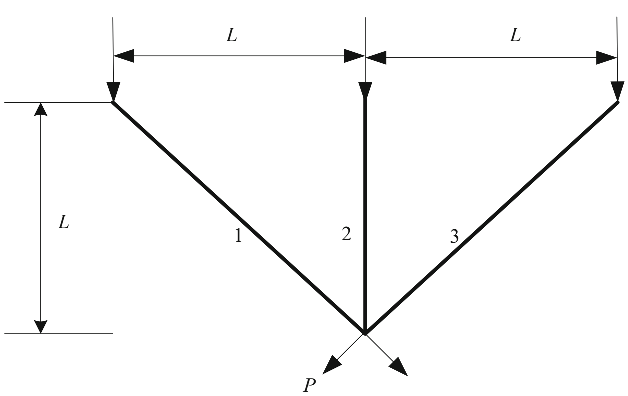


Fig. 10 Model of three-bar truss design problem model

The specific constraints and objective functions of this problem are listed as follows:

$$\text{Objective function::} f(X) = \left(2\sqrt{2}X_1 + X_2\right)*l$$

$$\text{Constraint conditions :}\quad g_1(X) = \frac{\sqrt{2}X_1 + X_2}{\sqrt{2}X_1 + 2X_1X_2}P - \sigma \leq 0$$

$$g_2(X) = \frac{X_2}{\sqrt{2}X_1 + 2X_1X_2}P - \sigma \leq 0$$

$$g_3(X) = \frac{1}{\sqrt{2}X_2 + X_1}P - \sigma \leq 0.$$

where, the variables X_1 and X_2 are the cross-sectional areas of rod 1 and rod 2 respectively, $0 \leq X_1, X_2 \leq 1$, $l = 100$ cm, $P = 2$ KN/cm, $\sigma = 2$ KN/cm.

Figure 11 shows the convergence curves of AOA and chaotic AOA to optimize the three-bar truss design problem. Table 8 records he best values of the original AOA algorithm and the arithmetic optimization algorithm based on the chaotic interference factor to optimize the three-bar truss problem. Each algorithm is run ten times, and then their average and variance are recorded in Table 9, and the best data in the table is bolded. It can be seen from Table 9 that SIAOA optimizing the three-bar truss design problem to obtain the best value, followed by the optimization results of CHAOA and SOAOA. From the average results, it can be seen that

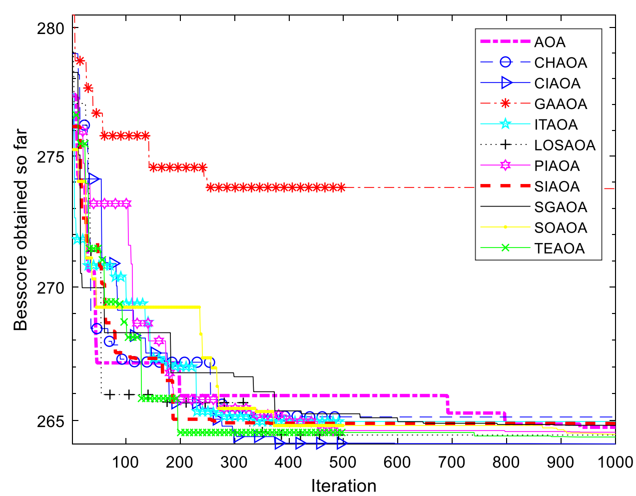


Fig. 11 Convergence curves of three-bar truss problem optimized by chaotic AOA

Table 8 Best solution obtained from chaotic AOA on the three-bar truss design problem

	AOA	CHAOA	CIAOA	GAAOA	ITAOA	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
$f(X)$	263.9003338	263.8956603	263.8958154	263.8974893	263.8961146	263.8977417	263.8959102	263.8956354	263.9003338	263.8956603	263.8960909
X_1	0.773382505	0.755787263	0.784053316	0.88233205	0.760195856	0.765305427	0.76652609	0.761117501	0.761627681	0.764935539	0.769428642
X_2	0.460074074	0.513653585	0.423724365	0.241903364	0.499480447	0.480025982	0.477608324	0.496213535	0.493995985	0.481344422	0.467630763

Table 9 Simulation results obtained from chaotic AOA on the three-bar truss design problem

	AOA	CHAOA	CIAOA	GAAOA	ITAOA	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
Best	263.9003338	263.8956603	263.8958154	263.8974893	263.8961146	263.8977417	263.8959102	**263.8956354**	263.9003338	263.8956603	263.8960909
Ave	264.953013	265.1346813	**263.9362199**	273.7518647	264.9642872	264.4640712	264.5675754	264.8983011	264.8208607	264.4912987	264.390773
Std	0.597660436	0.837560536	0.429521294	9.596328034	0.836743511	0.472045093	0.576899971	0.870117952	0.872141727	0.62239601	**0.309395747**

Table 10 Results obtained from competitor algorithms for the three-bar truss design problem

	CIAOA	HHO	EO	WHO	PSO	WOA	MPA	L-SHADE	LSHADE-EpSin
$f(X)$	263.8958154	263.8959528	263.8958734	263.8958433	263.8958433	263.8959383	263.8959615	263.9045254	263.9146378
X_1	0.784053316	0.703495852	0.804358928	0.793548935	0.758962123	0.774966457	0.769243675	0.805469253	0.786542541
X_2	0.423724365	0.453892456	0.496257952	0.506486528	0.654952352	0.510679455	0.446357129	0.469587226	0.536945852
Best	263.8958154	263.8959528	263.8958734	263.8958433	263.8958433	263.8959383	263.8959615	263.9045254	263.9146378
Ave	263.9362199	263.9419743	263.9463584	263.8958433	263.8959010	264.3105859	263.9754625	263.9946852	263.9657966
Std	0.429521294	0.046762128	2.47E-04	1.27E-13	5.39E-05	0.502907430	3.67E-04	0.003654852	0.03648538

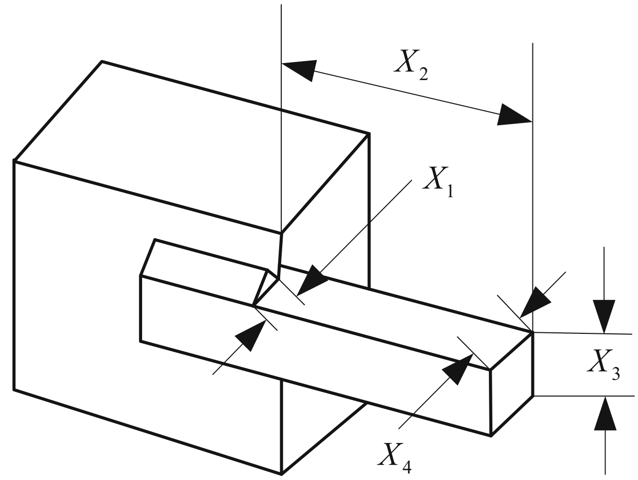


Fig. 12 Model of welded beam design problem

CIAOA optimizing three-bar truss problem is the best, but GAAOA is not good, and its volatility is relatively large.

Table 10 lists the experimental results of other competing algorithms to optimize the design of three-bar truss. Since CIAOA performed well in this engineering problem, the improved algorithm is chosen for comparison. Judging from the data in Table 10, CIAOA has obtained the best parameter values and the best objective function value. From the average point of view, the optimization effect of EO and PSO is relatively good, and CIAOA ranks third.

4.2.2 Welded beam problem

The problem of cantilever beam design is to minimize the manufacturing cost function under certain constraints. Figure 12 shows the model diagram of the cantilever beam design problem. The constraints of the engineering problem include buckling load on the steel bar, shear stress and bending stress in the beam, deflection of the end of the beam, and side constraints. The specific constraints and objective functions of this problem are listed as follows:

$$\text{Objective function}: f(X) = 1.1047X_1^2X_2 + 0.04811X_3X_4(14.0 + X_2)$$

$$\begin{aligned}
\text{Constraint conditions}: \quad & g_1(X) = \tau(X) - \tau_{\max} \leq 0 \\
& g_2(X) = \sigma(X) - \sigma_{\max} \leq 0 \\
& g_3(X) = \delta(X) - \delta_{\max} \leq 0 \\
& g_4(X) = X_1 - X_4 \leq 0 \\
& g_5(X) = P - P_c(X) \leq 0 \\
& g_6(X) = 0.125 - X_1 \leq 0 \\
& g_7(X) = 1.10471X_1^2 + 0.04811X_3X_4(14.0 + X_2) - 5.0 \leq 0
\end{aligned}$$

where, $\tau(X) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{X_2}{2R} + (\tau'')^2}$, $\tau' = \frac{P}{\sqrt{2}X_1}X_2$,

$\tau'' = \frac{MR}{J}$, $M = P\left(L + \frac{X_2}{2}\right)$, $R = \sqrt{\frac{X_2^2}{4} + \left(\frac{X_1 + X_3}{2}\right)^2}$, $J = 2$

Table 11 Best solution obtained from chaotic AOA on the welded beam problem

	AOA	CHAOA	CIAOA	GAAOA	ITAOA	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
$f(X)$	1.863232363	1.758032108	1.764252365	1.805413698	1.796630814	1.785272607	1.779739349	1.810121252	1.796477976	1.797764513	1.816353649
X_1	0.228305289	0.195088781	0.159691422	0.191760026	0.18368033	0.182212441	0.181453546	0.190493019	0.177298827	0.17720774	0.173092383
X_2	3.979138	4.28521233	4.660616354	4.567637292	4.187929547	4.237005341	4.380150727	4.216909707	4.238291221	4.970131431	5.472290829
X_3	9.055928912	9.834650375	9.936206605	9.980579369	9.571878305	9.916587725	9.95062591	9.963645718	9.923263497	9.990385599	9.946803093
X_4	0.265095652	0.209127257	0.201646699	0.201460637	0.228588902	0.205320322	0.201585538	0.203738425	0.201704077	0.201812164	0.201603366

Table 12 Simulation results obtained from chaotic AOA on the welded beam problem

	AOA	CHAOA	CIAOA	GAAOA	ITAOA	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
Best	1.863232363	**1.758032108**	1.764252365	1.805413698	1.796630814	1.785272607	1.779739349	1.810121252	1.796477976	1.797764513	1.816353649
Ave	2.25811996	2.017122494	1.95343655	2.046973763	1.944960839	1.928942491	**1.919208161**	2.110406558	1.94481698	2.031126803	1.937783041
Std	0.308976556	0.214212623	0.119709435	0.287746087	0.135246155	**0.110421535**	0.120066852	0.257071877	0.121047884	0.15894765	0.120892229

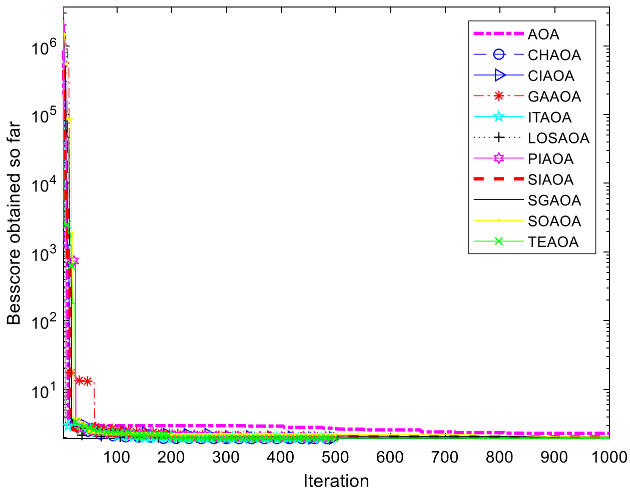


Fig. 13 Convergence curves of welded beam problem optimized by chaotic AOA

$\left\{\sqrt{2}X_1\ X_2\left[\frac{X_2^2}{4}+\left(\frac{X_1+X_3}{2}\right)^2\right]\right\}$, $\sigma(X)=\frac{6PL}{X_4X_3^2}$, $\delta(X)=\frac{6PL^3}{EX_3^2X_4}$,

$$P_c(X)=\frac{4.013E\sqrt{\frac{X_3^2X_4^6}{36}}}{L^2}\left(1-\frac{X_3}{2L}\sqrt{\frac{E}{4G}}\right),$$

$P = 6000$lb., $L = 14$in, $\delta_{max} = 0.25$ in, $E = 30 \times 1^6$psi, $G = 12 \times 10^6$psi, $\tau_{max} = 13600$psi, $\sigma_{max} = 30000$psi. X_1 is the thickness of the weld ($0.1 \le X_1 \le 2$), X_2 is length of attached part of bar ($0.1 \le X_2 \le 10$), X_3 represents the height of the bar ($0.1 \le X_3 \le 10$) and X_4 is the thickness of the bar ($0.1 \le X_4 \le 2$).

Figure 13 shows the convergence curves of AOA and the improved AOA based on the chaotic interference factor to optimize the welded beam design problem. Table 11 records the best values of the original AOA algorithm and the arithmetic optimization algorithm based on the chaotic interference factor to optimize the cantilever beam design problem. Each algorithm is run ten times, and then their average and variance are recorded in Table 12, and the best data in the table is bolded. It can be seen from Table 12 that the improved AOA on the welded beam problem has achieved better results. The best performance of optimizing the welded beam design problem is CHAOA, and the data result obtained by PIAOA ranks second. PIAOA runs ten times to get the best results.

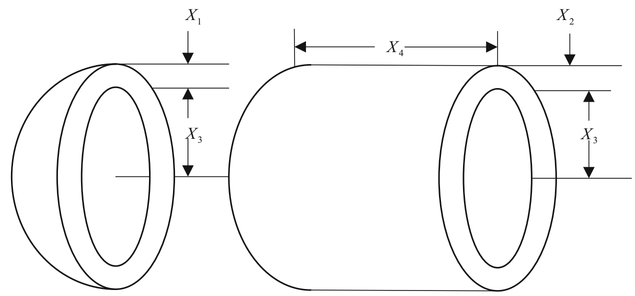


Fig. 14 Model of pressure vessel design problem

Table 13 refers to the experimental results of other competing algorithms to optimize the cantilever problem [16]. Because PIAOA performed well in this engineering problem, the improved algorithm is chosen for comparison. Judging from the data in Table 13, MPA has obtained the best parameter values and the best objective function value. From the average point of view, the optimization effects of MPA and EO are relatively good, and PIAOA ranks third.

4.2.3 Pressure vessel problem

The goal of the problem based on pressure vessel design is to minimize the manufacturing cost function. Figure 14 shows a model diagram of the pressure vessel problem. The container contains four design variables, and the manufacturing cost includes the material of the container, the molding of the container and the welding of the container. The specific constraints and objective functions of this problem are listed as follows:

$$\text{Objective function}: f(X)=0.6224X_1X_3X_4+1.7781X_2X_3^2+3.1661X_1^2X_4+19.84X_1^2X_3$$

Table 13 Results obtained from competitor algorithms for the welded beam problem

	PIAOA	HHO	EO	WHO	PSO	WOA	MPA	L-SHADE	LSHADE-EpSin
$f(X)$	1.779739349	1.8561	1.7449	1.7839	3.2731	2.3584	1.724853	2.0701	2.0157
X_1	0.181453546	0.2134	0.2057	0.2377	0.2843	0.329	0.205728	0.2389	0.2884
X_2	4.380150727	3.5601	3.4705	4.3647	7.5333	2.5471	3.470509	3.4067	3.1057
X_3	9.95062591	8.4629	9.0366	8.9634	7.6664	6.8078	9.036624	9.6383	9.3491
X_4	0.201585538	0.2346	0.2057	0.2846	0.3274	0.3789	0.205730	0.2901	0.2999
Best	1.779739349	1.8561	1.7449	1.7839	3.27314	2.3584	1.724853	2.0701	2.0157
Ave	1.919208161	1.9302	1.7555	1.9964	3.57351	2.5685	1.724861	2.1361	2.4165
Std	0.120066852	0.0647	1.86E-03	0.0965	4.00E-01	2.14E-01	6.41E-06	5.88E+01	4.49E+01

Table 14 Best solution obtained from chaotic AOA on the pressure vessel problem

	AOA	CHAOA	CIAOA	GAAOA	ITAOA
$f(X)$	7123.343021	5739.220871	5735.421334	5734.920568	5735.274335
X_1	0.772771587	0.846971149	0.849903235	0.843369831	0.842133933
X_2	0.71390865	0.416651072	0.416642347	0.414212897	0.412909412
X_3	46.39395548	45.55368133	45.70193845	45.36059782	45.27949309
X_4	195.7466882	154.6557073	154.7573705	150.9698682	159.4133316

	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
$f(X)$	5735.447058	5736.225866	5736.297101	5738.351437	5842.077062	5735.150342
X_1	0.867016596	0.917694149	0.841582141	0.941014553	0.806413327	0.858588706
X_2	0.424075203	0.445791312	0.413888314	0.456526054	0.496136098	0.419844946
X_3	46.51822984	49.12653797	45.28904823	50.39012775	42.9324112	46.13545358
X_4	145.6331478	133.083654	155.7817616	125.3755305	187.4321534	147.7183473

Table 15 Simulation results obtained from chaotic AOA on the pressure vessel problem

	AOA	CHAOA	CIAOA	GAAOA	ITAOA	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
Best	7123.343021	5739.220871	5735.421334	**5734.920568**	5735.274335	5735.447058	5736.225866	5736.297101	5738.351437	5842.077062	5735.150342
Ave	15,534.9322	6239.881424	6018.20791	5915.114851	6117.22766	6601.613222	6186.588581	**5822.608283**	6259.023298	6519.863043	6218.725465
Std	6155.972656	482.5626135	382.5802902	335.1907151	216.2207297	694.0836023	555.1755607	**111.0541144**	1085.468516	742.2062546	817.9789783

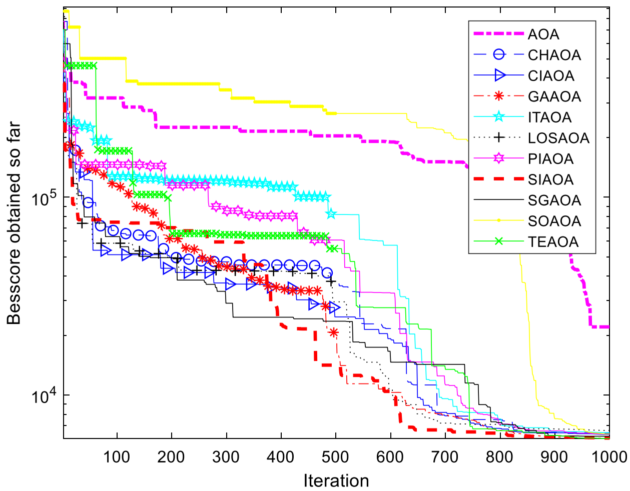


Fig. 15 Convergence curves of pressure vessel problem optimized by chaotic AOA

$$\text{Constraint condition}: \quad g_1(X) = 0.0193X_3 - X_1 \le 0$$
$$g_2(X) = 0.00954X_3 - X_2 \le 0$$
$$g_3(X) = 1296000 - \pi X_3^2 X_4 - {}^4/_3 \pi X_3^3 \le 0$$
$$g_4(X) = X_4 - 240 \le 0$$

where, X_1 and X_2 are head (Th) and cylinder wall thickness (Ts), $0.0625 \le X_1, X_2 \le 6.1875$; X_3 is the radius of the cylinder and head (R), X_4 is the cylinder length (L), $10 \le X_3, X_4 \le 200$. Among the four variables, X_1 and X_2 are uniformly discrete variables with an interval of 0.0625, X_3 and X_4 are continuous variables.

Figure 15 shows the convergence curves of AOA and the improved AOA based on the chaotic interference factor to optimize the pressure vessel problem. Table 14 records the best values of the original AOA algorithm and the arithmetic optimization algorithm based on the chaotic interference factor to optimize the pressure vessel design problem. Each algorithm is run ten times, and then their average and variance are recorded in Table 15, and the best data in the table is bolded. It can be seen from Table 15 that the optimization of improved AOA on the pressure vessel design problem has achieved better results. GAAOA optimizes the pressure vessel design problem to obtain the smallest minimum value. But from the average and variance point of view, SIAOA ranks first in the optimization process.

Table 16 refers to the experimental results of other competing algorithms to optimize the pressure vessel problem [16]. Since SIOAA performed well in this engineering problem, the improved algorithm is chosen for comparison. Judging from the data in Table 16, SIOAA has obtained the best parameter values and the best objective function value. From the average point of view, SIAOA is equally excellent, and its standard deviation is also the smallest, ranking first in the listed algorithms, and MPA and EO are ranked second and third respectively.

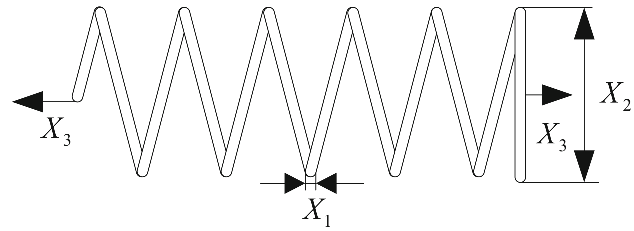


Fig. 16 Model of tension spring design problem

4.2.4 Tension spring problem

The goal of the tension spring design problem is to minimize the weight of the compression spring under certain constraints. Figure 16 shows the model diagram of the tension spring design problem. The design problem has three design variables. Constraints include vibration frequency, minimum deflection and shear stress. The specific constraints and objective functions of this problem are listed as follows:

$$\text{Objective function}: f(X) = (X_3 + 2)X_2X_1^2$$

Table 16 Results obtained from competitor algorithms for the pressure vessel problem

	SIAOA	HHO	EO	WHO	PSO	WOA	MPA	L-SHADE	LSHADE-EpSin
$f(X)$	5736.297101	6.39E+03	5.91E+03	5.96E+03	1.70E+04	7.11E+03	5885.3353	7.67E+03	6.85E+03
X_1	0.841582141	0.9833	0.7929	0.7962	1.6186	0.9730	0.77816876	0.8525	0.9330
X_2	0.413888314	0.4758	0.3914	0.4695	0.6296	0.6512	0.38464966	0.5775	0.6982
X_3	45.28904823	49.9297	41.1773	43.6952	59.2820	50.6804	40.3196208	56.3105	59.9952
X_4	155.7817616	98.9036	188.3950	159.5595	146.9644	93.0377	199.999993	65.7572	47.5678
Best	5736.297101	6.39E+03	5.91E+03	5.96E+03	1.701E+04	7.11E+03	6059.7144	7.67E+03	6.85E+03
Ave	5822.608283	6.61E+03	6.53E+03	6.98E+03	2.431E+04	1.05E+04	6102.8271	1.47E+04	1.87E+05
Std	111.0541144	2.54E+02	3.98E+02	360.69	9.91E+03	3.23E+03	106.61	7.74E+03	5.00E+05

Table 17 Best solution obtained from chaotic AOA on the tension spring problem

	AOA	CHAOA	CIAOA	GAAOA	ITAOA	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
$f(X)$	0.012481372	0.013598497	0.013601781	0.012830692	0.012129843	0.013601594	0.013604257	0.012247796	0.013599376	0.013609742	0.013600732
X_1	0.053456002	0.053845695	0.05384707	0.053752318	0.053643955	0.053843523	0.053844542	0.053568226	0.053846168	0.053848018	0.053844009
X_2	0.255047705	0.25	0.25	0.25	0.251004771	0.25	0.25	0.254893206	0.25	0.25	0.25
X_3	13.67026568	14.99243183	14.99250364	14.00363062	13.38998836	14.98778923	14.99279439	11.45465809	14.99676665	14.99878445	14.99290598

Table 18 Simulation results obtained from chaotic AOA on the tension spring problem

	AOA	CHAOA	CIAOA	GAAOA	ITAOA	LOAOA	PIAOA	SIAOA	SGAOA	SOAOA	TEAOA
Best	0.012481372	0.013598497	0.013601781	0.012830692	0.012129843	0.013601594	0.013604257	**0.012247796**	0.013599376	0.013609742	0.013600732
Ave	0.014488993	0.013606247	0.013610493	0.013330317	0.013608743	0.013605527	0.013609556	**0.012485346**	0.013606517	0.013612311	0.013608549
Std	0.00108086	2.45E-06	1.12E-06	0.000170224	1.54E-06	2.03E-06	2.34E-06	0.000662149	5.00E-06	**1.31E-07**	4.24E-06

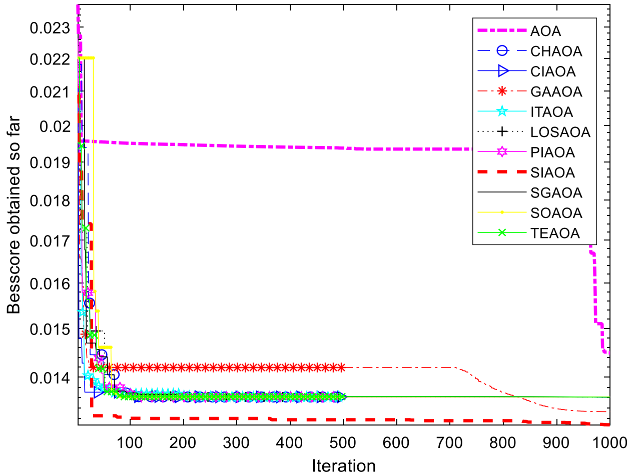


Fig. 17 Convergence curves of tension spring problem optimized by chaotic AOA

$$\text{Constraint condition}: \quad g_1(X) = 1 - \frac{X_2^3 X_3}{71785 X_1^4} \leq 0$$

$$g_2(X) = \frac{4X_2^2 - X_1 X_2}{12566\left(X_2 X_1^3 - X_1^4\right)} + \frac{1}{5108 X_1^2} \leq 0$$

$$g_3(X) = 1 - \frac{140.45 X_1}{X_2^2 X_3} \leq 0$$

$$g_4(X) = \frac{X_1 + X_2}{1.5} - 1 \leq 0$$

where, X_1 is wire diameter ($0.05 \leq X_1 \leq 2.00$), X_2 is the average coil diameter ($0.25 \leq X_2 \leq 1.30$), and X_3 represents the number of effective coils ($2.00 \leq X_3 \leq 15.0$).

Figure 17 shows the convergence curves of AOA and AOA based on the chaotic interference factor to optimize the tension spring design problem. Table 17 records the best values of the original AOA algorithm and the arithmetic optimization algorithm based on the chaotic interference factor to optimize the tension spring design problem. Each algorithm is run ten times, and then their average and variance are recorded in Table 18, and the best data in the table is bolded. It can be seen from Table 18 that the improved algorithms in the optimization of the tension spring design problem have achieved better results. The optimal value obtained by SIAOA for optimizing the tension spring design problem is the smallest when the constraints are met, followed by the optimal value by CHAOA. At the same time, the average value of SIAOA to optimize the spring design problem is the smallest and SOAOA is more stable.

Table 19 refers to the experimental results of other competing algorithms to optimize the tension spring problem [16]. Since SIOAA performed well in this engineering problem, the improved algorithm was chosen for comparison. Judging from the data in Table 19, SIOAA has obtained the best parameter values and the best objective function value. From the average point of view, SIAOA is equally excellent, ranking first in the listed algorithms, and MPA and EO are ranked second and third respectively.

Through the above comparison with competing algorithms, it can be found that the improved algorithm proposed in this paper performs well in solving actual engineering optimization problems. Among the four engineering problems, three can obtain the best parameter values, so as to obtain the best objective function value. They are the three-bar truss problem, the pressure vessel problem, and the spring tension problem. In terms of the average optimization effect, for the four engineering problems, the proposed improved algorithm in this paper can be ranked in top three. Especially when solving the pressure vessel problem and the spring tension problem, it ranked first. The optimization effects of MPA and EO are also relatively good.

Table 19 Results obtained from competitor algorithms for the tension spring problem

	SIAOA	HHO	EO	WHO	PSO	WOA	MPA	L-SHADE	LSHADE-EpSin
$f(X)$	0.012247796	0.013016	0.012682	0.012665	1.733190	0.012683	0.012665	0.018662	0.017227
X_1	0.053568226	0.0562	0.0512	0.0576	0.0582	0.0507	0.051724477	0.0555	0.0592
X_2	0.254893206	0.4754	0.3445	0.3986	0.7952	0.3339	0.35757003	0.4706	0.4983
X_3	11.45465809	6.6670	12.0455	8.3695	5.2794	12.7645	11.2391955	7.4552	8.8980
Best	0.012247796	0.013026	0.012682	0.012665	1.733190	0.012683	0.012665	0.018662	0.017237
Ave	0.012485346	0.014160	0.013536	0.015636	1.143211	0.014709	0.012665	0.019992	0.016214
Std	0.000662149	1.64E-03	2.20E-04	3.68E-03	8.48E+05	2.30E-03	5.55E-08	1.48E+02	1.01E+01

5 Conclusions

The improved AOA based on chaotic interference factors adopted in this paper improves the ability of the AOA to balance exploration and exploitation, avoids falling into local optimization, and improves the accuracy of convergence. This article has designed three schemes to carry out simulation experiments to test their performance. The optimization experiment results of unimodal functions, multimodal functions, mixed functions and complex functions show that the optimization performance of AOA based on chaotic interference factors are good. The optimization results show that the optimized average value of the arithmetic optimization algorithm based on the chaotic interference factor is lower than the initial algorithm in most of the test functions. By further comparing the optimization rankings of different chaotic variants, the CHAOA algorithm in which the chaotic interference factor is embedded in MOP and MOA at the same time has achieved the best effect among many chaotic variants. When compared with other optimization algorithms, the chaotic arithmetic optimization algorithm also has the upper hand. On the other hand, the convergence performance of the chaotic AOA to optimize the engineering design problem is also excellent. Among them, the optimized three-bar truss problem and the welded beam design problem are the most stable and the best is CIAOA and PIAOA respectively. For pressure vessel and spring design problems, the convergence speed and accuracy of the SIOA algorithm make it stand out. By comparing with other competing algorithms, the algorithm proposed in this article performs well in solving practical engineering problems.

It the future work, we plan to apply chaos theory to other optimization algorithms, not only considering the existing 2-dimensional chaos, but also considering the application of chaotic systems for optimization in order to extensively compare the performance of chaos in different heuristic algorithms. In addition, we will extend the application of the improved algorithm. First of all, the improved algorithm can be applied to the Frequent Itemsets Mining (FIM) [70] technology and the field of mining top-k most frequent association rules [71]. Secondly, CAOA can be applied to the recently more popular deep learning to optimize neural network parameters and improve the efficiency of network training. Finally, extending the algorithm to multi-objective optimization to solve combinatorial optimization problems can also be seen as a future contribution.

Acknowledgements This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. LJKZ0293), and the Project by Liaoning Provincial Natural Science Foundation of China (Grant No. 20180550700).

Author contributions Xu-Dong Li participated in the data collection, analysis, algorithm simulation, and draft writing. Jie-Sheng Wang participated in the concept, design, interpretation and commented on the manuscript. Wen-Kuo Hao, Min Zhang and Min Wang participated in the critical revision of this paper.

Declarations

Conflict of interest The authors declare that there is no conflict of interests regarding the publication of this article.

References

1. Abualigaha L, Diabat A, Mirjalili S et al (2021) The arithmetic optimization algorithm. Comp Methods Appl Mech Eng 376: 113609
2. Dutta T, Bhattacharyya S, Dey S, Platos J (2020) Border collie optimization. IEEE Access 8:109177–109197
3. S Arora, P Anand (2018) Chaotic grasshopper optimization algorithm for global optimization. Neural Comp Appl
4. Yang X-S, Gandomi AH, Talatahari S, Alavi AH (eds) (2012) Metaheuristics in water, geotechnical and transport engineering.Elsevier, Newnes
5. Abualigah L, Diabat A (2020) A comprehensive survey of the grasshopper optimization algorithm: results, variants, and applications. Neural Comput. Appl.:1–24
6. Shahrzad Saremi,Seyedali Mirjalili,Andrew Lewis (2014) Biogeography-based optimisation with chaos. Neural Comput & Applic
7. Kallioras NA, Lagaros ND, Avtzis DN (2018) Pity beetle algorithm–a new metaheuristic inspired by the behavior of bark beetles. AdvEng Softw 121:147–166
8. Talatahari S, Azizi M (2021) Chaos game optimization: a novel metaheuristic algorithm. Artif Intell Rev 54:917–1004
9. Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S (2020) Equilibrium optimizer: a novel optimization algorithm. Knowl-Based Syst 191:105190
10. Sadollah A, Sayyaadi H, Lee HM, Kim JH et al (2018) Mine blast harmony search: a new hybrid optimization method for improving exploration and exploitation capabilities. Appl Soft Comput 68: 548–564
11. Gholizadeh S, Danesh M, Gheyratmand C (2020) A new newton metaheuristic algorithm for discrete performance-based design optimization of steel moment frames. Comput Struct 234:106250
12. Abualigah L (2020) Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. Neural Comput Appl:1–24
13. Jordehi AR (2014) Particle swarm optimisation for dynamic optimisation problems:a review. Neural Comput Appl:1–10
14. El-Shorbagy MA, El-Refaey AM (2020) Hybridization of grasshopper optimization algorithm with genetic algorithm for solving system of non-linear equations. IEEE Access 8:220944–220961
15. et al (2021) A novel algorithm for global optimization: rat swarm optimizer. J Ambient Intell Human Comput 12:8457–8482Dhiman, G., Garg, M., Nagar, A.et al A novel algorithm for global optimization: rat swarm optimizer. J Ambient Intell Human Comput 12, 8457–8482 (2021)
16. Hashim FA, Hussain K, Houssein EH et al (2021) Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Appl Intell 51:1531–1551
17. Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S (2020) Equilibrium optimizer: a novel optimization algorithm. Knowledge-Based Syst 191
18. Abualigah L, Shehab M, Alshinwan M, Mirjalili S, Abd Elaziz M (2020) Ant lion optimizer: a comprehensive survey of its variants and applications. Arch Comput Methods Eng 28:1397–1416
19. Assiri AS, Hussien AG, Amin M (2020) Ant lion optimization: variants, hybrids, and applications. IEEE Access 8:77746–77764

20. Wang Y, Gao S, Yu Y, Wang Z, Cheng J, Yuki T (2020) A gravitational search algorithm with chaotic neural oscillators. IEEE Access 8:25938–25948
21. Mahamed GH, Omran MM (2008) Global-best harmony search. Appl Math Comput 198(2)
22. Beyer H, Sendhoff B (2017) Simplify your covariance matrix adaptation evolution strategy. IEEE Trans Evol Comp 21(5):746–759
23. Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. Soft Comput 23:715–734
24. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer [J]. Adv Eng Softw 69:46–61
25. Abualigah L, Diabat A (2020) A comprehensive survey of the grasshopper optimization algorithm: results, variants, and applications. Neural Comput & Applic 32:15533–15556
26. Eskandari S, Javidi MM (2020) A novel hybrid bat algorithm with a fast clustering-based hybridization. Evol Intel 13:427–442
27. Chen H, Li W, Yang X (2020) A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems. Expert Syst Appl 158:113612
28. Saha S, Mukherjee V (2018) A novel chaos-integrated symbiotic organisms search algorithm for global optimization. Soft Comput 22:3797–3816
29. Gandomi A, Yang X-S, Talatahari S, Alavi A (2013) Firefly algorithm with chaos. Commun Nonlinear Sci Numer Simul 18(1):89–98
30. Kaur A, Pal SK, Singh AP (2018) New chaotic flower pollination algorithm for unconstrained non-linear optimization functions[J]. Int J Syst Assur Eng Manag 9(4):853–865
31. Yu H (2020) Nannan Zhao, Pengjun Wang, Huiling Chen, Chengye Li, chaos-enhanced synchronized bat optimizer, applied mathematical modelling, volume 77. Part 2:1201–1215
32. D Prayogo (2021) Chaotic coyote algorithm applied to truss optimization problems, Comp Struct,242, Juliano Pierezan, Leandro dos Santos Coelho, Viviana Cocco Mariani, Emerson Hochsteiner de Vasconcelos Segundo
33. Sanaj MS, Joe Prathap PM (2020) Nature inspired chaotic squirrel search algorithm (CSSA) for multi objective task scheduling in an IAAS cloud computing atmosphere. Eng Sci Technol Int J 23(4)
34. Heidari AA, Abbaspour RA, Jordehi AR (2017) An efficient chaotic water cycle algorithm for optimization tasks. Neural Comput Appl 28(1):57–85
35. Gupta S, Deep K (2018) An opposition-based chaotic Grey wolf optimizer for global optimisation tasks[J]. J Exp Theor Artif Intell 31:1–29
36. Saha S, Mukherjee V (2017) A novel quasi-oppositional chaotic antlion optimizer for global optimization[J]. Appl Intell 48(9): 2628–2660
37. Gandomi AH, Yun GJ, Yang X-S, Talatahari S (2013) Chaos enhanced accelerated particle swarm optimization. Commun Nonlinear Sci Numer Simul 18(2):327–340
38. Arora S, Singh S (2017) An improved butterfly optimization algorithm with chaos. J Intell Fuzzy Syst 32(1):1079–1088
39. Han X, Chang X (2013) An intelligent noise reduction method for chaotic signals based on genetic algorithms and lifting wavelet transforms. Inf Sci 218:103–118
40. Coelho LDS (2009) Reliability–redundancy optimization by means of a chaotic differential evolution approach. Chaos Solitons Fractals 41:594–602
41. Lu Y, Zhou J, Qin H, Wang Y, Zhang Y (2011) Chaotic differential evolution methods for dynamic economic dispatch with valve-point effects. Eng Appl Artif Intell 24:378–387
42. Pan Q-K, Wang L, Gao L (2011) A chaotic harmony search algorithm for the flow shop scheduling problem with limited buffers. Appl Soft Comput 11:5270–5280
43. Ahmed A. Ewees, Mohamed Abd Elaziz, Zakaria Alameer, Haiwang Ye, Zhang Jianhua, Improving multilayer perceptron neural network using chaotic grasshopper optimization algorithm to forecast iron ore price volatility, Resources Policy, 65, 2020, 101555
44. Talatahari S, Farahmand Azar B, Sheikholeslami R, Gandomi A (2012) Imperialist competitive algorithm combined with chaos for global optimization. CommunNonlinear Sci Numer Simul 17: 1312–1319
45. Talatahari S, Kaveh A, Sheikholeslami R (2011) An efficient charged system search using chaos for global optimization problems. Int J Optim Civil Eng 2:305–325
46. Alatas B (2011) Uniform big bang–chaotic big crunch optimization. Commun Nonlinear Sci Numer Simul 16:3696–3703
47. Wu B., Fan S. (2011) Improved artificial bee Colony algorithm with chaos. In: Yu Y., Yu Z., Zhao J. (eds) Computer Science for Environmental Engineering and EcoInformatics. CSEEE 2011. Communications in Computer and Information Science, vol 158. Springer, Berlin, Heidelberg
48. Jordehi AR (2015) A chaotic artificial immune system optimisation algorithm for solving global continuous optimisation problems. Neural Comput Appl 26(4):827–833
49. Chuang L-Y, Tsai S-W, Yang C-H (2011) Chaotic catfish particle swarm optimization for solving global numerical optimization problems. Appl Math Comput 217(16):6900–6916
50. Saremi S, Mirjalili S, Lewis A (2014) Biogeography-based optimisation with chaos. Neural Comput Appl 25(5):1077–1097
51. Han X, Chang X (2012) A chaotic digital secure communication based on a modified gravitational search algorithm filter. Inf Sci 208:14–27
52. Niknam T, Narimani MR, Jabbari M et al (2011) A modified shuffle frog leaping algorithm for multi-objective optimal power flow. Energy 36:6420–6432
53. Prasad D, Mukherjee A, Shankar G, Mukherjee V (2017) Application of chaotic whale optimisation algorithm for transient stability constrained optimal power flow. IET Sci, Meas Technol
54. Gandomi AH, Yang X-S (2014) Chaotic bat algorithm. J. Comput. Sci. 5(2):224–232
55. Mukherjee A, Mukherjee V (2015) Solution of optimal reactive power dispatch by chaotic krill herd algorithm. IET Gener. Transm. Distrib 9(15):2351–2362
56. Zhu S, Zhu C, Cui H, Wang W (2019) A class of quadratic polynomial chaotic maps and its application in cryptography. IEEE Access 7:34141–34152
57. Anupadma S, Dharshini BS, Roshini S, Singh JK (2020) Random selective block encryption technique for image cryptography using chaotic cryptography. 2020 Int Conf Emerging Trends Inform Technol Eng (ic-ETITE):1–5
58. Banu SA, Amirtharajan R (2020) A robust medical image encryption in dual domain: chaos-DNA-IWT combined approach. Med Biol Eng Comput 58:1445–1458
59. Yu WB (2017) Application of Chaos in Image Processing and Recognition. 2017 Int Conf Comp Syst Elec Control (ICCSEC): 1108–1113
60. Chithra A, Raja Mohamed I (2017) Synchronization and chaotic communication in nonlinear circuits with nonlinear coupling. J Comput Electron 16:833–844
61. Naanaa A (2015) Fast chaotic optimization algorithm based on spatiotemporal maps for global optimization. Appl Math Comput 269:402–411
62. Lu H, Wang X, Fei Z, Qiu M (2014) The effects of using chaotic map on improving the performance of multiobjective evolutionary algorithms. Math Prob Eng 2014:16–16
63. Khennaoui AA, Ouannas A, Boulaaras S, Pham VT, Taher Azar A (2020) A fractional map with hidden attractors: chaos and control. Eur Phys J Spec Top 229:1083–1093
64. Yousri D, Allam D, Babu TS et al (2020) Fractional chaos maps with flower pollination algorithm for chaotic systems' parameters identification. Neural Comput & Applic 32:16291–16327

65. Zhuoran Z, Changqiang H, Hanqiao H, Shangqin T, Kangsheng D (April 2018) An optimization method: hummingbirds optimization algorithm. J Syst Eng Electron 29(2):386–404
66. Houssein EH, Helmy BE-D, Elngar AA, Abdelminaam DS, Shaban H (2021) An improved tunicate swarm algorithm for global optimization and image segmentation. IEEE Access 9:56066–56092
67. Kommadath R, Kotecha P (2017) Teaching learning based optimization with focused learning and its performance on CEC2017 functions[C]// evolutionary computation. IEEE:2397–2403
68. Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH (2020) Marine predators algorithm: a nature-inspired metaheuristic[J]. Expert Syst Appl 152:1–50
69. Naruei I, Keynia F (2021) Wild horse optimizer: a new meta-heuristic algorithm for solving engineering optimization problems. Eng Comput
70. Djenouri Y, Comuzzi M (2017) Combining Apriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem. Inf Sci 420:1–15
71. Liu X, Niu X, Fournier-Viger P (2021) Fast top-K association rule mining using rule generation property pruning. Appl Intell 51: 2077–2093

X. D. Li received the B.Eng. degree in electrical engineering and automation from Shanxi university, Taiyuan, Shanxi China, in 2020. He is currently pursuing the M.Eng. degree with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China. His research interests include pattern recognition, soft sensor technology, function optimization and multi-objective optimization.

J. S. Wang received his B. Sc. And M. Sc. degrees in Control Science and Engineering from University of Science and Technology Liaoning, China in 1999 and 2002, respectively, and his Ph. D. degree in Control Science and Engineering from Dalian University of Technology, China in 2006. He is currently a professor and Doctor's Supervisor in School of Electronic and Information Engineering, University of Science and Technology Liaoning. His main research interest is modeling of complex industry process, intelligent control and Computer integrated manufacturing.

W. K. Hao received the B.Eng. degree in electrical engineering and automation from Liren college of yanshan university, Qinhuangdao, Hebei China, in 2018. He is currently pursuing the M. Eng. degree with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China. His research interests include economic dispatch problem.

M. Zhang received the B. Eng. degree in Measurement and control technology and instrument from Qufu Normal University in Qufu, Shandong China, in 2020. She is currently pursuing the M. Eng. degree with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China. Her research interests include combinational optimization.

M. Wang received the B.Eng degree in automation engineering from Xuhai College of China University of mining and technology in Xuzhou, Jiangsu Province, China in 2020. She is currently pursuing the M. Eng.degree with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China. She is interested in multi-objective optimization algorithms.