

Received March 18, 2017, accepted April 10, 2017, date of publication April 19, 2017, date of current version May 17, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2695498

# Lévy Flight Trajectory-Based Whale Optimization Algorithm for Global Optimization

**YING LING<sup>1</sup>, YONGQUAN ZHOU<sup>1,2</sup>, AND QIFANG LUO<sup>1,2</sup>**

<sup>1</sup>College of Information Science and Engineering, Guangxi University for Nationalities, Nanning 530006, China

<sup>2</sup>Key Laboratories of Guangxi High Schools Complex System and Computational Intelligence, Nanning 530006, China

Corresponding author: Yongquan Zhou (yongquanzhou@126.com)

This work was supported in part by the National Science Foundation of China under Grant 61463007 and Grant 61563008 and in part by the Project of Guangxi Natural Science Foundation under Grant 2016GXNSFAA380264.

**ABSTRACT** The whale optimization algorithm (WOA) has been shown to be powerful in searching for an optimal solution. This paper proposes an improvement to the whale optimization algorithm that is based on a Lévy flight trajectory and called the Lévy flight trajectory-based whale optimization algorithm (LWOA). The LWOA makes the WOA faster and more robust and avoids premature convergence. The Lévy flight trajectory is helpful for increasing the diversity of the population against premature convergence and enhancing the capability of jumping out of local optimal optima. This method helps obtaining a better tradeoff between the exploration and exploitation of the WOA. The proposed algorithm is characterized by quick convergence and high precision, and it can effectively get rid of a local optimum. The LWOA is further compared with other well-known nature-inspired algorithms on 23 benchmarks and solving infinite impulse response model identification. The statistical results on the benchmark functions show that the LWOA can significantly outperform others on a majority of the benchmark functions, especially in solving an optimization problem that has high dimensionality. Additionally, the superior identification capability of the proposed algorithm is evident from the results obtained through the simulation study compared with other algorithms. All the results prove the superiority of the LWOA.

**INDEX TERMS** Lévy flight trajectory, whale optimization algorithm, global optimization, IIR system.

## I. INTRODUCTION

Currently, it is apparent that traditional methods find it difficult to solve engineering problems well. However, researchers have found meta-heuristic optimization techniques to be effective in providing optimal or near-optimal answers for engineering problems, and these techniques have become very popular and attracted the attention of many practitioners and researchers. Most of these algorithms originate from various natural phenomena, and two of the most popular and representative algorithms are particle swarm optimization (PSO) [1], which mimics the social behavior of flocking birds, and ant colony optimization (ACO) [2], which is inspired by the pheromones of ants. In addition to these two algorithms, some of the latest effective algorithms are the bat algorithm (BA) [3], which simulates the echolocation behavior of bats, grey wolf optimization (GWO) [4], which mimics the behavior of wolves, flower pollination algorithm (FPA) [5], which simulates flower pollination behavior, firefly algorithm (FA) [6], which mimics the behavior of fireflies as a result of their flashing characteristics, and

cuckoo search (CS) [7], which is based on the reproduction strategy of cuckoos.

The whale optimization algorithm (proposed by Mirjalili et al. [8]) is a newly proposed meta-heuristic that is inspired from the bubble-net hunting technique of humpback whales. It has been proven that the WOA can provide very competitive results, and when it is compared to other well-known metaheuristics, this algorithm has not only attracted the attention of many researchers but also been increasingly applied in many applications studies. Bhedsadiya, R. H. et al. have applied the WOA to train a multi-layer perceptron in a neural network [9]. A Kaveh has applied an enhanced WOA for the sizing optimization of a skeletal structure [10]. Haider J. Touma has used the WOA algorithm to solve the 30-bus problem of the economic dispatch problem [11]. Bhedsadiya, R. H. et al. proposed a novel adaptive WOA for global optimization [12]. D.B. Prakash and C. Lakshminarayana have used WOA algorithm to find optimal sizing and placement of capacitors for a typical radial distribution system [13]. Ibrahim Aljarah et al. have applied WOA

algorithm to optimize connection weights in neural networks [14]. Although the basic WOA has shown good performance compared with some traditional methods, it still has demerits with regard to slow convergence, low precision and easily trapped into local optimum due to a lack of population diversity. Lévy flight is a special class of random walk in which the step lengths are distributed according to a heavy power law tails. The large steps occasionally taken helps an algorithm to conduct a global search. Using the Lévy flight trajectory [7], [15] is helpful in obtaining a better trade-off between the exploration and exploitation of the algorithm, and it has merit in terms of the avoidance of local optimal optima. Therefore, this paper proposes an improved version of the WOA that is based on a Lévy flight trajectory and is called the LWOA, whose purpose is to improve the convergence rate and the precision of the basic WOA. The LWOA is tested with 23 benchmark functions and infinite impulse response (IIR) model identification. The simulation results show that the LWOA is feasible and effective and, most notably, has superior approximation capabilities in high-dimensional space.

The remainder of this paper is organized as follows: Section 2 briefly introduces the original WOA and the Lévy flight trajectory. The detailed description of the LWOA is presented in Section 3. To demonstrate the superior performance of the LWOA, a variety of experimental results and analysis that compares the LWOA with the other five well-known nature-inspired algorithms (including the original WOA) for 23 benchmark functions and infinite impulse response (IIR) system identification problem are presented, respectively, in Sections 4 and 5. Section 6 provides a discussion and analysis of the results. The last section shows the final conclusions.

## II. WHALE OPTIMIZATION ALGORITHM (WOA) AND THE LéVY FLIGHT TRAJECTORY

### A. WHALE OPTIMIZATION ALGORITHM

The Whale optimization algorithm [8] is a newly proposed meta-heuristic that is inspired from the bubble-net hunting technique of humpback whales. The WOA describes the special pursuing behavior of humpback whales, in which the whales attempt to encircle the prey (fish herds) near the surface of the water while creating bubbles that are in the shape of a circle. In the bubble-net hunting technique, the humpback whales dive approximately 12 meters down and then begin to make bubbles in a spiral shape around the prey and swim toward the surface.

### B. MATHEMATICAL MODEL DESCRIPTION/EXPLANATION

#### 1) ENCIRCLING THE PREY

Humpback whales first observe the locations of the prey (fish herds) and then encircle them. The WOA algorithm presumes that the objective prey that is closest to the perfect answer is the best candidate solution. After the best hunting agent is defined, the other whales' search operators will accordingly attempt to update their positions toward the best

hunting agent. During the optimization, to mathematically model the encircling mechanism, the following equations are proposed [8].

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

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

where  $\vec{A}$ ,  $\vec{C}$  are coefficient vectors,  $t$  denotes the current iteration,  $X^*$  is the position vector of the optimum solution acquired thus far, and  $X(t)$  is the position vector. Here,  $|\bullet|$  is the absolute value, and  $\cdot$  is the constituent-by-constituent multiplication. It must be mentioned here that  $X^*$  should be updated in each iteration if there is a better solution. The coefficient vectors  $\vec{A}$ ,  $\vec{C}$  are calculated as follows:

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

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

where the components of  $\vec{d}$  are linearly decreased from 2 to 0 over the course of the iterations (in both the exploration and exploitation stages), and  $r$  is a random vector in  $[0, 1]$ . Eq. (2) permits any search agent to update its position in the area of the current best solution and encircle the prey.

#### 2) BUBBLE-NET ATTACKING STRATEGY (EXPLOITATION PERIOD)

To express a mathematical equation for the bubble-net attacking behavior of the humpback whales, two strategies are modeled, as follows:

##### a: SHRINKING ENCIRCLING SYSTEM

This technique is achieved by decreasing linearly the value of  $\vec{d}$  from 2 to 0 via Eq. (3). Note that the fluctuation scope of  $\vec{A}$  is additionally decreased by  $\vec{d}$ . In other words, Vector  $\vec{A}$  is a random number in the range  $[-a, a]$ , where  $a$  is decreased from 2 to 0 throughout the course of the iterations. Setting  $\vec{A}$  randomly in  $[-1, 1]$ , the new position of a search agent can be characterized anywhere from the original position to the present best agent.

##### b: LOGARITHMIC SPIRAL UPDATING POSITION

First, the humpback whales search the prey and then calculate the distance from themselves to the prey. Then, the humpback whales move with a conical logarithmic spiral motion to prey on the fish herds. Each humpback whale is proposed to update its position according to the spiral flight path. This behavior is mathematically expressed as follows:

$$\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)| \quad (5)$$

$$\vec{X}'(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (6)$$

where  $\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)|$  indicates the distance from the  $i$ th whale to the prey (the best solution acquired thus far),  $l$  is a random value in  $[-1, 1]$ ,  $b$  is a constant for defining the logarithmic spiral shapes, and  $\cdot$  is a constituent-by-constituent multiplication.

It should be noted that humpback whales swim around the prey in a circle and, at the same time, move with a conical logarithmic spiral motion to prey on the fish herds. For simplicity, we assume that the positions of the humpback whales will be updated by Eq. (2) or Eq. (6), each with the possibility of 50%, which can be mathematically expressed as follows:

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

where  $p$  is a random number in the range  $[0, 1]$ . In addition to the bubble-net strategy, humpback whales have the additional interesting behavior of searching for prey randomly. The following is a mathematical model for the search behavior of the whales.

### 3) SEARCH FOR PREY (EXPLORATION PERIOD)

In fact, humpback whales scan for prey randomly when their positions are in a line with one other. In this phase, we want to focus on promising areas in the search space and force the search agent to swim far away from a reference whale. Therefore, vector  $\vec{A}$  is used for exploration to scan for prey, and its value is greater than 1 or less than  $-1$ . It is different from the exploitation period in that the position of a search agent in the exploration period will update in line with a randomly picked search agent rather than the best agent that has been discovered thus far. We use  $|\vec{A}| > 1$  to enforce exploration in the WOA algorithm to determine the global optimum and avoid local optima optimal. The mathematical model can be formulated as follows:

$$\vec{D} = |\vec{C} \cdot \vec{X}_{rand} - \vec{X}| \quad (8)$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D} \quad (9)$$

where  $\vec{X}_{rand}$  is chosen from the current generation and indicates a random position vector (a random whale). The general steps of the Whale optimization algorithm (WOA) can be summarized in the pseudo code shown in Algorithm 1 below [8].

### C. Lévy FLIGHT

The Lévy Flight trajectory was originally introduced by Lévy, and then Benoit Mandelbrot described it in detail. In general, Lévy flight is a type of random walk in which the steps are drawn from a Lévy distribution. A variety of studies have demonstrated that the behavior of flight for many animals and insects demonstrates the typical characteristics of Lévy flight [16]–[19]. From [19], we could learn that Reynolds and Frye have studied that fruits flies or *Drosophila melanogaster* search their landscape by using a set of straight flight paths punctuated by a sudden  $90^\circ$  turn(see Fig.1), which cause a Lévy-flight-style intermittent scale-free search pattern. P. Barthelemy et al. have concluded that even light is related to Lévy flight [20]. Subsequently, Lévy flight behavior has

---

### Algorithm 1 WOA Algorithm

---

```

Initialize the whales population  $X_i (i = 1, 2, \dots, n)$ 
Calculate the fitness of each search agent
 $X^* = \text{the best search agent}$ 
while ( $t < \max_i \text{tearation}$ )
  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 agent by Eq. (2)
    else if2 ( $|A| \geq 1$ )
      Select a random search agent ( $X_{rand}$ )
      Update the position of the current search agent by Eq. (9)
    end if2
    else if1 ( $p \geq 0.5$ )
      Update the position of the current search by Eq. (6)
    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  $X^*$ 

```

---

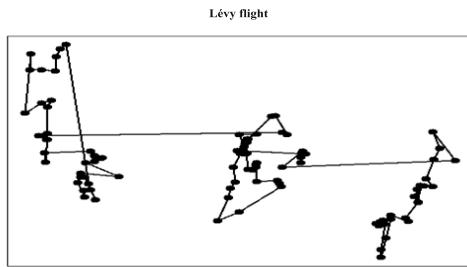
been used for optimization and optimal search, and preliminary results show that it has promising capability [17], [19], [21], [22].

### III. OUR APPROACH (LWOA)

The Whale Optimization algorithm (proposed by Mirjalili et al. [8]) can solve low-dimensional unimodal optimization problems easily. However, when handling high-dimensional and multi-modal optimization problems, we can clearly discover that the solutions obtained by the WOA are not very good. To enhance the exploration, local optima avoidance, exploitation and convergence of the WOA, this paper proposes an improved Lévy flight whale optimization (LWOA) algorithm. Lévy flight can maximize the diversification of search agents, which guarantees that the algorithm can explore the search place efficiently and accomplish local minima avoidance. This finding implies that the Lévy flight trajectory is helpful in obtaining a better trade-off between exploration and exploitation in the WOA. Therefore, the Lévy flight trajectory is used to update the humpback whales' positions after the position updating, which can be mathematically expressed as follows:

$$\vec{X}(t+1) = \vec{X}(t) + \mu \text{sign}[\text{rand} - 1/2] \oplus \text{Levy} \quad (10)$$

where  $\vec{X}(t)$  indicates the  $i$ th whale or the position vector  $\vec{X}$  at iteration  $t$ ,  $\mu$  is a random number that is consistent with a uniform distribution, the product  $\oplus$  means entrywise multiplication, and rand is a random number in the range of  $[0, 1]$ . It should be mentioned here that  $\text{sign}[\text{rand} - 1/2]$  has only



**FIGURE 1.** Simulations tracks of Lévy flights.

three values: 1, 0 and  $-1$ . Eq. (10) is essentially the stochastic equation for a random walk, and it helps the basic WOA to remove local minima and ensure that the search agent could explore the search place efficiently because its step length is much longer in the long run. The Lévy flight offers a random walk with the following distribution [7]:

$$\text{Levy} \sim u = t^{-\lambda}, 1 < \lambda \leq 3 \quad (11)$$

The Lévy flight is a special random walk in which the step-lengths have a probability distribution that is heavy-tailed. According to Fig.1, it is a figure about the simulations tracks of Lévy flights and it is typical that the steps are always small but occasionally have a large step. Mantegna's algorithm [23] is used to mimic a  $\lambda$ -stable distribution by generating random step lengths  $s$  that have the same behavior of the Lévy-flights, as follows:

$$s = \frac{\mu}{|\nu|^{1/\beta}} \quad (12)$$

where  $s$  is the step length of the Lévy flight, which is  $\text{Levy}(\lambda)$ , and  $\lambda$  in Eq. (11) obeys the formulation that  $\lambda = 1 + \beta$ , where  $\beta = 1.5$ ,  $\mu = N(0, \sigma_\mu^2)$  and  $\nu = N(0, \sigma_\nu^2)$  are both normal stochastic distributions with

$$\sigma_\mu = \left[ \frac{\Gamma(1 + \beta) \times \sin(\pi \times \beta/2)}{\Gamma(((1 + \beta)/2)) \times \beta \times 2^{(\beta-1)/2}} \right]^{1/\beta} \quad \text{and } \sigma_\nu = 1 \quad (13)$$

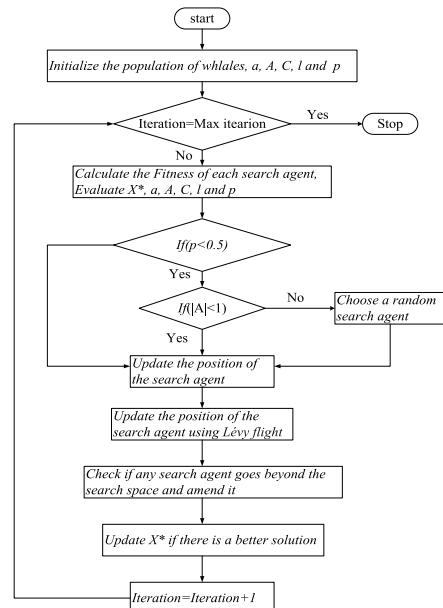
Lévy flight can significantly improve the search ability of the WOA, and then the results avoid local minima. This approach improves not only the intensification of the WOA but also its diversification. Intensification attempts to search around the present best objective solution and chooses the best objective solution, while diversification guarantees that the algorithm improves the global ability. Furthermore, it is observed to provide more significant and successful results, especially for unimodal and multimodal benchmark functions. Due to these distinguishing features, the proposed LWOA potentially outperforms the WOA. In the next section, several benchmark functions are employed to confirm the performance of the proposed algorithm in solving optimization problems. The main steps of the LWOA can be summarized in the pseudo code shown in Algorithm 2 below, and the flowchart of the LWOA is presented in Fig. 2.

## Algorithm 2 LWOA

```

Initialize the whales population  $X_i (i = 1, 2, \dots, n)$ 
Calculate the fitness of each search agent
 $X^* =$  the best search agent
while ( $t < \max_i \text{tearation}$ )
  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 agent by Eq. (2)
    else if2 ( $|A| \geq 1$ )
      Select a random search agent ( $X_{rand}$ )
      Update the position of the current search agent by Eq. (9)
    end if2
    else if1 ( $p \geq 0.5$ )
      Update the position of the current search by Eq. (6)
    end if1
  end for
  For each search agent
    Update the position of the current search agent using the Lévy flight
  end
  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 2.** The flowchart for the LWOA algorithm.

## IV. SIMULATION EXPERIMENTS

There are various benchmark functions in the literature [24]–[27] that are designed to benchmark the performance of metaheuristic methods. In this section,

**TABLE 1.** The parameter settings for six algorithms.

Algorithms	Parameter values
MFO [28]	$r$ is linearly decreased from -1 to -2 over the course of iteration, the population size is 20, and the maximum iteration number is 1000.
PSOGSA [29]	$G_0 = 1, \alpha = 20, C_1 = 0.5, C_2 = 1.5, \omega \in [0,1]$ the population size is 20, the maximum iteration number is 1000.
BA [3]	$A = 0.9, r = 0.5, f \in [-1,1]$ the population size is 20, the maximum iteration number is 1000.
ABC [30]	limit=50, the population size is 20, and the maximum iteration number is 1000.
WOA [8]	$\vec{\alpha}$ linear decrease from 2 to 0 was used, as recommended in [8], and the population size is 20. The maximum iteration number is 1000.
LWOA	$\vec{\alpha}$ linear decrease from 2 to 0 was used, as recommended in [8], and the population size is 20. The maximum iteration number is 1000.

**TABLE 2.** Unimodal benchmark functions.

Name	Benchmark function	Dim	Range	$f_{\min}$
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	50	$x_i \in [-100,100]$	0
Schwefel2.22	$f_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	50	$x_i \in [-10,10]$	0
Schwefel1.2	$f_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	50	$x_i \in [-100,100]$	0
Schwefel2.21	$f_4(x) = \max_i \{  x_i , 1 \leq i \leq D \}$	50	$x_i \in [-100,100]$	0
Resonbrock	$f_5(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	50	$x_i \in [-30,30]$	0
Step	$f_6(x) = \sum_{i=1}^n ( x_i + 0.5 )^2$	50	$x_i \in [-100,100]$	0
Quartic	$f_7(x) = \sum_{i=1}^n x_i^4 + \text{random}(0,1)$	50	$x_i \in [-1.28,1.28]$	0

23 benchmark functions are selected to exploit various characteristics to test the performance of the LWOA from different perspectives. In general, the benchmark functions can be divided into three groups: high-dimension unimodal test functions (Table II), high-dimension multimodal test functions (Table III), and fixed-dimensional multimodal test functions (Table IV). Whereas dim represents the dimension of the function, range indicates the boundary of the function's search space, and  $f_{\min}$  is the minimum value of the functions. As the names of the functions indicate, high-dimension unimodal test functions ( $f_1 \sim f_7$ ) that have only a unique global optimum can test the exploitation and convergence of an algorithm. In contrast, high-dimension multimodal test functions ( $f_8 \sim f_{13}$ ) have a variety of local optimal optima, which helps to benchmark exploration

and local optimal optima avoidance of the algorithms, and this characteristic makes them more difficult to optimize. Eventually, the same as for the high-dimension multimodal test functions, fixed-dimensional multimodal test functions ( $f_{14} \sim f_{23}$ ) also have various local optimal optima, but with some differences; the dimensionality of the fixed-dimensional multimodal test functions are low, and thus, they have a smaller number of local optimal optima. When we optimize the multimodal test functions, the algorithms are challenged to remove the local minima and are easily trapped into a nearby global optimum. Therefore, exploration and local minima can be tested by the multi-modal benchmark functions.

To verify the results of the LWOA, the other five state-of-the-art meta-heuristic algorithms are employed:

**TABLE 3.** Multimodal benchmark functions.

Name	Benchmark function	Dim	Range	$f_{\min}$
Alpine	$f_8(x) = \sum_{i=1}^n x_i \sin(x_i) + 0.1x_i$	50	$x_i \in [-10, 10]$	0
Rastrigin	$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	50	$x_i \in [-5.12, 5.12]$	0
Ackley	$f_{10}(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} - \exp \left( \frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i \right) \right) + 20 + e$	50	$x_i \in [-32, 32]$	0
Griewank	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n (x_i^2) - \prod_{i=1}^n \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$	50	$x_i \in [-600, 600]$	0
Penalized1	$f_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 \left[ 1 + 10 \sin^2(\pi y_{i+1}) \right] + (y_n - 1)^2 \right\}$ + $\sum_{i=1}^n \mu(x_i, 10, 100, 4)$	50	$x_i \in [-50, 50]$	0
	$y_i = 1 + \frac{x_i + 1}{4} \mu(x_i, a, k, m) = \begin{cases} k(x_i - a)^m > a \\ 0 - a < x_i < a \\ k(-x_i - a)^m x_i < -a \end{cases}$			
Zakharov	$f_{13}(x) = \sum_{i=1}^n x_i^2 + \left( \sum_{i=1}^n 0.5ix_i \right)^2 + \left( \sum_{i=1}^n 0.5ix_i \right)^4$	50	$x_i \in [-5, 10]$	0

moth-flame optimization algorithm (MFO) [28], a hybrid population-based algorithm with the combination of particle swarm optimization and gravitational search algorithm PSOGSA [29], bat algorithm (BA) [3], artificial bee colony algorithm (ABC) [30], and the whale optimization algorithm (WOA) [8]. To make a fair comparison, each algorithm is run on the benchmark functions 30 times, and the standard deviation of the best approximated solution in the final generation is provided. This feature shows us which algorithm behaves in a much more stable way. For all of the algorithms, the main parameters are set as shown in Table I. Such values represent the best parameter sets for these algorithms according to the original paper of MFO [28], PSOGSA [29], BA [3], ABC [30], and WOA [8].

The statistical results contain Best, Worst, Mean and Std of the objective function value, obtained by 30 independent runs for the 23 functions; these are listed in Tables V, VII and IX. All the algorithms are ranked according to the value of std. Moreover, in this paper, we also conduct Wilcoxon's non-parametric statistical test [31], [32] over these 30 runs to have a statistically meaningful conclusion. Such a statistical test

should be performed to confirm the significance of the results due to the stochastic nature of the meta-heuristics [33]. The Wilcoxon's nonparametric statistical test returns a parameter called *p-values*, which can be used to verify whether two sets of solutions are different to a statistically significant extent or not. Only if  $p < 0.5$  can we demonstrate a statistically significant superiority of the results. The *p-values* calculated in Wilcoxon's rank-sum test comparing WOA and other algorithms over all the benchmark functions are listed in Tables VI, VIII and X.

#### A. EXPERIMENTAL SETUP

All the algorithms are performed on a computer with an AMD Athlon (tm) II\*4640 processor and 4GB of RAM using MATLAB R2012a.

#### B. RESULTS OF THE ALGORITHMS ON UNIMODAL BENCHMARK FUNCTIONS

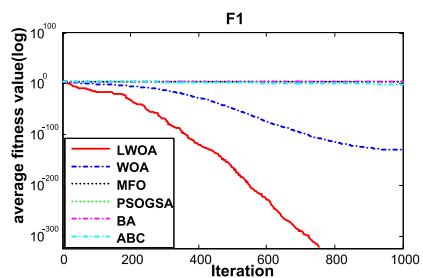
The functions  $f_1 \sim f_7$  are high-dimension unimodal because they have only one global optimum and there is no local solution for them. Therefore, they are very suitable for

**TABLE 4.** Fixed-dimension multimodal Benchmark functions.

Name	Benchmark function	Dim	Scope	$f_{\min}$
Shekel's Foxholes	$f_{14}(x) = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	2	$x_i \in [-65, 65]$	1
Kowalik	$f_{15}(x) = \sum_{i=1}^{11} \left[ a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	$x_i \in [-5, 5]$	0.0003075
Rastrigin	$f_{16}(x) = \left( x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 + 6 \right)^2 + 10 \left( 1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	$x_i \in [-5, 5]$	0.398
GoldStein Price	$f_{17}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$	2	$x_i \in [-5, 5]$	3
Shekel 1	$f_{18}(x) = - \sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$x_i \in [0, 10]$	-10.1532
Shekel 2	$f_{19}(x) = - \sum_{i=1}^7 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$x_i \in [0, 10]$	-10.4029
Shekel 3	$f_{20}(x) = - \sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$x_i \in [0, 10]$	-10.5364
Drop wave	$f_{21}(x) = - \frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$	2	$x_i \in [-5.12, 5.12]$	-1
Schaffer	$f_{22}(x) = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	2	$x_i \in [-100, 100]$	-1
Easom	$f_{23}(x) = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$	2	$x_i \in [-100, 100]$	-1

examining algorithms in terms of the convergence rate. The results of Table V show that the LWOA can provide very competitive results on unimodal functions. This algorithm outperforms other algorithms on the majority of the test cases. Table V shows that the LWOA outperforms all the others in  $f_1, f_2, f_3, f_4$ , and  $f_7$ , which are five benchmark functions. For the six functions ( $f_1, f_2, f_3, f_4, f_5$ , and  $f_7$ ) on unimodal functions, the standard deviation of LWOA is also less than that of the other algorithms. ABC has promising performance for optimizing  $f_6$ , and LWOA exhibits the third best performance on this function when compared using the standard deviation. All the statistical results show the superior performance of the LWOA in terms of solving high-dimension unimodal functions. The  $p$  – values in Table VI additionally present the superiority of the LWOA because most of the  $p$  – values are much smaller than 0.05.

The averaged convergence curves of the LWOA, MFO, PSOGSA, BA, ABC, and WOA when addressing all the unimodal functions over 30 independent runs are shown in Figs. 3-9. It must be mentioned here that all the convergence

**FIGURE 3.**  $D = 50$ , evolution curves of the fitness values for  $f_1$ .

curves in the following subsections are averaged curves. From Figs. 3-7 and 9, we can see clearly that the LWOA has the fastest convergence rate when finding the global optimum, and thus we can conclude that the LWOA is superior to the basic WOA during the process of optimization and, at the same time, outperformed all of the other four algorithms in these six high-dimensional unimodal test

**TABLE 5.** Results of the unimodal benchmark functions.

Benchmark functions	Result	Algorithm					Rank
		MFO	PSOGSA	BA	ABC	WOA	
$f_1 (D=50)$	Best	6.15000	1.80000	58700.00	0.0066133	1.93E-146	0
	Worst	20000.0	40000.0	93300.00	0.8727587	9.53E-129	0
	Mean	6656.14	7680.00	76649.85	0.1613981	3.48E-130	0
	Std	7500.00	10100.0	8760.00	0.2239980	1.73E-129	0
$f_2 (D=50)$	Best	0.71800	0.0799229	2.32E+11	0.0404188	5.06E-106	6.17E-302
	Worst	120.034	218.73276	4.75E+21	0.1881228	8.551E-95	2.83E-251
	Mean	63.6000	109.00000	1.70E+20	0.0905261	3.811E-96	9.45E-253
	Std	32.6000	74.420650	8.66E+20	0.0380328	1.593E-95	0
$f_3 (D=50)$	Best	25488.4152	17205.387	48187.346	39802.975	100980.48	0
	Worst	119231.193	65464.000	271913.27	71824.640	235329.02	0
	Mean	52969.0978	37602.818	129706.97	54842.006	154977.26	0
	Std	23032.2545	14127.625	49529.680	8250.2000	33434.339	0
$f_4 (D=50)$	Best	77.5100663	43.845901	69.714887	72.037700	13.714634	6.85E-239
	Worst	92.3926922	96.120853	83.529828	86.990567	95.806734	9.33E-188
	Mean	87.49358	76.242645	77.2798	81.701450	71.377289	3.11E-189
	Std	4.090000	15.956892	3.40000	3.320172	24.067995	0
$f_5 (D=50)$	Best	5270.0000	113.52911	262000	171.41463	47.087093	47.414227
	Worst	241459526	15988931	5384803.4	2579.3251	48.633277	48.697729
	Mean	27016318.9	21186373	2538617.0	738.48054	48.003822	48.216133
	Std	64227209.8	46205843	1240403.9	523.85824	0.4507025	0.3731767
$f_6 (D=50)$	Best	2.47997406	0.2597229	63858.780	0.0105582	0.4747958	0.6383553
	Worst	31900.000	29700.000	98100.000	1.2641610	2.5885035	3.8896131
	Mean	8330.0000	6330.9504	78500.00	0.1287937	1.3526089	1.9607244
	Std	8168.76401	7630.1711	8131.8392	0.2384516	0.5448619	0.8057334
$f_7 (D=50)$	Best	1.31221649	0.2094768	0.0695411	1.0206927	2.087E-05	7.096E-08
	Worst	97.7272991	0.7020370	0.1551359	2.3016816	0.0155463	0.0002797
	Mean	26.8406399	0.4122066	0.0944619	1.5742526	0.0026835	7.075E-05
	Std	28.2000000	0.1138668	0.0231043	0.3308606	0.0033584	7.439E-05

**TABLE 6.** Results of the *p*-value for the Wilcoxon rank-sum test on unimodal benchmark functions.

Functions	MFO vs LWOA	PSOGSA vs LWOA	BA vs LWOA	ABC vs LWOA	WOA vs LWOA
$f_1$	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
$f_2$	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
$f_3$	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
$f_4$	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
$f_5$	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.92E-02
$f_6$	6.7E-11	1.69E-09	3.02E-11	5.49E-11	3.34E-03
$f_7$	3.02E-11	3.02E-11	3.02E-11	3.02E-11	5.07E-10

functions. Fig. 8 illustrates that ABC eventually finds a better best-approximated solution than the LWOA; however, the LWOA is much more stable than ABC. Figs. 10-16 depict

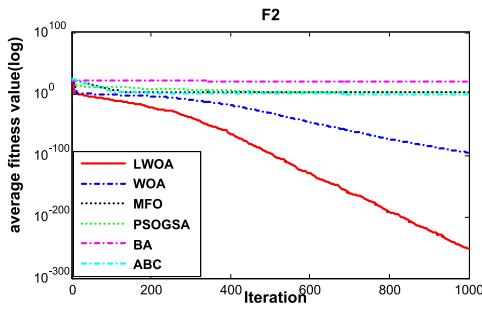
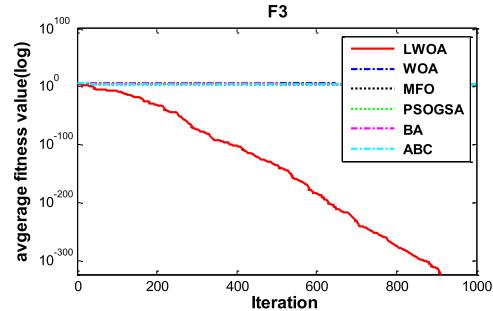
the Anova test of the global minimum of all the algorithms for the benchmark functions from  $f_1$  to  $f_7$ . From Table V and Figs. 10-16, it is obvious that the LWOA's standard

**TABLE 7.** Results of multimodal benchmark functions.

Benchmark functions	Result	Algorithm					Rank
		MFO	PSOGSA	BA	ABC	WOA	
$f_8 (D=50)$	Best	0.4957572	3.4273133	9.0027522	0.0715158	8.6E-109	9.87E-296
	Worst	32.0813744	22.389674	44.380242	0.6720755	1.82E-82	7.95E-241
	Mean	12.3003329	11.239291	20.171644	0.3005940	6.07E-84	2.65E-242
	Std	8.67722171	5.5162568	8.0592873	0.1476362	3.33E-83	0
$f_9 (D=50)$	Best	268.831115	175.11227	288.38874	11.12615	0	0
	Worst	450.00000	381.06687	466.00000	24.27988	0	0
	Mean	334.00000	279.11064	371.00000	17.00000	0	0
	Std	52.5572791	54.156122	41.240335	2.891041	0	0
$f_{10} (D=50)$	Best	18.4375531	16.061921	18.860989	0.1793727	8.88E-16	8.881E-16
	Worst	20.000000	20.00000	19.900000	1.3257102	7.99E-15	8.881E-16
	Mean	19.5586102	18.510762	19.391349	0.8037350	4.09E-15	8.881E-16
	Std	0.46043749	0.8579219	0.3195088	0.3440640	2.35E-15	0
$f_{11} (D=50)$	Best	1.08783843	1.2700000	817.0000	0.0101887	0	0
	Worst	271.309496	183.75765	1202.7643	0.5818773	0.2305737	0
	Mean	44.0597501	80.711588	1061.1506	0.1491764	0.0076857	0
	Std	65.6000000	72.756241	87.370446	0.1182633	0.0420968	0
$f_{12} (D=50)$	Best	7.72233888	13.774878	8861228	0.0003427	0.0065225	0.0034478
	Worst	512025046	51200006	48304869	0.0497181	0.3509396	0.0895898
	Mean	172000000	59700000	25300000	0.0099176	0.0407327	0.0379750
	Std	93455862.3	12902576	11358203	0.0133278	0.0660877	0.0219968
$f_{13} (D=50)$	Best	428.529327	249.31637	32.194495	918.48678	1305.1430	0
	Worst	2107.76072	988.12786	442.13725	1312.8759	1987.3408	0
	Mean	1258.84177	522.62671	78.612517	1166.1103	1632.0301	0
	Std	465.87576	210.10237	80.290840	93.132404	195.39490	0

**TABLE 8.** Results of the *p*-value Wilcoxon rank-sum test on multimodal benchmark functions.

Functions	MFO vs LWOA	PSOGSA vs LWOA	BA vs LWOA	ABC vs LWOA	WOA vs LWOA
$f_8$	3.02E-11	3.02E-11	3.02E-11	3.02E-11	1.73E-06
$f_9$	1.21E-12	1.21E-12	1.21E-12	1.21E-12	N/A
$f_{10}$	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.01E-08
$f_{11}$	1.21E-12	1.21E-12	1.21E-12	1.21E-12	3.34E-01
$f_{12}$	3.02E-11	3.02E-11	3.02E-11	2.03E-07	7.98E-02
$f_{13}$	1.21E-12	1.21E-12	1.21E-12	4.98E-11	2.71E-01

**FIGURE 4.**  $D = 50$ , evolution curves of the fitness value  $f_2$ .**FIGURE 5.**  $D = 50$ , evolution curves of the fitness value for  $f_3$ .

deviation is less than that of the other five algorithms. These experimental results show that the LWOA has an excellent exploitation ability for solving high-dimension unimodal benchmark functions.

### C. RESULTS OF THE ALGORITHMS ON HIGH-DIMENSION MULTIMODAL BENCHMARK FUNCTIONS

The statistical results of the algorithms on high-dimension multimodal benchmark functions are shown in Table VII.

**TABLE 9.** Results of fixed-dimension multimodal benchmark functions.

Benchmark functions	Result	Algorithms					Rank
		MFO	PSOGSA	BA	ABC	WOA	
$f_{14}$ ( $D=2$ )	Best	0.9980038	0.9980038	0.9980038	0.9980038	0.9980038	0.9980038
	Worst	12.670505	22.900634	23.809434	1.0261512	10.763180	12.670505
	Mean	4.4359917	6.8192764	12.687439	1.0003048	1.7534642	4.5602442
	Std	3.4900000	6.5005841	6.2782878	0.0059034	1.8831842	3.9392103
$f_{15}$ ( $D=4$ )	Best	0.0005096	0.0003074	0.0005192	0.0008085	0.0003076	0.0003129
	Worst	0.0203633	0.0225533	0.1188170	0.0025527	0.0023703	0.0007729
	Mean	0.0039700	0.0035837	0.0075723	0.0016161	0.0007381	0.0004278
	Std	0.0067092	0.0069304	0.024602	0.0004673	0.0004953	0.0001286
$f_{16}$ ( $D=2$ )	Best	0.3978873	0.3978873	0.3978873	0.3978959	0.3978873	0.3979886
	Worst	0.3978873	0.3978873	0.3978874	0.3992973	0.3978963	0.3981430
	Mean	0.3978873	0.3978873	0.3978873	0.3982247	0.3978895	0.3979272
	Std	0	0	2.15E-08	3.20E-04	2.864E-06	6.34E-05
$f_{17}$ ( $D=2$ )	Best	3	3	3	3.0038025	3	3
	Worst	3	3	84.000000	3.1018659	3.0069421	3.0036112
	Mean	3	3	15.600005	3.0393556	3.0003497	3.0003497
	Std	2.502E-15	2.161E-15	22.121299	0.0277987	0.0012937	0.0007191
$f_{18}$ ( $D=4$ )	Best	-10.1532	-10.1532	-10.15316	-10.00838	-10.15273	-10.152506
	Worst	-2.630471	-2.630471	-2.630450	-7.335544	-2.629024	-0.8927282
	Mean	-6.465878	-5.879811	-5.641223	-9.097271	-7.944815	-9.8191905
	Std	3.2040479	3.0224983	3.3740154	0.6832191	2.8033057	1.6861028
$f_{19}$ ( $D=4$ )	Best	-10.40294	-10.40294	-10.40284	-10.31932	-10.40263	-10.402239
	Worst	-1.837593	-2.751933	-2.751913	-6.265706	-2.611919	-10.156297
	Mean	-6.880537	-5.623942	-4.710459	-9.059769	-8.633117	-10.370955
	Std	3.6527812	3.2661134	2.7564982	0.9634577	2.7374898	0.0453235
$f_{20}$ ( $D=4$ )	Best	-10.53641	-10.53641	-10.53636	-10.34275	-10.53489	-10.535140
	Worst	-1.859480	-1.859480	-1.676550	-6.297999	-1.675821	-1.6460791
	Mean	-6.830401	-4.602916	-4.675314	-8.824212	-6.928500	-10.208377
	Std	3.6343660	3.1362511	3.1017606	1.1373033	3.3262606	1.6174411
$f_{21}$ ( $D=2$ )	Best	-1	-1	-0.936245	-0.999895	-1	-1
	Worst	-0.936245	-0.936245	-0.369127	-0.979408	-0.936245	-1
	Mean	-0.968122	-0.980873	-0.622707	-0.993835	-0.976623	-1
	Std	0.0324222	0.0297155	0.1815385	0.0058563	0.0312482	0
$f_{22}$ ( $D=2$ )	Best	-1	-1	-0.990284	-0.921810	-0.999929	-1
	Worst	-0.962775	-0.873009	-0.520430	-0.988677	-0.990284	-1
	Mean	-0.990338	-0.982707	-0.648434	-0.990734	-0.995465	-1
	Std	0.0059879	0.0227867	0.1155438	0.0022144	0.0049300	0
$f_{23}$ ( $D=2$ )	Best	-1	-1	-1	-0.999997	-1	-1
	Worst	-1	-1	-0.0000811	-0.999671	-0.999995	-0.9999425
	Mean	-1	-1	-0.633363	-0.999916	-0.999999	-0.9999923
	Std	0	0	0.4900927	8.86E-05	1.08E-06	1.201E-05
Count of Std	2	2	0	4	2	14	

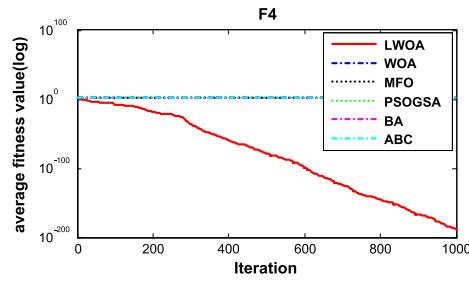
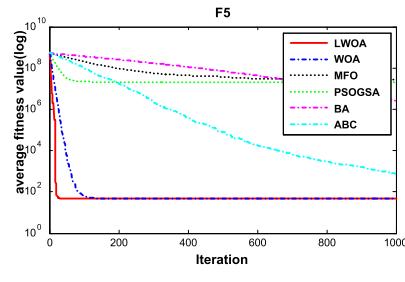
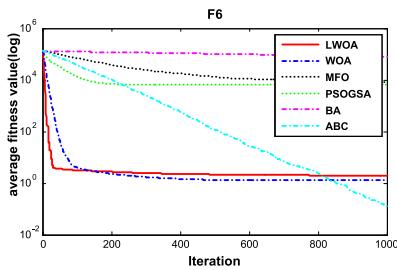
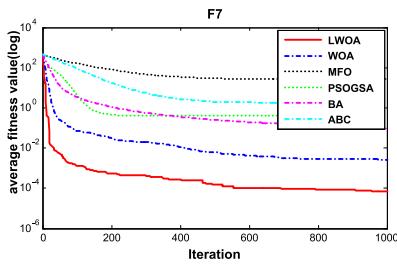
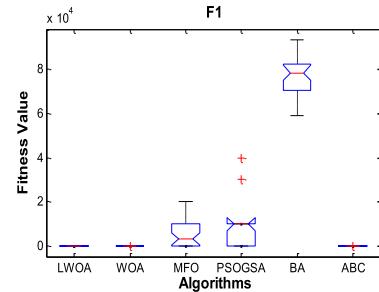
**FIGURE 6.**  $D = 50$ , evolution curves of the fitness value for  $f_4$ .**FIGURE 7.**  $D = 50$ , evolution curves of the fitness value for  $f_5$ .

Table VII shows that the LWOA outperforms all the others on five of the benchmark functions. For  $f_8, f_9, f_{10}, f_{11}$  and  $f_{13}$ , it can be seen that the best fitness value, worst fitness value, mean fitness value and standard deviation produced by

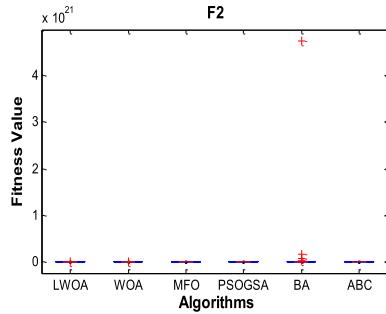
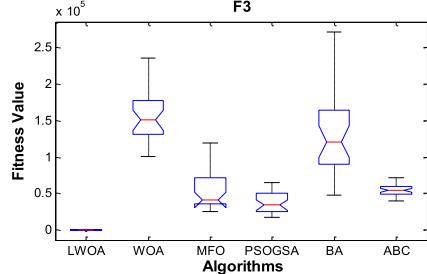
the LWOA are much better than those of other algorithms. Moreover, for  $f_9, f_{11}$  and  $f_{13}$ , the LWOA can achieve the theoretical global optimum of these three functions. The remaining results, which belong to  $f_{12}$ , show that the ABC algorithm can perform better than the LWOA, while LWOA

**TABLE 10.** Results of the  $p$ -value Wilcoxon rank-sum test on multimodal benchmark functions.

Functions	MFO vs LWOA	PSOGSA vs LWOA	BA vs LWOA	ABC vs LWOA	WOA vs LWOA
$f_{14}$	0.283155	0.482084	8.29E-06	6.77E-05	1.75E-05
$f_{15}$	1.46E-10	5.09E-08	3.82E-10	3.02E-11	6.10E-03
$f_{16}$	1.21E-12	1.21E-12	3.02E-11	2.02E-08	1.73E-07
$f_{17}$	2.81E-11	2.68E-11	1.52E-03	3.02E-11	1.50E-02
$f_{18}$	2.89E-01	1.89E-02	9.33E-02	5.57E-10	9.82E-01
$f_{19}$	1.00E+00	7.85E-03	9.51E-06	4.5E-11	1.00E+00
$f_{20}$	8.41E-01	2.79E-04	2.84E-04	5.57E-10	1.50E-02
$f_{21}$	1.1E-05	1.31E-03	1.21E-12	1.21E-12	3.12E-04
$f_{22}$	5.42E-12	1.2E-13	1.21E-12	1.21E-12	2.85E-05
$f_{23}$	1.21E-12	1.21E-12	1.12E-01	9.83E-08	1.73E-06

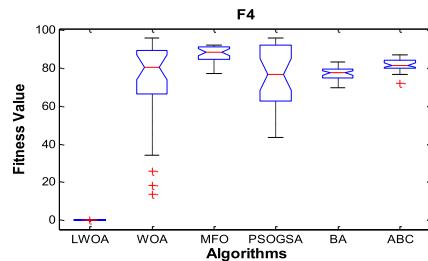
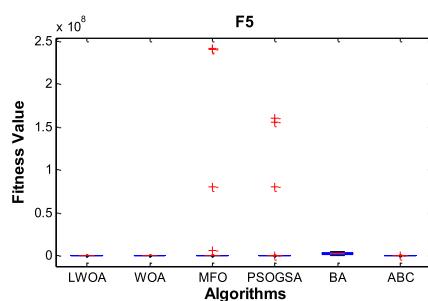
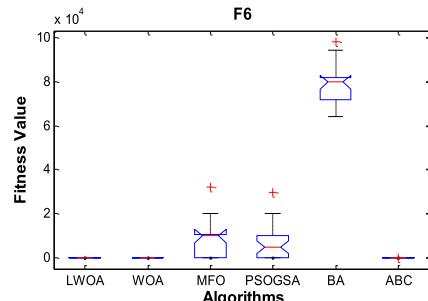
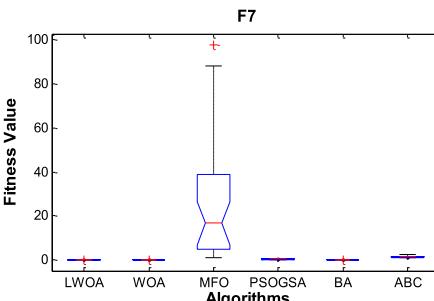
**FIGURE 8.**  $D = 50$ , evolution curves of the fitness value for  $f_6$ .**FIGURE 9.**  $D = 50$ , evolution curves of the fitness value for  $f_7$ .**FIGURE 10.**  $D = 50$ , ANOVA test of the global minimum for  $f_8$ .

is the second most effective, performing better than the other algorithms. The  $p$ -values reported in Table VIII show that the results of the LWOA in  $f_{11}, f_{12}$  and  $f_{13}$  are not significantly better than those for the basic WOA (5% significance

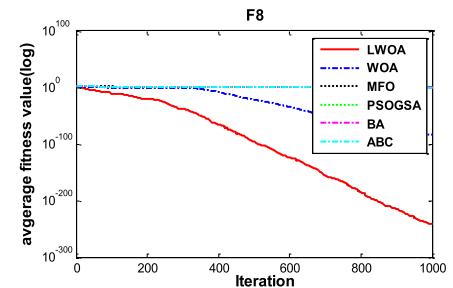
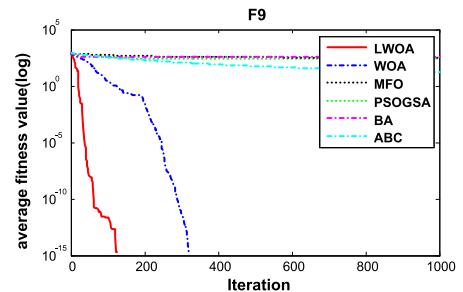
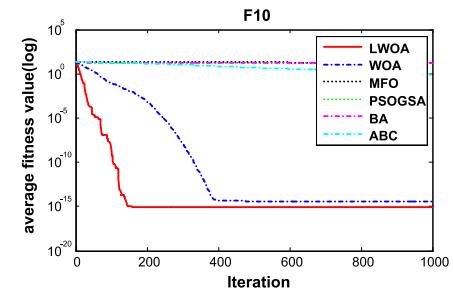
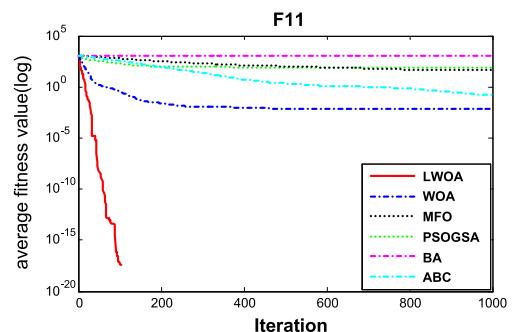
**FIGURE 11.**  $D = 50$ , ANOVA test of the global minimum for  $f_9$ .**FIGURE 12.**  $D = 50$ , ANOVA test of the global minimum for  $f_3$ .

level); however, in  $f_8, f_9$  and  $f_{10}$ , the LWOA is significantly better than the WOA. When compared with remaining algorithms, all of the  $p$ -values are much less than 0.05, which indicates that the LWOA shows significantly better results compared to the other algorithms. Because the multimodal functions have a variety of local optimal optima, these results show that the LWOA can explore the search space extensively and find promising regions of the search place, and it is evident that the LWOA has a strong ability to accomplish local minimum avoidance.

Figs. 17-22 illustrate the averaged convergence curves of all the algorithms that address all the high-dimension multimodal test functions over 30 independent runs. Fig. 21 depicts that ABC can finally find a better solution than the

FIGURE 13.  $D = 50$ , anova test of the global minimum for  $f_4$ .FIGURE 14.  $D = 50$ , anova test of the global minimum for  $f_5$ .FIGURE 15.  $D = 50$ , anova test of the global minimum for  $f_6$ .FIGURE 16.  $D = 50$ , anova test of the global minimum for  $f_7$ .

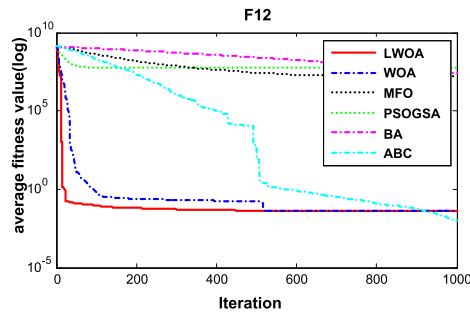
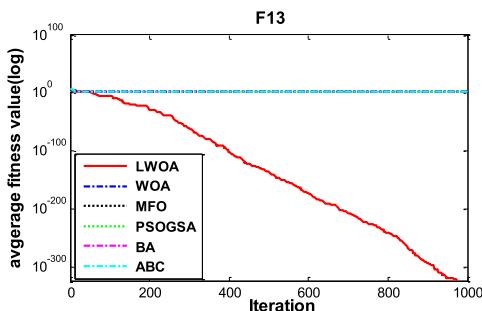
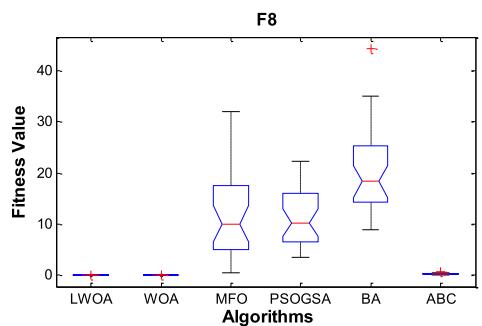
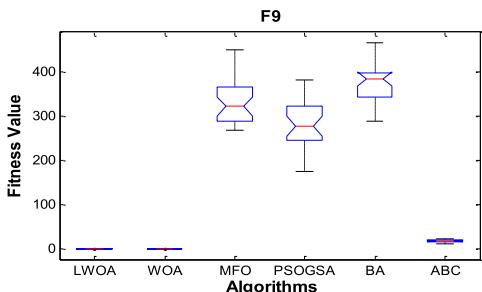
LWOA gain; however, the LWOA has the fastest convergence rate at the beginning and becomes much more stable than ABC. As can be seen from the remaining curves in the figures, the LWOA has the fastest convergence rate and reaches the approximate best solutions. Figs. 23-28 show the anova tests of the global minima for all the algorithms in  $f_8 \sim f_{13}$ . According to Table VII and Figs. 23-28, the standard deviations of the LWOA are smaller than those of the other algorithms. From all these findings, we can draw

FIGURE 17.  $D = 50$ , evolution curves of the fitness value for  $f_8$ .FIGURE 18.  $D = 50$ , evolution curves of the fitness value for  $f_9$ .FIGURE 19.  $D = 50$ , evolution curves of the fitness value for  $f_{10}$ .FIGURE 20.  $D = 50$ , evolution curves of the fitness value for  $f_{11}$ .

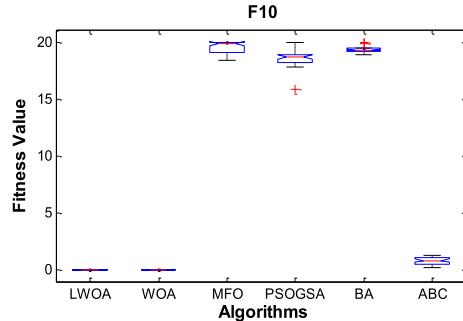
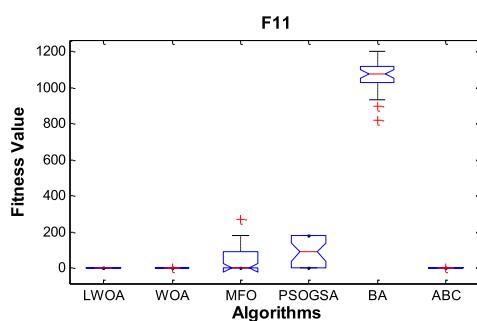
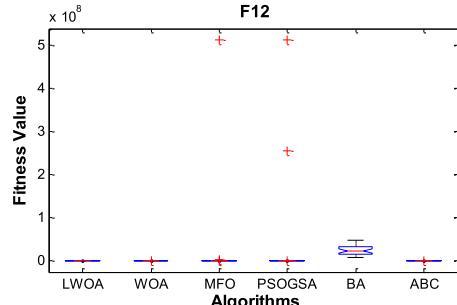
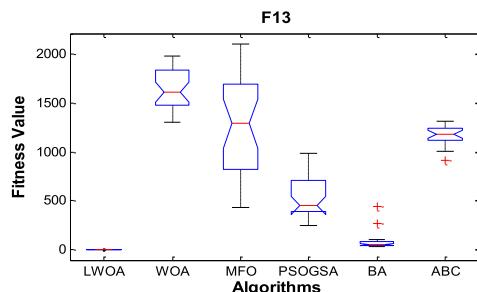
the conclusion that the LWOA is capable of avoiding local minima with a good convergence rate for high-dimension multimodal benchmark functions, and the LWOA has merit among the other algorithms, including the original WOA.

#### D. RESULTS OF THE ALGORITHMS ON FIXED-DIMENSION MULTIMODAL BENCHMARK FUNCTIONS

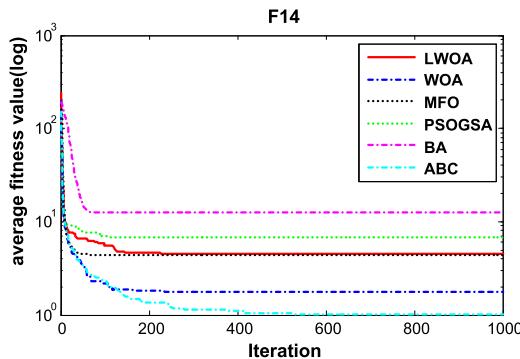
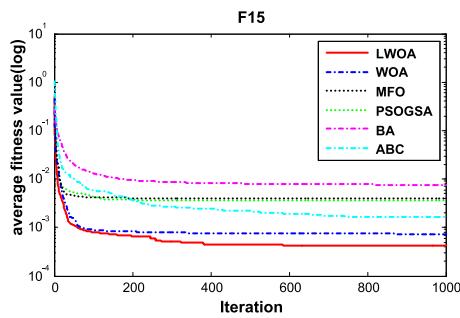
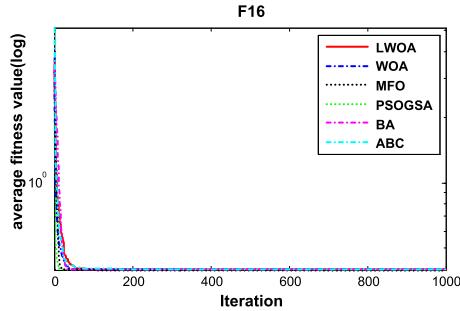
Similar to high-dimension multimodal benchmark functions, fixed-dimension multimodal benchmark functions also have

FIGURE 21.  $D = 50$ , evolution curves of the fitness value for  $f_{12}$ .FIGURE 22.  $D = 50$ , evolution curves of the fitness value for  $f_{13}$ .FIGURE 23.  $D = 50$ , ANOVA test of the global minimum for  $f_8$ .FIGURE 24.  $D = 50$ , ANOVA test of the global minimum for  $f_9$ .

more than one local minimum. The main difference between the two is that while the dimensions of the fixed-dimension multimodal benchmark functions are lower than the high-dimension multimodal functions, the number of local minima in the fixed-dimension multimodal benchmark functions is less than that for the high-dimension multimodal benchmark functions. Table IX includes the results on the

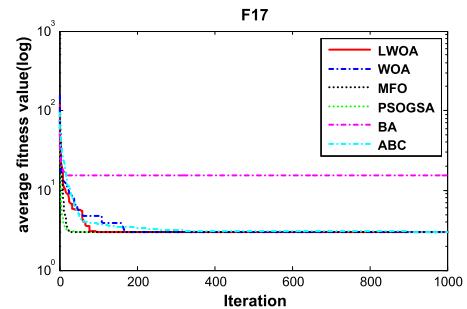
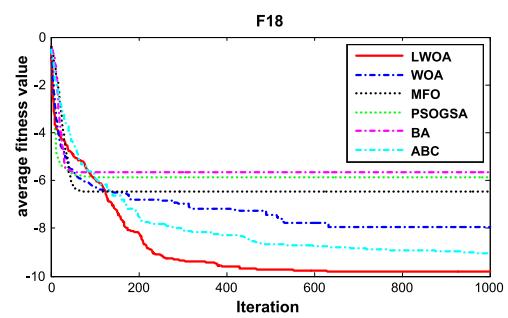
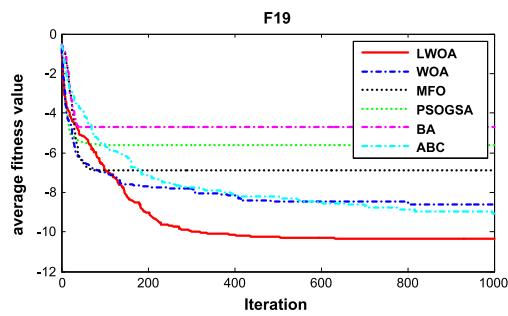
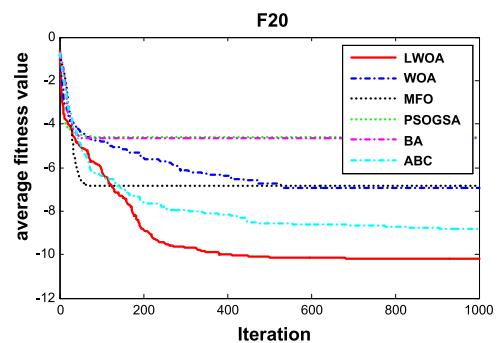
FIGURE 25.  $D = 50$ , ANOVA test of the global minimum for  $f_{10}$ .FIGURE 26.  $D = 50$ , ANOVA test of the global minimum  $f_{11}$ .FIGURE 27.  $D = 50$ , ANOVA test of the global minimum for  $f_{12}$ .FIGURE 28.  $D = 50$ , ANOVA test of the global minimum for  $f_{13}$ .

fixed-dimension multimodal benchmark functions. The results of the best values presented in Table VIII highlight the LWOA, and we can see clearly that the LWOA performs better than the others in six of the fixed-dimension multimodal benchmark functions. The LWOA failed to show

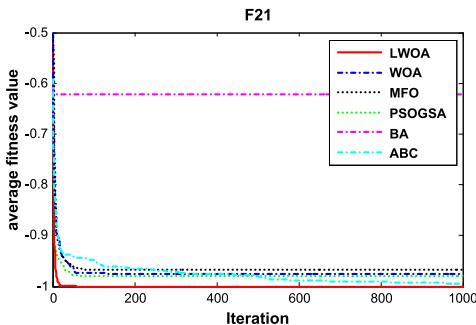
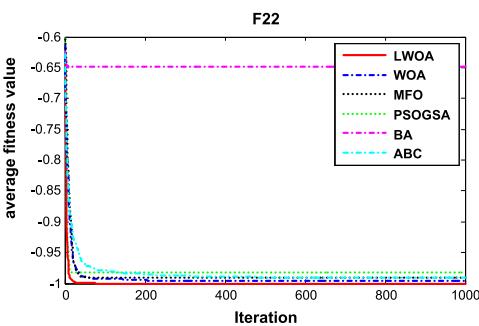
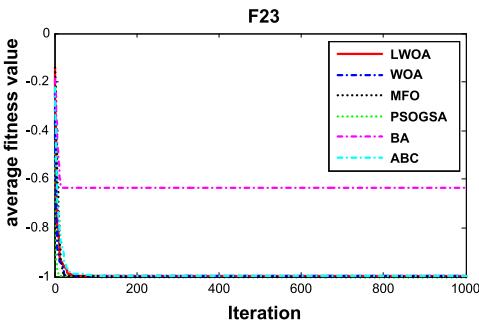
FIGURE 29.  $D = 50$ , evolution curves of the fitness value for  $f_{14}$ .FIGURE 30.  $D = 50$ , evolution curves of the fitness value for  $f_{15}$ .FIGURE 31.  $D = 50$ , evolution curves of the fitness value for  $f_{16}$ .

the best results on  $f_{15}, f_{18}, f_{19}$  and  $f_{20}$ , while at the same time, the LWOA can find the theoretic global optimum of  $f_{17}, f_{21}, f_{22}, f_{23}$  for these four test functions. In addition, for  $f_{15}, f_{19}, f_{21}$  and  $f_{22}$ , the standard deviations of the LWOA are smaller than those of the other algorithms. In fact, the LWOA is the most efficient or the second-best algorithm in the majority of test functions. The  $p$ -values in Table X also support the better results of the LWOA statistically. Inspecting the results of Table X, the LWOA is significantly better in six out of ten of the fixed-dimension benchmark functions. Thus, it can be claimed that the results of the LWOA in these benchmark functions are better than those for the other algorithms.

Figs. 29–38 depict the convergence curves of all the algorithms on fixed-dimension multimodal benchmark functions. As can be observed in Fig. 29, the ABC algorithm and the WOA are superior to the LWOA and the other algorithms,

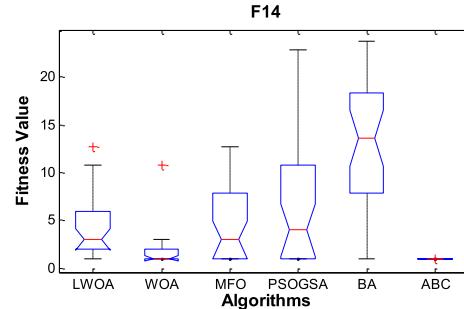
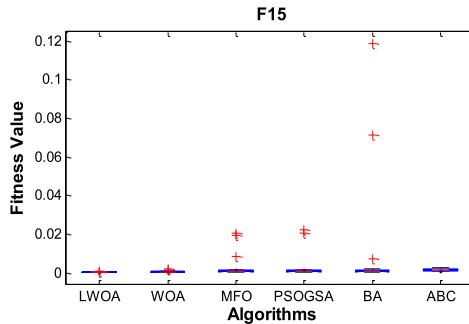
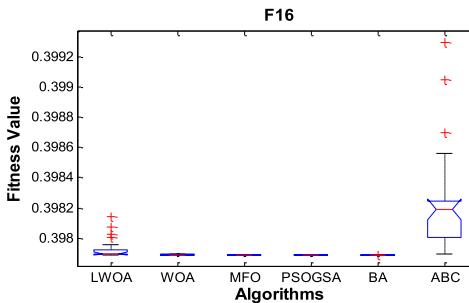
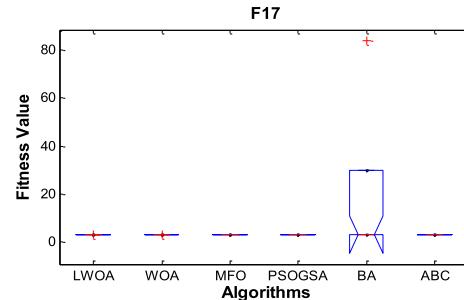
FIGURE 32.  $D = 50$ , evolution curves of the fitness value for  $f_{17}$ .FIGURE 33.  $D = 50$ , evolution curves of the fitness value for  $f_{18}$ .FIGURE 34.  $D = 50$ , evolution curves of the fitness value for  $f_{19}$ .FIGURE 35.  $D = 50$ , evolution curves of the fitness value for  $f_{20}$ .

while the LWOA also performs well compared with the other algorithms. For Fig. 31, the PSOGSA and MFO are slightly superior to the LWOA and the other algorithms, while the LWOA is the third best and obtains the same final optimum

FIGURE 36.  $D = 50$ , evolution curves of the fitness value for  $f_{21}$ .FIGURE 37.  $D = 50$ , evolution curves of the fitness value for  $f_{22}$ .FIGURE 38.  $D = 50$ , evolution curves of the fitness value for  $f_{23}$ .

with the PSOGSA and MFO algorithm. For the remaining figures, it is obvious that the LWOA outperforms all the other algorithms during the progress of the optimization. Statistically speaking, the figures concerning convergence curves show the superiority and high performance of the LWOA. Figs. 39-48 provide the anova test of the global minimum for all the algorithms in  $f_{14} \sim f_{23}$ . According to Table IX and Figs. 39-48, we can conclude that the standard deviations of the LWOA are smaller than those for the other algorithms for the majority of the test cases. These results show that the LWOA is potentially able to solve fixed-dimension multimodal test functions, and in addition, the results provide strong evidence for the superior ability of the LWOA.

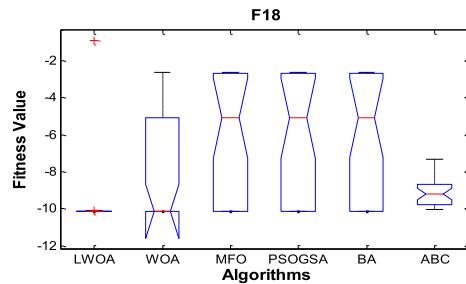
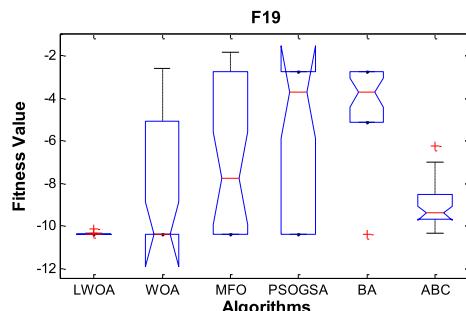
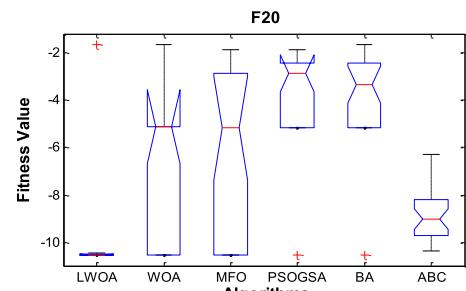
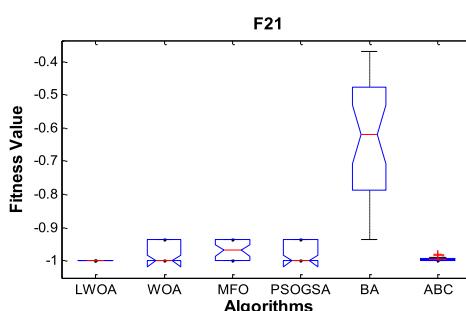
In summary, the results of this section experimentally prove that the LWOA shows very competitive results and outperforms other well-known algorithms on the three types

FIGURE 39.  $D = 50$ , anova test of the global minimum for  $f_{14}$ .FIGURE 40.  $D = 50$ , anova test of the global minimum for  $f_{15}$ .FIGURE 41.  $D = 50$ , anova test of the global minimum for  $f_{16}$ .FIGURE 42.  $D = 50$ , anova test of the global minimum for  $f_{17}$ .

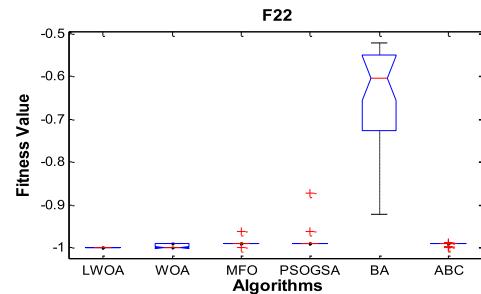
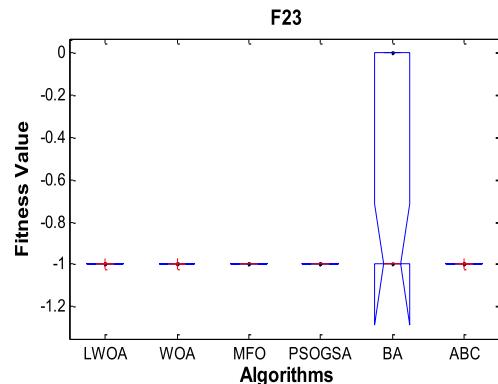
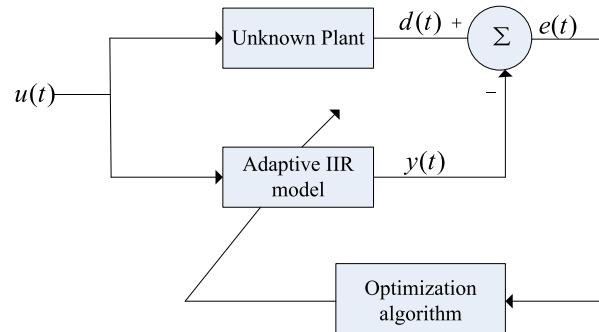
of benchmark functions, and we state that the LWOA has merit among the other five algorithms. The next section inspects the performance of the LWOA in solving the IIR model identification problem.

## V. IIR MODEL IDENTIFICATION

In this section, infinite systems identification is solved to further investigate the performance of proposed LWOA.

**FIGURE 43.**  $D = 50$ ,  $D = 50$ , anova test of the global minimum for  $f_{18}$ .**FIGURE 44.**  $D = 50$ , anova test of the global minimum for  $f_{19}$ .**FIGURE 45.**  $D = 50$ ,  $D = 50$ , anova test of the global minimum for  $f_{20}$ .**FIGURE 46.**  $D = 50$ , anova test of the global minimum for  $f_{21}$ .

Infinite system identification is an effective method to acquire the mathematics model of an unknown system by analyzing its input and output values. System identification based on infinite impulse response (IIR) models are preferably utilized in real world application than equivalent FIR (finite impulse response) since the former model physics plants more accurately. In addition, they are typically capable of meeting performance specifications using fewer filter

**FIGURE 47.**  $D = 50$ , anova test of the global minimum for  $f_{22}$ .**FIGURE 48.**  $D = 50$ , anova test of the global minimum for  $f_{23}$ .**FIGURE 49.** Block diagram of IIR system identification.

coefficients [34]. In a system identification configuration which is shown in Fig. 49, the optimization algorithm searches for the adaptive filter parameters until its input/output relationship matches closely to that of the unknown plant. The fitness function, which is used in the article could be defined as the following [35].

The transfer function of the IIR system can be written in the following general form:

$$\frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}} \quad (14)$$

where  $m$  and  $n$  are the number of numerator and denominator coefficients of the transfer function, respectively.

$a_i$  ( $i \in [1, \dots, n]$ ) and  $b_j$  ( $j \in [1, \dots, m]$ ) are filter coefficients that define its poles and zeros respectively. The Eq.(14)

can also be written in another general form as follows:

$$y(t) = \sum_{i=1}^n a_i \cdot y(t-i) + \sum_{j=0}^m b_j \cdot x(t-j) \quad (15)$$

where  $y(t)$  denotes the  $t$ th output of the system.  $u(t)$  denotes the  $t$ th input of the system (see Fig. 49). Consequently, the IIR system can be model by the series of unknown parameters which represents by  $\theta = \{a_1, \dots, a_n, b_0, \dots, b_m\}$ . Considering that the number of unknown parameters of  $\theta$  is  $(n+m+1)$ , the search space of IIR feasible values for  $\theta$  is  $\Re^{(n+m+1)}$ .

From the block of diagram of Fig. 48, the output of the unknown plant is  $d(t)$  while  $y(t)$  is the output of the IIR filter. The error function between the output of the adaptive filter and the output of the plant is  $e(t) = d(t) - y(t)$ . Hence, the design of this filter can be considered as a minimization problem of the function  $f(\theta)$  defined as follows:

$$f(\theta) = \frac{1}{w} \sum_{t=1}^w (d(t) - y(t))^2 \quad (16)$$

where  $w$  denotes the amount of the samples used for the calculation of the function  $f(\theta)$ . The main objective of the IIR system is to minimize the function  $f(\theta)$  by adjusting  $\theta$ . The optimal  $\theta^*$  or solution will be acquired as soon as the error function  $f(\theta)$  acquire its minimum value which could be stated as follows:

$$\theta^* = \arg \min(f(\theta)), \quad \theta \in S \quad (17)$$

#### The Results of the IIR Model Identification

The result is reported considering a superior-order plant through a high-order IIR model. The transfer functions of both

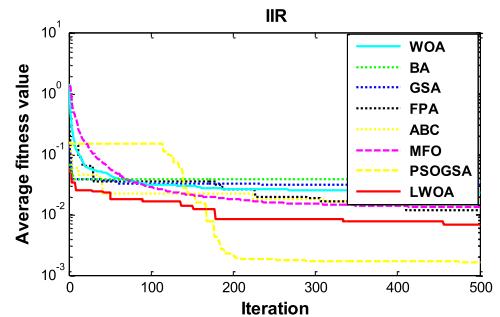
the unknown plant  $H_p$  and the IIR model  $H_M$  are defined as follows [33]:

$$H_p(z^{-1}) = \frac{1 - 0.4z^{-2} - 0.65z^{-4} + 0.26z^{-6}}{1 - 0.77z^{-2} - 0.8498z^{-4} + 0.6486z^{-6}}$$

$$H_M(z^{-1}) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4}}{1 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + a_4z^{-4}} \quad (18)$$

From Eq. (18), the plant is a sixth-order system and the IIR model a fourth-order system, the error surface  $f(\theta)$  is not unimodal but multimodal. The system input has used a white sequence with 100 samples in this case.

The results of the LWOA to solve IIR system identification and compare to others algorithms are reported in Table XI and Table XII. In this experiment, to make a fair comparison, each algorithm is run on this case 30 times. Table XI presents the best parameter values (ABP) and Table XII presents both the average  $f(\theta)$  value (AVE) and the standard deviation (STD). Fig. 50 depicts the average optimization convergence curves of the LWOA and other algorithms to in optimizing IIR system identification problem.



**FIGURE 50.** Comparison between LWOA and the other algorithms for IIR system identification.

According to the Table XII, it is obvious that the LWOA provides the best average results and the best standard deviation compare with FPA, GSA, BA, ABC, MFO, PSOGSA and WOA in terms of IIR system identification. The results show that LWOA provides better precision (AVE value) and robustness (STD). As can be observed in Fig. 50 that the LWOA has very high convergence speed in comparison to the other 6 algorithms except for the PSOGSA, though PSOGSA performs better than the LWOA in the convergence rate, however, the average value and standard deviation in Table XII shows LWOA is much more stable than PSOGSA. Therefore, we could draw a conclusion that LWOA could solve the IIR system identification very well.

To summarize, the LWOA algorithm shows desirable performance for this IIR system identification problem. By combining suitable IIR model, it will play an important role in IIR system identification. Thus in general the LWOA is a

potential candidate for identification of IIR plant compared to other evolutionary computing based approaches.

#### VI. DISCUSSION AND ANALYSIS OF THE RESULTS

Statistically speaking, the LWOA provides superior local avoidance and an optimum or near optimum in most of the datasets. In the majority of situations, the performance of the LWOA is superior to or at least highly competitive with the standard WOA, while given the same amount of time, the LWOA outperforms the other algorithms. There reasons that the LWOA performs excellently and efficiently are described next. First of all, using the Lévy flight trajectory helps to significantly improve the diversification of the algorithm to ensure its global search ability and to avoid local minima. In LWOA, the Lévy flight was introduced in to the basic WOA, that is, when each whale update the position by the original equations from the basic WOA, the objective function is not directly calculated, but it further updates individuals positions via Lévy flight as Eq. (10) and then calculated the value of objective function. Lévy flight strategy is an efficient way to carry out the exploration step and it could help search more efficiently. This strategy makes the LWOA has a better local search ability which can help the LWOA perform faster and more robustly than the WOA. In addition, according to the update mechanism of the humpback whales, whale search operators accordingly attempt to update their position towards the best hunting agent; this

**TABLE 11.** Results of the best parameter values (ABP).

Algorithms	ABP								
	a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>	a <sub>4</sub>	b <sub>0</sub>	b <sub>1</sub>	b <sub>2</sub>	b <sub>3</sub>	b <sub>4</sub>
FPA	0.068473	-0.24828	-0.0679	-0.63647	0.833805	0.082938	0.378238	0.06502	-0.52026
GSA	-0.24287	0.22186	0.037434	0.075601	0.411713	0.008314	0.136863	-0.07522	0.470426
BA	-0.09373	0.738731	-0.1249	0.293633	0.806225	-0.30992	0.825842	-0.60627	0.571276
ABC	0.547192	0.169075	0.825132	0.274598	-1.13193	0.732465	0.919517	-0.88647	0.05727
MFO	0.00990	-0.05390	-0.01050	-0.81220	0.998600	0.007200	0.295900	0.002400	-0.36060
PSOGSA	-0.1537	-0.10860	0.11140	-0.75200	0.866900	-0.17060	0.259700	0.028800	-0.20670
WOA	0.014386	-0.90321	0.031866	0.116009	0.880918	-0.06604	-0.57527	0.10358	0.287061
LWOA	0.135365	0.007952	0.04691	-0.49333	0.819424	0.211917	0.290214	0.129851	0.071382

**TABLE 12.** The average f (θ) value (ave) and the standard deviation (std).

Algorithms	AVE	STD
FPA	0.027896	9.20E-03
GSA	0.057665	1.65E-02
BA	0.268377	1.74E-01
ABC	0.046431	1.92E-02
MFO	0.032100	1.35E-02
PSOGSA	0.025100	2.58E-02
WOA	0.030306	2.09E-02
LWOA	0.021095	9.06E-03

mechanism promotes exploration of the search place and results in finding diverse search spaces .Third, due to the parameter (when  $|\vec{A}| > 1$ ), almost half of the iterations are used for exploration of the search space while in the same time the rest iterations are devoted to exploitation ( $|\vec{A}| < 1$ ) which helps to conduct a local search and in the same time as performing a global search. This mechanism is very helpful for the LWOA algorithm shows high local optimal optima avoidance and it's the main reason of the superiority of the LWOA algorithm. Finally, the advantage of the LWOA includes performing much more simply and easily and only having a few parameters to use.

Another finding in the results is the poor performance of the ABC, BA, MFO and PSOGSA. ABC, BA and MFO these three algorithms belong to the class of swarmed-based algorithms while PSOGSA is a hybrid algorithm combine with PSO and GSA these two algorithms. In contrary to evolutionary algorithms, there is no mechanism for significant abrupt movements in the search space and this is likely to be the reasons for the poor performance of ABC, BA, MFO and PSOGSA. The other reason is these four algorithms do not have operators to devote specific iteration to exploration or exploitation. In another words, these four algorithms use one formula to update the position of search agents, which may be the reason of increasing the likeness of stagnation in local minima. Although the LWOA algorithm is also a swarmed-based algorithm, its mechanisms described in the preceding are the reasons why it is advantageous in benchmark functions. From Section 4, it is clear that the LWOA performs better than the other algorithms on most of the benchmark functions because the Lévy flight trajectory provides the ability to smoothly balance exploration and exploitation.

The findings in Section 5 prove that the LWOA is very effective in solving IIR system identification system because the results on this problem show that the LWOA outperforms

the other methods. The main reason that the LWOA can perform well in this type of problem is that this algorithm is equipped with adaptive parameters to smoothly balance exploration and exploitation. Half of the iteration is devoted to exploration and the rest to exploitation. In addition, Lévy flight trajectory can increase the diversity of the population and make the algorithm jump out of local optimum more effectively. Therefore, combine with these two mechanism can let LWOA performs so well in the IIR system identification.

The comprehensive study conducted here showed that the LWOA has a stronger ability to find a global optimum and is more stable than other algorithms while at the same time efficiently solves the IIR system identification. All the statistical results prove the superiority of the LWOA.

## VII. CONCLUSION

This paper proposed an improved version of the WOA called the LWOA for optimization problems. To evaluate the performance of the LWOA, 23 benchmark functions were employed, and then we compared the LWOA with five other state-of-the-art meta-heuristic algorithms. Moreover, we use Wilcoxon's rank-sum nonparametric statistical test to judge whether the results of the LWOA differ from the best results of the other algorithms to a statistically significant extent. The results shown in Section 4 prove that the LWOA outperforms all the other algorithms in a majority of the test cases, and these results also demonstrate that the LWOA is a feasible and quite effective approach to solving global optimization problems. Furthermore, the results show the superior approximation capabilities of the LWOA in high-dimensional space.

We also applied our proposed approach of the LWOA to solve IIR system identification problem, as shown in Section 5.

Comparison of simulation results with other meta-heuristic clearly exhibits superior identification performance of the LWOA.

Our future work will focus on the following two issues: first, we would like to apply our proposed approach of the LWOA to solving more practical engineering optimization problems. Second, we would like to develop new meta-hybrid versions of the WOA.

## REFERENCES

- [1] J. Kennedy, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, Australia, vol. 4. Mar. 1995, pp. 1942–1948.

- [2] K. Socha and M. Dorigo, "Ant colony optimization for continuous domains," *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1155–1173, Mar. 2008.
- [3] X. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization*, C. Cruz, J. R. González, D. A. Pelta, N. Krasnogor, and G. Terrazas, Eds. Berlin, Germany: Springer, 2010, pp. 65–74.
- [4] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [5] X. S. Yang, "Flower pollination algorithm for global optimization," in *Unconventional Computation and Natural Computation* (Computer Science). New York, NY, USA: Springer, 2012, pp. 240–249.
- [6] X.-S. Yang, "Multi-objective firefly algorithm for continuous optimization," *Eng. Comput.*, vol. 29, no. 2, pp. 175–184, Mar. 2013.
- [7] X. S. Yang and D. Suash, "Cuckoo Search via Levy flights," in *World Congress Nature Biologically Inspired Computer (NaBIC)*. New York, NY, USA: IEEE Publication, 2009, pp. 210–214.
- [8] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
- [9] R. Bhesadiya, P. Jangir, N. Jangir, I. N. Trivedi, and D. Ladumor, "Training multi-layer perceptron in neural network using whale optimization algorithm," *Indian J. Sci. Technol.*, vol. 9, no. 19, pp. 28–36, May 2016.
- [10] A. Kaveh and M. I. Ghazaan, "Enhanced whale optimization algorithm for sizing optimization of skeletal structures," *Mech. Based Design Struct. Mach.*, vol. 1, pp. 1–18, Jul. 2016, doi: 10.1080/15397734.2016.1213639.
- [11] H. J. Touma, "Study of the economic dispatch problem on IEEE 30-bus system using whale optimization algorithm," *Int. J. Eng. Technol. Sci.*, vol. 5, no. 1, p. 1, Jun. 2016.
- [12] 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.
- [13] D. B. Prakash and C. Lakshminarayana, "Optimal siting of capacitors in radial distribution network using whale optimization algorithm," *Alexandria Eng. J.*, Nov. 2016.
- [14] I. Aljarah, H. Faris, and S. Mirjalili, "Optimizing connection weights in neural networks using the whale optimization algorithm," in *Proc. Soft Comput.*, 2016, pp. 1–15.
- [15] A. F. Kamruzaman, A. M. Zain, S. M. Yusuf, and A. Udin, "Levy flight algorithm for optimization problems—A literature review," *Appl. Mech. Mater.*, vol. 421, pp. 496–501, Sep. 2013.
- [16] C. T. Brown, L. S. Liebovitch, and R. Glendon, "Lévy flights in Dobe Ju/'hoansi Foraging patterns," *Human Ecol.*, vol. 35, no. 1, pp. 129–138, Dec. 2007.
- [17] I. Pavlyukevich, "Lévy flights, non-local search and simulated annealing," *J. Comput. Phys.*, vol. 226, no. 2, pp. 1830–1844, Oct. 2007.
- [18] I. Pavlyukevich, "Cooling down Lévy flights," *J. Phys. A, Math. Theor.*, vol. 40, no. 41, p. 12299, Sep. 2007.
- [19] A. M. Reynolds and M. A. Frye, "Free-flight odor tracking in *Drosophila* is consistent with an optimal intermittent scale-free search," *PLoS One*, vol. 2, no. 4, p. e354, Apr. 2007.
- [20] P. Barthelemy, J. Bertolotti, and D. S. Wiersma, "A Levy flight for light," *Nature*, vol. 453, no. 7194, pp. 495–498, May 2008.
- [21] M. F. Shlesinger, G. M. Zaslavsky, and U. Frisch, *Lévy Flights and Related Topics in Physics*. Berlin, Germany: Springer, 1995.
- [22] M. F. Shlesinger, "Mathematical physics: Search research," *Nature*, vol. 443, no. 7109, pp. 281–282, Sep. 2006.
- [23] R. N. Mantegna, "Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 49, no. 5, pp. 4677–4683, May 1994.
- [24] M. Molga and C. Smutnicki, *Test Functions for Optimization Needs*. Robert Marks.org, Apr. 2005.
- [25] X.-S. Yang. (2010). *Test Problems in Optimization*. [Online]. Available: <https://arxiv.org/abs/1008.0549>
- [26] S. Surjanovic and D. Bingham. (2013). *Virtual Library of Simulation Experiments: Test Functions and Datasets*, accessed on May 25, 2015. [Online]. Available: <http://www.sfu.ca/screen>
- [27] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Comput. Appl.*, vol. 26, no. 5, pp. 1257–1263, Dec. 2015.
- [28] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl. Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.
- [29] S. Mirjalili and S. Z. M. Hashim, "A new hybrid PSOGSA algorithm for function optimization," in *Proc. Int. Conf. Comput. Inf. Appl.*, 2010, pp. 374–377.
- [30] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Nov. 2007.
- [31] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, Dec. 1944.
- [32] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, p. 617, Dec. 2009.
- [33] 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 and Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011.
- [34] O. Kukr, "Analysis of the dynamics of a memoryless nonlinear gradient IIR adaptive notch filter," *Signal Process.*, vol. 91, no. 10, pp. 2379–2394, 2011.
- [35] E. Cuevas et al., "A comparison of evolutionary computation techniques for IIR model identification," *J. Appl. Math.*, vol. 2014, Dec. 2014, Art. no. 827206. [Online]. Available: <http://dx.doi.org/10.1155/2014/827206>

**YING LING** received the B.S. and M.S. degrees from the Shenyang Normal University of Computer Science and Technology, Shenyang, China, in 2015. Her current research interests are computational intelligence, evolutionary algorithms, and machine learning.



**YONGQUAN ZHOU** received the M.S. degree in computer science from Lanzhou University, Lanzhou, China, in 1993, and the Ph.D. degree in computation intelligence from Xidian University, Xi'an, China, in 2006. He is currently a Professor with Guangxi University for Nationalities. His research interests include computation intelligence, neural networks, and intelligence information processing.



**QIFANG LUO** received the M.S. degree in computer science from Guangxi University in 2005. Her research interests are in the areas of computation intelligence and intelligence information systems.

