



# Whale optimization algorithm based on dynamic pinhole imaging and adaptive strategy

Maodong Li<sup>1</sup> · Guanghui Xu<sup>1</sup> · Bo Fu<sup>1</sup> · Xilin Zhao<sup>1</sup>

Accepted: 25 September 2021 / Published online: 11 October 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

To solve the problems of premature convergence and easily falling into local optimum, a whale optimization algorithm based on dynamic pinhole imaging and adaptive strategy is proposed in this paper. In the exploitation phase, the dynamic pinhole imaging strategy allows the whale population to approach the optimal solution faster, thereby accelerating the convergence speed of the algorithm. In the exploration phase, adaptive inertial weights based on dynamic boundaries and dimensions can enrich the diversity of the population and balance the algorithm's exploitation and exploration capabilities. The local mutation mechanism can adjust the search range of the algorithm dynamically. The improved algorithm has been extensively tested in 20 well-known benchmark functions and four complex constrained engineering optimization problems, and compared with the ones of other improved algorithms presented in literatures. The test results show that the improved algorithm has faster convergence speed and higher convergence accuracy and can effectively jump out of the local optimum.

**Keywords** Whale optimization algorithm · Pinhole imaging · Dynamic boundary · Adaptive inertial weight · Local mutation

---

✉ Guanghui Xu  
xgh@hbut.edu.cn

Maodong Li  
limaodong2020@163.com

<sup>1</sup> Hubei Key Laboratory for High-Efficiency Utilization of Solar Energy and Operation Control of Energy Storage System, School of Electrical and Electronics Engineering, Hubei University of Technology, Wuhan 430068, China

## 1 Introduction

As a product of the combination of random algorithm and local search algorithm, the meta-heuristic algorithm proposes new feasible solutions for solving some complex engineering optimization problems. Compared with traditional algorithms, meta-heuristic algorithms search for spaces that may contain high-quality solutions by simulating natural or physical phenomena to give a better “global optimal solution.” When encountering difficult non-convex problems formed by complex engineering problems, the meta-heuristic algorithm is more effective than the gradient algorithm that is common in the past. This is mainly because of several major advantages of meta-heuristic algorithms: 1. The principle is simple and easy to implement; 2. Does not rely on gradient information; 3. It can be combined with different disciplines to solve diversified problems. 4. Able to give a better “global optimal solution.”

In recent years, with the continuous development of meta-heuristic algorithms, it can be divided into the following three categories. The first category is evolution-based algorithms. Evolutionary algorithms can simulate biological evolutionary behavior in nature. The most famous one is the genetic algorithm GA [1] proposed by learning Darwin’s biological evolution theory. In addition, there are genetic programming GP [2] and evolutionary strategy ES [3]. The second category is physics-based algorithms. Commonly used in this type of algorithms are the simulated annealing algorithm for learning natural annealing phenomena in nature [4], and the gravitational search algorithm GSA [5], which is inspired by Newton’s law of universal gravitation. Similar to these algorithms are the Moth–flame optimization algorithm MFO [6], the small world search algorithm WCA [7], the central force algorithm CFO [8] and the hydrological cycle algorithm WCA [9]. The third category is swarm-based algorithms. This type of algorithms is proposed after learning the clusters of biological populations and predation behavior in the natural world. For example, the particle swarm algorithm PSO [10] proposed by J. Kennedy and R. C. Eberhart et al. was inspired by the swarm behavior of birds. After decades of development, it has been widely used in different disciplines and achieved good results. Similar algorithms include gray wolf optimization algorithm GWO [11] inspired by wolf hunting behavior, sparrow search algorithm SSA [12], salvia colony algorithm SSA [13] and ant colony algorithm ACO [14].

Whale Optimization Algorithm (WOA) [15] is a swarm-based intelligent optimization algorithm proposed by Mirjalili et al in 2016. This algorithm simulates the unique bubble net predation behavior of humpback whales. Because of its simple structure and strong search capability, WOA is widely used in power economic dispatching, robot path planning, image segmentation, time series forecasting and other complex optimization problems. For example, Saha et al. [16] applied the improved WOA to the control of switched reluctance motors, and the test results showed that the algorithm has stable performance and reliability. Kong et al. [17] used the improved WOA to solve the complex parameter reduction problem, and the experimental results showed that the improved WOA has excellent performance. Sulaiman et al. [18] used the improved WOA to solve the PEHE multi-objective optimization

problem. The experimental results show that this scheme is better than other meta-heuristic algorithms.

At the same time, more in-depth researches and applications have also revealed the shortcomings of WOA itself. Aiming at the problems that WOA's convergence accuracy and convergence speed are insufficient, Gaganpreet Kaur et al. [19] proposed a way to combine WOA with chaos strategy. The test results show that the convergence speed of WOA has been improved to a certain extent. Coincidentally, Li et al. [20] also used the tent map in the chaotic map to initialize the population of WOA and achieved good results. These research contents not only put forward some improvement directions, but also show the importance of population initialization. In 2019, Hemasian-Etefagh et al. [21] proposed the idea of whale grouping to improve the convergence of WOA. Facts show that when WOA search methods and search capabilities are changed, WOA performance can be significantly improved. For example, Kaveh et al. [22] changed the original formula in WOA and applied it to specific engineering problems. The experimental results show that the performance of the improved WOA has been significantly improved. Along this research direction, more scholars have discovered the connection between the position update formula and population diversity. Ma et al. [23] introduced the inertia weight into the MFO, which effectively enriched the moth population and allowed the algorithm to maintain a higher search efficiency. At the same time, the introduction of inertia weight also effectively strengthens the adaptability of the algorithm itself. Qiu Xingguo et al. [24] used the sobel sequence, inertial weights and nonlinear strategies in the improvement of WOA. This scheme makes the initial population of WOA more abundant, so that the improved algorithm can continue to maintain a good search ability in the later stages of the iteration. At the same time, some scholars have also optimized WOA by introducing new content or combining WOA with other algorithms. Xiao Shuang et al. [25] combined Lévy flight with adaptive weights to increase the convergence speed of WOA, and achieved the best performance in comparison with other optimization schemes. Luo et al. [26] successfully introduced the differential mutation operator into WOA, and the addition of the mutation operator effectively improved the quality and reliability of solutions. Zhang et al. [27] combined WOA with the golden sine algorithm (GoldenSA) [28]. After extensive testing, they found that the golden sine operator can effectively accelerate the convergence speed of the algorithm. In general, these schemes have effectively improved the deficiencies of the algorithm to a certain extent. But in some typical non-convex problems, their performance is still very limited. Especially in some complex engineering problems, they still face the problem of premature convergence. In order to solve this problem effectively, this paper proposes a whale optimization algorithm based on dynamic pinhole imaging and adaptive strategies(DAWOA). The main contributions of this article are as follows:

- The theory of dynamic pinhole imaging is proposed and used to update the position of search agents. This strategy can adjust the position of the leader adaptively, allowing it to quickly approach the global optimal solution, thereby accelerating the convergence speed of the algorithm.

- Adaptive inertia weights based on dynamic boundaries and dimensions can adjust the algorithm in a feedback manner. It will adjust itself according to the location information of the individual population, so as to better balance the exploitation and the exploration capabilities of the algorithm.
- The local mutation strategy can dynamically adjust the shape of the logarithmic spiral according to the number of iterations, thereby changing the search range of the whale population. And, this can make the individual whale search state closer to the real whale.

At present, there are few studies on dynamic pinhole imaging and adaptive strategies in WOA algorithms. This analysis forms the basis of this work. The rest of the paper is structured as follows: Sect. 2 introduces the basic whale optimization algorithm. Section 3 discusses the three strategies proposed. Test problems and numerical results analysis are presented and discussed in Sects. 4 and 5, respectively. Section 6 is a summary of this paper and prospects for future work.

## 2 Principle and mathematical model of WOA

### 2.1 Shrink envelope

At this phase, the whale population will approach prey constantly. The number of whale populations is  $N$  and the dimension is  $d$ . The algorithm assumes that the position of the current optimal individual (search agent) is the global optimal position, and other whale individuals will use this as the target area to update their positions. The mathematical formula for this stage is:

$$D = \left| C\vec{X}^*(t) - X(t) \right| \quad (1)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - A \cdot \vec{D}, \quad (2)$$

where  $t$  is the current iteration number,  $A$  and  $C$  are coefficients.  $\vec{X}^*(t)$  is the current position vector of the best individual whale, and  $\vec{X}(t)$  is the position vector of the current whale. The mathematical model of coefficients  $A$  and  $C$  is as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (3)$$

$$C = 2r_2 \quad (4)$$

$$a = 2 - 2t/T_{\max}, \quad (5)$$

where  $r_1$  and  $r_2$  are random numbers in  $(0, 1)$ , the value of  $a$  will linearly decrease from 2 to 0 with the number of iterations.  $t$  is the current number of iterations, and  $T_{\max}$  is the maximum number of iterations.

## 2.2 Development phase

At this phase, the whale population will attack the prey. During the attack, the whale will continue to spirally update its position. Its mathematical model is as follows:

$$\vec{X}(t+1) = \vec{X}^*(t) + D_p e^{bl} \cos(2\pi l) \quad (6)$$

$$D_p = \left| \vec{X}^*(t) - \vec{X}(t) \right|, \quad (7)$$

where  $D_p$  is the distance between the whale and the prey,  $\vec{X}^*(t)$  is the current position vector of the best individual whale. It is worth noting that  $b$  is a constant, used to define the shape of the logarithmic spiral.  $l$  is a random number in  $(-1, 1)$ . In this link, the whale must continue to shrink and envelope while spiraling close to the prey. In this process, the whale population will repeat the behavior of the first stage. For this synchronous behavior model, in order to simulate the actual offensive process, assuming that the probability of the whale choosing the spiral update position is the same as the probability of choosing the shrink envelope, that is,  $p = 0.5$ . Then, the mathematical formula at this stage can be expressed as:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - A \cdot D & P < 0.5 \\ \vec{X}^*(t) + D_p e^{bl} \cos(2\pi l) & P \geq 0.5 \end{cases} \quad (8)$$

In the development stage, the value of  $A$  will change continuously with the change of the value of  $a$ . And, the value of  $a$  can be any value in  $[-1, 1]$ . Simply put, the individual whale can be in any position between the current position and the optimal position.

## 2.3 Search phase

At this stage, the whale population will conduct a large-scale search. The algorithm will randomly select a whale individual as the search agent to change the original predation area of the whale population. In this way, the whale population will conduct a more extensive search. Therefore, we only need to make small changes to Eq(1)(2) to get the mathematical model of this stage:

$$D = \left| C * \vec{X}_{\text{rand}} - \vec{X}(t) \right| \quad (9)$$

$$\vec{X}(t+1) = \vec{X}_{\text{rand}} - A \cdot D, \quad (10)$$

where  $\vec{X}_{\text{rand}}$  is the position vector of the randomly selected whale, which will be regarded as a new search agent. At this stage, due to  $|A| \geq 1$ , the whale population will be forced to leave its current location for a more extensive search. It is worth noting that the existence of this stage allows the algorithm to effectively avoid local optima.

### 3 whale optimization algorithm based on dynamic pinhole imaging and adaptive strategy

This section will give a detailed introduction to the three improvement strategies in the proposed improved algorithm (DAWOA).

#### 3.1 Dynamic pinhole imaging strategy

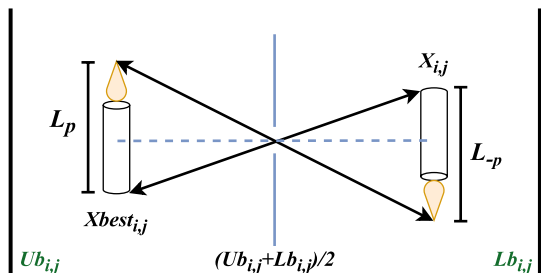
Generally speaking, the actual operation of an algorithm should be divided into two stages: particle divergence and particle shrinkage. Good population initialization can often bring faster convergence speed and higher convergence accuracy for the algorithm. If most of the population individuals can be initialized near the optimal solution, then in the particle shrinking phase of the algorithm, the population individuals only need a small amount of movement and search to reach the optimal value, which is undoubtedly a very ideal situation. In 2005, Tizhoosh proposed the Opposite-based learning theory [29], which revealed an efficient scheme for initializing the population. If the population individuals produce an opposite individual at the opposite position of the current position, the probability that the two individuals are closer to the optimal solution is 50%. Therefore, only a few operations are required to generate a higher quality population. This can be reminiscent of the theory of pinhole imaging in optics. Compared with ordinary Opposite-based learning, the pinhole imaging theory is more accurate and can produce more diversified points of opposition.

Figure 1 shows a typical theoretical model of pinhole imaging. Applying it to the search space of the population, the following mathematical model can be obtained:

$$\frac{X_{best_{i,j}} - (Ub_{i,j} + Lb_{i,j})/2}{(Ub_{i,j} + Lb_{i,j})/2 - X_{i,j}} = \frac{L_p}{L_{-p}}, \tag{11}$$

where  $X_{best_{i,j}}$  is the location of the current best individual (search agent),  $X_{i,j}$  is the opposite position in the theory of pinhole imaging.  $Ub_{i,j}$  and  $Lb_{i,j}$  are the dynamic boundaries of the  $i$ -th whale in the  $j$ -th dimension,  $L_p$  and  $L_{-p}$  are the length of the virtual candle in the current best position and the opposite position, respectively. It is worth noting that the position of the candle in the figure is also the position of the search agent, but the point representing the individual whale does not have an

Fig. 1 Dynamic pinhole imaging strategy



effective length. Therefore, the ratio of the two candles can be set as a variable  $K$ . From this, you can get:

$$X_{i,j} = \frac{(K + 1)(Ub_{i,j} + Lb_{i,j}) - 2Xbest_{i,j}}{2K} \quad (12)$$

It is not difficult to see from Eq.(12) that when two candles have the same length, this strategy evolves into a basic reverse learning strategy. Adjusting the value of  $K$  properly can change the position of the opposite point, allowing individual whales to have more search opportunities. In this paper,  $K$  is set to  $1.5 \times 10^4$ . In this way, every iteration, a new opposite point will be generated near the center line of the search space. When the opposite point is in a better position, its position will become a new boundary.

### 3.2 Adaptive strategy

When the algorithm is iterated to the later stage, the diversity of population individuals will decrease sharply. For the search space, the reduction of diversity will cause most of the population individuals to stagnate near the local optimal position. At this time, only a small number of whale populations are continuing to search, and the result is that the algorithm converges prematurely and falls into a local optimum. WOA is a swarm-based algorithm. Compared with an evolution-based algorithm, a swarm-based algorithm does not use a greedy strategy to discard poor values. WOA will retain the information of the previous generation for use in the next iteration. Therefore, an effective balance of algorithm's development and exploration capabilities is the key to enhancing the algorithm's search capabilities. To put it simply, introducing the inertia weight into the original WOA formula can effectively balance the local development capabilities and global search capabilities of WOA, thereby enriching the diversity of whale populations. This allows the algorithm to maintain a certain intensity of the search state in the later stages of the iteration, so as to avoid the population from falling into the local optimum. The updated Eqs. (8, 10) is as follows:

$$\vec{X}(t+1) = \begin{cases} \omega \cdot \vec{X} * (t) - A \cdot D & P < 0.5 \\ \omega \cdot \vec{X} * (t) + D_p e^{bl} \cos(2\pi l) & P \geq 0.5 \end{cases} \quad (13)$$

$$\vec{X}(t+1) = \omega \cdot \vec{X}_{rand} - A \cdot D, \quad (14)$$

where  $\omega$  is the adaptive inertia weight that can be adjusted with the number of iterations adaptively. The mathematical model is as follows:

$$\omega = \frac{C \times \sqrt{t}}{M} \times \left(1 - \frac{t}{T_{max}}\right) \quad (15)$$

$$M = \sqrt{H + N^2} \quad (16)$$

$$H = \frac{\sum_{j=1}^N (Ub_{i,j} - Lb_{i,j})^2}{t}, \quad (17)$$

where  $C = 0.1$ ,  $t$  is the current iteration number,  $T_{\max}$  is the maximum number of iterations, and  $N$  is the dimension. At the beginning of the iteration, almost only  $H$  and  $\sqrt{t}$  play a role. As the number of iterations progresses, when the position of the optimal solution is from farther to nearer, the value of  $M$  will decrease, and the value of  $\omega$  will increase at this time. This will allow the population individuals to escape from their current positions and conduct exploration with a larger search radius, thereby effectively enhancing the algorithm's extensive search capabilities to avoid falling into local optima. When the optimal solution is located farther away, the value of  $\omega$  will decrease relative to the previous iteration. As a result, individual whales will search in the high-quality solution space with a more precise spiral path to strengthen the local development capabilities of the algorithm. At the later stage of the iteration,  $H$  will hardly change again, and  $\omega$  at this time will become a steadily decreasing variable. This allows each individual whale to have a stronger search ability and a more diverse location. In fact, the purpose of this strategy is to tap the potential of individual populations and allow each individual whale to improve its own performance to varying degrees. In other words, as long as a small number of enhanced whales jump out of the local optimum successfully, they can lead the optimized population to the global optimum.

### 3.3 Local mutation strategy

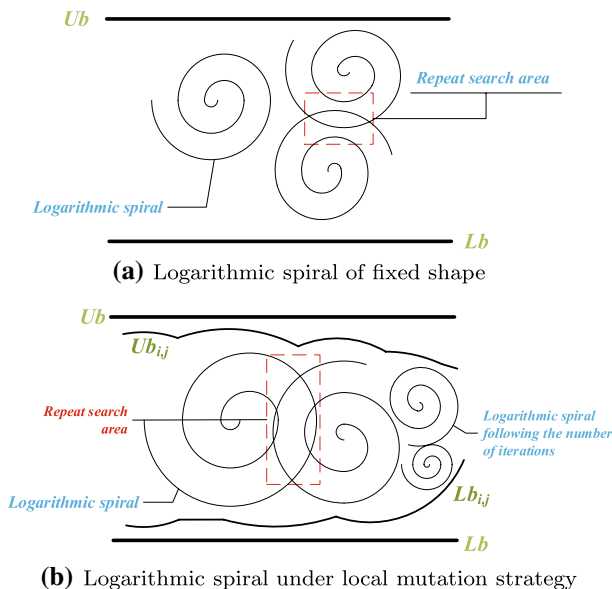
When a population individual finds a space that may contain high-quality solutions, using an effective strategy to approach the target area can not only help the algorithm quickly lock the optimal value, but also fully search this space. The advantage of this is that it can prevent the population individuals from gathering at a point in the space, which will cause the algorithm to stagnate. In Eq. (13), the parameter  $b$  can determine the shape of the logarithmic spiral. In other words,  $b$  determines the search path of the individual whale near the target area. Under normal circumstances,  $b$  is a fixed value, which will lead individual whales to perform local development in a fixed pattern in each iteration. When the better solution keeps a certain distance from the logarithmic spiral search path, these optimal values will lose their attractiveness to the whale population. Therefore, adding an appropriate mutation to  $b$  can help the individual whale better discover the global optimal solution that may exist in the surrounding space. This approach is similar to the actual hunting behavior of individual animals in nature. In the process of exploration, they will have a certain probability to deviate from the original path to develop possible prey because of the smell of the prey. The specific mathematical model is as follows:



$$b = \left(9 + \text{rand}\right)\left(1 - \frac{t}{T_{\max}}\right) \quad (18)$$

It is not difficult to see from Eq. (18) and Fig. 2 that as the iteration progresses, the search area will gradually shrink, so the logarithmic spiral will also continue to become smaller. Individuals of the population can perform more careful repeated searches in a limited space. When the iteration progresses to the later stage, the movement of the whale population will become very gradual, which can effectively increase the utilization of the current space. The mutation mechanism will give the population individuals more opportunities to explore possible solutions.

As a summary, the effective combination of the three strategies can bring the initial population closer to the optimal solution, and the richness and quality of the population will be higher. In the search process, adaptive inertial weights can help the algorithm find a dynamic balance point in exploration and development. The local mutation mechanism can adjust the whale's search range dynamically. As shown in Fig. 3, the original WOA will increase the possibility of falling into a local optimum due to insufficient population richness and too random initial position (Fig. 3a, b). The improved algorithm has richer population diversity and stronger search ability. The search agent in the current iteration process can also be initialized at a position closer to the optimal value (Fig. 3c, d). In this way, the algorithm can effectively jump out of the local optimal to find a better global optimal solution.



**Fig. 2** Local mutation strategy

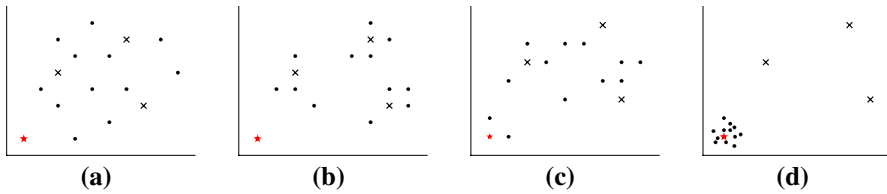


Fig. 3 DAWOA's search process

### 3.4 Analysis of DAWOA's computational complexity

The time complexity of the algorithm is the time resources consumed by the computer during the actual operation of the algorithm, which is an important reference for the actual performance of the algorithm. As far as DAWOA is concerned, the main factors that determine the time complexity are the total number of iterations ( $T$ ), population initialization, fitness sorting, three position update formulas, dynamic pinhole imaging strategy, adaptive inertial weighting and local mutation mechanism. Given that the number of populations is  $n$  and the dimension is  $D$ , then the time complexity for population initialization is  $O(n \times D)$ . Since the sorting process must be performed once in each iteration, its time complexity is  $O(T \times n^2)$ . Considering that the adaptive inertia weight and the WOA position update formula change together, the time complexity of the three formulas is  $O(T \times n \times D)$ . The dynamic pinhole imaging strategy needs to be compared and updated during each iteration. Therefore, its time complexity is  $O(T \times n \times D)$ . The local mutation strategy is equivalent to the successive change of parameter  $b$ , so its time complexity is  $O(T \times n \times D)$ . If the worst case is considered, the upper limit of the total time complexity of DAWOA is  $O(T \times (n^2 + n \times d))$ .

The pseudo-code of DAWOA algorithm is shown in Table 1.

## 4 Simulation experiment and numerical result analysis

20 well-known benchmark functions [30–33] composed of unimodal functions (F1–F7), multimodal functions (F8–F13) and fixed-dimensional functions (F14–F20) will be used to test the specifics of the improved algorithm Performance, some specific information of these test functions are shown in Table 2. In the test process, DAWOA will be used to compare the numerical results and convergence status with WOA, SSA, the recently proposed optimization algorithms EWOA [34] and MWOA [35].

### 4.1 Simulation test environment and evaluation standards

*Simulation test environment* The test environment for this experiment is as follows: Intel(R) core(TM) i5-6300HQ CPU @2.30 GHz 2.30 GHZ processor, 8G memory.

**Table 1** Pseudo code of DAWOA

---

Algorithm: The pseudo-code of DAWOA

---

```

01 begin
02   Initialize the population and determine
03   the search agent
04   While( $t < T_{\max}$ ) do
05     for  $i=1$  to  $N$  do
06       Local mutation strategy
07       Determine  $a$  and  $b$  according to Eq.(5)(18),
08       and determine the coefficients  $A$  and  $C$ 
09       according to Eq.(3) (4) (5)
10       Adaptive strategy
11       if( $p < 0.5$ )do
12         if( $|A| < 1$ ) do
13           Update the position of the current
14           individual according to Eq.(13)
15         else if ( $|A| \geq 1$ ) do
16           Randomly select an individual in the
17           population as the search agent, and
18           update the position with Eq.(14)
19         end if
20       else if ( $p \geq 0.5$ )do
21         Update the position according to Eq.(13)
22       end if
23     end for
24     Dynamic pinhole imaging strategy
25     Generate opposited-based individual
26     according to Eq.(12)
27     Calculate the fitness of the individual whale
28     population again, and update the search
29     agent position
30      $t = t + 1$ 
31   end while
32 end

```

---

The MATLAB version is MATLAB (2019b). During the test, the population size  $N$  is set to 30, and the maximum number of iterations  $T_{\max}$  is set to 500. In addition to the fixed dimension function, other test functions will be tested in the three dimensions of 10/30/50. Different algorithms will independently perform 30 tests of the same dimension in each test function and record the test results.

*Evaluation standards* In addition to introducing the mean (*Mean*) and standard deviation (*Std*) as the test criteria, this test will also introduce the Wilcoxon signed rank test [36] to compare the performance of each algorithm more reliably. Among the above detection standards, the average value test is used to detect the general performance and convergence accuracy of the algorithm; the standard deviation test is used to detect the overall stability of the algorithm. As a common nonparametric paired test, the Wilcoxon signed rank test is a fast and effective detection method.

**Table 2** Benchmark function

F	Function name	V-no	Range	Fmin
F1	Sphere	10/30/50	[- 100,100]	0
F2	Schwefel 2.22	10/30/50	[- 10,10]	0
F3	Schwefel 1.2	10/30/50	[- 100,100]	0
F4	Schwefel 2.21	10/30/50	[- 100,100]	0
F5	Rosenbrock	10/30/50	[- 30,30]	0
F6	Step	10/30/50	[- 100,100]	0
F7	Quartic	10/30/50	[- 1.28,1.28]	0
F8	Schwefel	10/30/50	[- 500,500]	- 418.9826D
F9	Rastrigin	10/30/50	[- - 5.12,5.12]	0
F10	Ackley	10/30/50	[- 32,32]	0
F11	Griewank	10/30/50	[- 600,600]	0
F12	Penalty1	10/30/50	[- 50,50]	0
F13	Penalty2	10/30/50	[- 50,50]	0
F14	Michalewicz	2	[- 65,65]	1
F15	Kowalik	4	[- 5,5]	0.0003
F16	Branin	3	[1,3]	0.398
F17	Hartmann 3D	6	[0,1]	- 3.86
F18	Shekel	4	[0,10]	- 10.1532
F19	Shekel	4	[0,10]	- 10.4028
F20	Shekel	4	[0,10]	- 10.5363

It only needs to provide a small number of samples for reliable numerical analysis. The rules of the inspection method are very simple. In the numerical test results, if the target algorithm is better than the comparison algorithm, it is recorded as  $R_+$ ; otherwise, it is recorded as  $R_-$ . Generally, speaking, the greater the number of  $R_+$ , the greater the advantage of the target algorithm. In order to facilitate recording and comparison, in the table, “+” will be used to indicate that the target algorithm is better than the comparison algorithm, “-” is used to indicate that the target algorithm is inferior to the comparison algorithm, and “=” indicates that the two algorithms are in the same degree of superiority and inferiority, and  $Gm$  represents the difference between the number of “+” and the number of “-”.

*Experimental parameter setting* In the comparative experiment in this chapter, the parameter settings of each algorithm are shown in Table 3.

## 4.2 Unimodal function test and result analysis

Unimodal test functions are mainly complex spherical or valley-shaped numerical problems. They only have a global minimum but are difficult to find. Therefore, it can be used to test the exploitation capabilities of the algorithm. As shown in Tables 4, 5, and 6, among the three dimensions, DAWOA almost occupies a dominant position in all unimodal functions, showing strong competitiveness. For the test functions F1–F4, although the convergence speed of MWOA is faster than DAWOA

**Table 3** Experimental parameter setting

Parameter	Optimization			
	This paper	WOA	MWOA	EWOA
$a$	[0, 2]	[0, 2]	[0, 2]	[0, 2]
$b$	N/A	1	1	1
$K$	15000	N/A	N/A	N/A
$l$	[-1, 1]	[-1, 1]	[-1, 1]	[-1, 1]
$\sigma$	N/A	N/A	0.1	N/A
$n$	N/A	N/A	12000	N/A
$b_1/b_2$	N/A	N/A	1/2	N/A
$r_1/r_2$	[0, 1]	[0, 1]	[0, 1]	[0, 1]

in every dimension, DAWOA has more obvious advantages in the remaining unimodal functions. For example, the test function F5 (Rosenbrock), which is a typical non-convex function, requires complex numerical calculations to solve it. The global minimum of the Rosenbrock function is located in a narrow parabolic valley. Although this valley is easy to find, it is difficult to converge to the lowest point. As shown in Figs. 4, 5, and 6, the performance of WOA, MWOA, EWOA and SSA in the test function F5 are all unsatisfactory. However, the improved DAWOA can reduce the numerical result to 4 decimal places, which is undoubtedly a big improvement. When the dimensions are 10 and 30, DAWOA's performance in F6 is second only to SSA. But when the dimension rose to 50, DAWOA recovered its own advantages. As shown in Fig. 4, during the convergence process, there were several cliff-like declines in the curve. This shows that the dynamic pinhole imaging strategy has been trying to adjust the position of the leader to help it quickly approach the optimal area. At the same time, adaptive inertia weights are also helping the population to maintain an efficient search state, thereby accelerating the convergence of the algorithm. The above conclusion fully shows that the local development capability of the algorithm has been significantly improved.

### 4.3 Multimodal function test and result analysis

Different from the unimodal functions, the multimodal functions have many local optimal values. Moreover, the scale of this type of problem will increase exponentially as the dimensionality increases. Therefore, the multimodal function can effectively test the exploration capabilities of the algorithm. In combination with Tables 4, 5, and 6, it can be seen that DAWOA performs significantly better than WOA, EWOA, MWOA and SSA in multimodal functions. For the test function F8, this is a typical multimodal function with a large number of local optimal values, and its shape is similar to a hill. Compared with WOA, EWOA and SSA, DAWOA's numerical results in F8 have made a big leap. Although the test result of MWOA is very close to the theoretical value, the convergence accuracy of DAWOA is significantly higher. This is because the addition of adaptive inertial weights increases the diversity of the whale population, allowing population individuals to be evenly

**Table 4** Comparison of test results of different algorithms

F	This paper	WOA	MWOA	EWOA	SSA
<i>F1</i>					
Mean	<b>0.000E+00</b>	6.214E-72	<b>0.000E+00</b>	9.527E-144	1.631E-07
Std	<b>0.000E+00</b>	3.334E-71	<b>0.000E+00</b>	3.152E-143	1.560E-07
Wilcoxon		+	=	+	+
<i>F2</i>					
Mean	<b>0.000E+00</b>	1.365E-45	<b>0.000E+00</b>	2.078E-81	2.497E+00
Std	<b>0.000E+00</b>	7.383E-45	<b>0.000E+00</b>	8.658E-81	1.543E+00
Wilcoxon		+	=	+	+
<i>F3</i>					
Mean	<b>0.000E+00</b>	5.347E+04	<b>0.000E+00</b>	4.306E-07	1.720E+03
Std	<b>0.000E+00</b>	1.519E+04	<b>0.000E+00</b>	1.306E-06	1.047E+03
Wilcoxon		+	=	+	+
<i>F4</i>					
Mean	<b>0.000E+00</b>	3.991E+01	<b>0.000E+00</b>	8.945E-46	1.186E+01
Std	<b>0.000E+00</b>	3.178E+01	<b>0.000E+00</b>	4.899E-45	3.811E+00
Wilcoxon		+	=	+	+
<i>F5</i>					
Mean	<b>5.762E-04</b>	2.785E+01	2.602E+01	2.768E+01	1.569E+02
Std	<b>1.078E-03</b>	4.963E-01	7.005E+00	5.608E-01	2.402E+02
Wilcoxon		+	+	+	+
<i>F6</i>					
Mean	1.671E-05	4.220E-01	1.998E-01	8.559E-01	<b>1.376E-07</b>
Std	2.291E-05	1.946E-01	7.153E-02	5.107E-01	<b>2.019E-07</b>
Wilcoxon		+	+	+	-
<i>F7</i>					
Mean	<b>4.187E-05</b>	3.528E-03	7.039E-05	3.001E-03	1.861E-01
Std	<b>2.788E-05</b>	3.342E-03	6.947E-05	2.680E-03	5.669E-02
Wilcoxon		+	+	+	+
<i>F8</i>					
Mean	<b>-1.257E+04</b>	-8.951E+03	-1.243E+04	-7.583E+03	-7.505E+03
Std	<b>9.056E-01</b>	1.329E+03	3.005E+02	9.403E+02	8.894E+02
Wilcoxon		+	+	+	+
<i>F9</i>					
Mean	<b>0.000E+00</b>	<b>0.000E+00</b>	<b>0.000E+00</b>	<b>0.000E+00</b>	5.801E+01
Std	<b>0.000E+00</b>	<b>0.000E+00</b>	<b>0.000E+00</b>	<b>0.000E+00</b>	1.807E+01
Wilcoxon		=	=	=	+
<i>F10</i>					
Mean	<b>8.882E-16</b>	3.967E-15	<b>8.882E-16</b>	3.375E-15	2.432E+00
Std	<b>4.012E-31</b>	2.421E-15	<b>4.012E-31</b>	1.656E-15	1.238E+00
Wilcoxon		+	=	+	+
<i>F11</i>					
Mean	<b>0.000E+00</b>	2.569E-02	<b>0.000E+00</b>	6.145E-03	1.583E-02

**Table 4** (continued)

F	This paper	WOA	MWOA	EWOA	SSA
Std	<b>0.000E+00</b>	6.881E-02	<b>0.000E+00</b>	2.355E-02	1.409E-02
Wilcoxon		+	=	+	+
<i>F12</i>					
Mean	<b>9.975E-07</b>	2.150E-02	8.582E-03	3.091E-02	6.989E+00
Std	<b>1.938E-06</b>	1.742E-02	5.419E-03	1.604E-02	2.896E+00
Wilcoxon		+	+	+	+
<i>F13</i>					
Mean	<b>1.247E-05</b>	4.792E-01	1.451E-01	8.504E-01	3.264E-03
Std	<b>1.859E-05</b>	2.196E-01	6.744E-02	3.719E-01	5.775E-03
Wilcoxon		+	+	+	+
<i>F14</i>					
Mean	<b>9.980E-01</b>	2.899E+00	1.622E+00	4.786E+00	1.097E+00
Std	<b>1.826E-06</b>	2.993E+00	2.000E+00	4.146E+00	3.033E-01
Wilcoxon		+	+	+	+
<i>F15</i>					
Mean	<b>3.323E-04</b>	7.660E-04	4.980E-04	6.573E-04	3.534E-03
Std	<b>1.834E-05</b>	4.753E-04	1.411E-04	2.557E-04	6.720E-03
Wilcoxon		+	+	+	+
<i>F16</i>					
Mean	3.980E-01	3.980E-01	3.982E-01	<b>3.980E-01</b>	-
Std	1.224E-04	1.694E-16	3.952E-04	<b>1.694E-16</b>	-
Wilcoxon		+	+	-	+
<i>F17</i>					
Mean	<b>- 3.859E+00</b>	- 3.826E+00	- 3.840E+00	- 3.857E+00	- 3.863E+00
Std	<b>2.724E-03</b>	1.403E-01	1.297E-02	9.939E-03	3.162E-15
Wilcoxon		+	+	+	+
<i>F18</i>					
Mean	<b>- 1.015E+01</b>	- 7.297E+00	- 1.008E+01	- 6.512E+00	- 6.811E+00
Std	<b>2.782E-03</b>	2.965E+00	1.510E-01	2.715E+00	3.498E+00
Wilcoxon		+	+	+	+
<i>F19</i>					
Mean	<b>- 1.040E+01</b>	- 7.420E+00	- 1.029E+01	- 7.456E+00	- 9.063E+00
Std	<b>2.300E-03</b>	3.038E+00	2.036E-01	2.783E+00	2.767E+00
Wilcoxon		+	+	+	+
<i>F20</i>					
Mean	<b>- 1.053E+01</b>	- 7.755E+00	- 1.046E+01	- 7.807E+00	- 8.134E+00
Std	<b>5.992E-03</b>	3.507E+00	1.039E-01	3.062E+00	3.504E+00
Wilcoxon		+	+	+	+
+/-!/=gm					
69/9/2/67					

The bolded results indicate the best results in the same group

“+” indicates that the proposed algorithm is better than the algorithm used for comparison

“-” indicates that the proposed algorithm is worse than the one used for comparison Algorithm

“=” indicates that the performance of the proposed algorithm is equivalent to that of the algorithm used for comparison

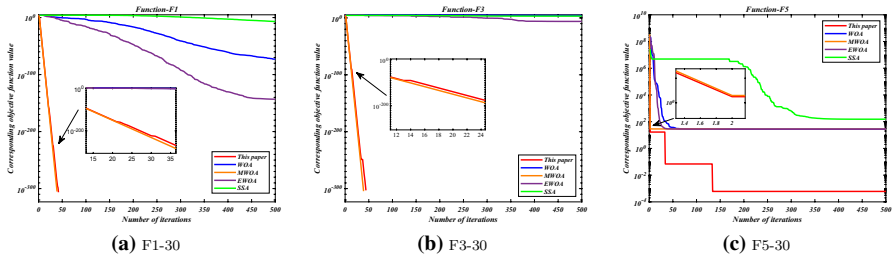


Fig. 4 Convergence curve comparison of some unimodal test functions (30 dimensions)

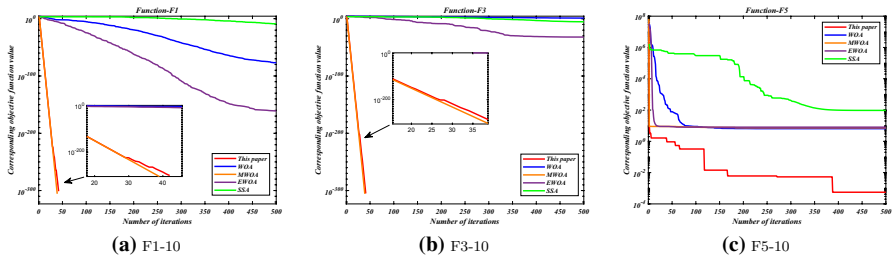


Fig. 5 Convergence curve comparison of some unimodal test functions (10 dimensions)

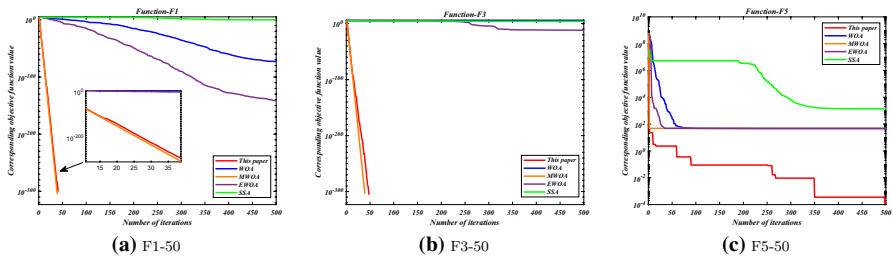


Fig. 6 Convergence curve comparison of some unimodal test functions (50 dimensions)

distributed in the solution space, thereby effectively avoiding local optima. Figs. 7, 8, and 9 show the convergence curves of several algorithms in functions F8, F11 and F12. In the early iteration of the algorithm, the dynamic aperture imaging strategy is playing an active role. The algorithm quickly approaches the target area, and the convergence curve drops sharply. As the number of iterations progressed, the local mutation strategy began to operate. After a short search, the whale population avoided the many local best points in the solution space and successfully reached the target area. When the dimensionality gradually increases, the complexity of the search space will increase significantly. It is not difficult to see from Table 5 and Table 6 that the increase in dimensions has not affected DAWOA's dominant position, and the collaborative work of the three strategies has helped it open up a



**Table 5** Comparison of the numerical results of unimodal and multimodal functions (10 dimensions)

F	This paper	WOA	MWOA	EWOA	SSA
<i>F1</i>					
Mean	<b>0.000E+00</b>	3.734E-63	<b>0.000E+00</b>	8.378E-150	9.065E-10
Std	<b>0.000E+00</b>	1.181E-62	<b>0.000E+00</b>	4.319E-149	3.467E-10
Wilcoxon		+	=	+	+
<i>F2</i>					
Mean	<b>0.000E+00</b>	3.891E-42	<b>0.000E+00</b>	1.163E-85	1.389E-02
Std	<b>0.000E+00</b>	1.448E-41	<b>0.000E+00</b>	3.980E-85	6.534E-02
Wilcoxon		+	=	+	+
<i>F3</i>					
Mean	<b>0.000E+00</b>	1.080E+02	<b>0.000E+00</b>	8.009E-30	4.423E-06
Std	<b>0.000E+00</b>	1.686E+02	<b>0.000E+00</b>	4.387E-29	1.627E-05
Wilcoxon		+	=	+	+
<i>F4</i>					
Mean	<b>0.000E+00</b>	1.058E+01	<b>0.000E+00</b>	5.892E-54	1.993E-05
Std	<b>0.000E+00</b>	1.084E+01	<b>0.000E+00</b>	2.015E-53	8.361E-06
Wilcoxon		+	=	+	+
<i>F5</i>					
Mean	<b>7.409E-04</b>	7.332E+00	7.245E+00	6.850E+00	9.828E+01
Std	<b>1.400E-03</b>	4.201E-01	1.400E+00	4.252E-01	2.002E+02
Wilcoxon		+	+	+	+
<i>F6</i>					
Mean	1.457E-05	4.862E-03	1.113E-02	2.950E-04	<b>9.859E-10</b>
Std	2.517E-05	2.449E-03	6.378E-03	5.145E-04	<b>2.519E-10</b>
Wilcoxon		+	+	+	-
<i>F7</i>					
Mean	<b>4.151E-05</b>	2.689E-03	7.305E-05	2.565E-03	1.446E-02
Std	<b>2.620E-05</b>	2.251E-03	7.913E-05	3.108E-03	9.910E-03
Wilcoxon		+	+	+	+
<i>F8</i>					
Mean	<b>-4.189E+03</b>	- 3.257E+03	- 4.154E+03	- 2.681E+03	- 2.754E+03
Std	<b>9.520E-01</b>	4.762E+02	1.329E+02	3.830E+02	3.417E+02
Wilcoxon		+	+	+	+
<i>F9</i>					
Mean	<b>0.000E+00</b>	4.737E-16	<b>0.000E+00</b>	6.384E+00	1.735E+01
Std	<b>0.000E+00</b>	2.595E-15	<b>0.000E+00</b>	1.535E+01	9.585E+00
Wilcoxon		+	=	+	+
<i>F10</i>					
Mean	<b>8.882E-16</b>	5.862E-15	<b>8.882E-16</b>	3.138E-15	7.551E-01
Std	<b>4.012E-31</b>	1.770E-15	<b>4.012E-31</b>	1.741E-15	9.430E-01
Wilcoxon		+	=	+	+
<i>F11</i>					
Mean	<b>0.000E+00</b>	4.132E-02	<b>0.000E+00</b>	1.185E-02	2.317E-01

**Table 5** (continued)

F	This paper	WOA	MWOA	EWOA	SSA
Std	<b>0.000E+00</b>	9.118E-02	<b>0.000E+00</b>	5.486E-02	1.388E-01
Wilcoxon		+	=	+	+
<i>F12</i>					
Mean	<b>1.166E-05</b>	4.619E-02	3.003E-03	1.674E-02	7.386E-01
Std	<b>1.892E-05</b>	2.042E-01	2.029E-03	2.655E-02	8.749E-01
Wilcoxon		+	+	+	+
<i>F13</i>					
Mean	<b>2.320E-05</b>	2.449E-02	1.845E-02	1.252E-01	1.696E-01
Std	<b>2.809E-05</b>	1.636E-02	1.325E-02	9.706E-02	9.123E-01
Wilcoxon		+	+	+	+
+/-/=gm					
44/7/1/43					

The bolded results indicate the best results in the same group

“+” indicates that the proposed algorithm is better than the algorithm used for comparison

“-” indicates that the proposed algorithm is worse than the one used for comparison Algorithm

“=” indicates that the performance of the proposed algorithm is equivalent to that of the algorithm used for comparison

broader search space. The above results fully demonstrate that the improved algorithm has good extensive search capabilities.

#### 4.4 Fixed dimension function test and result analysis

Fixed-dimensional functions are a type of complex multimodal functions with specific dimensional requirements. This type of problem can effectively test the overall performance of the algorithm, especially the ability to jump out of the local optimum. As far as the test function itself is concerned, most of the global optimal values of fixed-dimensional functions are not 0. This has the advantage of making the type of unknown solution space more comprehensive, and it also provides a more reliable test result and numerical basis for the specific performance test of the algorithm. It can be seen from Table 4 that so far, DAWOA has achieved the best performance in most of the test functions. In the test function F16 (Branin function), the improved algorithm is slightly worse than EWOA, but the numerical results obtained by DAWOA are already very close to the theoretical values. It can be seen from Fig. 10 that in the test functions F15, F17, F18, F19 and F20, after the improved algorithm quickly approximates the region where the optimal value is located, the population individuals do not gather near a local optimal value. The local mutation mechanism allows them to continue to maintain an efficient search state after reaching the local optimal area, which allows some whale individuals to perceive the location of a better solution successfully. In the test functions F15, F18, F19, F20, there are several obvious gradient drops in the convergence curve, which shows that the adaptive inertia

**Table 6** Comparison of the numerical results of unimodal and multimodal functions (50 dimensions)

F	This paper	WOA	MWOA	EWOA	SSA
<i>F1</i>					
Mean	<b>0.000E + 00</b>	3.261E-60	<b>0.000E + 00</b>	6.728E-141	9.756E-01
Std	<b>0.000E + 00</b>	1.604E-59	<b>0.000E + 00</b>	2.974E-140	1.390E + 00
Wilcoxon		+	=	+	+
<i>F2</i>					
Mean	<b>0.000E + 00</b>	1.178E-39	<b>0.000E + 00</b>	5.985E-14	8.065E + 00
Std	<b>0.000E + 00</b>	2.789E-39	<b>0.000E + 00</b>	3.278E-13	3.393E + 00
Wilcoxon		+	=	+	+
<i>F3</i>					
Mean	<b>0.000E + 00</b>	2.011E + 05	<b>0.000E-00</b>	1.557E-01	1.027E + 04
Std	<b>0.000E + 00</b>	4.474E + 04	<b>0.000E + 00</b>	8.381E-01	5.418E + 03
Wilcoxon		+	=	+	+
<i>F4</i>					
Mean	<b>0.000E + 00</b>	5.579E + 01	<b>0.000E + 00</b>	2.204E-47	1.987E + 01
Std	<b>0.000E + 00</b>	3.260E + 01	<b>0.000E + 00</b>	9.417E-47	3.586E + 00
Wilcoxon		+	=	+	+
<i>F5</i>					
Mean	<b>1.864E-03</b>	4.831E + 01	4.493E + 01	4.793E + 01	1.815E + 03
Std	<b>2.399E-03</b>	2.565E-01	1.222E + 01	5.436E-01	2.727E + 03
Wilcoxon		+	+	+	+
<i>F6</i>					
Mean	<b>1.927E-05</b>	5.349E-01	5.022E-01	2.221E + 00	9.781E-01
Std	<b>2.433E-05</b>	1.920E-01	1.620E-01	9.989E-01	1.535E + 00
Wilcoxon		+	+	+	+
<i>F7</i>					
Mean	<b>3.432E-05</b>	3.510E-03	9.119E-05	3.320E-03	5.556E-01
Std	<b>1.797E-05</b>	1.827E-03	9.556E-05	3.635E-03	1.992E-01
Wilcoxon		+	+	+	+
<i>F8</i>					
Mean	<b>-2.095E + 04</b>	- 1.637E+04	- 2.086E+04	- 1.221E+04	- 1.226E+04
Std	<b>9.893E + 00</b>	2.549E + 03	3.145E + 02	1.874E + 03	1.096E + 03
Wilcoxon		+	+	+	+
<i>F9</i>					
Mean	<b>0.000E + 00</b>	<b>0.000E + 00</b>	<b>0.000E + 00</b>	<b>0.000E + 00</b>	9.259E+01
Std	<b>0.000E + 00</b>	<b>0.000E + 00</b>	<b>0.000E + 00</b>	<b>0.000E + 00</b>	2.237E+01
Wilcoxon		=	=	=	+
<i>F10</i>					
Mean	<b>8.882E-16</b>	6.217E-15	<b>8.882E-16</b>	3.494E-15	4.455E + 00
Std	<b>4.012E-31</b>	2.237E-15	<b>4.012E-31</b>	1.598E-15	8.524E-01
Wilcoxon		+	=	+	+
<i>F11</i>					
Mean	<b>0.000E + 00</b>	4.261E-02	<b>0.000E + 00</b>	<b>0.000E + 00</b>	5.853E-01

**Table 6** (continued)

F	This paper	WOA	MWOA	EWOA	SSA
Std	<b>0.000E + 00</b>	1.328E-01	<b>0.000E + 00</b>	<b>0.000E + 00</b>	2.496E-01
Wilcoxon		+	=	=	+
<i>F12</i>					
Mean	<b>7.474E-06</b>	1.403E-02	1.177E-02	6.167E-02	1.323E + 01
Std	<b>2.037E-05</b>	3.661E-03	6.339E-03	3.238E-02	4.352E + 00
Wilcoxon		+	+	+	+
<i>F13</i>					
Mean	<b>2.729E-05</b>	4.411E-01	2.991E-01	2.031E + 00	7.948E + 01
Std	<b>2.406E-05</b>	1.546E-01	1.368E-01	7.023E-01	1.062E + 01
Wilcoxon		+	+	+	+

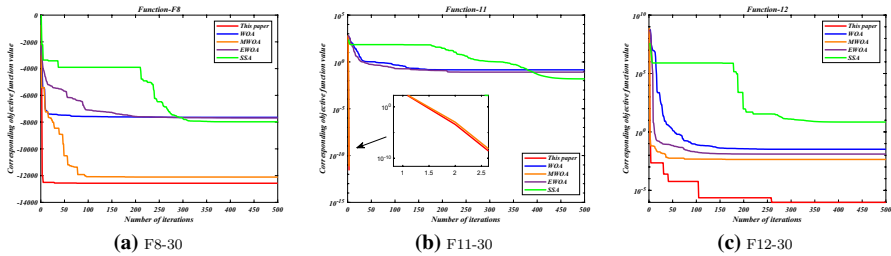
+/-/=/gm  
42/10/0/2

The bolded results indicate the best results in the same group

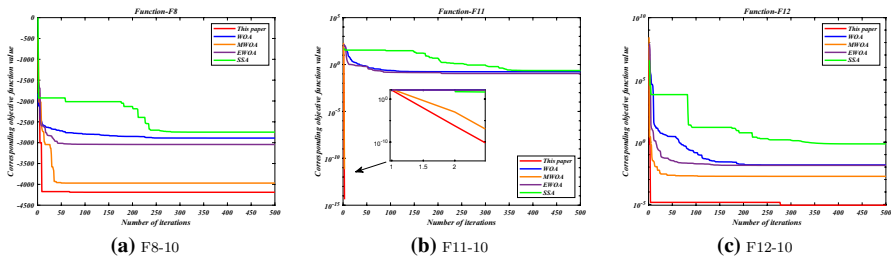
“+” indicates that the proposed algorithm is better than the algorithm used for comparison

“-” indicates that the proposed algorithm is worse than the one used for comparison

“=” indicates that the performance of the proposed algorithm is equivalent to that of the algorithm used for comparison



**Fig. 7** Convergence curve comparison of some multimodal functions (30 dimensions)



**Fig. 8** Convergence curve comparison of some multimodal functions (10 dimensions)

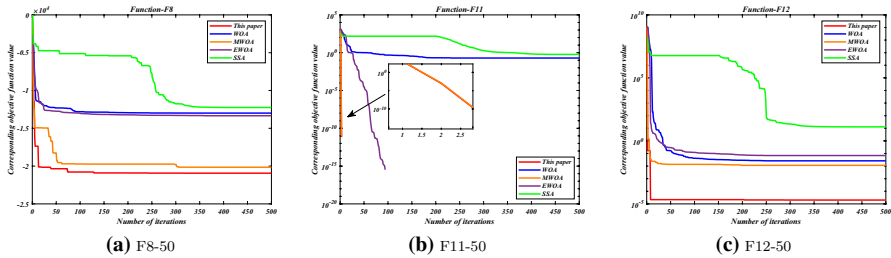


Fig. 9 Convergence curve comparison of some multimodal functions (50 dimensions)

weight effectively balances the algorithm’s extensive search capabilities and local development capabilities, allowing the information retained by the previous generation to be fully used. Therefore, the algorithm quickly found a better value after a short search and escaped the trap of local optimization. The total number of iterations in this test is only 500, which not only effectively reflects the search ability of the algorithm, but also makes the changes in the convergence curve more clear. In short, the cause of this phenomenon is easy to understand. The adaptive inertia weight is constantly receiving feedback information while adjusting the performance of the algorithm. In this way, the algorithm can successfully find a dynamic balance point between the exploration capability and the development capability to help the population quickly jump out of the local optimal value. The above data fully shows that the improved algorithm has a high success rate in solving such complex problems, and has a stronger ability to avoid local minimums.

### 4.5 Analysis of the effectiveness of DAWOA

In order to verify the effectiveness of each strategy in DAWOA, this section introduces effectiveness analysis. Adding the three strategies to WOA in the order of dynamic pinhole imaging strategy, local mutation strategy, and adaptive strategy can form three new algorithms: WOA-1, WOA-2 and WOA-3. In addition, this section selects three benchmark test functions to compare and test the four algorithms. The three benchmark functions are the unimodal function F1, F5 and the multimodal function F8 in turn. In order to ensure the fairness of the experiment, the dimension is uniformly set to 30. As shown in Table 7 and Fig. 11, no matter which test function it is, the three strategies of DAWOA are all effective. And, with the continuous addition of strategies, DAWOA’s performance is also continuously enhanced. This fully shows that the three strategies can not only work individually, but also work together to make the algorithm perform more comprehensively. At the same time, the specific effects of these strategies can also be briefly summarized as follows:

- (1) Dynamic pinhole imaging strategy (WOA-1). It is not difficult to see that the effect of the dynamic pinhole imaging strategy is significant. This shows that as the number of iterations increases, the dynamic pinhole imaging strategy has

been actively adjusting the position of the whale. This approach allows some whales to quickly jump to the vicinity of the optimal solution, thereby greatly improving the quality of leaders.

- (2) Adaptive strategy (WOA-2). Although the dynamic pinhole imaging strategy helped the whale obtain a higher-quality search area, the search ability of the individual whale has not been strengthened, which may cause WOA to fall into a local optimum to a large extent. It can be seen from Table 7 that the adaptive strategy can continuously adjust the search ability of the whale population under the action of the dynamic boundary to help the population quickly reach the target area. This approach can effectively improve the search capabilities of the algorithm.
- (3) Local mutation strategy (WOA-3). The significance of the local mutation strategy is to strengthen the local development capabilities of the algorithm. When the current two strategies help the algorithm approach the region where the optimal solution is located, the local mutation strategy can dynamically adjust the spiral search range of the algorithm following the number of iterations. With the continuous shrinking of the dynamic boundary, the dynamically reduced spiral path can effectively enhance the local search capability of the algorithm, allowing the algorithm to perform a more detailed search. This approach can significantly improve the convergence accuracy of the algorithm.

#### 4.6 Principle analysis

In order to more fully explain the role of the three strategies in DAWOA, this section will analyze DAWOA from a theoretical perspective. Undoubtedly, the search process of most algorithms can be divided into particle shrinkage and particle divergence phases. Therefore, we will discuss the advantages of DAWOA from the perspective of these two phases:

- (1) Particle divergence phase. At this phase, the population individuals of the algorithm will spread as far as possible across the entire search space to obtain richer diversity. However, for WOA, its exploration process is not sufficient, and the quality of leaders is not high enough. Although the exploration phase of WOA has a high priority, it must share the search process with the spiral update location phase. Therefore, the probability at this stage can be expressed as follows:

$$P_{|A|\geq 1} = \frac{1}{2} \int_1^2 \int_{1a}^1 d(2r - 1) da \approx 0.1535 \quad (19)$$

It is not difficult to see that this probability is much lower than expected. This is also the reason why WOA tends to fall into local optimum. When the dynamic pinhole imaging strategy comes into play, individuals in the population have more diverse positions. In many cases, some low-quality population individuals can quickly jump to the vicinity of the optimal solution under the action of these strategies, which undoubtedly brings more possibilities to the algorithm.

- (2) Particle shrinkage phase. For many typical non-convex problems, the algorithm is easy to find the target area. However, the ability of local search limits the accuracy of most algorithms, and WOA is no exception. The fixed search range makes it difficult for WOA to be attracted by the optimal solution near the search path, which leads to insufficient convergence accuracy. The local mutation strategy proposed in this paper can make the shape of the logarithmic spiral change with the number of iterations, which can to a large extent strengthen the development ability of the algorithm. At the same time, this strategy also contains an uncontrolled mutation, which can create a lot of opportunities for the algorithm when whales perform local searches. In this way, DAWOA can perform a more careful search during the particle shrinkage phase.

In addition, even if the algorithm has a better location and more flexible search range, if the whale's search ability is not improved, the performance of the algorithm will hardly be changed significantly. Therefore, this paper also proposes an adaptive strategy: the adaptive strategy allows the whale population to adjust its search ability according to changes in the environment to obtain a better global optimal solution. In this way, DAWOA can obtain more comprehensive performance under the synergy of the three strategies.

As a summary, the numerical experiment results in this section effectively prove the effectiveness of the improved algorithm. a). The unimodal function test results fully demonstrate the feasibility of the dynamic pinhole imaging strategy. Compared with EWOA, WOA and SSA, DAWOA can quickly find the global optimal value and converge. b). Multimodal function test results show that adaptive inertial weights based on dynamic boundary feedback can effectively increase the diversity of the population, make the distribution of population individuals more even, and it is easier to avoid local optima. c). The results of the fixed-dimensional function test show that when the search space becomes more complex, adaptive strategies and local mutation strategies can effectively balance the exploitation and exploration capabilities of the algorithm. This can prevent the population from gathering near the local optimal points that are easy to find in the space, leading to premature convergence of the algorithm. The effectiveness analysis further confirmed the above conclusions. In the following sections, DAWOA will be applied to more challenging engineering problems to further test its performance.

## 5 The application of DAWOA in engineering problems

The purpose of algorithm optimization is to better solve real-world problems. In order to test the actual effectiveness of DAWOA, four complex constrained engineering optimization problems will be used to test the specific performance of DAWOA. Among them, the specific parameters of the four engineering problems come from optimization literature [23, 37]. The detailed description of the problem and the analysis of the test results will be divided into four subsections to form the entire content of this section.

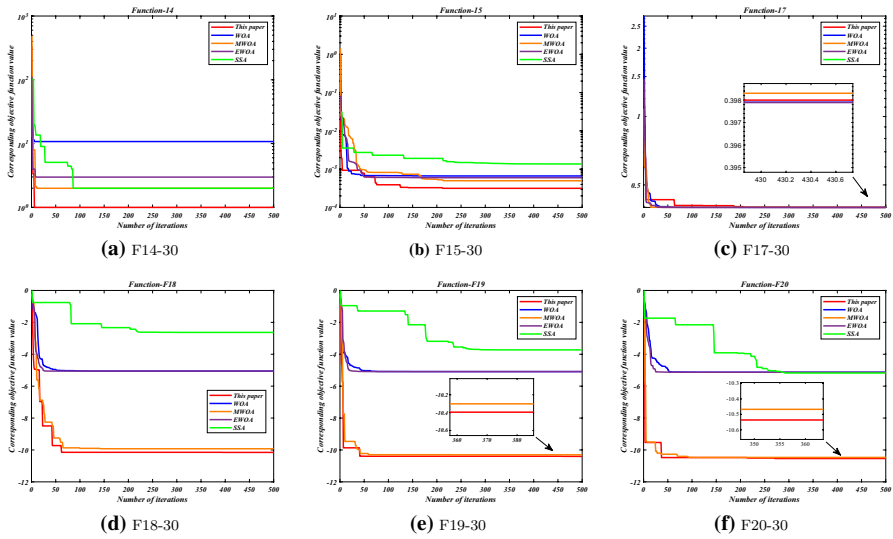


Fig. 10 Comparison of convergence curves of some fixed-dimensional test functions

Table 7 Test results of the four algorithms in some benchmark functions

F	WOA	DAWOA-1	DAWOA-2	DAWOA-3
<i>F1</i>				
Mean	6.214E-72	9.144E-141	<b>0.000E + 00</b>	<b>0.000E + 00</b>
Std	3.334E-71	4.577E-140	<b>0.000E + 00</b>	<b>0.000E + 00</b>
<i>F5</i>				
Mean	2.785E + 01	6.446E + 00	1.844E-02	<b>5.762E-04</b>
Std	4.963E-01	1.189E + 01	6.075E-02	<b>1.078E-03</b>
<i>F8</i>				
Mean	- 8.951E + 03	- 1.121E + 04	- 1.255E + 04	<b>- 1.257E + 04</b>
Std	1.329E + 03	1.731E + 03	4.427E + 01	<b>9.056E-01</b>

The bolded results indicate the best results in the same group

“+” indicates that the proposed algorithm is better than the algorithm used for comparison

“-” indicates that the proposed algorithm is worse than the one used for comparison

“=” indicates that the performance of the proposed algorithm is equivalent to that of the algorithm used for comparison

### 5.1 The problem of pressure vessel

The pressure vessel problem is a common optimization problem with complex mixed constraints in engineering. As shown in Fig. 12, this problem requires that the overall value be minimized while considering the cost of the structure, material, manufacturing, and welding of the vessel. Moreover, there are only four important design variables for the pressure vessel problem:



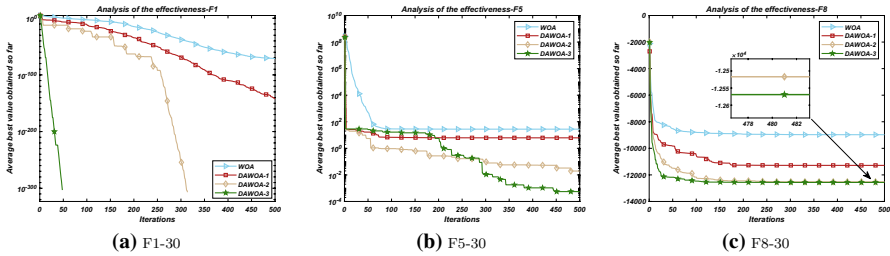


Fig. 11 Comparison of the convergence curves of the four algorithms in some benchmark functions

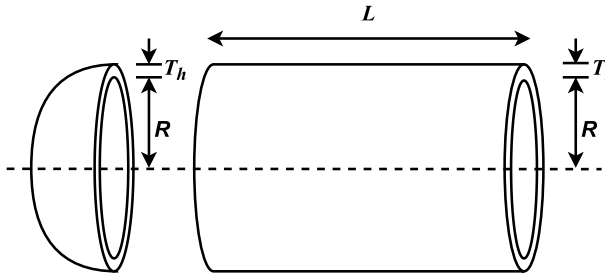


Fig. 12 Problem of pressure vessel

Table 8 Comparison of results of pressure vessel problems

Algorithm	Optimal values for variables				Optimum cost
	$X_1$	$X_2$	$X_3$	$X_4$	
This paper	0.7782424	0.3847675	40.323	200	<b>5886.7645</b>
MBA	0.7802	0.3856	40.4292	198.4964	5889.3216
RCGA-rdn	N/A	N/A	N/A	N/A	5893.8627
LSA-SM	0.8103764	0.4005695	41.98842	178.0048	5942.6966
DMMFO	0.743	0.3842	40.319619	200	6032.5484
WOA	0.8125	0.4375	42.098269	176.639	6059.741
PSO	0.8125	0.4375	42.091266	176.7465	6061.0777
GA	0.9375	0.5	48.329	112.679	6410.3811
DE	0.8125	0.4375	42.098411	176.63769	6059.734
ACO	0.8125	0.4375	42.103624	176.57266	6059.0888
MFO	0.8125	0.4375	42.098445	176.6366	6059.7143
GSA	1.125	0.625	55.98866	84.454203	8538.8359
ES	0.8125	0.4375	42.098087	176.64052	6059.7456

The bolded results indicate the best results in the same group

“+” indicates that the proposed algorithm is better than the algorithm used for comparison

“-” indicates that the proposed algorithm is worse than the one used for comparison

“=” indicates that the performance of the proposed algorithm is equivalent to that of the algorithm used for comparison

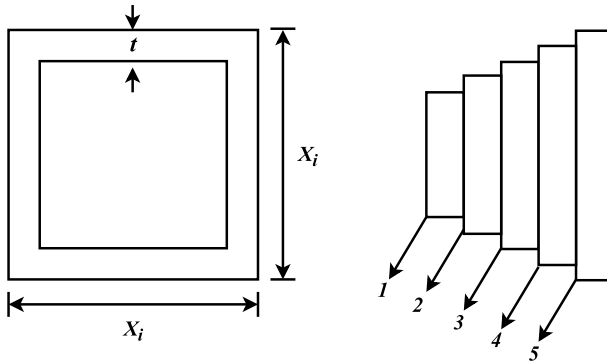


Fig. 13 Problem of cantilever beam

- $X_1$ —Thickness of the two lids of the container.
- $X_2$ —The thickness of the container shell.
- $X_3$ —Inner radius of the container.
- $X_4$ —The length of the cylinder in the middle of the container.

Table 8 shows the best data comparison between DAWOA and WOA, LSA-SM, DE, ACO, GA, RCGA-rdn [38], MBA, DMMFO [23], GSA and ES. Most of the data used for comparison comes from the literature [15, 23, 39]. It is not difficult to see that the performance of DAWOA is better than other comparison algorithms, and the numerical results obtained by DAWOA are even better.

Table 9 Comparison of numerical results of cantilever beam problem

Algorithm	Optimal values for variables					Optimum cost
	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	
This paper	6.0217	5.3125	4.4881	3.5011	2.1504	<b>1.33652294</b>
HWOANM	6.01812	5.31142	4.48836	3.49751	2.15833	1.33995
EWOA	6.01867	5.31481	4.49132	3.49907	2.15234	1.3399591
CS	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
GOA	6.01167	5.31297	4.48307	3.50279	2.16333	1.33996
MVO	6.02394	5.30601	4.49501	3.49602	2.15273	1.3399595
MMA	6.01	5.3	4.49	3.49	2.15	1.34
SOS	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996

The bolded results indicate the best results in the same group

“+” indicates that the proposed algorithm is better than the algorithm used for comparison

“-” indicates that the proposed algorithm is worse than the one used for comparison Algorithm

“=” indicates that the performance of the proposed algorithm is equivalent to that of the algorithm used for comparison

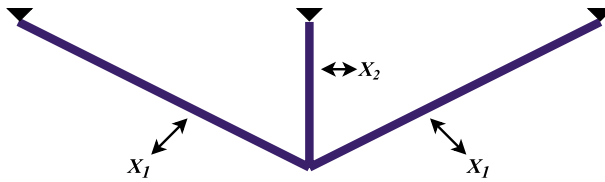


Fig. 14 Problem of three-bar truss

## 5.2 The problem of cantilever beam

The purpose of the cantilever beam problem is to minimize the weight of the square-section cantilever beam. As shown in Fig. 13, five hollow blocks of different sizes constitute a typical cantilever beam. Among them, the number of hollow blocks determines the number of parameters, and  $X_i$  is the side length of the square block. Table 9 is the comparison of the numerical results of DAWOA and some algorithms. Among them, the test data of all the comparison algorithms are from the literature

**Table 10** Comparison of results of three-bar truss problems

Algorithm	Optimal values for variables		Optimum cost
	$X_1$	$X_2$	
This paper	0.78867	0.40826	<b>263.8958442</b>
MBA	0.78857	0.40856	263.8958522
CS	0.78867	0.40902	263.9716
WDO	0.78662	0.41418	263.9085
Ray	0.795	0.395	264.3
MFO	0.78824	0.40947	263.8959796
GOA	0.7889	0.40762	263.8958810

The bolded results indicate the best results in the same group

“+” indicates that the proposed algorithm is better than the algorithm used for comparison

“-” indicates that the proposed algorithm is worse than the one used for comparison Algorithm

“=” indicates that the performance of the proposed algorithm is equivalent to that of the algorithm used for comparison

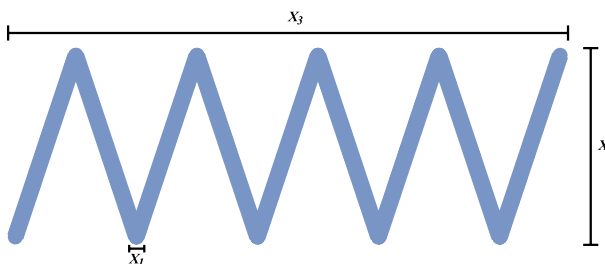


Fig. 15 Problem of tension/compression spring

**Table 11** Comparison of results of tension/compression spring problems

Algorithm	Optimal values for variables			Optimum cost
	$X_1$	$X_2$	$X_3$	
This paper	0.0516801	0.356503	11.3016	<b>0.012665</b>
WOA	0.051207	0.345215	12.004032	0.0126763
PSO	0.51728	0.357644	11.244543	0.012675
ES	0.51989	0.363965	10.890522	0.012681
GA	0.05148	0.351661	11.632201	0.012705
DE	0.051609	0.354714	11.410831	0.01267
SaC	0.0521602	0.3681587	10.6484422	0.0126692
MFO	0.0519945	0.3641093	10.868422	0.0126669
NM-PSO	0.05162	0.355498	11.333327	0.0126706
CPSO	0.051728	0.357644	11.24454	0.0126747

The bolded results indicate the best results in the same group

“+” indicates that the proposed algorithm is better than the algorithm used for comparison

“-” indicates that the proposed algorithm is worse than the one used for comparison Algorithm

“=” indicates that the performance of the proposed algorithm is equivalent to that of the algorithm used for comparison

[15, 37–39]. It is not difficult to see that DAWOA is still the best optimizer among them.

### 5.3 The problem of three-bar truss

The three-bar truss problem is a complex structural optimization problem. An obvious feature of this problem is that the algorithm is required to find the minimum value in a narrow search space. Therefore, the three-bar truss problem has a fairly broad research prospect. As shown in Fig. 14, there are two adjustable variables in the three-bar truss problem. The focus of the problem is to keep the weight of the entire structure as small as possible. DAWOA was used to compare with MBA, CS, GOA, WDO [40], Ray [41], MFO, and the comparison results are shown in Table 10. Among all the algorithms, DAWOA's numerical results are better than other algorithms and have the most reliable performance.

### 5.4 The problem of tension/compression spring

The tension/compression spring problem was first proposed by Belegundu in 1982. It is a very common and well-known constrained engineering optimization problem. As shown in Fig. 15, the problem contains three important parameters, the purpose is to minimize the weight of the entire spring while ensuring the performance of the spring. Table 11 is the comparison result of DAWOA and some algorithms. The data of these algorithms comes from literature [15, 40]. It can be seen from

the comparison results that the numerical results obtained by DAWOA are the best among all the algorithms (Table 11).

As a summary, this section tests the performance of DAWOA in four well-known engineering optimization problems with complex constraints. From the comparison of numerical results, DAWOA performs better than other comparison algorithms. These experimental results fully show that DAWOA can effectively solve some practical problems and has a certain practicability.

## 6 Conclusion and future work

This paper proposes three strategies to help WOA improve its performance. The dynamic pinhole imaging strategy inspired by optics allows the leader to quickly approach areas that may contain high-quality solutions, thereby effectively accelerating the convergence speed of the algorithm. Adaptive inertial weights based on dynamic boundaries and dimensions can properly balance the exploitation and exploration capabilities of the algorithm, making the algorithm's population diversity richer. So as to effectively strengthen the search ability of the algorithm and prevent the algorithm from premature convergence. The local mutation strategy can dynamically adjust the search range of the algorithm according to the number of iterations, so that the whale population can have more diverse spiral paths. The improved algorithm has been extensively tested in 20 well-known benchmark functions and four complex constrained engineering optimization problems, and compared with other optimization algorithms presented in the literature. The test results show that the improved algorithm has faster convergence speed and higher convergence accuracy, and can effectively jump out of the local optimum.

Although the performance of the improved algorithm DAWOA has been significantly improved to a certain extent, its performance in some test functions will gradually decrease when the dimension increase. This situation reflects the deficiencies of DAWOA to a certain extent. If the leader is always unable to reach the vicinity of the global optimal solution, no matter how many iterations, the entire whale population will most likely fall into a local optimum. How to ensure that the leader's position update is efficient and reliable, this is the problem we are about to solve. In the future, we will conduct more in-depth research on WOA for ultra-high-dimensional large-scale problems and multi-objective problems to solve more real-world problems.

**Acknowledgements** This work was supported by the National Natural Science Foundation of China (No. 61603127). The authors would like to thank all the anonymous referees for their valuable comments and suggestions to further improve the quality of this work.

## References

1. Holland JH (1973) Genetic algorithms and the optimal allocation of trials. *SIAM J Comput* 2(2):88–105
2. Koza JR (1992) Genetic programming

3. Rechenberg I (1978) *Evolutions strategien*. Springer, Berlin, Heidelberg, pp 83–114
4. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
5. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. *Inf Sci* 179(13):2232–2248
6. Seyedali M (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl-Based Syst* 89:228–249
7. Du H, Wu X, Zhuang J (2006) Small-world optimization algorithm for function optimization. In: Jiao L, Wang L, Gao X, Liu J, Wu F (eds.), *Advances in natural computation. ICNC (2006) Lecture Notes in Computer Science*, vol 4222. Springer, Berlin, Heidelberg
8. Richard F (2007) Central force optimization: a new metaheuristic with applications in applied electromagnetics. *Prog Electromag Res* 77:425–491
9. Eskandar H, Sadollah A, Bahreininejad A et al (2012) Water cycle algorithm: a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput Struct* 110:151–166
10. Kennedy J, Eberhart RC (2002) Particle swarm optimization. In: *Proceedings of the 1995 IEEE International Conference on Neural Networks* 4:1942–1948
11. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
12. Xue J, Shen B (2020) A novel swarm intelligence optimization approach: sparrow search algorithm. *Syst Sci Control Eng* 8(1):22–34
13. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm. *Adv Eng Softw* 114:163–191
14. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. *IEEE Comput Intell Mag* 1(4):28–39
15. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
16. Saha N, Panda S (2020) Cosine adapted modified whale optimization algorithm for control of switched reluctance motor. *Comput Intell*. <https://doi.org/10.1111/coin.12310>
17. Kong Z, Zhao J, Yang Q, Ai J, Wang L (2020) Parameter reduction in fuzzy soft set based on whale optimization algorithm. *IEEE Access* 8:217268–217281
18. Sulaiman M, Samiullah I, Hamdi A, Hussain Z (2019) An improved whale optimization algorithm for solving multi-objective design optimization problem of PFHE. *J Intell Fuzzy Syst* 37:3815–3828
19. Gaganpreet K, Sankalap A (2018) Chaotic whale optimization algorithm. *J Comput Des Eng* 5(3):275–284
20. Li Y, Han M, Guo Q (2020) Modified whale optimization algorithm based on tent chaotic mapping and its application in structural optimization. *KSCE J Civ Eng* 24:3703–3713
21. Hemasian-Etefagh F, Safi-Esfahani F (2020) Group-based whale optimization algorithm. *Soft Comput* 24:3647–3673
22. Kaveh A, Ilchi Ghazaan M (2017) Enhanced whale optimization algorithm for sizing optimization of skeletal structures. *Mech Based Des Struct Mach* 45(3):345–362
23. Ma L, Wang C, Xie NG et al (2021) Moth-flame optimization algorithm based on diversity and mutation strategy. *Appl Intell* 51:5836–5872
24. Xingguo Q, Ruizhi W, Weiguo Z, Zhaozhao Z, Jing Z (2021) Improved whale optimization algorithm based on hybrid strategy. *Comput Eng Appl* 1–12
25. Shuang X, Jingmin Z (2021) Hybrid WOAMFO algorithm based on Lévy flight and adaptive weights. *Math Pract Understanding*: 1–11[2021-05-15]
26. Luo J, Shi B (2019) A hybrid whale optimization algorithm based on modified differential evolution for global optimization problems. *Appl Intell* 49:1982–2000
27. Zhang J, Wang JS (2020) Improved whale optimization algorithm based on nonlinear adaptive weight and golden sine operator. *IEEE Access* 8:77013–77048
28. Tanyildizi E, Demir G (2017) Golden sine algorithm: a novel math inspired algorithm. *Adv Electr Comput Eng* 17(2):71–78
29. Tizhoosh HR (2015) Opposition-based learning: a new scheme for machine intelligence. In: *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*. Austria, Vienna, pp 695–701
30. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. *IEEE Trans Evol Comput* 3:82–102

31. Digalakis J, Margaritis K (2001) On benchmarking functions for genetic algorithms. *Int J Comput Math* 77:481–506
32. Molga M, Smutnicki C (2005) Test functions for optimization needs
33. Yang X-S (2010) Firefly algorithm, stochastic test functions and design optimisation. *Int J Bio-Inspired Comput* 2(2):78–84
34. Wentao F, Kekang S (2020) An enhanced whale optimization algorithm. *Comput Simul* 37(11):275–279
35. Zhang Damin X, Yirou HW, Song T, Wang L (2021) Whale optimization algorithm for embedded circle mapping and one-dimensional learning based small hole imaging. *Control Decis* 36(05):1173–1180
36. Seyedali M (2016) SCA: a sine cosine algorithm for solving optimization problems. *Knowl-Based Syst* 96:120–133
37. Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a meta-heuristic approach to solve structural optimization problems. *Eng Comput* 29:17–35
38. Song Y, Wang F, Chen X (2019) An improved genetic algorithm for numerical function optimization. Springer, US
39. Yan Z et al (2021) Nature-inspired approach: an enhanced whale optimization algorithm for global optimization. *Math Comput Simul* 185:17–46
40. Bayraktar Z, Komurcu M, Bossard JA et al (2013) The wind driven optimization technique and its application in electromagnetics. *IEEE Trans Antennas Propagation* 6(5):2745–2755
41. Ray T, Saini P (2001) Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng Optim* 33(6):735–748

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.