IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 17, 2019, accepted March 6, 2019, date of publication March 14, 2019, date of current version April 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2905009

Whale Optimization Algorithm Based on Lamarckian Learning for Global Optimization Problems

QIANG ZHANG[1] AND LIJIE LIU[2]
[1]College of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China
[2]College of Electrical and Information, Heilongjiang Bayi Agricultural University, Daqing 163319, China

Corresponding author: Qiang Zhang (dqpi_zq@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702093, and in part by the Natural Science Foundation of Heilongjiang Province under Grant 2018003.

ABSTRACT Whale optimization algorithm (WOA) is a population-based meta-heuristic imitating the hunting behavior of humpback whales, which has been successfully applied to solve many real-world problems. Although WOA has a good convergence rate, it cannot achieve good results in finding the global optimal solution of high-dimensional complex optimization problems. The learning mechanism of Lamarckian evolutionism has the advantages of speeding up and strengthening local search. Through this learning mechanism, solutions with certain conditions can acquire higher adaptability with a higher probability by active learning. To enhance the global convergence speed and get better performance, this paper presents a WOA based on Lamarckian learning (WOALam) for solving high-dimensional function optimization problems. First, the population is initialized by good point set theory so that individuals can be evenly distributed in the solution space. Second, the upper confidence bound algorithm is used to calculate the development potential of the individual. Finally, based on the evolutionary theory of Lamarck, individuals with more development potentials are selected to perform the local enhanced search to improve the performance of the algorithm. The WOALam was compared with six variants of WOA on 44 benchmark functions. The experiments proved that the proposed algorithm can balance the global exploring ability and the exploiting ability well. It could obtain better results with fewer iterations and had good convergence speed and accuracy.

INDEX TERMS Whale optimization algorithm, Lamarckian learning, good point set, upper confidence bound, optimization.

I. INTRODUCTION

Traditional optimization algorithms are based on gradient information when solving such problems, which is only applicable to the case where the objective function and constraint conditions are differentiable. With the increasing complexity of optimization problems, traditional optimization algorithms become more and more powerless. In recent years, a class of optimization algorithm based on iterative process has been widely studied. Compared with traditional optimization algorithm, meta-heuristic algorithm is a group-based search technology and is not restricted by whether the search space is continuous or differentiable. It has the characteristics of simple operation, strong versatility and parallel processing. Metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring (global search) and exploiting (local search) the search space [1]. Many new swarm intelligence algorithms have been proposed by scholars in recent years. These algorithms have the characteristics of fewer parameters, relatively simple evolutionary process and fast operation speed. Such as: Magnetic Charged System Search (MCSS) [2], Dolphin Echolocation Optimization (DEO) [3], Colliding Bodies Optimization (CBO) [4], Chicken Swarm Optimization (CSO) [5], Water Evaporation Optimization(WEO) [6], Cuttlefish optimization algorithm (CFA) [7], Gray Wolf Optimizer(GSO) [8], Fruit Fly Optimization Algorithm(FFOA) [9], Multi-Verse Optimizer (MVO) [10], etc.

In 2016, a new intelligent optimization algorithm, called Whale Optimization Algorithm (WOA), was proposed by Mirjalili and Lewis [11]. WOA was developed through

The associate editor coordinating the review of this manuscript and approving it for publication was Bora Onat.

simulating the hunting behavior of humpback whales to update the location of populations for finding the optimal in search space. It has the advantages of simple concept, less adjustable parameters and easy implementation. The superiority of WOA over Particle Swarm Optimization (PSO) [12], Gravitational Search Algorithm (GSA) [13], Differential Evolution (DE) [14] and Fast Evolutionary Programing (FEP) [15] has been demonstrated. Owing to its impressive advantages such as easy implementation, lesser adjustable parameters and quick convergence, the WOA has been successfully applied to diverse problems. For example, 0-1 knapsack problem [16], the permutation flow shop scheduling problem [17], clustering and classification [18], [19], optimal control problems [20], [21], Multi-objective optimization [22], [23], routing optimization [24], support vector machines and neural networks [25]–[27], Feature selection [28], economic load dispatch problems [29], image segmentation [30], fuzzy controller [31], and parameter estimation [32].

In original WOA algorithm, the current best individual is the target prey and the other individuals attempt to modify their positions towards this best individual in each iteration process. This update process may lead the algorithm into local optimum. Researchers have proposed some variants of WOA to enhance the convergence capability of algorithm when dealing with global optimization problems in recent years. Trivedi *et al.* [33] proposed an adaptive whale optimization algorithm (AWOA). The results show that compared with the standard WOA optimization algorithm, the AWOA has better competitive performance. Hu *et al.* [34] proposed an improved whale optimization algorithm based on inertia weight (IWOA) to adjust the impact on the current optimal solution. Bentouati *et al.* [35] proposed a new power system planning strategy by combining Whale Optimization Algorithm (WOA) with pattern search algorithm (PS). Simulation results clearly reveal the effectiveness and the rapidity of the proposed algorithm for solving the OPF problem. Abdel-Basset *et al.* [36] proposed a memetic algorithm using the Whale optimization Algorithm (WA) Integrated with a Tabu Search (WAITS) for solving Quadratic Assignment Problem. Ling *et al.* [37] proposed an improved version of the WOA (LWOA) based on Lévy flight trajectory. The simulation results show that the LWOA is feasible and effective, and has superior approximation capabilities for global optimization problems with dimension up to 50. Sun *et al.* [38] proposed a modified whale optimization algorithm (MWOA) for solving LSGO problems, Lévy-flight strategy and quadratic interpolation method is applied to enhance the local exploitation ability and improve the solution accuracy. Trivedi *et al.* [39] proposed a hybrid PSO-WOA which is a combination of PSO used for exploitation phase and WOA for exploration phase in uncertain environment, analysis of competitive results obtained from PSO-WOA validates its effectiveness compared to standard PSO and WOA algorithm. Yan *et al.* [23] proposed an ameliorative whale optimization algorithm(AWOA), the logistic map is used to obtain the initial solution and inertia weight is introduced in the WOA to improve the algorithm, the problem of water resource allocation optimization is well solved by the AWOA. Sun *et al.* [40] proposed an improved whale optimization algorithm based on different searching paths and perceptual disturbance(PDWOA), the effect of several other spiral curves on the performance of the algorithm has been verified , the simulation results also proved that the performance of the equal-pitch Archimedean spiral curve is superior to other types of spiral curve. Abd Elaziz and Oliva [32] proposed an improved opposition-based whale optimization algorithm(OBWOA) ,the proposed method has been tested over different benchmark optimization functions to verify its exploration capabilities and has also been applied to estimate the parameters of solar cells using three different diode models. Kaveh and Ilchi Ghazaan [41] proposed an enhanced whale optimization algorithm (EWOA). Mafarja *et al.* [42] proposed a hybrid WOA-SA algorithm, the SA was used to enhance the exploitation by searching the most promising regions located by WOA algorithm. Kaur and Arora [43] proposed an chaotic whale optimization algorithm (CWOA). The experimental results show that logistic, cubic, sine, sinusoidal, singer, tent, piecewise and gauss/mouse chaotic maps are able to enhance the performance of WOA algorithm successfully. Abdel-Basset *et al.* [44], [45] proposed improved Lévy-based whale optimization algorithm (ILWOA) which parameters a, A, L are updated by Levy distribution and p is generated by logistic chaos map. The 1-D bin packing problems and bandwidth-efficient virtual machine placement are well solved by the proposed algorithm.

Although WOA has been greatly improved, it still has several problems, such as slow convergence and deteriorating optimization capability when dealing with complex high-dimensional problems. The essence of these strategies of the above literature is to improve the individual evolutionary mode by balance exploring (global search) and exploiting (local search). However, it is found that the invariance of individual evolutionary strategy is also an important reason for the low efficiency of the algorithm in the evolutionary process. For example, the individual update is subjected to randomly selected or the current best individual of whale population in the evolutionary process , the inheritance of past group experience is missing and has not been effectively utilized . For multimodal function optimization problems, if the current best individual falls into local optimum, the probability of the whole population falling into local optimum will increase. In this paper, Whale Optimization Algorithm based on Lamarckian learning (WAOLam) is proposed. First, the population is initialized by good point set theory so that individuals can be evenly distributed in the solution space. Second, the upper confidence bound algorithm is used to calculate development potential of the individual. Finally, based on evolutionary theory of lamarck, individuals with more development potentials are selected to perform local enhanced search to improve the performance of the algorithm.

Algorithm 1 The Procedure of the Original WOA

Initialize the whales population X_i (i = 1, 2, ..., n)
Calculate the fitness of each search agent
X^* = the best search agent
while (t < maximum number of iterations)
for each search agent
Update a, A, C, l, and p
if1 (p<0.5)
if2 (|A| < 1)
Update the position of the current search by the Eq. (1)
else if2 (|A| >= 1)
Update the position of the current search by the Eq. (9)
end if2
else if1 (p>=0.5)
Update the position of the current search by the Eq. (5)
end if1
end for
Check if any search agent goes beyond the search space and amend it
Calculate the fitness of each search agent
Update X^* if there is a better solution
t=t+1
end while
return X^*

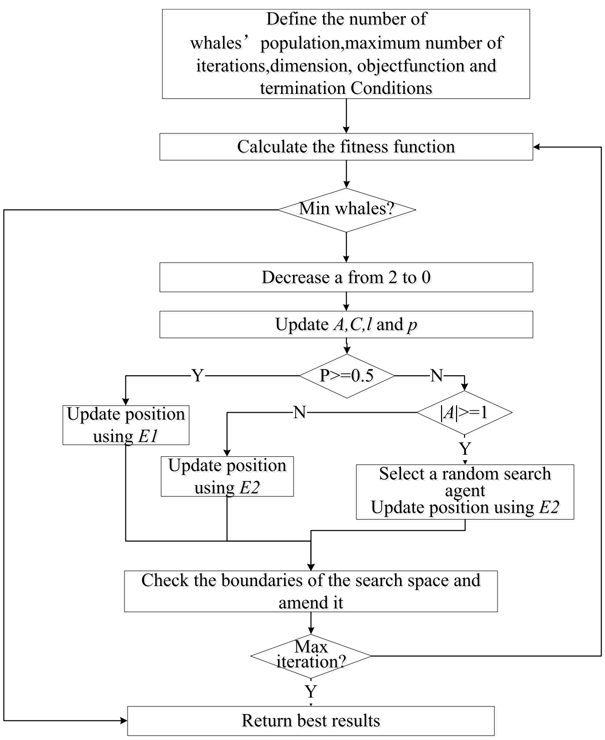


FIGURE 1. The optimization procedure of the WOA.

The remainder of this paper is organized as follows. Section 2 provides a summary of the original WOA to better understand the procedure and principles of WOA. The proposed algorithm, called WOALam, is described in detail

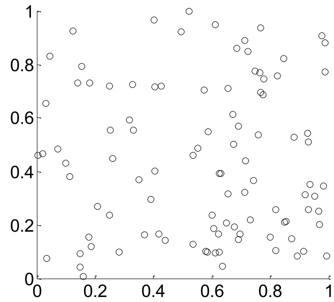


FIGURE 2. 100 points generated by random method.

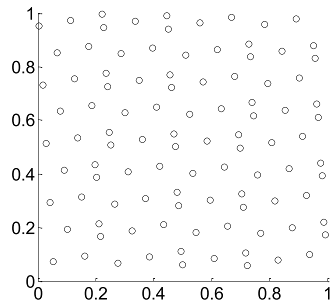


FIGURE 3. 100 points generated by good point set method.

in Section 3. Some experimental studies regarding the numerical benchmark functions, along with analysis and discussions are summarized in Section 4. Finally, Section 5 is devoted to conclusions and the future work.

II. WHALE OPTIMIZATION ALGORITHM

This algorithm is inspired by the hunting mechanism of humpback whales in nature, and simulates the shrinking encircling, spiral updating position, and random hunting mechanisms of humpback whale pods. This algorithm is composed of the following three stages: encircling prey, bubbling-net attacking, and search for prey. And it works as follows.

A. ENCIRCLING PREY

For encircling prey, the WOA algorithm assumes that the current optimal solution is the target prey. Other individual whales then try to update their positions toward the optimal position. This behavior is represented by the following equations:

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

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

where t represents the current iteration, $\vec{X}(t)$ is the position vector, $\overrightarrow{X^*}(t)$ is the position vector of the optimal solution obtained so far, || is the absolute value, and $\cdot$ is an element-by-element multiplication. $\vec{A}$ and $\vec{C}$ are coefficient vectors and are calculated as follows:

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

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

TABLE 1. Comparison algorithms.

Algorithm	Method	Authors and reference
WOA	Whale optimization algorithm	Mirjalili.et al. [11]
MWOA	Modified Whale Optimization Algorithm	Mahdavi et al.[37]
PDWOA	Whale Optimization Algorithm Based on Different Searching Paths and Perceptual Disturbance	Sun, W. et al. [40]
OWOA	Opposition-based whale optimization algorithm	Abd Elaziz, M . et al. [32]
WOASA	Hybrid Whale Optimization Algorithm with simulated annealing	Mafarja. et al. [43]
EWOA	Enhanced whale optimization algorithm	Kaveh, A. et al. [42]

Algorithm 2 The Procedure of the WOALam Algorithm

Initialize the whales population X_i (i = 1, 2, ..., n) according to ***A***.
Calculate the fitness of each search agent
X^*=the best search agent
while ($t <$ maximum number of iterations)
if1 (t mod LSI == 0)
Calculate the variance of the optimal solution of the past generations by the Eq. (13)
if2 σ^2 less than or equal to preset threshold
Calculate individual development potential according to ***B.***
Perform a partial search according to ***C.***
end if2
end if1
foreach search agent
Update a, A, C, l, and p
If3 (p<0.5)
If4 ($|A| < 1$)
Update the position of the current search by the Eq. (1)
else if4 ($|A| >= 1$)
Select random individuals by the Eq. (15)
Update the position of the current search by the Eq. (9)
end if4
else if1 (p>=0.5)
Update the position of the current search by the Eq. (5)
end if3
end for
Check if any search agent goes beyond the search space and amend it
Calculate the fitness of each search agent
Update X^* if there is a better solution
$t = t + 1$
end while
return X^*

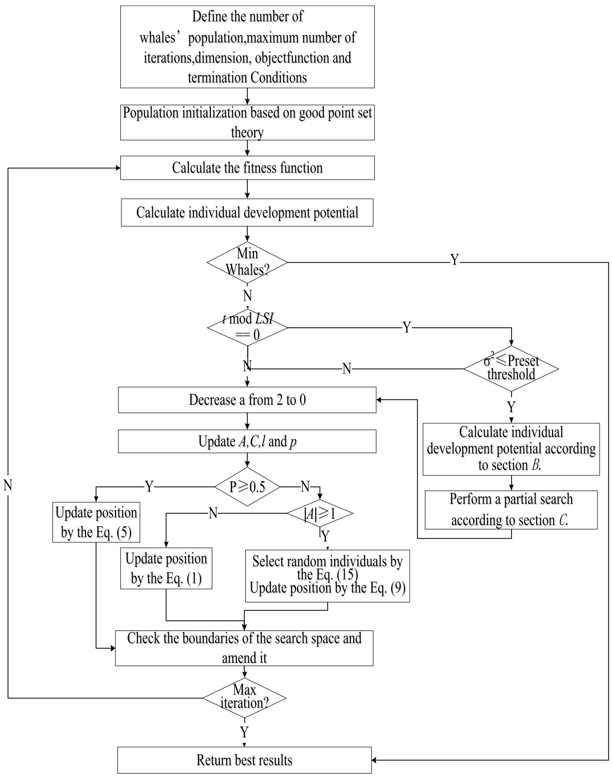


FIGURE 4. The optimization procedure of the WOALam.

where $\vec{a}$ is linearly decreased from 2 to 0 over the course of iterations (in both exploration and exploitation phases) and $\vec{r}$ is a random vector in [0, 1].

B. BUBBLE-NET ATTACKING METHOD (EXPLOITATION PHASE)

The bubble-net behavior of humpback whales includes the shrinking encircling mechanism and spiral updating position.

1) SHRINKING ENCIRCLING MECHANISM

This mechanism is mainly achieved by decreasing the value of control parameter a. $\vec{A}$ is a random value in the interval $[-a, a]$. When random values for $\vec{A}$ are in the interval $[-1, 1]$, the new position of the individual whales can be defined anywhere between the original position and the current best position. The mathematical modeling is expressed by Equations (1) and (2).

2) SPIRAL UPDATING POSITION

A spiral equation is then created between the position of whale and prey to mimic the helix-shaped movement of humpback whales as follows:

$$\overrightarrow{X'}(t+1) = \overrightarrow{D'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^*}(t) \tag{5}$$

$$\overrightarrow{D'} = \left|\overrightarrow{X^*}(t) - \vec{X}(t)\right| \tag{6}$$

TABLE 2. Test functions.

No.	Test Functions	Search space	Optimum
1	$f_1 = \sum_{i=1}^{n} x_i^2$	$[-100,100]^n$	0
2	$f_2 = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$[-10,10]^n$	0
3	$f_3 = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)$	$[-100,100]^n$	0
4	$f_4 = max(abs(x_i))$	$[-100,100]^n$	0
5	$f_5 = \sum_{i=1}^{n-1} [100 * (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30,30]^n$	0
6	$f_6 = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	$[-100,100]^n$	0
7	$f_7 = \sum_{i=1}^{n} i x_i^4 + random(0,1)$	$[-1.28,1.28]^n$	0
8	$f_8 = \sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i \rvert})]$	$[-500,500]^n$	$-418.98289 \times n$
9	$f_9 = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)] + 10n$	$[-5.12,5.12]^n$	0
10	$f_{10} = 20 + \exp(1) - 20\exp\left[-\frac{1}{5}\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right] - \exp\left[\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right]$	$[-32,32]^n$	0
11	$f_{11} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left[\frac{x_i}{\sqrt{i}}\right] + 1$	$[-600,600]^n$	0
12	$f_{12} = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 1000, 4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a \le x_i \le a \\ k(-x_i - a)^m, x_i < -a \end{cases}, y_i = 1 + \frac{x_i + 1}{4}$	$[-50,50]^n$	0
13	$f_{13} = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)[1 + \sin 2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, x_i > a \\ 0, -a \le x_i \le a \\ k(-x_i - a)^m, x_i < -a \end{cases}, y_i = 1 + \frac{x_i + 1}{4}$	$[-50,50]^n$	0
14	$f_{14} = \left[\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right]^{-1}, a_{ij} = \begin{pmatrix} -32,16,0,16,32,-32,\ldots,0,16,32 \\ -32,-32,-32,-32,-32,-16,\ldots,32,32,32 \end{pmatrix}$	$[-65.536,65.536]^n$, $n = 2$	1
15	$f_{15} = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)^2}{b_i^2 + b_i x_3 + x_4}\right]$	$[-5,5]^n, n = 4$	0.0003075
16	$f_{16} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	$[-5,5]^n, n = 2$	-1.0316285
17	$f_{17} = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	$x_1 \in [-5,10]$ $x_2 \in [0,15]$	0.398
18	$\mathrm{f}_{18} = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x^2 + 6x_1 x_2 + 3x_2^3)]$ $[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x^2 - 36x_1 x_2 + 27x_2^2)]$	$[-2,2]^n, n = 2$	3
19	$f_{19} = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{n} a_{ij}(x_j - p_{ij})^2\right]$	$[0,1]^n, n = 3$	-3.86
20	$f_{20} = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{n} a_{ij}(x_j - p_{ij})^2\right]$	$[0,1]^n, n = 6$	-3.32

TABLE 2. ***(Continued.)*** **Test functions.**

21	$f_{21}=-\sum_{i=1}^{n}[(x_i-a_i)(x_i-a_i)^T+c_i]^{-1}$	$[0,1]^n, n=5$	-10.1513
22	$f_{22}=-\sum_{i=1}^{n}[(x_i-a_i)(x_i-a_i)^T+c_i]^{-1}$	$[0,1]^n, n=7$	-10.4028
23	$f_{23}=-\sum_{i=1}^{n}[(x_i-a_i)(x_i-a_i)^T+c_i]^{-1}$	$[0,1]^n, n=10$	-10.5363
24	$f_{24}=(x_1-1)^2+\sum_{i=1}^{n} i(2x_i^2-xi-1)^2$	$[-10,10]^n$	0
25	$f_{25}=\sum_{i=1}^{n} x_i^2+(\sum_{i=1}^{n} 0.5ix_i)^2+(\sum_{i=1}^{n} 0.5ix_i)^4$	$[-5,10]^n$	0
26	$f_{26}=\sum_{i=1}^{n} ix_i^2$	$[-5.12,5.12]^n$	0
27	$f_{27}=\sum_{i=1}^{n}\sum_{j=1}^{i} x_i^2$	$[-65.536,65.536]^n$	0
28	$f_{28}=\sum_{i=1}^{n}\lvert x_i\rvert^{(i+1)}$	$[-1,1]^n$	0
29	$f_{29}=\sum_{i=1}^{n}\lvert x_i\sin(x_i+0.1x_i)\rvert$	$[0,10]^n$	0
30	$f_{30}=-\exp(-0.5\sum_{i=1}^{n} x_i^2)$	$[-1,1]^n$	-1
31	$f_{31}=1-\cos(2\pi\sqrt{\sum_{i=1}^{n}x_i^2})+0.1\sqrt{\sum_{i=1}^{n}x_i^2}$	$[-100,100]^n$	0
32	$f_{32}=-\sum_{i=1}^{n-1}\left\{e^{\left[\frac{[-(x_i^2+x_{i+1}^2+0.5x_ix_{i+1})]}{8}\right]}\cos(4\times\sqrt{x_i^2+x_{i+1}^2+0.5x_ix_{i+1}})\right\}$	$[-5,5]^n$	$-n+1$
33	$f_{33}=1-\cos(2\pi(\sqrt{\sum_{i=1}^{n}x_i^2})+0.1\sqrt{\sum_{i=1}^{n}x_i^2})$	$[-100,100]^n$	0
34	$f_{34}=\sum_{i=1}^{n}(y_i-10\cos(2\pi y_i)+10),\ y_i=\begin{cases}x_i, \lvert x_i\rvert<1/2\\ round(2x_i)/2, \lvert x_i\rvert\geq 1/2\end{cases}$	$[-5.12,5.12]^n$	0
35	$f_{35}=\sin^2(\pi\omega_1)+\sum_{i=1}^{n-1}(\omega_i-1)^2[1+10\sin^2(\pi\omega_i+1)]+(\omega_d-1)^2[1+\sin^2(2\pi\omega_d)]$ $\omega_i=1+\frac{x_i-1}{4}$	$[-10,10]^n$	0

where b is a constant for defining the shape of the logarithmic spiral, l is a random number in $[-1, 1]$, and $\cdot$ is an element-by-element multiplication.

When humpback whales attack their prey, they move simultaneously within a shrinking encircling circle and along a spiral-shaped path. The WOA assumes there is a 50% probability of choosing between the shrinking encircling mechanism and the spiral model to update the position of whales during optimization. The mathematical model is as follows:

$$\overrightarrow{X^*}(t+1)=\begin{cases}\overrightarrow{X^*}(t)-\vec{A}\cdot\vec{D} & \text{if } p<0.5\\ \overrightarrow{D'}\cdot e^{bl}\cdot\cos(2\pi l)+\overrightarrow{X^*}(t) & \text{if } p\geq 0.5\end{cases} \tag{7}$$

C. SEARCH FOR PREY (EXPLORATION PHASE)

In fact, humpback whales search randomly according to the position of each other. In the search for prey or exploration phase, the positions of other individual whales are updated according to a randomly chosen individual whale. In order to find other better prey, individual whales are forced to move far away from reference whales by setting $|\vec{A} > 1|$. WOA algorithm can perform global search by this way. The mathematical model can be expressed as:

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

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

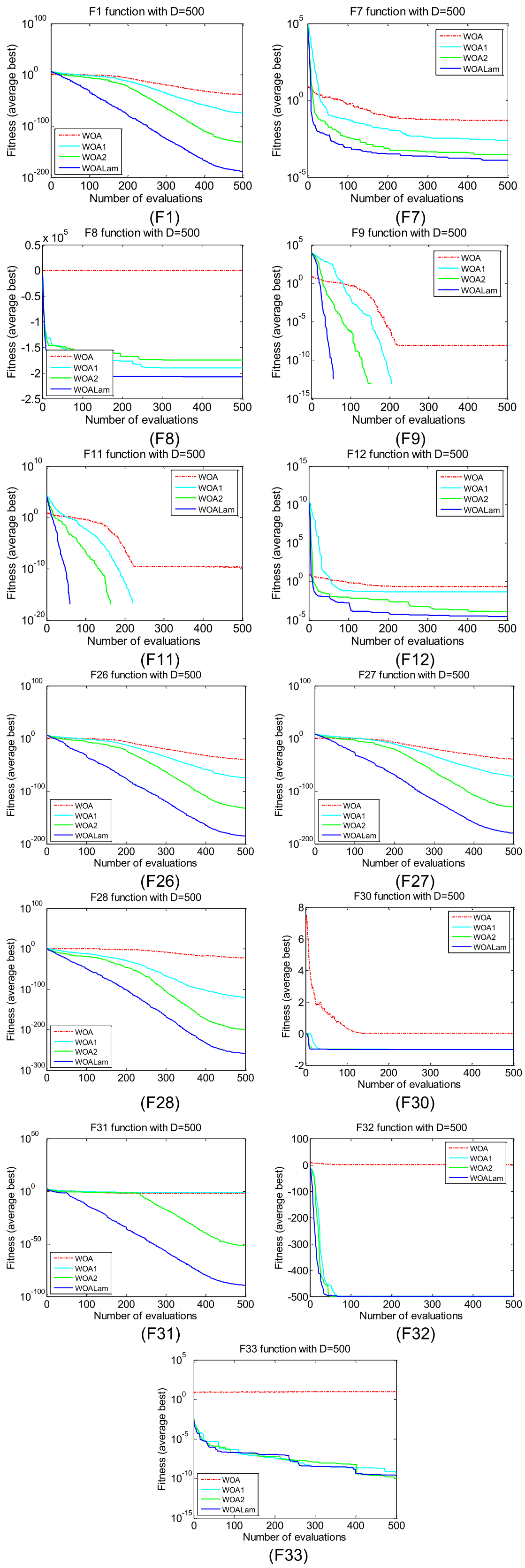


FIGURE 5. Performance comparison of three different components.

where X_{rand} is a position vector randomly selected from the current whale population. The pseudocode and flow chart of the original WOA are shown in Algorithm 1 and Figure 1, respectively.

III. WHALE OPTIMIZATION ALGORITHM BASED ON LAMARCKIAN LEARNING (WOALam)

However, the current best individual of the WOA is only the optimal position that an individual can find in the current iteration process. It can only represent the current best level of all individuals, but it can't represent the overall level and direction of evolution. Especially in solving high-dimensional optimization problems with multiple local optimal solutions. If the current best individual is a local optimal solution, other individuals updating the position according to the local optimal individual may lead the WOA to fall into the local optimal solution. Evolutionary studies have shown that organisms have great adaptability to the environment, environmental changes will cause changes in the organisms, and organisms will improve their behavior. Diversification of the environment is the root cause of biodiversification. This paper proposes an individual development potential evaluation method. Based on Lamarck's evolutionary theory, individuals with more development potentials are selected to perform local enhanced search to improve the performance of the algorithm.

A. POPULATION INITIALIZATION BASED ON GOOD POINT SET THEORY

WOA starts iterative optimization from random initialization individuals. If there are optimal solutions near some individuals, the convergence of the algorithm can be accelerated and the performance of the algorithm can be improved to a certain degree. The good point set theory [46], [47] has been proved that the weighted sum of n good points is less error than that of any other n points when the approximate computation function is integrated in the s dimensional Euclidean space unit cube. The specific definitions are as follows:

Suppose G_s is a unit cube in s dimensional Euclidean space.If $r \in G_s$, $p_n(k) = \{(\{r_1^{(n)} * k\}, \{r_2^{(n)} * k\}, \ldots, \{r_s^{(n)} * k\}), 1 \le k \le n\}$, its deviation $\varphi(n)$ satisfies $\varphi(n) = C(r, \varepsilon)n^{-1+\varepsilon}$. Among them, $C(r, \varepsilon)$ is a constant only related to r and $\varepsilon(\varepsilon > 0)$. $p_n(k)$ is called good point set. In general, $r = \{2\cos(2\pi k/p), 1 \le k \le s\}$, p is the smallest prime number that satisfies $(p - s)/2 \ge s$. Figure 2 and Figure 3 shows the initial population distribution of 100 points constructed by random point distribution method and good point set distribution method respectively.

B. INDIVIDUAL LOCAL SEARCH STRATEGY BASED ON THE LAMARCKIAN THEORY

The lamarckian evolution theory, which existed concurrently with Darwin's theory of evolution, believes that the learning

TABLE 3. Experimental results of the optimization functions with fixed dimension.

Function	Result	Algorithms						
		WOA	MWOA	PDWOA	OBWOA	WOASA	EWOA	WOALam
F14	Best	9.98E-01	9.98E-01	**9.98E-01**	9.98E-01	9.98E-01	9.98E-01	9.98E-01
	Worst	5.93E+00	2.98E+00	**9.98E-01**	2.98E+00	2.98E+00	2.98E+00	1.27E+01
	Mean	1.69E+00	2.19E+00	**9.98E-01**	1.40E+00	1.20E+00	1.49E+00	2.66E+00
	SD	1.62E+00	1.03E+00	**7.58E-08**	6.94E-01	6.27E-01	8.43E-01	3.61E+00
F15	Best	3.13E-04	**3.08E-04**	3.18E-04	3.11E-04	3.15E-04	3.16E-04	3.24E-04
	Worst	1.52E-03	**1.43E-03**	1.57E-03	2.25E-03	2.25E-03	2.11E-03	9.09E-04
	Mean	6.55E-04	**4.52E-04**	8.48E-04	5.68E-04	6.96E-04	7.00E-04	5.58E-04
	SD	4.17E-04	**3.58E-04**	4.98E-04	5.95E-04	6.56E-04	5.95E-04	2.19E-04
F16	Best	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	**-1.03E+00**
	Worst	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	**-1.03E+00**
	Mean	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	**-1.03E+00**
	SD	3.63E-09	4.28E-11	2.01E-08	8.98E-10	9.71E-10	3.15E-07	**1.89E-11**
F17	Best	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	**3.98E-01**
	Worst	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	**3.98E-01**
	Mean	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	**3.98E-01**
	SD	3.03E-05	4.55E-08	9.19E-06	2.68E-05	1.12E-05	3.07E-05	**6.81E-10**
F18	Best	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	**3.00E+00**
	Worst	3.00E+00	3.00E+01	3.02E+00	3.00E+00	3.00E+00	3.00E+00	**3.00E+00**
	Mean	3.00E+00	5.70E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	**3.00E+00**
	SD	1.30E-05	8.54E+00	7.41E-03	2.34E-05	2.10E-05	6.96E-04	**5.38E-06**
F19	Best	-3.86E+00	**-3.86E+00**	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00	-3.86E+00
	Worst	-3.86E+00	**-3.86E+00**	-3.86E+00	-3.75E+00	-3.75E+00	-3.86E+00	-3.65E+00
	Mean	-3.86E+00	**-3.86E+00**	-3.86E+00	-3.84E+00	-3.85E+00	-3.86E+00	-3.82E+00
	SD	2.10E-03	**6.59E-06**	2.61E-03	4.66E-02	3.67E-02	3.76E-05	4.00E-02
F20	Best	-3.32E+00	-3.32E+00	**-3.32E+00**	-3.32E+00	-3.32E+00	-3.32E+00	-3.20E+00
	Worst	-3.14E+00	-3.20E+00	**-3.11E+00**	-3.06E+00	-3.15E+00	-3.18E+00	-3.13E+00
	Mean	-3.27E+00	-3.27E+00	**-3.29E+00**	-3.20E+00	-3.24E+00	-3.26E+00	-3.18E+00
	SD	7.31E-02	6.16E-02	**7.41E-02**	1.03E-01	7.37E-02	6.51E-02	2.77E-02
F21	Best	-1.02E+01	-1.02E+01	-1.02E+01	-1.02E+01	-1.01E+01	-1.02E+01	**-1.02E+01**
	Worst	-2.63E+00	-1.02E+01	-5.06E+00	-1.02E+01	-2.63E+00	-2.63E+00	**-1.02E+01**
	Mean	-8.13E+00	-1.02E+01	-8.11E+00	-1.02E+01	-9.34E+00	-7.87E+00	**-1.02E+01**
	SD	3.30E+00	7.63E-05	2.63E+00	3.47E-04	2.36E+00	3.03E+00	**1.87E-05**
F22	Best	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	-1.04E+01	**-1.04E+01**
	Worst	-5.09E+00	-1.04E+01	-5.08E+00	-1.04E+01	-1.01E+01	-2.77E+00	**-1.04E+01**
	Mean	-8.24E+00	-1.04E+01	-9.34E+00	-1.04E+01	-1.03E+01	-7.24E+00	**-1.04E+01**
	SD	2.71E+00	6.11E-04	2.24E+00	6.76E-04	9.83E-02	3.40E+00	**4.12E-05**
F23	Best	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	-1.05E+01	**-1.05E+01**
	Worst	-2.42E+00	-1.05E+01	-5.06E+00	-1.05E+01	-3.84E+00	-1.68E+00	**-1.05e+01**
	Mean	-7.20E+00	-1.05E+01	-7.28E+00	-1.05E+01	-9.82E+00	-6.93E+00	**-1.05E+01**
	SD	3.61E+00	6.94E-05	2.80E+00	1.23E-03	2.10E+00	3.97E+00	**5.95E-06**

and adaptive behaviors produced by the direct influence of organisms can be inherited to the offspring to a certain extent. According to Lamarck, the traits acquired by the organisms can be directly fed back to the genotypes and passed on to the offspring through genetics. This mechanism enables individuals to directly replace the corresponding individuals in the group by the excellent individuals obtained by local search. This thinking mode has considerable reference value for optimization algorithm [48]. Liu *et al.* [49] designed a Lamarck learning rule and established a framework of Lamarck genetic algorithm, and proved its good convergence performance and local search ability in engineering theory. Luan *et al.* [50] proposed Lamarck individual learning mechanism based on the new concept of ''learning potential'', so that the advantages of local learning can be fully utilized. The results of the above literature research show that the local learning mechanism based on lamarckian evolutionism has the advantages of speeding up and strengthening

TABLE 4. Experimental results of the optimization functions with 500D.

Function	Result	Algorithms						
		WOA	MWOA	PDWOA	OBWOA	WOASA	EWOA	WOALam
F1	Best	8.81E-82	2.41E-103	5.88E-182	1.90E-97	7.94E-90	5.84E-69	**7.73E-207**
	Worst	1.50E-70	1.52E-85	3.84E-56	6.68E-81	1.21E-78	1.86E-54	**1.34E-173**
	Mean	5.02E-72	1.06E-86	1.28E-57	2.24E-82	4.51E-80	8.10E-56	**4.47E-175**
	SD	2.75E-71	3.52E-86	7.00E-57	1.22E-81	2.20E-79	3.42E-55	**0.00E+00**
F2	Best	1.73E-54	3.82E-63	3.80E-182	4.15E-56	8.60E-54	4.11E-53	**3.03E-107**
	Worst	1.25E-45	1.77E-53	3.74E-56	2.13E-48	3.57E-48	1.88E-42	**3.88E-98**
	Mean	6.91E-47	1.04E-54	1.25E-57	1.35E-49	2.63E-49	1.07E-43	**1.96E-99**
	SD	2.33E-46	3.36E-54	6.82E-57	4.48E-49	8.26E-49	4.11E-43	**7.08E-99**
F3	Best	2.62E+07	2.15E+02	2.11E+07	6.08E+06	2.80E+06	2.31E+07	**3.26E-158**
	Worst	4.83E+07	2.06E+07	4.76E+07	2.15E+07	4.34E+07	3.04E+07	**3.25E-123**
	Mean	3.44E+07	8.15E+06	3.18E+07	1.42E+07	2.11E+07	2.68E+07	**8.13E-124**
	SD	1.05E+07	8.89E+06	1.21E+07	6.48E+06	1.68E+07	3.01E+06	**1.63E-123**
F4	Best	2.06E+01	1.03E-17	1.21E-02	8.78E-19	1.32E-01	6.53E+01	**5.47E-100**
	Worst	9.89E+01	7.16E-09	2.12E+00	2.06E-09	9.89E+01	9.89E+01	**1.43E-85**
	Mean	7.99E+01	3.47E-10	4.92E-01	8.77E-11	6.99E+01	9.41E+01	**4.79E-87**
	SD	1.91E+01	1.34E-09	4.09E-01	3.77E-10	3.02E+01	6.45E+00	**2.61E-86**
F5	Best	4.95E+02	**4.06E-01**	4.94E+02	1.04E+00	3.82E-01	4.94E+02	2.20E+00
	Worst	4.97E+02	**4.94E+02**	4.95E+02	4.95E+02	4.94E+02	4.94E+02	4.94E+02
	Mean	4.96E+02	**1.01E+02**	4.95E+02	2.47E+02	2.02E+02	4.94E+02	4.77E+02
	SD	4.78E-01	**1.67E+02**	2.10E-01	2.37E+02	2.43E+02	4.34E-02	8.97E+01
F6	Best	1.10E+01	4.99E-02	2.01E-01	1.26E-02	4.59E-02	3.22E+00	**1.79E-04**
	Worst	3.27E+01	2.12E+00	3.44E+00	1.01E+01	3.78E-01	5.60E+00	**6.36E-01**
	Mean	2.16E+01	6.68E-01	6.59E-01	1.82E+00	1.61E-01	4.42E+00	**6.25E-02**
	SD	5.69E+00	5.13E-01	5.91E-01	2.05E+00	7.58E-02	6.20E-01	**1.27E-01**
F7	Best	1.53E-05	1.19E-04	1.47E-04	**1.90E-05**	4.05E-06	2.77E-04	3.88E-06
	Worst	5.78E-03	4.24E-03	4.32E-02	**1.42E-04**	9.11E-03	2.14E-02	4.74E-04
	Mean	2.25E-03	1.24E-03	7.94E-03	**5.18E-05**	2.12E-03	6.41E-03	1.58E-04
	SD	1.93E-03	1.62E-03	1.29E-02	**3.68E-05**	2.60E-03	8.04E-03	1.55E-04
F8	Best	-2.10E+05	-2.10E+05	-2.10E+05	-2.10E+05	-2.10E+05	-2.07E+05	**-2.10E+05**
	Worst	-1.47E+05	-2.05E+05	-2.10E+05	-2.10E+05	-1.98E+05	-1.32E+05	**-2.10E+05**
	Mean	-1.95E+05	-2.08E+05	-2.10E+05	-2.10E+05	-2.07E+05	-1.61E+05	**-2.10E+05**
	SD	2.27E+04	1.73E+03	4.28E+00	1.76E+00	3.84E+03	2.49E+04	**1.12E+00**
F9	Best	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	0.00E+00	0.00E+00	**0.00E+00**
	Worst	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	9.10E-13	9.10E-13	**0.00E+00**
	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	9.10E-14	1.82E-13	**0.00E+00**
	SD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.88E-13	3.84E-13	**0.00E+00**
F10	Best	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	4.44E-15	**8.88E-16**
	Worst	4.44E-15	4.44E-15	4.44E-15	7.99E-15	7.99E-15	1.51E-14	**8.88E-16**
	Mean	3.38E-15	1.95E-15	2.67E-15	3.38E-15	3.73E-15	6.93E-15	**8.88E-16**
	SD	1.72E-15	1.72E-15	1.87E-15	2.40E-15	2.25E-15	3.37E-15	**0.00E+00**
F11	Best	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	0.00E+00	**0.00E+00**
	Worst	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.11E-16	**0.00E+00**
	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.11E-17	**0.00E+00**
	SD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.51E-17	**0.00E+00**
F12	Best	2.85E-02	1.14E-04	1.97E-04	3.22E-05	1.65E-04	6.74E-03	**3.99E-08**
	Worst	8.62E-02	1.56E-03	2.63E-03	2.99E-03	6.82E-04	1.40E-02	**6.08E-05**
	Mean	5.05E-02	6.41E-04	6.83E-04	7.73E-04	4.23E-04	1.02E-02	**1.35E-05**
	SD	1.83E-02	4.59E-04	7.43E-04	1.04E-03	1.64E-04	2.52E-03	**1.88E-05**
F13	Best	6.84E+00	8.01E-02	4.40E-02	7.05E-02	2.06E-02	2.13E+00	**3.16E-04**
	Worst	2.02E+01	5.20E-01	5.05E-01	5.04E+00	2.25E-01	3.69E+00	**6.19E-01**
	Mean	1.11E+01	2.15E-01	2.30E-01	1.17E+00	1.42E-01	2.82E+00	**7.49E-02**
	SD	3.90E+00	1.49E-01	1.42E-01	1.54E+00	5.60E-02	5.54E-01	**1.91E-01**

TABLE 4. *(Continued.)* **Experimental results of the optimization functions with 500D.**

F24	Best	2.67E-01	6.73E-01	6.67E-01	2.66E-01	3.16E-01	9.77E-01	**1.83E-01**
	Worst	6.94E-01	1.00E+00	6.75E-01	9.99E-01	8.60E-01	1.00E+00	**4.98E-01**
	Mean	3.77E-01	8.55E-01	6.69E-01	7.42E-01	6.54E-01	9.96E-01	**3.12E-01**
	SD	9.31E-03	1.20E-01	2.27E-03	2.14E-01	1.33E-01	7.00E-03	**5.03E-04**
F25	Best	6.21E+03	2.18E+03	6.46E+03	6.18E+02	7.41E+03	7.63E+03	**2.52E-98**
	Worst	8.32E+03	8.83E+03	9.98E+03	1.84E+04	8.56E+03	1.19E+04	**6.68E-64**
	Mean	7.51E+03	7.21E+03	8.14E+03	8.28E+03	7.86E+03	9.00E+03	**6.79E-65**
	SD	6.96E+02	2.36E+03	1.23E+03	4.34E+03	4.52E+02	1.43E+03	**2.11E-64**
F26	Best	3.38E-83	4.21E-95	8.226e-321	8.94E-96	2.75E-88	9.09E-73	**4.01E-199**
	Worst	2.02E-72	4.57E-89	1.89E-88	7.96E-86	2.59E-79	1.64E-58	**2.76E-185**
	Mean	2.02E-73	5.08E-90	1.89E-89	1.76E-86	2.59E-80	2.20E-59	**2.98E-186**
	SD	6.39E-73	1.43E-89	5.98E-89	3.04E-86	8.19E-80	5.20E-59	**0.00E+00**
F27	Best	6.23E-80	8.55E-93	2.58E-109	3.96E-86	1.43E-89	6.73E-65	**2.68E-200**
	Worst	1.55E-75	3.47E-87	1.13E-97	1.34E-81	1.53E-75	4.87E-58	**8.05E-187**
	Mean	4.01E-76	8.92E-88	2.83E-98	6.58E-82	3.83E-76	1.22E-58	**2.02E-187**
	SD	7.66E-76	1.72E-87	5.65E-98	7.58E-82	7.64E-76	2.43E-58	**0.00E+00**
F28	Best	2.36E-144	1.44E-148	0.00E+00	1.52E-151	9.77E-146	1.85E-111	**1.87E-288**
	Worst	4.94E-122	1.13E-124	3.51E-116	5.81E-128	3.59E-110	4.51E-76	**5.85E-267**
	Mean	4.94E-123	1.13E-125	3.51E-117	5.81E-129	3.59E-111	4.51E-77	**5.87E-268**
	SD	1.56E-122	3.58E-125	1.11E-116	1.84E-128	1.14E-110	1.43E-76	**0.00E+00**
F29	Best	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Worst	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	SD	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F30	Best	-1.00E+00	**-1.00E+00**	**-1.00E+00**	**-1.00E+00**	**-1.00E+00**	-1.00E+00	**-1.00E+00**
	Worst	-1.00E+00	**-1.00E+00**	**-1.00E+00**	**-1.00E+00**	**-1.00E+00**	-1.00E+00	**-1.00E+00**
	Mean	-1.00E+00	**-1.00E+00**	**-1.00E+00**	**-1.00E+00**	**-1.00E+00**	-1.00E+00	**-1.00E+00**
	SD	3.70E-17	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.70E-17	**0.00E+00**
F31	Best	1.11E-33	1.17E-44	1.41E-80	1.49E-44	9.99E-02	7.37E-29	**3.98E-102**
	Worst	3.00E-01	2.00E-01	2.00E-01	9.99E-02	9.99E-02	3.00E-01	**1.80E-88**
	Mean	1.30E-01	9.99E-02	1.20E-01	4.99E-02	9.99E-02	1.40E-01	**1.80E-89**
	SD	9.48E-02	4.71E-02	7.88E-02	5.26E-02	6.72E-15	9.66E-02	**5.70E-89**
F32	Best	-4.99E+02	**-4.99E+02**	**-4.99E+02**	**-4.99E+02**	**-4.99E+02**	-4.99E+02	**-4.99E+02**
	Worst	-4.99E+02	**-4.99E+02**	**-4.99E+02**	**-4.99E+02**	**-4.99E+02**	-4.99E+02	**-4.99E+02**
	Mean	-4.99E+02	**-4.99E+02**	**-4.99E+02**	**-4.99E+02**	**-4.99E+02**	-4.99E+02	**-4.99E+02**
	SD	1.90E-14	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.79E-14	**0.00E+00**
F33	Best	5.64E-14	8.22E-15	**0.00E+00**	1.59E-12	3.20E-14	5.15E-13	1.70E-14
	Worst	2.88E-08	1.57E-09	**0.00E+00**	1.89E-08	1.13E-08	5.62E-09	6.59E-10
	Mean	4.66E-09	3.21E-10	**0.00E+00**	4.11E-09	2.52E-09	7.89E-10	8.00E-11
	SD	9.07E-09	4.74E-10	**0.00E+00**	7.18E-09	3.82E-09	1.73E-09	2.04E-10
F34	Best	0.00E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**0.00E+00**
	Worst	2.18E+02	2.00E+00	1.12E+02	5.75E+01	2.03E+01	1.12E+03	**0.00E+00**
	Mean	3.00E+01	1.10E+00	1.71E+01	5.75E+00	4.35E+00	2.88E+02	**0.00E+00**
	SD	6.73E+01	3.16E-01	3.43E+01	1.82E+01	8.39E+00	4.39E+02	**0.00E+00**
F35	Best	5.71E+00	1.10E-02	1.06E+00	**2.45E-03**	1.39E-01	4.64E+00	7.46E-03
	Worst	1.44E+01	5.74E-01	1.27E+01	**1.87E-01**	8.99E-01	2.42E+01	6.25E-01
	Mean	9.34E+00	2.56E-01	5.17E+00	**4.44E-02**	3.58E-01	9.87E+00	2.51E-01
	SD	2.78E+00	1.78E-01	3.95E+00	**5.62E-02**	2.48E-01	7.15E+00	2.31E-01

local search. However, Lamarck learning does not predict the validity of the search object before local search, but requires traversing all individuals. This traversal mechanism not only enhances the global search ability of the algorithm, but also increases the probability of the algorithm falling into local optimum. At the same time, some useless calculations have

TABLE 5. **Mean and standard deviation(±SD) of the optimization functions with 50D.**

Function	WOA	MWOA	PDWOA	OBWOA	WOASA	EWOA	WOALam
F01	3.52E-76(0.00E+00)	1.26E-85(6.89E-85)	7.44E-51(4.08E-50)	1.56E-85(8.13E-85)	1.62E-80(8.83E-80)	3.03E-59(1.65E-58)	**3.15E-187(0.00E+00)**
F02	2.69E-48(0.00E+00)	8.62E-56(4.34E-55)	4.86E-67(2.66E-66)	1.85E-51(8.71E-51)	4.01E-50(1.99E-49)	1.93E-46(5.72E-46)	**4.33E-95(2.37E-94)**
F03	1.71E+05(2.37E-94)	1.02E+04(1.50E+04)	1.63E+05(3.77E+04)	2.42E+04(3.25E+04)	1.51E+05(5.07E+04)	2.03E+05(4.98E+04)	**2.56E-157(1.39E-156)**
F04	4.98E+01(1.39E-156)	2.14E-11(7.41E-11)	4.20E-01(2.96E-01)	2.02E-11(5.81E-11)	3.18E+01(3.26E+01)	8.49E+01(7.56E+00)	**7.48E-89(2.59E-88)**
F05	4.80E+01(2.59E-88)	1.03E+01(1.91E+01)	4.69E+01(8.68E+00)	**1.62E+01(2.22E+01)**	2.26E+01(2.42E+01)	4.85E+01(1.61E-02)	4.80E+01(2.18E-01)
F06	5.99E-01(2.18E-01)	3.50E-03(8.53E-04)	3.69E-02(2.60E-02)	8.26E-02(1.06E-01)	4.17E-03(2.54E-03)	1.75E-01(4.15E-02)	**1.25E-03(2.08E-04)**
F07	2.48E-03(2.08E-02)	1.65E-03(2.81E-03)	5.75E-03(8.42E-03)	**1.32E-04(2.48E-04)**	2.11E-03(1.87E-03)	1.01E-02(7.40E-03)	1.87E-04(2.96E-04)
F08	-1.82E+04(2.96E-04)	-2.09E+04(4.51E+01)	-2.09E+04(1.15E+02)	-2.10E+04(8.49E-01)	-2.07E+04(4.67E+02)	-1.55E+04(2.80E+03)	-2.07E+04(8.51E+02)
F09	0.00E+00(8.51E+02)	**0.00E+00(0.00E+00)**	7.77E-01(4.26E+00)	**0.00E+00(0.00E+00)**	1.90E-15(1.04E-14)	1.27E+01(6.38E+01)	**0.00E+00(0.00E+00)**
F10	4.68E-15(0.00E+00)	2.78E-15(2.03E-15)	3.14E-15(2.38E-15)	3.26E-15(2.16E-15)	3.73E-15(2.54E-15)	5.74E-15(3.79E-15)	**8.88E-16(0.00E+00)**
F11	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	3.70E-18(2.03E-17)	**0.00E+00(0.00E+00)**	1.94E-03(1.06E-02)	3.77E-02(1.16E-01)	**0.00E+00(0.00E+00)**
F12	1.49E-02(0.00E+00)	6.13E-05(2.57E-05)	1.73E-03(2.64E-03)	8.05E-04(1.46E-03)	3.03E-04(1.39E-04)	1.68E-02(4.62E-02)	**1.80E-06(2.86E-04)**
F13	7.47E-01(2.86E-04)	5.28E-03(1.20E-02)	3.74E-02(4.41E-02)	1.16E-01(1.19E-01)	1.41E-02(1.20E-02)	2.75E-01(1.07E-01)	**3.23E-03(4.27E-03)**
F24	6.67E-01(1.64E+00)	4.79E-01(2.09E-01)	6.67E-01(5.83E-04)	5.08E-01(2.19E-01)	**4.82E-01(2.18E-01)**	7.02E-01(2.10E-02)	6.67E-01(6.53E-04)
F25	8.88E+02(6.53E-04)	8.09E+02(1.63E+02)	8.64E+02(1.36E+02)	4.87E+02(2.34E+02)	9.04E+02(2.52E+02)	8.90E+02(1.60E+02)	**1.09E-123(3.55E-123)**
F26	1.87E-79(3.55E-123)	1.34E-89(7.07E-89)	1.86E-85(5.69E-85)	5.66E-87(1.89E-86)	4.16E-82(1.60E-81)	1.59E-60(6.94E-60)	**6.72E-181(0.00E+00)**
F27	1.06E-75(0.00E+00)	5.44E-88(1.87E-87)	4.10E-59(1.90E-58)	1.46E-83(6.05E-83)	1.51E-82(4.56E-82)	5.11E-56(2.80E-55)	**1.21E-185(0.00E+00)**
F28	4.86E-118(0.00E+00)	9.01E-118(4.94E-117)	3.16E-135(1.73E-134)	4.93E-123(2.69E-122)	2.57E-106(8.57E-106)	6.33E-82(3.01E-81)	**1.48E-262(0.00E+00)**
F29	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
F30	**-1.00E+00(0.00E+00)**	**-1.00E+00(0.00E+00)**	-1.00E+00(3.57E-17)	**-1.00E+00(0.00E+00)**	-1.00E+00(5.46E-17)	-1.00E+00(5.05E-17)	**-1.00E+00(0.00E+00)**
F31	1.10E-01(0.00E+00)	8.32E-02(4.61E-02)	1.20E-01(7.61E-02)	6.99E-02(4.66E-02)	1.17E-01(6.47E-02)	1.43E-01(8.58E-02)	**3.85E-93(1.97E-92)**
F32	-4.90E+01(1.97E-92)	**-4.90E+01(0.00E+00)**	-4.43E+01(8.07E+00)	**-4.90E+01(0.00E+00)**	-4.71E+01(7.24E+00)	-4.49E+01(1.08E+01)	**-4.90E+01(0.00E+00)**
F33	1.16E-09(0.00E+00)	7.33E-12(1.63E-11)	**0.00E+00(0.00E+00)**	1.04E-09(3.07E-09)	8.46E-10(2.07E-09)	5.08E-10(6.53E-10)	1.01E-11(3.55E-11)
F34	3.00E+00(3.55E-11)	1.10E+00(2.31E+00)	1.60E+00(2.03E+00)	8.75E-01(3.69E+00)	3.97E+00(9.07E+00)	3.64E+01(8.37E+01)	**0.00E+00(0.00E+00)**
F35	6.82E-01(0.00E+00)	2.36E-03(7.45E-04)	5.04E-01(3.69E-01)	2.77E-02(5.05E-02)	5.02E-03(3.19E-03)	2.17E+01(4.66E+01)	**6.01E-05(7.93E-04)**

been increased. Therefore, development potential of individual is needed to evaluate before performing local search in the algorithm. This paper proposes the concept of ''development potential''. The size of development potential directly reflects the improvement effect of local search. Before performing the local search, the probability that each individual can search the global optimum through learning can be predicted by evaluating the improvement ability. Local search is carried out for individuals with high improvement ability, which reduces the computational complexity required by the algorithm. The evaluation method of development potential is given as follows.

In the absence of any prior knowledge, development potential can be made more quickly by coordinating the use of acquired knowledge and the exploration of new knowledge. Upper Confidence Bound (UCB) [51], [52] is composed of two parts, one of which is the current reward value, and the other is related to the size of the unilateral confidence interval to ensure that the expected reward can fall within the range of reward with a great possibility. This algorithm has been applied to the computer Go program and has achieved good results. Based on the immediate value of the current situation on the chessboard and the future value of the optional sub-points, the UCB algorithm is used to calculate the maximum confidence interval of the upper limit as the falling sub-point of the current situation. Therefore, the calculation of individual development potential can be analogous to computer Go decision making. The mathematical model for calculating individual development potential can be expressed as:

$$Potential_t(x_i) = \frac{\left|f(x_i) - f(x_i^p)\right|}{t - Succ(x_i)} + \sqrt{\frac{C_0 \times \log(t)}{Succ(x_i)}} \tag{10}$$

where, t is the number of current iterations. The role of C_0 is to balance the utilization of individual knowledge and the need for exploration, which is set to 2 according to the default value of UCB. $Potential_t(x_i)$ is the individual development potential of x_i in the t iteration. x_i^p is the best location that x_i has searched. $Succ(x_i)$ is the number of iterations that x_i have been improved.

C. LOCAL SEARCH STRATEGY BASED ON INDIVIDUALS WITH BETTER DEVELOPMENT POTENTIAL

Suppose that the WOA algorithm has been iterated several times, individuals in the whale population have reliable experience and knowledge at certain locations in the solution space. For example, $X_{\min} = [0, 0, 0, 0, 0]$ is the optimal solution for the minimized objective function $f(X) = \sum_{i=1}^{n} x_i^2$, $n = 5$. Also suppose that X_1 and X_2 be two solution vectors

TABLE 6. Mean and standard deviation of the optimization functions with 100D.

Function	WOA	MWOA	PDWOA	OBWOA	WOASA	EWOA	WOALam
F01	1.48E-75(7.93E-03)	2.49E-89(7.85E-89)	1.20E-55(6.21E-55)	3.35E-82(1.76E-81)	7.27E-80(3.78E-79)	9.96E-61(3.68E-60)	**1.01E-177(0.00E+00)**
F02	5.94E-47(3.01E-46)	2.19E-54(7.03E-54)	1.49E-58(8.09E-58)	6.97E-52(1.88E-51)	1.44E-49(5.64E-49)	3.51E-46(1.02E-45)	**2.02E-98(9.31E-98)**
F03	9.88E+05(1.88E+05)	2.13E+05(3.00E+05)	7.13E+05(1.69E+05)	1.94E+05(1.98E+05)	8.02E+05(2.79E+05)	1.06E+06(2.60E+05)	**1.09E-147(5.97E-147)**
F04	8.12E+01(1.56E+01)	2.92E-11(8.55E-11)	5.55E-01(3.38E-01)	1.40E-09(7.56E-09)	4.61E+01(3.79E+01)	9.14E+01(4.91E+00)	**2.43E-88(1.30E-87)**
F05	9.80E+01(2.56E-01)	**2.85E+01(4.27E+01)**	8.55E+01(3.25E+01)	3.66E+01(4.47E+01)	3.32E+01(4.65E+01)	9.80E+01(1.13E-02)	9.76E+01(1.19E-01)
F06	2.40E+00(7.56E-01)	2.21E-02(1.16E-02)	9.59E-02(7.88E-02)	2.40E-01(2.35E-01)	1.58E-02(7.50E-03)	5.97E-01(1.14E-01)	**8.93E-03(2.14E-02)**
F07	1.79E-03(1.50E-03)	1.64E-03(1.42E-03)	4.36E-03(5.27E-03)	**1.27E-04(2.58E-04)**	2.54E-03(2.25E-03)	1.09E-02(7.47E-03)	1.64E-04(2.28E-04)
F08	-3.59E+04(5.49E+03)	-4.16E+04(7.18E+02)	-3.59E+04(5.49E+03)	-4.19E+04(5.66E+00)	-4.16E+04(6.66E+02)	-3.08E+04(5.15E+03)	**-4.18E+04(2.69E+00)**
F09	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	1.52E-14(4.94E-14)	**0.00E+00(0.00E+00)**
F10	4.44E-15(2.47E-15)	3.14E-15(1.98E-15)	3.73E-15(2.70E-15)	3.38E-15(2.12E-15)	4.32E-15(2.38E-15)	5.03E-15(2.48E-15)	**8.88E-16(0.00E+00)**
F11	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	3.70E-18(2.03E-17)	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	1.80E-02(9.86E-02)	**0.00E+00(0.00E+00)**
F12	2.59E-02(9.20E-03)	1.80E-04(1.24E-04)	1.13E-03(1.45E-03)	1.33E-03(4.39E-03)	2.65E-04(1.21E-04)	8.93E-03(2.60E-03)	**9.31E-05(1.88E-04)**
F13	1.82E+00(7.64E-01)	**9.32E-03(8.05E-03)**	5.39E-02(6.00E-02)	2.00E-01(2.53E-01)	2.05E-02(1.37E-02)	6.64E-01(2.12E-01)	1.44E-02(2.95E-02)
F24	6.67E-01(1.04E-03)	6.29E-01(1.46E-01)	6.68E-01(8.36E-04)	6.06E-01(1.94E-01)	6.13E-01(1.40E-01)	8.38E-01(7.26E-02)	**6.07E-01(5.94E-06)**
F25	1.66E+03(2.36E+02)	1.59E+03(3.68E+02)	1.72E+03(2.91E+02)	1.59E+03(7.37E+02)	1.74E+03(2.22E+02)	1.79E+03(2.21E+02)	**9.58E-100(4.10E-99)**
F26	3.57E-77(9.53E-77)	2.29E-88(1.21E-87)	1.26E-78(6.92E-78)	8.24E-86(4.27E-85)	4.56E-80(2.46E-79)	6.18E-59(2.58E-58)	**1.87E-186(0.00E+00)**
F27	4.47E-76(1.70E-75)	2.33E-86(7.78E-86)	1.98E-63(1.08E-62)	3.23E-82(1.75E-81)	3.62E-80(1.33E-79)	1.80E-54(9.21E-54)	**4.19E-178(0.00E+00)**
F28	1.74E-116(9.49E-116)	2.23E-124(1.05E-123)	6.72E-123(3.68E-122)	1.27E-125(6.93E-125)	6.33E-106(3.47E-105)	1.73E-80(9.38E-80)	**1.55E-256(0.00E+00)**
F29	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**	**0.00E+00(0.00E+00)**
F30	-1.00E+00(3.57E-17)	**-1.00E+00(0.00E+00)**	-1.00E+00(2.06E-17)	**-1.00E+00(0.00E+00)**	-1.00E+00(7.14E-17)	-1.00E+00(5.83E-17)	**-1.00E+00(0.00E+00)**
F31	1.10E-01(5.48E-02)	7.99E-02(4.06E-02)	1.23E-01(8.97E-02)	5.99E-02(4.98E-02)	1.13E-01(6.81E-02)	1.43E-01(9.71E-02)	**1.14E-91(4.57E-91)**
F32	-9.90E+01(6.46E-15)	**-9.90E+01(0.00E+00)**	-9.43E+01(1.44E+01)	**-9.90E+01(0.00E+00)**	-9.90E+01(3.73E-15)	-9.16E+01(2.27E+01)	**-9.90E+01(0.00E+00)**
F33	4.42E-10(6.77E-10)	9.12E-11(2.51E-10)	**0.00E+00(0.00E+00)**	1.93E-09(5.89E-09)	1.12E-09(2.64E-09)	6.95E-10(1.37E-09)	5.16E-11(1.71E-10)
F34	4.68E+00(1.34E+01)	2.64E+00(7.06E+00)	4.93E+00(1.53E+01)	7.00E-01(1.93E+00)	5.48E+00(1.38E+01)	1.58E+01(2.45E+01)	**0.00E+00(0.00E+00)**
F35	1.69E+00(7.85E-01)	3.48E-02(8.63E-02)	1.28E+00(6.57E-01)	6.22E-02(7.57E-02)	2.27E-02(1.51E-02)	6.41E+00(1.63E+01)	**1.54E-02(1.69E-02)**

of WOA with corresponding fitness values as follows:

$$X_1 = [0.0, 0.0, 0.0, 5.0, 0.0], \quad f(X_1) = 25$$
$$X_2 = [2.0, 2.0, 2.0, 1.0, 2.0], \quad f(X_2) = 17 \tag{11}$$

It can be seen from equation (11) that the fitness of X_2 is better than that of X_1, but from the composition of the solution, X_1 is closer to the optimal solution $X_{\min}$ only by changing the value of the fourth position. X_1 can be considered as an individual with greater development potential. X_1 which is modified by the value of the fourth position can quickly get a better fitness value then X_2. Although the above example exists in the actual iteration process, all WOAs guide individual evolution based on the optimal fitness value except for a few, desired decision variables might be modified. So, local search strategy is proposed to use the value of the optimal individual to modify the value of each dimension of other individuals. Through learning, individual can better adapt to the environment and improve their fitness. In local search process, inappropriate decision variables of individuals with the greater development potential have the opportunity to be modified by using the most appropriate decision variables of optimal individual . In other words, every decision variable x_i of X is replaced by a decision variable x_j of X_{best}. X_{best} is the current optimal individual. As it can be deduced from local search strategy, the computational complexity of search strategy is $O(n^2)$. Based on this, the algorithm complexity of WOALam can be deduced as follows:

$$Complexity\,(WOALam) \propto O\left(MaxT + MaxT \times m \times n^2\right) \tag{12}$$

where m is the number of whale individuals that were searched locally, $MaxT$ is the total number of iterations of the algorithm, n is the number of decision variables. It is obvious that such complexity might be undesired for high dimensional optimization problems. So for local search, how to solve the contradiction between the sufficiency of local search and the calculation time is a primary issue that we need to consider. In order to overcome this problem, three methods will be integrated into the proposed algorithm:

1) SELECTION OF ITERATIONS OF THE LOCAL SEARCH START

If the fitness of each iteration increases slowly, it may fall into the local optimum or have found the global optimum solution. It is necessary to judge whether local search strategy is adopted in this case. The variance of fitness can reflect the degree of convergence of individuals. The smaller the variance, the more likely it is to fall into the local optimum or to find the global optimal solution. Therefore, the algorithm is judged by the variance of individual fitness value, and the

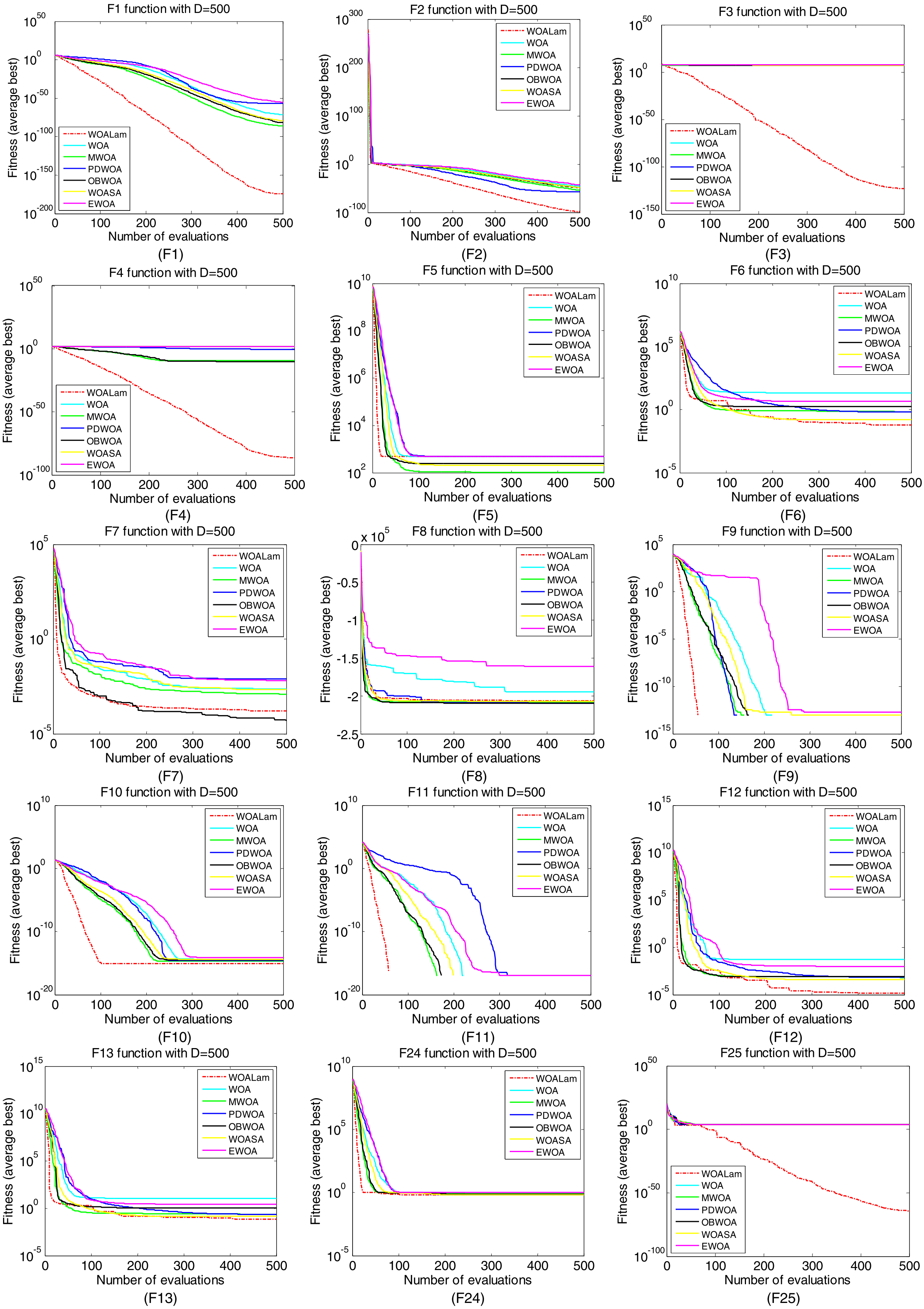


FIGURE 6. Average fitness curves for some selected functions with 500D.

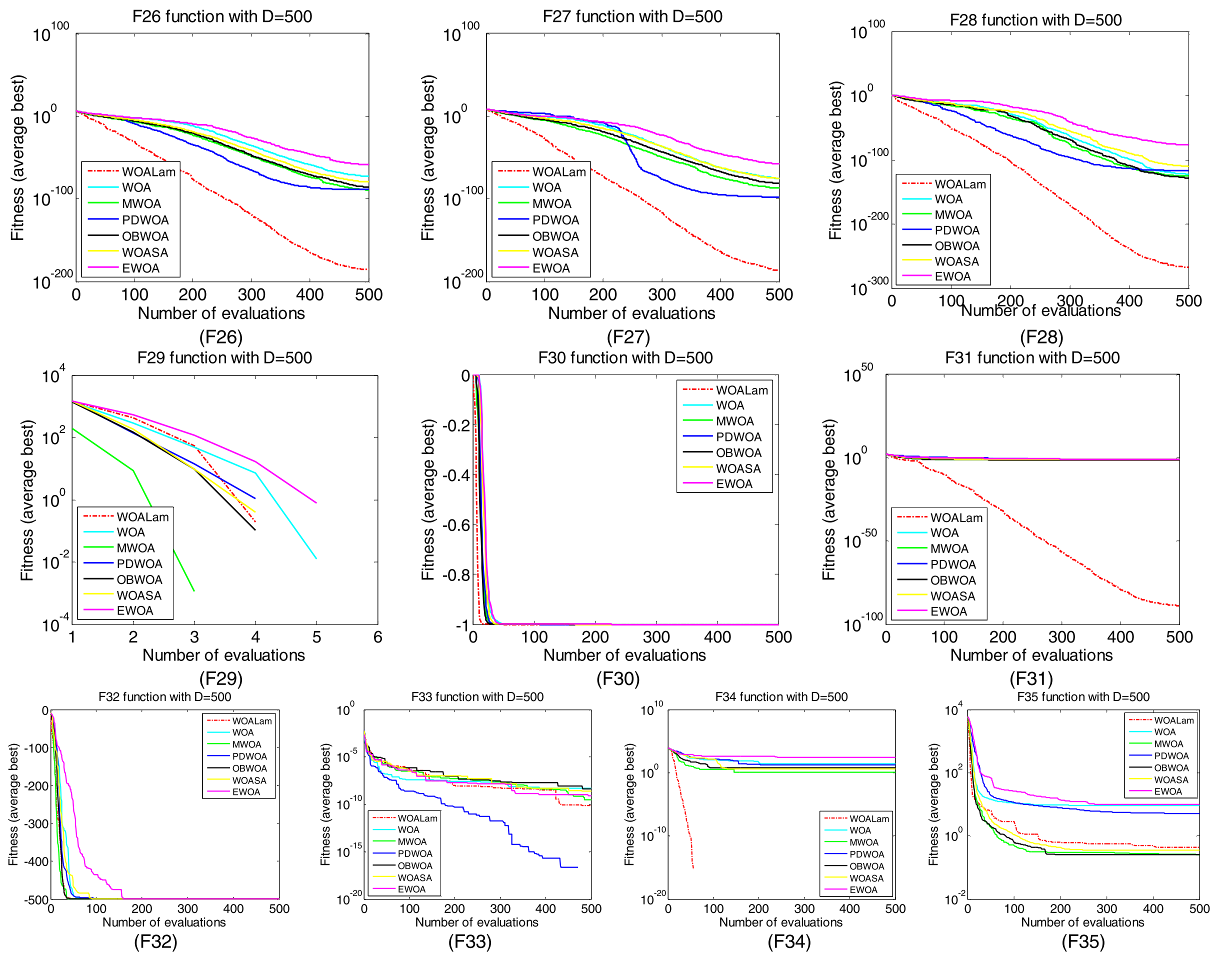


FIGURE 6. ***(Continued.)*** **Average fitness curves for some selected functions with 500D.**

calculation method is presented by Equation (13).

$$\sigma^2 = \frac{\sum_{i=t-count}^{t} (f_i - \bar{f})^2}{count} \tag{13}$$

where, t is represents the current number of iterations, $count$ is the number of fitness values ($count$ is set to 20 in this paper), f_i is the fitness value of each iteration, and $\bar{f}$ is the average of fitness values.

According to Lamarck's evolutionary theory, improved parent individuals can inherit the learning good genes to their offspring and obtain more adaptive individuals. Therefore, individuals need to undergo certain iterations before passing on the acquired good genes to their offspring. In order to gain good genes and avoid frequent calculation variance, Local Search Start Interval (*LSSI*) is set in this paper (The default value of *LSSI* is set to 50). Where *LSSI* is the parameter that controls the period in which local search and variance calculation occur. In other words, when t mod *LSSI* is equal to 0 and the variance satisfies a preset threshold, local search strategy is performed.

2) SELECTION OF INDIVIDUALS WITH BETTER DEVELOPMENT POTENTIAL

From Equation (12), m is the number of individuals who need to perform local search strategy. The larger the value of m, the greater the amount of work that needs to be calculated . Each individual has a certain development potential ($Potential(x_i)$), and the individual with the development potential greater than the average development potential of the population is selected for local search. In the early stage of evolution, the value of m should be more to maintain the diversity of the population, and in the later stage of evolution, the value of m should be less to accelerate the convergence of the algorithm. Therefore, the calculation method of m is presented by Equation (14).

$$m = Number - \frac{Number - 1}{MaxT} \times t \tag{14}$$

where, t is the number of current iterations. $MaxT$ is the maximum number of iterations. $Number$ is the number of individuals greater than the average development potential of the population.

TABLE 7. The results of wilcoxon's rank sum test for functions with 100D.

Function		WOALam Vs					
		WOA	MWOA	PDWOA	OBWOA	WOASA	EWOA
F1	*p*-value	6.80E-08	6.80E-08	1.20E-06	6.80E-08	6.80E-08	6.80E-08
	h-value	1	1	1	1	1	1
F2	*p*-value	6.80E-08	6.80E-08	5.25E-01	6.80E-08	6.80E-08	6.80E-08
	h-value	1	1	0	1	1	1
F3	*p*-value	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08
	h-value	1	1	1	1	1	1
F4	*p*-value	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08
	h-value	1	1	1	1	1	1
F5	*p*-value	5.09E-04	1.80E-06	1.20E-01	3.15E-02	1.23E-02	6.80E-01
	h-value	1	-1	0	1	1	0
F6	*p*-value	6.80E-08	4.68E-05	1.58E-06	4.54E-06	2.56E-03	6.80E-08
	h-value	1	1	1	1	1	1
F7	*p*-value	3.50E-06	8.60E-06	2.96E-07	4.99E-01	4.17E-05	1.24E-07
	h-value	1	1	1	0	1	1
F8	*p*-value	3.06E-03	6.95E-01	1.26E-01	1.78E-03	5.61E-01	1.81E-05
	h-value	1	0	0	1	0	1
F9	*p*-value	NaN	NaN	NaN	NaN	NaN	1.62E-01
	h-value	0	0	0	0	0	0
F10	*p*-value	8.03E-06	5.68E-06	3.66E-04	5.74E-05	8.03E-06	1.68E-08
	h-value	1	1	1	1	1	1
F11	*p*-value	NaN	NaN	1.27E-04	NaN	NaN	1.63E-05
	h-value	0	0	1	0	0	1
F12	*p*-value	6.80E-08	6.22E-04	7.95E-07	5.90E-05	1.10E-05	6.80E-08
	h-value	1	1	1	1	1	1
F13	*p*-value	6.80E-08	1.44E-02	1.79E-04	4.54E-06	3.97E-03	6.80E-08
	h-value	1	-1	1	1	1	1
F24	*p*-value	5.23E-07	3.61E-02	6.02E-07	2.85E-01	9.62E-02	6.80E-08
	h-value	1	1	1	0	0	1
F25	*p*-value	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08
	h-value	1	1	1	1	1	1
F26	*p*-value	6.80E-08	6.80E-08	2.23E-02	6.80E-08	6.80E-08	6.80E-08
	h-value	1	1	1	1	1	1
F27	*p*-value	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08	6.80E-08
	h-value	1	1	1	1	1	1
F28	*p*-value	6.80E-08	6.80E-08	8.36E-04	6.80E-08	6.80E-08	6.80E-08
	h-value	1	1	1	1	1	1
F29	*p*-value	NaN	NaN	NaN	NaN	NaN	NaN
	h-value	0	0	0	0	0	0
F30	*p*-value	3.42E-01	NaN	1.62E-01	NaN	1.62E-01	1.95E-02
	h-value	0	0	0	0	0	1
F31	*p*-value	6.80E-08	6.78E-08	5.95E-08	6.80E-08	6.80E-08	6.80E-08
	h-value	1	1	1	1	1	1
F32	*p*-value	1.67 E-01	NaN	1.63E-03	NaN	1.62E-01	3.42E-03
	h-value	0	0	1	0	0	1
F33	*p*-value	4.32E-03	4.57E-01	1.22E-08	2.75E-04	2.47E-04	1.44E-02

TABLE 7. *(Continued.)* The results of wilcoxon's rank sum test for functions with 100D.

	h-value	1	0	-1	1	1	1
F34	p-value	8.02E-06	1.38E-04	5.65E-08	1.63E-05	2.45E-05	3.30E-06
	h-value	1	1	1	1	1	1
F35	p-value	6.80E-08	3.61E-02	6.80E-08	1.78E-03	1.06E-01	6.80E-08
	h-value	1	1	1	1	0	1

3) SUBSET SELECTION OF OPTIMAL INDIVIDUALS FOR LOCAL SEARCH

Another way to decrease the computational complexity imposed by local search is to use a subset of decision variables of X_{best}. Similar to the above two methods, this method allows the algorithm to avoid too much complexity without losing the benefits of local search. The number of selected decision variables is determined by a parameter called Subset Length (SL), which is usually considered much smaller than n. Suppose that SL is used to indicate the length of the selected subset (SL is set to $n/3$ in this paper), n is the dimension of the problem to be optimized. The process of subset selection consists of two steps:

Step 1: Randomly generate an integer between 1 and (n-SL) as the starting point of the subset;

Step 2: From the starting point, successive SL decision variables of the optimal individual are selected as a subset of the local search.

Integrated into these three improved methods, the number of necessary fitness evaluations consumed by WOALam changes from $O\left(MaxT + MaxT \times m \times n^2\right)$ into $O(MaxT + \left\lfloor \frac{MaxT}{LSI} \right\rfloor \times m \times n \times SL)$.

D. IMPROVEMENT OF INDIVIDUAL RANDOM SELECTION METHOD IN ORIGINAL WOA

In the search for prey (exploration) stage, the reference individuals in the original whale algorithm are randomly selected. Although this method is conducive to increasing the diversity of the population, it also has blindness. It is possible to use the solution obtained from the local search as the reference individual (similar to the current optimal solution), which is not conducive to exploration. In addition, if the current fitness value is not much different from fitness of the reference individual, it means that the individual itself has no better improvement. This paper proposes the following methods to select the reference individuals in the exploration phase. The mathematical formula for individual selection is as follows:

$$x_i| \max(\frac{1}{f_t(x_i) - f_b(x_i)}), \quad i = 1, 2, \ldots, n \tag{15}$$

where, x_i is the selected reference individual. $f_t(x_i)$ is the fitness value of the t-th iteration of the x_i. $f_b(x_i)$ is the best fitness value currently obtained by x_i. The pseudocode and flow chart of the WOALam algorithm are shown in Algorithm 2 and Figure 4, respectively.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. EXPERIMENTAL SETTINGS

To extensively investigate the performance of WOALam algorithm, we compared it with six other variants of WOA. In the field of evolutionary computation, it is common to compare different algorithms using a large test set, especially when the test involves function optimization. In order to verify whether an algorithm is better than another, a function set containing 44 benchmark problems with different characteristics is employed. Although it is not exhaustive, it includes many different types of problems such as unimodal, multimodal, separable, non-separable and multidimensional.

In this section, the benchmark functions are introduced. The experimental settings and results are presented, and several conclusions are provided. All experiments are implemented on Intel(R) Core(TM) i5 7200U CPU dual core processor 2.50 GHz and 2.71 GHz with 4.00GB memory physical address extension. All algorithms are coded and carried out by MATLAB R2014a version under the environment of Microsoft Windows 10 Professional. Six improved whale algorithms proposed in recent years were tested for comparison. The source information of the comparison algorithms are listed in Table 1.

B. PERFORMANCE COMPARISON OF THREE DIFFERENT COMPONENTS OF THE WOALam

The algorithms were tested on 35 well-known benchmark functions based on previous work (Wang *et al.* [53], Eesa *et al.* [54], Mirjalili *et al.* [55]). The test functions are listed in Table 2. For all experiments, Maximum number of iterations of the algorithm ($MaxT$) is set to 500, which is the same as that of the other WOA variants used for comparisons. The population size N of all algorithms is set to 30. Moreover, in our experiments, 30 independent runs are carried out in each case. To compare the performance of the proposed algorithm, all the experiments in this paper run according to the above settings unless a change is mentioned. In experimental study, we compared WOA with WOA1 (WOA is improved only by initializing the population through Good Point Sets), WOA2 (WOA is improved only by initializing the population through Good Point Sets and embedding Lamarckian local search strategy) and WOALam. 25 high-dimensional optimization functions (F1-F13, F24-F35) were selected and compared. All the functions are tested in 500 dimensions.

TABLE 8. The results of wilcoxon's rank sum test for functions with 500D.

Function		WOALam Vs					
		WOA	MWOA	PDWOA	OBWOA	WOASA	EWOA
F1	*p*-value	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
	h-value	1	1	1	1	1	1
F2	*p*-value	1.83E-04	1.83E-04	4.73E-01	1.83E-04	1.83E-04	1.83E-04
	h-value	1	1	0	1	1	1
F3	*p*-value	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
	h-value	1	1	1	1	1	1
F4	*p*-value	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
	h-value	1	1	1	1	1	1
F5	*p*-value	1.83E-04	2.11E-02	1.83E-04	3.45E-01	7.91E-01	1.83E-04
	h-value	1	-1	1	0	0	1
F6	*p*-value	1.83E-04	1.40E-02	2.46E-04	3.30E-04	9.11E-03	1.83E-04
	h-value	1	1	1	1	1	1
F7	*p*-value	2.11E-02	2.83E-03	1.71E-03	9.11E-03	2.58E-02	1.83E-04
	h-value	1	1	1	-1	1	1
F8	*p*-value	2.83E-03	1.04E-01	5.71E-01	6.23E-01	6.40E-02	4.40E-04
	h-value	1	0	0	0	0	1
F9	*p*-value	NaN	NaN	NaN	NaN	1.52 E-03	3.68E-03
	h-value	0	0	0	0	1	1
F10	*p*-value	1.98E-03	1.68E-01	1.68E-01	6.04E-04	1.37E-02	1.57E-04
	h-value	1	0	0	1	1	1
F11	*p*-value	NaN	NaN	NaN	NaN	NaN	3.68E-01
	h-value	0	0	0	0	0	0
F12	*p*-value	1.83E-04	1.32E-03	4.40E-04	7.69E-04	7.69E-04	1.83E-04
	h-value	1	1	1	1	1	1
F13	*p*-value	1.83E-04	3.61E-03	2.20E-03	2.46E-04	1.01E-03	1.83E-04
	h-value	1	1	1	1	1	1
F24	*p*-value	1.83E-04	1.83E-04	1.83E-04	2.83E-03	1.83E-04	1.83E-04
	h-value	1	1	1	1	1	1
F25	*p*-value	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
	h-value	1	1	1	1	1	1
F26	*p*-value	1.83E-04	1.83E-04	5.21E-01	1.83E-04	1.83E-04	1.83E-04
	h-value	1	1	0	1	1	1
F27	*p*-value	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04	1.83E-04
	h-value	1	1	1	1	1	1
F28	*p*-value	1.83E-04	1.83E-04	2.58E-02	1.83E-04	1.83E-04	1.83E-04
	h-value	1	1	1	1	1	1
F29	*p*-value	NaN	NaN	NaN	NaN	NaN	NaN
	h-value	0	0	0	0	0	0
F30	*p*-value	7.67E-02	NaN	NaN	NaN	1.37E-02	NaN
	h-value	0	0	0	0	1	0
F31	*p*-value	1.83E-04	1.82E-04	1.72E-04	1.83E-04	1.83E-04	1.83E-04
	h-value	1	1	1	1	1	1
F32	*p*-value	3.68E-01	NaN	NaN	NaN	NaN	3.68E-01
	h-value	0	0	0	0	0	0
F33	*p*-value	1.40E-02	5.21E-01	6.39E-05	2.11E-02	3.76E-02	2.11E-02

TABLE 8. ***(Continued.)*** **The results of wilcoxon's rank sum test for functions with 500D.**

	h-value	1	0	-1	1	1	1
F34	*p*-value	2.09E-03	9.66E-05	2.43E-05	7.76E-03	7.08E-04	2.21E-03
	h-value	1	1	1	1	1	1
F35	*p*-value	1.83E-04	9.11E-03	1.83E-04	1.73E-02	3.76E-02	1.83E-04
	h-value	1	1	1	0	1	1

TABLE 9. **Mean and standard deviation of the optimization functions with 10D.**

CEC 2017 Function	Result	Algorithms						
		WOA	MWOA	PDWOA	OBWOA	WOASA	EWOA	WOALam
F1	Best	3.99E+08	6.29E+08	8.87E+06	4.17E+08	4.50E+07	**6.10E+06**	4.39E+06
	Worst	5.67E+09	4.86E+09	3.20E+08	5.35E+09	4.16E+08	**3.66E+07**	1.18E+08
	Mean	2.83E+09	1.77E+09	1.03E+08	2.48E+09	2.04E+08	**1.79E+07**	2.96E+07
	SD	1.88E+09	1.34E+09	9.21E+07	1.44E+09	1.39E+08	**1.01E+07**	3.58E+07
F3	Best	9.96E+02	7.03E+03	2.46E+03	2.40E+03	2.69E+03	7.48E+03	**6.16E+02**
	Worst	1.81E+04	2.96E+04	2.15E+04	1.15E+04	1.79E+04	1.61E+04	**1.08E+04**
	Mean	6.09E+03	1.35E+04	8.52E+03	8.32E+03	5.94E+03	1.26E+04	**4.15E+03**
	SD	6.14E+03	7.21E+03	5.99E+03	2.76E+03	4.70E+03	2.72E+03	**2.86E+03**
F4	Best	6.50E+00	9.65E+00	**8.19E+00**	5.91E+01	1.34E+01	3.82E+00	1.01E+01
	Worst	1.57E+02	4.00E+02	**6.08E+01**	2.53E+02	1.49E+02	1.77E+02	8.16E+01
	Mean	6.22E+01	1.62E+02	**3.02E+01**	1.31E+02	7.53E+01	4.46E+01	3.48E+01
	SD	5.76E+01	1.07E+02	**2.17E+01**	6.49E+01	4.62E+01	5.63E+01	2.48E+01
F5	Best	2.11E+01	3.20E+01	3.14E+01	2.82E+01	3.29E+01	4.01E+01	**3.37E+00**
	Worst	9.50E+01	6.34E+01	6.82E+01	1.12E+02	8.21E+01	1.07E+02	**6.31E+00**
	Mean	5.56E+01	4.57E+01	4.56E+01	6.06E+01	6.18E+01	5.97E+01	**4.43E+00**
	SD	2.37E+01	1.03E+01	1.23E+01	2.75E+01	1.78E+01	2.16E+01	**3.48E+00**
F6	Best	1.31E+01	2.45E+01	1.32E+01	3.09E+01	1.22E+01	1.82E+01	**1.12E+00**
	Worst	7.06E+01	6.69E+01	4.47E+01	5.33E+01	5.80E+01	5.82E+01	**4.40E+00**
	Mean	3.73E+01	4.87E+01	3.03E+01	4.19E+01	3.16E+01	3.65E+01	**2.52E+00**
	SD	1.86E+01	1.56E+01	1.05E+01	8.03E+00	1.45E+01	1.14E+01	**1.33E+00**
F7	Best	7.39E+01	7.57E+01	5.97E+01	6.70E+01	**4.84E+01**	5.36E+01	5.48E+01
	Worst	1.43E+02	1.25E+02	1.16E+02	1.31E+02	**8.46E+01**	9.57E+01	1.65E+02
	Mean	1.12E+02	9.49E+01	8.59E+01	9.87E+01	**6.78E+01**	7.46E+01	9.37E+01
	SD	2.19E+01	1.69E+01	2.18E+01	1.60E+01	**1.27E+01**	1.54E+01	3.66E+01
F8	Best	2.09E+01	1.50E+01	2.94E+01	3.23E+01	2.32E+01	1.74E+01	**2.42E+00**
	Worst	8.71E+01	6.78E+01	6.71E+01	6.71E+01	7.27E+01	7.30E+01	**6.49E+00**
	Mean	4.69E+01	4.15E+01	4.28E+01	4.98E+01	4.69E+01	5.18E+01	**3.87E+00**
	SD	2.25E+01	1.58E+01	1.18E+01	1.06E+01	1.40E+01	1.69E+01	**1.16E+00**
F9	Best	1.64E+02	3.17E+02	1.38E+02	1.99E+02	2.72E+02	9.43E+01	**1.22E+01**
	Worst	1.72E+03	7.67E+02	1.06E+03	9.64E+02	9.39E+02	8.10E+02	**8.20E+01**
	Mean	6.54E+02	5.50E+02	4.99E+02	5.70E+02	7.04E+02	5.17E+02	**4.74E+01**
	SD	4.44E+02	1.72E+02	3.45E+02	2.58E+02	1.96E+02	2.24E+02	**2.50E+01**
F10	Best	1.14E+03	1.01E+03	5.61E+02	7.17E+02	7.74E+02	8.82E+02	**7.57E+02**
	Worst	1.78E+03	1.86E+03	1.66E+03	1.77E+03	1.70E+03	1.79E+03	**1.46E+03**
	Mean	1.46E+03	1.39E+03	1.15E+03	1.47E+03	1.20E+03	1.35E+03	**1.10E+03**
	SD	2.36E+02	3.21E+02	3.32E+02	3.05E+02	3.11E+02	2.34E+02	**2.33E+02**

In comparison of 500-dimensional optimization performance of 25 functions, WOA2 and WOALam have better optimization performance than WOA. The 13 functions optimized by WOA1 are better than WOA, as shown in Figure 5. The optimization performance of the remaining 12 functions is similar to that of WOA. From the comparison results, it can be seen that the good point set initialization method is beneficial to improve the performance of the algorithm in

TABLE 10. Mean and standard deviation of the optimization functions with 30D.

CEC 2017 Function	Result	Algorithms						
		WOA	MWOA	PDWOA	OBWOA	WOASA	EWOA	WOALam
F1	Best	3.07E+10	1.78E+10	6.73E+09	9.16E+09	5.34E+09	**1.28E+09**	1.40E+09
	Worst	5.82E+10	4.37E+10	1.43E+10	3.93E+10	9.07E+09	**3.57E+09**	5.33E+09
	Mean	4.09E+10	2.66E+10	1.03E+10	3.16E+10	6.55E+09	**2.23E+09**	3.10E+09
	SD	8.81E+09	6.79E+09	2.58E+09	8.97E+09	1.24E+09	**9.14E+08**	1.28E+09
F3	Best	1.29E+05	1.75E+05	9.77E+04	9.26E+04	1.55E+05	1.72E+05	**7.07E+03**
	Worst	4.22E+05	4.14E+05	3.53E+05	2.98E+05	3.92E+05	3.62E+05	**9.52E+03**
	Mean	2.41E+05	2.58E+05	2.39E+05	1.98E+05	2.79E+05	2.57E+05	**8.09E+03**
	SD	9.07E+04	7.16E+04	8.11E+04	7.70E+04	6.48E+04	5.41E+04	**9.41E+02**
F4	Best	3.01E+02	3.16E+03	1.24E+03	2.30E+03	7.96E+02	1.27E+03	**3.40E+02**
	Worst	1.24E+03	9.33E+03	3.93E+03	1.16E+04	1.82E+03	8.72E+03	**8.21E+02**
	Mean	6.10E+02	5.53E+03	2.12E+03	7.57E+03	1.15E+03	4.48E+03	**5.77E+02**
	SD	2.64E+02	1.81E+03	8.44E+02	3.46E+03	3.23E+02	2.31E+03	**1.33E+02**
F5	Best	2.32E+02	3.19E+02	2.85E+02	3.13E+02	2.47E+02	2.65E+02	**2.49E+01**
	Worst	3.78E+02	4.86E+02	4.78E+02	4.56E+02	3.90E+02	4.75E+02	**3.87E+01**
	Mean	3.36E+02	3.76E+02	3.42E+02	3.70E+02	3.28E+02	3.43E+02	**3.08E+01**
	SD	4.97E+01	5.09E+01	5.23E+01	4.91E+01	4.79E+01	5.88E+01	**4.90E+00**
F6	Best	6.93E+01	6.33E+01	6.20E+01	6.84E+01	5.40E+01	6.59E+01	**1.58E+00**
	Worst	9.45E+01	8.93E+01	8.27E+01	9.47E+01	9.75E+01	9.35E+01	**9.46E+00**
	Mean	8.10E+01	7.89E+01	7.17E+01	8.32E+01	7.42E+01	8.32E+01	**3.89E+00**
	SD	9.93E+00	8.13E+00	6.98E+00	7.78E+00	1.37E+01	8.91E+00	**2.31E+00**
F7	Best	4.73E+02	5.16E+02	5.57E+02	5.55E+02	4.43E+02	4.30E+02	**4.16E+01**
	Worst	6.84E+02	7.15E+02	8.56E+02	8.18E+02	7.21E+02	7.42E+02	**5.98E+01**
	Mean	5.71E+02	6.45E+02	6.83E+02	6.85E+02	6.02E+02	6.23E+02	**5.11E+01**
	SD	6.00E+01	6.10E+01	8.53E+01	8.36E+01	1.11E+02	1.05E+02	**1.38E+01**
F8	Best	1.88E+02	2.25E+02	2.50E+02	2.67E+02	2.34E+02	1.98E+02	**1.86E+02**
	Worst	3.27E+02	3.35E+02	3.61E+02	3.82E+02	3.26E+02	3.60E+02	**3.12E+02**
	Mean	2.62E+02	2.80E+02	2.83E+02	3.24E+02	2.82E+02	3.06E+02	**2.43E+02**
	SD	3.99E+01	4.20E+01	3.43E+01	3.29E+01	3.08E+01	4.84E+01	**4.13E+01**
F9	Best	6.12E+03	5.29E+03	6.25E+03	8.66E+03	4.51E+03	4.85E+03	**6.11E+02**
	Worst	1.79E+04	1.05E+04	1.38E+04	1.45E+04	1.17E+04	1.77E+04	**1.29E+03**
	Mean	1.04E+04	7.93E+03	9.01E+03	1.04E+04	9.68E+03	1.24E+04	**7.63E+02**
	SD	4.17E+03	1.67E+03	2.55E+03	1.76E+03	1.99E+03	4.07E+03	**1.99E+02**
F10	Best	6.13E+03	6.62E+03	6.17E+03	6.94E+03	5.79E+03	5.62E+03	**4.48E+03**
	Worst	8.23E+03	8.46E+03	6.74E+03	8.34E+03	7.43E+03	7.91E+03	**7.87E+03**
	Mean	7.31E+03	7.38E+03	6.40E+03	7.45E+03	6.69E+03	6.87E+03	**6.36E+03**
	SD	7.32E+02	6.50E+02	1.96E+02	5.03E+02	5.63E+02	6.99E+02	**1.03E+03**

most cases. The proposed Lamarckian local search strategy has the advantages of speeding up and strengthening local search. Improvement of individual random selection method has stronger local escape ability and can effectively prevent the population from entering the predicament of local optimum. The combination of these three improved methods can obtain better optimization results.

C. COMPARISON OF FIXED ITERATION NUMBER WITH SIX IMPROVED WHALE ALGORITHMS

In experimental study, We compared WOALam with six WOA variants. All the functions are tested in 50 dimensions, 100 dimensions and 500 dimensions. Limited to the length of the paper, only the best value, worst value, mean and standard deviation (SD) of the 500-dimensional optimization functions are given in this paper. They are reported in TABLES 3 and 4. The mean and standard deviation (SD) of the 50 and 100 dimensional optimization functions are reported in TABLES 5 and 6. The best ones are written in bold.

1) COMPARISON AND ANALYSIS OF EXPERIMENTAL RESULTS

From TABLES 3-6, it can be observed that the proposed algorithm WOALam is superior to the other 6 WOA algorithms in most cases. As can be seen from TABLE 3, the WOALam

TABLE 11. Mean and standard deviation of the optimization functions with 50D.

CEC 2017 Function	Result	Algorithms						
		WOA	MWOA	PDWOA	OBWOA	WOASA	EWOA	WOALam
F1	Best	7.37E+10	4.32E+10	2.84E+10	4.78E+10	1.52E+10	**9.08E+09**	1.04E+10
	Worst	9.35E+10	8.60E+10	5.31E+10	1.01E+11	2.86E+10	**1.64E+10**	1.96E+10
	Mean	8.44E+10	6.89E+10	3.91E+10	7.44E+10	2.18E+10	**1.21E+10**	1.48E+10
	SD	5.96E+09	1.23E+10	7.90E+09	1.73E+10	3.70E+09	**2.29E+09**	3.28E+09
F3	Best	1.80E+05	1.82E+05	1.61E+05	2.37E+05	1.85E+05	2.41E+05	**1.55E+04**
	Worst	3.24E+05	4.06E+05	6.54E+05	4.05E+05	4.28E+05	4.42E+05	**2.83E+04**
	Mean	2.67E+05	2.95E+05	3.43E+05	2.87E+05	2.83E+05	3.05E+05	**2.17E+04**
	SD	4.66E+04	7.37E+04	1.40E+05	4.70E+04	9.13E+04	5.80E+04	**3.98E+01**
F4	Best	2.09E+03	1.25E+04	3.04E+03	1.74E+04	3.38E+03	2.17E+03	**1.66E+03**
	Worst	2.87E+04	2.82E+04	1.08E+04	3.52E+04	9.21E+03	4.36E+03	**4.20E+03**
	Mean	8.20E+03	1.97E+04	6.85E+03	2.69E+04	5.14E+03	3.33E+03	**2.86E+03**
	SD	8.25E+03	5.81E+03	2.14E+03	5.64E+03	1.73E+03	7.78E+02	**8.28E+02**
F5	Best	4.55E+02	5.38E+02	5.53E+02	5.95E+02	4.60E+02	5.03E+02	**3.71E+02**
	Worst	9.61E+02	7.15E+02	6.73E+02	7.25E+02	7.02E+02	7.83E+02	**7.33E+02**
	Mean	5.89E+02	6.17E+02	6.06E+02	6.81E+02	6.05E+02	6.46E+02	**5.13E+02**
	SD	1.42E+02	5.95E+01	4.16E+01	3.46E+01	6.98E+01	1.12E+02	**1.37E+02**
F6	Best	8.16E+01	8.24E+01	7.92E+01	7.91E+01	8.13E+01	7.43E+01	**1.34E+01**
	Worst	1.10E+02	1.03E+02	9.60E+01	1.16E+02	1.06E+02	1.05E+02	**5.18E+01**
	Mean	9.73E+01	9.16E+01	8.93E+01	9.88E+01	9.66E+01	9.09E+01	**3.18E+01**
	SD	8.43E+00	6.27E+00	5.57E+00	1.22E+01	8.87E+00	1.07E+01	**1.37E+01**
F7	Best	1.07E+03	1.06E+03	1.12E+03	1.20E+03	9.37E+02	9.81E+02	**6.28E+01**
	Worst	1.40E+03	1.23E+03	1.42E+03	1.42E+03	1.20E+03	1.37E+03	**1.38E+02**
	Mean	1.22E+03	1.17E+03	1.26E+03	1.34E+03	1.07E+03	1.23E+03	**9.82E+01**
	SD	1.07E+02	5.74E+01	9.68E+01	6.08E+01	9.34E+01	1.20E+02	**2.46E+01**
F8	Best	4.76E+02	4.91E+02	4.96E+02	6.11E+02	5.18E+02	4.60E+02	**3.96E+01**
	Worst	6.52E+02	7.00E+02	6.70E+02	7.08E+02	6.44E+02	8.07E+02	**5.68E+01**
	Mean	5.68E+02	6.00E+02	6.01E+02	6.68E+02	5.75E+02	6.21E+02	**4.73E+01**
	SD	5.47E+01	6.19E+01	5.15E+01	3.29E+01	4.15E+01	1.06E+02	**5.84E+00**
F9	Best	2.04E+04	2.79E+04	1.95E+04	2.96E+04	1.93E+04	2.27E+04	**1.67E+03**
	Worst	5.30E+04	4.03E+04	4.03E+04	4.15E+04	4.03E+04	5.20E+04	**3.31E+03**
	Mean	3.53E+04	3.59E+04	3.05E+04	3.59E+04	3.25E+04	3.64E+04	**2.41E+03**
	SD	9.45E+03	4.30E+03	6.02E+03	4.04E+03	6.25E+03	8.31E+03	**4.90E+02**
F10	Best	1.01E+04	1.15E+04	1.10E+04	1.26E+04	1.08E+04	1.14E+04	**8.92E+03**
	Worst	1.37E+04	1.47E+04	1.32E+04	1.48E+04	1.38E+04	1.27E+04	**1.01E+04**
	Mean	1.21E+04	1.32E+04	1.21E+04	1.38E+04	1.26E+04	1.19E+04	**9.01E+03**
	SD	1.37E+03	9.44E+02	8.54E+02	7.44E+02	1.01E+03	4.59E+02	**1.82E+03**

algorithm has better or similar results compared to the other 6 WOA algorithms for the 10 tested functions, except for F14, F15, F19 and F20. But for WOALam, the best result of these four functions have reached the optimal value more than once, the solution obtained by the algorithm is very close to the global optimal solution. At the same time, it can be seen in Table 3 that for some functions, the optimal results of WOA, MWOA, PDWOA, OBWOA and WOASA are very close to WOALam, but the variance is somewhat different. The results are written in underline. It should be pointed out that, compared with other algorithms, the fitness is evaluated twice in each iteration of OBWOA. But even so, the results of WOALam are better than that of OBWOA. As can be seen from Figure 6, OBWOA shows better optimization performance before *MaxT*/2 iteration. Even though the fitness is evaluated twice in each iteration, the optimization performance of the OBWOA decreases significantly or remains basically unchanged after *MaxT* /2 iteration. It shows that although the improved mechanism based on opposition learning can improve the performance of WOA in the early stage of optimization, it can not overcome the shortcomings of WOA in solving high-dimensional optimization problems in the later stage of iteration.

As can be seen from TABLE 4, taking the best, worst, mean and standard deviation (SD) criteria into account, the WOALam algorithm outperforms the other WOA

TABLE 12. Mean and standard deviation of the optimization functions with 100D.

CEC 2017 Function	Result	Algorithms						
		WOA	MWOA	PDWOA	OBWOA	WOASA	EWOA	WOALam
F1	Best	8.33E+10	1.76E+11	1.37E+11	2.09E+11	8.82E+10	6.28E+10	**1.66E+10**
	Worst	1.07E+11	2.29E+11	1.89E+11	2.56E+11	1.10E+11	1.04E+11	**2.65E+10**
	Mean	9.45E+10	2.11E+11	1.64E+11	2.30E+11	9.76E+10	8.30E+10	**2.25E+10**
	SD	6.95E+09	1.55E+10	1.74E+10	1.67E+10	7.17E+09	1.38E+10	**3.27E+08**
F3	Best	7.75E+05	7.44E+05	7.58E+05	3.98E+05	5.69E+05	7.92E+05	**3.59E+04**
	Worst	1.19E+06	1.08E+06	1.38E+06	9.28E+05	1.22E+06	1.29E+06	**3.78E+04**
	Mean	9.76E+05	8.91E+05	9.74E+05	6.93E+05	8.99E+05	9.68E+05	**3.68E+04**
	SD	1.23E+05	1.38E+05	2.21E+05	2.06E+05	1.68E+05	1.35E+05	**2.79E+01**
F4	Best	1.12E+04	4.47E+04	2.78E+04	4.95E+04	1.87E+04	1.08E+04	**2.36E+03**
	Worst	2.21E+04	9.37E+04	4.79E+04	9.95E+04	3.01E+04	1.95E+04	**1.21E+04**
	Mean	1.52E+04	6.18E+04	3.34E+04	7.57E+04	2.28E+04	1.43E+04	**3.67E+03**
	SD	3.35E+03	1.52E+04	6.38E+03	1.42E+04	3.46E+03	2.60E+03	**2.23E+03**
F5	Best	1.30E+03	1.36E+03	1.39E+03	1.51E+03	1.37E+03	1.28E+03	**9.44E+02**
	Worst	1.53E+03	1.60E+03	1.57E+03	1.64E+03	1.57E+03	1.74E+03	**1.59E+03**
	Mean	1.44E+03	1.50E+03	1.49E+03	1.56E+03	1.46E+03	1.45E+03	**1.17E+03**
	SD	8.17E+01	7.96E+01	5.80E+01	4.28E+01	7.21E+01	1.23E+02	**2.66E+02**
F6	Best	9.66E+01	9.13E+01	9.16E+01	9.91E+01	9.42E+01	1.06E+02	**1.69E+01**
	Worst	1.35E+02	1.08E+02	1.12E+02	1.26E+02	1.16E+02	1.37E+02	**1.05E+02**
	Mean	1.11E+02	1.01E+02	1.01E+02	1.11E+02	1.04E+02	1.17E+02	**5.18E+01**
	SD	1.40E+01	5.35E+00	5.61E+00	8.08E+00	7.07E+00	9.36E+00	**2.94E+01**
F7	Best	2.94E+03	2.89E+03	2.89E+03	3.06E+03	2.72E+03	2.98E+03	**1.57E+03**
	Worst	3.27E+03	3.15E+03	3.28E+03	3.36E+03	3.20E+03	3.34E+03	**2.84E+03**
	Mean	3.05E+03	2.98E+03	3.17E+03	3.23E+03	3.02E+03	3.12E+03	**2.22E+03**
	SD	1.27E+02	7.54E+01	1.20E+02	1.05E+02	1.55E+02	1.19E+02	**4.61E+02**
F8	Best	1.42E+03	1.55E+03	1.52E+03	1.65E+03	1.43E+03	1.38E+03	**1.02E+02**
	Worst	1.74E+03	1.74E+03	1.73E+03	1.90E+03	1.74E+03	1.85E+03	**1.73E+02**
	Mean	1.59E+03	1.63E+03	1.64E+03	1.74E+03	1.57E+03	1.60E+03	**1.31E+02**
	SD	1.18E+02	6.41E+01	6.44E+01	6.82E+01	8.47E+01	1.56E+02	**2.46E+01**
F9	Best	6.04E+04	6.10E+04	5.85E+04	7.05E+04	5.24E+04	5.67E+04	**4.23E+03**
	Worst	1.13E+05	8.13E+04	7.25E+04	8.81E+04	9.41E+04	1.59E+05	**7.37E+03**
	Mean	8.62E+04	7.28E+04	6.51E+04	7.65E+04	6.96E+04	9.37E+04	**5.93E+03**
	SD	1.82E+04	6.69E+03	4.05E+03	5.72E+03	1.36E+04	3.20E+04	**8.92E+02**
F10	Best	2.69E+04	2.88E+04	2.65E+04	2.97E+04	2.17E+04	2.40E+04	**2.07E+04**
	Worst	3.07E+04	3.10E+04	3.11E+04	3.29E+04	3.04E+04	3.10E+04	**2.69E+04**
	Mean	2.89E+04	3.00E+04	2.94E+04	3.09E+04	2.72E+04	2.83E+04	**2.29E+04**
	SD	2.57E+03	8.01E+02	1.71E+03	9.38E+02	1.18E+03	1.94E+03	**3.18E+02**

algorithms for 25 functions, except for F5, F7, F33 and F35. For F9,F11, F29, F30 and F32, the results of several other algorithms are the same as WOALam, but WOALam has fewer iterations than them. Similar conclusions can be obtained according to TABLES 5 and 6, and will not be repeated here. This indicates that WOALam algorithm has strong global searching ability and stability. Though some algorithms perform better than the WOALam algorithm in some special cases, WOALam algorithm has an overall edge in terms of performance. According to the above observations, we concluded that the performance of the WOALam is superior to the other six methods when used to solve most of these optimization problems. When the dimension is increased, optimization performance of WOALam is relatively stable. It can also get an ideal solution.

2) CONVERGENCE ANALYSIS

The convergence curve of 25 optimization functions for 500-dimensional is given in Figure 6. As depicted in Figure 6, WOALam not only converges quickly towards the global optimal solution but also achieves higher accuracy. On the contrary, the compared algorithms converge easily to the local optima for most functions, and the convergence speed will be very slow when they converge toward the global optimal solution. From the Figure 6, it is obvious that WOALam converges very fast and reaches optimal results for the most

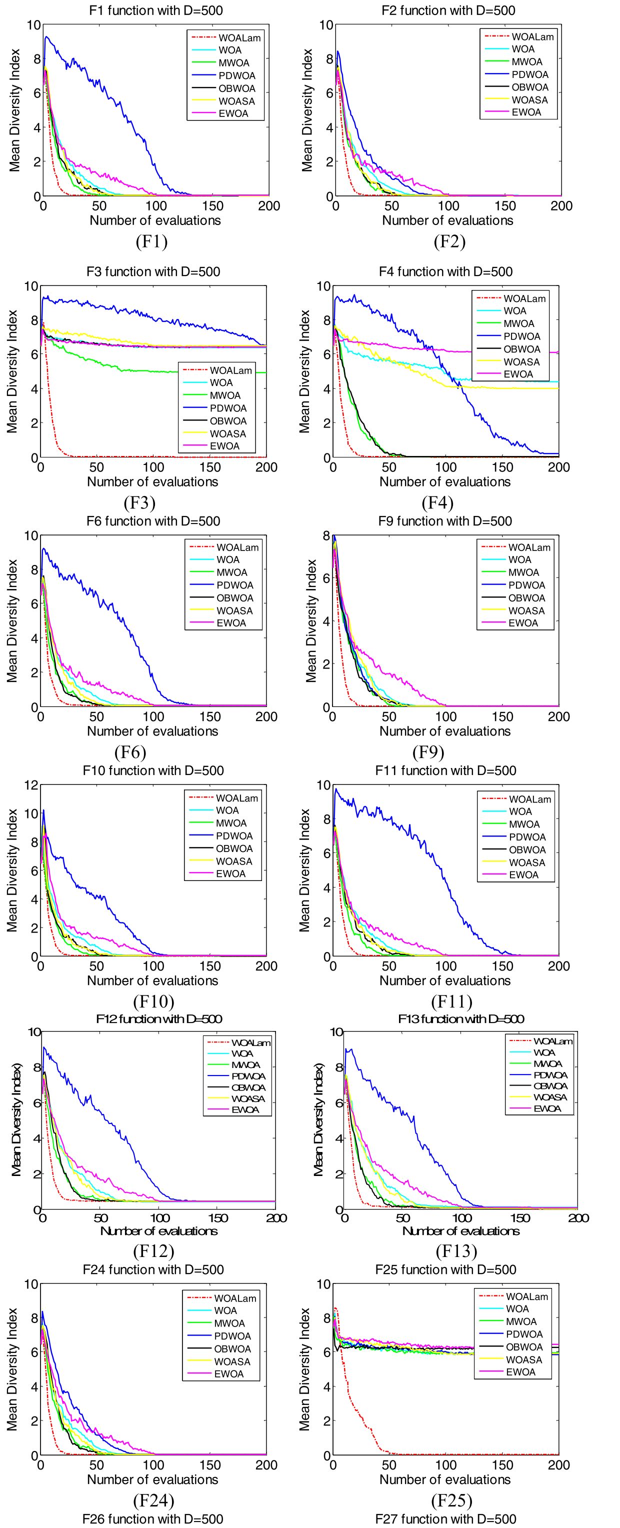


FIGURE 7. **Comparison of mean diversity index with different algorithms.**

functions, especially for the functions F1, F2, F3, F4, F9, F11, F25, F26, F27, F28, F29, F30, F31, F32 and F34. The WOALam has strong exploitation power (the convergence curve keeps decreasing) in the later stage of search process.

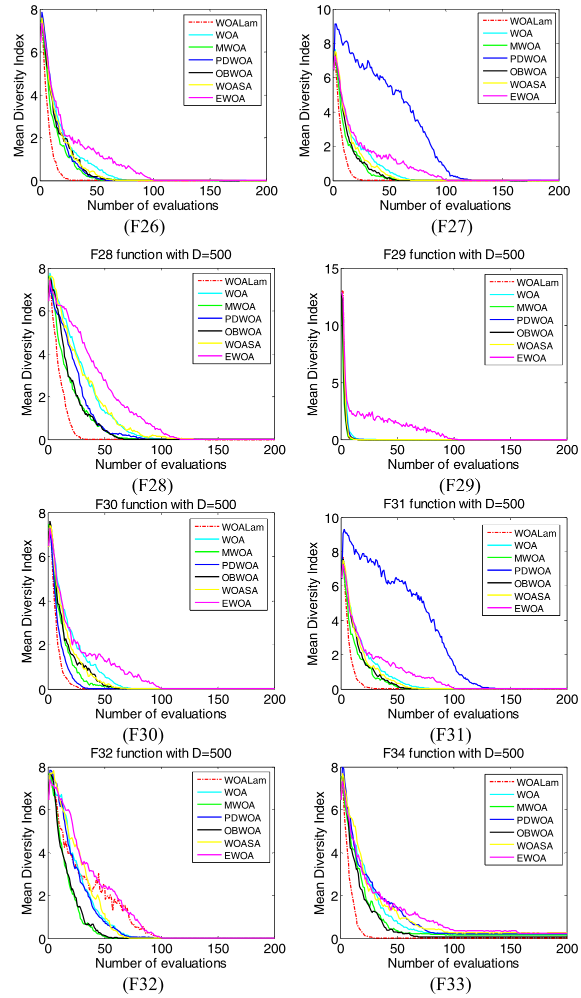


FIGURE 7. ***(Continued.)*** **Comparison of mean diversity index with different algorithms.**

But for functions F5, F7, F33 and F35, the result of WOALam is not superior to other comparison algorithms. It also shows that "there is no free lunch theorem" from other side. But from the iterative curves of functions F5, F7, and F35, it can be seen that the iteration curve of WOALam is not much different from the best iteration curve obtained. From the overall optimization results, the WOALam algorithm has good optimization performance compared with six comparison algorithms, especially in the process of dealing with high-dimensional functions with good adaptability.

From the iteration curve of Figure 6, it can be seen that other six algorithms can hardly jump out of the local optimum after reaching a certain precision. However, the iteration curve of WOALam shows that even if the dimension reaches 500, the algorithm can update the optimal solution in a faster iterative manner. And it can be seen that WOALam can get better results with a smaller number of iterations. This shows that the local search technology based on lamarckian learning can effectively enhance the individual optimization performance and avoid some local oscillation computation.

This optimization characteristic is especially suitable for optimization problems with complex fitness computation.

3) WILCOXON's RANK SUM TEST RESULTS

To estimate the statistical significance difference between the algorithms, the Wilcoxon rank sum test (Derrac *et al.* [56]) at the 5% significance level was employed to determine whether the difference between other results and the WOALam result is significant. Specifically, An h value of 1 or -1 indicates whether the results achieved by WOALam are significantly better or worse than those of the compared algorithms. An h value of 0 means that no significant difference exists between WOALam and the compared algorithms. In addition, if the optimization results of the comparative algorithms are exactly the same, the p-value is NaN.The results of comparing WOALam with the other six algorithms on the benchmark functions with D=100, 500 are presented in TABLES 7-8 respectively.

4) COMPARISON OF DIVERSITY INDEX WITH DIFFERENT ALGORITHMS

A diversity index [57] is defined in order to assess the exploration/exploitation behavior exhibited by the each algorithm in the search process. The mathematical formula for diversity index is as follows:

$$Diversityindex = \frac{1}{N}\sum_{j=1}^{N}\sqrt{\sum_{i=1}^{Dim}\left(\frac{GB(i)-X_j(i)}{X_{i,max}-X_{i,min}}\right)^2} \quad (16)$$

where $X_j(i)$ is the value of the *ith* variable of the *jth*individual; *GB(i)* is the value of the *ith* variable of global optimal individual; $X_{i,min}$ and $X_{i,max}$ are the minimum and maximum values of the *ith* variable, respectively; *Dim* is the number of design variables and N is the number of individuals. According to formula (15), if the optimal solution is found, the value of diversity index is 0. The mean values for the diversity indices are plotted against iteration numbers for all of the algorithms in different examples. It will be seen that the differences between the performances of algorithms can be interpreted by analyzing the differences between their diversity index curves to some extent. Twenty functions with better optimization results than other algorithms are selected as shown in Figure 7. According to Table 4 and Figure 7, In the process of WOALam optimization, high values of diversity are provided in the early stages of the optimization process. As the optimization process continues, the individuals focus on more promising regions of the search space in order to perform local search and diversity index values gradually decrease. As can be seen from Table 4, the results of several other algorithms are the same as WOALam for F9,F11, F29, F30 and F32, but WOALam has fast convergence speed from Figure 7.For PDWOA and EWOA the curves maintain a relatively high distance from the horizontal axis in the process of optimization. This means that the individuals are far apart from global optimal location and the algorithms do not perform a proper local search phase.

D. COMPARISON OF CEC2017 BENCHMARK FUNCTIONS AMONG VARIANTS OF THE WHALE ALGORITHM

In experimental study, WOALam was experimentally validated using F1, F3-F10 of optimization problems for the CEC 2017 special session. F2 has been excluded because it shows unstable behavior especially for higher dimensions. Matlab codes for CEC'17 test suite can be downloaded from the website given in [58]. Search range: $[-100, 100]$. All the functions are tested in 10 dimensions, 30 dimensions, 50 dimensions and 100 dimensions in the literature. Among them, F1 and F3 are unimodal functions. F1 is shifted and rotated bent cigar function, F3 is shifted and rotated zakharov function. F4 – F10 are multimodal functions. F4 is shifted and rotated rosenbrock's function, F5 is shifted and rotated rastrigin's function, F6 is shifted and rotated expanded scaffer's function, F7 is shifted and rotated lunacek bi_rastrigin function, F8 is shifted and rotated non-continuous rastrigin's function, F9 is shifted and rotated levy function, F10 is shifted and rotated schwefel's function.To evaluate the performance of the algorithm WOALam, six variants of WOA algorithms were used for comparison. The parameter settings of various algorithms are the same as those above.

It is evident from Table 9-12 that WOALam is very efficient and outperforms six other algorithms. For functions F3, F4, F5, F6, F7, F8, F9, F10, the solution obtained by WOALam algorithm is superior to that obtained by other algorithms in the performance comparison of 30, 50 and 100-dimensional optimization. In the comparison of 10 and 30-dimensional optimization performance, the solution obtained by EWOA algorithm is superior to that obtained by other algorithms for function F1. In the comparison of 10-dimensional optimization performance, PDWOA obtained relatively better results for F4, WOASA obtained relatively better results for F7. However, From table 9, The mean values of WOALam, PDWOA and EWOA are similar on F1,F4,F7, but the variance of PDWOA and EWOA is smaller than WOALam. Also from Table 10-12, we know that the optimization performance of WOALam is better than other algorithms with the increase of dimension. Statistical results show that the proposed algorithm has better performance than others in solving high-dimensional function optimization problems. From the results, although the seven algorithms have better optimization performance for solving Zakharov function, Rosenbrock function, rastrigin, levy function, schwefel's function and so on from TABLE4-6, the optimization effect is not good for their shifted and rotated functions. The reason for the analysis is that it is influenced by the evolutionary behavior of whale optimization algorithm. Therefore, it is a good research direction to improve the evolutionary behavior of the WOA algorithm itself in the future.

V. CONCLUSIONS

In this paper, Whale Optimization Algorithm based on Lamarckian learning (WOALam) is proposed to enhance the performance of WOA. In the WOALam, population distribution is homogenized through population initialization based on good point set theory. It is helpful for enhancing the ergodicity of the solution space and finding the global optimal solution better. The individual evaluation mechanism of development potential is proposed based on lamarck theory, and the excellent experience in the evolution process is directly passed on to the next generation through local search strategy. This learning mechanism has the advantages of speeding up and strengthening local search. Improvement of individual random selection method has stronger local escape ability and can effectively prevent the population from entering the predicament of local optimum. At the same time, the reasons for affecting the efficiency and speed of algorithm optimization are analyzed, and corresponding improvement measures are proposed. The test results based on the high dimension benchmark function showed that the proposed algorithm has a better global optimization ability and a faster convergence speed, and can obtain more satisfactory optimization results in less iterative times. Thus, it is suitable for solving high dimensional optimization problems, which some fitness calculations are complex and time-consuming. It has also carried out new explorations and attempts for the study of evolutionary computation. In addition, it is a good research direction to improve the evolutionary behavior of the WOA algorithm itself in the future.

REFERENCES

[1] A. Kaveh, *Advances in Metaheuristic Algorithms for Optimal Design of Structures*. 2nd ed. Cham, Switzerland: Springer, 2017, pp. 1–5.

[2] A. Kaveh, M. A. M. Share, and M. Moslehi, "Magnetic charged system search: A new meta-heuristic algorithm for optimization," *Acta Mechanica*, vol. 224, no. 1, pp. 85–107, Jan. 2013. doi: 10.1007/s00707-012-0745-6.

[3] A. Kaveh and N. Farhoudi, "A new optimization method: Dolphin echolocation," *Adv. Eng. Softw.*, vol. 59, no. 5, pp. 53–70, May 2013. doi: 10.1016/j.advengsoft.2013.03.004.

[4] A. Kaveh and V. R. Mahdavi, "Colliding bodies optimization: A novel meta-heuristic method," *Comput. Struct.*, vol. 139, pp. 18–27, Jan. 2014. doi: 10.1016/j.compstruc.2014.04.005.

[5] X. Meng, Y. Liu, X. Gao, and H. Zhang, "A new bio-inspired algorithm: Chicken swarm optimization," in *Advances in Swarm Intelligence*. Hefei, China, Oct. 2014, pp. 86–94.

[6] A. Kaveh and T. Bakhshpoori, "Water evaporation optimization: A novel physically inspired optimization algorithm," *Comput. Struct.*, vol. 167, pp. 69–85, Apr. 2016. doi: 10.1016/j.compstruc.2016.01.008.

[7] A. S. Eesa, A. M. A. Brifcani, and Z. Orman, "Cuttlefish algorithm—A novel bio-inspired optimization algorithm," *Int. J. Sci. Eng. Res.*, vol. 4, no. 9, pp. 1978–1986, Sep. 2013.

[8] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, no. 3, pp. 46–61, Mar. 2014. doi: 10.1016/j.advengsoft.2013.12.007.

[9] W.-T. Pan, "A new fruit fly optimization algorithm: Taking the financial distress model as an example," *Knowl.-Based Syst.*, vol. 26, pp. 69–74, Feb. 2012. doi: 10.1016/j.knosys.2011.07.001.

[10] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, 2016. doi: 10.1007/s00521-015-1870-7.

[11] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016. doi: 10.1016/j.advengsoft.2016.01.008.

[12] J. P. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, WA, Australia, 1995, pp. 1942–1948.

[13] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009. doi: 10.1016/j.ins.2009.03.004.

[14] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997. doi: 10.1023/a:1008202821328.

[15] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999. doi: 10.1109/4235.771163.

[16] M. Abdel-Basset, D. El-Shahat, and A. K. Sangaiah, "A modified nature inspired meta-heuristic whale optimization algorithm for solving 0–1 knapsack problem," *Int. J. Mach. Learn.*, vol. 10, no. 3, pp. 495–514, Mar. 2019. doi: 10.1007/s13042-017-0731-3.

[17] M. Abdel-Basset, G. Manogaran, D. El-Shahat, and S. Mirjalili, "A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem," *Future Gener. Comput. Syst.*, vol. 85, pp. 129–145, Aug. 2018. doi: 10.1016/j.future.2018.03.020.

[18] J. Nasiri, F. M. Khiyabani, and A. Yoshise, "A whale optimization algorithm (WOA) approach for clustering," *Cogent Math. Statist.*, vol. 5, no. 1, Jun. 2018, Art. no. 1483565. doi: 10.1080/25742558.2018.1483565.

[19] A. E. Hegazy, M. A. Makhlouf, and G. S. El-Tawel, "Dimensionality reduction using an improved whale optimization algorithm for data classification," *Int. J. Modern Educ. Comput. Sci.*, vol. 10, no. 7, pp. 37–49, Jul. 2018. doi: 10.5815/ijmecs.2018.07.04.

[20] N. M. Laskar, K. Guha, I. Chatterjee, S. Chanda, K. L. Baishnab, and P. K. Paul, "HWPSO: A new hybrid whale-particle swarm optimization algorithm and its application in electronic design optimization problems," *Appl. Intell.*, vol. 49, no. 1, pp. 265–291, Jan. 2019. doi: 10.1007/s10489-018-1247-6.

[21] H. H. Mehne and S. Mirjalili, "A parallel numerical method for solving optimal control problems based on whale optimization algorithm," *Knowl.-Based Syst.*, vol. 151, pp. 114–123, Jul. 2018. doi: 10.1016/j.knosys.2018.03.024.

[22] J. Wang, P. Du, T. Niu, and W. Yang, "A novel hybrid system based on a new proposed algorithm—Multi-objective whale optimization algorithm for wind speed forecasting," *Appl. Energy*, vol. 208, pp. 344–360, Dec. 2017. doi: 10.1016/j.apenergy.2017.10.031.

[23] Z. Yan, J. Sha, B. Liu, W. Tian, and J. Lu, "An ameliorative whale optimization algorithm for multi-objective optimal allocation of water resources in Handan, China," *Water*, vol. 10, no. 1, p. 87, Jan. 2018. doi: 10.3390/w10010087.

[24] H. Yadav, U. Lithore, and N. Agrawal, "An Enhancement of whale optimization algorithm using ANN for routing optimization in ad-hoc network," *Int. J. Adv. Technol. Eng. Explor.*, vol. 4, no. 36, pp. 161–167, Nov. 2017. doi: 10.19101/IJATEE.2017.436004.

[25] H. Zhao, S. Guo, and H. Zhao, "Energy-related CO_2 emissions forecasting using an improved LSSVM model optimized by whale optimization algorithm," *Energies*, vol. 10, no. 7, pp. 874–889, Jun. 2017. doi: 10.3390/en10070874.

[26] I. Aljarah, H. Faris, and S. Mirjalili, "Optimizing connection weights in neural networks using the whale optimization algorithm," *Soft Comput.*, vol. 22, no. 1, pp. 1–15, Jan. 2018. doi: 10.1007/s00500-016-2442-1.

[27] D. Oliva, M. A. El Aziz, and A. E. Hassanien, "Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm," *Appl. Energy*, vol. 200, pp. 141–154, Aug. 2017. doi: 10.1016/j.apenergy.2017.05.029.

[28] M. Sharawi, H. M. Zawbaa, E. Emary, H. M. Zawbaa, and E. Emary, "Feature selection approach based on whale optimization algorithm," in *Proc. 9th Int. Conf. Adv. Comput. Intell. (ICACI)*, Doha, Qatar, Feb. 2017, pp. 163–168.

[29] A. Kumar, V. Bhalla, P. Kumar, T. Bhardwaj, and N. Jangir, "Whale optimization algorithm for constrained economic load dispatch problems—A cost optimization," in *Ambient Communications and Computer Systems* (Advances in Intelligent Systems and Computing), vol. 696. Mar. 2018, pp. 353–366. doi: 10.1007/978-981-10-7386-1_31.

[30] M. A. El Aziz, A. A. Ewees, and A. E. Hassanien, "Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation," *Expert Syst. Appl.*, vol. 83, pp. 242–256, Oct. 2017. doi: 10.1016/j.eswa.2017.04.023.

[31] R. Sivalingam, S. Chinnamuthu, and S. S. Dash, "A modified whale optimization algorithm-based adaptive fuzzy logic PID controller for load frequency control of autonomous power generation systems," *Automatika*, vol. 58, no. 4, pp. 410–421, Dec. 2017. doi: 10.1080/00051144.2018.1465688.

[32] M. A. El Aziz and D. Oliva, "Parameter estimation of solar cells diode models by an improved opposition-based whale optimization algorithm," *Energy Convers. Manage.*, vol. 171, pp. 1843–1859, Sep. 2018. doi: 10.1016/j.enconman.2018.05.062.

[33] I. N. Trivedi, J. Pradeep, J. Narottam, K. Arvind, and L. Dilip, "Novel adaptive whale optimization algorithm for global optimization," *Indian J. Sci. Technol.*, vol. 9, no. 38, pp. 1–6, Aug. 2016. doi: 10.17485/ijst/2016/v9i38/101939.

[34] H. Hu, Y. Bai, and T. Xu, "A whale optimization algorithm with inertia weight," *WSEAS Trans. Comput.*, vol. 15, pp. 319–326, 2016.

[35] B. Bentouati, L. Chaib, and S. Chettih, "A hybrid whale algorithm and pattern search technique for optimal power flow problem," in *Proc. 8th Int. Conf. Modelling, Identificat. Control (ICMIC)*, Algiers, Algeria, Nov. 2016, pp. 1048–1053.

[36] M. Abdel-Basset, G. Manogaran, D. El-Shahat, and S. Mirjalili, "Integrating the whale algorithm with Tabu search for quadratic assignment problem: A new approach for locating hospital departments," *Appl. Soft Comput.*, vol. 73, pp. 530–546, Dec. 2018. doi: 10.1016/j.asoc.2018.08.047.

[37] Y. Ling, Y. Zhou, and Q. Luo, "Lévy flight trajectory-based whale optimization algorithm for global optimization," *IEEE Access*, vol. 5, pp. 6168–6186, Apr. 2017. doi: 10.1109/ACCESS.2017.2695498.

[38] Y. Sun, X. Wang, Y. Chen, and Z. Liu, "A modified whale optimization algorithm for large-scale global optimization problems," *Expert Syst. Appl.*, vol. 114, pp. 563–577, Dec. 2018. doi: 10.1016/j.eswa.2018.08.027.

[39] I. N. Trivedi, P. Jangir, A. Kumar, N. Jangir, and R. Totlani, "A novel hybrid PSO–WOA algorithm for global numerical functions optimization," in *Advances in Computer and Computational Sciences* (Advances in Intelligent Systems and Computing), vol. 554, S. Bhatia, K. Mishra, S. Tiwari, and V. Singh, Eds. Singapore: Springer, Sep. 2017, pp. 53–60. doi: 10.1007/978-981-10-3773-3_6.

[40] W.-Z. Sun, J.-S. Wang, and X. Wei, "An improved whale optimization algorithm based on different searching paths and perceptual disturbance," *Symmetry*, vol. 10, no. 6, pp. 1–31, Jun. 2018. doi: 10.3390/sym10060210.

[41] A. Kaveh and M. I. Ghazaan, "Enhanced whale optimization algorithm for sizing optimization of skeletal structures," *Mech. Based Des. Struct. Mach.*, vol. 45, no. 3, pp. 345–362, Jul. 2016. doi: 10.1080/15397734.2016.1213639.

[42] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, Oct. 2017. doi: 10.1016/j.neucom.2017.04.053.

[43] G. Kaur and S. Arora, "Chaotic whale optimization algorithm," *J. Comput. Des. Eng.*, vol. 5, no. 3, pp. 275–284, Jul. 2018. doi: 10.1016/j.jcde.2017.12.006.

[44] M. Abdel-Basset, G. Manogaran, L. Abdel-Fatah, and S. Mirjalili, "An improved nature inspired meta-heuristic algorithm for 1-D bin packing problems," *Pers. Ubiquitous Comput.*, vol. 22, nos. 5–6, pp. 1117–1132, Oct. 2018. doi: 10.1007/s00779-018-1132-7.

[45] M. Abdel-Basset, L. Abdle-Fatah, and A. K. Sangaiah, "An improved Lévy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment," *Cluster Comput.*, pp. 1–16, Jan. 2018. doi: 10.1007/s10586-018-1769-z.

[46] H. Luo-Geng and W. Yuan, *The Application of Number Theory in Modern Analysis*. Beijing, China: Science Press, 1978, pp. 1–99.

[47] X.-J. Bi and Z. Lei, "Dual population constrained optimization algorithm with hybird strategy," *Control Decis.*, vol. 30, no. 4, pp. 715–720, Apr. 2015. doi: 10.13195/j.kzyjc.2014.0223.

[48] K.-H. Liang, X. Yao, and C. Newton, "Lamarckian evolution in global optimization," in *Proc. 26th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, vol. 4, Nagoya, Japan, Oct. 2000, pp. 2975–2980.

[49] F. Liu, X. J. Zuo, and K. Zhang, "Novel genetic algorithm," *Comput. Eng. Appl.*, vol. 46, no. 12, pp. 46–50, Dec. 2010. doi: 10.3778/j.issn.1002-8331.2010.12.012.

[50] Z.-B. Luan, Q. T. Huang, H. Z. Jiang, and H. R. Li, "Mixed application of two learning mechanisms in genetic algorithm," *Syst. Eng. Electron.*, vol. 31, no. 8, pp. 1985–1988, Aug. 2009.

[51] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, nos. 2–3, pp. 235–256, May 2002. doi: 10.1023/A:1013689704352.

[52] N. R. Sturtevant, "An analysis of UCT in multi-player games," *Comput. Games*, vol. 31, no. 4, pp. 37–49, Dec. 2008. doi: 10.1007/978-3-540-87608-3_4.

[53] L. Wang, Y. Shi, and S. Liu, "An improved fruit fly optimization algorithm and its application to joint replenishment problems," *Expert Syst. Appl.*, vol. 42, no. 9, pp. 4310–4323, Jun. 2015. doi: 10.1016/j.eswa.2015.01.048.

[54] A. S. Eesa, A. M. A. Brifcani, and Z. Orman, "A novel bio-inspired optimization algorithm," *Int. J. Sci. Eng. Res.*, vol. 4, no. 9, pp. 1978–1986, Jun. 2013.

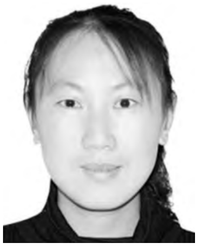
[55] X. Meng, Y. Liu, X. Gao, and H. Zhang, "A new bio-inspired algorithm: Chicken swarm optimization," in *Advances in Swarm Intelligence*, vol. 8794. Cham, Switzerland: Springer, Oct. 2014, pp. 86–94. doi: 10.1007/978-3-319-11857-4_10.

[56] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011. doi: 10.1016/j.swevo.2011.02.002.

[57] A. Kaveh and A. Zolghadr, "Comparison of nine meta-heuristic algorithms for optimal design of truss structures with frequency constraints," *Adv. Eng. Softw.*, vol. 76, no. 9, pp. 9–30, Oct. 2014. doi: 10.1016/j.advengsoft.2014.05.012.

[58] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization," Nanyang Technol. Univ., Singapore Jordan Univ. Sci. Technol. Zhengzhou Univ., Zhengzhou, China, Tech. Rep., 2016.

QIANG ZHANG received the M.S. and Ph.D. degrees in computer science from Northeast Petroleum University, Daqing, China, in 2015, where he is currently an Associate Professor. His research interests include computational intelligence, evolutionary algorithms, and machine learning.



LIJIE LIU received the M.S. degree in computer software and theory from Harbin Engineering University, Harbin, China, in 2012. Her current research interests include neural networks and evolutionary algorithms.