# A modified whale optimization algorithm with multi-strategy mechanism for global optimization problems

**Mingyuan Li**[1] · **Xiaobing Yu**[1] · **Bingbing Fu**[1] · **Xuming Wang**[1]

## Abstract

Whale Optimization Algorithm (WOA) is an outstanding nature-inspired algorithm widely used to solve many complex engineering optimization problems. However, WOA has a poor balance in exploration and exploitation, which converges to local optimum easily. This article proposes a Modified Whale Optimization Algorithm (MWOA) with multi-strategy mechanism, which introduces the elite reverse learning strategy, nonlinear convergence factor, DE/rand/1 mutation strategy and Lévy flight disturbance strategy. MWOA can improve the convergent ability and maintain the balance of exploitation and exploration to avoid local optimum. Compared with WOA, PSO, MFO, SOA, SCA and other four WOA variants on the CEC2017 benchmark suite, MWOA has strong competitiveness and can better improve the efficiency of WOA according to the experimental results and analysis.

**Keywords** Whale optimization algorithm · Multi-strategy · Elite reverse learning strategy · Nonlinear convergence factor

## 1 Introduction

Meta-heuristic algorithms are the product of the improvement of the heuristic algorithm, which combines the random algorithm and the local search algorithm. They are random search methods characterized by imitating various operational mechanisms [1]. Since this kind of algorithm can reduce the amount of calculation, they are used in computer science, engineering optimization design and other fields [2–5].

According to the mechanism they are inspired by, meta-heuristic optimization algorithms fell generally into two groups: algorithms that imitate biological processes and algorithms based on physical principles. The first division includes evolutionary-based and swarm-based algorithms. Genetic Algorithm (GA) [6] is the earliest evolutionary-based algorithm. Particle Swarm Optimization (PSO) [7]

opens up a precedent for swarm-based algorithms. Then many specialists have successively proposed lots of swarm intelligence optimization algorithms. The main inspiration for Moth-Flame Optimization (MFO) [8] comes from the navigation method of moths called 'horizontal orientation'. In light of the fact that ants can always discover the best route when foraging, Ant Colony Optimization (ACO) [9] is established. In addition, there are also some well-known algorithms, such as Chimp Optimization Algorithm (ChOA) [10] and Artificial Bee Colony (ABC) [11].

As a novel kind of meta-heuristic algorithms, Whale Optimization Algorithm (WOA) [12] imitates the special bubble-net predation strategy of humpback whales. WOA has fewer parameters and operates more easily. It is extensively used to manage engineering optimization problems [13–16]. According to the experimental results of Ref. [12], WOA converges faster than DE and PSO, and has strong competitiveness and more significant application potential.

However, due to the characteristics of the algorithm itself, there are still some deficiencies in the optimization process. The main disadvantages of the WOA are that it is easy to fall into the local optimum and the convergence

✉ Xiaobing Yu
yuxiaobing@nuist.edu.cn

1   School of Management Science and Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China

accuracy is not high. Because of the random initialization and parameter randomization, the exploration and exploitation are unbalanced, which converges to the local optimum easily [17]. Further, the convergent ability needs to be improved. The WOA algorithm gradually exposes some shortcomings in solving complex optimization problems. To overcome the shortcomings, some scholars have proposed WOA variants to enhance its performance.

According to the existing literature, the improvements are generally classified into three parts: the improvement achieved by adjusting the parameters, the improvement achieved by the adjustment of the search strategy and the improvement achieved by mixing WOA with other algorithms.

Many works have improved the algorithm by adjusting the parameters. Kaur and Arora [18] applied chaos theory in WOA and proposed CWOA, which used various chaotic maps to adjust the main parameters. Zhong and Long [19] have proposed a nonlinear strategy for different control parameters. Li et al. [20] used a nonlinear tuning parameter in an improved WOA (IWOA). Due to the few parameters of WOA, this kind of improvement is relatively simple, and the improvement effect is general.

The second is to enhance the performance by adjusting the search strategy, generally used in initializing the population and updating the location. Rashmi proposed [21] a modified WOA based on roulette wheel selection to balance exploration and exploitation. Elaziz and Oliva [22] employed an opposition-based learning strategy for a better-initialized population. Zhang and Wang [23] used the nonlinear adaptive weight to enable whales to search adaptively. Chen et al. [24] proposed the RDWOA, which used the random spare strategy to overcome the shortcoming of low convergence speed. To enhance the global exploration ability, Liu et al. [25] used the Lévy flights strategy to increase the search space. Chakraborty [26] proposed a novel improved WOA method (ImWOA) with increased diversity in the solution. The entire iteration is clearly divided into exploration and exploration. Mohammad [27] proposed an enhanced WOA with three new effective search strategies named migrating, preferential selecting, and enriched encircling prey.

Barring the above two types of improvements, some works are devoted to hybridizing the WOA algorithm with other intelligent algorithms to solve complex problems. Indrajit and Pradeep [28] raised a hybrid PSO-WOA algorithm, where WOA was used for exploration. Hardi and Tarik [29] proposed an improved algorithm hybridized with GWO to get better solutions. The hunting mechanism of GWO was applied in the exploitation stage of WOA. Kaveh and Rastegar [30] hybridized WOA with some concepts of Collider Bodies Optimization. Seyed and Samaneh [31] hybridized WOA with Differential Evolution (DE) to obtain improved WOA. This approach combined the exploration stage of DE to get a better solution. Bentouati et al. [32] combined the Pattern Search algorithm (PS) with WOA to solve power system problems. Tang [33] proposed a WOA mixed with the Artificial Bee Colony (ACWOA) to solve the problems of slow convergence and low precision. Dey [34] used a novel hybrid WOA-Sine cosine algorithm (SCA) to minimize the generation cost of a low voltage (LV) grid-connected microgrid system.

Although a series of works on the improvement of WOA has been carried out and the efficiency of WOA has been upgraded, there are some shortages. On the one hand, the strategies are relatively simple, using only nonlinear convergence parameters or adaptive weight. The single strategy cannot effectively achieve significant improvements in WOA performance. It is challenging to solve complex optimization problems. According to Ref. [23], the control parameter $a$ was altered by different nonlinear adjustment strategies, and the improvement was not particularly effective. On the other hand, the computational complexity is higher than WOA, which will increase a lot of workloads. Chakraborty et al. [35] proposed an enhanced WOA, which introduced the mutualism phase of the Symbiotic Organisms Search. However, the complexity was higher than WOA. In addition, the improvement of WOA mainly based on the hybrid algorithm has resulted in losing evolutionary direction, which is challenging to realize and easy to sink into local optimum [36].

By consulting a lot of literature about the improvement of WOA, it is concluded that the effect of single strategy improvement is not very obvious. It is difficult to solve complex problems. In addition, most improvements only unilaterally improve the performance, such as increasing the convergence speed, improving the exploration capability and other single aspects. In contrast, the combination of multiple strategies can greatly boost the efficiency of WOA. Yuan et al. [37] proposed the Multi-Strategy Ensemble WOA, which employed the chaos strategy, an improved random search mechanism, Lévy flight, and an enhanced position correction mechanism. It balanced local and global searches to avoid premature algorithm convergence. Sun [38] devised a modified WOA based on multi-strategy to address somewhat deficiencies of the original WOA, which used a tent map function, inertia weight and an optimal feedback strategy. At present, many types of research are devoted to introducing different new strategies into WOA. The effective combination of multiple strategies may enhance the performance of WOA [39]. Elite reverse learning-related strategies, mutation strategies and Lévy flight-related strategies have been proved to be simple and effective in enhancing meta-heuristic optimization algorithms. Inspired by these references, we introduce these strategies into WOA.

Hence we propose a modified WOA (MWOA) with Multi-strategy mechanism. The above drawbacks can be

solved by introducing new strategies to the WOA algorithm. For the problem of slow convergence and local optimum, MWOA improves the initialization, control parameters and search strategy of WOA, respectively. Improving at different stages of the algorithm and assisting each other can lead to better results. For the random initialization of WOA, the elite reverse learning strategy is introduced to get a better population. Then in terms of improvement of parameters, the nonlinear convergence factor is used for strengthening exploitation and exploration and maintains a balance of them. The search strategy is boosted by introducing the mutation strategy and a Lévy flight disturbance strategy. To enhance the exploration capacity and avoid local optimum, the DE/rand/1 mutation strategy is introduced into the exploration phase of WOA. Lastly, utilizing the Lévy fight disturbance strategy to boost the spatial search and accelerate convergence, which can strengthen the global search ability. Compared with WOA, PSO, MFO, SOA, SCA and other four WOA variants on the CEC 2017 benchmark suite, MWOA has strong competitiveness and can enhance the performance of WOA.

The remainder of this article is as below. WOA is briefly introduced in Sect. 2. MWOA is presented in Sect. 3, which introduces the mechanism of the four strategies and the pseudocode of MWOA. In Sect. 4, the experimental results are presented and analyzed. Lastly, we conclude in Sect. 5.

## 2 WOA

WOA simulates the unique hunting behavior of humpback whales. These whales usually live in groups in nature. When the prey is found, groups of humpback whales spiral upward from the bottom while exhaling air bubbles to surround their prey. These humpback whales usually hunt through three phases (1) Surround the prey; (2) Bubble-net attacking strategy; (3) Global search for prey. According to the three stages of whale foraging modeling, the WOA is obtained.

### 2.1 Surround the prey

Whales must first recognize the location of the prey. Because they cannot obtain specific location information of prey in advance, the location of the best whale is taken as the target location. Other individuals will move toward the best whale, and this behavior is expressed in Eqs. (1–2).

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

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

where $t$ is the index of iterations, $\vec{X}(t)$ is the position of the current population, $\overrightarrow{X^*}(t)$ is the optimal position that has been searched, $\vec{X}(t+1)$ deems the position that needs to be updated in the next iteration, $\vec{A}$ indicates the convergence factor and $\vec{C}$ represents the disturbance factor, computed as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r_1} - \vec{a} \tag{3}$$

$$\vec{C} = 2\vec{r_2} \tag{4}$$

where random vectors $\vec{r_1}$ and $\vec{r_2}$ are in [0,1], $a$ represents the convergence factor decreased from 2 to 0 during the iteration, whose changes can be expressed by:

$$a = 2 - (2t/t_{\max}) \tag{5}$$

where $t$ is the index of iterations.

### 2.2 Bubble-net attacking strategy

Whales will conduct spiral bubble net attacks on the target prey. There are two main ways: the shrinking encircling mechanism and the spiral updating position. The following equation can realize the spiral updating position.

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

where $\overrightarrow{D_p} = \left| \overrightarrow{X^*}(t) - \vec{X}(t) \right|$ represents the distance between the individual and the target prey, $b$ denotes a constant constrained to a logarithmic spiral shape, and the random number $l$ is in [−1,1].

Whales randomly update their location. There is a 50 percent chance of choosing either contraction encirclement or spiral updating position, $p$ is defined as a random number in [0, 1]. This behavior can be obtained by:

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A} \cdot \vec{D}, & p < 0.5 \\ \overrightarrow{X^*}(t) + \overrightarrow{D_p} \cdot e^{bl} \cdot \cos(2\pi l), & p \geq 0.5 \end{cases} \tag{7}$$

### 2.3 Global search for prey

Assuming a random whale as the best whale. Other whales will approach this whale. Humpback whales search for prey by a random walk. When |A| > 1, it focuses on exploration and performs the global search. This behavior is obtained as follows:

$$\vec{D} = \left| \vec{C} \cdot \overrightarrow{X_{rand}}(t) - \vec{X}(t) \right| \tag{8}$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}}(t) - \vec{A} \cdot \vec{D} \tag{9}$$

In the WOA algorithm, individuals update the positions based on the best whale or a random whale. It can realize the transition between exploitation and exploration by

adjusting the parameter $a$. Whales will search for prey randomly when $|\vec{A}| > 1$. When $|\vec{A}| < 1$, whales will attack their prey, and WOA is devoted to exploitation at this time.

The flowchart of WOA is presented in Fig. 1. Although WOA has some advantages compared with other algorithms, it also exposes some shortcomings. The main defect is the slow convergence rate [36]. The exploration and exploitation are unbalanced, which will lead WOA into local optimum in complex environments. Aiming at these problems, we propose MWOA to overcome deficiencies and enhance performance.

# 3 MWOA

MWOA adopts the elite reverse learning strategy, nonlinear convergence factor, DE/rand/1 mutation strategy and Lévy flight disturbance strategy. Elite reverse learning strategy is used at initialization to accelerate the

convergence and obtain an excellent population. The nonlinear convergence factor can well balance exploitation and exploration. Then introducing the mutation strategy to enhance exploration capabilities. MWOA introduces a new hybrid operator $m$ to control exploration and exploitation. Finally, applying the Lévy flight perturbation strategy to improve the global search capability and avoid local optimum. The details are as follows.

## 3.1 Elite reverse learning strategy

At initialization, the individuals generally update the location information randomly. The search time will be longer when it converges to the optimal solution. Therefore, initialization can significantly influence the search performance of WOA. Using the elite reverse learning strategy to initialize the population can obtain a better-initialized population and reduce the time for searching for the optimal solution [40].

**Fig. 1** The schematic diagram of WOA
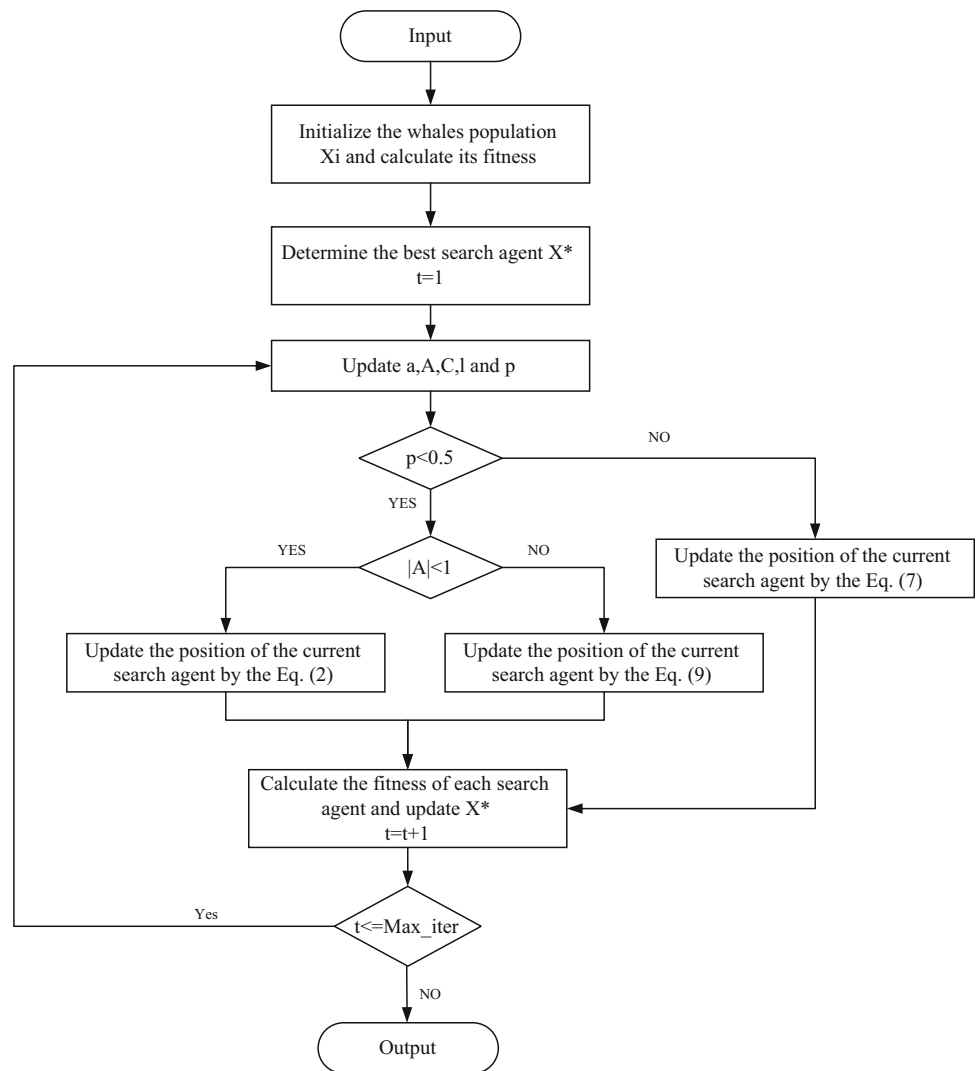
Elite reverse learning strategy is used to generate reverse learning individual $\overline{X_i}$, which is given by:

$$\overline{X_i} = k(lb + ub) - X_i \qquad (10)$$

where $lb$ represents the upper boundary, $ub$ represents the lower boundary, and $X_i$ indicates the current individual location information of the initialized population. Reverse individuals are obtained through elite reverse learning. Calculating the fitness of $\overline{X_i}$ and comparing it with the fitness of the $X_i$. Better individuals are retained, which can be expressed by Eq. (11).

$$X_i = \begin{cases} \overline{X_i}, f(\overline{X_i}) < f(X_i) \\ X_i, f(\overline{X_i}) > f(X_i) \end{cases} \qquad (11)$$

## 3.2 Nonlinear convergence factor

The coefficient $a$ is the main factor affecting convergence performance. Because it decreases linearly, the exploration and exploitation of WOA cannot reach a balance [41]. According to Ref. [19], Zhong and Long proposed a nonlinear tuning strategy with different control parameters in the improved algorithm. The improvement using the cosine nonlinear convergence parameter was proved to be the best in several experiments. This paper uses this strategy and the nonlinear convergence factor $a_1$ calculated by Eq. (12).

$$a_1 = (a_{\max} - a_{\min}) \cdot \cos\left(7 \cdot \frac{t}{t_{\max}}\pi\right) \qquad (12)$$

where the $a_{\max}$, $a_{\min}$ are the maximal and minimal of the parameter $a$.

## 3.3 Mutation strategy

The DE algorithm [42] has only been developed for more than 20 years since it was proposed. However, it has shown structural advantages such as firm performance, simple structure and parallel operation. The DE algorithm has a good exploration ability for the function optimization problem [43]. By introducing the mutation strategy of DE to other meta-heuristic optimization algorithms, better solutions can be effectively obtained, and the exploration ability can be enhanced [44]. Hu et al. [45] used the mutation operator of differential evolution at mutation phase of BWO which helps the algorithm escape from the local optima. In this section, we give a brief introduction to the DE algorithm. This is mainly to reference the mutation policy in the DE algorithm. This strategy prevents stagnation in the search process, and it can deviate from the local optimum. WOA is prone to fall into local optimum and is particularly prone to premature convergence.

MWOA introduces the DE/rand/1/ mutation strategy into the exploration stage of WOA, which can boost the exploration capability and avoid local optimum. The mutation formula is as follows:

$$V_i(t) = X_{w_1}(t) + F \cdot (X_{w_2}(t) - X_{w_3}(t)) \qquad (13)$$

The mutation vector $V_i(t)$ is obtained through the mutation operation, $w_1$, $w_2$ and $w_3$ are random integers generated in $[1, N]$, they are not equal to each other. $F$ denotes the mutation operator, which controls the magnification of the deviation vector in practice. $N$ is the population size.

Then, to increase the vector's diversity, a crossover operation is introduced on the population. It can be presented by:

$$U_i(t) = \begin{cases} V_i(t), & rand(0, 1) \leq CR \, or \, j = j_{rand} \\ X_i(t), & others \end{cases} \qquad (14)$$

where $U_i(t)$ is the test vector obtained by the crossover operation, and $CR$ is the crossover operator.

Finally, the selection operation is performed. The generated test vector $U_i(t)$ is compared with the original individual $X_i(t)$. Better individuals are retained. The selection process can be represented by:

$$X_i(t + 1) = \begin{cases} U_i(t), & fU_i(t) < fX_i(t) \\ X_i(t), & \text{otherwise} \end{cases} \qquad (15)$$

In MWOA, we introduce a new hybrid operator $m$ to adjust the exploration and exploitation. When $rand \leq m$, it performs the exploration part. When and $> m$, it performs the exploitation part. The operator $m$ is calculated by Eq. (16).

$$m = 1 - \frac{t}{t_{\max}} \qquad (16)$$

## 3.4 Lévy flight disturbance strategy

Since Yang [46] proposed the cuckoo search algorithm which used two mechanisms of cuckoo parasitic reproduction and Lévy flight. The method has been widely used to improve the meta-heuristic algorithms [47–49]. In recent years, relevant scholars have conducted a series of research and proved that it helps to enhance the exploration of the optimization algorithm [50]. In WOA, individuals gradually approach to the optimal whale. The whale population is relatively concentrated, which can weaken global search capabilities and trap in the local optimum.

The Lévy flight disturbance strategy introduced in this paper perturbs the population after each position updates. It can improve the phenomenon of local optimization and premature convergence. The new equation for updating the position can be expressed by:

$$U_i(t) = U_i(t) + \alpha \cdot Lev\acute{y}(\lambda) \qquad (17)$$

where $\alpha$ denotes the step size control factor, $U_i(t)$ is the

position of the current population, and *Lévy* ($\lambda$) is the path of random search, calculated by Eq. (18)

$$Le\acute{v}y(\lambda) = \frac{u}{|v|^{1/\beta}} \qquad (18)$$

where $u$ and $v$ follow normal distribution, $u \sim N(0, \delta_u^2)$, $v \sim N(0, \delta_v^2)$, $\delta_v = 1$ and $\delta_u$ can be calculated by Eq. (19).

$$\delta_u = \left\{ \frac{\Gamma(1 + \beta) \sin(\pi\beta/2)}{\Gamma[(1 + \beta)/2]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta} \qquad (19)$$

where $\Gamma$ is the standard gamma function. $\beta$ usually takes the value of 3/2.

The pseudocode of MWOA is shown below.

| | **Algorithm 1** The MWOA algorithm |
|---|---|
| 1 | Initialize each whale $X_i (i = 1, 2 \dots, n)$ |
| 2 | Compute the fitness of each whale |
| 3 | Elite reverse learning strategy is used for generating reverse learning individual $\overline{X}_i$ |
| 4 | Compare the fitness function values of $X_i$ and $\overline{X}_i$, and retain excellent individuals |
| 5 | $X^* =$ the best search agent |
| 6 | while ($t < Max\_iter$) |
| 7 |    for each individual |
| 8 |    Update $a_1, A, C, l, p, q, j_{rand}$ and $m$ |
| 9 |    **if** ($q \leq m$) |
| 10 |      **if** rand $\leq$ CR or $j == j_{rand}$ |
| 11 |      Select three random search agents($w_1, w_2, w_3$) |
| 12 |        $U_i(j) = X_{w_1}(j) + F \cdot (X_{w_2}(j) - X_{w_3}(j))$ |
| 13 |      **else** |
| 14 |      Select a random search agent $(\overrightarrow{X_{rand}})$ |
| 15 |        $\vec{D} = |\vec{C} \cdot \overrightarrow{X_{rand}} - \overrightarrow{X_i}|$ |
| 16 |        $U_i(j) = \overrightarrow{X_{rand}}(j) - \vec{A} \cdot \vec{D}$ |
| 17 |      **end** if |
| 18 |    **else if**( $q > m$) |
| 19 |      **if** $p \leq 0.5$ |
| 20 |        $\vec{D} = |\vec{C} \cdot \overrightarrow{X^*} - \overrightarrow{X_i}|$ |
| 21 |        $U_i(j) = \overrightarrow{X^*}(j) - \vec{A} \cdot \vec{D}$ |
| 22 |      **elseif** $p > 0.5$ |
| 23 |        $\overrightarrow{D_p} = |\overrightarrow{X^*} - \overrightarrow{X_i}|$ |
| 24 |        $U_i(j) = \overrightarrow{X^*}(j) + \overrightarrow{D_p} \cdot e^{bl} \cdot \cos(2\pi l)$ |
| 25 |      **end** if |
| 26 |    **end** if |
| 27 |    Update the position using the Lévy flight disturbance strategy by Eq. (17) |
| 28 |    **end for** |
| 29 |    Correct the agent which is out of the boundary |
| 30 |    Calculate the fitness of $X_i$ and $U_i$ |
| 31 |    **if** $U_i$ is superior to $X_i$ |
| 32 |      Update the whale individuals, $X_i = U_i$ |
| 33 |      **if** $U_i$ is superior to $X^*$ |
| 34 |        Update the best whale individual, $X^* = U_i$ |
| 35 |      **end if** |
| 36 |    **end if** |
| 37 |    $t = t + 1$ |
| 38 | **end while** |

**Table 1** Comparison of MWOA with WOA, PSO, MFO, SOA, and SCA on CEC 2017 at $D = 30$

| Functions | Mean/Std | MWOA | WOA | PSO | MFO | SOA | SCA |
|---|---|---|---|---|---|---|---|
| F1 | Mean | **3.377E + 03** | 4.589E + 06 | 7.210E + 10 | 1.309E + 10 | 6.166E + 09 | 1.217E + 10 |
|  | Std | **2.476E + 03** | 4.619E + 06 | 6.842E + 08 | 7.011E + 09 | 2.449E + 09 | 1.819E + 09 |
| F3 | Mean | **1.038E-01** | 1.273E + 04 | 8.281E + 04 | 1.069E + 05 | 3.192E + 04 | 3.569E + 04 |
|  | Std | **4.823E-01** | 4.590E + 03 | 1.132E + 03 | 6.545E + 04 | 1.012E + 04 | 6.661E + 03 |
| F4 | Mean | **7.028E + 01** | 1.898E + 02 | 2.685E + 04 | 1.133E + 03 | 4.512E + 02 | 9.534E + 02 |
|  | Std | **4.630E + 01** | 5.073E + 01 | 3.767E + 02 | 1.047E + 03 | 1.343E + 02 | 2.344E + 02 |
| F5 | Mean | **4.704E + 01** | 2.173E + 02 | 4.494E + 02 | 2.153E + 02 | 1.540E + 02 | 2.763E + 02 |
|  | Std | **1.398E + 01** | 5.488E + 01 | 8.810E + 00 | 5.287E + 01 | 2.894E + 01 | 2.260E + 01 |
| F6 | Mean | **3.905E-01** | 6.470E + 01 | 9.091E + 01 | 4.052E + 01 | 3.102E + 01 | 4.951E + 01 |
|  | Std | **2.648E-01** | 1.031E + 01 | 2.319E + 00 | 1.150E + 01 | 4.876E + 00 | 4.671E + 00 |
| F7 | Mean | **8.436E + 01** | 6.235E + 02 | 7.324E + 02 | 4.695E + 02 | 4.096E + 02 | 4.240E + 02 |
|  | Std | 1.686E + 01 | 9.522E + 01 | **1.277E + 01** | 1.808E + 02 | 5.809E + 01 | 3.890E + 01 |
| F8 | Mean | **4.776E + 01** | 2.966E + 02 | 3.441E + 02 | 2.146E + 02 | 1.518E + 02 | 2.440E + 02 |
|  | Std | 1.167E + 01 | 6.997E + 01 | **9.549E + 00** | 4.531E + 01 | 2.765E + 01 | 2.054E + 01 |
| F9 | Mean | **3.360E + 01** | 7.792E + 03 | 9.354E + 03 | 7.676E + 03 | 4.242E + 03 | 4.782E + 03 |
|  | Std | **3.296E + 01** | 3.152E + 03 | 5.994E + 02 | 2.552E + 03 | 1.343E + 03 | 1.089E + 03 |
| F10 | Mean | **3.272E + 03** | 4.703E + 03 | 7.441E + 03 | 4.196E + 03 | 4.522E + 03 | 7.110E + 03 |
|  | Std | 6.548E + 02 | 7.857E + 02 | **1.642E + 02** | 7.027E + 02 | 8.777E + 02 | 3.351E + 02 |
| F11 | Mean | **9.247E + 01** | 3.861E + 02 | 1.096E + 08 | 7.834E + 03 | 1.115E + 03 | 1.052E + 03 |
|  | Std | **4.320E + 01** | 9.953E + 01 | 1.411E + 05 | 1.025E + 04 | 6.204E + 02 | 4.193E + 02 |
| F12 | Mean | **3.286E + 05** | 1.179E + 08 | 2.411E + 10 | 9.037E + 08 | 2.418E + 08 | 1.121E + 09 |
|  | Std | **2.110E + 05** | 5.882E + 07 | 3.685E + 08 | 1.392E + 09 | 5.031E + 08 | 2.836E + 08 |
| F13 | Mean | **1.437E + 04** | 9.640E + 04 | 3.487E + 10 | 5.232E + 07 | 5.553E + 07 | 3.761E + 08 |
|  | Std | **5.086E + 03** | 9.578E + 04 | 3.341E + 08 | 1.600E + 08 | 1.181E + 08 | 1.621E + 08 |
| F14 | Mean | **2.759E + 03** | 2.747E + 05 | 5.686E + 08 | 5.139E + 05 | 1.794E + 05 | 1.300E + 05 |
|  | Std | **2.163E + 03** | 2.158E + 05 | 1.257E + 06 | 1.096E + 06 | 1.304E + 05 | 8.845E + 04 |
| F15 | Mean | **3.685E + 03** | 5.154E + 04 | 4.126E + 09 | 7.135E + 06 | 4.987E + 06 | 1.356E + 07 |
|  | Std | **1.964E + 03** | 6.062E + 04 | 2.078E + 06 | 5.069E + 07 | 1.540E + 07 | 1.210E + 07 |
| F16 | Mean | **7.685E + 02** | 1.636E + 03 | 1.878E + 04 | 1.447E + 03 | 9.752E + 02 | 2.023E + 03 |
|  | Std | 2.676E + 02 | 4.076E + 02 | 2.974E + 02 | 4.344E + 02 | 2.514E + 02 | **2.121E + 02** |
| F17 | Mean | **2.433E + 02** | 9.121E + 02 | 1.094E + 05 | 6.980E + 02 | 6.421E + 02 | 7.220E + 02 |
|  | Std | 1.531E + 02 | 2.990E + 02 | 3.407E + 02 | 2.391E + 02 | 2.434E + 02 | **1.373E + 02** |
| F18 | Mean | **6.032E + 04** | 5.163E + 06 | 2.513E + 09 | 5.645E + 05 | 3.433E + 05 | 2.919E + 06 |
|  | Std | **3.393E + 04** | 5.023E + 06 | 2.628E + 07 | 2.152E + 06 | 3.111E + 05 | 1.384E + 06 |
| F19 | Mean | **3.186E + 03** | 6.508E + 05 | 4.299E + 09 | 6.431E + 05 | 3.396E + 05 | 2.446E + 07 |
|  | Std | **2.227E + 03** | 6.852E + 05 | 1.401E + 06 | 1.970E + 06 | 4.423E + 05 | 1.124E + 07 |
| F20 | Mean | **3.292E + 02** | 7.146E + 02 | 1.902E + 03 | 6.364E + 02 | 4.232E + 02 | 6.037E + 02 |
|  | Std | **1.211E + 02** | 1.826E + 02 | 1.973E + 01 | 2.168E + 02 | 1.351E + 02 | 1.186E + 02 |
| F21 | Mean | 2.477E + 02 | 1.781E + 02 | 8.866E + 02 | 7.467E + 02 | 4.270E + 02 | 4.526E + 02 |
|  | Std | **1.158E + 01** | 4.049E + 01 | 1.289E + 01 | 5.796E + 02 | 1.977E + 02 | 1.979E + 01 |
| F22 | Mean | **1.017E + 02** | 2.336E + 02 | 8.010E + 03 | 1.877E + 02 | 1.529E + 02 | 5.744E + 03 |
|  | Std | **2.387E + 00** | 4.452E + 01 | 1.708E + 02 | 4.609E + 01 | 3.232E + 01 | 2.607E + 03 |
| F23 | Mean | **4.004E + 02** | 8.682E + 02 | 3.934E + 03 | 6.695E + 02 | 6.390E + 02 | 6.936E + 02 |
|  | Std | **1.343E + 01** | 1.242E + 02 | 8.622E + 01 | 3.262E + 01 | 4.593E + 01 | 3.233E + 01 |
| F24 | Mean | **4.781E + 02** | 3.086E + 02 | 2.588E + 03 | 1.106E + 03 | 1.076E + 03 | 7.596E + 02 |
|  | Std | 1.970E + 01 | 3.337E + 02 | **1.449E + 01** | 4.239E + 01 | 1.045E + 02 | 3.352E + 01 |
| F25 | Mean | 3.961E + 02 | **2.245E + 02** | 4.556E + 03 | 1.112E + 03 | 9.350E + 02 | 7.031E + 02 |
|  | Std | **1.107E + 01** | 9.948E + 01 | 1.184E + 02 | 7.314E + 02 | 1.813E + 02 | 7.233E + 01 |

**Table 1** (continued)

| Functions | Mean/Std | MWOA | WOA | PSO | MFO | SOA | SCA |
|-----------|----------|------|-----|-----|-----|-----|-----|
| F26 | Mean | 1.661E + 03 | **1.435E + 03** | 1.120E + 04 | 4.025E + 03 | 3.424E + 03 | 4.404E + 03 |
| | Std | 1.803E + 02 | 2.309E + 03 | **1.387E + 02** | 5.381E + 02 | 3.753E + 02 | 2.815E + 02 |
| F27 | Mean | **5.416E + 02** | 1.246E + 03 | 5.695E + 03 | 9.038E + 02 | 8.187E + 02 | 7.010E + 02 |
| | Std | **1.355E + 01** | 1.913E + 02 | 1.064E + 02 | 7.469E + 01 | 5.500E + 01 | 3.675E + 01 |
| F28 | Mean | **3.472E + 02** | 4.129E + 02 | 6.097E + 03 | 2.355E + 03 | 2.480E + 03 | 1.036E + 03 |
| | Std | **5.512E + 01** | 5.182E + 02 | 8.105E + 01 | 3.861E + 02 | 6.310E + 01 | 1.356E + 02 |
| F29 | Mean | **6.725E + 02** | 1.476E + 03 | 7.902E + 04 | 1.199E + 03 | 9.751E + 02 | 1.781E + 03 |
| | Std | **1.263E + 02** | 4.500E + 02 | 2.214E + 03 | 2.813E + 02 | 1.972E + 02 | 2.425E + 02 |
| F30 | Mean | **1.296E + 04** | 2.082E + 06 | 7.861E + 09 | 1.612E + 06 | 9.085E + 06 | 7.579E + 07 |
| | Std | **4.699E + 03** | 2.421E + 06 | 7.973E + 07 | 2.780E + 06 | 1.990E + 07 | 2.821E + 07 |

# 4 Experimental results

## 4.1 CEC2017 benchmark suite

CEC 2017 benchmark suite [51] is used to test the performance of MWOA. It consists of 30 functions, which are separated into four groups: Unimodal (F1–F3); Simple Multimodal (F4–F10); Hybrid (F11–F20) and Composition (F21–F30). Different test functions are used to test various properties. Since F2 is unstable during the experiment using MATLAB software, this experiment used 29 test functions other than F2.

## 4.2 Experimental settings

MWOA is compared with five classical algorithms and four WOA variants. In this paper, the total number of evaluations is 300,000. The population size is 30 and $D$ is taken as 30. $F$ and $CR$ are 0.5 and 0.9, respectively, and $\alpha$ is 0.01. During the experiments, each algorithm runs 51 times to avoid the occasional error. The optimal value is preserved in each iteration. We use the mean error as the statistical index. We calculate the mean (Mean) and standard (Std) deviation of the data to reduce the statistical error. The same hardware and software platform is used for this experiment, and the running environment is Windows 10 operating system with Intel(R) Core(TM) i7-8565U CPU. MWOA and other algorithms are realized in MATLAB 2020(a).

## 4.3 Comparison of MWOA with other classic algorithms

We compare MWOA with WOA and four classic algorithms. Four algorithms include: PSO [7], MFO [8], SOA [52], SCA [53]. The 29 test functions are independently tested 51 times. The results from six algorithms are shown in Table 1. Bold data is the optimal value among six algorithms.

In terms of convergence accuracy, MWOA obtains the best results on 27 functions. WOA obtains optimal solutions on 2 functions. PSO, MFO, SOA and SCA have not achieved the best results on any function. MWOA is significantly better than PSO, MFO, SOA and SCA on all test functions, and it is the second best on F25 and F26 among all algorithms. The standard deviation of MWOA is the smallest on 22 functions, so the stability of MWOA is superior to other algorithms.

We further study the convergence capability of MWOA. The mean error of 51 independent runs is calculated. Figure 2 is the convergence curve, showing the iteration trend on some selected functions. The ordinate represents the logarithm of the average function error value, and the abscissa represents the iteration number.

As seen in Fig. 2, the searchability of MWOA is better. MWOA has a good convergence speed and better convergence accuracy than WOA, PSO, MFO, SOA and SCA, showing a strong convergence ability in early iterations. The convergence accuracy of MWOA is superior to other algorithms on most functions. On F1, F5-F9, F12, F19, F20 and F29, the MWOA has the fastest convergence speed. Overall, MWOA can always get better solutions when handling 30-dimensional problems.

According to the figure, the strategies applied by MWOA have different effects on the algorithm. MWOA shows a faster convergence in early iterations, and it can save time for convergence to the optimal solution. This is the result of using elite reverse learning strategy. MWOA remains stable in mid-iteration. Exploration and exploitation are balanced because of the improvement of nonlinear parameters. The application of the mutation strategy and Lévy flight disturbance strategy makes the algorithm continue to search globally in the later stage to avoid local optimum. Using multiple strategies enables MWOA to
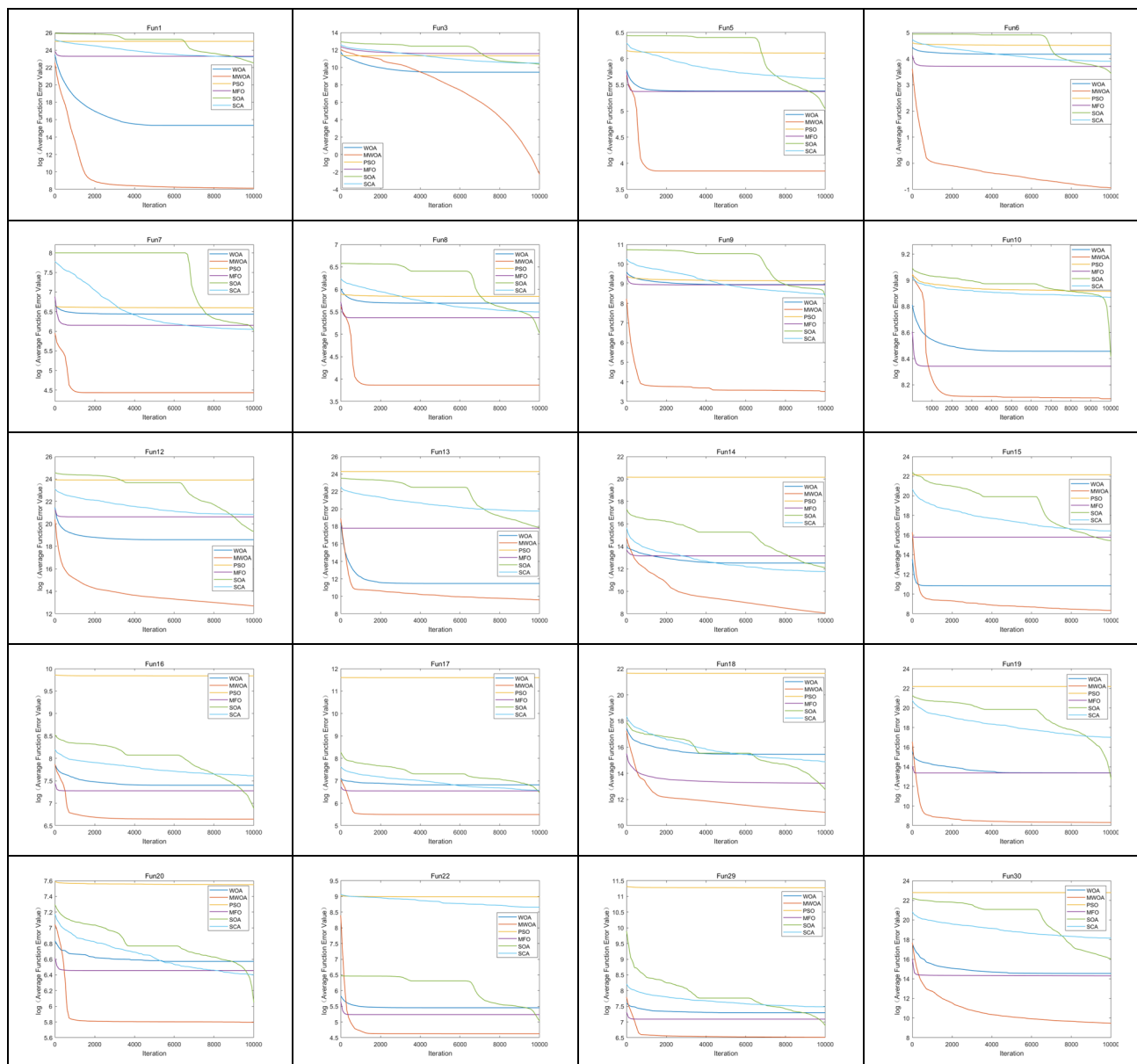
**Fig. 2** Convergence curves of WOA, PSO, MFO, SOA, SCA and MWOA on 29 test functions

obtain better convergence performance than other algorithms.

## 4.4 Comparison of MWOA with other WOA variants

In this part, We compare MWOA with other WOA variants, which are MSWOA [54], IWOA [55], CWOA [18] and OBWOA [22]. These four algorithms are representative improvement algorithms, which can improve the original WOA from different perspectives and use different mechanisms, and the improvement is relatively good. So they have better comparison value. MSWOA is proposed

by Yang, using chaotic initialization, adaptive weights and EPD strategy. IWOA is proposed by Li and Luo with a nonlinear adjustment strategy for cosine control parameters and Gaussian perturbation operator. CWOA introduces chaotic logical sequences during population initialization, which is proposed by Kaur and Arora. OBWOA employs an Opposition-based Learning strategy. For fairness, MWOA and other algorithms run independently 51 times on the CEC2017 benchmark suite. The mean error is calculated as an evaluation index. Experimental results of MWOA, MSWOA, IWOA, CWOA and OBWOA are demonstrated in Table 2. Bold data reveals the optimal value between MWOA and other algorithms.

**Table 2** Comparison of MWOA with MSWOA, IWOA, CWOA and OBWOA on CEC2017 at $D = 30$

| Functions | Mean/Std | MWOA | MSWOA | IWOA | CWOA | OBWOA |
|---|---|---|---|---|---|---|
| F1 | Mean | **3.377E + 03** | 4.002E + 08 | 1.175E + 10 | 4.863E + 06 | 2.764E + 06 |
|  | Std | **2.476E + 03** | 3.971E + 07 | 5.174E + 09 | 4.267E + 06 | 1.942E + 06 |
| F3 | Mean | **1.038E-01** | 5.628E + 04 | 3.330E + 04 | 1.280E + 04 | 1.386E + 05 |
|  | Std | **4.823E-01** | 8.780E + 03 | 7.231E + 03 | 5.316E + 03 | 4.868E + 04 |
| F4 | Mean | **7.028E + 01** | 1.806E + 02 | 6.110E + 02 | 1.750E + 02 | 1.422E + 02 |
|  | Std | 4.630E + 01 | 5.024E + 01 | 3.043E + 02 | 3.786E + 01 | **3.889E + 01** |
| F5 | Mean | **4.704E + 01** | 2.695E + 02 | 3.330E + 04 | 2.187E + 02 | 2.788E + 02 |
|  | Std | **1.398E + 01** | 4.757E + 01 | 7.231E + 03 | 5.184E + 01 | 6.090E + 01 |
| F6 | Mean | **3.905E-01** | 6.692E + 01 | 3.091E + 01 | 6.322E + 01 | 6.779E + 01 |
|  | Std | **2.648E-01** | 7.311E + 00 | 5.571E + 00 | 1.012E + 01 | 1.059E + 01 |
| F7 | Mean | **8.436E + 01** | 5.896E + 02 | 4.153E + 02 | 6.732E + 02 | 5.498E + 02 |
|  | Std | **1.686E + 01** | 7.688E + 01 | 8.152E + 01 | 1.137E + 02 | 8.896E + 01 |
| F8 | Mean | **4.776E + 01** | 2.146E + 02 | 1.752E + 02 | 3.019E + 02 | 2.102E + 02 |
|  | Std | **1.167E + 01** | 3.925E + 01 | 2.390E + 01 | 6.536E + 01 | 4.783E + 01 |
| F9 | Mean | **3.360E + 01** | 6.606E + 03 | 3.979E + 03 | 7.952E + 03 | 7.328E + 03 |
|  | Std | **3.296E + 01** | 9.984E + 02 | 9.475E + 02 | 3.017E + 03 | 3.102E + 03 |
| F10 | Mean | **3.272E + 03** | 5.545E + 03 | 3.880E + 03 | 5.015E + 03 | 5.154E + 03 |
|  | Std | 6.548E + 02 | **2.028E + 02** | 6.547E + 02 | 1.056E + 03 | 7.252E + 02 |
| F11 | Mean | **9.247E + 01** | 5.675E + 02 | 1.410E + 03 | 3.869E + 02 | 4.295E + 02 |
|  | Std | **4.320E + 01** | 1.816E + 02 | 8.781E + 02 | 1.157E + 02 | 3.581E + 02 |
| F12 | Mean | **3.286E + 05** | 1.160E + 08 | 3.154E + 08 | 1.429E + 08 | 3.285E + 07 |
|  | Std | **2.110E + 05** | 2.508E + 07 | 5.785E + 08 | 6.871E + 07 | 2.572E + 07 |
| F13 | Mean | **1.437E + 04** | 8.995E + 06 | 3.960E + 07 | 6.457E + 08 | 1.400E + 05 |
|  | Std | **5.086E + 03** | 3.816E + 06 | 5.788E + 07 | 4.610E + 09 | 7.619E + 04 |
| F14 | Mean | **2.759E + 03** | 3.780E + 05 | 3.960E + 07 | 2.458E + 05 | 8.650E + 05 |
|  | Std | **2.163E + 03** | 1.523E + 05 | 5.788E + 07 | 1.628E + 05 | 9.616E + 05 |
| F15 | Mean | **3.685E + 03** | 1.199E + 06 | 8.691E + 05 | 3.976E + 04 | 6.321E + 04 |
|  | Std | **1.964E + 03** | 3.427E + 05 | 6.058E + 06 | 4.719E + 04 | 4.642E + 04 |
| F16 | Mean | **7.685E + 02** | 1.644E + 03 | 9.359E + 02 | 1.753E + 03 | 1.904E + 03 |
|  | Std | 2.676E + 02 | 4.037E + 02 | **2.515E + 02** | 5.866E + 02 | 4.119E + 02 |
| F17 | Mean | **2.433E + 02** | 8.907E + 02 | 3.827E + 02 | 9.811E + 02 | 8.609E + 02 |
|  | Std | 1.531E + 02 | 3.865E + 02 | **1.420E + 02** | 3.139E + 02 | 2.631E + 02 |
| F18 | Mean | **6.032E + 04** | 1.202E + 07 | 4.901E + 05 | 4.492E + 06 | 2.423E + 06 |
|  | Std | **3.393E + 04** | 7.247E + 06 | 1.058E + 06 | 3.405E + 06 | 2.548E + 06 |
| F19 | Mean | **3.186E + 03** | 5.084E + 06 | 7.404E + 05 | 5.085E + 05 | 2.889E + 06 |
|  | Std | **2.227E + 03** | 5.648E + 06 | 4.910E + 06 | 4.748E + 05 | 2.189E + 06 |
| F20 | Mean | **3.292E + 02** | 7.667E + 02 | 3.879E + 02 | 8.300E + 02 | 6.738E + 02 |
|  | Std | 1.211E + 02 | 1.872E + 02 | **1.114E + 02** | 1.877E + 02 | 1.792E + 02 |
| F21 | Mean | 2.477E + 02 | 4.688E + 02 | 6.371E + 02 | **1.018E + 02** | 4.718E + 02 |
|  | Std | **1.158E + 01** | 4.814E + 01 | 4.982E + 02 | 1.222E + 01 | 6.586E + 01 |
| F22 | Mean | **1.017E + 02** | 5.447E + 03 | 1.596E + 02 | 1.017E + 02 | 4.680E + 03 |
|  | Std | **2.387E + 00** | 4.137E + 02 | 2.542E + 01 | 1.171E + 01 | 1.960E + 03 |
| F23 | Mean | **4.004E + 02** | 8.524E + 02 | 7.130E + 02 | 8.660E + 02 | 7.489E + 02 |
|  | Std | **1.343E + 01** | 1.378E + 02 | 5.540E + 01 | 1.541E + 02 | 9.856E + 01 |
| F24 | Mean | 4.781E + 02 | 7.278E + 02 | 1.134E + 03 | **3.324E + 02** | 7.790E + 02 |
|  | Std | **1.970E + 01** | 3.596E + 01 | 1.180E + 02 | 3.669E + 02 | 8.714E + 01 |
| F25 | Mean | 3.961E + 02 | 4.652E + 02 | 9.218E + 02 | **2.184E + 02** | 4.436E + 02 |
|  | Std | 1.107E + 01 | **4.411E + 00** | 2.195E + 02 | 9.214E + 01 | 2.703E + 01 |

**Table 2** (continued)

| Functions | Mean/Std | MWOA | MSWOA | IWOA | CWOA | OBWOA |
|---|---|---|---|---|---|---|
| F26 | Mean | 1.661E + 03 | 5.729E + 03 | 3.634E + 03 | **1.448E + 03** | 4.604E + 03 |
| | Std | **1.803E + 02** | 7.819E + 02 | 9.534E + 02 | 2.324E + 03 | 1.309E + 03 |
| F27 | Mean | 5.416E + 02 | **5.000E + 02** | 9.621E + 02 | 1.260E + 03 | 6.545E + 02 |
| | Std | 1.355E + 01 | **1.180E-05** | 9.249E + 01 | 2.105E + 02 | 1.007E + 02 |
| F28 | Mean | **3.472E + 02** | 5.000E + 02 | 2.137E + 03 | 6.266E + 02 | 5.076E + 02 |
| | Std | 5.512E + 01 | **1.416E-04** | 5.875E + 02 | 1.661E + 03 | 2.404E + 01 |
| F29 | Mean | **6.725E + 02** | 1.929E + 03 | 6.867E + 02 | 1.228E + 03 | 1.923E + 03 |
| | Std | **1.263E + 02** | 3.560E + 02 | 2.001E + 02 | 4.444E + 02 | 3.984E + 02 |
| F30 | Mean | **1.296E + 04** | 3.025E + 07 | 9.885E + 05 | 2.108E + 06 | 1.064E + 07 |
| | Std | **4.699E + 03** | 2.236E + 07 | 4.002E + 06 | 2.319E + 06 | 8.452E + 06 |
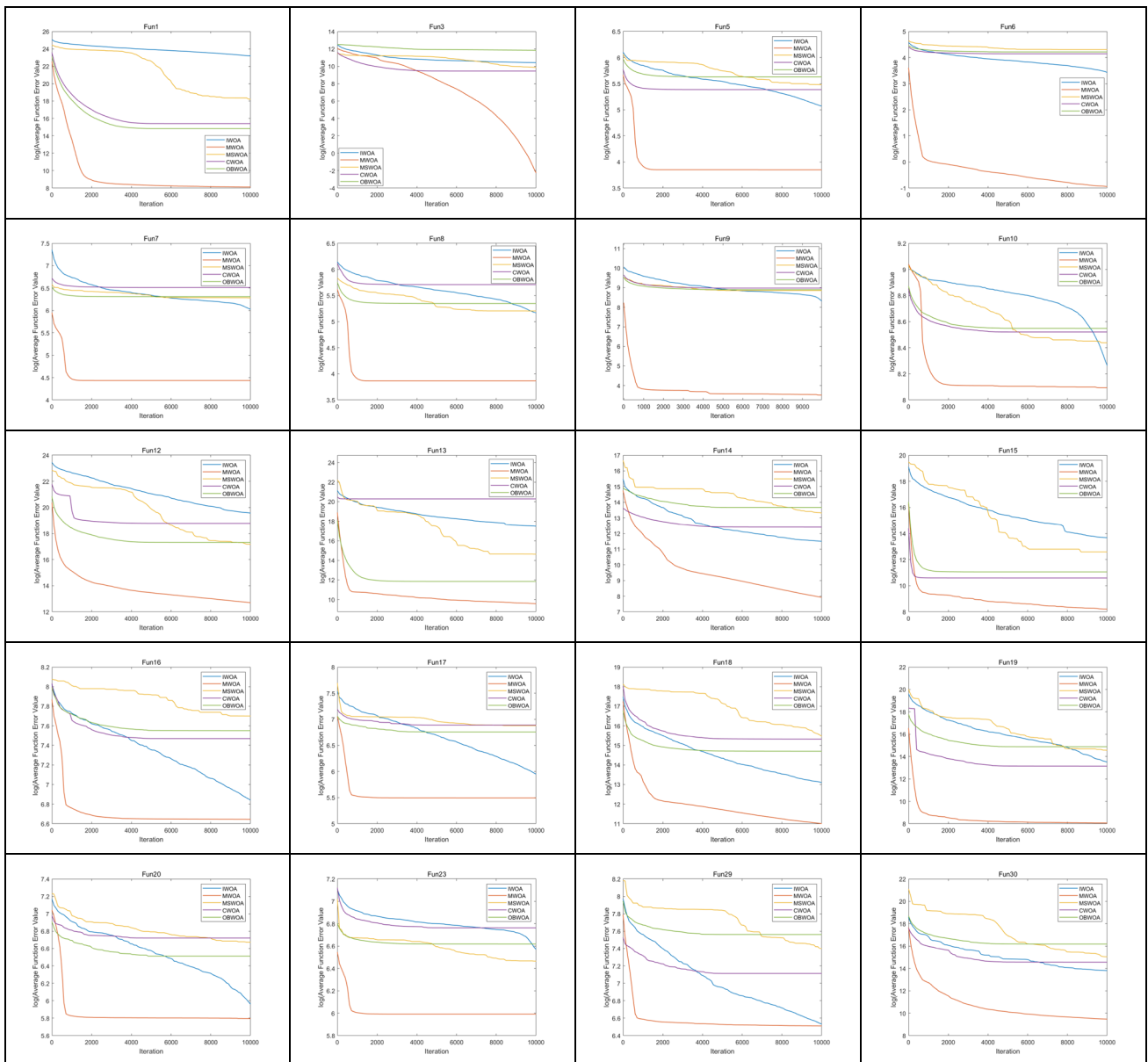


**Fig. 3** Convergence curves of MSWOA, IWOA, CWOA, OBWOA and MWOA on 29 test functions

**Table 3** The statistical results based on Wilcoxon's signed rank test (D = 30)

| MWOA vs | $R^+$ | $R^-$ | P-value | $\alpha = 0.05$ |
|---------|-------|-------|---------|-----------------|
| MSWOA | 434 | 1 | 0.000002849 | Yes |
| IWOA | 435 | 0 | 0.000002563 | Yes |
| CWOA | 427 | 8 | 0.000028631 | Yes |
| OBWOA | 435 | 0 | 0.000002563 | Yes |
| WOA | 409 | 26 | 0.000034603 | Yes |
| PSO | 435 | 0 | 0.000002563 | Yes |
| MFO | 435 | 0 | 0.000002563 | Yes |
| SOA | 435 | 0 | 0.000002563 | Yes |
| SCA | 435 | 0 | 0.000002563 | Yes |

According to Table 2, MWOA obtains the best solutions on 24 functions. CWOA acquires the optimal solutions on four functions which are F21, F24, F25 and F26. MSWOA gets the optimal solutions on F27. IWOA and OBWOA do not obtain the optimal solutions on any functions. Moreover, the results of MWOA on F21, F24, F25, F26 and F27 are the second-best among all algorithms. MWOA can generally achieve better results in dealing with various global problems. In terms of stability, MWOA has the slightest standard deviation on 21 functions, and the performance is relatively stable and superior to other WOA variants. The embedding of the mutation strategy strengthens the exploration ability and greatly improves the convergence accuracy. In general, the strategies used by MWOA greatly improve the convergence accuracy and stability, which is better than other WOA variants to a certain extent.

We make the convergence curves to study the performance of MWOA. Figure 3 is the convergence curves of MWOA, MSWOA, IWOA, CWOA and OBWOA over 51 independent runs on the selected functions.

We can conclude that MWOA performs best than other four WOA variants. MWOA has apparent advantages in the convergence speed on F1, F5–F12, F16–F20, F23 and F30. MWOA exhibits a good convergence speed at the beginning of the iteration. Then it keeps exploration and exploitation stable. There is a tendency to continue to accelerate in the later iterations, especially on F9, F14 and F18. It shows that the algorithm continues to perform a global search at the later iterative stage, which helps avoid local optimum. Lévy flight is used for updating the position, which increases the search step and accelerates the algorithm convergence. It can strengthen the global search ability. Moreover, it enables the algorithm to find the best solution rapidly and effectively. In general, MWOA is better than MSWOA, IWOA, CWOA and OBWOA in

terms of the improvement effect. MWOA can consistently achieve better results when solving global optimization problems.

In addition to the above experiments, MATLAB and EXCEL software are used to perform Wilcoxon's signed rank test on MWOA and other nine algorithms. The experimental results are listed in Table 3. $R^+$ represents positive rank sum, $R^-$ represents negative rank sum. The P-value is a parameter used to determine the results of the hypothesis test and indicates the significant difference between MWOA and other algorithms. The $R^+$ value of MWOA is larger than $R^-$ value in all comparisons. The value of the $R^-$ is 0 in contrast with IWOA, PSO, MFO, SOA, and SCA. At the confidence level of $\alpha = 0.05$, the P-value is much less than 0.05. So MWOA significantly differs from other algorithms. Through the experimental results, MWOA is superior to other algorithms when handling 30-dimensional problems.

## 5 Conclusion

In WOA, the individual generally initializes the position randomly. Since the optimal solution cannot be obtained easily, random initialization will increase the search range of feasible solutions and search time. The linear convergence factor makes the WOA encounter the phenomenon of unbalanced exploration and exploitation in optimization. It tends to converge prematurely and fall into local optimum. We propose a modified WOA with multi-strategy mechanism to solve the existing problems of WOA and improve its performance. The innovations of MWOA are as follows:

1. MWOA introduces the elite reverse learning strategy to obtain a better population, which can enhance population diversity and speed up convergence.
2. The nonlinear convergence factor is introduced to maintain the balance between exploration and exploitation.
3. Using mutation strategy can strengthen the global exploration capability. Furthermore, the ability to mine the optimal solution is effectively improved, so it can prevent the algorithm from falling into local optimum.
4. The application of the Lévy flight disturbance strategy can increase the search range of the algorithm space. It is beneficial for finding the global optimal solution. After updating the location, the Lévy fight strategy perturbs the population position to disperse the population. This operation can avoid premature convergence.

The CEC2017 benchmark suite is used to test the performance of MWOA. MWOA is compared with WOA, PSO, MFO, SOA, SCA, and other four WOA variants. It has been proved by many experiments that MWOA has strong competitiveness. The proposed algorithm has noticeable improvements in convergence speed and accuracy. It can well balance exploration and exploitation to avoid local optimum. In general, MWOA demonstrates outstanding performance and has great competitiveness in dealing with optimization problems.

Although this paper overcomes some shortcomings of WOA and enhances the performance of WOA to some extent, the global theoretical optimum is not found for some functions. In addition, there are some parameters in MWOA, which need to be adjusted. The algorithm proposed in this paper needs to be applied to solve practical problems. In the next work, we will apply WOA to solveengineering optimization problems.

**Author contributions** Mingyuan Li: Conceptualization, Methodology, Writing- Original draft preparation. Xiaobing Yu: Reviewing and Editing. Bingbing Fu: Data curation, Investigation. Xuming Wang: Editing. Xianrui Yu: Editing.

**Data availability** The data that support the findings of this study are not openly available the university's data sharing guidelines but are available from the corresponding author upon reasonable request.

## Declarations

**Conflict of interest** The authors declare that there is no conflict of interest.

## References

1. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
2. Yang B et al (2020) Comprehensive overview of meta-heuristic algorithm applications on PV cell parameter identification. Energy Convers Manage 208:112595
3. Wu Y (2021) A survey on population-based meta-heuristic algorithms for motion planning of aircraft. Swarm Evol Comput 62:100844
4. Lu P et al (2021) Review of meta-heuristic algorithms for wind power prediction: methodologies, applications and challenges. Appl Energy 301:117446
5. Hu G et al (2022) An enhanced manta ray foraging optimization algorithm for shape optimization of complex CCG-Ball curves. Knowl-Based Syst 240:108071
6. Mirjalili S (2019) Genetic algorithm. In: Evolutionary algorithms and neural networks. Springer, pp 43–55
7. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. IEEE
8. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249
9. Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B (Cybernetics) 26(1): 29–41
10. Hu G et al (2022) An enhanced chimp optimization algorithm for optimal degree reduction of Said-Ball curves. Math Comput Simul 197:207–252
11. Basturk B (2006) An artificial bee colony (ABC) algorithm for numeric function optimization. In: IEEE swarm intelligence symposium, Indianapolis, IN, USA
12. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
13. Gharehchopogh FS, Gholizadeh H (2019) A comprehensive survey: whale optimization algorithm and its applications. Swarm Evol Comput 48:1–24
14. Nasiri J, Khiyabani FM (2018) A whale optimization algorithm (WOA) approach for clustering. Cogent Math Stat 5(1):1483565
15. Cui D (2017) Application of whale optimization algorithm in reservoir optimal operation. Adv Sci Technol Water Resour 37(3):72–79
16. Pham Q-V et al (2020) Whale optimization algorithm with applications to resource allocation in wireless networks. IEEE Trans Veh Technol 69(4):4285–4297
17. Srivastava V, Srivastava S (2019) Whale optimization algorithm (WOA) based control of nonlinear systems. In: 2019 2nd International conference on power energy, environment and intelligent control (PEEIC). IEEE
18. Kaur G, Arora S (2018) Chaotic whale optimization algorithm. J Comput Des Eng 5(3):275–284
19. Zhong M, Long W (2017) Whale optimization algorithm with nonlinear control parameter. In: MATEC web of conferences. 2017. EDP Sciences
20. Li S, Luo X, Wu L (2021) An improved whale optimization algorithm for locating critical slip surface of slopes. Adv Eng Softw 157:103009
21. Kushwah R, Kaushik M, Chugh K (2021) A modified whale optimization algorithm to overcome delayed convergence in artificial neural networks. Soft Comput 25(15):10275–10286
22. Abd Elaziz M, Oliva D (2018) Parameter estimation of solar cells diode models by an improved opposition-based whale optimization algorithm. Energy conversion and management, 171: 1843–1859
23. Zhang J, Wang J-S (2020) Improved whale optimization algorithm based on nonlinear adaptive weight and golden sine operator. IEEE Access 8:77013–77048
24. Chen H et al (2020) An efficient double adaptive random spare reinforced whale optimization algorithm. Expert Syst Appl 154:113018
25. Liu J, et al (2022) A novel enhanced global exploration whale optimization algorithm based on Lévy flights and judgment mechanism for global continuous optimization problems. Eng Comput, 1–29
26. Chakraborty S, et al. (2022) A novel improved whale optimization algorithm to solve numerical optimization and real-world applications. Artif Intell Rev, 1–112
27. Nadimi-Shahraki MH, Zamani H, Mirjalili S (2022) Enhanced whale optimization algorithm for medical feature selection: a COVID-19 case study. Comput Biol Med 148:105858
28. Trivedi IN, et al (2018) A novel hybrid PSO–WOA algorithm for global numerical functions optimization. In: Advances in computer and computational sciences. Springer, p 53–60

29. Mohammed H, Rashid T (2020) A novel hybrid GWO with WOA for global numerical optimization and solving pressure vessel design. Neural Comput Appl 32(18):14701–14718

30. Kaveh A, Rastegar Moghaddam M (2018) A hybrid WOA-CBO algorithm for construction site layout planning problem. Sci Iranica 25(3): 1094–1104

31. Mostafa Bozorgi S, Yazdani S (2019) IWOA: an improved whale optimization algorithm for optimization problems. J Comput Des Eng 6(3): 243–259

32. Bentouati B, Chaib L, Chettih S (2016) A hybrid whale algorithm and pattern search technique for optimal power flow problem. In: 2016 8th international conference on modelling, identification and control (ICMIC). IEEE

33. Tang C et al (2022) A hybrid whale optimization algorithm with artificial bee colony. Soft Comput 26(5):2075–2097

34. Dey B, Bhattacharyya B (2022) Comparison of various electricity market pricing strategies to reduce generation cost of a microgrid system using hybrid WOA-SCA. Evol Intel 15(3):1587–1604

35. Chakraborty S et al (2021) A novel enhanced whale optimization algorithm for global optimization. Comput Ind Eng 153:107086

36. Jin Q, Xu Z, Cai W (2021) An improved whale optimization algorithm with random evolution and special reinforcement dual-operation strategy collaboration. Symmetry 13(2):238

37. Yuan X et al (2020) Multi-strategy ensemble whale optimization algorithm and its application to analog circuits intelligent fault diagnosis. Appl Sci 10(11):3667

38. Sun G et al (2022) An improved whale optimization algorithm based on nonlinear parameters and feedback mechanism. Int J Comput Intell Syst 15(1):1–17

39. Li X, et al (2022) A multi-strategy hybrid adaptive whale optimization algorithm. J Comput Des Eng

40. Xiao Z-Y, Liu S (2019) Study on elite opposition-based golden-sine whale optimization algorithm and its application of project optimization. Acta Electon Sin 47(10): 2177

41. Tang C, et al (2019) A hybrid improved whale optimization algorithm. In: 2019 IEEE 15th international conference on control and automation (ICCA). IEEE

42. Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359

43. Xue Y, Tong Y, Neri F (2022) An ensemble of differential evolution and Adam for training feed-forward neural networks. Inf Sci 608:453–471

44. Chakraborty S, et al (2021) A hybrid whale optimization algorithm for global optimization. J Ambient Intell Human Comput, 1–37

45. Hu G et al (2022) An enhanced black widow optimization algorithm for feature selection. Knowl-Based Syst 235:107638

46. Yang X-S, Deb S (2009) Cuckoo search via Lévy flights. In: 2009 World congress on nature and biologically inspired computing (NaBIC). IEEE

47. Houssein EH et al (2020) Lévy flight distribution: a new meta-heuristic algorithm for solving engineering optimization problems. Eng Appl Artif Intell 94:103731

48. Liu M, Yao X, Li Y (2020) Hybrid whale optimization algorithm enhanced with Lévy flight and differential evolution for job shop scheduling problems. Appl Soft Comput 87:105954

49. Mohseni S et al (2021) Lévy-flight moth-flame optimisation algorithm-based micro-grid equipment sizing: an integrated investment and operational planning approach. Energy AI 3:100047

50. Mohiz MJ et al (2021) Application mapping using cuckoo search optimization with Lévy flight for NoC-based system. IEEE Access 9:141778–141789

51. Mzanh A, et al (2016) Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization

52. Dhiman G, Kumar V (2019) Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. Knowl-Based Syst 165:169–196

53. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133

54. Yang W et al (2022) A multi-strategy Whale optimization algorithm and its application. Eng Appl Artif Intell 108:104558

55. Li Y et al (2019) An adaptive whale optimization algorithm using Gaussian distribution strategies and its application in heterogeneous UCAVs task allocation. IEEE Access 7:110138–110158