Research Article

Biological Flower Pollination Algorithm with Orthogonal Learning Strategy and Catfish Effect Mechanism for Global Optimization Problems

Weijia Cui and Yuzhu He

School of Instrumentation Science and Opto-Electronics Engineering, Beihang University, Beijing 100191, China

Correspondence should be addressed to Weijia Cui; cuiweijia@buaa.edu.cn

Received 20 November 2017; Revised 13 February 2018; Accepted 25 February 2018; Published 8 April 2018

Academic Editor: Ricardo Soto

Copyright © 2018 Weijia Cui and Yuzhu He. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The flower pollination algorithm (FPA) is a novel optimization technique derived from the pollination behavior of flowers. However, the shortcomings of the FPA, such as a tendency towards premature convergence and poor exploitation ability, confine its application in engineering problems. To further strengthen FPA optimization performance, an orthogonal learning (OL) strategy based on orthogonal experiment design (OED) is embedded into the local pollination operator. OED can predict the optimal factor level combination by constructing a smaller but representative test set based on an orthogonal array. Using this characteristic of OED, the OL strategy can extract a promising solution from various sources of experience information, which leads the population to a potentially reasonable search direction. Moreover, the catfish effect mechanism is introduced to focus on the worst individuals during the iteration process. This mechanism explores new valuable information and maintains superior population diversity. The experimental results on benchmark functions show that our proposed algorithm significantly enhances the performance of the basic FPA and offers stronger competitiveness than several state-of-the-art algorithms.

1. Introduction

Conventional optimization methods face serious challenges in modern sciences because the characteristics of optimization problems are often noncontinuous, nonlinear, multivariate, or nonconvex [1]. Currently, swarm intelligence algorithms are efficiently used to solve complex optimization problems. Most swarm intelligence algorithms are developed by simulation of foraging behavior, migration patterns, or the evolutionary approach in natural species, and these algorithms include the genetic algorithm (GA) [2], particle swarm optimization (PSO) [3], differential evolution (DE) [4], shuffled frog leaping algorithm (SFLA) [5], biogeography-based optimization (BBO) [6], cuckoo search (CS) [7], krill herd algorithm (KH) [8], fruit fly optimization (FFO) [9], pigeon inspired optimization (PIO) [10], invasive weed optimization (IWO) [11], and bat algorithm (BA) [12].

As a novel heuristic algorithm, the flower pollination algorithm (FPA) is inspired by the pollination behavior of

flowers. In nature, the pollination methods for flowers involve two main types: cross-pollination and self-pollination [13]. In cross-pollination, certain birds act as global pollinators that transfer pollen to the flowers of more distant plants. By contrast, in self-pollination, pollen is spread by the wind and only between adjacent flowers at the same plant. As such, the FPA is developed by mapping the two forms of cross-pollination and self-pollination into global pollination and local pollination operators, respectively. The FPA has attracted significant attention due to its merits of simple principles, few parameters, and ease of operation.

However, similar to other swarm intelligence algorithms, the FPA also suffers from a lack of perfect compromise between global exploration and local exploitation. Therefore, researchers have proposed various improved strategies to enhance the FPA search ability. Nabil [14] introduced the clone selection mechanism into the local pollination operator to obtain more accurate solutions. Hoang et al. [15] integrated the mutation, cross, and selection operators of DE to replace

the original update method for local pollination. Zhou et al. [16] developed the elite opposition mechanism, the local self-adaptive greedy strategy, and a dynamic switch probability method to balance exploration and exploitation. Salgotra and Singh [17] analyzed the influence of the various mutation techniques on the performance of the basic FPA. These variants exhibited excellent performance on benchmark functions. Additionally, the FPA has achieved favorable results in selected practical engineering applications. Abdelaziz et al. [18] used the basic FPA to solve the economic load dispatch problem. Wang et al. [19] conducted cluster analysis using bee pollinator-based FPA. Zhang et al. [20] proposed a chaotic local search-based FPA to forecast wind speed. Xu et al. [21] hybridized the FPA with the good point set theory and Deb's heuristic rules to identify multipass turning parameters. Salgotra and Singh [22] implemented the bat algorithm and the FPA in parallel to synthesize the linear antenna arrays.

As is known to all, standard FPA adopts the differential learning method to achieve the global exploration and refine exploitation. However, this simple evolution pattern is inefficient and presents some drawbacks. First, the random search behavior around the current individual is lack of an effective mechanism to avoid the premature convergence; moreover, the update formula works on the complete solution vector rather than dimension by dimension; it may cause the consequence of "two steps forward, on step back" [23]. The main reason is that different solution vectors may contain different qualities of components; thus some high-quality components at some dimensions may improve the current individual, while some inferior components may result in a degeneration phenomenon at some certain dimensions. Through the above analysis, how to keep the population diversity and make better use of the various experience information to exploit the higher-quality solution becomes the critical factors of improving the optimizing capability.

Orthogonal experiment design (OED) as a typically analytical tool of experiment scheme can effectively predict the best combination of levels at different factors through choosing a smaller but most representative test set. In this paper, considering that potentially valuable information might exist in certain dimensions in different candidate solution vectors, an OL strategy based on orthogonal experiment design (OED) [24] is developed to obtain more promising candidate solution by combining the useful information among the best individual, current individual, and a random selected individual. In this way, the convergence accuracy and speed would be significantly improved. However, although OL strategy can improve the optimizing efficiency, the algorithm still suffers from the loss of diversity. To overcome this problem, the catfish effect mechanism [25, 26] is introduced to enhance the population diversity by replacing the worst individuals with their own opposition information. Specifically, as the iterations progress, the worst individuals are coming closer and closer to the best individual, and once the algorithm traps into the local convergence, catfish effect mechanism can drive the worst subpopulation to explore the new region. In this way, the population diversity is kept. In sum, the incorporation of OL strategy and catfish effect mechanism into the FPA guarantees the search stability

of algorithm and can effectively improve the convergence performance. The experimental results demonstrate that our proposed algorithm is superior to the original FPA and other advanced evolutionary algorithms in testing on benchmark functions.

The remainder of this paper is described as follows. Section 2 describes the model of global optimization problems, Section 3 reviews the principle of FPA in detail, Section 4 elaborates on the OL strategy and catfish effect mechanism, Section 5 demonstrates the optimization performance of OCFPA on the benchmark functions, and Section 6 summarizes the conclusions.

2. Global Optimization Problems

Many engineering problems can be considered as global optimization problems [27, 28], which can be described as follows:

$$\min \quad (\text{or max}) \quad f(SX)$$

$$SX = (sx_1, \dots, sx_d, \dots, sx_D) \in S = \prod_{d=1}^D [l_d, u_d], \quad (1)$$

where SX is the vector form of the decision variable, D is the number of decision variables, $f(SX)$ is the objective function, S defines the decision space (i.e., the search space of the optimization algorithm), and l_d and u_d are the lower and upper limits of decision variable sx_d , respectively. Global optimization problem is to obtain the best decision variable SX by minimizing or maximizing the object function $f(SX)$.

3. Flower Pollination Algorithm

The FPA is a population-based global optimization algorithm. In [13], Yang et al. summarized the characteristics of the flower pollination process into four ideal rules as follows.

Rule 1. Cross-pollination conducted by the pollinators such as birds is viewed as a global pollination phase and the pollinators act with levy flight behavior.

Rule 2. Self-pollination occurring on the nearby flowers is viewed as a local pollination process.

Rule 3. The flower constancy is treated as the reproductive rate, which is direct ratio with the similar level of two involved flowers.

Rule 4. Global pollination and local pollination are implemented based on a switch probability.

According to the above rules, the FPA contains a global pollination operator and a local pollination operator. In the FPA, each pollen item is treated as a solution SX_i , and the solutions are initialized with random vectors in the feasible search space. The initial formula is given as follows:

$$SX_i = \text{Lower} + Rv(\text{Upper} - \text{Lower}), \quad (2)$$

where $i \in \{1, \dots, NP\}$, NP is the population size; Rv is a D -dimensional random vector in $[0, 1]^D$; the lower limit of

search space is Lower = $[l_1, \dots, l_d, \dots, l_D]$, and the upper limit is Upper = $[u_1, \dots, u_d, \dots, u_D]$.

In the global pollination operation, pollinators such as birds have a relatively large movement range and can carry pollen over a long distance. Thus, Rules 1 and 3 are formulated as follows:

$$SX_i^{t+1} = SX_i^t + \gamma L(\lambda) (gbest - SX_i^t), \quad (3)$$

where SX_i^t is the solution i at iteration t , $gbest$ is the current global best solution, γ is a step factor, the flight feature of birds can be numerically imitated by a levy distribution denoted by $L(\lambda)$ in (3), and $L(\lambda)$ can also be considered a varying step factor to quantify the intensity of pollination. The levy distribution when $L > 0$ can be described as follows:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, \quad (4)$$

$$s \gg s_0 > 0, \quad s_0 = 0.1,$$

where $\Gamma(\lambda)$ is the standard gamma function with $\lambda = 1.5$ and s is determined with two Gaussian distributions U and V as follows:

$$s = \frac{U}{|V|^{1/\lambda}}, \quad U \sim N(0, \delta^2), \quad V \sim N(0, 1), \quad (5)$$

$$\delta^2 = \left\{ \frac{\Gamma(1+\lambda)}{\lambda \Gamma[(1+\lambda)/2]} \cdot \frac{\sin(\pi\lambda/2)}{2^{(\lambda-1)/2}} \right\}^{1/\lambda},$$

where $N(0, \delta^2)$ denotes the normal distribution with the mean value 0 and variance δ^2 ; $N(0, 1)$ denotes the standard normal distribution.

If the pollination activities involve local pollination, then pollen are spread to a local neighbor, and the model can be formulated with Rules 2 and 3 as follows:

$$SX_i^{t+1} = SX_i^t + \xi (SX_j^t - SX_k^t), \quad (6)$$

where SX_j and SX_k are the pollen randomly selected from different flowers in the same plant, where j and $k \in \{1, \dots, NP\}$ and ξ is a D -dimensional random vector in $[0, 1]^D$. In addition, from Rule 4, the two pollination activities occur randomly and are determined by a probability p . In other words, if a random value $rand$ in $[0, 1]$ is smaller than p , then global pollination is conducted, otherwise, vice versa. The flowchart of the FPA is shown in Figure 1.

4. OCFPA

4.1. Orthogonal Experimental Design. It is assumed that an experimental system contains F factors and Q levels per factor. To obtain the best combination of levels from each factor, the direct method is used to assess the overall Q^F combinations. However, the calculation cost of this process is relatively high, particularly when F and Q are large. Although OED can achieve this purpose by testing only a smaller number of experiments, OED is performed based on an orthogonal array (OA) and factor analysis (FA) [29].

The OA denoted by $L_M(Q^F)$ is used to create the M most representative combinations for OED, where L represents the OA, $M = Q^J$, and J is a minimum integer that satisfies $(Q^J - 1)/(Q - 1) = F$. Additionally, the OA is constructed by generating the basic columns and nonbasic columns; the detailed procedures are given in [24]. In this work, an example of OA $L_8(2^6)$ is shown in (7). It is clear that there are 6 factors (column number), two levels per factor (1 or 2), and 8 combinations (row number).

$$L_8(2^6) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 2 & 2 & 1 & 1 & 2 \\ 1 & 2 & 2 & 2 & 2 & 1 \\ 2 & 1 & 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 2 & 1 & 2 \\ 2 & 2 & 1 & 1 & 2 & 2 \\ 2 & 2 & 1 & 2 & 1 & 1 \end{bmatrix}. \quad (7)$$

To more clearly illustrate the use of OED, a four-dimensional *SumSquare* function, that is, $R = A^2 + 2B^2 + 3C^2 + 4D^2$, is chosen as a test case. In this case, there are four factors (designated A , B , C , and D), and each factor has two levels: $[A_1, B_1, C_1, D_1] = [2 \ 8 \ 3 \ 1]$ and $[A_2, B_2, C_2, D_2] = [5 \ 3 \ 5 \ 7]$. Because a subarray of an OA is also an OA, the first four columns of $L_8(2^6)$ are chosen as the OA of this case. In Table 1, each row corresponds to a combination such as A_1, B_1, C_1 , and D_1 in combination 1 or A_1, B_1, C_1 , and D_2 in combination 2. Thus, 8 combinations are constructed by combining the various levels of the factors based on the OA. All combinations are tested based on the function formula, and the evaluation results $R_{OA} : \{R_1, R_2, \dots, R_8\}$ are also given in Table 1.

FA is used to further measure the average effect of each level at each factor based on the evaluation results R_{OA} . The FA can effectively obtain the most beneficial level at each factor and then may predict one better combination of levels. The average effect E_{gk} of level k of factor g is computed by the following formula:

$$E_{gk} = \frac{\sum_{i=1}^M R_i w_{igk}}{\sum_{i=1}^M w_{igk}}, \quad (8)$$

where R_i is the evaluation result of combination i , and if the level of factor g in combination i is k , $w_{igk} = 1$; otherwise, $w_{igk} = 0$. For a minimization problem, the smaller the value of E_{gk} is, the better the level k is at factor g . The procedure of FA in this case is shown in Table 2. We can see that a new combination $[A_1, B_2, C_1, D_1] = [2, 3, 3, 1]$ in the FA result is discovered by choosing the level with a smaller E_{gk} at each factor, and, apparently, the evaluation result R_{FA} is superior to R_{OA} .

4.2. OL Strategy. In the local pollination operator, the exploitation ability of differential search patterns is limited [30]. To enhance the performance, an OL strategy based on

FIGURE 1: Flowchart of the FPA.

TABLE 1: OA of the test case.

Combination	Factors				R_{OA}
	A	B	C	D	
(1)	$A_1(2)$	$B_1(8)$	$C_1(3)$	$D_1(1)$	$R_1 = 163$
(2)	$A_1(2)$	$B_1(8)$	$C_1(3)$	$D_2(7)$	$R_2 = 355$
(3)	$A_1(2)$	$B_2(3)$	$C_2(5)$	$D_1(1)$	$R_3 = 101$
(4)	$A_1(2)$	$B_2(3)$	$C_2(5)$	$D_2(7)$	$R_4 = 293$
(5)	$A_2(5)$	$B_1(8)$	$C_2(5)$	$D_1(1)$	$R_5 = 232$
(6)	$A_2(5)$	$B_1(8)$	$C_2(5)$	$D_2(7)$	$R_6 = 424$
(7)	$A_2(5)$	$B_2(3)$	$C_1(3)$	$D_1(1)$	$R_7 = 74$
(8)	$A_2(5)$	$B_2(3)$	$C_1(3)$	$D_2(7)$	$R_8 = 266$

TABLE 2: FA of the test case.

Factor	Level	E_{gk}	FA result	R_{FA}
A	2	$(R_1 + R_2 + R_3 + R_4)/4 = \mathbf{228}$	2 (A_1)	53
	5	$(R_5 + R_6 + R_7 + R_8)/4 = 249$		
B	8	$(R_1 + R_2 + R_5 + R_6)/4 = 293.5$	3 (B_2)	
	3	$(R_3 + R_4 + R_7 + R_8)/4 = \mathbf{183.5}$		
C	3	$(R_1 + R_2 + R_7 + R_8)/4 = \mathbf{214.5}$	3 (C_1)	
	5	$(R_3 + R_4 + R_5 + R_6)/4 = 262.5$		
D	1	$(R_1 + R_3 + R_5 + R_7)/4 = \mathbf{142.5}$	1 (D_1)	
	7	$(R_2 + R_4 + R_6 + R_8)/4 = 334.5$		

OED is incorporated into the local search phase. The motivation for this choice is that by applying the principle of OED, OL can systematically and reasonably combine the useful information on each dimension from various individuals, and a high-quality solution can be implemented to effectively guide the population towards a promising search direction.

To add sufficient information to the search, the OL strategy should act on three individuals: one is the current best individual $gbest$ used to supply the advanced guidance information, and the other two are the current individual SX_i and a randomly selected individual SX_k , which are used to share their search experiences and enrich the diversity of information. Thus, a novel local update method of SX_i is formed based on the OL strategy as follows:

$$SX_i^t = \text{OED}(SX_i^t, gbest, SX_k^t). \quad (9)$$

In OL, the factor number F corresponds to the problem dimensionality D , and the level number Q in each factor is equal to the number of the selected individuals, that is, 3. Once the OL strategy is implemented, the value lv_{di} of level i ($i = 1, 2, 3$) at dimension d can be determined:

$$\begin{aligned} lv_{d1} &= SX_{id}, \\ lv_{d2} &= gbest_d, \\ lv_{d3} &= SX_{kd}. \end{aligned} \quad (10)$$

Subsequently, M candidate solutions C are generated based on OA $L_M(3^D)$ as follows:

$$\begin{aligned} C_1 &= (lv_{11}, lv_{21}, \dots, lv_{d1}, \dots, lv_{D1}) \\ &\vdots \\ C_m &= (lv_{12}, lv_{23}, \dots, lv_{d2}, \dots, lv_{D2}) \\ &\vdots \\ C_M &= (lv_{13}, lv_{23}, \dots, lv_{d3}, \dots, lv_{D3}). \end{aligned} \quad (11)$$

The candidate solutions C are evaluated by the specified objective function, and the FA is conducted. The implementation of the OL strategy is shown in Algorithm 1.

As shown in steps (16)–(21) of Algorithm 1, it should be noted that each execution of OL increases M fitness evaluations. Taking a 30-dimensional problem, for example, when OL strategy works on three individuals, there are $3^4 = 81$ solutions to be generated based on OA and then evaluated by the specified objective function. Therefore, the use of OL strategy will result in larger computational complexity. To maintain the computational efficiency at every iteration, the OL strategy is applied only when the current solution i enters into local pollination and the index i is equal to the random integer u , where $u \in \{1, \dots, NP\}$. In this manner, the local exploitation ability can be enhanced with smaller consumption of computational overhead.

4.3. Catfish Effect Mechanism. In real life, to maintain the freshness of sardines, fishermen always place catfish into a sardine pond. The catfish disturb the living environment of the sardines to activate their survival ability. The catfish effect is derived from this phenomenon and has been successfully incorporated into PSO [25, 26].

To avoid the premature convergence in the FPA, the catfish effect mechanism is introduced to force the worst individuals to explore new regions and obtain potentially better candidate solutions. The working principle of this mechanism is that if the fitness value of the current best solution has not been enhanced in n consecutive iterations, the worst “sardine” individuals WX in 10% of the population are replaced with new “catfish” individuals CX . In this paper, the “catfish” individual is considered an opposition solution of the “sardine” individual and can be computed as follows:

$$CX_{id} = l_d + u_d - WX_{id}, \quad (12)$$

where i is the index of the “sardine” individuals. The catfish effect mechanism is shown in Algorithm 2.

4.4. Main Procedure of the OCFPA. By embedding the OL strategy and catfish effect mechanism into the basic FPA, the modified method OCFPA is developed. To better demonstrate our proposed algorithm, the complete pseudocode of the OCFPA is shown in Algorithm 3.

5. Experimental Verification and Comparisons

To verify the effectiveness of our proposed algorithm, fourteen benchmark functions [31, 32] were tested. Table 3 lists the information for each function, including the function name, scope of variables, optimal solution, and global optimum. As such, f_{01} – f_{07} are unimodal functions used primarily to test the optimal efficiency, and f_{08} – f_{14} are multimodal functions used primarily to test the ability to escape local optima. The parameter settings of FPA and OCFPA are $\gamma = 3$ and $p = 0.8$. The number of consecutive iterations n in catfish effect mechanism is set to 7. To guarantee fairness of comparison, the population size NP and fitness evaluations FEs are set to 20 and 2×10^5 , respectively. All statistical results are obtained with 30 independent runs using the involved algorithms. The better results are marked in bold. The subsequent experiments are executed on a PC with a 3.2 GHz CPU and 2.5 GB memory running the Windows XP operating system.

5.1. Comparison of FPA and OCFPA. This subsection investigates the improvement degree of our proposed algorithm against the basic FPA. Table 4 lists the statistical results, including the mean and standard deviation values (Std), for the FPA and OCFPA. From Table 4, the OCFPA is superior to the FPA on all benchmark functions. In solving the unimodal functions, the OCFPA finds better solutions than the FPA on f_{01} , f_{03} , f_{05} , and f_{07} , and although it is challenging to find the global optima of f_{02} and f_{04} due to their complex function characteristics, the OCFPA still outperforms the FPA. The results indicate that the OCFPA has excellent


```

(01) Input
(02) Number of factors ( $F = D$ ) and number of levels ( $Q = 3$ ).
(03) Three individuals:  $SX_i$ ,  $gbest$ ,  $SX_k$ .
(04) Objective function  $f(solution)$  of minimization or maximization problems.
(05) Number of fitness evaluations FEs = 0.
(06) Output
(07) OL result.
(08) Begin
(09) % Construct the OA based on  $F$  and  $Q$ .
(10) Compute the minimum integer  $J$  that satisfies  $(Q^J - 1)/(Q - 1) = F$ , then  $M = Q^J$ .
(11) Construct an OA  $L_M(3^D)$  based on [24].
(12) % Compute the value of each level at each dimension by using (10).
(13) For  $d = 1 : D$ 
(14)    $lv_{d1} = SX_{id}$ ;  $lv_{d2} = gbest_d$ ;  $lv_{d3} = SX_{kd}$ .
(15) End for
(16) Generate  $M$  solutions  $C_1, \dots, C_m, \dots, C_M$  based on  $L_M(3^D)$ .
(17) The  $M$  solution vectors are formulated as Equation (11).
(18) % Evaluate the  $M$  solutions with objective function  $f$ .
(19) The  $M$  results are:  $f(C_1), \dots, f(C_m), \dots, f(C_M)$ .
(20) % Accumulate the number of fitness evaluations.
(21) FEs = FEs +  $M$ .
(22) % Choose the best solution  $C_{OA}$ .
(23)  $C_{OA} = \arg \min \text{ or } \max [f(C_1), \dots, f(C_m), \dots, f(C_M)]$ .
(24) % Compute the average effect  $E_{dk}$  of each level  $k(1, 2, 3)$  at each dimension  $d$ .
(25) For  $d = 1 : D$ 
(26) % Find the row index that having the same level at dimension  $d$  based on OA.
(27) % Compute the mean value of fitness results according to the index vector.
(28)    $Index\_1 = \text{find}(OA(:, d) == 1)$ ,  $E_{d1} = \text{mean}(f(Index\_1))$ ;
(29)    $Index\_2 = \text{find}(OA(:, d) == 2)$ ,  $E_{d2} = \text{mean}(f(Index\_2))$ ;
(30)    $Index\_3 = \text{find}(OA(:, d) == 3)$ ,  $E_{d3} = \text{mean}(f(Index\_3))$ ;
(31)   Compare  $E_{d1}$ ,  $E_{d2}$ ,  $E_{d3}$ , select the most beneficial level  $k$  at dimension  $d$ .
(32) End for
(33) Construct  $C_{FA}$  by combining all the most beneficial level values  $lv_{dk}$ .
(34) Compare  $C_{OA}$  and  $C_{FA}$ , and choose the best one as the OL result.
(35) End

```

ALGORITHM 1: OL strategy.

TABLE 3: Benchmark functions.

f	Function name	Scope	Optimal solution	Optimum
f_{01}	Sphere function	$[-100, 100]^D$	$SX^* = (0, 0, \dots, 0)$	0
f_{02}	Quartic function with noise	$[-1.28, 1.28]^D$	$SX^* = (0, 0, \dots, 0)$	0
f_{03}	High conditioned elliptic function	$[-100, 100]^D$	$SX^* = (0, 0, \dots, 0)$	0
f_{04}	Rosenbrock function	$[-30, 30]^D$	$SX^* = (0, 0, \dots, 0)$	0
f_{05}	Schwefel problem 2.22 function	$[-10, 10]^D$	$SX^* = (0, 0, \dots, 0)$	0
f_{06}	Step function	$[-100, 100]^D$	$SX^* = (0, 0, \dots, 0)$	0
f_{07}	Dixon-price function	$[-10, 10]^D$	$SX^* = (0, 0, \dots, 0)$	0
f_{08}	Schwefel function	$[-500, 500]^D$	$SX^* = (420.9687, \dots)$	-418.9829D
f_{09}	Rastrigin function	$[-5.12, 5.12]^D$	$SX^* = (0, 0, \dots, 0)$	0
f_{10}	Griewank function	$[-600, 600]^D$	$SX^* = (0, 0, \dots, 0)$	0
f_{11}	Ackley function	$[-32, 32]^D$	$SX^* = (0, 0, \dots, 0)$	0
f_{12}	Noncontinuous rastrigin function	$[-5.12, 5.12]^D$	$SX^* = (0, 0, \dots, 0)$	0
f_{13}	Generalized penalized function 1	$[-10, 10]^D$	$SX^* = (0, 0, \dots, 0)$	0
f_{14}	Generalized penalized function 2	$[-10, 10]^D$	$SX^* = (0, 0, \dots, 0)$	0

```

(01) Input
(02) Number of population NP, population dimensionality  $D$ .
(03) Current iteration number  $t$ , current iterative population  $SX$ .
(04) Objective function  $f(solution)$  of minimization or maximization problems.
(05) The storage vector  $f_{best}$  of historical best fitness.
(06) Consecutive iteration number  $n$ .
(07) Output
(08) "Catfish" individuals  $CX$ .
(09) Begin
(10) If  $t > n$ 
(11) % The global best individual has not improved in  $n$  consecutive iterations.
(12) If  $\min(f(SX)) - f_{best}(t - n) == 0$ 
(13) % Sort the population from good to bad based on fitness.
(14)  $[sort\_f, sort\_ind] = \text{sort}(f(SX))$ .
(15) % Choose 10% worst individuals as  $WX$  based on  $sort\_ind$ .
(16)  $WX = SX(sort\_ind(0.9NP + 1 : NP), :)$ .
(17) % Generate the "catfish" individuals  $CX$  by using (12).
(18) For  $i = 1$  to the number of  $WX$ 
(19) For  $d = 1 : D$ 
(20)  $CX_{id} = l_d + u_d - WX_{id}$ .
(21) End for
(22) End for
(23) Compute the fitness value  $f(CX_i)$ .
(24) End if
(25) Replace all the worst individuals  $WX$  with "catfish" individuals  $CX$ .
(26) End if
(27) End

```

ALGORITHM 2: Catfish effect mechanism.

TABLE 4: Comparison of the FPA and OCFPA ($D = 30$).

f	FPA		OCFPA	
	Mean	Std	Mean	Std
f_{01}	$1.56E - 25$	$3.20E - 25$	$1.09E - 86$	$3.27E - 86$
f_{02}	$3.37E - 02$	$1.56E - 02$	$1.70E - 03$	$9.08E - 04$
f_{03}	$6.89E - 22$	$2.10E - 21$	$2.09E - 81$	$7.67E - 81$
f_{04}	$2.41E + 01$	$1.81E + 01$	$1.08E + 01$	$7.11E + 00$
f_{05}	$5.51E - 17$	$6.12E - 17$	$2.38E - 47$	$6.13E - 47$
f_{06}	0	0	0	0
f_{07}	$5.94E - 21$	$3.25E - 20$	$1.36E - 86$	$4.54E - 86$
f_{08}	-10308	$4.92E + 02$	-12569	$4.05E - 12$
f_{09}	$4.12E + 01$	$9.59E + 00$	0	0
f_{10}	$1.40E - 03$	$3.20E - 03$	0	0
f_{11}	$8.21E - 01$	$6.40E - 01$	$4.32E - 15$	$6.49E - 16$
f_{12}	$2.22E + 01$	$5.08E + 00$	0	0
f_{13}	$1.35E - 27$	$2.01E - 27$	$1.61E - 32$	$1.18E - 33$
f_{14}	$1.47E - 24$	$7.99E - 24$	$2.09E - 32$	$4.56E - 32$

exploitation ability and optimal efficiency. In addition, for multimodal functions f_{08} – f_{14} with many local optima, the OCFPA achieves global or near-global optimal solutions and surpasses the FPA in accuracy and robustness. This result indicates that the OCFPA can avoid local convergence and obtain more accurate solutions. In short, the comparative results illustrate that our proposed algorithm has a faster

optimal speed and can converge to a higher accuracy than the basic FPA.

5.2. Influence of Two Strategies. We introduced the OL strategy and catfish effect mechanism to enhance the search ability of the FPA. Therefore, it is necessary to analyze the contribution of each strategy to the optimization performance.

```

(01) Input
(02) Pollen  $SX$ .
(03) Objective function  $f(SX)$  of minimization or maximization problems.
(04) Number of pollen (NP), number of fitness evaluations (FEs).
(05) Switch probability ( $p$ ), step factor ( $\gamma$ ).
(06) Output
(07) Global best pollen ( $gbest$ ).
(08) Begin
(09) % Initialize the population of pollen randomly.
(10) For  $i = 1$  to number of pollen (NP)
(11)  $SX_i = \text{Lower} + Rv(\text{Upper} - \text{Lower})$ .
(12) Compute the fitness value  $f(SX_i)$  and store it.
(13) End for
(14) The pollen with the best fitness value is chosen as the current best pollen.
(15) While (the maximum number of fitness evaluations is not reached)
(16) Draw a random integer  $u \in \{1, \dots, NP\}$ .
(17) For  $i = 1$  to number of pollen (NP)
(18) If a random number in  $[0, 1] < \text{switch probability } (p)$ 
(19) % conduct global pollination.
(20)  $SX_i^{t+1} = SX_i^t + \gamma L(\lambda) (gbest - SX_i^t)$ .
(21) Else
(22) % conduct local pollination.
(23) If  $i \sim u$ 
(24) Draw a random vector  $\xi \in [0, 1]^D$ .
(25) Draw two random integers  $j$  and  $k \in \{1, \dots, NP\}$ .
(26)  $SX_i^{t+1} = SX_i^t + \xi(SX_j^t - SX_k^t)$ .
(27) Else
(28) Generate  $SX_i^{t+1}$  by Performing OL strategy according to Algorithm 1.
(29) End if
(30) End if
(31) Compute the fitness value  $f(SX_i)$ .
(32) Update  $SX_i$  if the current individual is superior to its previous one.
(33) End for
(34) Find the pollen with the best fitness in the population.
(35) Update  $gbest$  if the current best pollen beats the previous best pollen.
(36) Perform catfish effect mechanism according to Algorithm 2.
(37) Return to the next generation until stop criterion is reached.
(38) End while
(39) Output  $gbest$ .
(40) End

```

ALGORITHM 3: The pseudocode of the OCFPA.

To accomplish this aim, two strategies are combined with the FPA, namely, the OL-based FPA (OFPA) and the catfish-effect-based FPA (CFPA).

Table 5 shows the experimental results of the OFPA, CFPA, and OCFPA in the form of “mean (standard deviation).” Relative to the OFPA, the OCFPA is superior on 13 functions except for f_{04} . The results indicate that it is unlikely that the OL strategy will enhance the quality of the solution once the algorithm becomes trapped in local optima, because OL mainly serves to recombine various pieces of information and is incapable of exploring new regions to find more valuable information. With respect to the CFPA, the OCFPA is clearly superior on 10 functions, but for f_{06} , f_{10} , and f_{11} , the optimization performances of the CFPA and OCFPA are equivalent. This result implies that the catfish

effect mechanism has the ability to expand the search space to obtain a more promising candidate solution.

By analyzing the experimental results from Table 5, it can be concluded that the mutual interaction of the two strategies is beneficial to OCFPA performance. In detail, the catfish effect mechanism can explore the new search space and supply valuable individuals for the OL strategy, and the OL strategy can take full advantage of the useful information to acquire a high-precision solution.

5.3. Influence of Population Size. To study the influence of different population sizes on the search ability of our proposed algorithm, the FPA and OCFPA with $NP = 20, 50, 80$, and 100 are implemented in comparative experiments. The results of these four cases are listed in the form of “mean (standard deviation)” in Table 6. It is clear that, for

TABLE 5: Comparison of the OFPA, CFPA, and OCFPA ($D = 30$).

f	OFPA	CFPA	OCFPA
f_{01}	$2.31E - 58$ ($1.26E - 57$)	$1.18E - 51$ ($1.74E - 51$)	$1.09E - 86$ ($3.27E - 86$)
f_{02}	$3.17E - 02$ ($1.71E - 02$)	$1.20E - 02$ ($4.72E - 04$)	$1.70E - 03$ ($9.08E - 04$)
f_{03}	$1.98E - 70$ ($9.05E - 70$)	$5.52E - 48$ ($6.79E - 48$)	$2.09E - 81$ ($7.67E - 81$)
f_{04}	$2.13E + 00$ ($2.02E - 00$)	$2.38E + 01$ ($3.02E - 01$)	$1.08E + 01$ ($7.11E + 00$)
f_{05}	$7.99E - 02$ ($2.15E - 01$)	$5.66E - 26$ ($6.88E - 26$)	$2.38E - 47$ ($6.13E - 47$)
f_{06}	$4.33E - 01$ ($1.65E + 00$)	0 (0)	0 (0)
f_{07}	$1.14E - 13$ ($6.26E - 13$)	$1.32E - 52$ ($1.65E - 32$)	$1.36E - 86$ ($4.54E - 86$)
f_{08}	-12712 ($2.15E + 02$)	-9922.6 ($5.49E + 02$)	-12569 ($4.05E - 12$)
f_{09}	$9.88E + 00$ ($3.23E + 00$)	$2.21E + 01$ ($2.11E + 01$)	0 (0)
f_{10}	$5.62E - 02$ ($5.98E - 02$)	0 (0)	0 (0)
f_{11}	$3.45E + 00$ ($1.38E + 00$)	$4.32E - 15$ ($6.49E - 16$)	$4.32E - 15$ ($6.49E - 16$)
f_{12}	$7.67E + 00$ ($2.78E + 00$)	$2.21E + 01$ ($7.20E + 00$)	0 (0)
f_{13}	$1.39E - 13$ ($7.61E - 13$)	$5.14E - 24$ ($2.70E - 23$)	$1.61E - 32$ ($1.18E - 33$)
f_{14}	$3.96E - 02$ ($1.83E - 01$)	$3.00E - 03$ ($1.65E - 02$)	$2.09E - 32$ ($4.56E - 32$)

TABLE 6: Comparison of different population sizes ($D = 30$, FEs = 2×10^5).

f	NP = 20		NP = 50	
	FPA	OCFPA	FPA	OCFPA
f_{01}	$1.56E - 25$ ($3.20E - 25$)	$1.09E - 86$ ($3.27E - 86$)	$1.26E - 12$ ($6.73E - 13$)	$1.69E - 35$ ($5.05E - 35$)
f_{02}	$3.37E - 02$ ($1.56E - 02$)	$1.70E - 03$ ($9.08E - 04$)	$1.44E - 02$ ($4.40E - 03$)	$5.60E - 03$ ($2.00E - 03$)
f_{03}	$6.89E - 22$ ($2.10E - 21$)	$2.09E - 81$ ($7.67E - 81$)	$2.53E - 09$ ($1.56E - 09$)	$1.41E - 32$ ($2.87E - 32$)
f_{04}	$2.41E + 01$ ($1.81E + 01$)	$1.08E + 01$ ($7.11E + 00$)	$1.97E + 01$ ($2.52E + 00$)	$1.04E + 01$ ($1.13E + 01$)
f_{05}	$5.51E - 17$ ($6.12E - 17$)	$2.38E - 47$ ($6.13E - 47$)	$3.70E - 07$ ($1.40E - 07$)	$3.11E - 20$ ($6.40E - 20$)
f_{06}	0 (0)	0 (0)	0 (0)	0 (0)
f_{07}	$5.94E - 21$ ($3.25E - 20$)	$1.36E - 86$ ($4.54E - 86$)	$1.60E - 13$ ($7.74E - 14$)	$5.87E - 37$ ($1.62E - 36$)
f_{08}	-10308 ($4.92E + 02$)	-12569 ($4.05E - 12$)	-8989.7 ($3.33E + 02$)	-12569 ($2.07E - 01$)
f_{09}	$4.12E + 01$ ($9.59E + 00$)	0 (0)	$5.76E + 01$ ($9.18E + 00$)	0 (0)
f_{10}	$1.40E - 03$ ($3.20E - 03$)	0 (0)	$2.85E - 08$ ($8.15E - 08$)	0 (0)
f_{11}	$8.21E - 01$ ($6.40E - 01$)	$4.32E - 15$ ($6.49E - 16$)	$2.62E - 04$ ($3.11E - 04$)	$2.56E - 12$ ($2.05E - 12$)
f_{12}	$2.22E + 01$ ($5.08E + 00$)	0 (0)	$4.63E + 01$ ($7.92E + 00$)	0 (0)
f_{13}	$1.35E - 27$ ($2.01E - 27$)	$1.61E - 32$ ($1.18E - 33$)	$6.41E - 12$ ($1.23E - 11$)	$1.57E - 32$ ($6.24E - 35$)
f_{14}	$1.47E - 24$ ($7.99E - 24$)	$2.09E - 32$ ($4.56E - 32$)	$7.05E - 13$ ($8.28E - 13$)	$1.69E - 35$ ($5.05E - 35$)
f	NP = 80		NP = 100	
	FPA	OCFPA	FPA	OCFPA
f_{01}	$4.14E - 07$ ($1.32E - 07$)	$3.62E - 20$ ($7.87E - 20$)	$5.37E - 05$ ($1.83E - 05$)	$2.62E - 15$ ($3.61E - 15$)
f_{02}	$2.40E - 02$ ($4.50E - 03$)	$1.29E - 02$ ($4.70E - 03$)	$3.20E - 02$ ($8.60E - 03$)	$1.74E - 02$ ($6.00E - 03$)
f_{03}	$1.10E - 03$ ($5.27E - 04$)	$7.69E - 17$ ($1.71E - 16$)	$1.25E - 01$ ($3.22E - 02$)	$8.32E - 12$ ($1.62E - 11$)
f_{04}	$2.22E + 01$ ($2.85E + 00$)	$6.32E + 00$ ($6.09E + 00$)	$2.49E + 01$ ($2.99E + 00$)	$1.32E + 01$ ($9.82E + 00$)
f_{05}	$4.09E - 04$ ($1.01E - 04$)	$5.74E - 12$ ($6.91E - 12$)	$5.40E - 03$ ($9.98E - 04$)	$1.02E - 08$ ($1.93E - 08$)
f_{06}	0 (0)	0 (0)	0 (0)	0 (0)
f_{07}	$5.95E - 08$ ($2.26E - 08$)	$1.86E - 21$ ($4.71E - 21$)	$6.39E - 06$ ($1.83E - 06$)	$7.03E - 16$ ($1.46E - 15$)
f_{08}	-8711.7 ($3.47E + 02$)	-12569 ($2.80E - 12$)	-8518.8 ($2.52E + 02$)	-12569 ($2.62E - 04$)
f_{09}	$6.94E + 01$ ($1.02E + 01$)	$2.77E - 11$ ($1.52E - 10$)	$6.86E + 01$ ($1.06E + 01$)	$6.43E - 10$ ($3.46E - 09$)
f_{10}	$2.02E - 05$ ($3.37E - 05$)	$3.56E - 14$ ($8.80E - 14$)	$5.59E - 04$ ($2.40E - 04$)	$7.72E - 06$ ($4.17E - 05$)
f_{11}	$8.53E - 02$ ($6.44E - 02$)	$1.97E - 06$ ($2.01E - 06$)	$6.22E - 01$ ($3.38E - 01$)	$1.01E - 04$ ($4.85E - 05$)
f_{12}	$5.75E + 01$ ($8.36E + 00$)	$1.18E - 15$ ($5.51E - 15$)	$6.26E + 01$ ($7.58E + 00$)	$4.07E - 11$ ($1.67E - 10$)
f_{13}	$1.08E - 06$ ($9.97E - 07$)	$4.68E - 21$ ($9.92E - 21$)	$3.12E - 05$ ($2.54E - 05$)	$5.42E - 16$ ($1.03E - 15$)
f_{14}	$1.01E - 06$ ($7.37E - 07$)	$8.25E - 19$ ($3.03E - 18$)	$1.30E - 04$ ($7.19E - 05$)	$2.10E - 13$ ($3.02E - 13$)

TABLE 7: Comparison of different dimensions (NP = 20).

f	$D = 10/\text{FEs} = 1 \times 10^5$		$D = 50/\text{FEs} = 3 \times 10^5$	
	FPA	OCFPA	FPA	OCFPA
f_{01}	$4.69E - 37$ ($2.17E - 36$)	$3.06E - 76$ ($1.48E - 75$)	$1.06E - 25$ ($1.23E - 25$)	$8.84E - 63$ ($2.25E - 62$)
f_{02}	$3.20E - 03$ ($1.70E - 03$)	$1.20E - 03$ ($5.71E - 04$)	$9.16E - 02$ ($3.36E - 02$)	$2.10E - 03$ ($9.92E - 04$)
f_{03}	$1.09E - 34$ ($2.89E - 34$)	$1.58E - 73$ ($4.13E - 73$)	$9.09E - 22$ ($2.67E - 21$)	$1.23E - 58$ ($4.42E - 58$)
f_{04}	$1.32E + 01$ ($7.27E - 01$)	$1.33E - 01$ ($7.27E - 01$)	$7.68E + 01$ ($3.64E + 01$)	$3.65E + 01$ ($1.25E + 01$)
f_{05}	$6.49E - 20$ ($1.12E - 19$)	$4.92E - 43$ ($1.15E - 42$)	$6.15E - 18$ ($4.15E - 18$)	$7.85E - 33$ ($3.97E - 32$)
f_{06}	0 (0)	0 (0)	0 (0)	0 (0)
f_{07}	$2.83E - 38$ ($1.15E - 38$)	$6.13E - 77$ ($3.03E - 76$)	$3.19E - 26$ ($4.74E - 26$)	$5.75E - 62$ ($3.11E - 61$)
f_{08}	-4143.8 ($6.12E + 01$)	-4189.8 ($2.77E - 12$)	-15278 ($9.06E + 02$)	-20949 ($3.43E - 01$)
f_{09}	$2.77E + 00$ ($9.58E - 01$)	0 (0)	$8.04E + 01$ ($1.96E + 01$)	0 (0)
f_{10}	$5.35E - 02$ ($3.32E - 02$)	0 (0)	$2.70E - 03$ ($4.70E - 03$)	0 (0)
f_{11}	$7.70E - 02$ ($2.93E - 01$)	$4.20E - 15$ ($9.01E - 16$)	$1.44E + 00$ ($4.44E - 01$)	$4.32E - 15$ ($6.48E - 16$)
f_{12}	$2.81E + 00$ ($1.02E + 00$)	0 (0)	$5.41E + 01$ ($2.43E + 01$)	0 (0)
f_{13}	$4.83E - 32$ ($5.31E - 33$)	$4.72E - 32$ ($2.95E - 34$)	$4.10E - 03$ ($2.27E - 02$)	$8.15E - 19$ ($4.46E - 18$)
f_{14}	$5.49E - 33$ ($4.22E - 33$)	$3.12E - 33$ ($8.38E - 33$)	$3.00E - 03$ ($1.65E - 02$)	$4.25E - 33$ ($8.76E - 33$)
f	$D = 75/\text{FEs} = 4.5 \times 10^5$		$D = 100/\text{FEs} = 6 \times 10^5$	
	FPA	OCFPA	FPA	OCFPA
f_{01}	$6.30E - 28$ ($2.09E - 27$)	$1.57E - 91$ ($5.32E - 91$)	$3.07E - 30$ ($3.79E - 30$)	$1.10E - 118$ ($5.90E - 118$)
f_{02}	$1.64E - 01$ ($3.86E - 02$)	$1.40E - 03$ ($8.59E - 04$)	$2.60E - 01$ ($6.04E - 02$)	$1.30E - 03$ ($9.24E - 04$)
f_{03}	$5.17E - 25$ ($5.90E - 25$)	$1.11E - 85$ ($6.09E - 85$)	$7.28E - 27$ ($7.76E - 27$)	$2.9E - 117$ ($1.0E - 116$)
f_{04}	$1.48E + 02$ ($5.53E + 01$)	$5.79E + 01$ ($1.61E + 01$)	$1.91E + 02$ ($5.37E + 01$)	$6.87E + 01$ ($2.75E + 01$)
f_{05}	$1.67E - 20$ ($7.22E - 21$)	$6.12E - 50$ ($1.85E - 49$)	$1.92E - 22$ ($7.67E - 23$)	$4.74E - 65$ ($2.08E - 64$)
f_{06}	0 (0)	0 (0)	0 (0)	0 (0)
f_{07}	$9.11E - 29$ ($1.46E - 28$)	$1.11E - 91$ ($5.82E - 91$)	$1.88E - 29$ ($9.77E - 29$)	$4.91E - 119$ ($2.72E - 118$)
f_{08}	-22236 ($1.54E + 03$)	-31424 ($5.36E - 01$)	-29152 ($1.68E + 03$)	-41898 ($5.16E - 02$)
f_{09}	$1.43E + 02$ ($3.28E + 01$)	0 (0)	$1.75E + 02$ ($4.11E + 01$)	0 (0)
f_{10}	$4.00E - 03$ ($1.35E - 02$)	0 (0)	$2.50E - 03$ ($8.00E - 03$)	0 (0)
f_{11}	$1.63E + 00$ ($2.62E - 01$)	$4.32E - 15$ ($6.48E - 16$)	$2.07E + 01$ ($3.14E - 01$)	$4.32E - 15$ ($6.48E - 16$)
f_{12}	$6.08E + 01$ ($1.19E + 01$)	0 (0)	$7.99E + 01$ ($2.19E + 01$)	0 (0)
f_{13}	$1.40E - 03$ ($7.60E - 03$)	$6.95E - 33$ ($6.54E - 33$)	$2.10E - 03$ ($7.90E - 03$)	$3.82E - 25$ ($1.93E - 24$)
f_{14}	$3.40E - 03$ ($1.85E - 02$)	$2.07E - 30$ ($1.04E - 29$)	$9.40E - 03$ ($2.88E - 02$)	$2.57E - 33$ ($3.52E - 33$)

f_{01} – f_{14} , the OCFPA performs better than the FPA on the four cases. However, we note that the optimal accuracy of the FPA and OCFPA exhibits downward trends as the population size increases. The comparison results reflect the fact that a population size that is too large can result in a rapid consumption of fitness evaluations and prevent the algorithm from obtaining the global optimum. By contrast, a population that is too small can also have a negative influence on the search ability owing to the reduction in population diversity. Therefore, choosing a proper population size for various complex problems is a key step in achieving a better optimal effect.

5.4. Influence of Problem Dimensionality. The complexity of the problem increases with larger dimensions. Therefore, the optimization performance of OCFPA was investigated in different dimensionality situations. As shown in Table 7, the experiment involved four cases of $D = 10/\text{FEs} = 1 \times 10^5$, $D = 50/\text{FEs} = 3 \times 10^5$, $D = 75/\text{FEs} = 4.5 \times 10^5$, and $D = 100/\text{FEs}$

$= 6 \times 10^5$. The simulation results show that, for the four different dimensions, both the FPA and OCFPA can find the optimal solution 0 on f_{06} but that the OCFPA outperforms the FPA on the other remaining 13 functions. Furthermore, the convergence accuracies and optimal speeds of OCFPA in the four cases are considerable. These results prove that the introduction of these two strategies significantly improves the search ability of FPA, even when solving high-dimensional optimization problems.

5.5. Comparison with Several Evolutionary Algorithms. To further demonstrate the superiority of the OCFPA, several advanced evolutionary algorithms are introduced for comparison on the benchmark functions with $D = 30$, including CS [7], GSA [33], CMAES [34], OXDE [35], SGHS [36], and CLPSO [37]. The related parameters of these algorithms are in accordance with the references [7, 33–37]. Tables 8 and 9 list the results in the form of “Mean/(Std)/Rank.” The ranking of

FIGURE 2: Iterative curves on f_{01} , f_{02} , f_{05} , and f_{07} .

the performance of OCFPA and the other algorithms is based on the mean and Std values.

The comparative results of OCFPA and six competitors on the unimodal functions are presented in Table 8. The OCFPA has a better global optimization ability than the other algorithms on f_{01} , f_{02} , f_{03} , f_{05} , and f_{07} . For f_{04} , the OCFPA surpasses the CS, GSA, SGHS, and CLPSO methods but is surpassed by the CMAES and OXDE methods. For f_{06} , all algorithms converge to the optimal solution 0.

Table 9 lists the comparative results on seven multimodal functions; for f_{08} , f_{09} , and f_{12} , the OCFPA achieves the global optimal solutions and outperforms the other algorithms. For f_{10} , both the OCFPA and CLPSO converge to the global optimum 0 and outperform the other five algorithms. For f_{11} , the OCFPA surpasses the other algorithms in terms of search accuracy and robustness. For f_{13} and f_{14} , CLPSO performs slightly better than the OCFPA, but we can observe that CLPSO, OCFPA, and OXDE on f_{13} and CLPSO, OCFPA, OXDE, and CMAES on f_{14} achieve the same order of magnitude of accuracy.

To more intuitively reflect the optimal effect, Figures 2 and 3 show the iterative curves of the OCFPA compared with those of the competitors on several functions. The comparative results illustrate that the OCFPA is superior to these advanced intelligent algorithms in terms of convergence speed and accuracy on most functions.

Figures 4 and 5 show the box-and-whisker plots of the optimal solution set of 30 independent runs on unimodal

functions f_{01} , f_{02} , f_{05} , and f_{07} and multimodal functions f_{08} , f_{09} , f_{10} , and f_{11} . We note that our proposed algorithm has fewer outliers and that the dispersion degree is lower than that of the other algorithms, indicating that the proposed OCFPA has excellent stability characteristics.

5.6. Nonparametric Statistical Analysis. In this section, the Wilcoxon sign-rank test [38, 39], a nonparametric statistic technique, is introduced for pairwise comparison of the significant difference of OCFPA against other state-of-the-art competitors on the final results over 30 independent runs. The significance level α is fixed at 0.05. In Table 10, the p value is compared with α to evaluate the significance of the results; R^+ denotes the sum of ranks for which OCFPA outperforms other competitors, and R^- indicates the inverse relation; “+,” “=,” and “−” denote that OCFPA is superior, equivalent, or inferior to the compared algorithm, respectively, and “ $w/t/l$ ” is used to count the number of OCFPA wins on w , ties on t , and loses on l functions against the corresponding competitor.

From Table 10, we observe that, for “CS versus OCFPA,” “GSA versus OCFPA,” and “SGHS versus OCFPA,” the p values are much smaller than 0.05 on 13 functions. For “CMAES versus OCFPA” and “OXDE versus OCFPA,” CMAES and OXDE surpass OCFPA only on f_{04} . For “CLPSO versus OCFPA,” OCFPA is inferior to CLPSO only on f_{13} and f_{14} . In summary, from the “ $w/t/l$ ” results, we can conclude that

TABLE 8: Comparison of OCFPA with evolutionary algorithms on unimodal functions ($D = 30$).

f		CS	GSA	CMAES	OXDE	SGHS	CLPSO	OCFPA
f_{01}	Mean	$6.64E - 32$	$1.27E - 16$	$1.57E - 29$	$3.32E - 61$	$1.47E - 10$	$2.61E - 40$	$1.09E - 86$
	Std	$(1.41E - 31)$	$(4.04E - 17)$	$(2.22E - 30)$	$(8.22E - 61)$	$(8.53E - 11)$	$(3.43E - 40)$	$(3.27E - 86)$
	Rank	4	6	5	2	7	3	1
f_{02}	Mean	$1.23E - 01$	$6.10E - 02$	$3.97E - 02$	$1.17E - 02$	$7.20E - 03$	$3.90E - 03$	$1.70E - 03$
	Std	$(5.79E - 02)$	$(1.50E - 02)$	$(1.32E - 02)$	$(7.90E - 03)$	$(3.40E - 03)$	$(1.10E - 03)$	$(9.08E - 04)$
	Rank	7	5	6	4	3	2	1
f_{03}	Mean	$1.16E - 27$	$3.33E + 03$	$5.73E - 24$	$2.41E - 58$	$2.97E - 02$	$1.42E - 38$	$2.09E - 81$
	Std	$(3.92E - 27)$	$(2.77E + 03)$	$(4.72E - 25)$	$(4.68E - 58)$	$(1.27E - 01)$	$(1.65E - 38)$	$(7.67E - 81)$
	Rank	4	7	5	2	6	3	1
f_{04}	Mean	$2.70E + 01$	$2.29E + 01$	$2.65E - 01$	$7.97E - 01$	$9.23E + 01$	$2.93E + 01$	$1.08E + 01$
	Std	$(3.35E + 01)$	$(2.34E + 01)$	$(1.01E + 00)$	$(1.62E + 00)$	$(2.34E + 02)$	$(2.72E + 01)$	$(7.11E + 00)$
	Rank	5	4	1	2	7	6	3
f_{05}	Mean	$7.47E - 20$	$5.32E - 08$	$1.38E - 02$	$1.11E - 37$	$4.61E - 05$	$7.89E - 26$	$2.38E - 47$
	Std	$(3.42E - 19)$	$(7.64E - 09)$	$(5.85E - 02)$	$(1.80E - 37)$	$(1.17E - 05)$	$(5.70E - 26)$	$(6.13E - 47)$
	Rank	4	5	7	2	6	3	1
f_{06}	Mean	0	0	0	0	0	0	0
	Std	(0)	(0)	(0)	(0)	(0)	(0)	(0)
	Rank	1	1	1	1	1	1	1
f_{07}	Mean	$4.69E - 33$	$1.18E - 15$	$2.90E - 28$	$7.03E - 60$	$5.09E - 04$	$2.54E - 41$	$1.33E - 87$
	Std	$1.58E - 32$	$4.09E - 16$	$3.59E - 29$	$3.79E - 59$	$2.80E - 03$	$3.55E - 41$	$4.61E - 87$
	Rank	4	6	5	2	7	3	1

FIGURE 3: Iterative curves on f_{08} , f_{09} , f_{10} , and f_{11} .

TABLE 9: Comparison of OCFPA with evolutionary algorithms on multimodal functions ($D = 30$).

f		CS	GSA	CMAES	OXDE	SGHS	CLPSO	OCFPA
f_{08}	Mean	-11461	-2427.8	-7045.7	-12514	-12547	-12542	-12569
	Std	(422.65)	(515.24)	(641.36)	(103.09)	(124.88)	(5.09E + 01)	(4.05E - 12)
	Rank	5	7	6	4	2	3	1
f_{09}	Mean	1.53E + 01	3.25E + 01	2.38E + 02	1.65E + 01	1.33E - 01	3.32E - 02	0
	Std	(5.21E + 00)	(9.15E + 00)	(3.81E + 01)	(5.13E + 01)	(4.32E - 01)	(1.82E - 01)	(0)
	Rank	4	6	7	5	3	2	1
f_{10}	Mean	1.30E - 02	1.80E - 03	2.46E - 04	1.11E - 02	4.40E - 02	0	0
	Std	(1.73E - 02)	(4.90E - 03)	(1.40E - 03)	(1.76E - 02)	(4.06E - 02)	(0)	(0)
	Rank	6	4	3	5	7	1	1
f_{11}	Mean	5.61E + 00	9.45E - 09	1.94E + 01	9.16E - 01	1.03E - 05	9.77E - 15	4.32E - 15
	Std	(1.97E + 00)	(1.70E - 09)	(2.27E - 01)	(8.67E - 01)	(3.31E - 06)	(2.91E - 15)	(6.49E - 16)
	Rank	6	3	7	5	4	2	1
f_{12}	Mean	1.80E + 01	4.41E + 01	2.43E + 02	2.19E + 01	2.67E - 01	2.00E - 01	0
	Std	(7.37E + 00)	(9.39E + 00)	(4.53E + 01)	(5.54E + 00)	(6.92E - 01)	(4.07E - 01)	(0)
	Rank	4	6	7	5	3	2	1
f_{13}	Mean	1.04E - 02	3.50E - 03	9.27E - 01	1.64E - 32	2.29E - 12	1.57E - 32	1.61E - 32
	Std	3.16E - 02	1.89E - 02	9.96E - 01	1.14E - 33	2.64E - 12	5.56E - 48	1.18E - 33
	Rank	6	5	7	3	4	1	2
f_{14}	Mean	4.77E - 02	1.38E - 17	6.05E - 30	3.79E - 32	2.94E - 11	1.50E - 33	2.09E - 32
	Std	(1.10E - 01)	(3.78E - 18)	(7.91E - 31)	(1.51E - 31)	(2.22E - 11)	(0)	(4.56E - 32)
	Rank	7	5	4	3	6	1	2

FIGURE 4: Box-and-whisker plot on f_{01} , f_{02} , f_{05} , and f_{07} .

TABLE 10: Statistical analysis using the Wilcoxon sign-rank test.

f	CS versus OCFPA				GSA versus OCFPA				CMAES versus OCFPA			
	p value	R^+	R^-	Win	p value	R^+	R^-	Win	p value	R^+	R^-	Win
f_{01}	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
f_{02}	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
f_{03}	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
f_{04}	$1.09E-01$	310	155	+	$4.44E-05$	431	34	+	$1.73E-06$	0	465	-
f_{05}	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
f_{06}	1	232.5	232.5	=	1	232.5	232.5	=	1	232.5	232.5	=
f_{07}	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
f_{08}	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
f_{09}	$1.72E-06$	465	0	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
f_{10}	$2.90E-04$	419.5	45.5	+	$6.79E-02$	289.5	175.5	+	$3.17E-01$	247.5	217.5	=
f_{11}	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
f_{12}	$1.72E-06$	465	0	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
f_{13}	$4.49E-06$	455.5	9.5	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
f_{14}	$3.07E-06$	459	6	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
$w/t/l$	13/1/0				13/1/0				11/2/1			
f	OXDE versus OCFPA				SGHS versus OCFPA				CLPSO versus OCFPA			
	p value	R^+	R^-	Win	p value	R^+	R^-	Win	p value	R^+	R^-	Win
f_{01}	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
f_{02}	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+	$1.92E-06$	464	1	+
f_{03}	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
f_{04}	$7.06E-06$	15	450	-	$1.47E-04$	417	48	+	$2.10E-03$	382	83	+
f_{05}	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
f_{06}	1	232.5	232.5	=	1	232.5	232.5	=	1	232.5	232.5	=
f_{07}	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+
f_{08}	$5.90E-03$	252	213	+	$2.10E-06$	464.5	0.5	+	$1.04E-02$	230	235	=
f_{09}	$1.73E-06$	465	0	+	$1.73E-06$	465	0	+	$3.17E-01$	247.5	217.5	=
f_{10}	$4.29E-04$	412.5	52.5	+	$1.73E-06$	465	0	+	1	232.5	232.5	=
f_{11}	$8.00E-05$	437.5	27.5	+	$1.73E-06$	465	0	+	$6.83E-07$	465	0	+
f_{12}	$1.64E-06$	465	0	+	$1.73E-06$	465	0	+	$1.43E-02$	315	150	+
f_{13}	$2.37E-01$	282	183	=	$1.73E-06$	465	0	+	$4.14E-04$	52.5	412.5	-
f_{14}	$8.87E-01$	223.5	241.5	=	$1.73E-06$	465	0	+	$2.20E-03$	85.5	379.5	-
$w/t/l$	10/3/1				13/1/0				8/4/2			

our proposed algorithm shows excellent or at least higher competitive performance than the advanced evolutionary algorithms.

6. Conclusion

We offer a novel variant of the FPA by combining the OL strategy and catfish effect mechanism to solve complex optimization problems. The OCFPA uses the OL strategy to improve the local search ability of the basic FPA by sharing and exchanging useful information at each dimension from various solution vectors. Furthermore, the catfish effect mechanism is introduced to force the FPA to

escape the local optima and improve the search space of the population to explore more promising regions. With the aid of these two strategies, the OCFPA offers a favorable balance of exploration and exploitation ability. The experimental results on benchmark functions demonstrate the effectiveness and efficiency of our proposed algorithms.

In future work, although the OL strategy has been successfully incorporated into the FPA, the potential of the OL method should be further exploited by acting on different information sources. In addition, we plan to apply the OL-based FPA to real-world engineering problems, such as parameter identification and feature selection.

FIGURE 5: Box-and-whisker plot on f_{08} , f_{09} , f_{10} , and f_{11} .

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] G.-G. Wang, S. Deb, A. H. Gandomi, and A. H. Alavi, "Opposition-based krill herd algorithm with Cauchy mutation and position clamping," *Neurocomputing*, vol. 177, pp. 147–157, 2016.
- [2] M. Battarra, S. Benedettini, and A. Roli, "Leveraging saving-based algorithms by masterslave genetic algorithms," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 4, pp. 555–566, 2011.
- [3] K. Kameyama, "Particle swarm optimization—a survey," *IEICE Transaction on Information and Systems*, vol. 92, no. 7, pp. 1354–1361, 2009.
- [4] N. Karaboga, "Digital IIR filter design using differential evolution algorithm," *EURASIP Journal on Applied Signal Processing*, vol. 2005, Article ID 856824, 2005.
- [5] I. Perez, M. Gomez-Gonzalez, and F. Jurado, "Estimation of induction motor parameters using shuffled frog-leaping algorithm," *Electrical Engineering*, vol. 95, no. 3, pp. 267–275, 2013.
- [6] P. K. Roy and D. Mandal, "Quasi-oppositional biogeography-based optimization for multi-objective optimal power flow," *Electric Power Components and Systems*, vol. 40, no. 2, pp. 236–256, 2011.
- [7] X.-S. Yang and S. Deb, "Multiobjective cuckoo search for design optimization," *Computers & Operations Research*, vol. 40, no. 6, pp. 1616–1624, 2013.
- [8] E. Fattahi, M. Bidar, and H. R. Kanan, "Fuzzy Krill Herd (FKH): An improved optimization algorithm," *Intelligent Data Analysis*, vol. 20, no. 1, pp. 153–165, 2016.
- [9] L. Wu, C. Zuo, H. Zhang, and Z. Liu, "Bimodal fruit fly optimization algorithm based on cloud model learning," *Soft Computing*, vol. 21, no. 7, pp. 1877–1893, 2017.
- [10] R. Dou and H. Duan, "Lévy flight based pigeon-inspired optimization for control parameters optimization in automatic carrier landing system," *Aerospace Science and Technology*, vol. 61, pp. 11–20, 2017.
- [11] S. Peng, A.-J. Ouyang, and J. J. Zhang, "An adaptive invasive weed optimization algorithm," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 2, Article ID 1559004, 2015.
- [12] F. Xue, Y. Cai, Y. Cao, Z. Cui, and F. Li, "Optimal parameter settings for bat algorithm," *International Journal of Bio-Inspired Computation*, vol. 7, no. 2, pp. 125–128, 2015.
- [13] X.-S. Yang, M. Karamanoglu, and X. He, "Flower pollination algorithm: a novel approach for multiobjective optimization," *Engineering Optimization*, vol. 46, no. 9, pp. 1222–1237, 2014.
- [14] E. Nabil, "A modified flower pollination algorithm for global optimization," *Expert Systems with Applications*, vol. 57, pp. 192–203, 2016.
- [15] N.-D. Hoang, D. Tien Bui, and K.-W. Liao, "Groutability estimation of grouting processes with cement grouts using Differential Flower Pollination Optimized Support Vector Machine," *Applied Soft Computing*, vol. 45, pp. 173–186, 2016.
- [16] Y. Zhou, R. Wang, and Q. Luo, "Elite opposition-based flower pollination algorithm," *Neurocomputing*, vol. 188, pp. 294–310, 2016.

- [17] R. Salgotra and U. Singh, "Application of mutation operators to flower pollination algorithm," *Expert Systems with Applications*, vol. 79, pp. 112–129, 2017.
- [18] A. Y. Abdelaziz, E. S. Ali, and S. M. Abd Elazim, "Combined economic and emission dispatch solution using Flower Pollination Algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 80, pp. 264–274, 2016.
- [19] R. Wang, Y. Zhou, S. Qiao, and K. Huang, "Flower pollination algorithm with bee pollinator for cluster analysis," *Information Processing Letters*, vol. 116, no. 1, pp. 1–14, 2016.
- [20] W. Zhang, Z. Qu, K. Zhang, W. Mao, Y. Ma, and X. Fan, "A combined model based on CEEMDAN and modified flower pollination algorithm for wind speed forecasting," *Energy Conversion and Management*, vol. 136, pp. 439–451, 2017.
- [21] S. Xu, Y. Wang, and F. Huang, "Optimization of multi-pass turning parameters through an improved flower pollination algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 89, no. 1–4, pp. 503–514, 2017.
- [22] R. Salgotra and U. Singh, "A novel bat flower pollination algorithm for synthesis of linear antenna arrays," *Neural Computing and Applications*, pp. 1–14, 2016.
- [23] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.
- [24] J. Wang, B. Zhou, and S. Zhou, "An Improved Cuckoo Search Optimization Algorithm for the Problem of Chaotic Systems Parameter Estimation," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 2959370, 8 pages, 2016.
- [25] L.-Y. Chuang, S.-W. Tsai, and C.-H. Yang, "Chaotic catfish particle swarm optimization for solving global numerical optimization problems," *Applied Mathematics and Computation*, vol. 217, no. 16, pp. 6900–6916, 2011.
- [26] L.-Y. Chuang, S.-W. Tsai, and C.-H. Yang, "Catfish particle swarm optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '08)*, IEEE, St. Louis, Mo, USA, September 2008.
- [27] S. G. de-los-Cobos-Silva, M. A. Gutierrez-Andrade, R. A. Mora-Gutierrez et al., "An efficient algorithm for unconstrained optimization," *Mathematical Problems in Engineering*, vol. 2015, Article ID 178545, 17 pages, 2015.
- [28] S. Tuo, L. Yong, F. Deng, Y. Li, Y. Lin, and Q. Lu, "HSTLBO: A hybrid algorithm based on Harmony Search and Teaching-Learning- Based Optimization for complex highdimensional optimization problems," *PLoS ONE*, vol. 12, no. 4, Article ID e0175114, 2017.
- [29] W.-F. Gao, S.-Y. Liu, and L.-L. Huang, "A novel artificial bee colony algorithm based on modified search equation and orthogonal learning," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 1011–1024, 2013.
- [30] A. Draa, "On the performances of the flower pollination algorithm—qualitative and quantitative analyses," *Applied Soft Computing*, vol. 34, pp. 349–371, 2015.
- [31] J. Niu, W. Zhong, Y. Liang, N. Luo, and F. Qian, "Fruit fly optimization algorithm based on differential evolution and its application on gasification process operation optimization," *Knowledge-Based Systems*, vol. 88, pp. 253–263, 2015.
- [32] L. Wu, C. Zuo, and H. Zhang, "A cloud model based fruit fly optimization algorithm," *Knowledge-Based Systems*, vol. 89, pp. 603–617, 2015.
- [33] A. Yadav and K. Deep, "Constrained optimization using gravitational search algorithm," *National Academy of Science Letters*, vol. 36, no. 5, pp. 527–534, 2013.
- [34] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [35] Y. Wang, Z. Cai, and Q. Zhang, "Enhancing the search ability of differential evolution through orthogonal crossover," *Information Sciences*, vol. 185, pp. 153–177, 2012.
- [36] Q.-K. Pan, P. N. Suganthan, M. F. Tasgetiren, and J. J. Liang, "A self-adaptive global best harmony search algorithm for continuous optimization problems," *Applied Mathematics and Computation*, vol. 216, no. 3, pp. 830–848, 2010.
- [37] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [38] 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 Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
- [39] D. Shilane, J. Martikainen, S. Dudoit, and S. J. Ovaska, "A general framework for statistical performance comparison of evolutionary computation algorithms," *Information Sciences*, vol. 178, no. 14, pp. 2870–2879, 2008.

Submit your manuscripts at
www.hindawi.com