

Hybrid Whale Optimization Algorithm Based on Three Spiral Searching Strategies and Sine Cosine Operator with Convergence Factor

Ji Zhang

University of Science and Technology Liaoning

Kai Yang

University of Science and Technology Liaoning

jiesheng wang (✉ wang_jiesheng@126.com)

University of Science and Technology Liaoning <https://orcid.org/0000-0002-8853-1927>

Research Article

Keywords: Whale optimization algorithm, Spiral search, Sine cosine operator, Convergence factor, Optimization

Posted Date: May 18th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-215608/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Hybrid Whale Optimization Algorithm Based on Three Spiral Searching Strategies and Sine Cosine Operator with Convergence Factor

Ji Zhang ¹, Kai Yang ^{1*}, Jie-Sheng Wang ²

1. School of Computer and Software Engineering, University of Science and Technology Liaoning, Anshan 114000, Liaoning

2. School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114000, Liaoning

*Correspondence: yangkai@ustl.edu.cn (K. Yang); Tel.: +86-0412-2538355.

Abstract: Whale Optimization Algorithm (WOA) is a swarm intelligence algorithm inspired by whale hunting behavior. Aiming at the defect that the spiral update mechanism in WOA may exceed the search range, three different spiral searching strategies are first proposed. The agents search with a more reasonable and broader route distribution so as to improve population diversity and traversal. Secondly, an improved sine cosine operator based on the convergence factor was proposed to improve the search efficiency of WOA, where sine search is used for global exploration and cosine search is used for local exploitation. The proposed convergence factor enables search agents to adaptively balance the exploration and exploitation phases with iterations. In the simulation experiment, the effectiveness of three spiral search strategies and sine cosine operator is verified. Then, the whale optimization algorithm (WOA), salp swarm algorithm (SSA), firefly algorithm (FA), moth-flame optimization (MFO) algorithm, fireworks algorithm (FWA), sine cosine algorithm (SCA) and improved WOA are selected for comparison experiments. Finally, the improved WOA is applied to two engineering problems (three-bar truss design problem and the welded beam optimization problem). The experimental results show that compared with other optimization algorithms, the improved WOA has the advantages of high search accuracy, fast convergence speed, and avoiding falling into local optimal values.

Keywords: Whale optimization algorithm; Spiral search; Sine cosine operator; Convergence factor; Optimization

1. Introduction

Exact approaches and meta-heuristics are two types of algorithms for solving optimization problems. Exact approaches are traditional algorithms in the field of operations research. They create mathematical models based on optimization problems. When the scale of the optimization problem is small, the exact approaches can find the optimal solution in an acceptable time. However, exact approaches have a complex structure and fall into the local optimum when solving large-scale optimization problems. Meta-heuristic algorithms are inspired by simple principles and have no derivation mechanism. At the same time, meta-heuristic algorithms treat the optimization problem as a black box, which can solve optimization problems in different fields, which has attracted the attention of researchers. Meta-heuristic algorithms simulate biological or natural worlds to build mathematical models. With the deepening of research, meta-heuristic algorithms have developed the main branches: algorithms based on evolution, swarm intelligence based algorithms and physical phenomena based algorithms. Evolution-based

algorithms are inspired by Darwin's theory of evolution. In order to realize the process of survival of the fittest, the algorithm retains eligible individuals through selection, crossover, and mutation, and eliminates other individuals. Evolution-based algorithms include Genetic Algorithm (GA) [1], Evolution Strategy (ES) [2], Differential Evolution Strategy (DE) [3], Probabilistic-Based Incremental Learning (PBIL) [4], etc. Swarm Intelligence (SI)-based algorithms are inspired by the intelligent behavior (hunting methods, moving routes) of biological swarm. For example, Particle Swarm Optimization (PSO) algorithm [5] simulates the foraging behavior of bird swarm, treating each bird as a particle. The algorithm assumes that the bird swarm (particle swarm) only knows the distance to the food (optimal solution). The simple and effective method is to search for the area of the bird closest to the food. SI-based algorithms include Firefly Algorithm (FA) [6], Bat Algorithm (BA) [7], Bird Mating Optimization (BMO) Algorithm [8], Cuckoo Search (CS) [9], Grasshopper Optimization Algorithm (GOA) [10], Dragonfly Algorithm (DA) [11], Dolphin Echo Location Algorithm (DE) [12], Fruit fly Optimization Algorithm (FOA) [13], Krill Herd (KH) [14], Wasp Swarm Algorithm (WSA) [15], Bee Collecting Pollen Algorithm (BCPA) [16], etc. Physical phenomena based algorithms mainly include Black Hole (BH) Algorithm [17], Center Force Optimization (CFO) Algorithm [18], Gravity Search Algorithm (GSA) [19], Artificial Chemical Reaction Optimization Algorithm (ACROA) [20], Curve Space Optimization (CSO) Algorithm [21], Charged System Search (CSS) [22], Simulated Annealing (SA) Algorithm [23]. Some algorithms are also inspired by life or human society, such as the Fireworks Algorithm (FWA) [24], Imperialist Competition Algorithm (ICA) [25], Soccer league Competition (SLC) Algorithm [26], Brainstorm Optimization (BSO) Algorithm [27], Anarchic Society Optimization (ASO) [28], Social Based Algorithm (SBA) [29], Mine Blast Algorithm (MBA) [30], Teaching Learning Based Optimization (TLBO) [31], Taboo Search (TS) [32], etc. SI-based algorithms are superior to evolution based algorithms and physical phenomena based algorithms. Firstly, SI-based algorithms often retain the information of search space. For example, the Gray Wolf Optimizer (GWO) [33] sets levels based on the fitness value of individuals, and low-level individuals follow high-level individuals to hunt and migrate. Secondly, SI-based algorithms have fewer parameters and operators, which makes them suitable for most optimization problems. It is worth mentioning that mathematical rules based algorithms are novel algorithms that have been proposed in recent years. Mathematical rules based algorithms include the Basic Optimization Algorithm (BOA) [34-35], and the Sine Cosine Algorithm (CSA) [36].

The whale optimization algorithm (WOA) was proposed by Seyedali Mirjalili in 2016 [37]. Similar to other swarm intelligence-based algorithms, WOA simulates the predatory behavior of whale populations to establish the mathematical model. Due to simplicity and flexibility, it has attracted the attention of researchers. Researchers have proposed many improvements and applications of WOA. Mafarja M. et al. proposed two binary variants of WOA to solve the classification of the best feature subsets [38]. The first method uses tournament and roulette selection mechanisms to eliminate the accidental effects of WOA random operators. The second method uses crossover and mutation operations to enhance WOA. The results show that the proposed method can improve the classification accuracy. Kaveh A et al. proposed an enhanced whale optimization algorithm (EWOA) for truss optimization [39]. Experimental results prove that EWOA is more effective than WOA. Ling Y et al. used the Levy flight trajectory to update the position of population, and proposed WOA based on the Levy flight trajectory [40]. The statistical results show that the performance of LWOA is excellent, which can solve the identification problem of infinite

impulse response model. With a large number of application backgrounds, the flow shop scheduling problem is one of the most important scheduling problems. Abdel-Basset M. et al. introduced a local search strategy into WOA and proposed the hybrid whale algorithm (HWA) [41], which can improve the optimization results of flow shop scheduling problems. Prakash DB et al. applied WOA to find the optimal capacitor size and location for a typical radial power distribution system [42]. The results show WOA can reduce operating costs and power consumption. The algorithm is more efficient in maintaining the voltage distribution. Parameter extraction of solar photovoltaic (PV) models is a nonlinear multivariate optimization problem. Xiong G. et al. proposed an improved whale optimization algorithm (IWOA) [43]. The comparison results fully prove that IWOA can accurately extract the parameters of different PV models. The above literature shows that WOA can effectively solve different optimization problems. However, WOA still has the defects of low optimization accuracy, slow convergence speed, and easy to fall into local optimal values. The improvement method of WOA needs further research. In addition, according to no free lunch theorem NFL [44], an algorithm can only solve specific optimization problems. In other words, improved algorithms may outperform other algorithms. Therefore, the motivation and necessity of this paper are supported.

This paper proposes an improved WOA based on three spiral searching strategies and sine cosine operator with convergence factor. Firstly, three spiral search strategies were used to improve the search efficiency of the spiral update mechanism and population diversity in WOA. Secondly, this paper introduces an improved sine and cosine operator, which can adaptively balance the exploitation and exploration capabilities of the algorithm. The section of this paper are arranged as follows. Section 2 introduces the whale optimization algorithm. Section 3 introduces the improved WOA. Section 4 first selects the salp swarm algorithm (SSA) [44], firefly algorithm (FA), moth-flame optimization (MFO) algorithm [45], fireworks algorithm (FWA), sine cosine algorithm (SCA) and the improved WOA for comparison experiments. Then, the effectiveness of the improved strategies were verified. Finally, the improved WOA is applied to high-dimensional optimization and engineering optimization problems. Section 5 is the conclusion of this paper.

2. Whale Optimization Algorithm

2.1 Mathematical Model of Whale Optimization Algorithm

WOA regards the optimal solution of the optimization problem as a prey. Similar to whale recognition and surrounding prey, searching agents move towards the optimal solution through the location update mechanism [47]. The two-dimensional mathematical model of WOA is shown in Fig. 1, where (x, y) indicates the search agent, and (x^*, y^*) indicates the optimal solution (prey). In order to transform the two parts of whale predation into a mathematical model, WOA designed two position update mechanisms, namely the shrink enclosing mechanism and the spiral update mechanism. The choice of the two location update mechanisms is determined by the parameter p . If $p < 0.5$, the shrink-wrap mechanism is performed. If $p \geq 0.5$, the spiral update mechanism is performed. The mathematical model of the whale optimization algorithm can be described as:

$$\vec{X}(t+1) = \begin{cases} \vec{X}_p(t) - \vec{A} \cdot |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| & \text{if } p < 0.5 \\ \vec{D} \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}_p(t) & \text{if } p \geq 0.5 \end{cases} \quad (1)$$

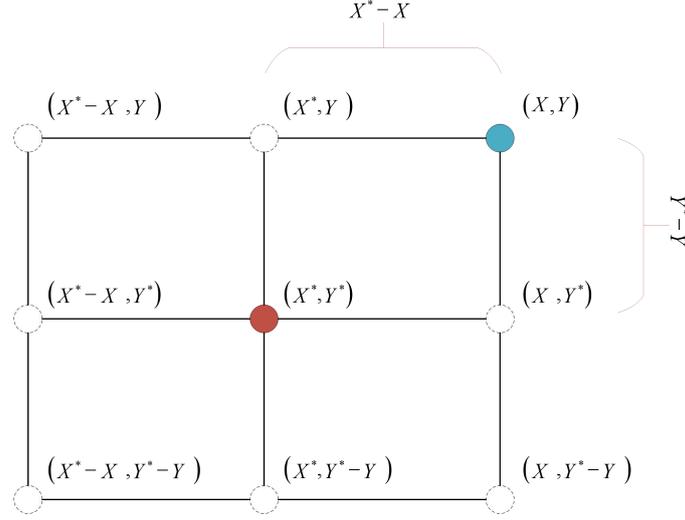


Fig. 1 2D mathematical model of WOA

2.1.1 Shrink Enclosing Mechanism

The shrink enclosing mechanism is one of the position update mechanisms of WOA in order to implement the process of whale predation, which can be described as:

$$\bar{X}(t+1) = \bar{X}_p(t) - \bar{A} \cdot |\bar{C} \cdot \bar{X}_p(t) - \bar{X}(t)| \quad (2)$$

where, \bar{X} indicates the position of the search agent, t is the current number of iterations, $\bar{X}_p(t)$ indicates the position of the optimal solution. $\bar{A} \cdot |\bar{C} \cdot \bar{X}_p(t) - \bar{X}(t)|$ indicates the indentation distance, which is used to calculate the distance between the search agent and the optimal solution. \bar{A} and \bar{C} are used to control the indentation distance, which can be calculated by:

$$\bar{A} = 2\bar{a} \cdot \bar{r} - \bar{a} \quad (3)$$

$$\bar{C} = 2 \cdot \bar{r} \quad (4)$$

where, \bar{r} is a random number in $[0,1]$. In order to realize the process of whale surrounding the prey, \bar{a} decreases from 2 to 0 with the iteration, which drives the search agent closer to the target value.

In addition, the whales will search for prey randomly. In order to achieve this process, the algorithm randomly selects a search agent to search the area outside the target value, that is to say:

$$\bar{X}(t+1) = \bar{X}_{rand}(t) - \bar{A} \cdot |\bar{C} \cdot \bar{X}_{rand}(t) - \bar{X}(t)| \quad (5)$$

where, $\bar{X}_{rand}(t)$ indicates a search agent that is randomly selected.

The shrink enclosing mechanism can realize the search method of enclosing prey and random hunting. The choice of the two search methods is determined by the coefficient vector \bar{A} . The model of the shrink enclosing mechanism is shown in Fig. 2. When $|\bar{A}| \leq 1$, the search agent performs a shrinking envelop on the target value so as to increase the local exploitation capability of the algorithm. When $|\bar{A}| > 1$, the search agent randomly searches for areas other than the target value. The random search method can reduce the probability of the algorithm falling into a local optimum, which reflects the global exploration capability. Two search methods of the shrink enclosing

mechanism can find a balance between exploitation and exploration through the parameter \bar{A} .

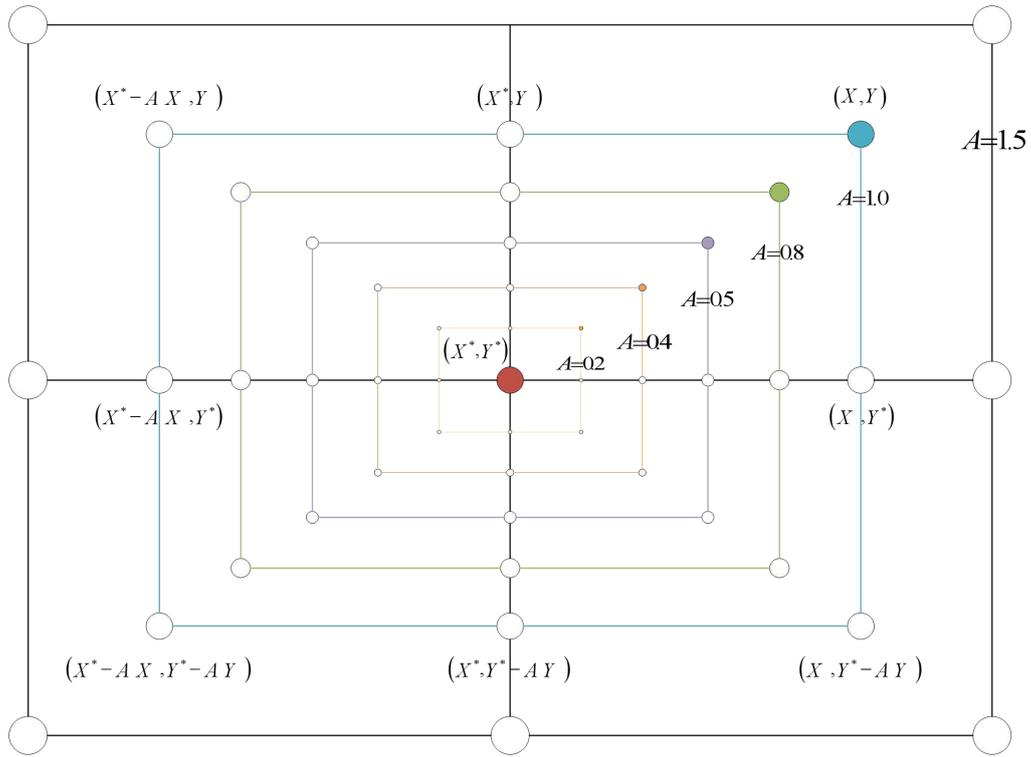


Fig. 2 Shrink enclosing mechanism model

2.1.2 Spiral Update Mechanism

Another position update mechanism of WOA is the spiral update mechanism. This method first calculates the distance between the search agent and the target value. Then, a logarithmic spiral is generated between the search agent and the target value. Search agents can implement location updates along a spiral route. The spiral update mechanism is described as follows:

$$\bar{X}(t+1) = \bar{D} \cdot e^{bl} \cdot \cos(2\pi l) + \bar{X}_P(t) \quad (6)$$

$$\bar{D} = \bar{X}_P(t) - \bar{X}(t) \quad (7)$$

where, \bar{D} indicates the distance between the search agent and the target value, b is a constant to control the shape of the spiral, l is a random number in $[-1, 1]$ to control the location update effect of the search agent.

The spiral update mechanism model is shown in Fig. 3, which shows not only the search method of the spiral route, but also the influence of the parameter l on the search position. When the parameter l is in $[0.5, 1]$, the algorithm suppresses the effect of the search agent moving closer to the target value. The search of the search agent beyond the target value reflects the exploration capability of the algorithm. When the parameter l is in $[-1, 0.5]$, the algorithm enhances the effect of the search agent moving closer to the target value. The search agent explores in the target value area, which reflects the exploitation ability of the algorithm.

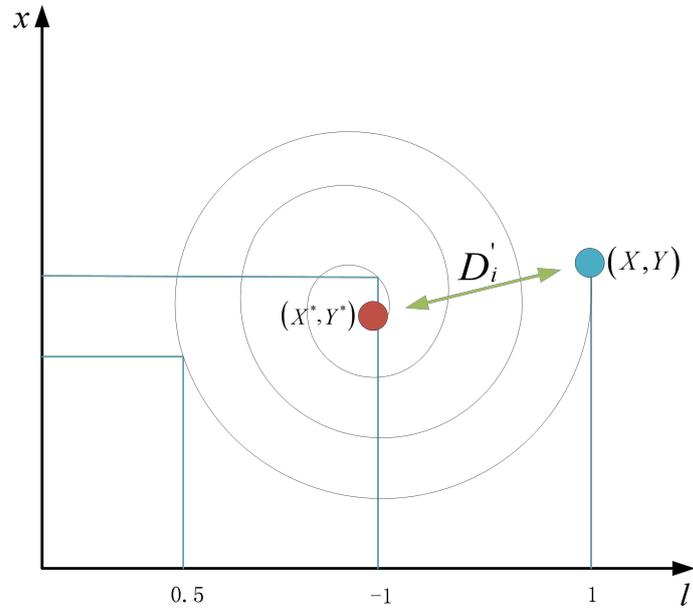


Fig. 3 Spiral update mechanism model

2.2 Pseudo Code of WOA

The pseudo code of WOA is described as follows and the flowchart of WOA is shown in Fig. 4 .

Initialize the whales population $X_i (i = 1, 2, 3 \dots n)$

Calculate the fitness of each search agent

X_p =the best search agent

$t=1$

While ($t < \text{maxmum numer of iterations}$)

For each search agent

Update a , A , C , l , and p

If ($p < 0.5$)

If ($|A| < 1$)

Update the position of search agent by the Eq.(2)

Else If ($|A| \geq 1$)

Update the position of search agent by the Eq.(5)

End If

Else If ($p > 0.5$)

Update the position of the search agent by the Eq.(6)

End If

End For

$t=t+1$

End While

Return X_p

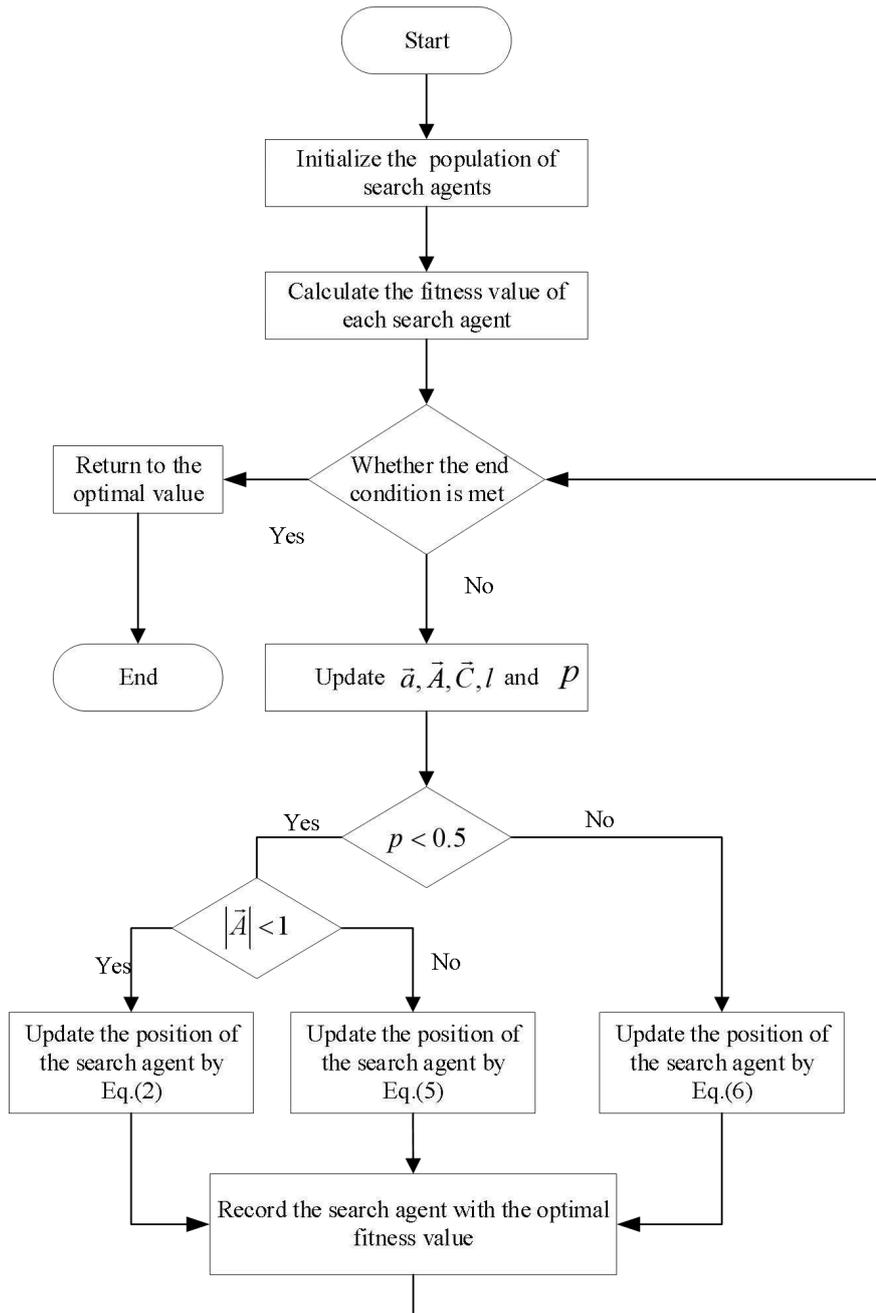


Fig. 4 Flow chart of WOA

As a swarm intelligence optimization algorithm, the structure of WOA has simplicity. At the same time, the mathematical model (shrink enclosing mechanism and the spiral update mechanism) of WOA designed parameters for balanced exploitation and exploration. In addition, the initialized random solutions are continuously improved through the location update mechanism to avoid falling into a local optimum. Finally, WOA has an information exchange mechanism. Search agents can update locations based on target values, increasing the interactivity of the population. However, the shortcomings of WOA also deserve attention. First, the parameters of balanced exploitation and exploration are randomly generated, and whether it brings contingency to the algorithm. Second, how to further improve the performance of WOA is an important research direction. Finally, whether the design of

the shrink enclosing mechanism and the spiral update mechanism are flawed.

3. Improved WOA Based on Spiral Search Strategy and Sine Cosine Operator with Convergence Factor

3.1 Spiral Search Strategy

The moth-flame optimization (MFO) algorithm [46] firstly proposed a search strategy of logarithmic spiral routes, which is inspired by the laterally positioned moth trajectory. The logarithmic spiral path can make the moth effectively fly to the light source at a fixed angle. In the WOA, the spiral update mechanism creates a logarithmic spiral route between the search agent and the target value. The search agent can move towards the target value along the logarithmic spiral. The logarithmic spiral is also called equiangular spiral, which refers to a spiral whose distance increases in geometric progression. The two-dimensional image of a logarithmic spiral is shown in Fig. 5. The logarithmic spiral is still a logarithmic spiral after various suitable transformations, which can be described as:

$$\begin{cases} x = a \cdot e^l \cdot \cos(n\pi l) \\ y = a \cdot e^l \cdot \sin(n\pi l) \end{cases} \quad (8)$$

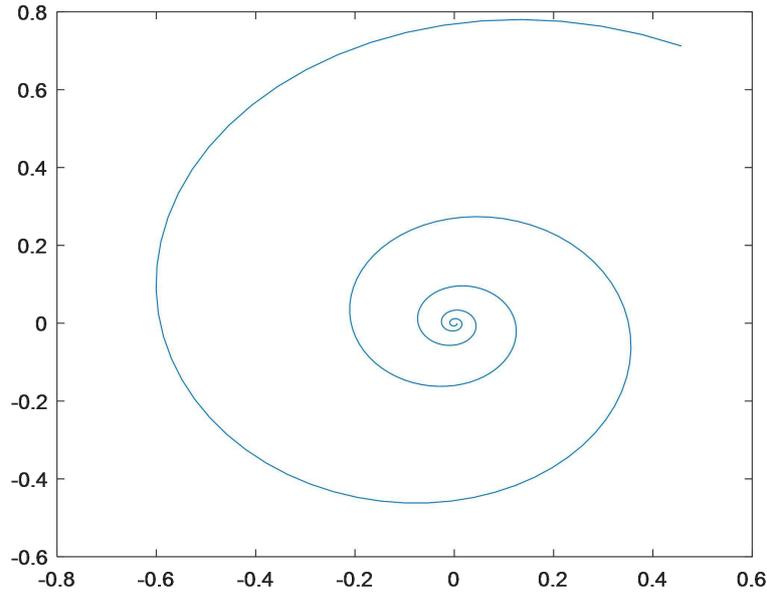


Fig. 5 2D image of logarithmic spiral

As the distance of the spirals increases in geometric progression, the logarithmic spiral path generated between the search agent and the target value may exceed the searching space. Spiral routes beyond the search space are invalid, and search agents will also fail along invalid routes. The failed search agent reduces the diversity of the population. In addition, due to the large distance of the spirals, the distribution position of the logarithmic spiral in space is sparse, which reduces the population diversity of the search agent. Aiming at the shortcomings of the logarithmic spiral, this paper proposes three different spiral search strategies, namely Archimedes spiral search, Rose spiral search and Hypotrochoid spiral search.

1) Archimedes spiral searching strategy

Archimedes spiral was named after the Greek mathematician Archimedes in the third century BC. The Archimedes spiral is a trajectory generated when a point leaves a fixed point at a uniform speed while rotating around the fixed point at a fixed angular velocity. When the peripheral speed and the linear speed double at the same time, the shape of the Archimedes spiral will not change. Therefore, the Archimedes spiral belongs to the constant velocity ratio spiral. At the same time, the Archimedes spiral is also called an equidistant spiral because it expands equidistantly in each rotation cycle. Two-dimensional image of the Archimedes spiral is shown in Fig. 6. The parametric equation of the Archimedes spiral is described as follows:

$$\begin{cases} x = a + b \cdot l \cdot \cos(2\pi l) \\ y = a + b \cdot l \cdot \sin(2\pi l) \end{cases} \quad (9)$$

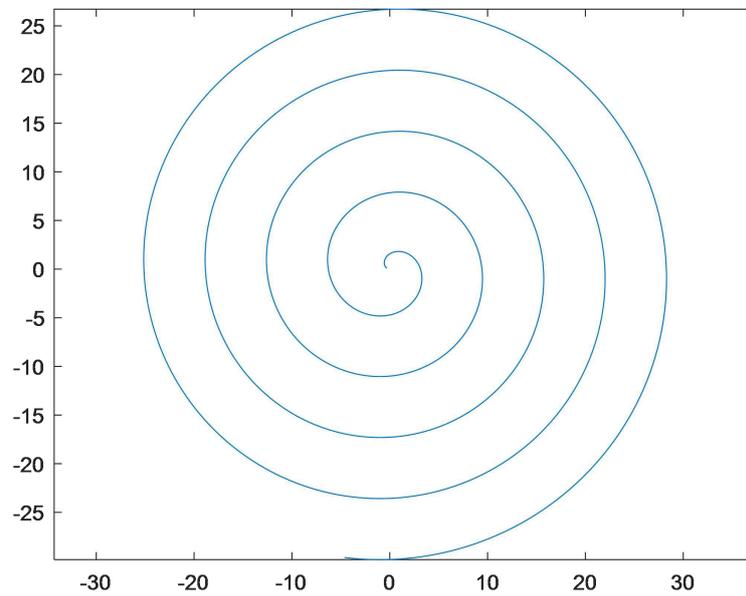


Fig. 6 2D image of Archimedes spiral

2) Rose spiral searching strategy

Rose spiral is a periodic arc spiral. The parameters controls the shape of the rose spiral. Among them, parameter a determines the length of the petals, and parameter n determines the number, size, and length period of the petals. Different parameters can generate rose spirals with different structures. Two-dimensional image of a rose petal with four petals is shown in Fig. 7. The parametric equation of the rose spiral is described as follows.

$$\begin{cases} x = a \cdot \cos(n\pi l) \cdot \sin(l) \\ y = a \cdot \sin(n\pi l) \cdot \cos(l) \end{cases} \quad (10)$$

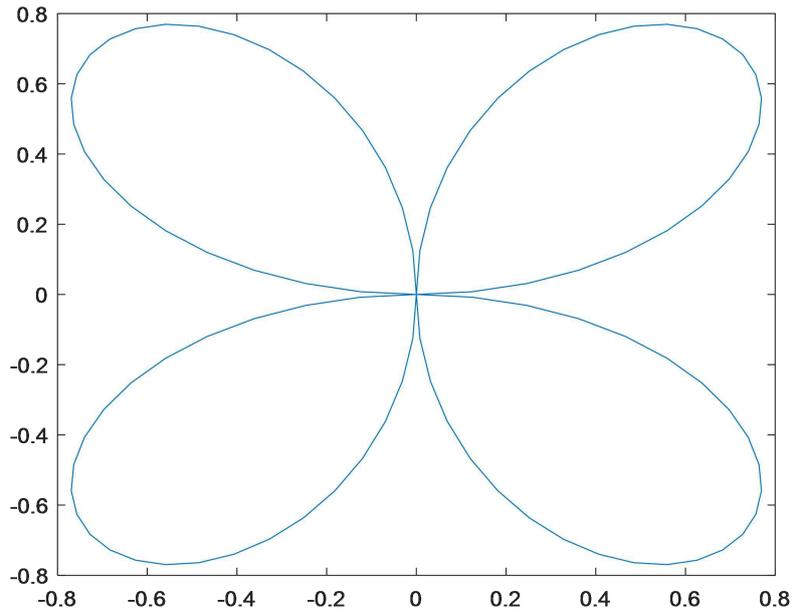


Fig. 7 2D image of rose spiral

3) Hypotrochoid spiral searching strategy

Hypotrochoid spiral is a trace obtained by tracking a point attached to a circle with radius a that rolls around the inside of a fixed circle with radius b . The distance from this point to the center of the inner scrolling circle is c . Two-dimensional image of the Hypotrochoid spiral is shown in Fig. 8. The parametric equation of the Hypotrochoid spiral is described as follows.

$$\begin{cases} x = (a-b) \cdot \cos(n\pi l) + c \cdot \cos\left(\frac{a-b}{b}n\pi l\right) \\ y = (a-b) \cdot \sin(n\pi l) - c \cdot \sin\left(\frac{a-b}{b}n\pi l\right) \end{cases} \quad (11)$$

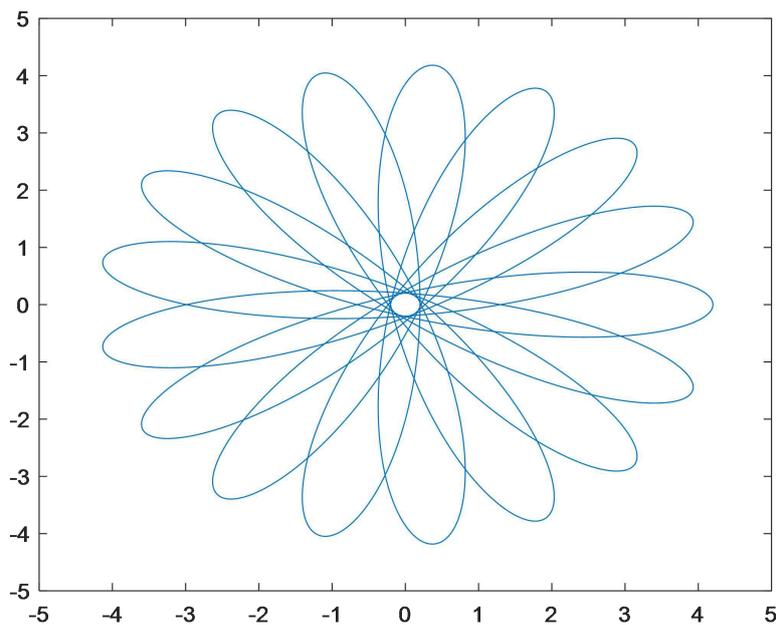


Fig. 8 2D image of Hypotrochoid spiral

3.2 Sine Cosine Algorithm

Sine Cosine Algorithm (SCA) is a mathematical rule-based algorithm inspired by sine and cosine functions in mathematics. Its advantage is that it can well balance the global exploration and local exploitation capabilities. The sine function is similar to the cosine function. The sine function and the cosine function are converted to each other through the induced formula or a half-angle formula. That is to say the sine function and cosine function are converted to each other by translation. Exploration and exploitation are two indispensable capabilities for optimization algorithms. In order to balance the two phases and show their combined ability as much as possible, SCA uses sine search for global exploration and cosine search for local exploitation. These two search methods can be defined as:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \cdot \sin(r_2) \cdot |r_3 p_i^t - X_i^t| & \text{if } r_4 < 0.5 \\ X_i^t + r_1 \cdot \cos(r_2) \cdot |r_3 p_i^t - X_i^t| & \text{if } r_4 \geq 0.5 \end{cases} \quad (12)$$

where, X_i^t indicates the position of the i -th dimension individual in the t -th iteration. p_i^t indicates the position of the target value of the i -th dimension in the t -th iteration. Parameter r_2 is a random number in $[0, 2\pi]$. Parameter r_3 provides random weights for the individual, which is used to strengthen ($r_3 > 1$) or weaken ($r_3 < 1$) the moving distance of the individual. Parameter r_4 is a random number in $[0, 1]$, which is used to make the two search methods equally selected. Parameter r_1 is used to guide the movement direction of individual, which can be calculated by:

$$r_1 = a - t \frac{a}{T} \quad (13)$$

where, t indicates the current number of iterations, T indicates the maximum number of iterations, and a is a constant and $a = 3$.

It can be seen from Eq. (13) that parameter r_1 has a large value in the early stage of the iteration, which can drive individuals to search in a wide range outside the target value. The global exploration capability of the algorithm has been reflected. With the iteration of the algorithm, the parameter r_1 gradually becomes smaller, which leads the individual to move closer to the target value region. The local exploitation ability of the algorithm is reflected. Eq. (11) combines sine search and cosine search. This model is shown in Fig. 9, which shows that parameter r_1 can guide individuals to search for areas outside the target value, and can also be used to guide individuals to search for areas inside the target value. This mechanism adaptively balances exploration and exploitation. It should be noted that by expanding the range of the sine and cosine functions, these two searching methods can expand the search range to suit different search spaces. The wide search range guarantees that the algorithm can search both in the target value area and the area outside the target value. The sine and cosine functions in the range $[-2, 2]$ are shown in Fig. 10.

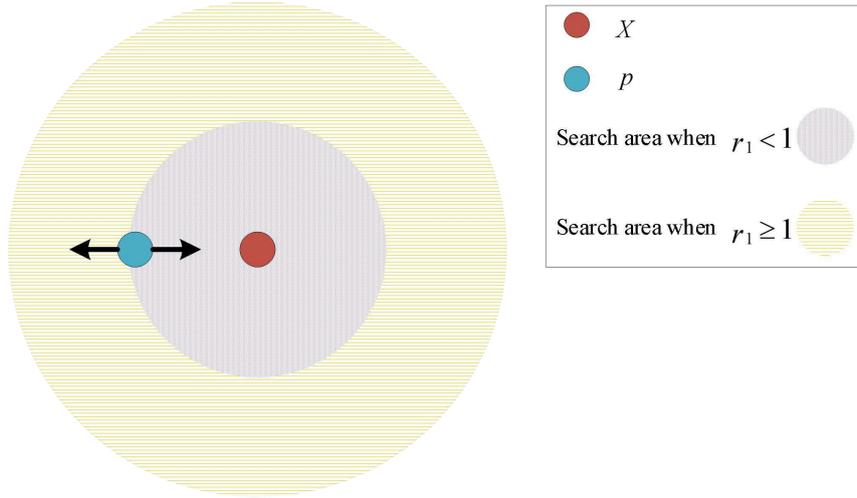


Fig. 9 The model combining sine and cosine search

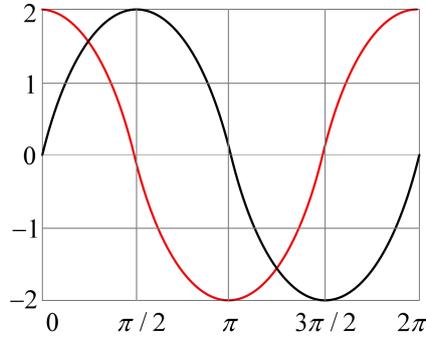


Fig. 10 Sine and cosine function with range of [-2,2]

3.3 Improved WOA Based on Spiral Searching Strategy and Sine Cosine Operator with Convergence Factor

Logarithmic spiral routes may exceed the searching space. At the same time, the distribution of logarithmic spirals in space will reduce the population diversity of search agents. Aiming at the shortcomings of the spiral update mechanism, this paper proposes three different spiral search strategies, namely Archimedes spiral search, Rose spiral search and Hypotrochoid spiral search. The Archimedes spiral is an equidistant spiral, so it does not exceed the search space. In addition, the Archimedes spiral route can search more locations, which can effectively improve the population diversity of search agents. After introducing the Archimedes spiral searching strategy, Eq. (6) can be updated as follows:

$$\bar{X}(t+1) = \bar{D} \cdot bl^4 \cdot \sin(2\pi l) + \bar{X}_p(t) \quad (14)$$

The rose spiral is a periodic arc spiral. From the two-dimensional image, it can be seen that the route of the rose spiral is wider than that of the logarithmic spiral, which increases the location diversity of the search agent. After introducing the rose spiral search, Eq. (6) can be defined as:

$$\bar{X}(t+1) = \bar{D} \cdot \sin(l) \cdot \cos(2\pi l) + \bar{X}_p(t) \quad (15)$$

Hypotrochoid spiral is internal trajectories generated along a circle, so it does not exceed the searching space. In addition, compared with the logarithmic spiral, the two-dimensional image shows that the route distribution of the Hypotrochoid spiral can search most areas in the space. After introducing the Hypotrochoid spiral searching strategy, Eq. (6) can be updated as:

$$\vec{X}(t+1) = \vec{D} \cdot \sin(2\pi l) - 0.3 \cdot \vec{D} \cdot \cos(2\pi l) + \vec{X}_p(t) \quad (16)$$

The mathematical model of WOA (shrink enclosing mechanism and spiral update mechanism) uses parameter r_1 to control the transition between exploitation and exploration. However, the generation of parameter r_1 is random, which brings randomness to the algorithm. In order to further improve the optimization ability of the WOA, this paper proposes an improved sine cosine operator. Firstly, the improved sine cosine operator improves parameter r_1 of the sine cosine algorithm, which improves the convergence effect of the search agent. Secondly, this paper proposes the merit based strategy to eliminate the shortcomings of the SCA in determining the search methods by probability. It can be seen from Eq. (11) that the purpose of designing parameter r_1 of SCA is to make the searching agent converge from the exploration stage to the exploitation stage. However, the convergence effect of the parameter r_1 needs to be verified. This paper proposes a convergence factor c_1 to replace parameter r_1 , that is to say:

$$c_1 = 2 \cdot e^{-(4t/T)^2} \quad (17)$$

where, t indicates the current number of iterations and T indicates the maximum number of iterations.

In order to verify the convergence effect of the convergence factor, c_1 and r_1 with different parameters are selected for comparison. The comparison results in Fig. 11 show that the convergence effect of c_1 is better than r_1 . Convergence factor c_1 can guide search agents to quickly transition from the exploration to the exploitation phase.

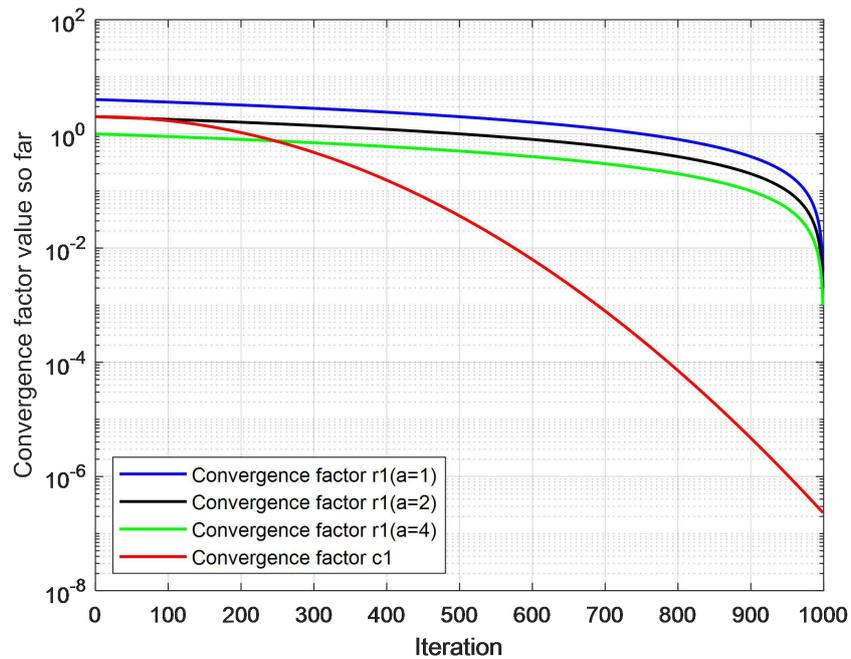


Fig. 11 Comparison of convergence effects

The sine and cosine searching methods are updated to the following equation after introducing the

convergence factor c_1 :

$$\bar{X}(t+1) = \bar{X}(t) + c_1 \cdot \sin(r_2) \cdot |r_3 \bar{X}_p(t) - \bar{X}(t)| \quad (18)$$

$$\bar{X}(t+1) = \bar{X}(t) + c_1 \cdot \cos(r_2) \cdot |r_3 \bar{X}_p(t) - \bar{X}(t)| \quad (19)$$

where, parameters r_2 and r_3 have the same meaning as Eq. (11). Fig. 12 shows that the convergence factor c_1 can make the sine and cosine functions gradually decrease with iteration, which ensures that the search agent develops towards the area of the target value.

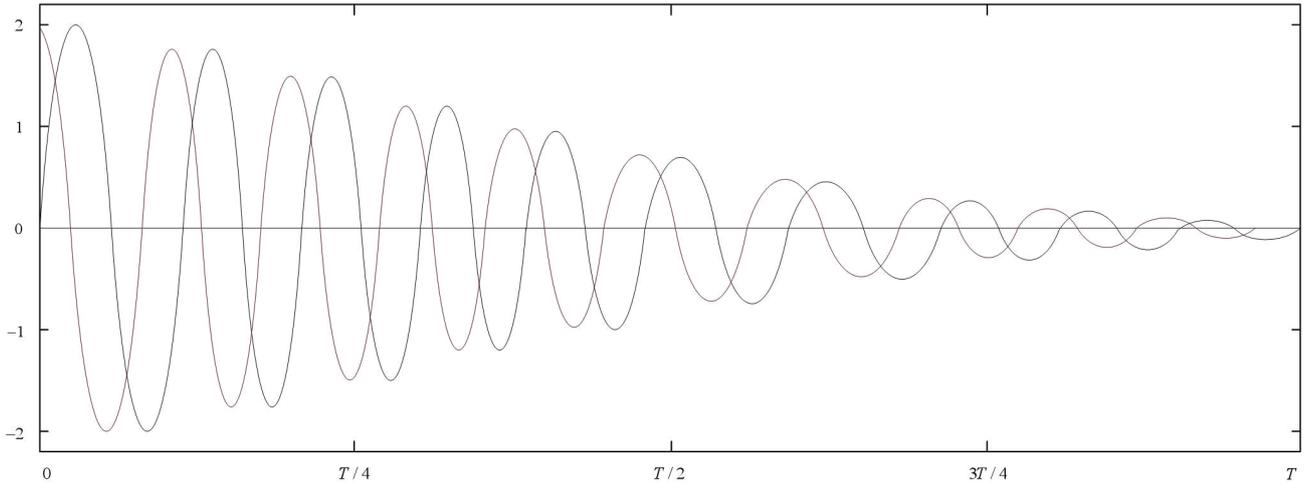


Fig. 12 The model of sine and cosine function by reducing ranges

In addition, improved sine cosine operator introduce the merit based strategy. SCA adopts sine search for global exploration and cosine search for local exploitation. The probability of two searching methods being selected is determined by the random parameter r_4 . When $r_4 < 0.5$, the algorithm uses sine search for position update. When $r_4 \geq 0.5$, the algorithm uses cosine search. This mechanism guarantees the fairness of two search methods. However, the equal selection method has defects, that is, the selected search method is not as effective as the non-selected search method, which increases the randomness of the algorithm. The merit based strategy firstly performs two search methods and calculates fitness values. Then, two searches are retained based on fitness values. The merit based strategy does not use parameter r_4 , which avoids the shortcomings of randomness. The flowchart of the merit based strategy is shown in Fig. 13.

Improved sine cosine operator allow the function to be extended to accommodate a wider search range. At the same time, the search agent can search different areas based on the value returned by the function. Fig. 14 shows the effect of the function returned value in the range $[-2,2]$ on the search methods. In the range of $(1, 2]$ and $[-2, -1)$, the search agent performs global exploration in areas outside the target value. In the range of $[-1,1]$, the search agent performs local exploitation in the target value area. In addition, the convergence factor can also guide searching agents from the exploration phase to the exploitation phase. Therefore, the introduction of improved sine cosine operator ensures the transition between exploration and exploitation.

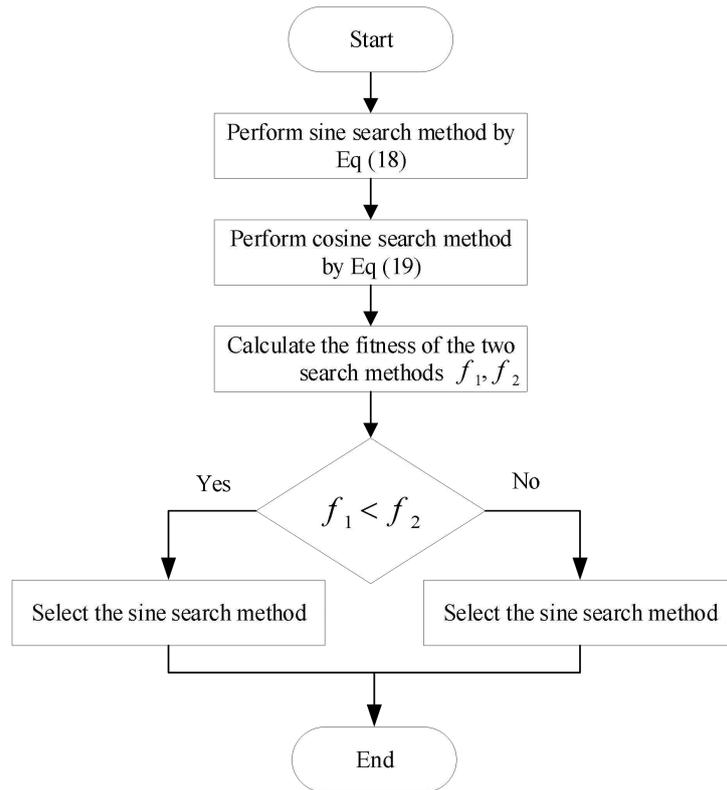


Fig. 13 Flow chart of merit based strategy

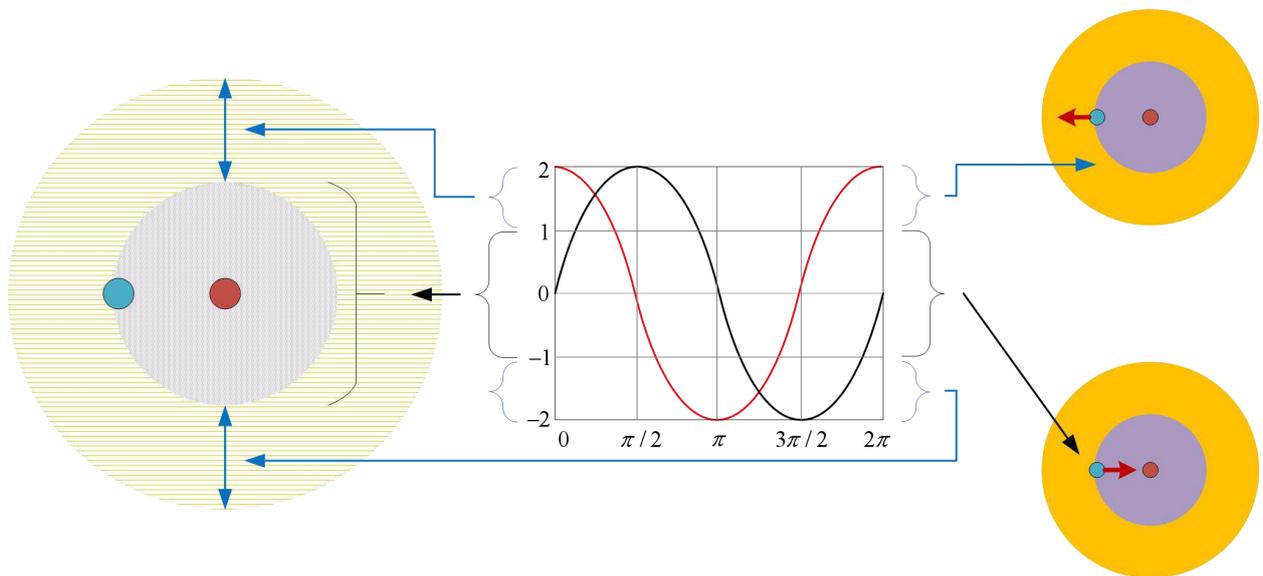


Fig. 14 Effect of the function returned value on the search area

The flowchart of the improved WOA is shown in Fig. 15 and its pseudo code is described as follows.

Initialize the whales population $X_i (i = 1, 2, 3 \dots n)$

Calculate the fitness of each search agent

X_p = the best search agent

$t=1$

While ($t < \text{maximum number of iterations}$)

For each search agent
 Update a , A , C , l , and p
 If ($p < 0.5$)
 If ($|A| < 1$)
 Update the position of search agent by the Eq.(2)
 Else If ($|A| \geq 1$)
 Update the position of search agent by the Eq.(5)
 End If
 Else If ($p > 0.5$)
 Perform the spiral search by the Eq.(14/15/16)
 End If
 End For
 Update c_1 , r_2 , and r_3
 Perform improved sine and cosine operator by the Eq.(18) and Eq.(19)
 Perform the merit based strategy by the Fig.13
 $t = t + 1$
 End While
 Return X_p

3.4 Time Complexity Analysis

Time complexity is used to describe the running time of the algorithm, which is expressed by a big O symbol. To calculate the time complexity, the number of operating units of an algorithm is usually estimated. Therefore, the time complexity of the optimization algorithm is related to the operating unit and algorithm structure. For WOA, the time complexity mainly depends on the number of search agents, the number of iterations, and the location update mechanism. The improved WOA introduces improved strategies based on WOA. The impact of the improved strategy on the time cost of the algorithm is unknown and needs to be analyzed.

The time complexity of each operation unit in WOA is described as follows.

- 1) The N search agents are distributed in the D -dimensional search space, which needs to run $N \cdot D$ times.
- 2) Calculate the fitness of search agent and select the optimal agent as the target value, which needs to run $[N \cdot (N - 1)]/2$ times.
- 3) Parameters a , A , C , l and p are updated, which needs to run 5 times.
- 4) The position update operation of N search agents in the D -dimensional space, which needs to run $N \cdot D$ times.
- 5) Output the optimal value needs to run 1 times.

Operation units has undergone T iterations. Therefore, the total time complexity of WOA is $O(WOA) = T \cdot [ND + (N^2 - N)/2 + 6]$.

The time complexity of each operation unit of the improved WOA algorithm is described as follows.

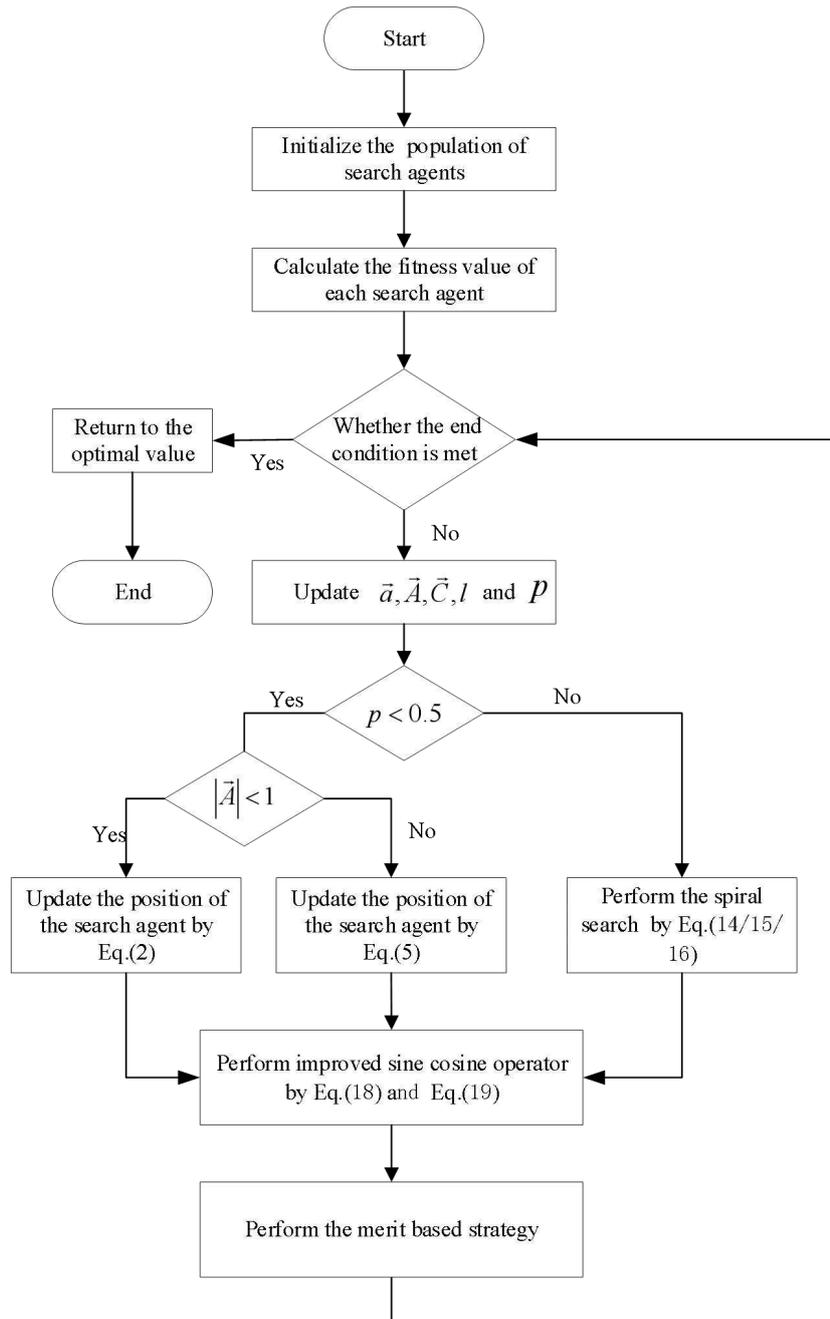


Fig. 15 Flow chart of improved WOA

- 1) The N search agents are distributed in the D -dimensional search space, which needs to run $N \cdot D$ times.
- 2) Calculate the fitness of search agent and select the optimal agent as the target value, which needs to run $[N \cdot (N - 1)]/2$ times.
- 3) Parameters a , A , C , l and p are updated respectively, which needs to run 5 times.
- 4) The position update operation of N search agents in the D -dimensional space, which needs to run $N \cdot D$ times.
- 5) Parameters c_1 , r_2 , and r_3 are updated, which needs to run 3 times.
- 6) N search agents perform the improved sine cosine operation in the D -dimensional space, which needs to

run $N \cdot D$ times.

7) Output the optimal value needs to run 1 times.

The total time complexity of the improved WOA is $O(\text{improved-WOA}) = T \cdot [ND + (N^2 - N)/2 + 9]$. From the time complexity analysis, it can be seen that the improvement strategy is simple and does not increase the calculation cost.

4. Simulation Experiments and Results Analysis

4.1 Benchmark Functions

The benchmark functions are used to verify the performance of the algorithms. The test function creates the searching space based on the number of variables, constraints, and dimensions. In general, the test functions have a global optimal solution, so it belongs to the single objective optimization problem. 22 test functions were used to evaluate the performance of the improved WOA. The test functions and their specific information are shown in Table 1. Functions $F_1 - F_7$ are unimodal functions with a global optimum and a small number of local optimums. The unimodal function can evaluate the exploitation ability and convergence speed of the algorithm. Unlike unimodal functions, multimodal functions $F_8 - F_{13}$ have multiple local optimums, and the algorithm is prone to fall into local optimums. Multimodal functions can therefore evaluate the global exploration capabilities of algorithm. In addition, the local optimal value of the multimodal function increases with increasing dimensions. The fixed-dimensional multimodal functions $F_{14} - F_{22}$ have fewer dimensions and have a small number of local optimal values. The feature of fixed-dimensional multimodal functions is that they provide different searching spaces and do not allow adjustment of dimensions. They can evaluate the optimization accuracy of the algorithm.

Table 1 Test functions and specific information

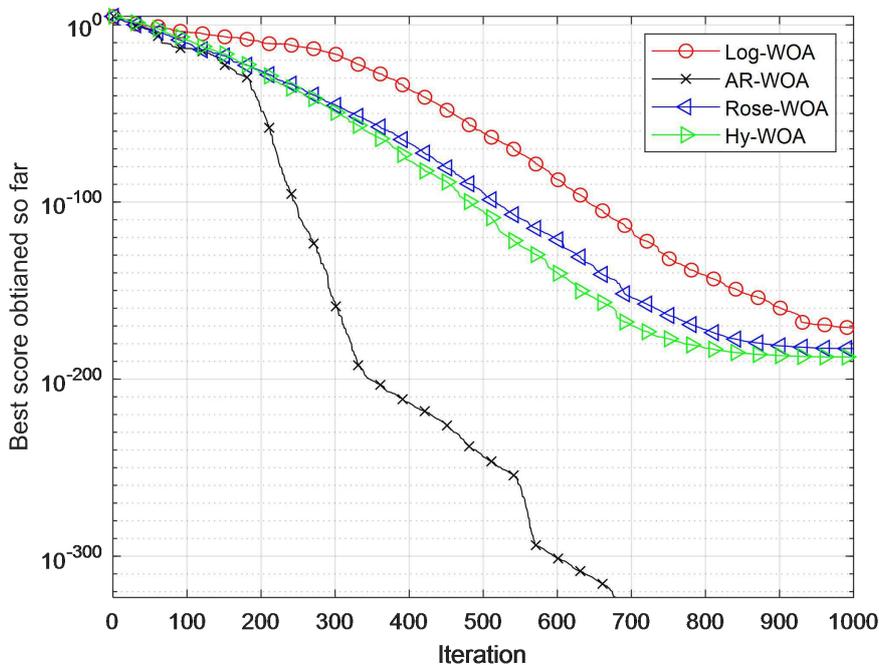
Function	Dim	Range	f_{\min}
$F_1(x) = \sum_{i=1}^n x_i^2$	30	[-100,100]	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10,10]	0
$F_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^j x_j \right)^2$	30	[-100,100]	0
$F_4(x) = \max_i \{ x_i , 1 \leq i \leq n \}$	30	[-100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	30	[-30,30]	0
$F_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30	[-10,10]	0
$F_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0,1)$	30	[-1.28,1.28]	0
$F_8(x) = \sum_{i=1}^n -x_i^2 \sin(\sqrt{ x_i })$	30	[-500,500]	-418.9829×Dim
$F_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0

$F_{10}(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	[-32,32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0
$F_{12}(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^n (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+4})]\right\}$ $+ \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50,50]	0
$F_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)]\}$ $+ (x_n - 1)^2 [1 + \sin^2(3\pi x_n)] + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	[-50,50]	0
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}\right)^{-1}$	2	[-65,65]	1
$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5,5]	0.00030
$F_{16}(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right) \cos x_1 + 10$	2	[-5,5]	0.398
$F_{17}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)]$ $\times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$	2	[-2,2]	3
$F_{18}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2\right)$	3	[1,3]	-3.86
$F_{19}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2\right)$	6	[0,1]	-3.32
$F_{20}(x) = -\sum_{i=1}^5 [(X - a)_i (X - a)_i^2 + c_i]^{-1}$	4	[0,10]	-10.1532
$F_{21}(x) = -\sum_{i=1}^7 [(X - a)_i (X - a)_i^2 + c_i]^{-1}$	4	[0,10]	-10.4028
$F_{22}(x) = -\sum_{i=1}^{10} [(X - a)_i (X - a)_i^2 + c_i]^{-1}$	4	[0,10]	-10.5363

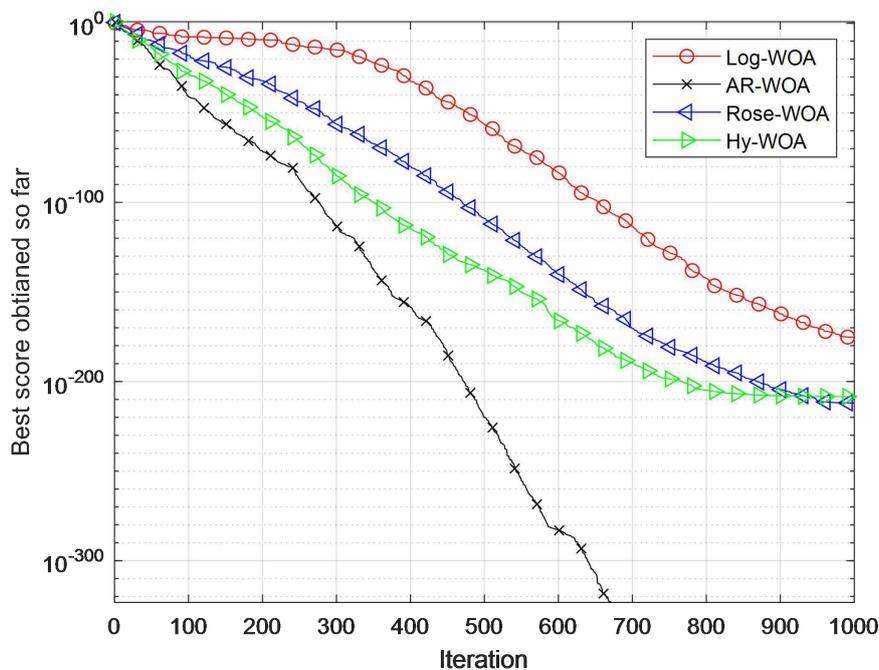
4.2 Comparison of Three Spiral searching Strategies

In order to improve the searching efficiency and the diversity of the population, this paper proposes three different spiral searching strategies. They are Archimedes spiral search (AR), rose spiral search (Rose) and Hypotrochoid spiral search (Hy). The logarithmic spiral (Log) of the original WOA and three spiral searching strategies are selected for comparison on some functions. The convergence curves in Fig. 16 shows that the Archimedes spiral search has the best convergence speed and optimization accuracy. For each function, the algorithm runs independently 10 times. The statistical results in Table 2 show the average accuracy and robustness of different spiral searching strategies. It can be seen from Table 2 that the Archimedes spiral search performs well

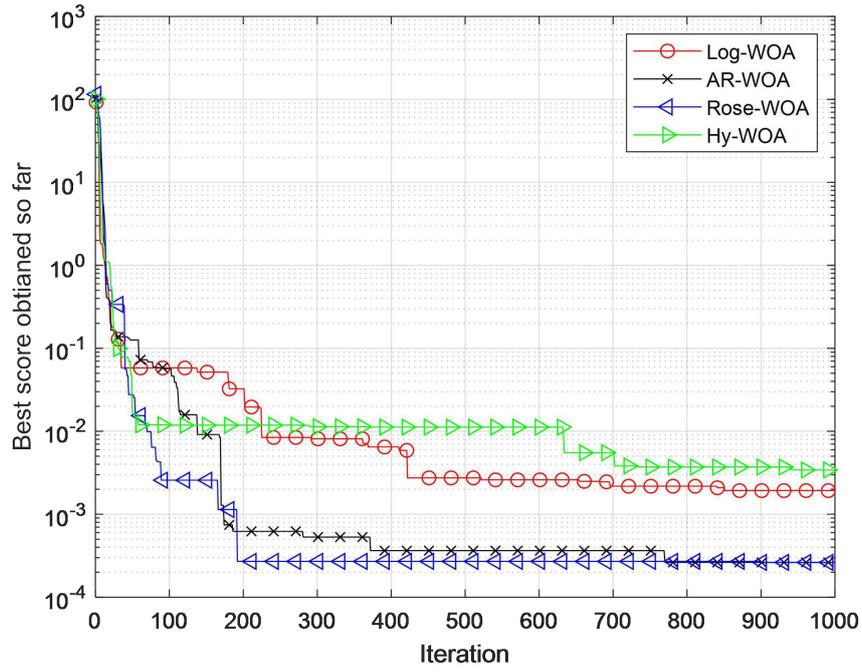
and is more competitive than other spiral search strategies. On the unimodal functions, all three spiral search strategies are better than the logarithmic spiral. However, the rose spiral search and the Hypotrochoid spiral search did not show an advantage in multimodal functions. Only the Archimedes spiral search showed advantages in both unimodal and multimodal functions. The Archimedes spiral search is an isometric spiral that does not exceed the searching space. At the same time, the route distribution of Archimedes spiral is wider and more reasonable, which can effectively improve the population diversity of search agents.



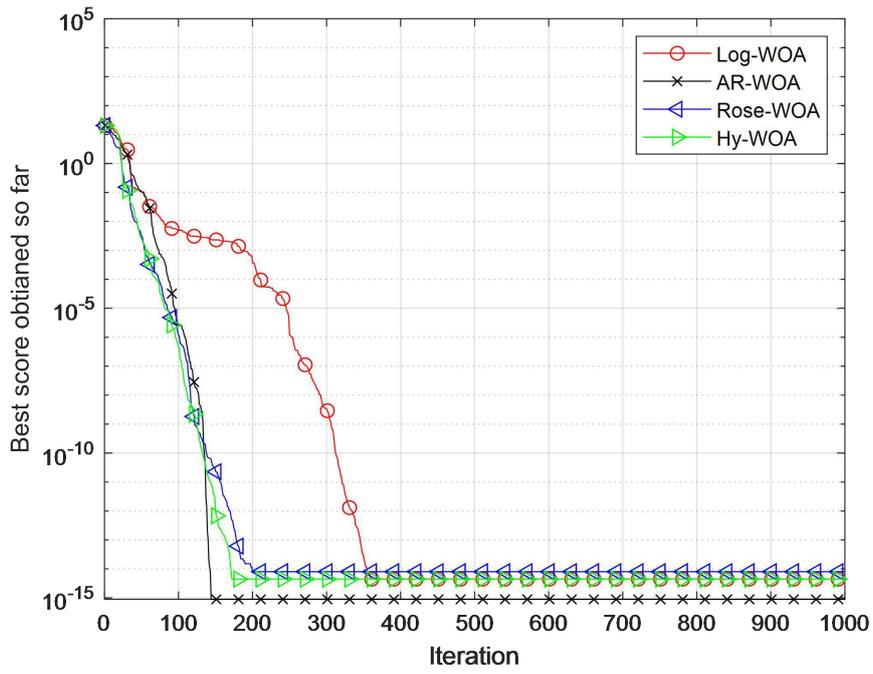
(1) Convergence curves of F_1



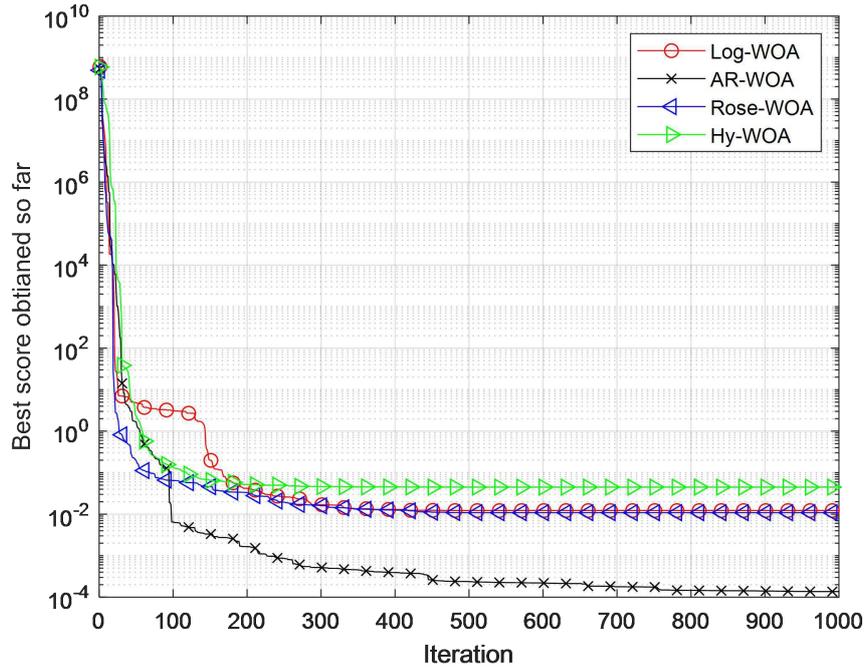
(2) Convergence curves of F_2



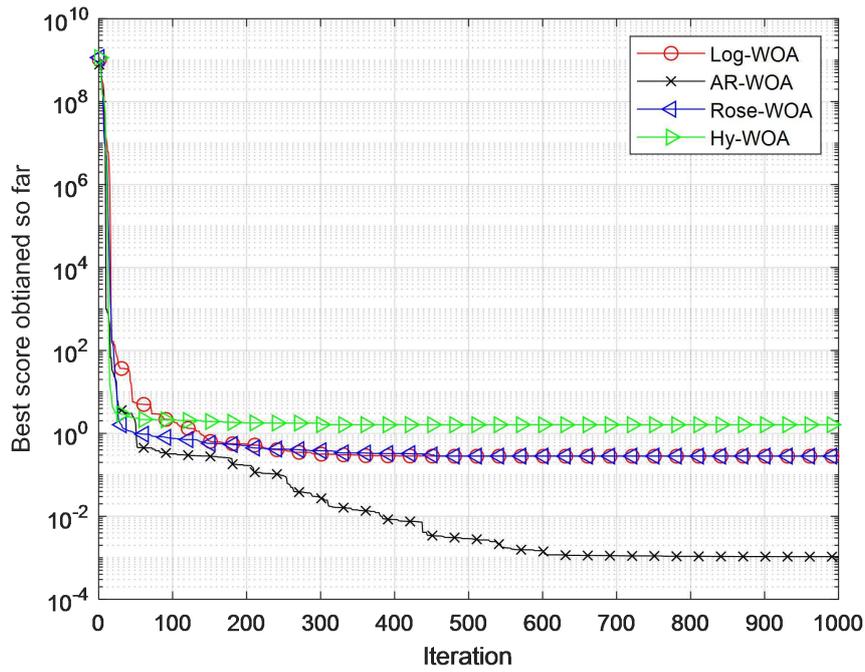
(3) Convergence curves of F_7



(4) Convergence curves of F_{10}



(5) Convergence curves of F_{12}



(6) Convergence curves of F_{13}

Fig. 16 Simulation results

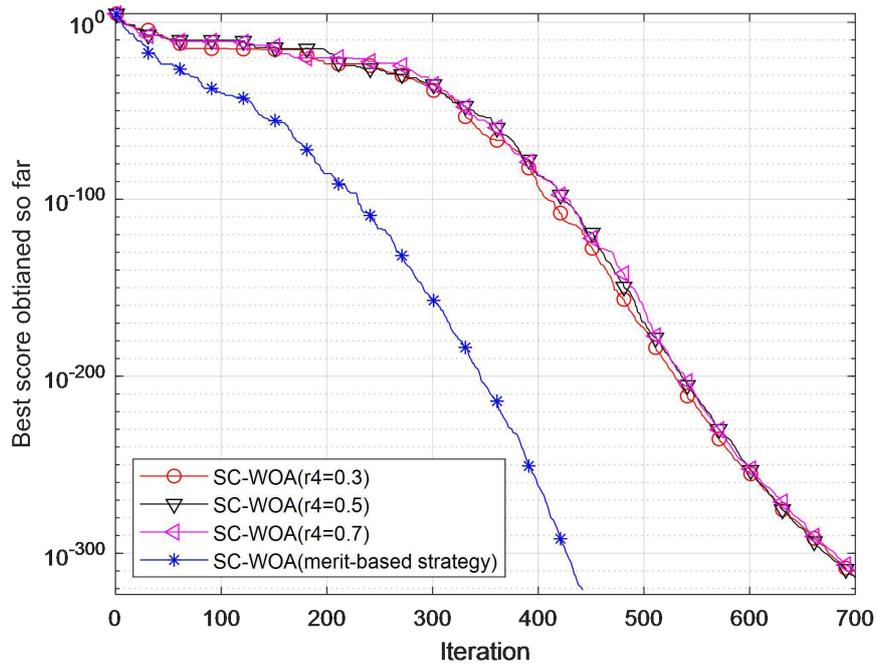
Table 2 Simulation performance comparison

Function		Log-WOA	AR-WOA	Rose-WOA	Hy-WOA
F1	Best	6.27E-176	0	3.59E-195	4.45E-202
	Ave	8.20E-169	2.54E-148	7.32E-177	6.39E-178
	Std	0	8.04E-148	0	0
Best		3.54E-180	0	8.36E-213	2.44E-221

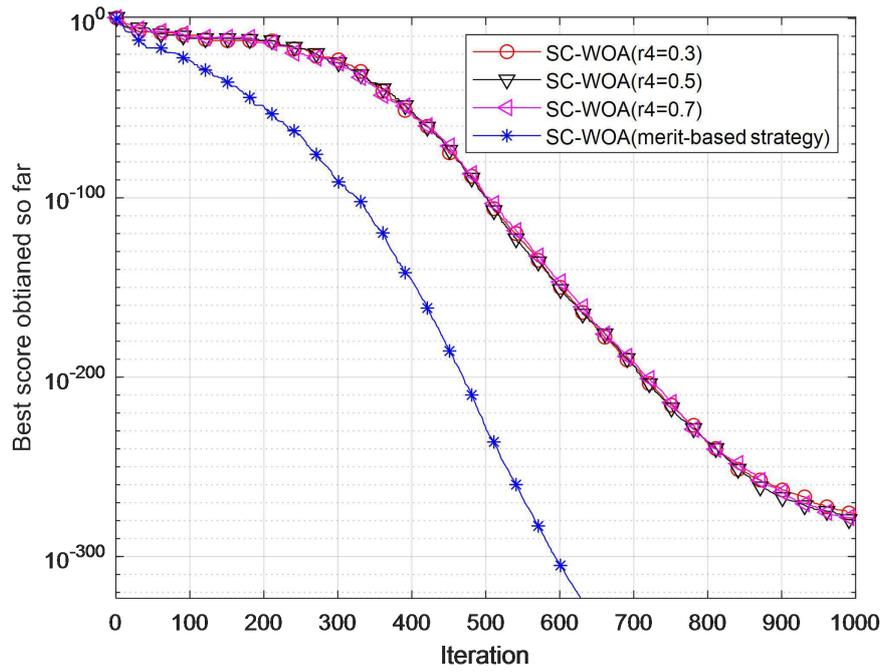
F2	Ave	6.83E-163	3.49E-247	8.56E-191	2.70E-199
	Std	2.22E-162	0	0	0
F7	Best	4.23E-04	6.98E-05	7.75E-05	1.50E-04
	Ave	2.14E-03	2.04E-03	1.07E-03	2.05E-03
	Std	2.26E-03	3.11E-03	1.43E-03	2.41E-03
F10	Best	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	Ave	3.02E-15	1.60E-15	4.09E-15	4.80E-15
	Std	2.48E-15	2.25E-15	2.62E-15	2.62E-15
F12	Best	3.11E-04	1.26E-04	3.12E-03	1.29E-02
	Ave	3.66E-03	2.56E-04	2.09E-02	4.36E-02
	Std	5.54E-03	1.28E-04	1.37E-02	2.61E-02
F13	Best	3.11E-03	3.48E-04	2.53E-01	5.06E-01
	Ave	6.42E-02	1.07E-02	8.54E-01	1.26E+00
	Std	6.85E-02	1.73E-02	3.59E-01	4.62E-01

4.3 Effectiveness Analysis of Improved Sine Cosine Operator

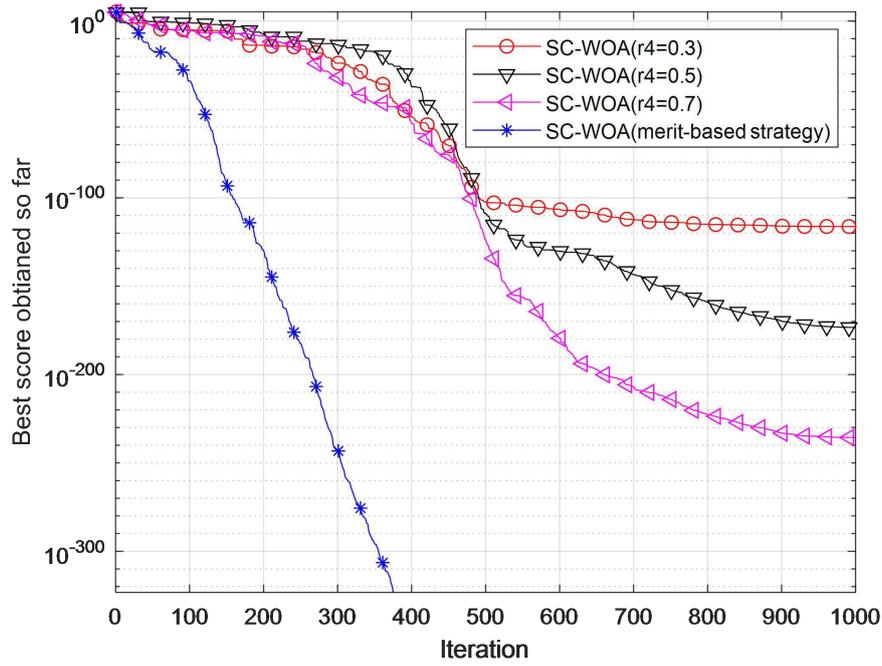
In order to further improve the exploration and exploitation capabilities of WOA, this paper proposes an improved sine cosine operator. Improved sine cosine operator first improves the parameter r_1 and improves the convergence effect of the search agent. Secondly, the merit based strategy is introduced to eliminate the shortcomings of the searching method determined by probability. The original sine cosine operator with different parameter r_4 and improved sine cosine operator are selected for comparison experiments. The convergence curves of some functions are shown in Fig. 17. For each function, the algorithm runs independently 10 times. The statistical results are listed in Table 3. The improved sine cosine operator have advantages in both unimodal and multimodal functions. The change of parameter r_4 does not significantly improve the performance of the sine cosine operator. The improved sine cosine operator has better optimization effect, and is significantly better than the original sine cosine operator in terms of convergence speed or optimization accuracy. Comparative experiments show that the convergence factor can drive the searching agent to switch from exploration to exploitation, which improves the convergence speed of the algorithm. In addition, the merit based strategy retains the optimal search method, which eliminates randomness and further improves the optimization accuracy of the algorithm.



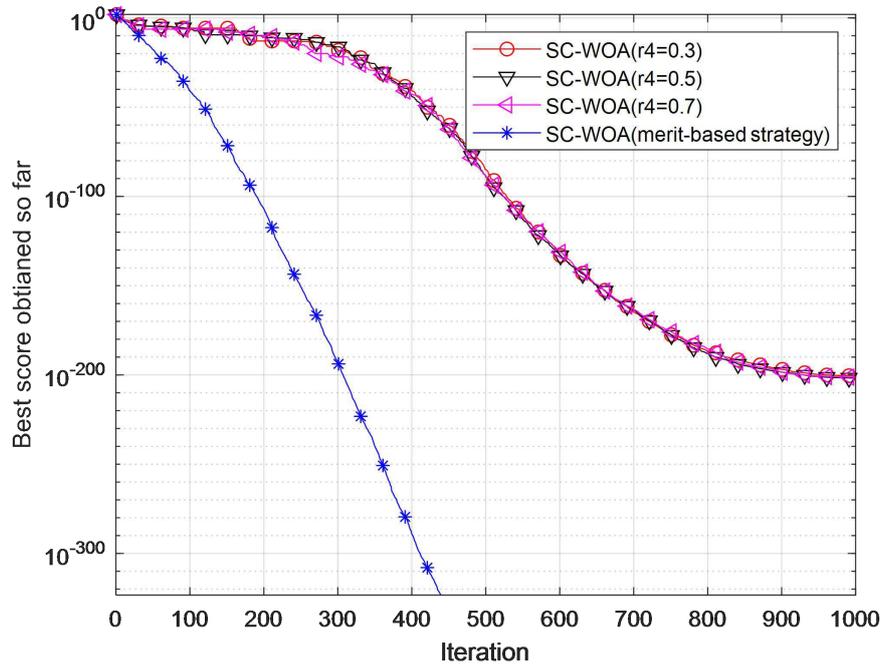
(1) Convergence curves of F_1



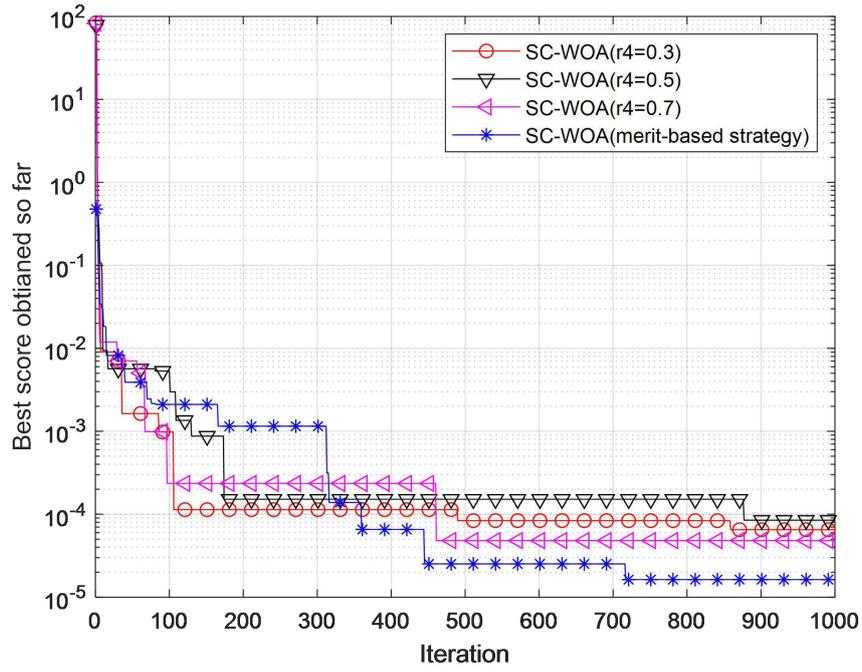
(2) Convergence curves of F_2



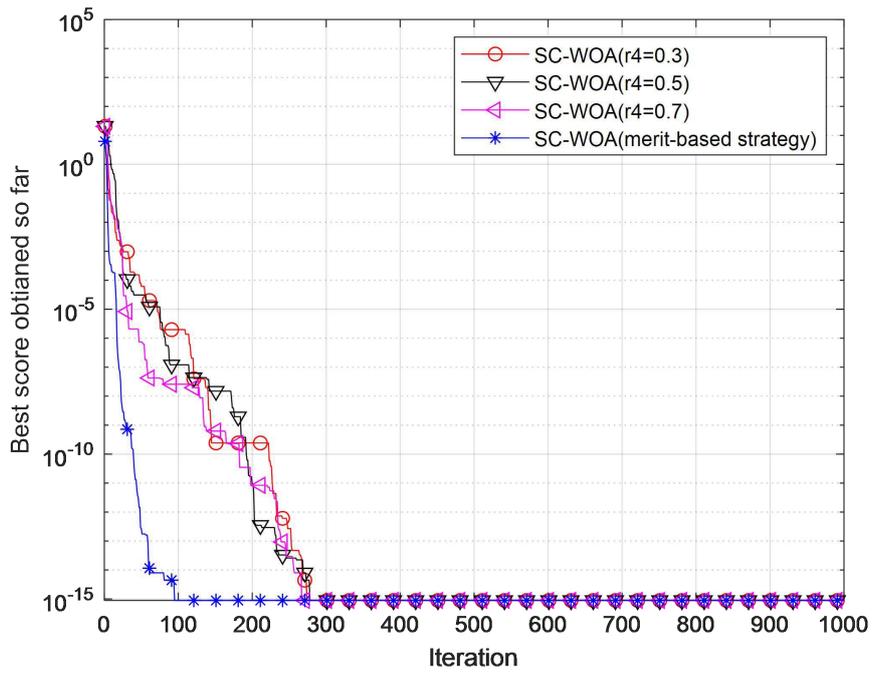
(3) Convergence curves of F_3



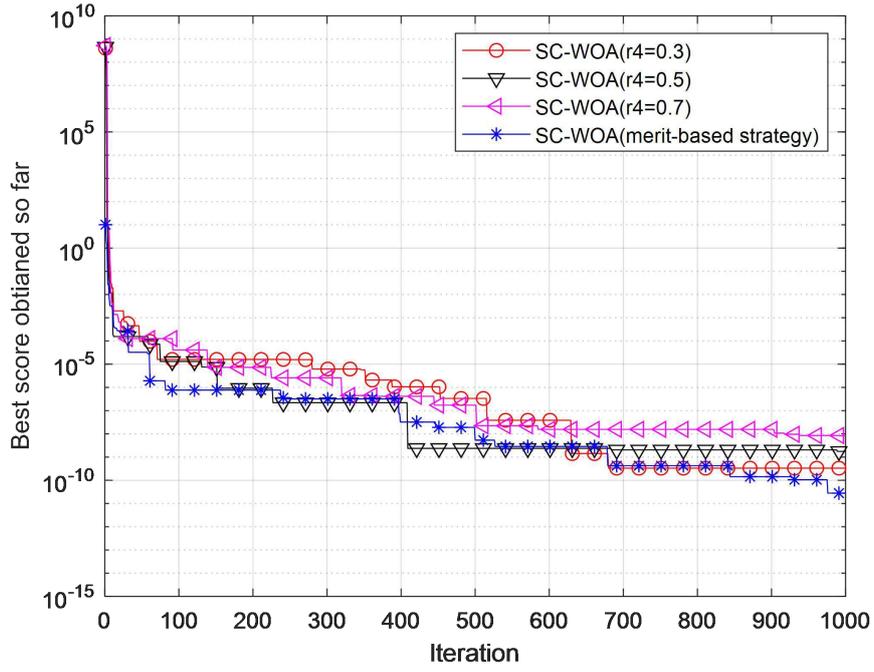
(4) Convergence curves of F_4



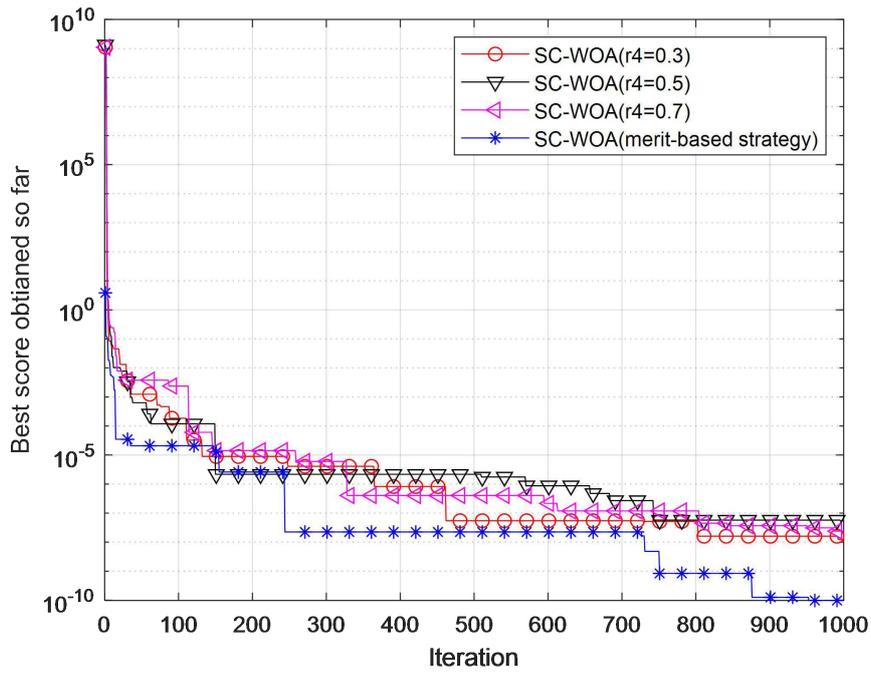
(5) Convergence curves of F_7



(6) Convergence curves of F_{10}



(7) Convergence curves of F_{12}



(8) Convergence curves of F_{13}

Fig. 17 Simulation results

Table 3 Simulation performance comparison

Function		SC-WOA			
		r4=0.3	r4=0.5	r4=0.7	merit-based strategy
F1	Best	0	0	0	0
	Ave	0	0	0	0
	Std	0	0	0	0

F2	Best	9.19E-283	5.36E-281	1.71E-284	0
	Ave	1.87E-268	9.99E-273	4.90E-271	0
	Std	0	0	0	0
F3	Best	2.07E-215	3.33E-200	3.25E-216	0
	Ave	5.86E-148	5.93E-134	6.33E-95	0
	Std	1.79E-147	1.87E-133	2.00E-94	0
F4	Best	4.67E-209	6.44E-207	5.72E-206	0
	Ave	1.24E-200	2.02E-192	2.64E-194	0
	Std	0	0	0	0
F7	Best	3.74E-06	5.41E-06	1.80E-06	2.73E-06
	Ave	8.04E-05	3.93E-05	7.41E-05	2.44E-05
	Std	8.59E-05	3.12E-05	9.52E-05	2.14E-05
F10	Best	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	Ave	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	Std	0	0	0	0
F12	Best	3.92E-12	3.64E-12	2.27E-11	3.91E-12
	Ave	2.57E-09	1.95E-09	4.38E-09	1.45E-10
	Std	4.71E-09	4.46E-09	4.56E-09	2.11E-10
F13	Best	9.74E-11	1.85E-10	4.34E-11	7.30E-12
	Ave	1.52E-08	4.63E-08	2.02E-08	8.24E-10
	Std	2.25E-08	5.62E-08	2.87E-08	1.09E-09

4.4 Comparison of Improved WOA with Other Algorithms

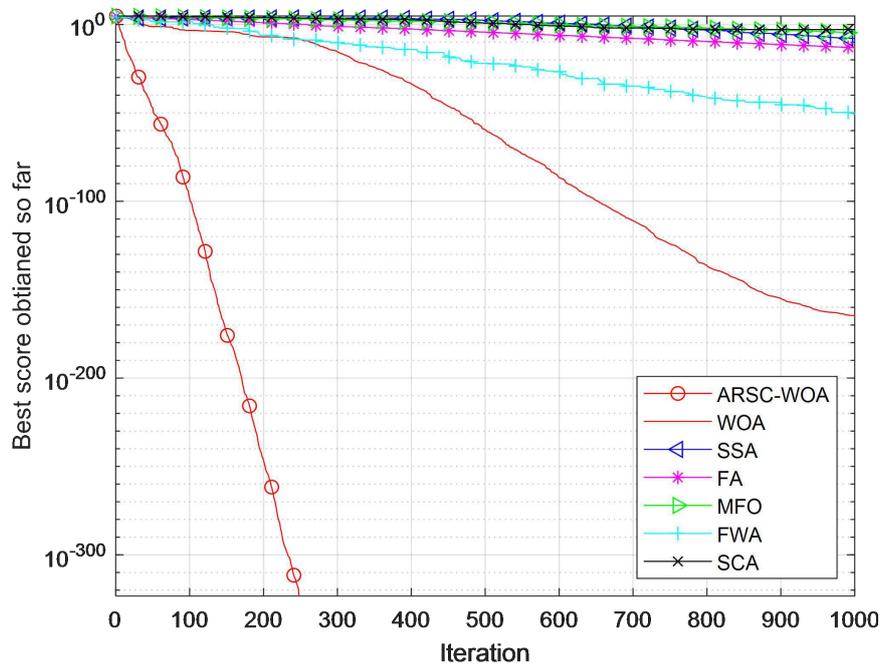
Section 4.2 has verified that the Archimedean spiral searching strategy has the best optimization effect. At the same time, the validity of the improved sine cosine operator is verified in Section 4.3. Therefore, in this section, an improved WOA based on the Archimedes spiral search strategy and sine cosine operator (ARSC-WOA) is selected in the experiments. In order to show the performance comparison between the improved WOA and other algorithms, this paper select the whale optimization algorithm (WOA), salp swarm algorithm (SSA), firefly algorithm (FA), moth-flame optimization (MFO) algorithm, fireworks algorithm (FWA), sine cosine algorithm (SCA) and improved WOA for comparison experiments. Fairness is the basic step to eliminate experimental errors, so the algorithm has uniform parameter settings. Table 4 shows the detailed parameter settings of the algorithms. The convergence curves under different algorithm are shown in Fig. 18. The convergence curves can intuitively show the convergence speed and optimization accuracy of the algorithms, but it can not statistics the average accuracy and robustness of the algorithms. For each function, the algorithm runs independently 10 times. Table 5 shows the mean and variance, which can show the average accuracy and robustness of the algorithms.

Table 4 The specific parameter settings of the algorithm

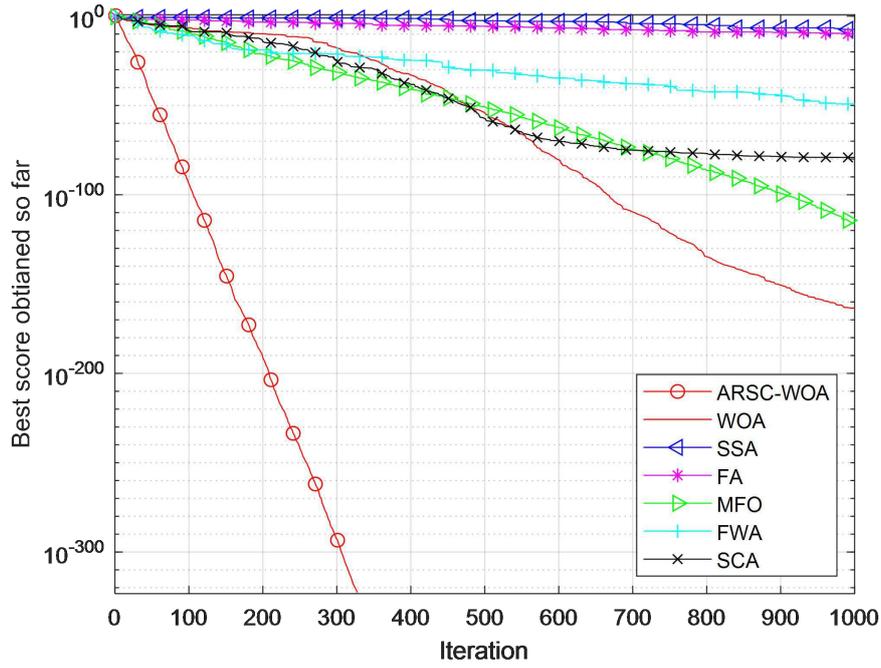
Algorithm	Main parameter settings
-----------	-------------------------

ARSC-WOA Population size n=40; Number of iterations T=1000

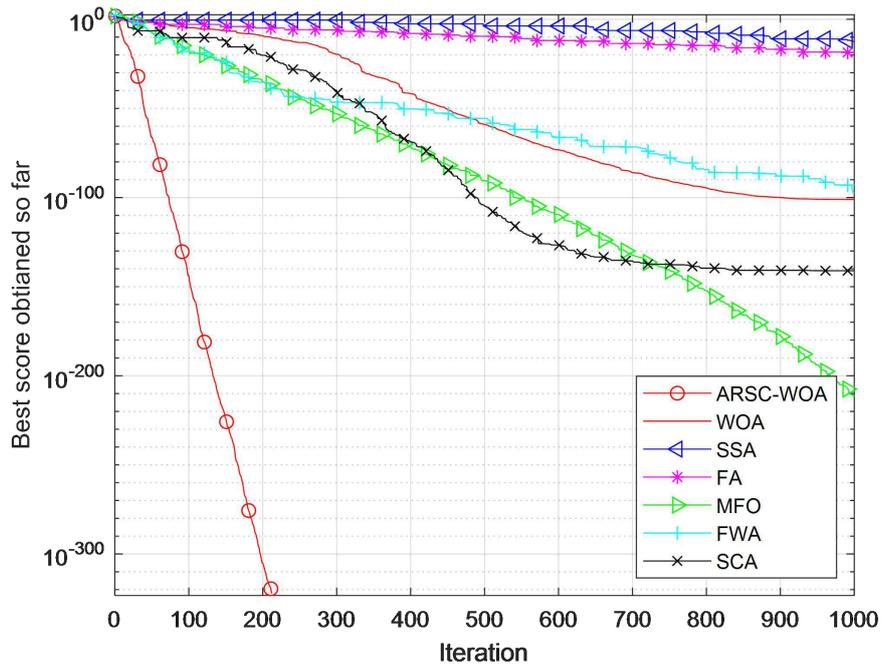
WOA	Population size n=40; Number of iterations T=1000
SSA	Population size n=40; Number of iterations T=1000
FA	Population size n=40; Number of iterations T=1000; $\beta_0 = 1$; $\gamma = 1$
MFO	Population size n=40; Number of iterations T=1000
FWA	Population size n=40; Number of iterations T=1000
SCA	Population size n=40; Number of iterations T=1000



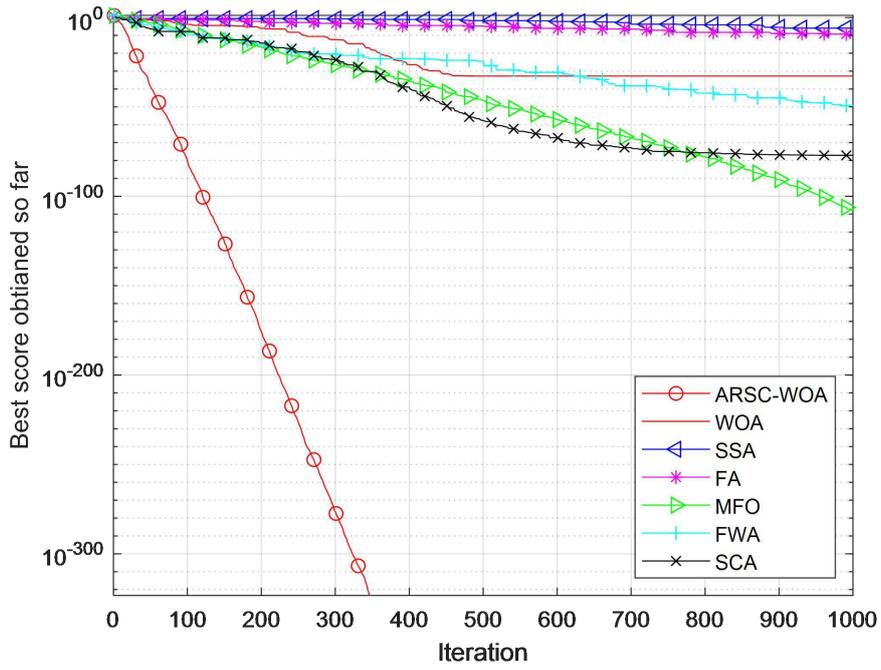
(1) Convergence curves of F_1



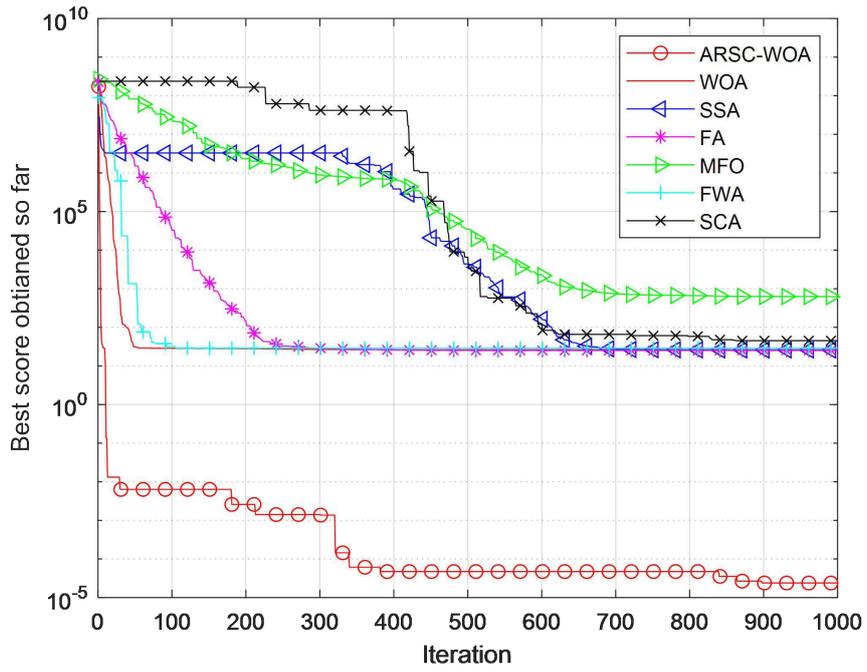
(2) Convergence curves of F_2



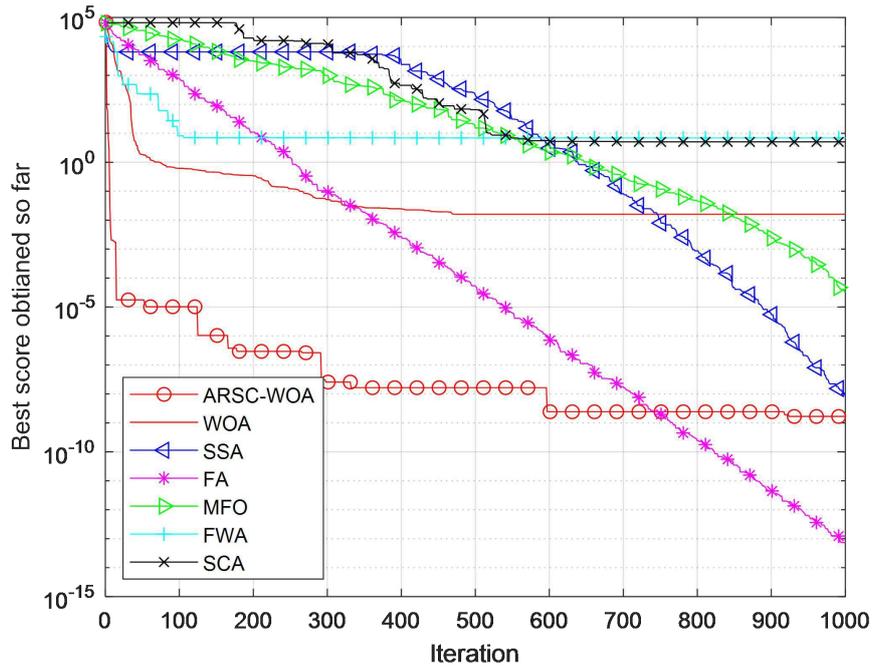
(3) Convergence curves of F_3



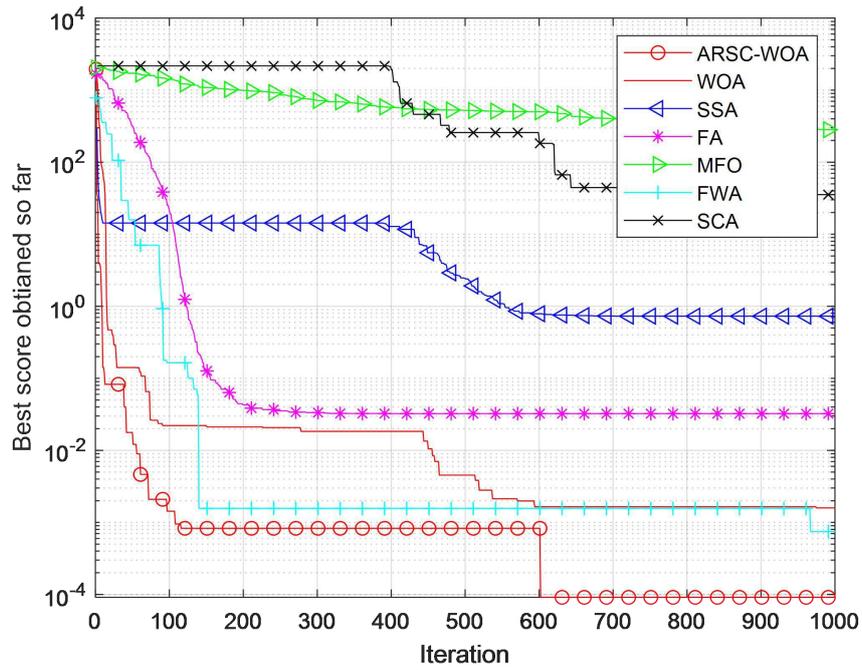
(4) Convergence curves of F_4



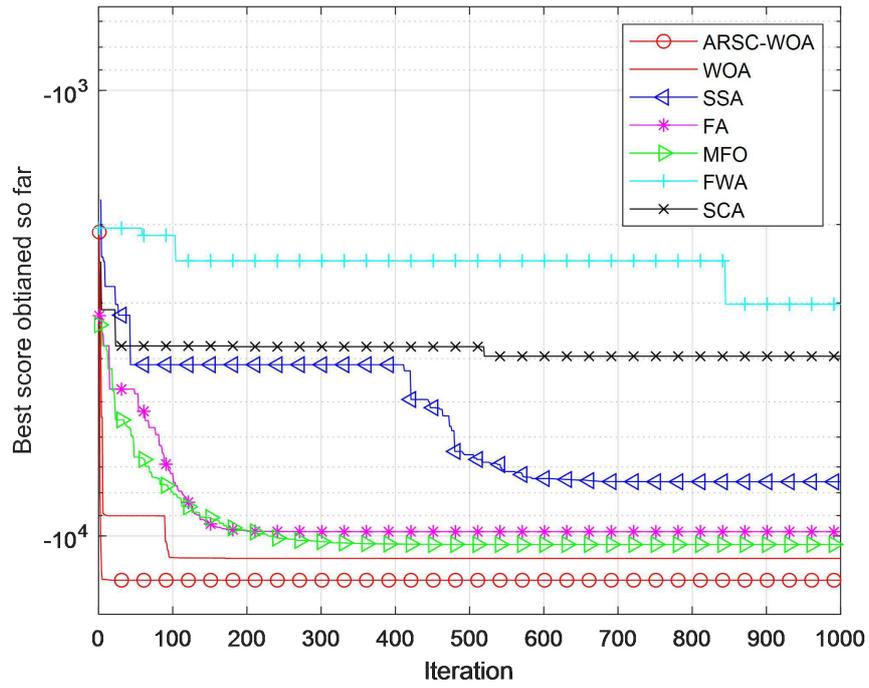
(5) Convergence curves of F_5



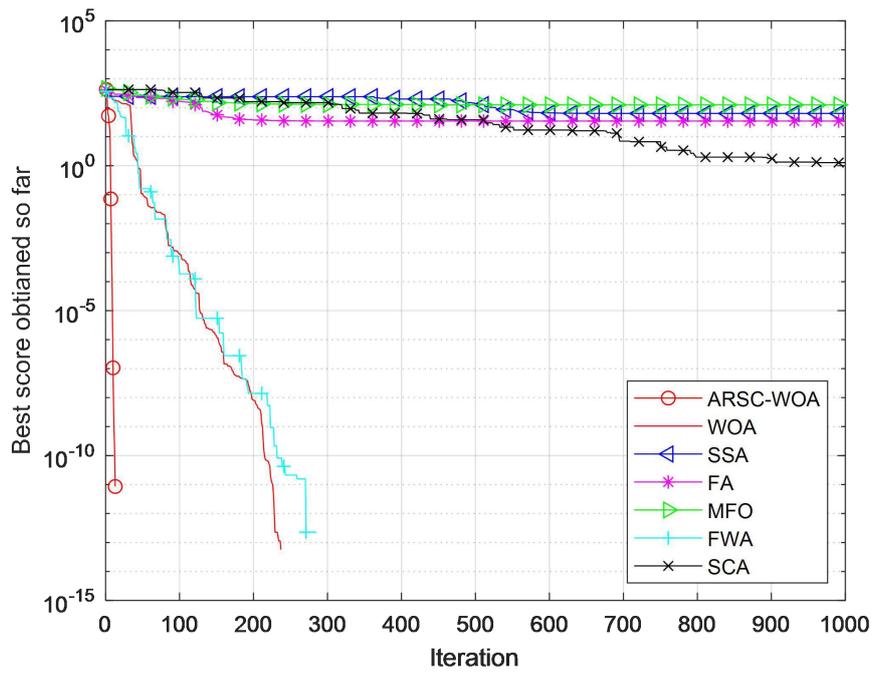
(6) Convergence curves of F_6



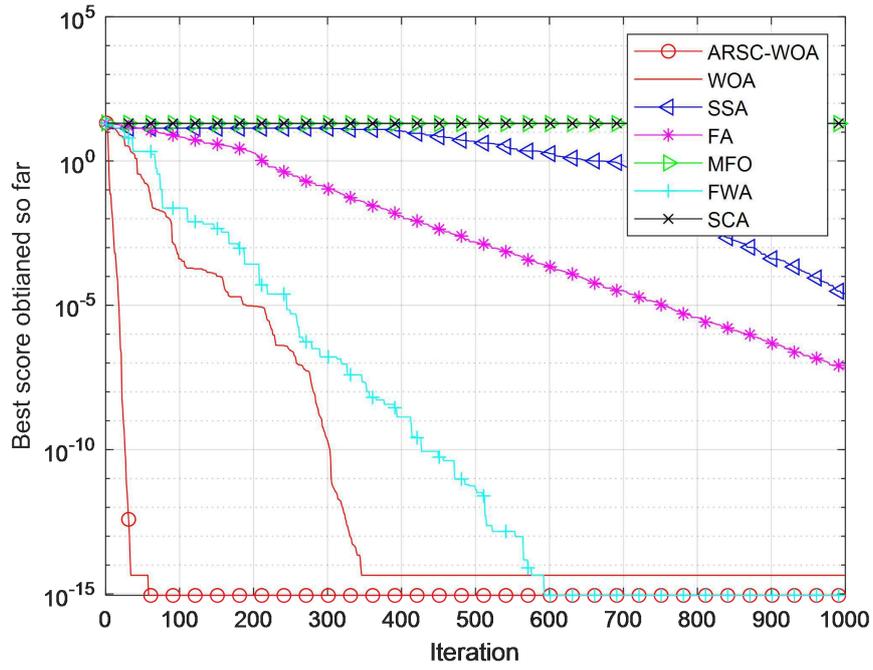
(7) Convergence curves of F_7



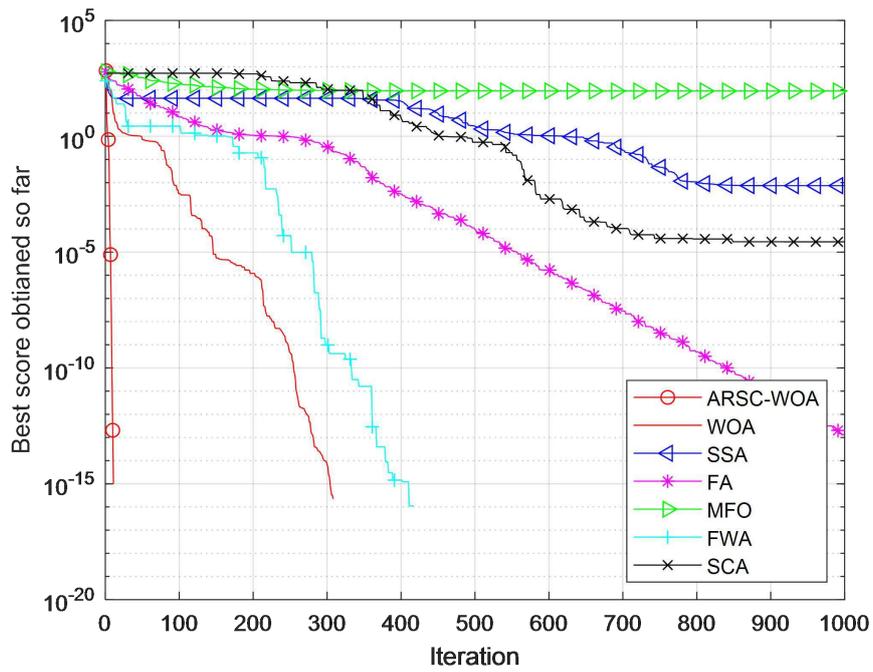
(8) Convergence curves of F_8



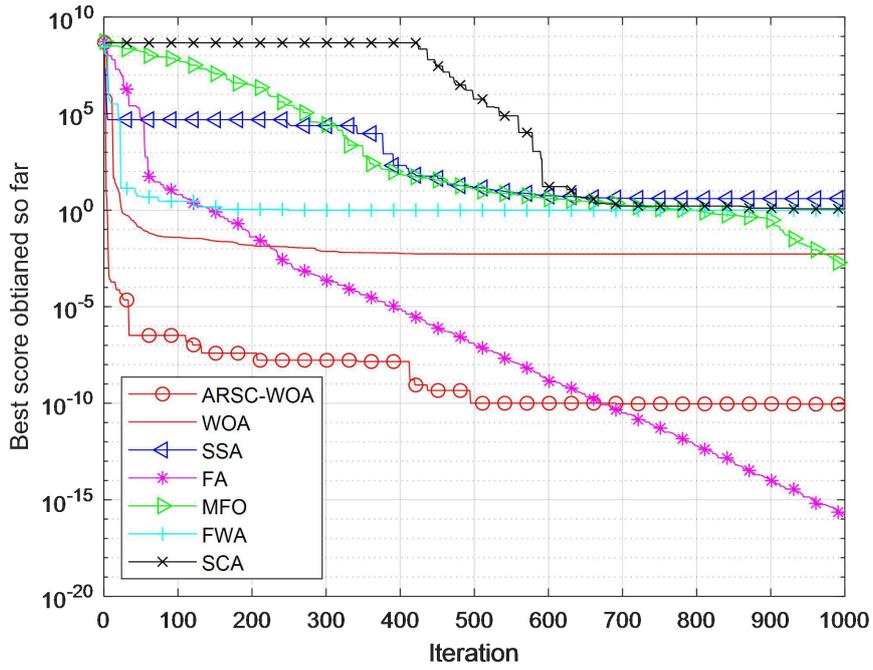
(9) Convergence curves of F_9



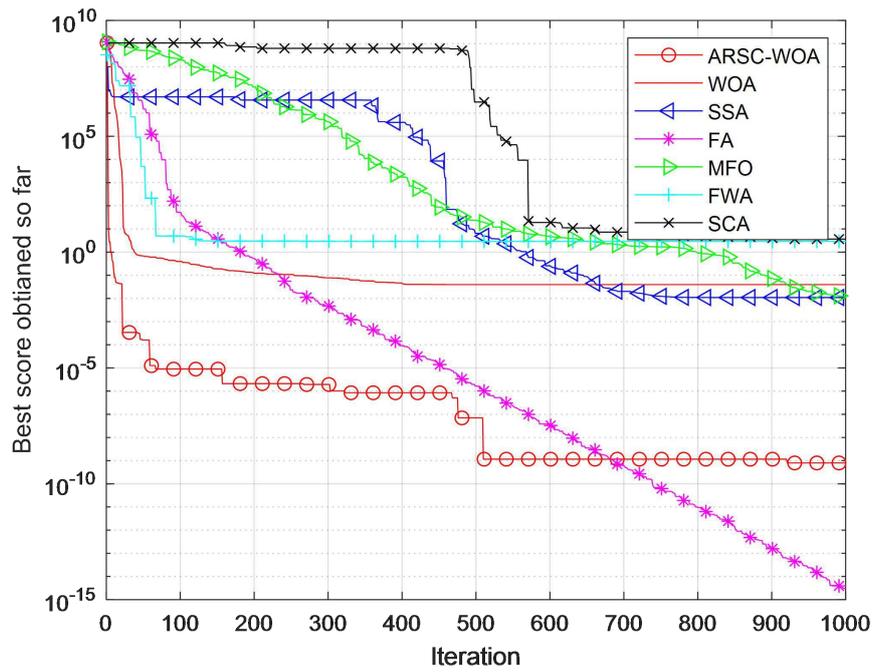
(10) Convergence curves of F_{10}



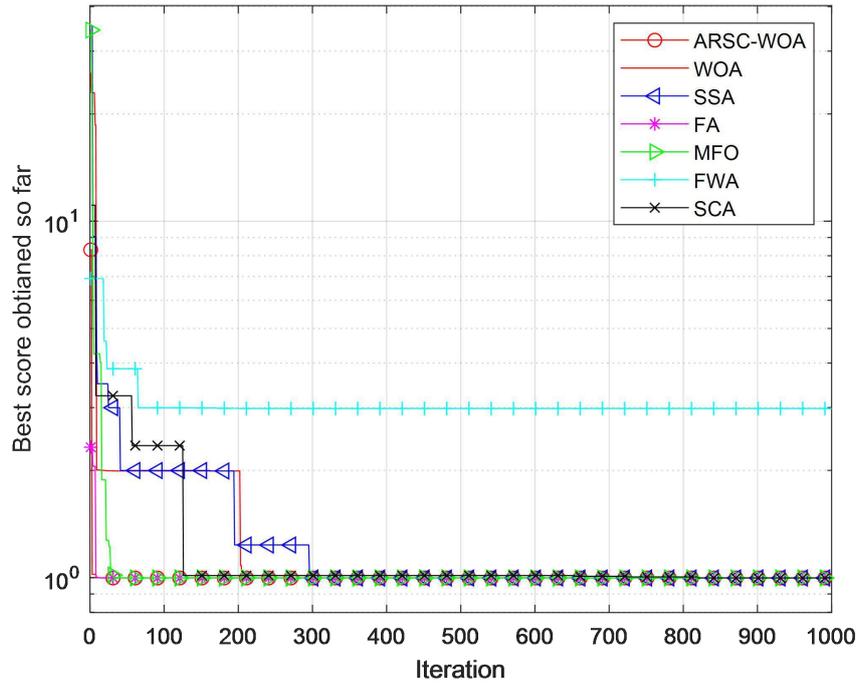
(11) Convergence curves of F_{11}



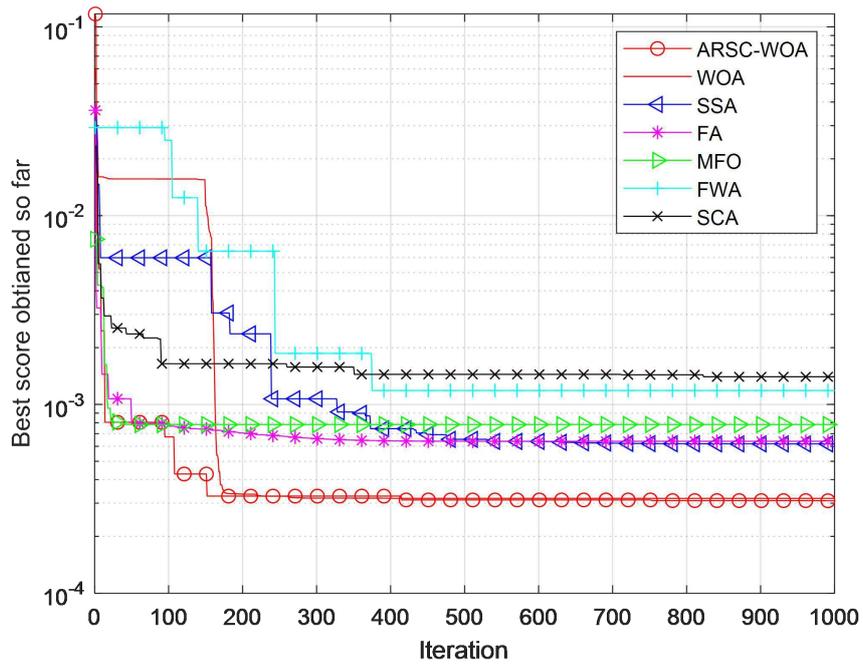
(12) Convergence curves of F_{12}



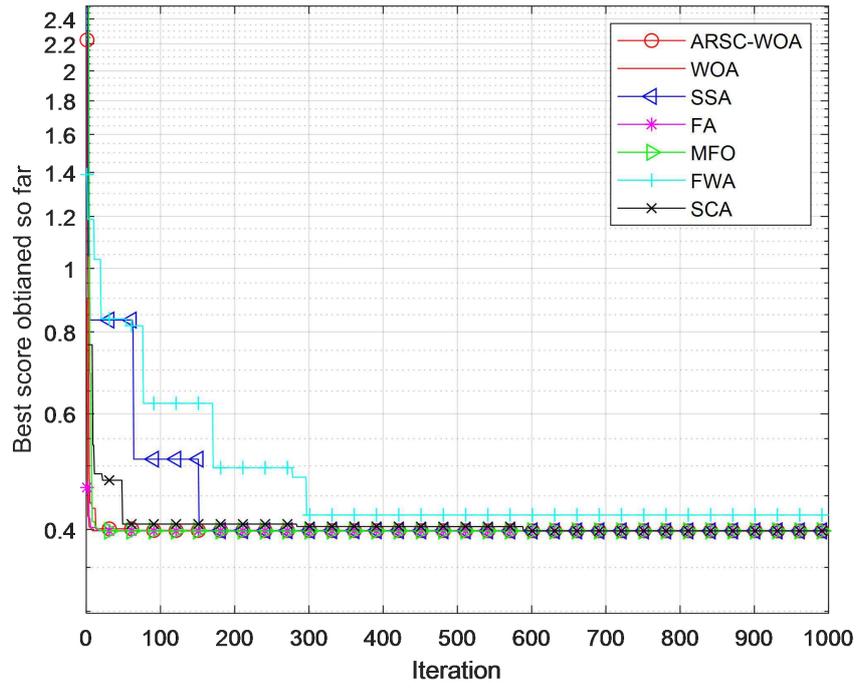
(13) Convergence curves of F_{13}



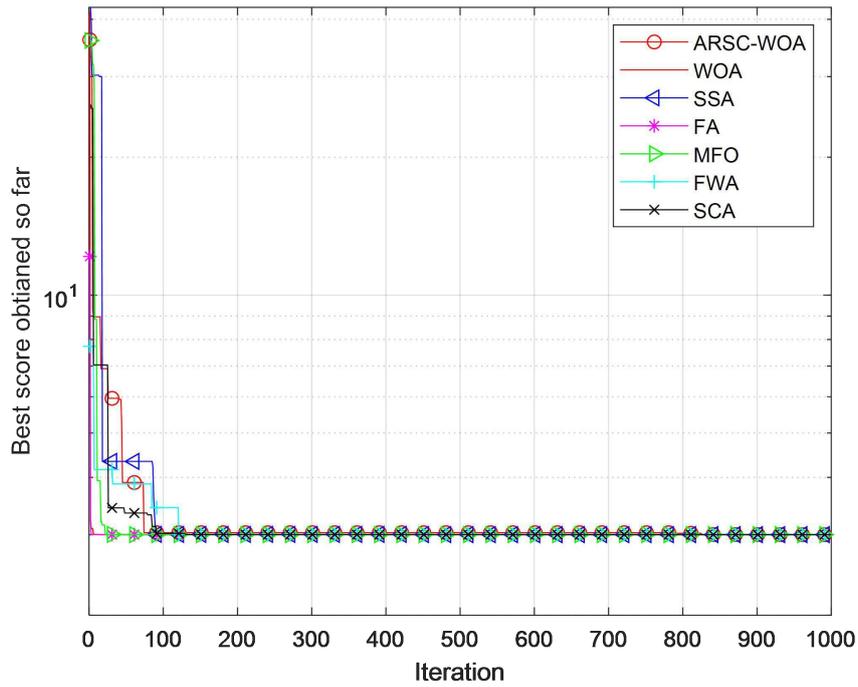
(14) Convergence curves of F_{14}



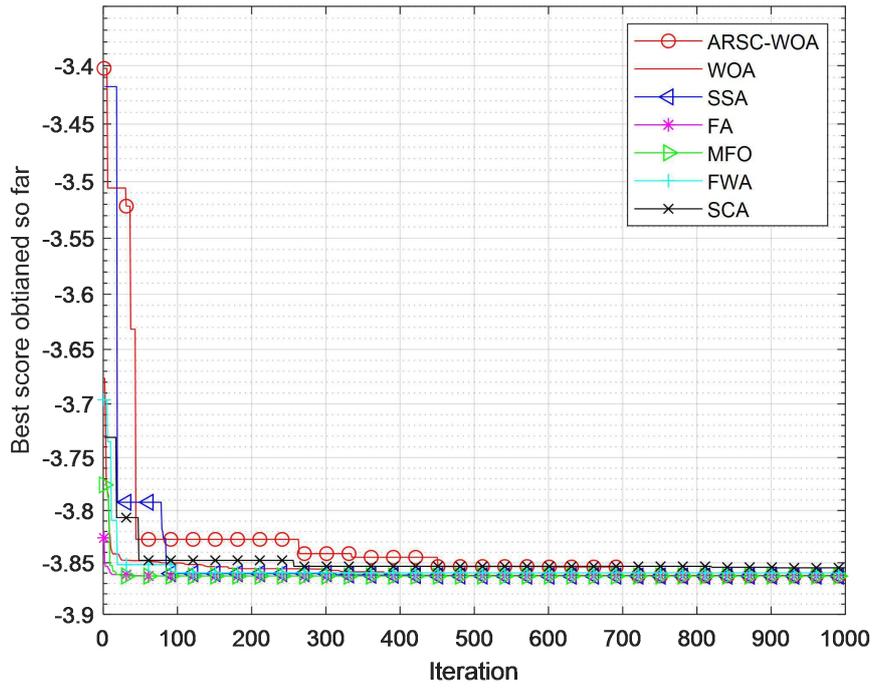
(15) Convergence curves of F_{15}



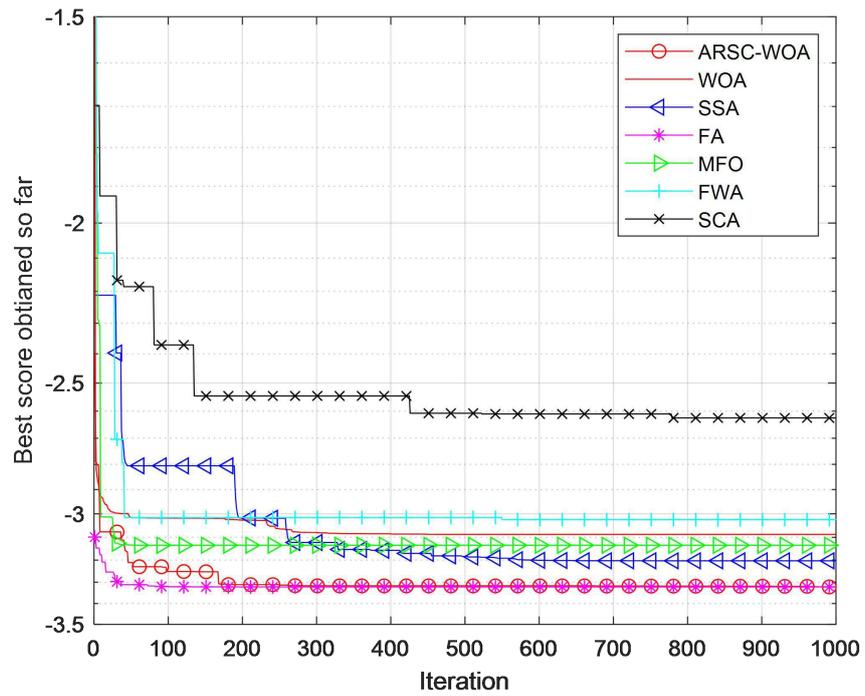
(16) Convergence curves of F_{16}



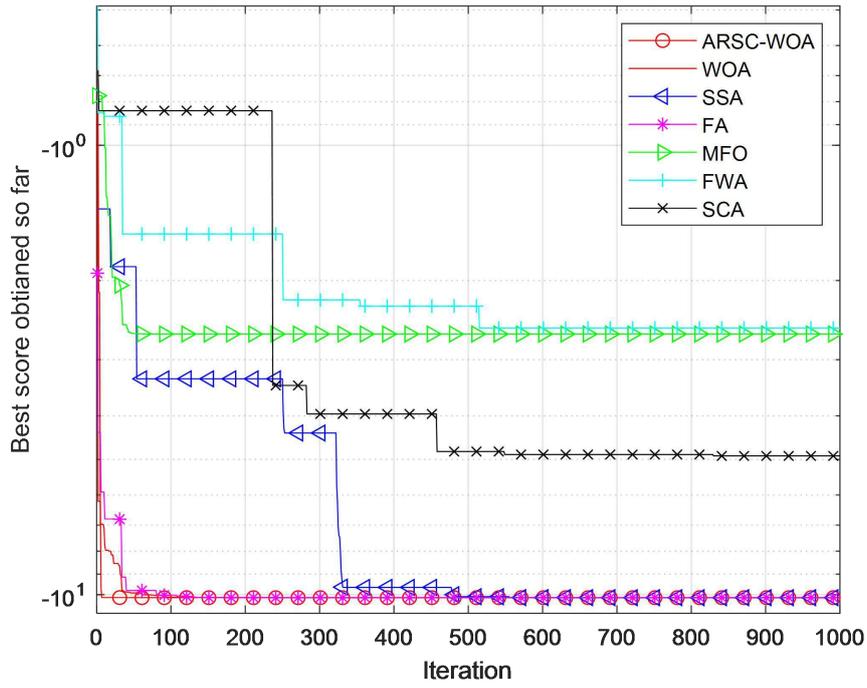
(17) Convergence curves of F_{17}



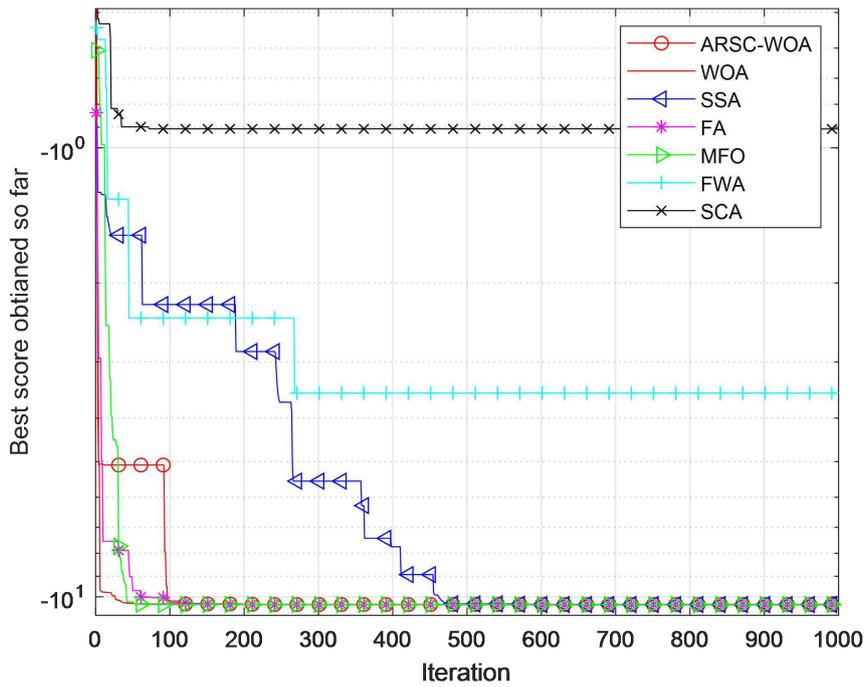
(18) Convergence curves of F_{18}



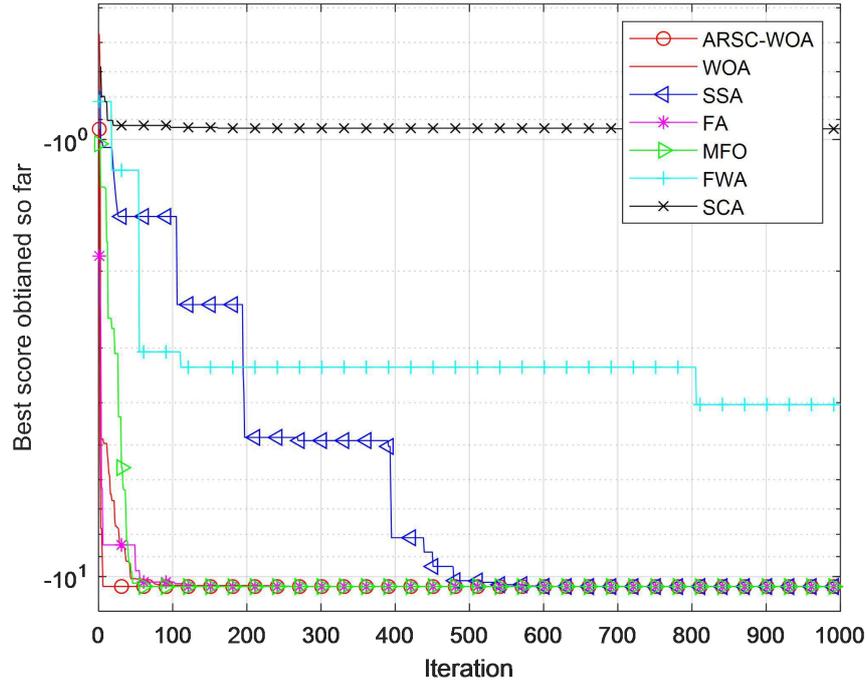
(19) Convergence curves of F_{19}



(20) Convergence curves of F_{20}



(21) Convergence curves of F_{21}



(22) Convergence curves of F_{22}

Fig. 18 Simulation results

Table 5 Simulation performance comparison

Function	ARSC-WOA	WOA	SSA	FA	MFO	FWA	SCA	
F1	Best	0	6.27E-176	5.53E-09	5.40E-14	1.17E-05	4.37E-55	8.42E-06
	Ave	0	8.20E-169	8.82E-09	7.96E-14	4.77E-05	2.46E-41	9.94E-03
	Std	0	0	2.51E-09	1.23E-14	3.25E-05	7.74E-41	1.62E-02
F2	Best	0	3.54E-180	1.15E-08	1.63E-11	1.70E-120	3.77E-68	2.41E-82
	Ave	0	6.83E-163	5.89E-08	6.99E-11	7.79E-115	3.06E-52	1.08E-74
	Std	0	2.22E-162	4.77E-08	3.57E-11	1.55E-114	6.61E-52	2.95E-74
F3	Best	0	1.62E-108	3.75E-15	7.66E-21	1.52E-209	3.15E-151	2.47E-146
	Ave	0	8.57E-96	1.19E-13	4.02E-19	4.84E-196	9.50E-87	3.99E-134
	Std	0	2.25E-95	2.35E-13	5.38E-19	0	3.00E-86	1.08E-133
F4	Best	0	2.87E-40	1.74E-07	2.44E-10	6.37E-114	9.57E-69	6.17E-79
	Ave	0	2.43E-30	4.66E-07	5.81E-10	2.96E-108	9.36E-51	3.00E-72
	Std	0	6.74E-30	3.15E-07	2.50E-10	8.65E-108	2.35E-50	8.97E-72
F5	Best	7.28E-06	2.60E+01	2.36E+01	2.38E+01	8.87E+01	2.90E+01	2.89E+01
	Ave	2.24E+01	2.67E+01	1.28E+02	2.49E+01	3.64E+04	2.90E+01	3.94E+01
	Std	1.18E+01	3.00E-01	1.60E+02	7.76E-01	4.62E+04	8.60E-03	1.84E+01
F6	Best	2.66E-11	3.86E-03	6.42E-09	7.41E-14	6.93E-06	6.5687	3.5375
	Ave	1.71E-09	9.46E-03	1.03E-08	8.30E-14	1.01E+03	1.79E+01	4.7038
	Std	1.90E-09	7.40E-03	2.55E-09	7.15E-15	3.19E+03	1.17E+01	1.2479
F7	Best	2.72E-05	4.23E-04	2.94E-02	2.26E-02	4.37E+01	1.51E-04	1.12E+01
	Ave	2.67E-04	2.14E-03	7.35E-02	2.97E-02	1.76E+02	1.45E+01	4.50E+01
	Std	4.56E-04	2.26E-03	3.33E-02	4.92E-03	1.23E+02	1.53E+01	2.88E+01

F8	Best	-12569.4866	-12569.3488	-8687.6661	-10431.5066	-10056.4573	-3345.4754	-4927.4869
	Ave	-1.26E+04	-1.19E+04	-7.43E+03	-9.94E+03	-8.74E+03	-1.37E+03	-4.04E+03
	Std	1.02E-05	1.05E+03	7.79E+02	4.39E+02	8.38E+02	1.49E+03	3.30E+02
F9	Best	0	0	2.98E+01	2.79E+01	9.45E+01	0	1.37E-04
	Ave	0	0	6.07E+01	3.98E+01	1.42E+02	0	1.32E+01
	Std	0	0	1.94E+01	6.7968	31.002	0	2.28E+01
F10	Best	8.88E-16	8.88E-16	2.19E-05	6.15E-08	7.04E-04	8.88E-16	4.34E-04
	Ave	8.88E-16	3.02E-15	1.5547	6.47E-08	1.37E+01	8.88E-16	1.01E+01
	Std	0	2.48E-15	8.25E-01	2.39E-09	8.9782	0	10.5845
F11	Best	0	0	4.19E-08	1.05E-13	1.92E-05	0	4.40E-04
	Ave	0	2.67E-03	1.65E-02	2.71E-03	1.82E+01	0	2.12E-01
	Std	0	8.43E-03	1.34E-02	4.42E-03	3.83E+01	0	1.80E-01
F12	Best	5.69E-14	3.11E-04	1.0563	1.61E-16	2.18E-04	8.94E-01	3.36E-01
	Ave	1.69E-10	3.66E-03	3.9496	2.06E-16	1.45E-01	1.2753	6.88E-01
	Std	2.79E-10	5.54E-03	2.2658	2.86E-17	1.71E-01	0.23608	3.97E-01
F13	Best	3.44E-11	3.11E-03	5.34E-10	2.33E-15	1.63E-04	9.33E-01	4.95E-01
	Ave	2.50E-09	6.42E-02	5.40E-03	2.98E-15	4.88E-02	1.4229	2.0347
	Std	2.53E-09	6.85E-02	7.55E-03	4.18E-16	6.19E-02	1.77E-01	3.1389
F14	Best	0.998	0.998	0.998	0.998	0.998	1.0294	0.998
	Ave	0.998	2.5698	0.998	0.998	2.2843	1.9222	1.39E+00
	Std	2.34E-14	3.027	1.28E-16	1.48E-16	1.6816	7.98E-01	8.37E-01
F15	Best	3.08E-04	3.08E-04	6.21E-04	3.07E-04	3.82E-04	6.67E-04	5.21E-04
	Ave	5.16E-04	6.09E-04	2.91E-03	5.81E-04	8.40E-04	1.45E-03	7.88E-04
	Std	4.37E-04	3.01E-04	6.14E-03	1.48E-04	3.55E-04	6.71E-04	2.51E-04
F16	Best	0.39789	0.39789	0.39789	0.39789	0.39789	0.40007	0.39813
	Ave	0.39789	0.39789	0.39789	0.39789	0.39789	0.41362	0.39868
	Std	8.46E-07	2.28E-07	3.01E-14	0	0	1.63E-02	3.96E-04
F17	Best	3	3	3	3	3	3.0008	3
	Ave	3	3	3	3	3	3.0097	3
	Std	1.55E-05	8.03E-06	4.70E-14	8.24E-16	4.68E-16	8.72E-03	1.51E-05
F18	Best	-3.8628	-3.8628	-3.8628	-3.8628	-3.8628	-3.8624	-3.8548
	Ave	-3.8627	-3.86	-3.8628	-3.8628	-3.8628	-3.8606	-3.8544
	Std	2.88E-05	3.19E-03	2.37E-14	4.44E-16	9.36E-16	9.90E-04	4.07E-04
F19	Best	-3.3194	-3.3219	-3.322	-3.322	-3.322	-3.1758	-3.1981
	Ave	-2.9496	-3.2944	-3.2377	-3.2863	-3.2551	-3.1273	-2.9925
	Std	3.57E-01	5.76E-02	5.82E-02	5.74E-02	7.30E-02	3.65E-02	2.74E-01
F20	Best	-10.1532	-10.1531	-10.1532	-10.1532	-10.1532	-5.1001	-5.8208
	Ave	-10.1532	-9.6427	-10.1532	-10.1532	-7.1382	-3.5819	-2.2504
	Std	1.13E-06	1.6119	3.23E-11	1.87E-15	3.3011	7.46E-01	2.0658
F21	Best	-10.4028	-10.4028	-10.4029	-10.4029	-10.4029	-6.9583	-6.328
	Ave	-10.4028	-9.2029	-10.4029	-10.4029	-8.8727	-3.7699	-4.6722
	Std	4.04E-05	2.5486	2.48E-11	2.37E-15	3.23E+00	1.2861	8.71E-01

	Best	-10.5363	-10.5364	-10.5364	-10.5364	-10.5364	-4.1393	-6.6962
F22	Ave	-10.5363	-8.7809	-8.4608	-10.5364	-7.205	-3.3155	-4.6614
	Std	2.58E-05	2.8411	3.402	1.87E-15	3.61E+00	0.60007	1.4862

The convergence curves show that ARSC-WOA has obvious advantages in most functions. For unimodal functions F1-F7, ARSC-WOA is inferior to FA algorithm only on function F6. For functions F1-F5 and F7, ARSC-WOA is significantly better than other algorithms. It shows that the improved WOA has strong convergence speed and local exploitation ability. For multimodal functions F8-F13, ARSC-WOA is inferior to FA algorithm only on functions F12 and F13. ARSC-WOA has shown strong competitiveness in other functions. It is worth mentioning that for functions F9 and F11, the improved WOA found the theoretical optimal value and exited within 100 iterations. The FWA and WOA found the theoretical optimal value within 500 iterations. However, other algorithms have not shown obvious global convergence and local optimal stagnation has occurred. The convergence curves of the multimodal function shows that ARSC-WOA has a good global exploration ability, and the searching agent can develop towards the optimal value area. The fixed-dimensional multimodal functions F14-F22 are mainly used to evaluate the optimization accuracy of the algorithm, so the difference between the convergence curves of the algorithm is not obvious. The statistical results of the optimization accuracy have listed in Table 5. The statistical results in Table 5 can test the optimal accuracy, average accuracy, and robustness. In terms of optimal accuracy, ARSC-WOA found a theoretical optimal value on 50% functions. WOA found a theoretical optimal value on 18% of the functions. The ratio of SSA, FA and FWA to find the theoretical optimal value is 9%. The ratio of MFO and SCA to find the theoretical optimal value is 4%. In terms of average accuracy and robustness, ARSC-WOA is superior to other algorithms on 64% of functions. The statistical results show that the improved WOA has the best optimization accuracy in this experiment. The strong robustness makes the algorithm less susceptible to randomness.

In order to show the advantages of the improved WOA more intuitively, this paper uses Wilcoxon rank sum test [48] to further analyze the statistical results. Wilcoxon rank sum test is also called order sum test, and it is a non-parametric test. Wilcoxon rank sum test does not depend on the specific form of the overall distribution, it does not consider the distribution of the research object and whether the distribution is known, so it is more practical. Wilcoxon rank sum test assesses whether the difference between the two samples is significant and is recorded as the p-value. If the p-value is less than 0.05 and is close to 0, it can be considered that there is a significant difference between two samples. If the p-value is greater than 0.05, it is considered that there is no significant difference between two samples. If the p-value is NaN, there is no difference between two samples. Table 6 summarizes the p-value results of ARSC-WOA compared with other algorithms. The statistical results show that the improved WOA has obvious advantages, which further validates the performance of the algorithm.

In summary, ARSC-WOA showed the best performance in this experiment. It is not only better than WOA, but also better than SCA, which shows that the improved strategies proposed in this paper are effective. First, the Archimedes spiral searching strategy improves the population diversity of search agents. Secondly, the introduction of improved sine cosine operator enables the search agent to adaptively switch search methods, and the exploration and exploitation capabilities are further improved.

Table 6 P-value of Wilcoxon test

Function	ARSC-WOA	ARSC-WOA	ARSC-WOA	ARSC-WOA	ARSC-WOA	ARSC-WOA
	vs WOA	vs SSA	vs FA	vs MFO	vs FWA	vs SCA
F1	6.39E-05	6.39E-05	6.39E-05	6.39E-05	6.39E-05	6.39E-05
F2	6.39E-05	6.39E-05	6.39E-05	6.39E-05	6.39E-05	6.39E-05
F3	6.39E-05	6.39E-05	6.39E-05	6.39E-05	6.39E-05	6.39E-05
F4	6.39E-05	6.39E-05	6.39E-05	6.39E-05	6.39E-05	6.39E-05
F5	2.57E-02	2.11E-02	2.57E-02	1.83E-04	1.83E-04	4.40E-04
F6	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
F7	1.73E-02	1.83E-04	1.83E-04	1.83E-04	5.21E-01	1.83E-04
F8	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
F9	NaN	6.39E-05	6.39E-05	6.39E-05	NaN	6.39E-05
F10	1.80E-03	6.39E-05	6.39E-05	6.39E-05	NaN	6.39E-05
F11	3.68E-01	6.39E-05	6.39E-05	6.39E-05	NaN	6.39E-05
F12	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
F13	1.83E-04	3.45E-01	1.83E-04	1.83E-04	1.83E-04	1.83E-04
F14	3.30E-04	6.39E-05	6.39E-05	1.70E-03	1.83E-04	1.83E-04
F15	3.08E-01	8.90E-02	6.78E-01	6.39E-02	1.13E-02	8.90E-02
F16	1.13E-02	1.74E-04	6.39E-05	6.39E-05	1.83E-04	1.83E-04
F17	7.57E-02	1.83E-04	6.39E-05	6.39E-05	4.40E-04	1.86E-01
F18	2.20E-03	1.71E-04	6.39E-05	6.39E-05	1.83E-04	4.40E-04
F19	7.30E-03	2.57E-02	1.40E-03	5.40E-03	5.71E-01	0.8501
F20	1.83E-04	2.80E-03	6.39E-05	1.00E+00	1.83E-04	1.83E-04
F21	2.20E-03	2.80E-03	6.39E-05	2.12E-02	1.83E-04	1.83E-04
F22	1.70E-03	0.1405	6.39E-05	1.83E-04	1.83E-04	1.83E-04

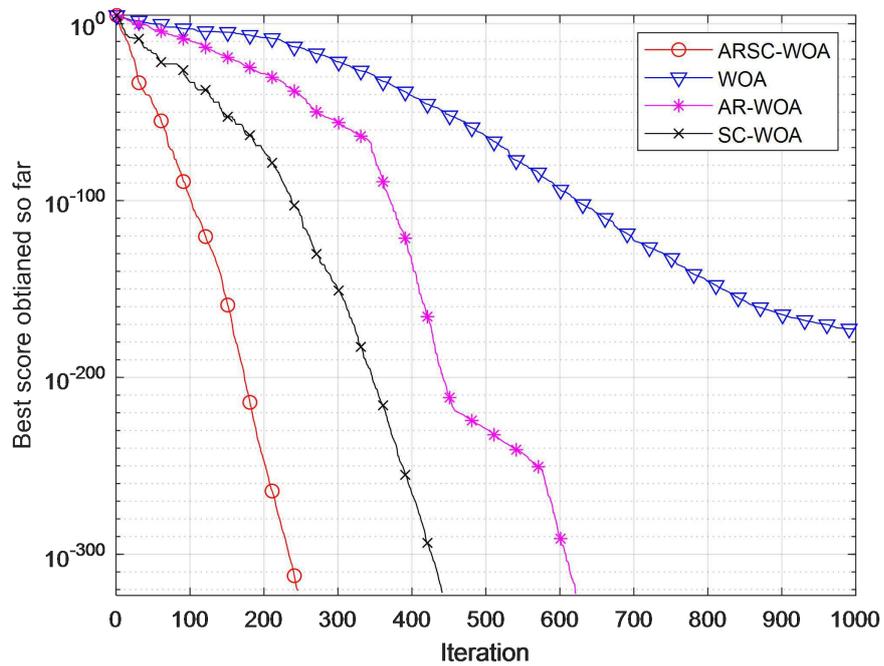
4.5 Effectiveness Analysis of Improved Strategies

The proposed ARSC-WOA introduces the Archimedes spiral search strategy and improved sine and cosine operator. Comparative experiments with other algorithms have verified the performance of the proposed algorithm. However, the rationality and effectiveness of the two improvement strategies need to be further verified. In this paper, WOA based on the Archimedes spiral search strategy (AR-WOA), WOA based on sine cosine operator (SC-WOA), Whale Optimization Algorithm (WOA), and the improved WOA (ARSC-WOA) are selected for carrying out the comparison simulation experiments.

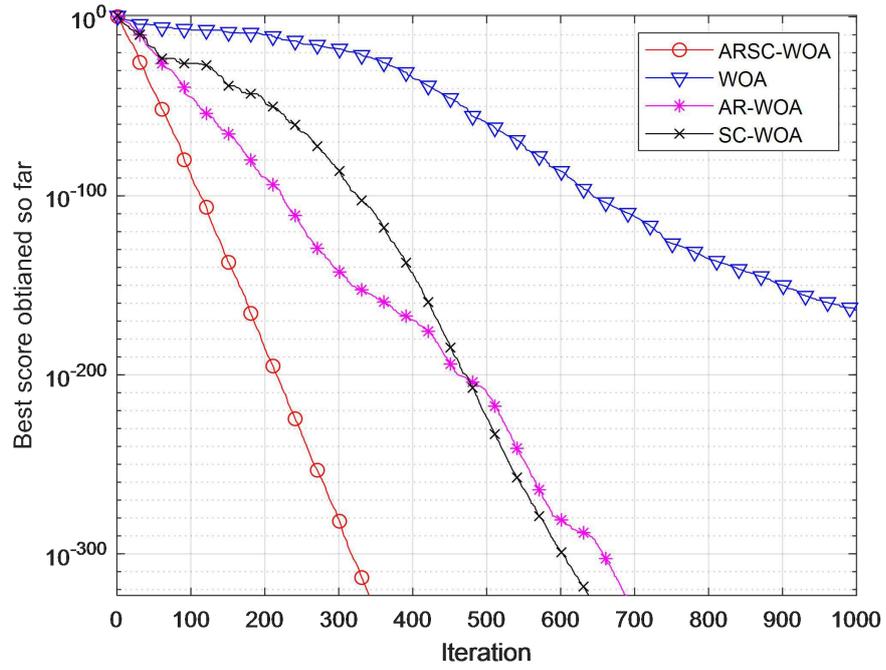
Fig. 19 shows the convergence curves of these algorithms. The statistical results are listed in Table 7. It can be seen from the convergence curves that ARSC-WOA has the best convergence speed and optimization accuracy for both unimodal and multimodal functions. Statistical results show that ARSC-WOA finds the theoretical optimal value on 50% of the functions. AR-WOA found the theoretical optimal value of the 23% functions. SC-WOA found the theoretical optimal value on 36% of the functions. ARSC-WOA also excels in average accuracy and robustness. Table 8 shows the p-value results of the Wilcoxon rank sum test. It can be seen from the statistical results that

ARSC-WOA is significantly different from other algorithms, indicating that the improved strategy can effectively improve the performance of WOA. Among them, the difference between WOA and ARSC-WOA is the most obvious. The second obvious difference is AR-WOA and ARSC-WOA. The difference between SC-WOA and ARSC-WOA is not particularly obvious. The statistical results show that improved sine and cosine operator is the most effective in improving the performance of the algorithm.

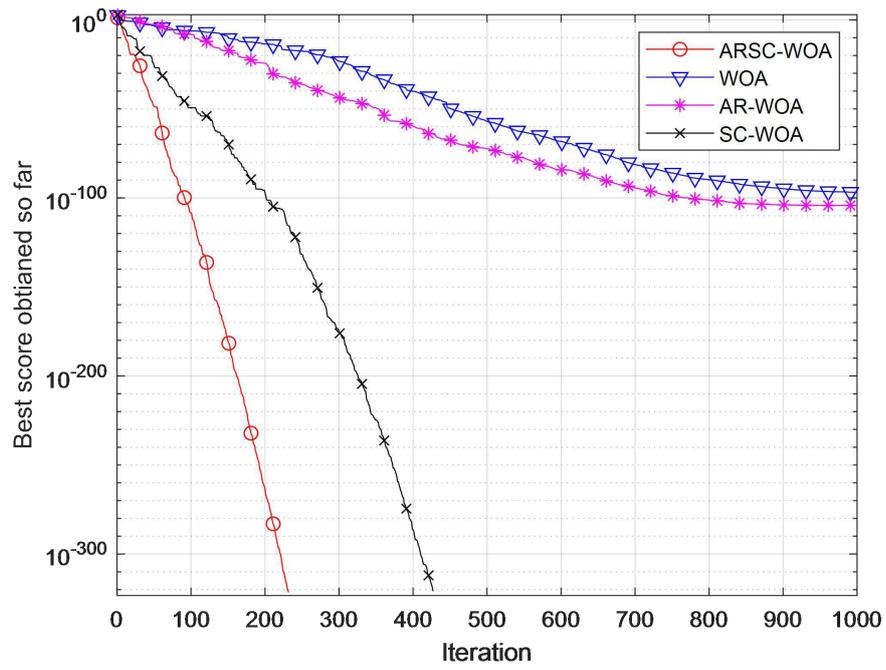
In summary, the experimental results show that AR-WOA and SC-WOA are superior to WOA in most functions, which proves that two improved strategies proposed in this paper are reasonable. In addition, the experimental results show that ARSC-WOA is superior to AR-WOA and SC-WOA in most functions, indicating that multiple improved strategies are more effective than single improved strategy.



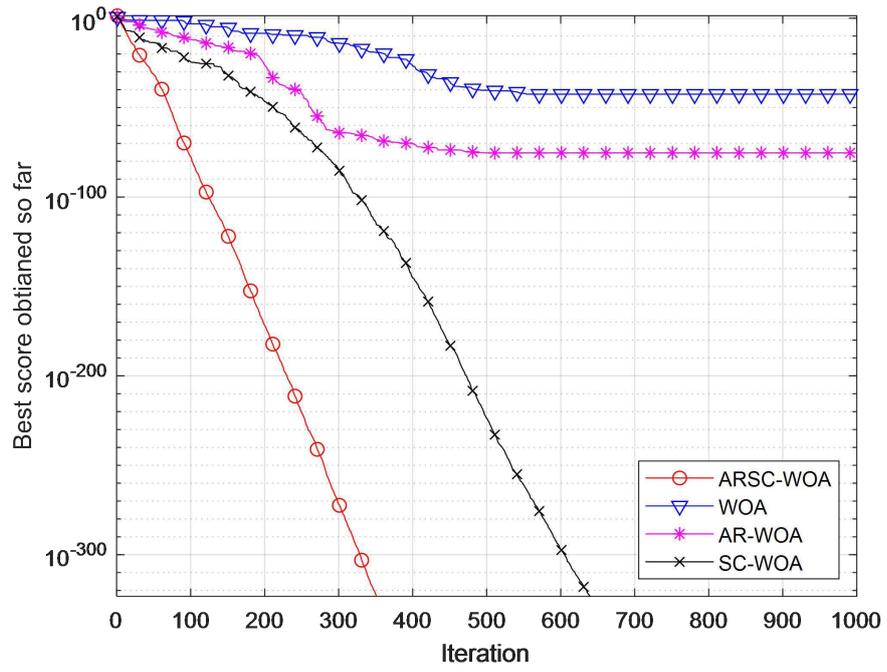
(1) Convergence curves of F_1



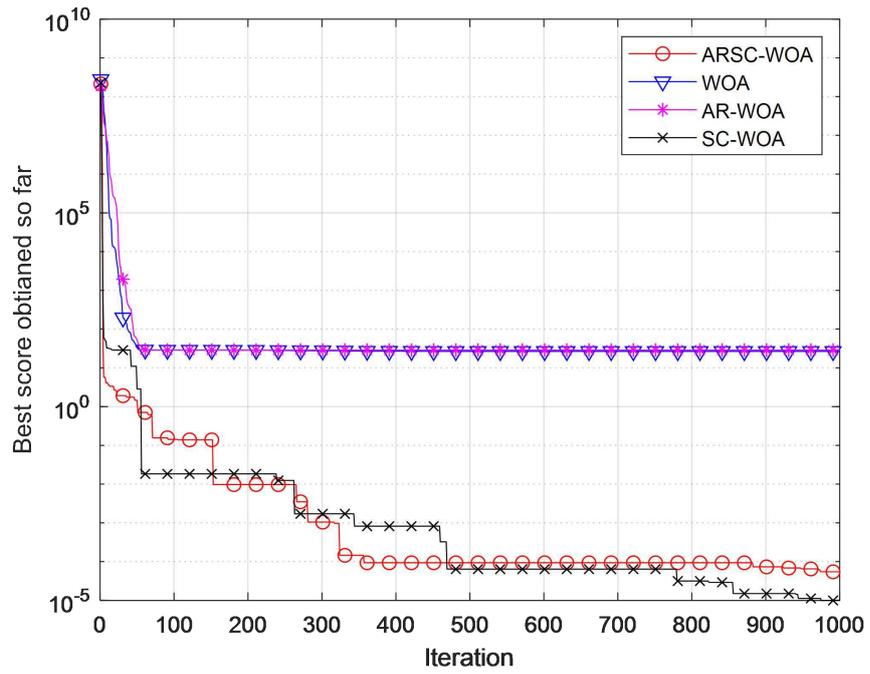
(2) Convergence curves of F_2



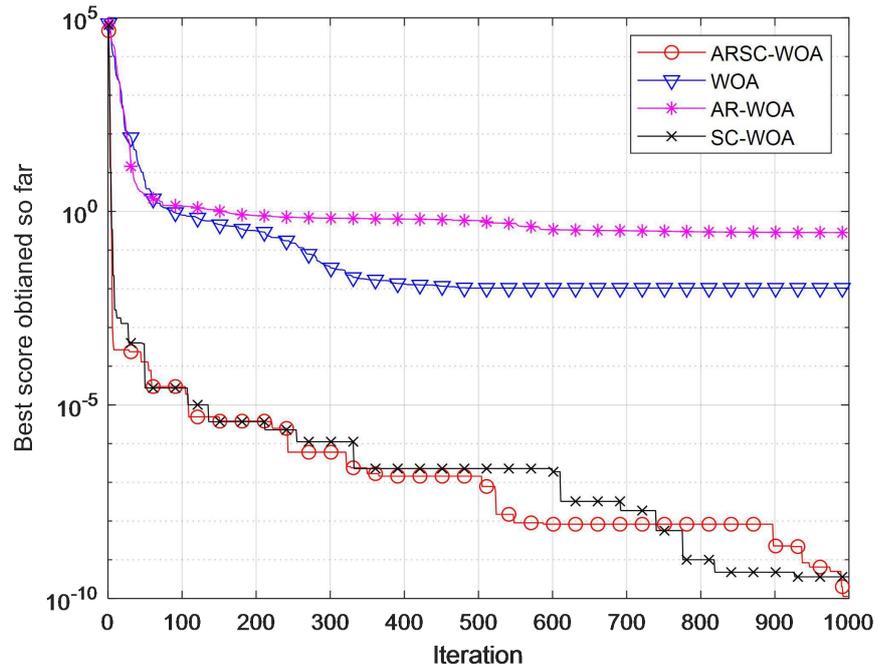
(3) Convergence curves of F_3



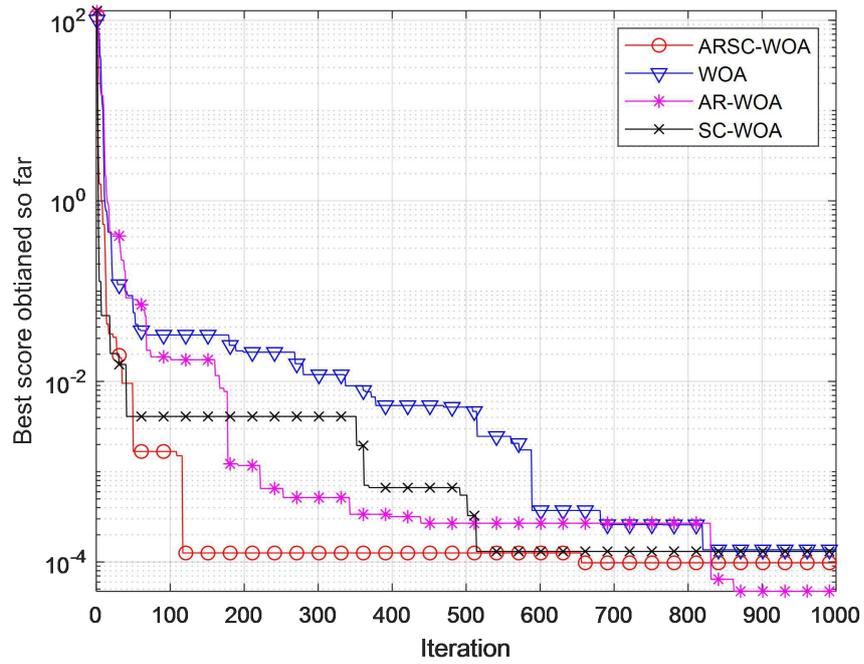
(4) Convergence curves of F_4



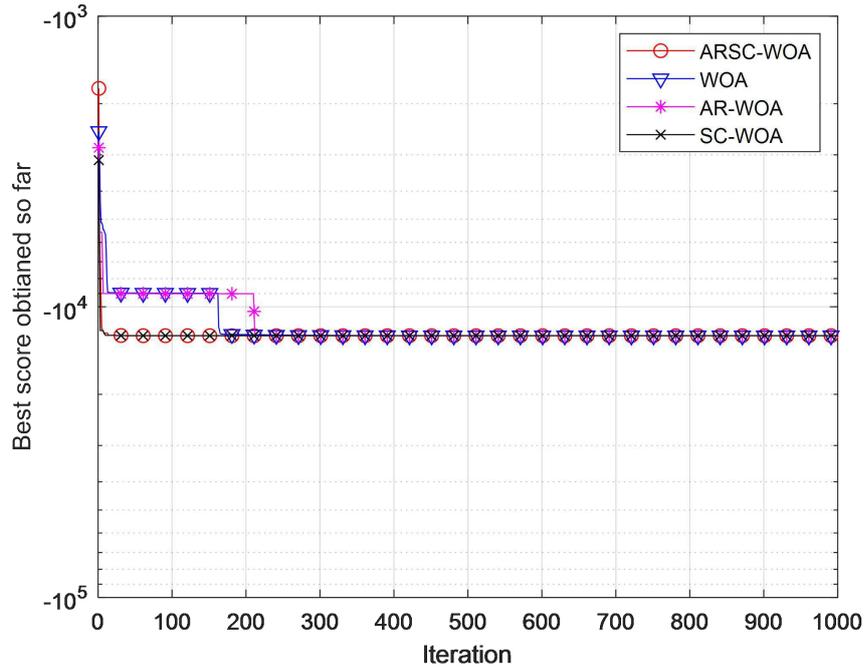
(5) Convergence curves of F_5



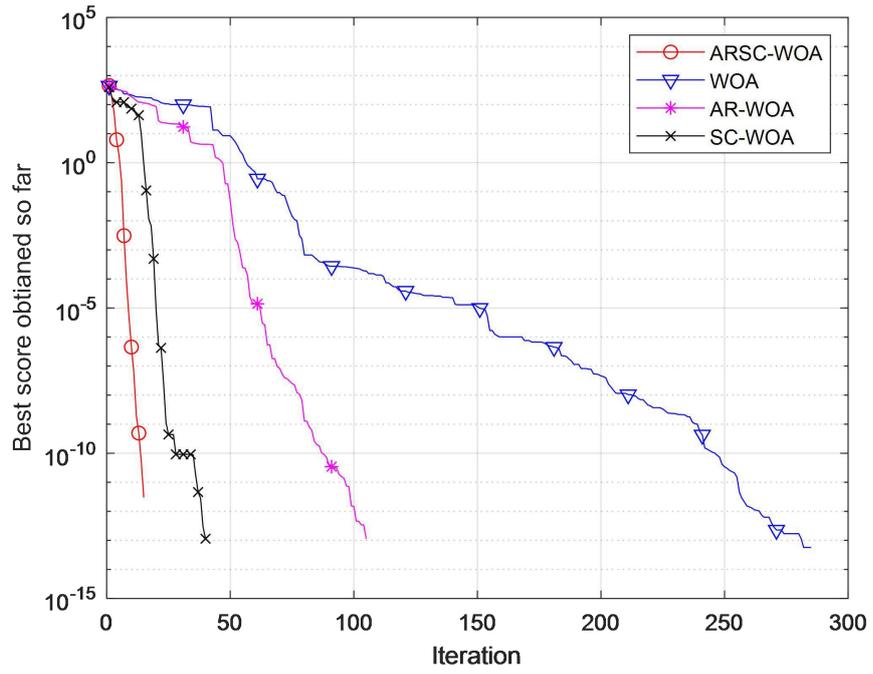
(6) Convergence curves of F_6



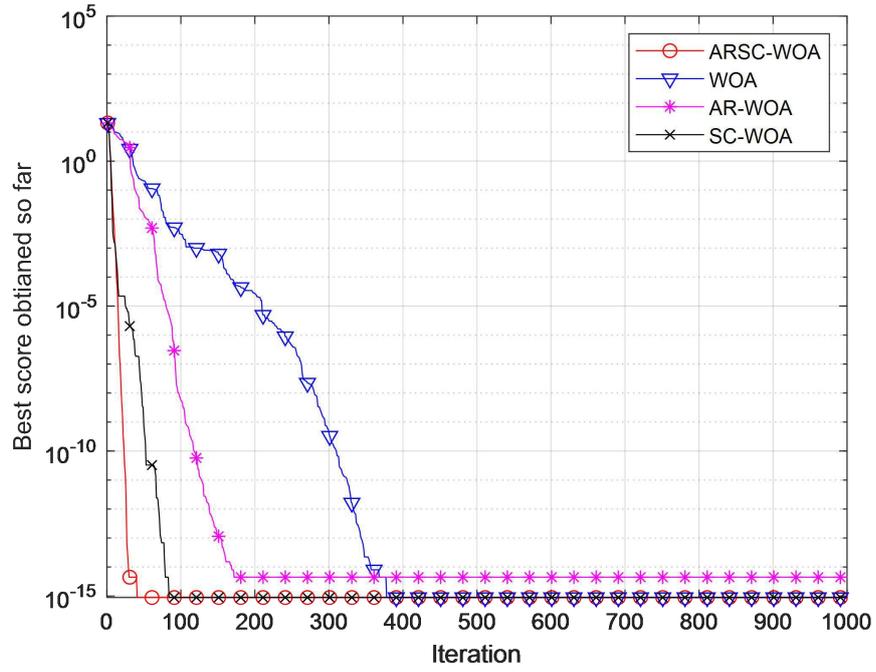
(7) Convergence curves of F_7



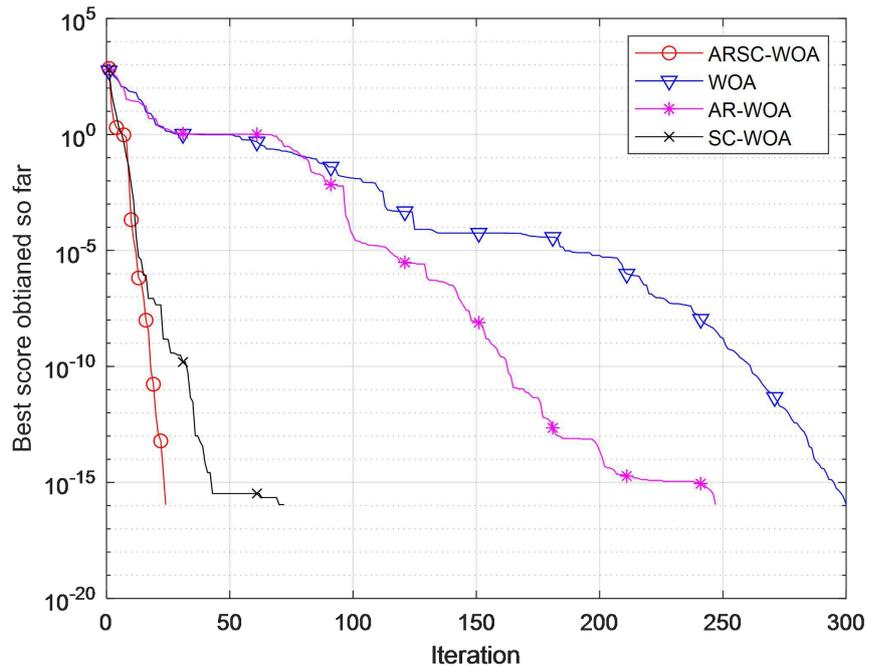
(8) Convergence curves of F_8



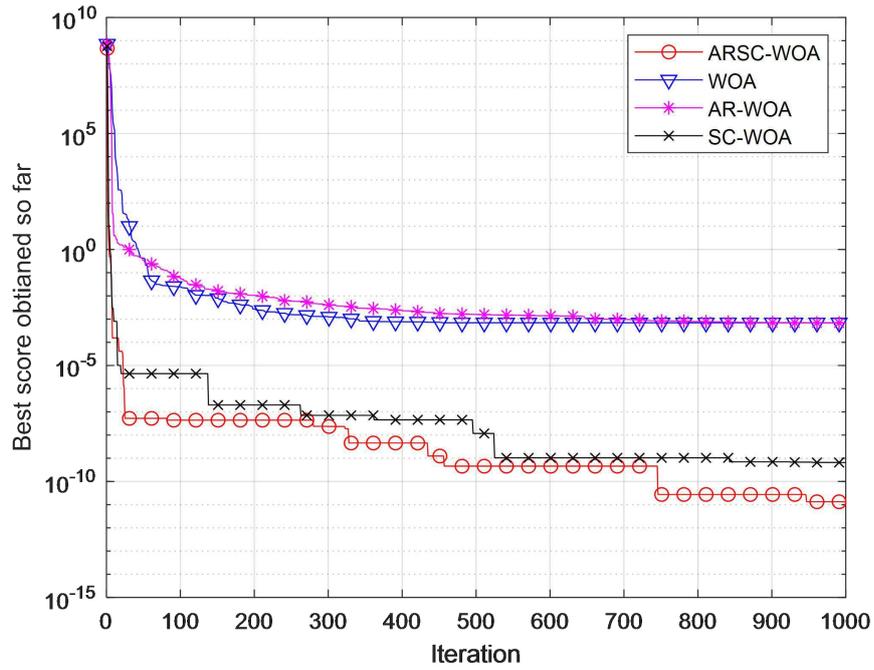
(9) Convergence curves of F_9



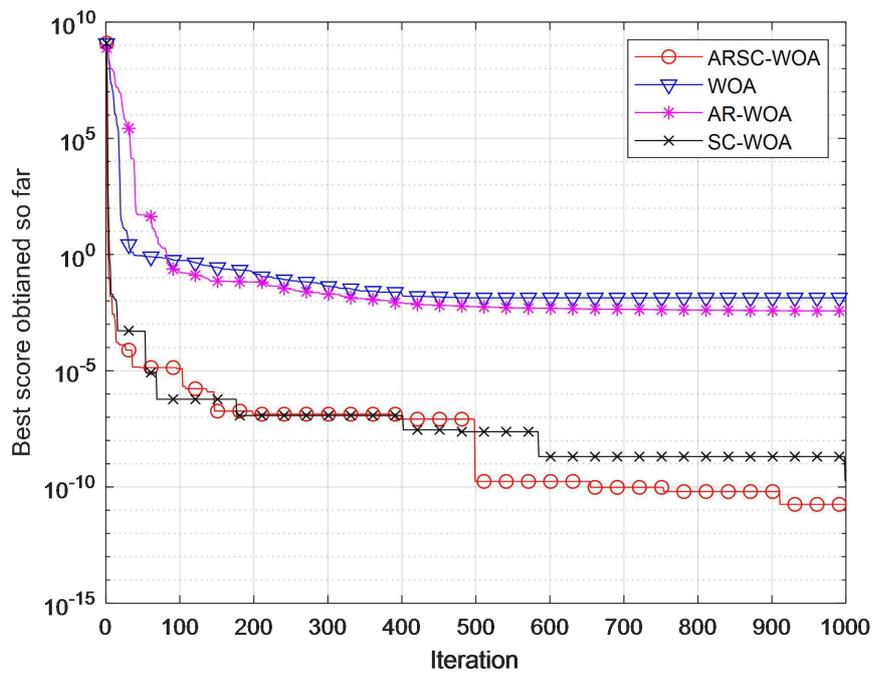
(10) Convergence curves of F_{10}



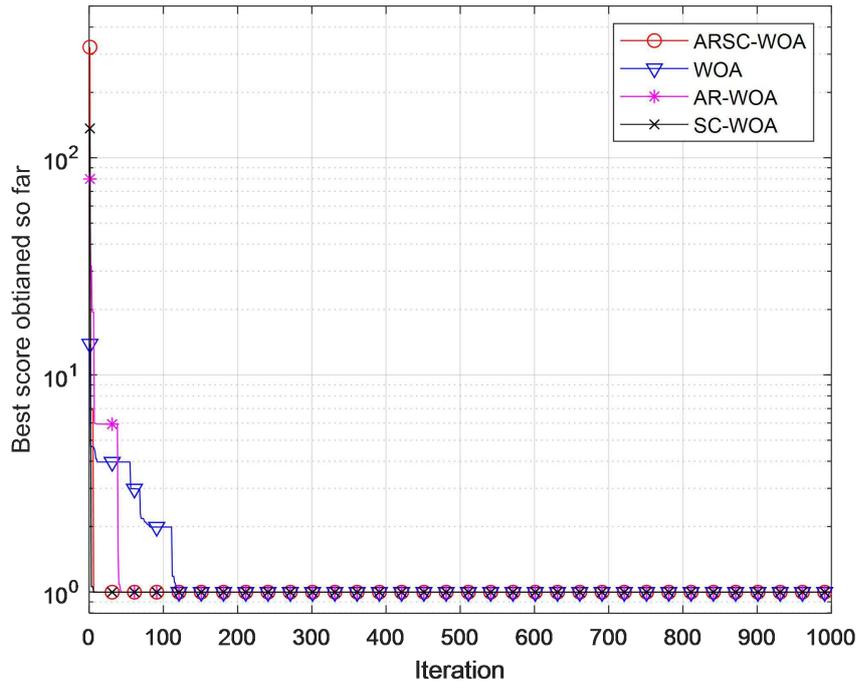
(11) Convergence curves of F_{11}



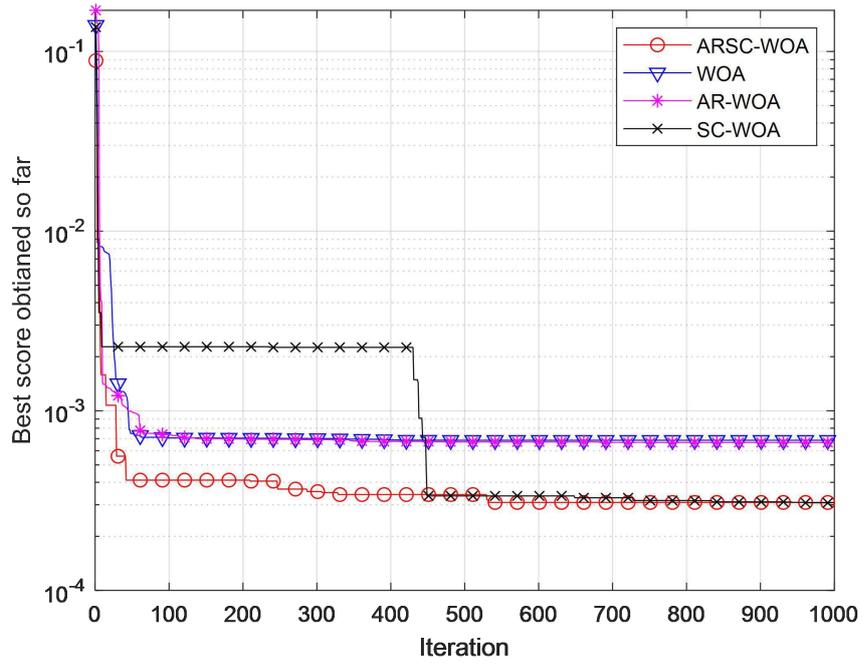
(12) Convergence curves of F_{12}



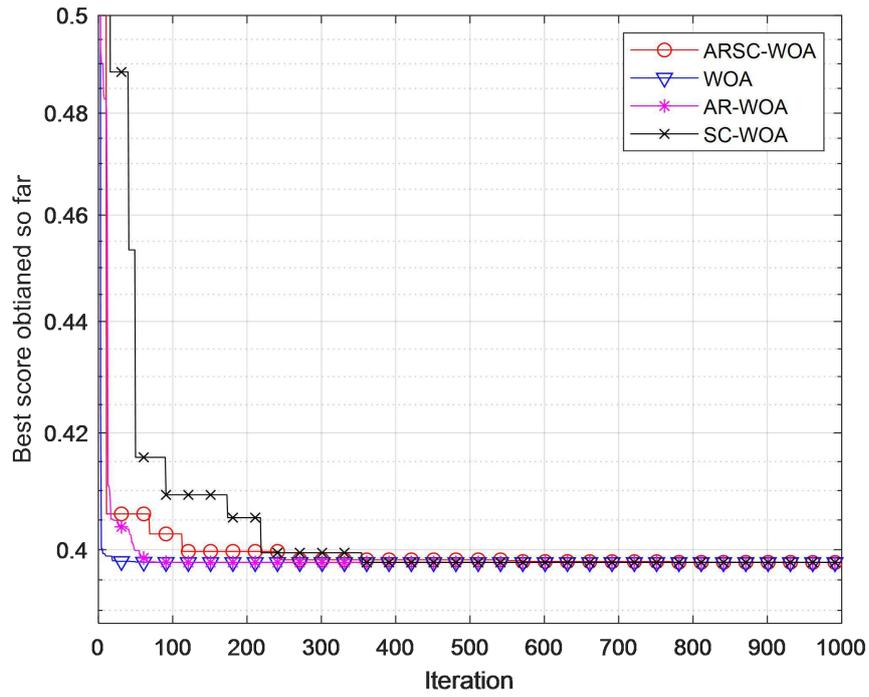
(13) Convergence curves of F_{13}



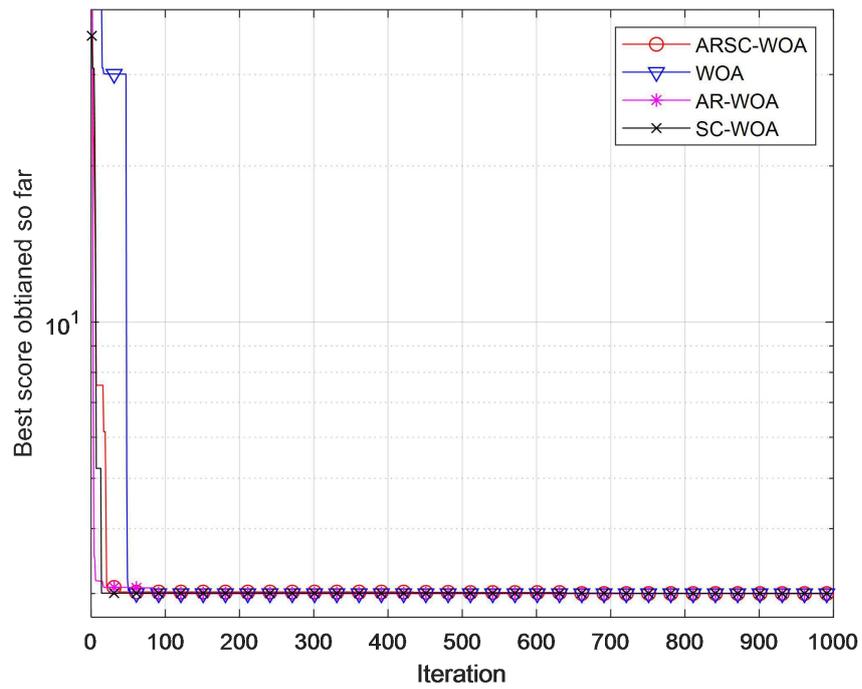
(14) Convergence curves of F_{14}



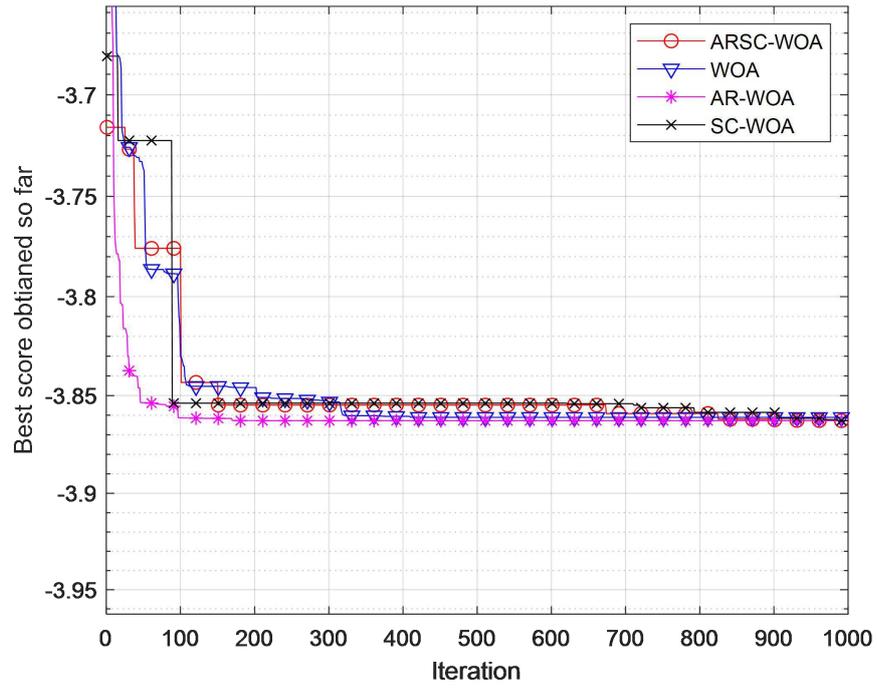
(15) Convergence curves of F_{15}



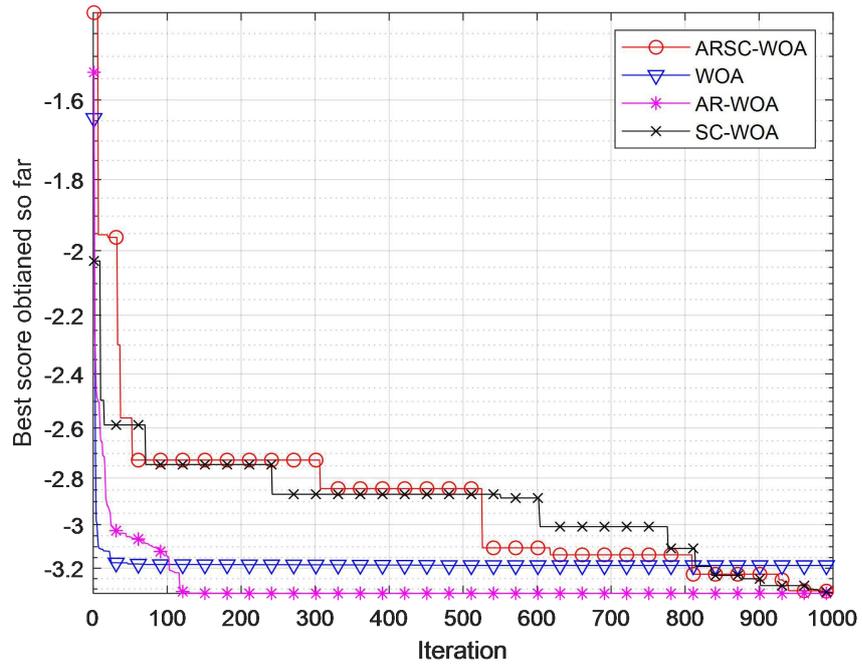
(16) Convergence curves of F_{16}



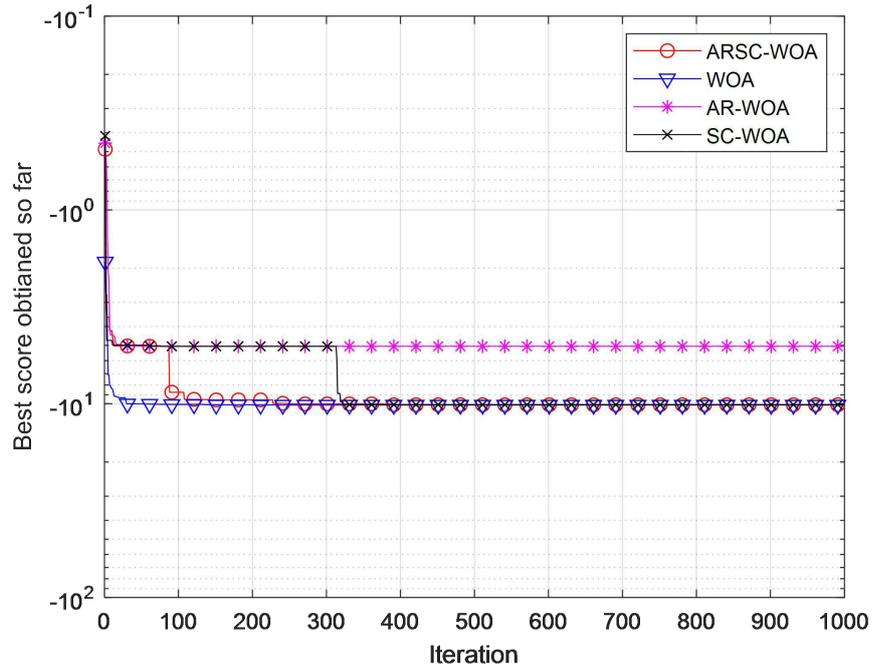
(17) Convergence curves of F_{17}



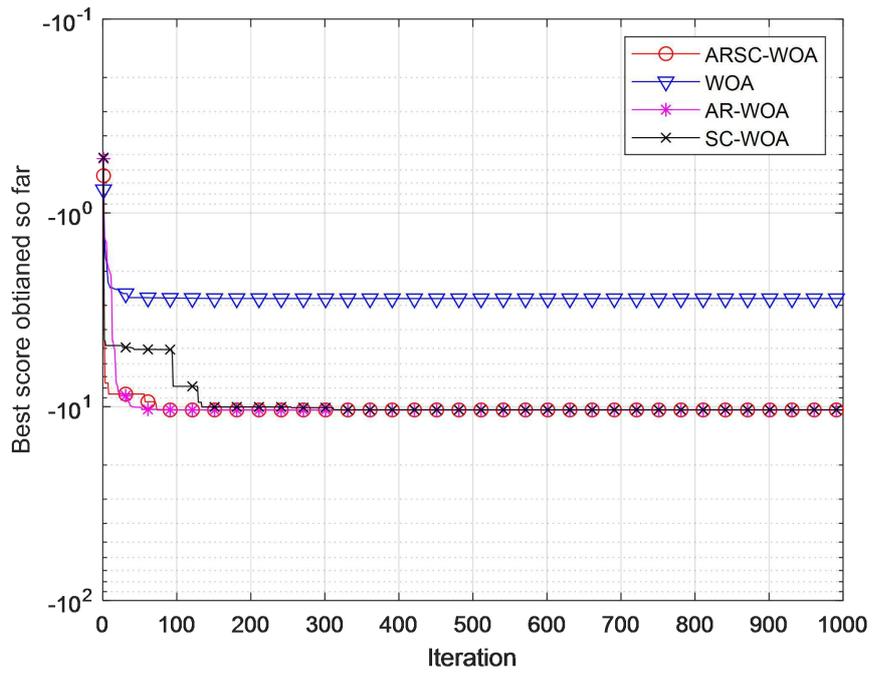
(18) Convergence curves of F_{18}



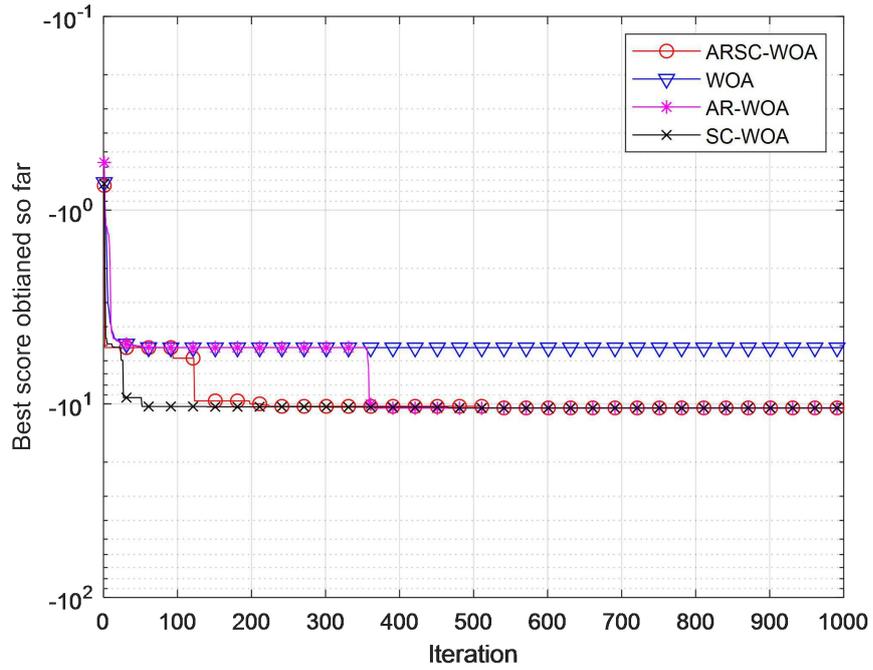
(19) Convergence curves of F_{19}



(20) Convergence curves of F_{20}



(21) Convergence curves of F_{21}



(22) Convergence curves of F_{22}

Fig. 19 Simulation results

Table 7 Comparison of improvement strategies

Function		ARSC-WOA	WOA	AR-WOA	SC-WOA
F1	Best	0	6.27E-176	0	0
	Ave	0	8.20E-169	2.54E-186	0
	Std	0	0	0	0
F2	Best	0	3.54E-180	0	0
	Ave	0	6.83E-163	5.49E-285	0
	Std	0	2.22E-162	0	0
F3	Best	0	1.62E-108	2.34E-132	0
	Ave	0	8.57E-96	4.93E-94	0
	Std	0	2.25E-95	1.56E-93	0
F4	Best	0	2.87E-40	7.09E-120	0
	Ave	0	2.43E-30	4.44E-79	0
	Std	0	6.74E-30	1.40E-78	0
F5	Best	7.28E-06	2.60E+01	2.70E+01	1.77E-08
	Ave	2.24E+01	2.67E+01	2.80E+01	1.11E+01
	Std	1.18E+01	3.00E-01	7.70E-01	1.43E+01
F6	Best	2.66E-11	3.86E-03	2.56E-03	4.87E-11
	Ave	1.71E-09	9.46E-03	6.35E-01	2.27E-09
	Std	1.90E-09	7.40E-03	7.03E-01	3.36E-09
F7	Best	2.72E-05	4.23E-04	5.61E-05	3.10E-06
	Ave	2.67E-04	2.14E-03	1.88E-03	8.38E-05
	Std	4.56E-04	2.26E-03	3.13E-03	7.97E-05

F8	Best	-12569.4866	-12569.3488	-12569.4178	-12569.4866
	Ave	-1.26E+04	-1.19E+04	-1.26E+04	-1.26E+04
	Std	1.02E-05	1.05E+03	8.25E-02	1.64E-05
F9	Best	0	0	0	0
	Ave	0	0	0	0
	Std	0	0	0	0
F10	Best	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	Ave	8.88E-16	3.02E-15	2.31E-15	8.88E-16
	Std	0	2.48E-15	1.83E-15	0.00E+00
F11	Best	0	0	0	0
	Ave	0	2.67E-03	3.21E-03	0
	Std	0	8.43E-03	1.01E-02	0
F12	Best	5.69E-14	3.11E-04	1.04E-04	1.64E-11
	Ave	1.69E-10	3.66E-03	2.61E-04	4.10E-10
	Std	2.79E-10	5.54E-03	1.53E-04	4.84E-10
F13	Best	3.44E-11	3.11E-03	9.51E-04	1.98E-11
	Ave	2.50E-09	6.42E-02	9.73E-03	4.79E-09
	Std	2.53E-09	6.85E-02	9.27E-03	5.78E-09
F14	Best	0.998	0.998	0.998	0.998
	Ave	0.998	2.5698	0.998	0.998
	Std	2.34E-14	3.027	9.25E-12	6.74E-12
F15	Best	3.08E-04	3.08E-04	3.36E-04	3.08E-04
	Ave	5.16E-04	6.09E-04	7.76E-04	7.70E-04
	Std	4.37E-04	3.01E-04	4.16E-04	5.96E-04
F16	Best	0.39789	0.39789	0.39789	0.39789
	Ave	0.39789	0.39789	0.39789	0.39789
	Std	8.46E-07	2.28E-07	8.00E-14	2.66E-06
F17	Best	3	3	3	3
	Ave	3	3	3.0005	3
	Std	1.55E-05	8.03E-06	9.38E-04	6.01E-05
F18	Best	-3.8628	-3.8628	-3.8628	-3.8628
	Ave	-3.8627	-3.86	-3.8619	-3.8627
	Std	2.88E-05	3.19E-03	1.99E-03	5.54E-05
F19	Best	-3.3194	-3.3219	-3.322	-3.3218
	Ave	-2.9496	-3.2944	-3.2422	-3.2393
	Std	3.57E-01	5.76E-02	6.99E-02	7.43E-02
F20	Best	-10.1532	-10.1531	-10.1532	-10.1532
	Ave	-10.1532	-9.6427	-8.114	-10.1532
	Std	1.13E-06	1.6119	2.6326	1.78E-06
F21	Best	-10.4028	-10.4028	-10.4029	-10.4029
	Ave	-10.4028	-9.2029	-8.8073	-10.4028
	Std	4.04E-05	2.5486	2.5693	3.99E-05

	Best	-10.5363	-10.5364	-10.5364	-10.5364
F22	Ave	-10.5363	-8.7809	-9.4531	-10.5363
	Std	2.58E-05	2.8411	2.2837	3.68E-05

Table 8 P-value of Wilcoxon test

Function	ARSC-WOA	ARSC-WOA	ARSC-WOA	SC-WOA
	vs WOA	vs AR-WOA	vs SC-WOA	vs AR-WOA
F1	6.39E-05	2.20E-03	NaN	2.20E-03
F2	6.39E-05	7.51E-04	NaN	7.51E-04
F3	6.39E-05	6.39E-05	NaN	6.39E-05
F4	6.39E-05	6.39E-05	NaN	6.39E-05
F5	2.57E-02	1.86E-01	5.39E-02	1.13E-02
F6	1.83E-04	1.83E-04	7.34E-01	1.83E-04
F7	1.73E-02	3.76E-02	3.85E-01	2.80E-03
F8	1.83E-04	1.83E-04	1.62E-01	1.83E-04
F9	NaN	NaN	NaN	NaN
F10	1.80E-03	3.36E-02	NaN	3.36E-02
F11	3.68E-01	3.68E-01	NaN	3.68E-01
F12	1.83E-04	1.83E-04	3.08E-01	1.83E-04
F13	1.83E-04	1.83E-04	7.34E-01	1.83E-04
F14	3.30E-04	9.40E-01	1.30E-03	1.60E-01
F15	3.08E-01	1.21E-01	6.78E-01	3.85E-01
F16	1.13E-02	1.32E-04	7.34E-01	1.32E-04
F17	7.57E-02	1.13E-02	9.10E-01	3.12E-02
F18	2.20E-03	4.27E-01	5.71E-01	8.50E-01
F19	7.30E-03	2.57E-02	5.39E-02	4.73E-01
F20	1.83E-04	4.73E-01	2.12E-01	4.73E-01
F21	2.20E-03	1.62E-01	5.21E-01	1.41E-01
F22	1.00E-03	2.57E-02	6.23E-01	2.57E-02

4.6 High-dimensional Function Optimization

High dimensionality and non-linearity are common in combinatorial optimization problems. For example, traditional methods such as back propagation and gradient descent can simply and effectively optimize neural networks. However, the quality of the global optimal solution depends on the initial solution, which is easy to fall into the local optimal. Meta-heuristics as a trainer for neural networks can avoid local optimization. At the same time, due to the differences in neural network structure and optimization parameters, the dimensions that the meta-heuristic algorithm needs to optimize may be more than 50 dimensions [49-50], which tests the performance of the algorithm. In this paper, the dimensions of functions F1-F13 are increased to verify the possibility of improved WOA to solve high-dimensional optimization problems. The search range of the unimodal functions F1-F7 will increase as the dimensions increase. In this case, the global convergence of the search agent is reduced.

The multimodal functions F8-F13 have a large number of local optimal values. The increase of the dimensions causes the local optimal values to increase sharply, which interferes with the tendency of the search agent to develop towards the global optimal value. If the algorithm does not have a strong global exploration capability, it will fall into the local optimal value stagnation.

For each dimension of the function, the algorithm runs independently 10 times. For different dimensions, the test results listed in Table 9 show that ARSC-WOA found the theoretical optimal value in 54% of the functions. It should be noted that the optimal value of the function F8 is $f_{\min} = -418.9829 \cdot D$, and the global optimal value will shift with the dimensions. For different dimensions, ARSC-WOA can find the theoretical optimal value of F8. It shows that the improvement of the dimension will not have a negative impact on the performance of ARSC-WOA. The introduction of improved strategies enables searching agents to avoid local optimization. Table 10 shows the p-value results of the Wilcoxon rank sum test. Statistical results show that the performance of ARSC-WOA when dealing with high-dimensional optimization problems is not much different from low-dimensional optimization. The improved WOA did not fall into the dimensional disaster, which lays a theoretical foundation for ARSC-WOA to deal with complex optimization problems of high dimensions.

Table 9 High-dimensional optimization results

Function	Performance	ARSC-WOA			
		D=50	D=100	D=150	D=200
F1	Best	0	0	0	0
	Ave	0	0	0	0
	Std	0	0	0	0
F2	Best	0	0	0	0
	Ave	0	0	0	0
	Std	0	0	0	0
F3	Best	0	0	0	0
	Ave	0	0	0	0
	Std	0	0	0	0
F4	Best	0	0	0	0
	Ave	0	0	0	0
	Std	0	0	0	0
F5	Best	5.22E-08	4.05E-08	1.47E+02	2.26E-05
	Ave	2.41E+01	5.88E+01	1.47E+02	1.38E+02
	Std	2.54E+01	5.06E+01	8.93E-02	9.51E+01
F6	Best	3.24E-11	8.30E-11	2.64E-10	5.44E-11
	Ave	3.28E-09	1.48E-08	1.35E-08	1.22E-08
	Std	3.50E-09	1.54E-08	1.49E-08	1.19E-08
F7	Best	3.13E-05	5.86E-06	2.83E-05	2.72E-05
	Ave	3.24E-04	8.46E-05	2.49E-04	7.19E-04
	Std	4.29E-04	6.80E-05	3.01E-04	1.13E-03
F8	Best	-20949.1444	-41898.2887	-62847.4331	-83796.5775
	Ave	-2.09E+04	-4.19E+04	-6.28E+04	-8.38E+04

	Std	4.50E-06	3.46E-05	3.40E-05	7.64E-05
F9	Best	0	0	0	0
	Ave	0	0	0	0
	Std	0	0	0	0
F10	Best	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	Ave	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	Std	0	0	0	0
F11	Best	0	0	0	0
	Ave	0	0	0	0
	Std	0	0	0	0
F12	Best	9.15E-12	1.38E-13	1.10E-12	7.08E-13
	Ave	9.55E-11	1.35E-10	8.45E-11	3.87E-11
	Std	1.02E-10	1.76E-10	8.50E-11	4.66E-11
F13	Best	4.04E-11	3.76E-13	7.26E-10	1.65E-10
	Ave	3.18E-09	1.36E-09	5.03E-09	7.69E-09
	Std	3.82E-09	1.02E-09	5.46E-09	1.19E-08

Table 10 P-value of Wilcoxon test

Function	ARSC-WOA			
	D=50	D=100	D=150	D=200
	vs D=30	vs D=30	vs D=30	vs D=30
F1	NaN	NaN	NaN	NaN
F2	NaN	NaN	NaN	NaN
F3	NaN	NaN	NaN	NaN
F4	NaN	NaN	NaN	NaN
F5	6.78E-01	3.85E-01	1.83E-04	5.39E-02
F6	5.21E-01	5.80E-03	1.13E-02	3.12E-02
F7	6.78E-01	5.21E-01	8.50E-01	1.62E-01
F8	1.83E-04	1.83E-04	1.83E-04	1.83E-04
F9	NaN	NaN	NaN	NaN
F10	NaN	NaN	NaN	NaN
F11	NaN	NaN	NaN	NaN
F12	7.34E-01	8.50E-01	7.34E-01	1.21E-01
F13	8.50E-01	4.27E-01	2.12E-01	3.45E-01

4.7 Engineering Optimization Design

In order to more fully evaluate the optimization performance of ARSC-WOA, this paper applies the improved WOA to solve engineering optimization problems. The three-bar truss design problem is a classic engineering optimization problem with the purpose of minimizing the weight of the bar truss. The structure and parameters of the three-bar truss design problem are shown in Fig. 20. Two variables that need to be optimized are A_1 and A_2 .

The objective function, optimization constraints, and optimization variables of the three-bar truss optimization problem can be described as:

$$\text{Consider } \bar{x} = [x_1 \ x_2] = [A_1 \ A_2] \quad (20)$$

$$\text{Minimize } f(\bar{x}) = (2\sqrt{2}x_1 + x_2) \cdot l$$

$$\text{Subject } g_1(\bar{x}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} p - \sigma \leq 0$$

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

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

$$\text{Variable range } 0 \leq x_1, x_2 \leq 1$$

$$\text{where } l = 100\text{cm}, p = 2\text{kn/cm}^2, \sigma = 2\text{kn/cm}^2$$

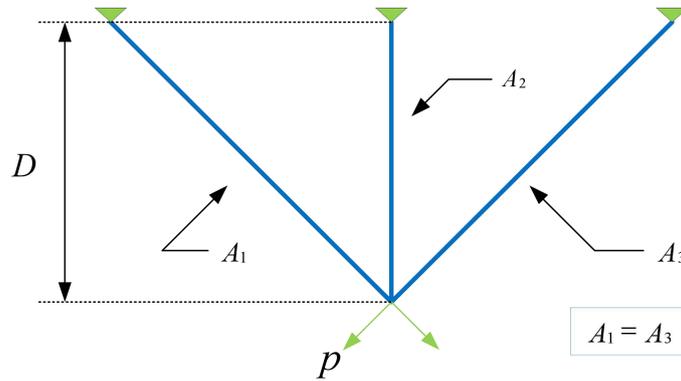


Fig. 20 Three-bar truss design problem

In order to more objectively show the performance of ARSC-WOA, the optimization costs of different algorithms are selected for comparison. The optimization results in Table 11 show that ARSC-WOA has the best optimization cost. The comparison of the statistical results in Table 12 shows that ARSC-WOA is superior to SSA, GOA and WOA in mean and variance. ARSC-WOA is not only better than other algorithms, but also better than WOA. The improved WOA can effectively solve the three-bar truss design problem.

Table 11 Comparison of optimized weight

Algorithm	Optimum variables		Optimum weight
	A1	A2	
ARSC-WOA	0.786230529	0.289207095	186.3861
WOA	0.912618406	0.093722977	192.125
SSA[45]	0.788665414	0.408275784	263.8958434
GOA[51]	0.788897556	0.40761957	263.8958815
DEDS[52]	0.78867513	0.40824828	263.8958434
PSO-DE[53]	0.7886751	0.4082482	263.8958433

MBA[54]	0.788565	0.4085597	263.8958522
Ray and Sain[55]	0.795	0.395	264.3
Tsa[56]	0.788	0.408	263.68
CS[57]	0.78867	0.40902	263.9716

Table 12 Comparison of statistical results

Algorithm	Average	Standard Deviation
ARSC-WOA	186.3895	0.0069833
WOA	188.5398	2.7514
SSA	265.3652	0.1854
GOA	265.8606	0.0532

In addition, this paper also applies the improved WOA to solve the welded beam design problem. The purpose of this engineering problem is to minimize welding costs. The structure and parameters of the welded beam design problem are shown in Fig. 21. The four constraints are shear stress τ , bending stress in the beam θ , buckling load p_c , and deflection of the beam δ . These four variables that need to be optimized are the thickness of the weld seam h , the length of the clamp beam l , the height of the bar t , and the thickness of the bar b . The objective function, optimization constraints, and optimization variables of the welding beam optimization problem can be described as:

$$\text{Consider } \vec{x} = [x_1 \ x_2 \ x_3 \ x_4] = [h \ l \ t \ b] \quad (21)$$

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

$$\text{Subject } g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \leq 0$$

$$g_2(\vec{x}) = \theta(\vec{x}) - \theta_{\max} \leq 0$$

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

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

$$g_5(\vec{x}) = p - p_c(\vec{x}) \leq 0$$

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

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

$$\text{Variable range } 0.1 \leq x_1 \leq 2$$

$$0.1 \leq x_2 \leq 10$$

$$0.1 \leq x_3 \leq 10$$

$$0.1 \leq x_4 \leq 2$$

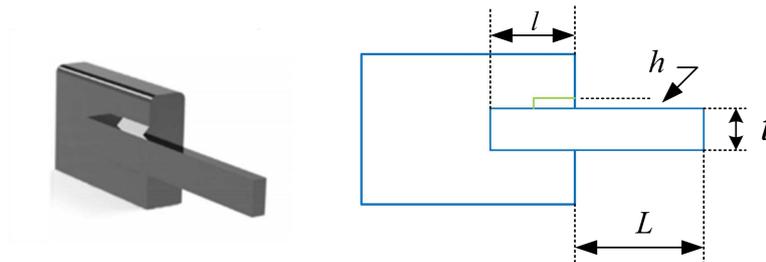


Fig. 21 Welded beam optimization problem

In order to more objectively show the performance of ARSC-WOA, the optimization costs of different algorithms are selected for comparison. In addition, mathematical methods[61] such as random method, simplex method, and successive linear approximation are selected in this paper. The optimization results in Table 13 show that ARSC-WOA has the best optimization cost. The comparison of the statistical results in Table 14 shows that ARSC-WOA is superior to GSA, PSO and WOA in mean and variance. Experimental results verify that ARSC-WOA is not only better than other algorithms, but also better than WOA. This shows that the optimization performance of the improved WOA is strong and can effectively solve the welded beam design problem.

Table 13 Comparison of optimization cost

Algorithm	Optimum variables				Optimum cost
	h	l	t	b	
ARSC-WOA	0.196814651	3.464542879	9.035922994	0.206103275	1.713
WOA[43]	0.205396	3.484293	9.037426	0.206276	1.730499
GSA[43]	0.182129	3.856979	10	0.202376	1.879952
GA (Deb)[58,59]	0.2489	6.173	8.1789	0.2533	2.4331
HS (Lee and Geem)[60]	0.2442	6.2231	8.2915	0.2443	2.3807
Random[61]	0.4575	4.7313	5.0853	0.66	4.1185
Simplex[61]	0.2792	5.6256	7.7512	0.2796	2.5307
David[61]	0.2434	6.2552	8.2915	0.2444	2.3841
APPROX[61]	0.2444	6.2189	8.2915	0.2444	2.3815

Table 14 Comparison of statistical results

Algorithm	Average	Standard Deviation
ARSC-WOA	1.7128	0.012822
WOA	1.732	0.0226
PSO	1.7422	0.01275
GSA	3.5761	1.2874

5. Conclusions

As a swarm intelligence based algorithm, WOA relies less on parameters and operators. Aiming at the shortcomings of the spiral update mechanism in WOA, this paper proposes three kinds of spiral search strategies, namely Archimedes spiral search, Rose spiral search and Hypotrochoid spiral search. The experimental results show that the optimization effect of Archimedes spiral search is the best, and it can avoid exceeding the search range. At the same time, Archimedes spiral search can effectively improve the population diversity of search agents. In addition, the introduction of improved sine cosine operator enables the algorithm to use sine search for global exploration and cosine search for local exploitation. The convergence factor enables the search agent to smoothly transition between exploration and exploitation with iterative adaptation. The algorithm retains the optimal search method through the merit based strategy. The sine search and cosine search can also search different regions based on the returned value, and the optimization performance has been further improved. In the experimental simulation part, the improved WOA based on the Archimedes spiral search strategy and the sine cosine operator (ARSC-WOA)

has shown strong competitiveness compared with other algorithms. The introduction of improved strategies enables the algorithm to effectively solve high-dimensional optimization and engineering optimization problems. The future research direction is to apply the improved WOA to more combinatorial optimization problems, such as multi-objective optimization problems, binary optimization problems, and regression prediction models.

Compliance with Ethical Standards: This article does not contain any studies with human participants or animals performed by any of the authors.

Acknowledgement: This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. 2017FWDF10), and the Project by Liaoning Provincial Natural Science Foundation of China (Grant No. 20180550700).

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

Author Contributions: Ji Zhang participated in the data collection, algorithm simulation and draft writing. Kai Yang and Jie-Sheng Wang participated in the concept, design, interpretation and commented on the manuscript and the critical revision of this paper.

References

- [1] Holland J H. Genetic algorithms[J]. Scientific american, 1992, 267(1): 66-73.
- [2] Rechenberg I. Evolutionsstrategien[M]//Simulationsmethoden in der Medizin und Biologie. Springer, Berlin, Heidelberg, 1978: 83-114.
- [3] Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of global optimization, 1997, 11(4): 341-359.
- [4] Evolutionary algorithms in engineering applications[M]. Springer Science & Business Media, 2013.
- [5] Eberhart R, Kennedy J. Particle swarm optimization[C]//Proceedings of the IEEE international conference on neural networks. 1995, 4: 1942-1948.
- [6] Yang, She X . Firefly algorithm, stochastic test functions and design optimisation[J]. International Journal of Bio-Inspired Computation, 2010, 2(2):78-84.
- [7] Yang X S. A new metaheuristic bat-inspired algorithm[M]//Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, Berlin, Heidelberg, 2010: 65-74.
- [8] Askarzadeh A, Rezazadeh A. A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: bird mating optimizer[J]. International Journal of Energy Research, 2013, 37(10): 1196-1204.
- [9] Yang X S, Deb S. Cuckoo search via Lévy flights[C]//2009 World Congress on Nature & Biologically Inspired Computing (NaBIC). IEEE, 2009: 210-214.
- [10] Saremi S, Mirjalili S, Lewis A. Grasshopper optimisation algorithm: theory and application[J]. Advances in Engineering Software, 2017, 105: 30-47.
- [11] Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems[J]. Neural Computing and Applications, 2016, 27(4): 1053-1073.
- [12] Kaveh A, Farhodi N. A new optimization method: Dolphin echolocation[J]. Advances in Engineering Software, 2013, 59: 53-70.
- [13] Pan W T. A new fruit fly optimization algorithm: taking the financial distress model as an example[J].

Knowledge-Based Systems, 2012, 26: 69-74.

- [14] Gandomi A H, Alavi A H. Krill herd: a new bio-inspired optimization algorithm[J]. Communications in nonlinear science and numerical simulation, 2012, 17(12): 4831-4845.
- [15] Pinto P C, Runkler T A, Sousa J M C. Wasp swarm algorithm for dynamic MAX-SAT problems[C]//International Conference on Adaptive and Natural Computing Algorithms. Springer, Berlin, Heidelberg, 2007: 350-357.
- [16] Lu X, Zhou Y. A novel global convergence algorithm: bee collecting pollen algorithm[C]//International Conference on Intelligent Computing. Springer, Berlin, Heidelberg, 2008: 518-525.
- [17] Hatamlou A. Black hole: A new heuristic optimization approach for data clustering[J]. Information sciences, 2013, 222: 175-184.
- [18] Formato R A. Central force optimization[J]. Prog Electromagn Res, 2007, 77: 425-491.
- [19] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm[J]. Information sciences, 2009, 179(13): 2232-2248.
- [20] Alatas B. ACROA: artificial chemical reaction optimization algorithm for global optimization[J]. Expert Systems with Applications, 2011, 38(10): 13170-13180.
- [21] Moghaddam F F, Moghaddam R F, Cheriet M. Curved space optimization: a random search based on general relativity theory[J]. arXiv preprint arXiv:1208.2214, 2012.
- [22] Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search[J]. Acta Mechanica, 2010, 213(3-4): 267-289.
- [23] Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by simulated annealing[J]. science, 1983, 220(4598): 671-680.
- [24] Tan Y, Zhu Y. Fireworks algorithm for optimization[C]//International conference in swarm intelligence. Springer, Berlin, Heidelberg, 2010: 355-364.
- [25] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition[C]//2007 IEEE congress on evolutionary computation. IEEE, 2007: 4661-4667.
- [26] Moosavian N, Roodsari B K. Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks[J]. Swarm and Evolutionary Computation, 2014, 17: 14-24.
- [27] Shi Y. An optimization algorithm based on brainstorming process[M]//Emerging Research on Swarm Intelligence and Algorithm Optimization. IGI Global, 2015: 1-35.
- [28] Shayeghi H, Dadashpour J. Anarchic society optimization based PID control of an automatic voltage regulator (AVR) system[J]. Electrical and Electronic Engineering, 2012, 2(4): 199-207.
- [29] Ramezani F, Lotfi S. Social-based algorithm (SBA)[J]. Applied Soft Computing, 2013, 13(5): 2837-2856.
- [30] Sadollah A, Bahreininejad A, Eskandar H, et al. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems[J]. Applied Soft Computing, 2013, 13(5): 2592-2612.
- [31] Rao R V, Savsani V J, Vakharia D P. Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems[J]. Computer-Aided Design, 2011, 43(3): 303-315.
- [32] Glover F. Tabu search—part I[J]. ORSA Journal on computing, 1989, 1(3): 190-206.
- [33] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in engineering software, 2014, 69: 46-61.
- [34] Pelikan M, Goldberg D E, Cantú-Paz E. BOA: The Bayesian optimization algorithm[C]//Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1. Morgan Kaufmann Publishers Inc., 1999: 525-532.
- [35] Pelikan M, Goldberg D E. Hierarchical BOA solves Ising spin glasses and MAXSAT[C]//Genetic and Evolutionary Computation Conference. Springer, Berlin, Heidelberg, 2003: 1271-1282.
- [36] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems[J]. Knowledge-Based Systems,

2016, 96: 120-133.

- [37] Mirjalili S, Lewis A. The whale optimization algorithm[J]. *Advances in engineering software*, 2016, 95: 51-67.
- [38] Mafarja M, Mirjalili S. Whale optimization approaches for wrapper feature selection[J]. *Applied Soft Computing*, 2018, 62: 441-453.
- [39] Kaveh A, Ghazaan M I. Enhanced whale optimization algorithm for sizing optimization of skeletal structures[J]. *Mechanics Based Design of Structures and Machines*, 2017, 45(3): 345-362.
- [40] Ling Y, Zhou Y, Luo Q. Lévy flight trajectory-based whale optimization algorithm for global optimization[J]. *IEEE access*, 2017, 5: 6168-6186.
- [41] Abdel-Basset M, Manogaran G, El-Shahat D, et al. A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem[J]. *Future Generation Computer Systems*, 2018, 85: 129-145.
- [42] Prakash D B, Lakshminarayana C. Optimal siting of capacitors in radial distribution network using whale optimization algorithm[J]. *Alexandria Engineering Journal*, 2017, 56(4): 499-509.
- [43] Xiong G, Zhang J, Shi D, et al. Parameter extraction of solar photovoltaic models using an improved whale optimization algorithm[J]. *Energy conversion and management*, 2018, 174: 388-405.
- [44] Wolpert D H, Macready W G. No free lunch theorems for optimization[J]. *IEEE transactions on evolutionary computation*, 1997, 1(1): 67-82.
- [45] Mirjalili S, Gandomi A H, Mirjalili S Z, et al. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems[J]. *Advances in Engineering Software*, 2017, 114: 163-191.
- [46] Mirjalili S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm[J]. *Knowledge-Based Systems*, 2015, 89: 228-249.
- [47] Hof P R, Van der Gucht E. Structure of the cerebral cortex of the humpback whale, *Megaptera novaeangliae* (Cetacea, Mysticeti, Balaenopteridae)[J]. *The Anatomical Record: Advances in Integrative Anatomy and Evolutionary Biology: Advances in Integrative Anatomy and Evolutionary Biology*, 2007, 290(1): 1-31.
- [48] Rosner B, Glynn R J, Ting Lee M L. Incorporation of clustering effects for the Wilcoxon rank sum test: a large - sample approach[J]. *Biometrics*, 2003, 59(4): 1089-1098.
- [49] Mirjalili S. How effective is the Grey Wolf optimizer in training multi-layer perceptrons[J]. *Applied Intelligence*, 2015, 43(1): 150-161.
- [50] Xie W, Wang J S, Xing C, et al. Extreme Learning Machine Soft-Sensor Model With Different Activation Functions on Grinding Process Optimized by Improved Black Hole Algorithm[J]. *IEEE Access*, 2020, 8: 25084-25110.
- [51] Saremi S, Mirjalili S, Lewis A. Grasshopper optimisation algorithm: theory and application[J]. *Advances in Engineering Software*, 2017, 105: 30-47.
- [52] Zhang M, Luo W, Wang X. Differential evolution with dynamic stochastic selection for constrained optimization[J]. *Information Sciences*, 2008, 178(15): 3043-3074.
- [53] Liu H, Cai Z, Wang Y. Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization[J]. *Applied Soft Computing*, 2010, 10(2): 629-640.
- [54] Sadollah A, Bahreininejad A, Eskandar H, et al. Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems[J]. *Applied Soft Computing*, 2013, 13(5): 2592-2612.
- [55] Ray T, Saini P. Engineering design optimization using a swarm with an intelligent information sharing among individuals[J]. *Engineering Optimization*, 2001, 33(6): 735-748.
- [56] Tsai J F. Global optimization of nonlinear fractional programming problems in engineering design[J]. *Engineering Optimization*, 2005, 37(4): 399-409.
- [57] Gandomi A H, Yang X S, Alavi A H. Cuckoo search algorithm: a metaheuristic approach to solve structural

- optimization problems[J]. *Engineering with computers*, 2013, 29(1): 17-35.
- [58] Deb K. Optimal design of a welded beam via genetic algorithms[J]. *AIAA journal*, 1991, 29(11): 2013-2015.
- [59] Deb K. An efficient constraint handling method for genetic algorithms[J]. *Computer methods in applied mechanics and engineering*, 2000, 186(2-4): 311-338.
- [60] Lee K S, Geem Z W. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice[J]. *Computer methods in applied mechanics and engineering*, 2005, 194(36-38): 3902-3933.
- [61] Ragsdell K M, Phillips D T. Optimal design of a class of welded structures using geometric programming[J]. 1976.

Figures

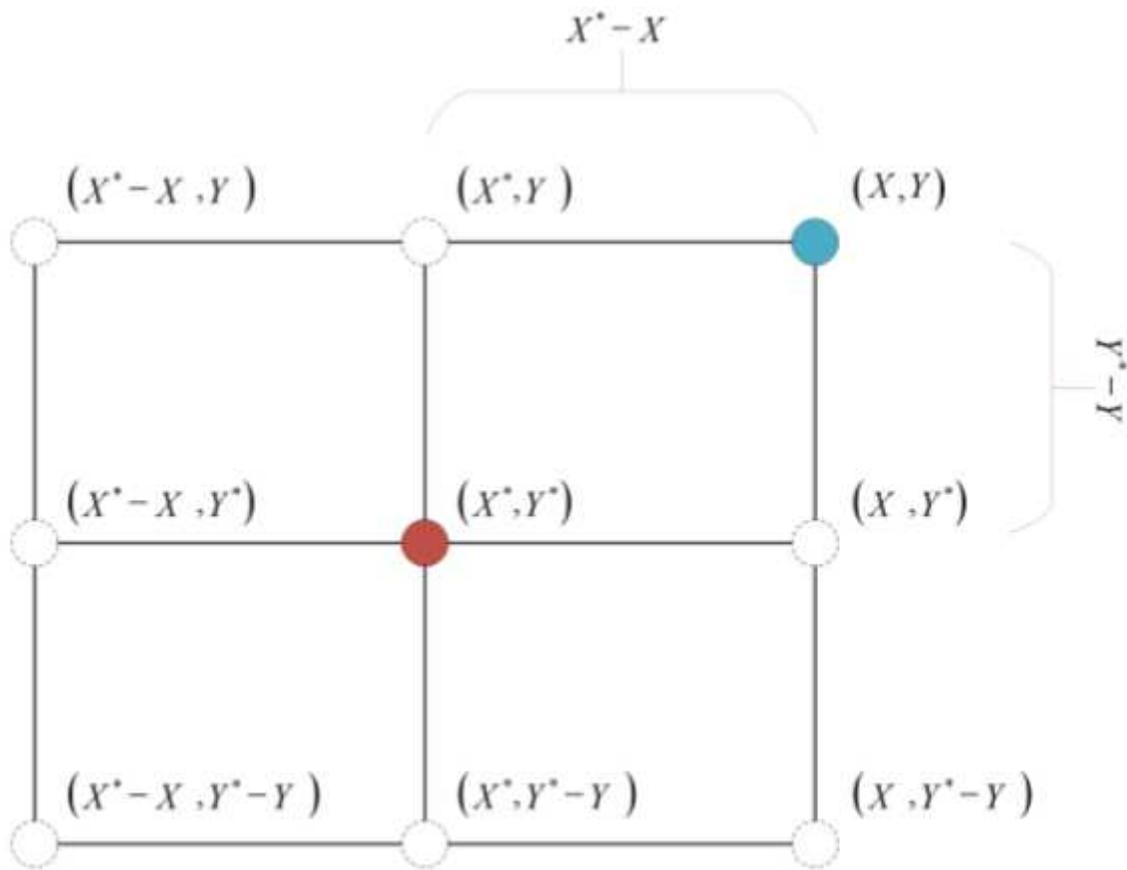


Figure 1

2D mathematical model of WOA

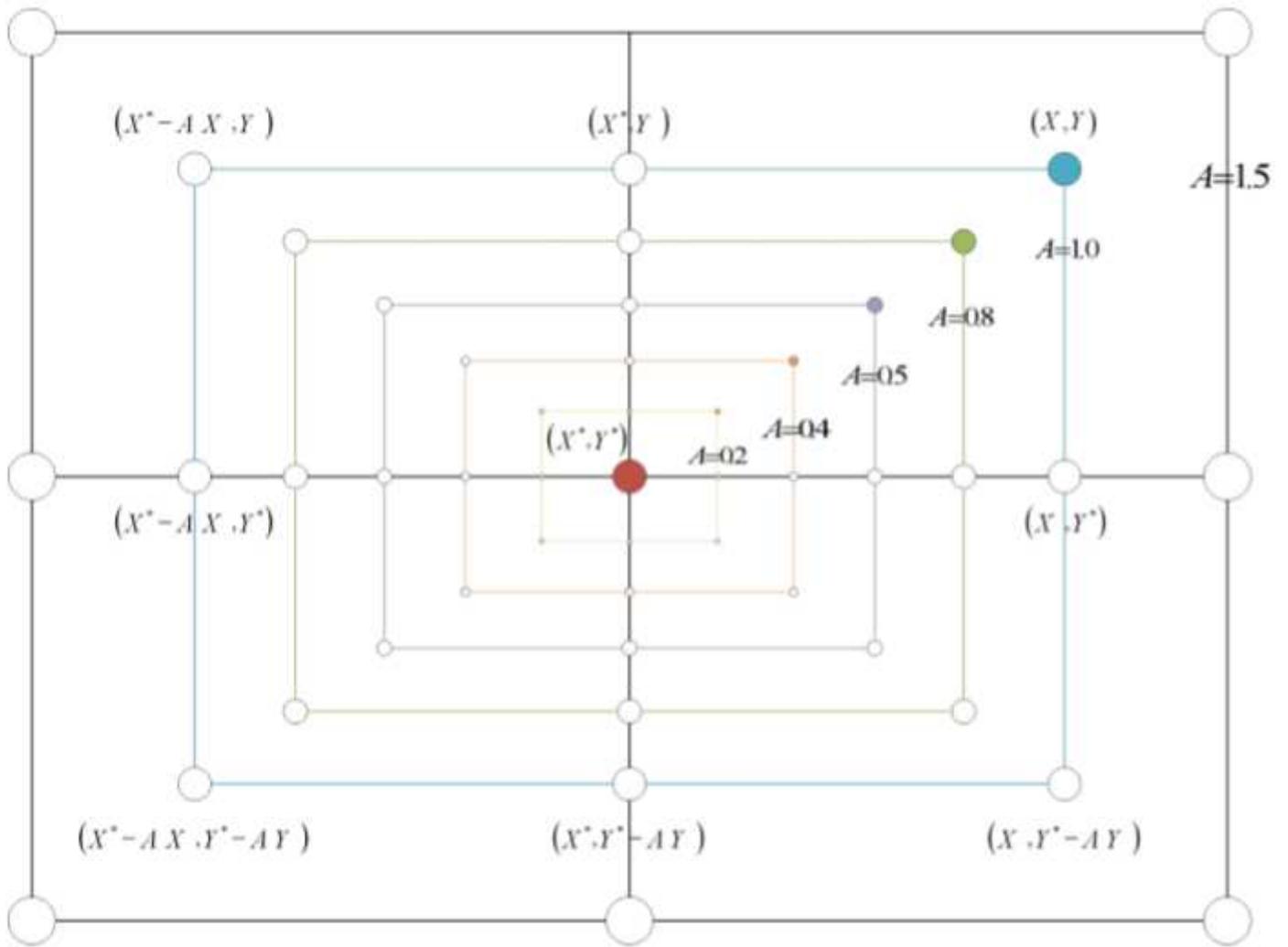


Figure 2

Shrink enclosing mechanism model

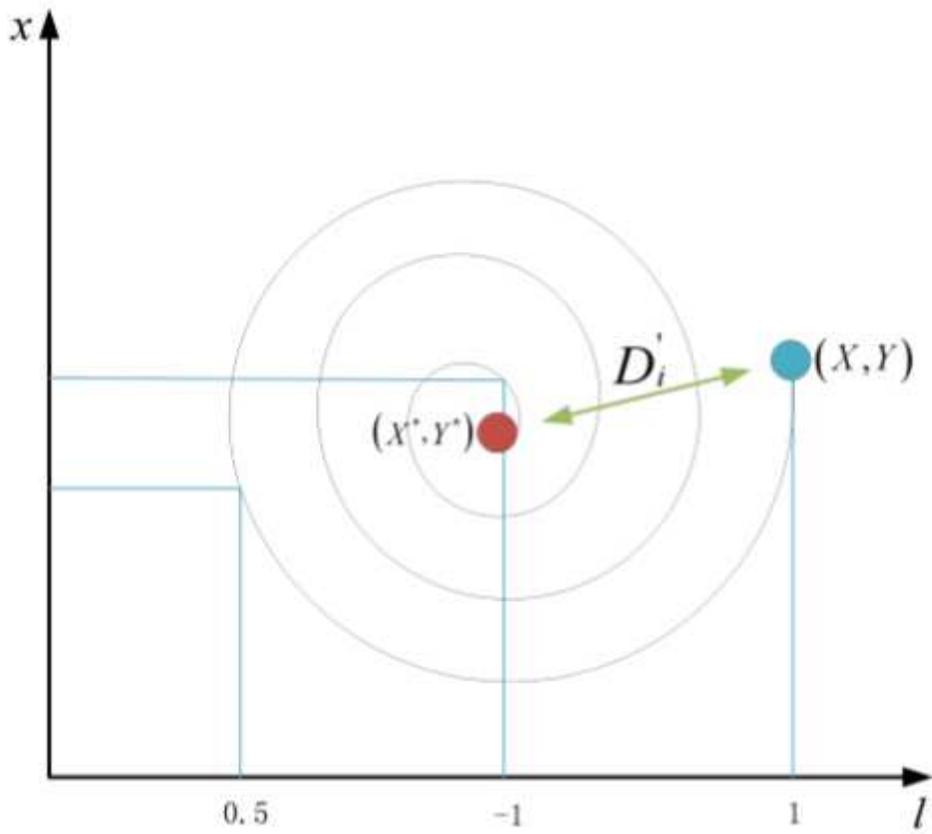


Figure 3

Spiral update mechanism model

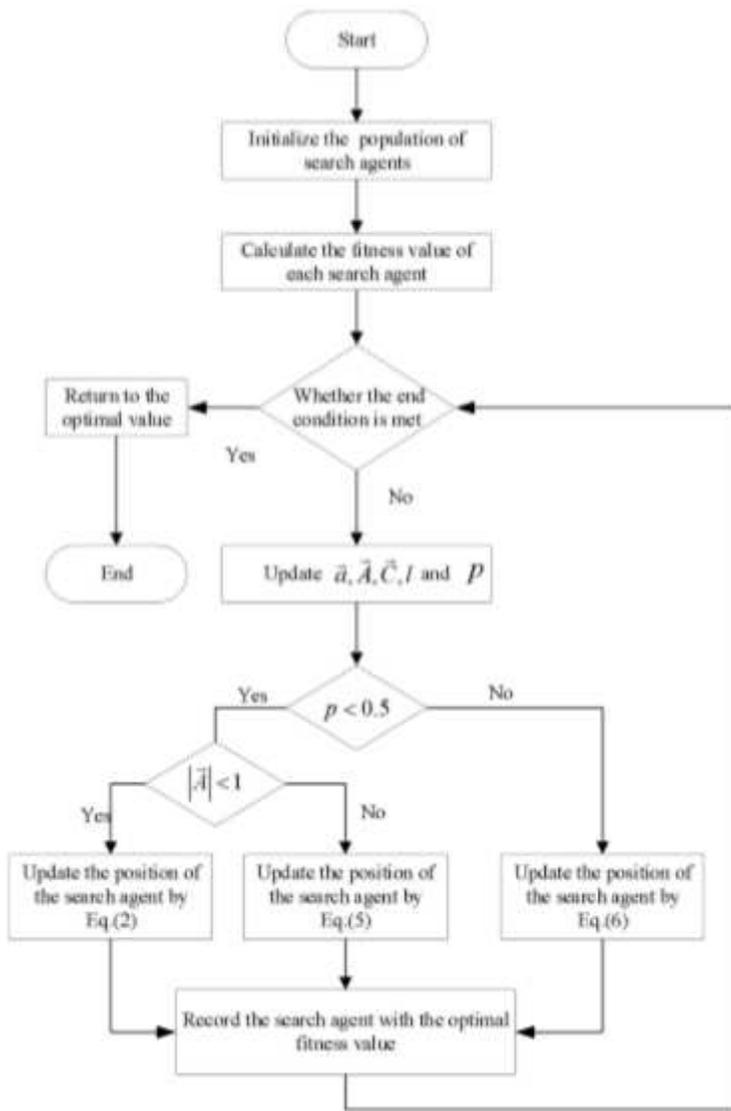


Figure 4

Flow chart of WOA

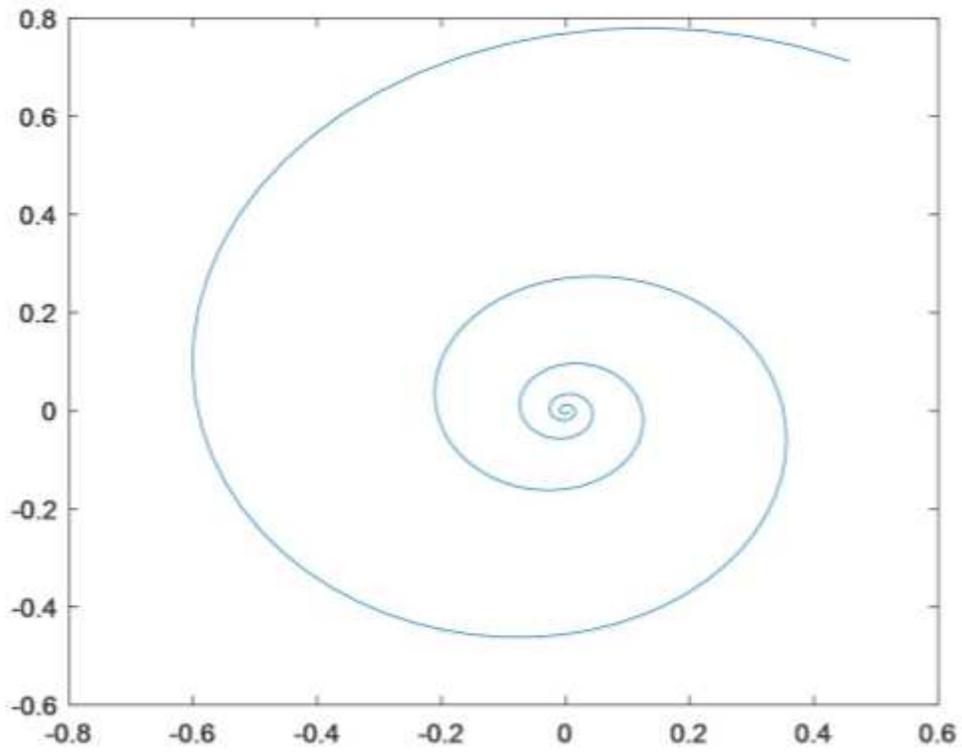


Figure 5

2D image of logarithmic spiral

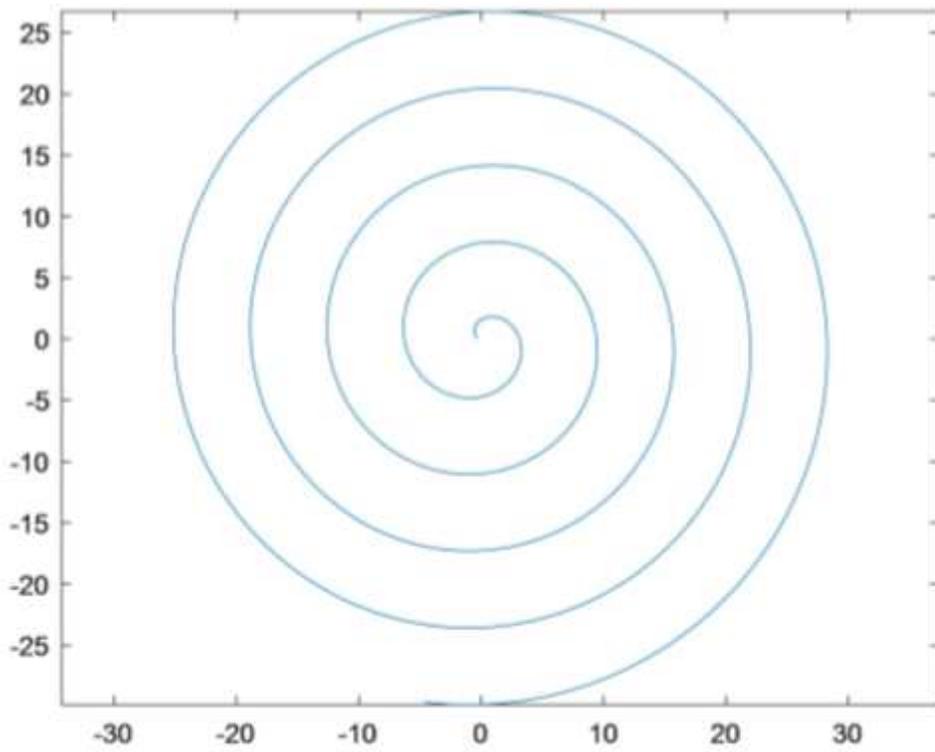


Figure 6

2D image of Archimedes spiral

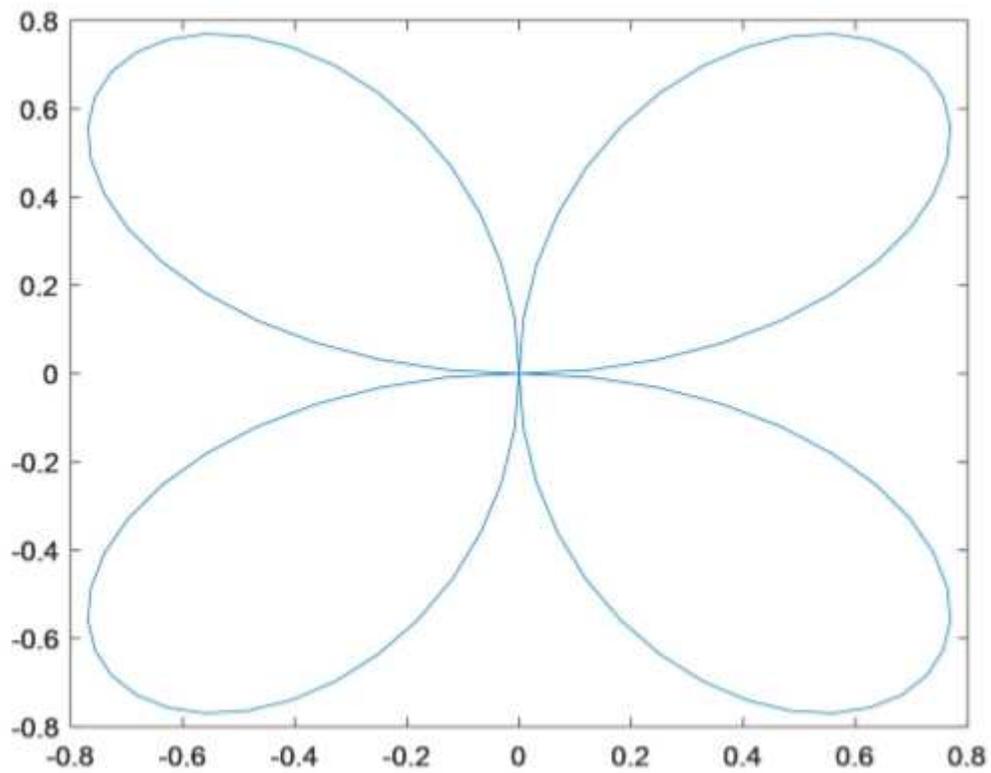


Figure 7

2D image of rose spiral

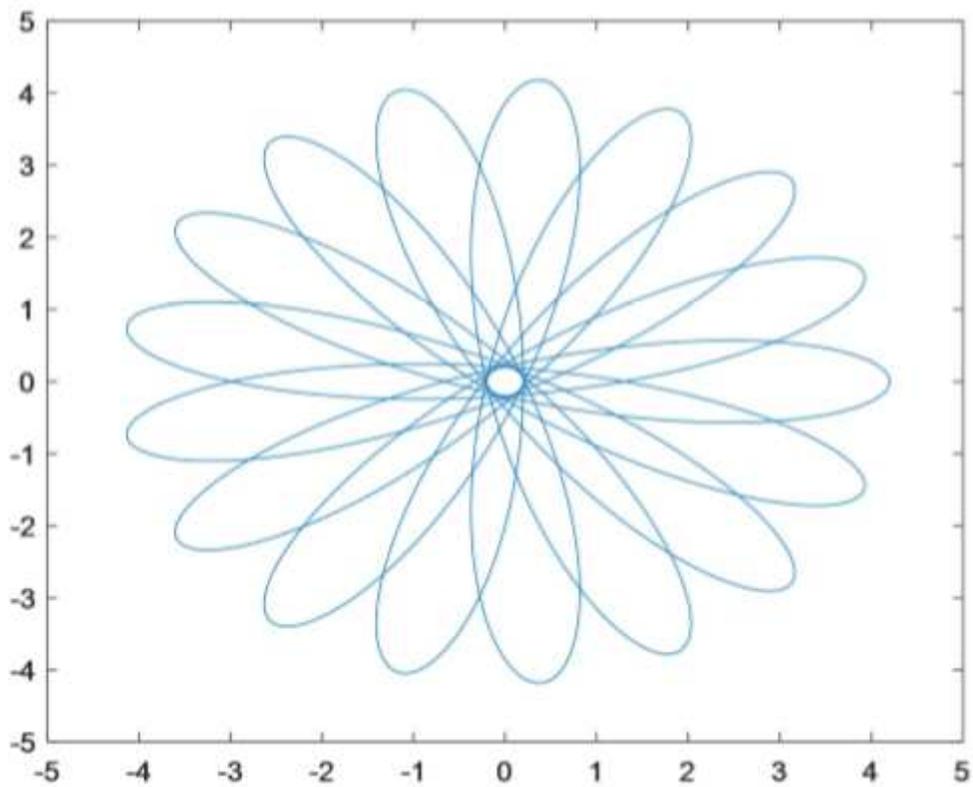


Figure 8

2D image of Hypotrochoid spiral

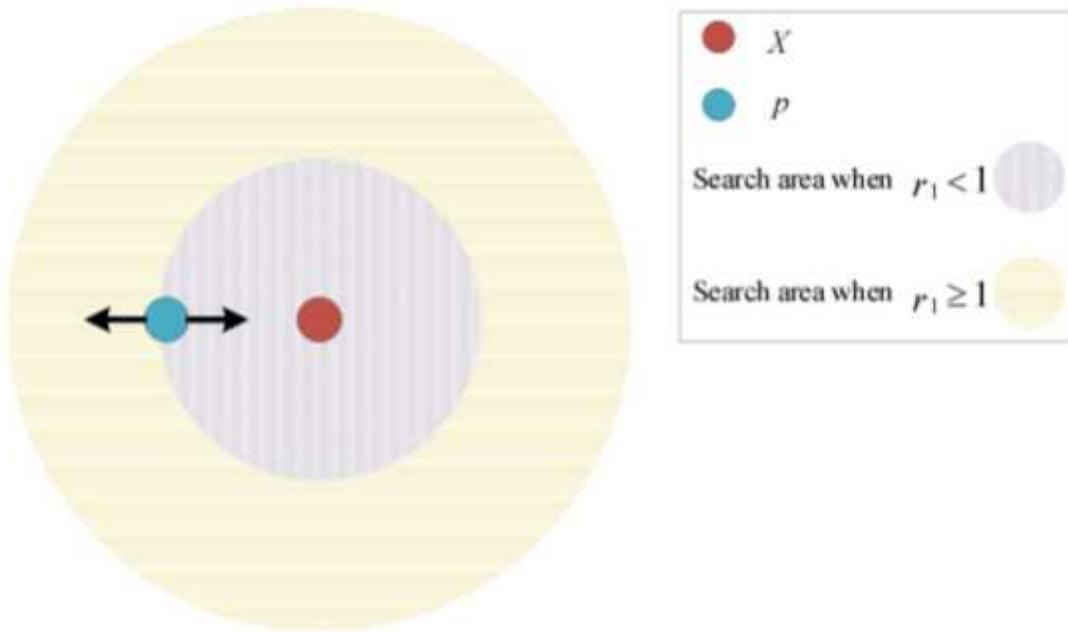


Figure 9

The model combining sine and cosine search

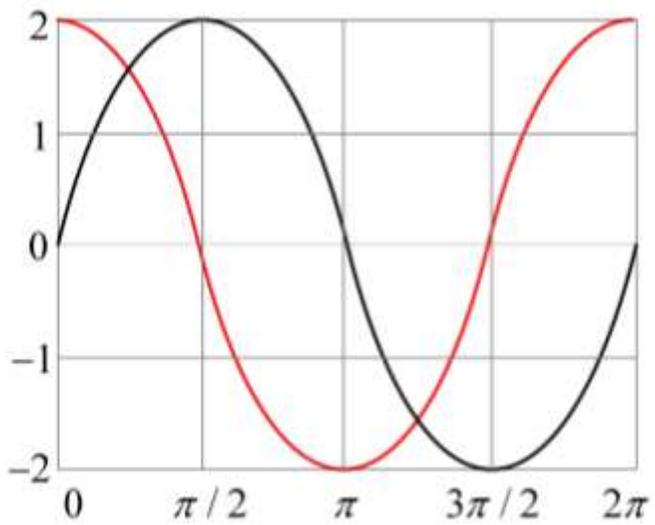


Figure 10

Sine and cosine function with range of $[-2, 2]$

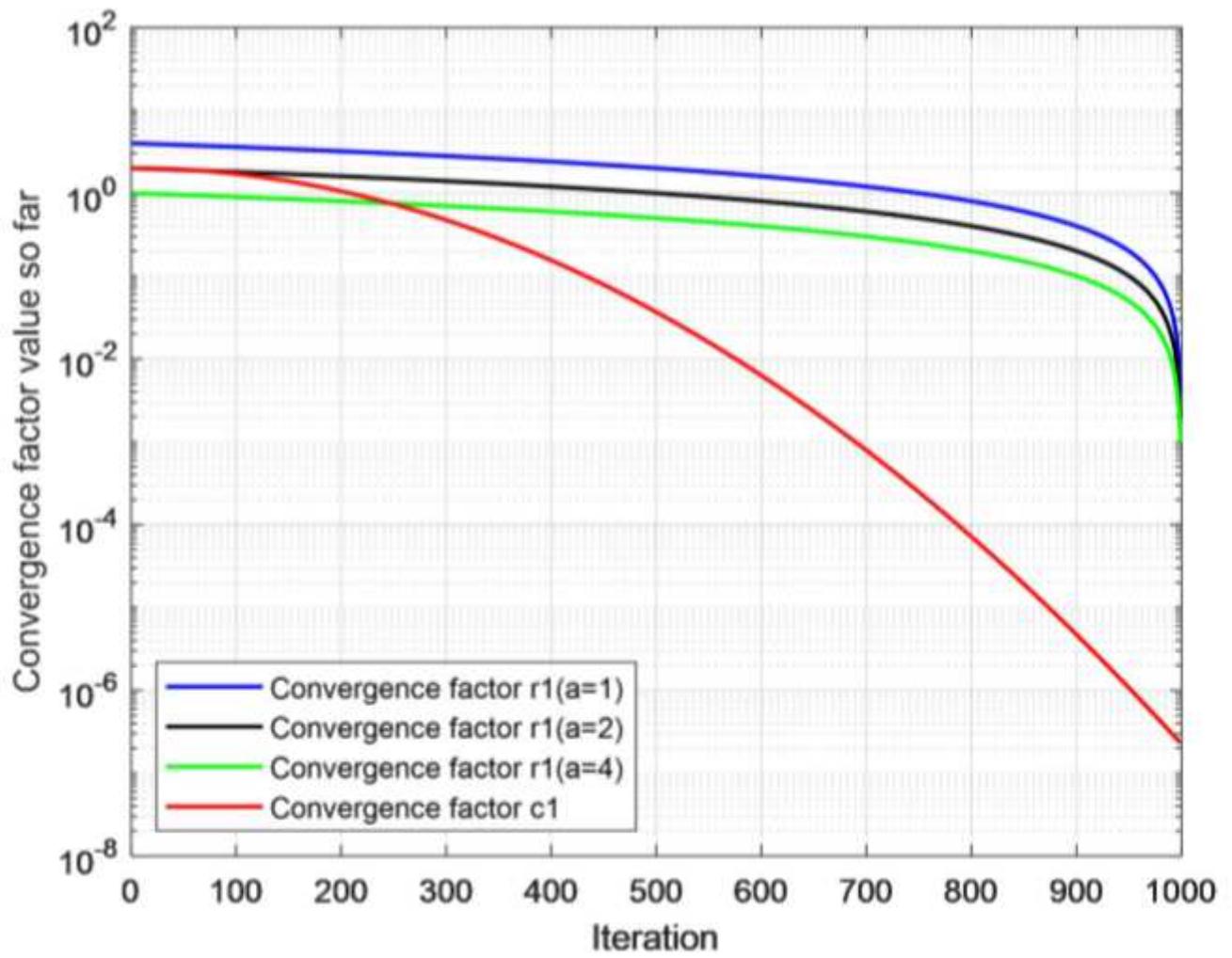


Figure 11

Comparison of convergence effects

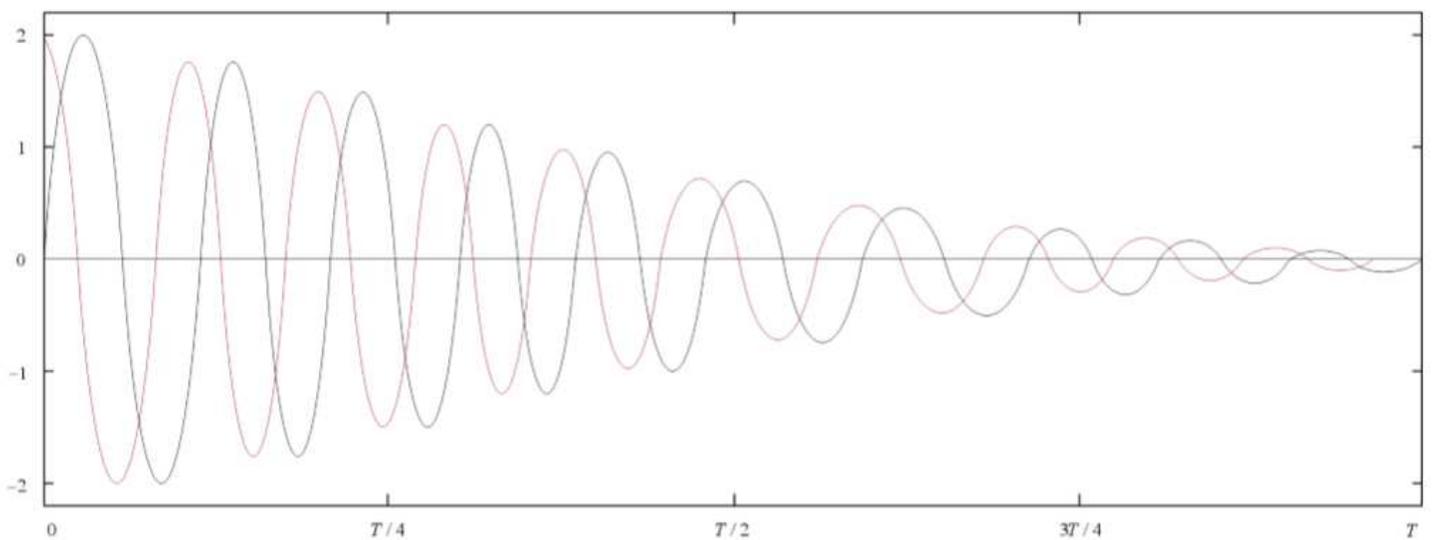


Figure 12

The model of sine and cosine function by reducing ranges

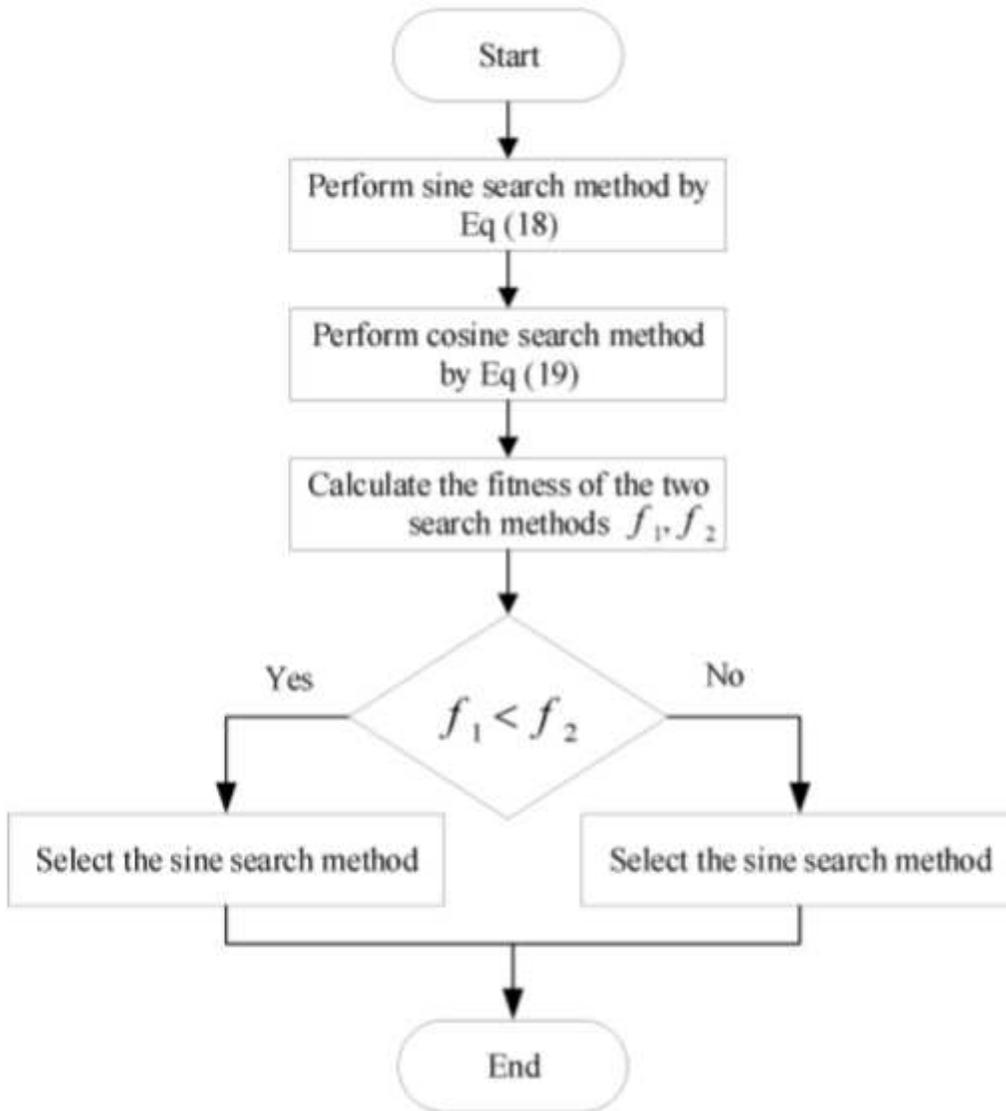


Figure 13

Flow chart of merit based strategy

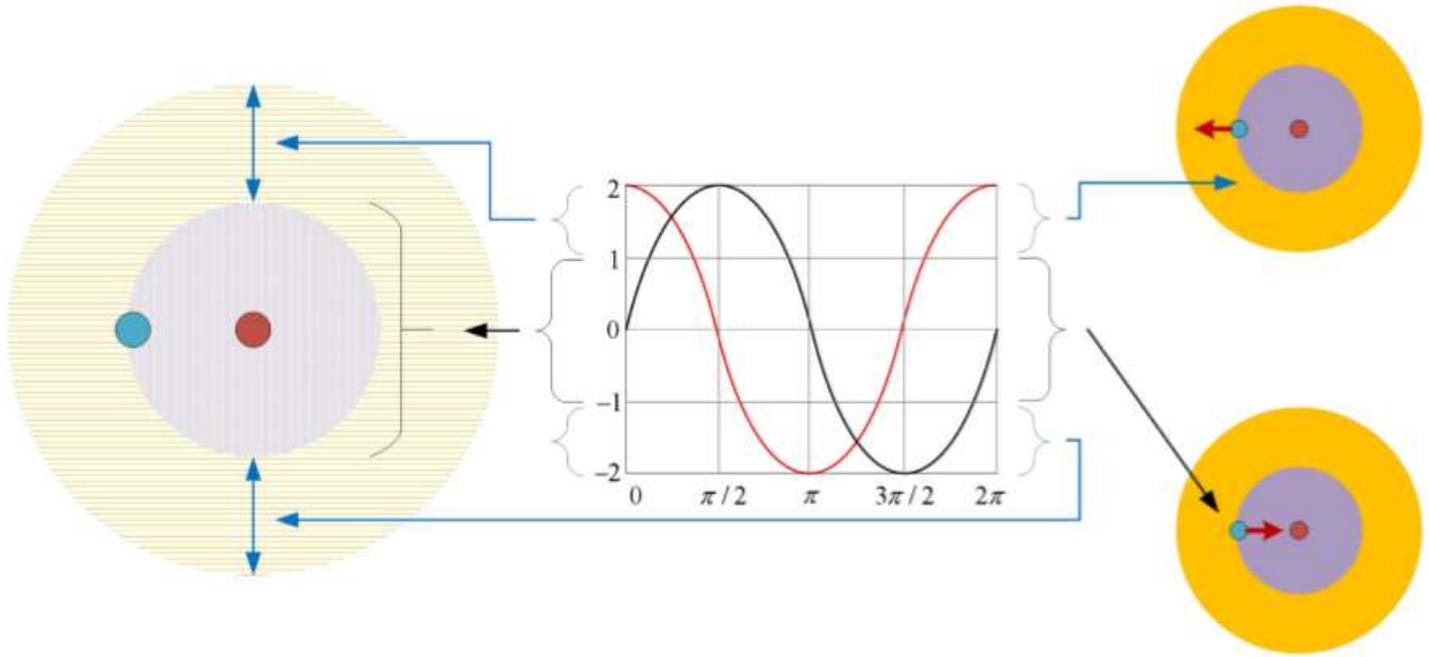


Figure 14

Effect of the function returned value on the search area

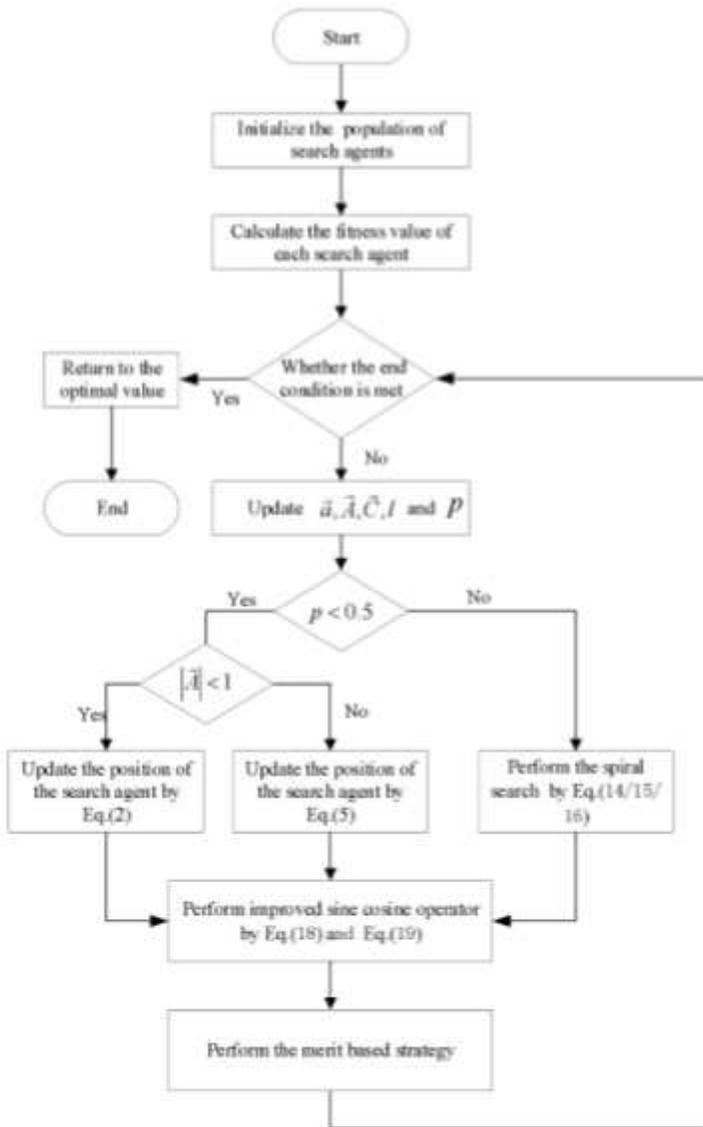


Figure 15

Flow chart of improved WOA

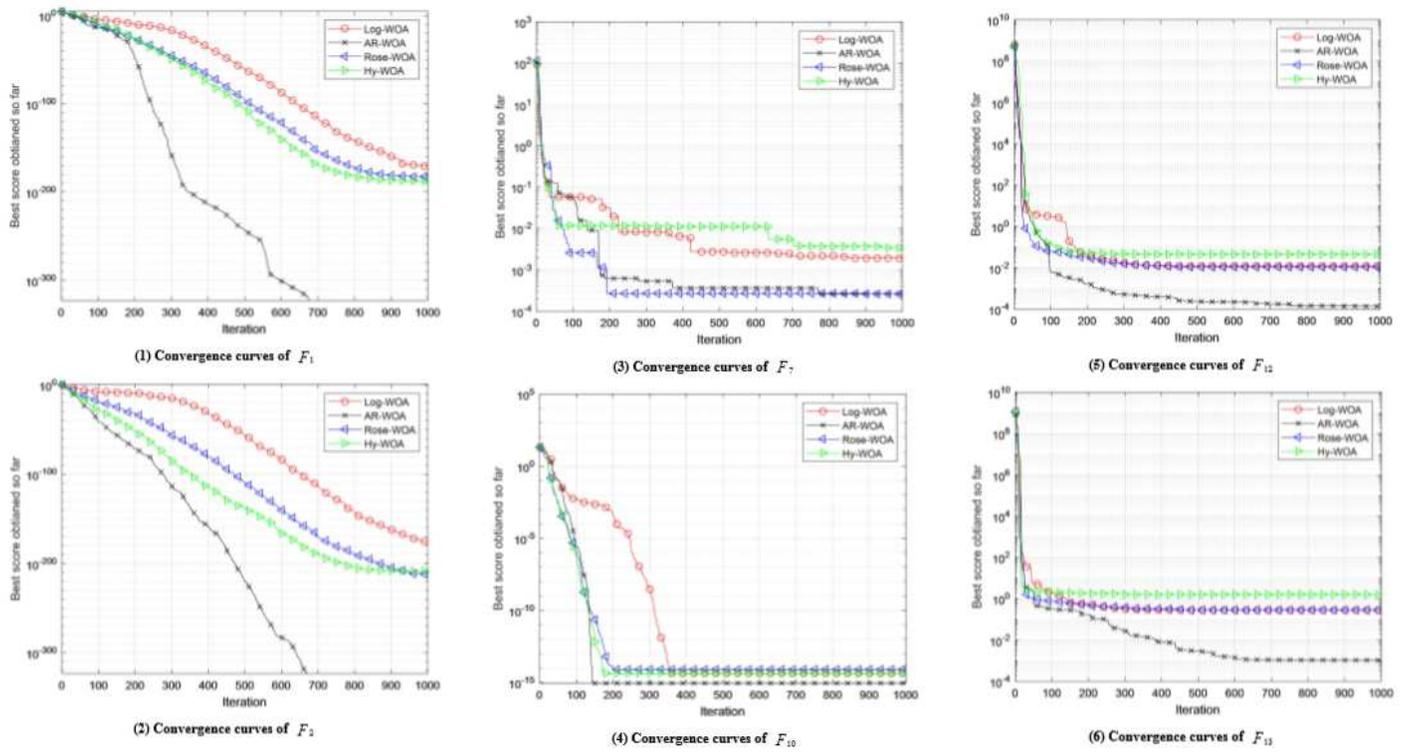


Figure 16

Simulation results

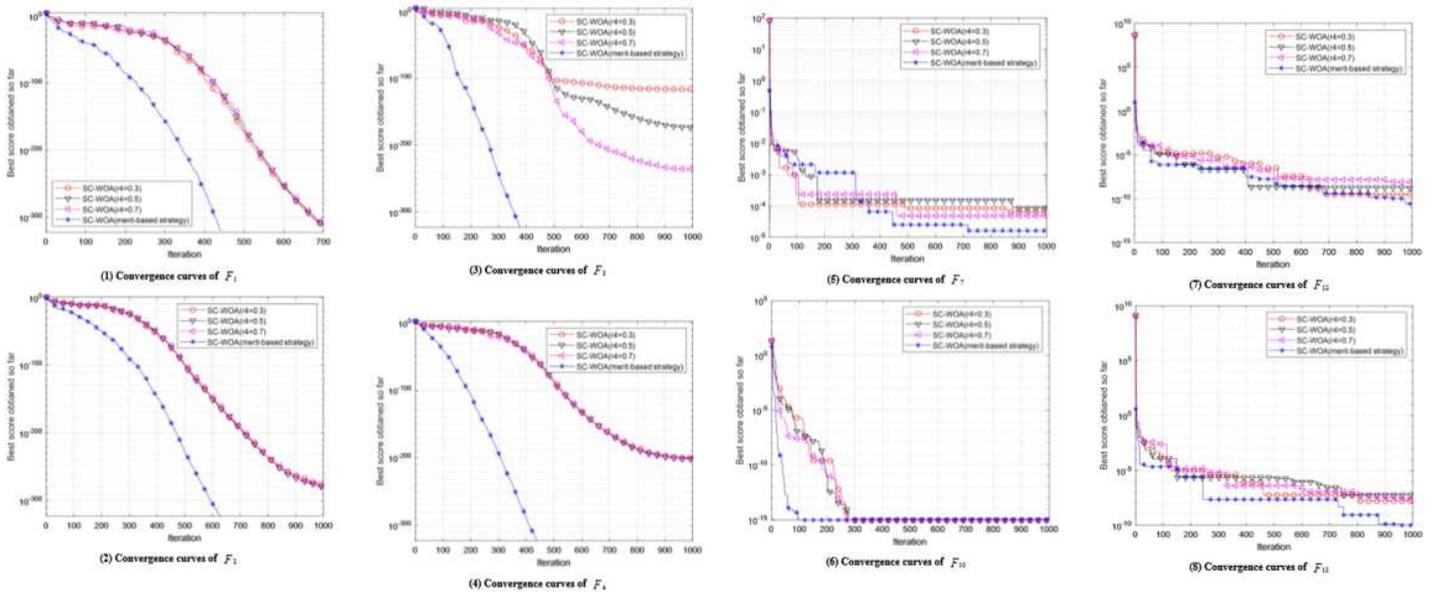


Figure 17

Simulation results

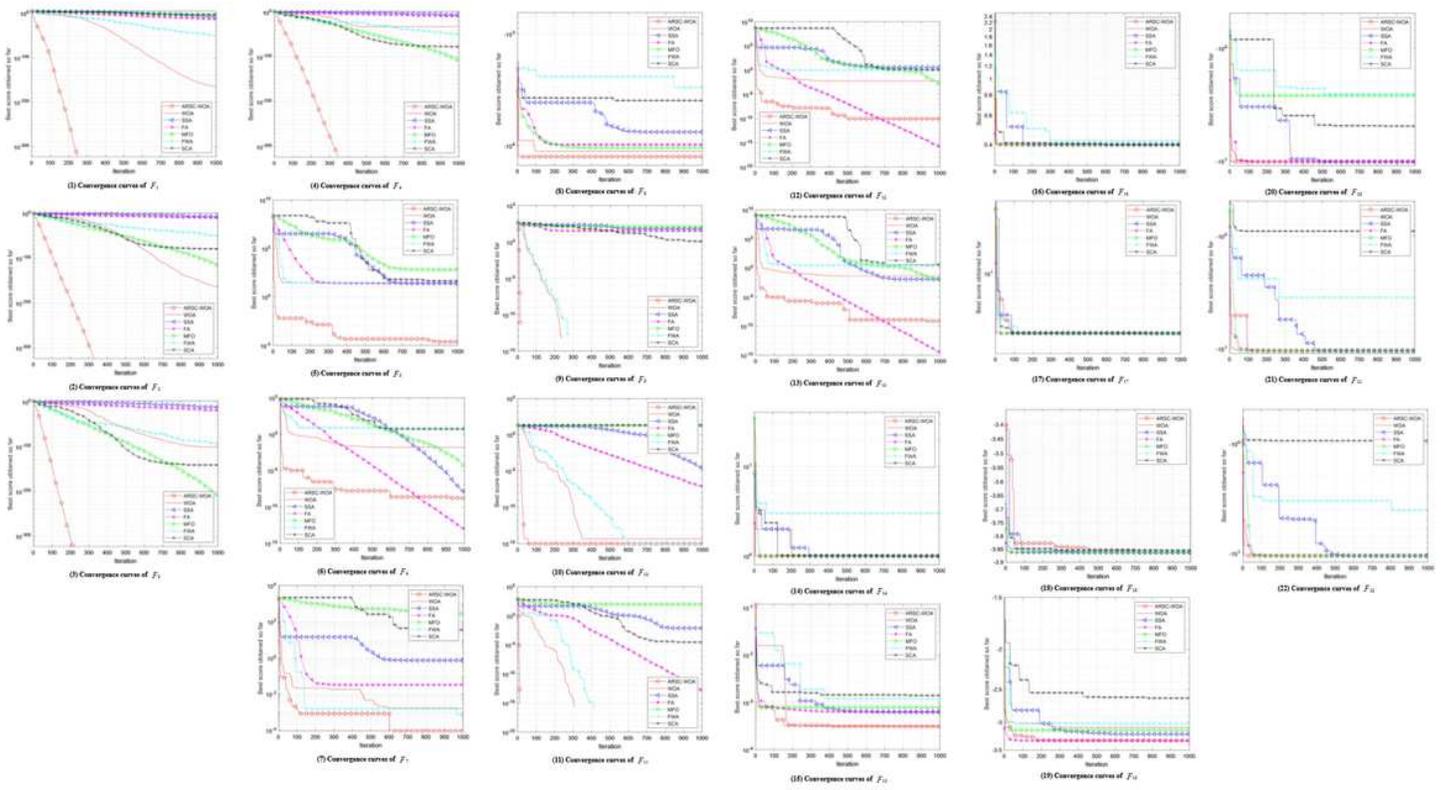


Figure 18

Simulation results

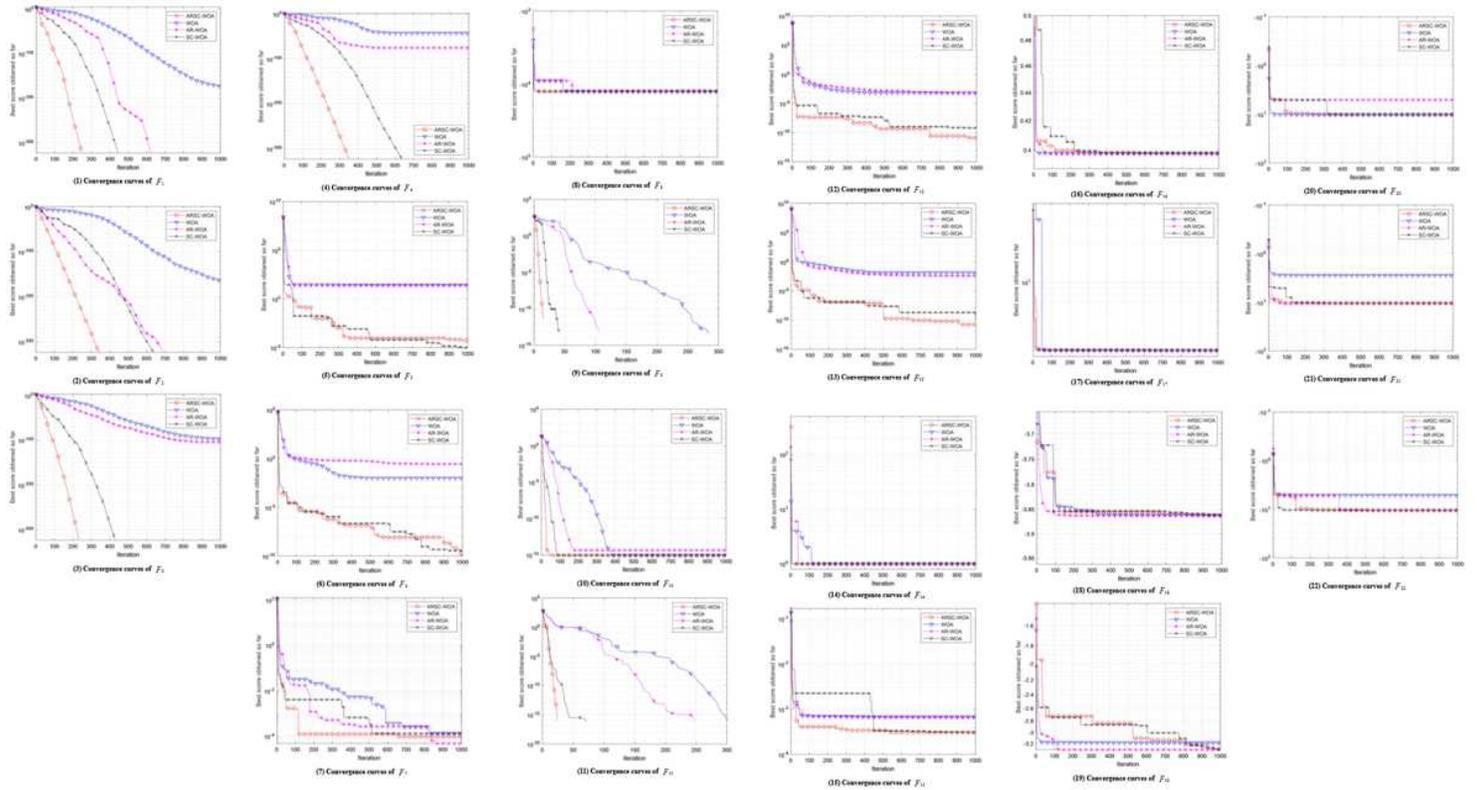


Figure 19

Simulation results

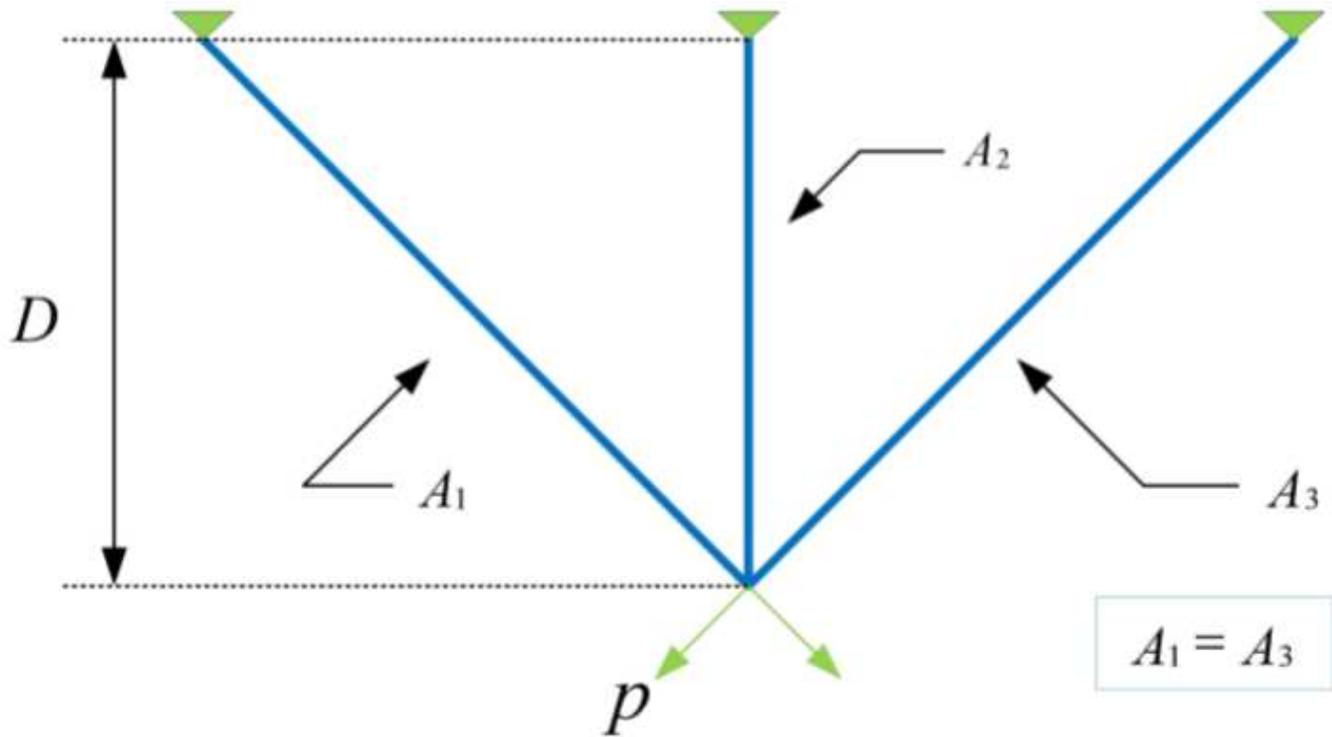


Figure 20

Three-bar truss design problem

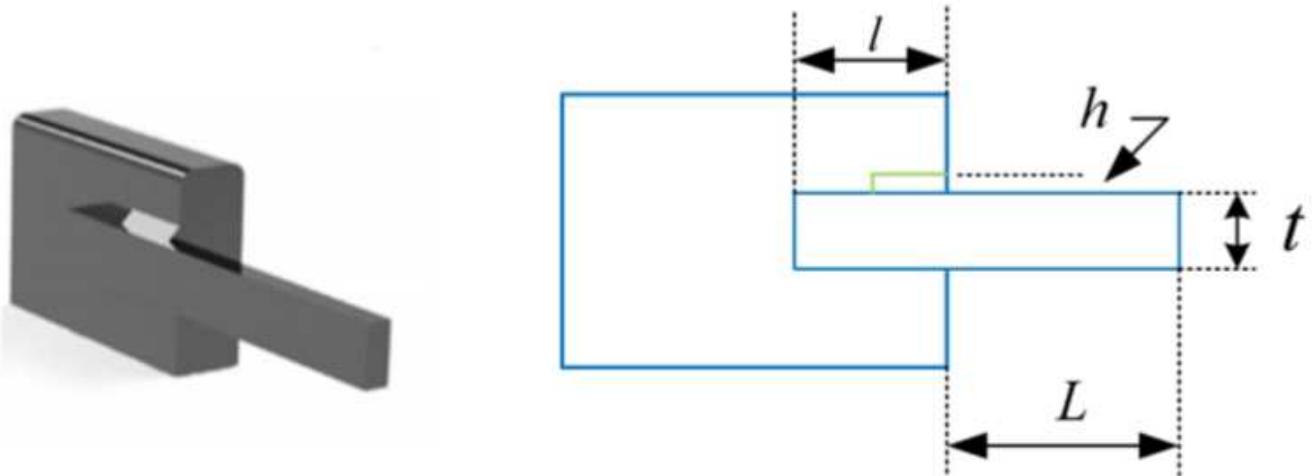


Figure 21

Welded beam optimization problem