

Research Article

Modification of Fish Swarm Algorithm Based on Lévy Flight and Firefly Behavior

Zhenrui Peng , Kangli Dong , Hong Yin, and Yu Bai

School of Mechatronic Engineering, Lanzhou Jiaotong University, Lanzhou, China

Correspondence should be addressed to Zhenrui Peng; pzrui@163.com

Received 11 June 2018; Revised 18 July 2018; Accepted 9 August 2018; Published 13 September 2018

Academic Editor: Raşit Köker

Copyright © 2018 Zhenrui Peng et al. 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.

Artificial fish swarm algorithm easily converges to local optimum, especially in solving the global optimization problem of multidimensional and multiextreme value functions. To overcome this drawback, a novel fish swarm algorithm (LFFSA) based on Lévy flight and firefly behavior is proposed. LFFSA incorporates the moving strategy of firefly algorithm into two behavior patterns of fish swarm, i.e., chasing behavior and preying behavior. Furthermore, Lévy flight is introduced into the searching strategy. To limit the search band, nonlinear view and step size based on dynamic parameter are considered. Finally, the proposed algorithm LFFSA is validated with several benchmark problems. Numerical results demonstrate that LFFSA has a better performance in convergence speed and optimization accuracy than the other test algorithms.

1. Introduction

Optimization plays an important role in modern industry as well as in scientific world. Due to the computational costs of the existing numerical methods, researchers have to rely on metaheuristic algorithms to solve complex optimization problems.

Artificial fish swarm algorithm (AFSA) was proposed as a metaheuristic algorithm based on fish swarm behaviors—preying behavior, swarming behavior, and chasing behavior [1]. With its simple principle and such good features as robustness and tolerance of parameter setting, AFSA has become an increasingly important tool in swarm intelligence optimization. Some algorithms derived from AFSA have been presented: Wang et al. eliminated the step restriction and added new leaping behavior to improve the stability of the algorithm [2]. Another modified form of AFSA used particle swarm optimization to reformulate AFSA, integrated AFSA with communication behavior, and created formulas for major AFSA parameters [3]. Luan et al. adopted normal distribution function, Cauchy distribution function, multiparent crossover operator, mutation operator, and modified minimal generation gap model to overcome the drawback of slow convergence speed in later

iterations [4]. Another improved AFSA with crossover and culture algorithm was proposed to enhance its optimization efficiency [5]. Hu et al. integrated the merits of the self-adaptation strategy, mutation strategy, and hybrid strategy into the social behaviors of AFSA [6].

Over the last decades, AFSA and its improved algorithms have been successfully applied to various engineering optimization problems. Kumar et al. adopted AFSA to optimize renewable energy sources in a microgrid [7]. By categorizing the social behaviors of fish swarm into foraging, reproductive, and random behaviors, a novel artificial fish swarm algorithm was advocated for solving large-scale reliability-redundancy application problem [8]. A derivative of AFSA was used to solve the multiobjective disassembly line balancing problem with fuzzy disassembly times [9]. A binary fish swarm algorithm was presented to solve profit-based unit commitment problem in generation companies [10].

Firefly algorithm (FA) is another metaheuristic algorithm based on the idealized behavior of the flashing characteristics of fireflies [11]. FA can adaptively adjust the radius of the induction and parallel search the optimum in multiple peaks. FA has natural advantages in solving multimodal optimization problem. Some drawbacks of searching strategy and

parameters in FA were targeted and improved by researchers. A derivative of FA with directed movement of fireflies was proposed by Farahani et al. [12]. Chaotic sequence was introduced into the basic FA to adjust parameters γ and α by Coelho et al. [13]. Eagle Strategy, which combines Lévy flight search with FA, was introduced by Yang and Deb [14]. Elitist FA was presented [15], which tried to enhance the best solution position by generating m uniform random vectors and moving in the direction of best solution.

Firefly algorithm and its modified forms were also successfully applied to numerous practical problems. Jagatheesan et al. used firefly algorithm to design a controller for an automatic generation control of multiarea power thermal systems [16]. An FA-inspired band selection and optimized extreme learning machine were proposed for hyperspectral image classification [17]. A self-adaptive firefly algorithm was developed for placement of FACTS devices [18]. Teshome et al. modified FA to counteract some inherent problems that may hinder the performance of the maximum power point tracking [19]. Alb et al. used FA to solve a shielding/shunting electromagnetic problem [20]. Mishra et al. proposed a method for optimal placement of interline power flow controller by using FA [21]. Other applications of improved FA include image compression [22], financial forecasting [23], image segmentation [24], structural optimization [25], classification problem [26], unconstrained optimization [27], economic dispatch problems [28], clustering [29], image retrieval [30], and mechanical optimal design [31]. Some researchers presented comprehensive reviews of existing FA and its modified forms to encourage new researchers to employ FA for solving their own problems [32, 33].

At the same time, numerous studies have shown that Lévy flight is similar to flight characteristics of given animals and insects [34–36], which has been widely used in swarm intelligence algorithms. Subsequently, Lévy flight has been applied to optimization, and preliminary results showed its potential capabilities. Jensi and Jiji proposed an enhanced particle swarm optimization with Lévy flight [37]. Tang et al. proposed a new framework of shuffled frog-leaping algorithm based on the exploration and exploitation mechanism by using Lévy flight [38]. Yahya and Saka proposed a multiobjective artificial bee colony algorithm with Lévy flight and applied it to construction site layout planning [39].

Hybridization is recognized to be an important aspect of high performing algorithms in recent years [40]. Owing to some drawbacks of traditional AFSA and FA, they are not suitable for solving highly nonlinear and multimodal problems. By integrating the merits of AFSA with Lévy flight and FA, this paper proposes a novel hybrid algorithm, named LFFSA (fish swarm algorithm based on Lévy flight and firefly behavior), for global optimization. The highlights of the new algorithm are as follows:

- (i) Attraction degree is involved in the definition of artificial fish
- (ii) Lévy flight is used to adjust the search route of artificial preying fishes

- (iii) By analyzing the relationship between swarming behavior and chasing behavior, unnecessary behavior (swarming behavior) is excluded instead of improving AFSA through adding new behaviors [3–6]
- (iv) Time complexity of the improved algorithm is also further analyzed to demonstrate the effectiveness of the improvement

The remainder of this paper is organized as follows: Section 2 describes the basic AFSA, FA, and Lévy flight respectively; Section 3 proposes and explains LFFSA algorithm in details; in Section 4, the superiority of proposed algorithm LFFSA is validated by several benchmark problems; Section 5 outlines the conclusion.

2. Background

2.1. Artificial Fish Swarm Algorithm. AFSA is a swarm intelligence algorithm, which can be employed to solve the optimization problem by imitating swarming, chasing, and preying behaviors of artificial fishes [1]. As shown in Figure 1, let X_i be the current position of one artificial fish, X_v be the viewpoint of artificial fish at one moment, Visual be the visual scope of each individual, X_a and X_b be fishes within the Visual of X_i , Step be the biggest step of artificial fish, and δ be the congestion factor of fish swarm. The food concentration is proportional to the fitness function $f(X)$. The behavior patterns of fish swarms can be described as follows.

Swarming behavior: if $f(X_c) > f(X_i)$, where X_c is the central point inside the Visual of the point X_i , swarming behavior is to be executed. Take X_c as X_v . The fish at X_i will take a step toward the point X_c .

Chasing behavior: if the point (denoted by X_{\max}) having the best objective function value inside the Visual satisfies $f(X_{\max}) > f(X_i)$, and if the Visual of X_i is not crowded, chasing behavior is to be executed. Take X_{\max} as X_v . The fish at X_i will take a step toward the point X_{\max} .

Preying behavior: preying behavior is tried in the following situations:

- (1) $f(X_c) < f(X_i)$, $f(X_{\max}) < f(X_i)$, and the Visual is not crowded
- (2) The Visual is crowded

Here, a point X_j inside the Visual of X_i is randomly selected. If $f(X_j) > f(X_i)$, the preying behavior is to be executed. Take X_j as X_v . The fish at X_i will take a step toward the point X_j . Otherwise, it will move a step randomly within its Visual.

The best solution obtained in each iteration is marked as “board.” After the specified iterations, search process is terminated and the result on the “board” is regarded as the final solution.

For artificial preying fishes, the position-updating can be formulated as

$$X_{\text{next}} = X_i + \text{rand} \cdot \frac{\text{step} \times (X_j - X_i)}{\text{norm}(X_j - X_i)}. \quad (1)$$

where X_{next} is the next position of artificial fish; X_i is the current position of artificial fish; X_j is the position which has

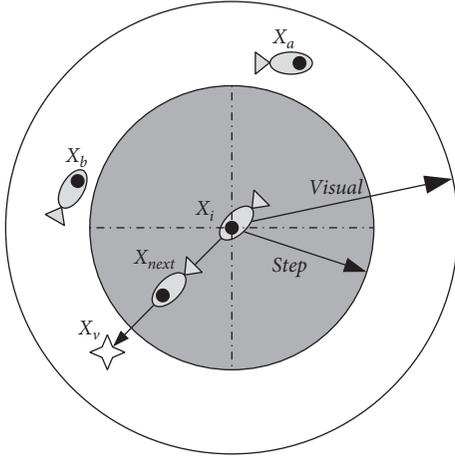


FIGURE 1: Vision concept of the artificial fish.

a better objective function value; rand is a random number in $[-1, 1]$; and $\text{norm}(X_j - X_i)$ is the distance between two position vectors.

For artificial swarming fishes, the position-updating can be formulated as

$$X_{\text{next}} = X_i + \text{rand} \cdot \frac{\text{step} \times (X_c - X_i)}{\text{norm}(X_c - X_i)}. \quad (2)$$

For artificial chasing fishes, the position-updating can be formulated as

$$X_{\text{next}} = X_i + \text{rand} \cdot \frac{\text{step} \times (X_{\text{max}} - X_i)}{\text{norm}(X_{\text{max}} - X_i)}. \quad (3)$$

The flowchart of AFSA is shown in Figure 2.

2.2. Firefly Algorithm. Firefly algorithm (FA) [11] is another swarm intelligence algorithm. It achieves swarming phenomenon by using the fluorescent signal between two firefly individuals.

The attraction between fireflies depends on their light intensities and attraction degree. The light intensity is in proportion to the objective function value of firefly's position. Attraction degree is in proportion to the light intensity. The brighter the light intensity is, the higher attraction degree will be. Besides, the farther the distance is, the lower the light intensity and attraction degree will be.

In the simplest form, the light intensity $I(r)$ varies with the distance r monotonically and exponentially as

$$I(r) = I_0 e^{-\gamma r}. \quad (4)$$

where I_0 is the original light intensity and γ is the light absorption coefficient. As firefly attraction degree is proportional to the light intensity seen by adjacent fireflies, the attraction degree beta of a firefly can be defined as

$$\text{beta} = \text{beta}_0 \cdot e^{-\gamma r^2}, \quad (5)$$

where beta_0 is the attraction degree at $r = 0$.

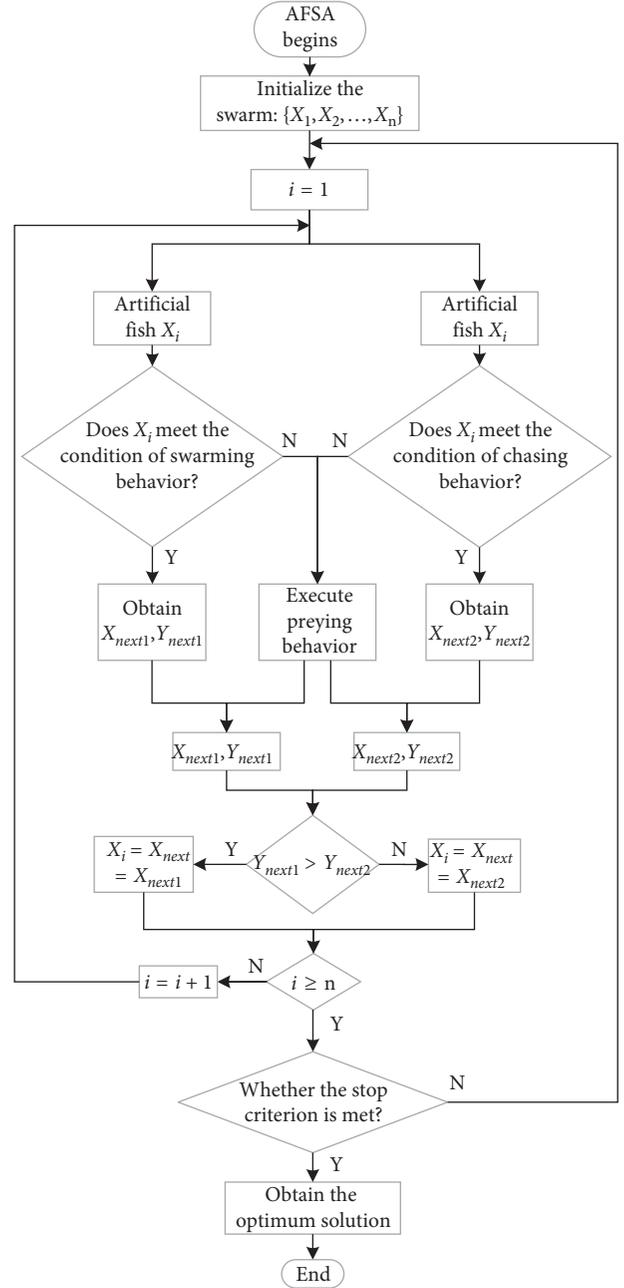


FIGURE 2: Flowchart of AFSA.

The distance r_{ij} between any two fireflies i and j at x_i and x_j , respectively, is the Cartesian distance, which is calculated as

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{d=1}^n (x_{i,d} - x_{j,d})^2}, \quad (6)$$

where n is the dimensionality of the given problem.

2.3. Lévy Flight. Lévy flight is one kind of random searching strategy [35]. Flying step satisfies a heavy-tailed Lévy distribution, which can be represented by a clear power-law equation as

$$L(s) \sim |s|^{1-\beta}, \quad (7)$$

where s is random Lévy step. For searching problems inside a wide range of unknown space, the variance of Lévy movement increases faster than the dimensional Brownian movement.

To some extent, the foraging behavior of nature animals is a kind of random movement behavior. Because next movement usually depends on the current position and the probability of moving to next position, the effectiveness of each random movement becomes greatly important. Recent studies show that Lévy flight is one of the best searching strategies in random movement model [35, 41–43].

3. Fish Swarm Algorithm Based on Lévy Flight and Firefly Behavior

AFSA has several disadvantages in solving nonlinear and multimodal problems. Firstly, AFSA uses swarming behavior and chasing behavior to execute parallel search in a simple and fast way. However, after determining the direction, each artificial fish moving with random step will be unable to approach the target point effectively. Secondly, artificial fish will execute preying behavior when it does not meet the conditions of swarming behavior and chasing behavior. This kind of searching strategy is inefficient and can easily miss the optimum point. Thirdly, too many moving patterns can increase the algorithm complexity, which may cause slow convergence speed.

The above-listed drawbacks of AFSA are improved in the proposed LFFSA. FA has the unique moving strategy using attraction between fireflies, which can be used to fix the random moving after determining the direction in AFSA. And, the preying behavior can be improved using Lévy flight to specify the behavior of artificial fishes. In LFFSA, attraction degree is involved in the definition of artificial fishes, which allows each individual move according to attraction degree; Lévy flight is also considered in the definition of the artificial preying fish to avoid falling into the local optimum; chasing behavior is excluded to decrease the algorithm complexity.

The flowchart of LFFSA is shown in Figure 3. Pseudo code of LFFSA is described in Algorithm 1. To show the difference between AFSA and LFFSA vividly, mechanisms of both algorithms are provided in Figure 4.

The main improvements of LFFSA are summarized as follows:

- (a) *Improvement 1*: FA-based moving strategy. Attraction degree is involved in the definition of artificial fishes, which can be formulated as

$$\text{beta}_{ij} = \text{beta}_0 \cdot e^{-\gamma r_{ij}}, \quad (8)$$

where r_{ij} is the Cartesian distance between artificial fishes i and j given by Equation (6); γ is the light intensity coefficient, which can be set as a constant; and beta_0 , the largest attraction degree, is attraction degree of an artificial fish at $r_{ij} = 0$.

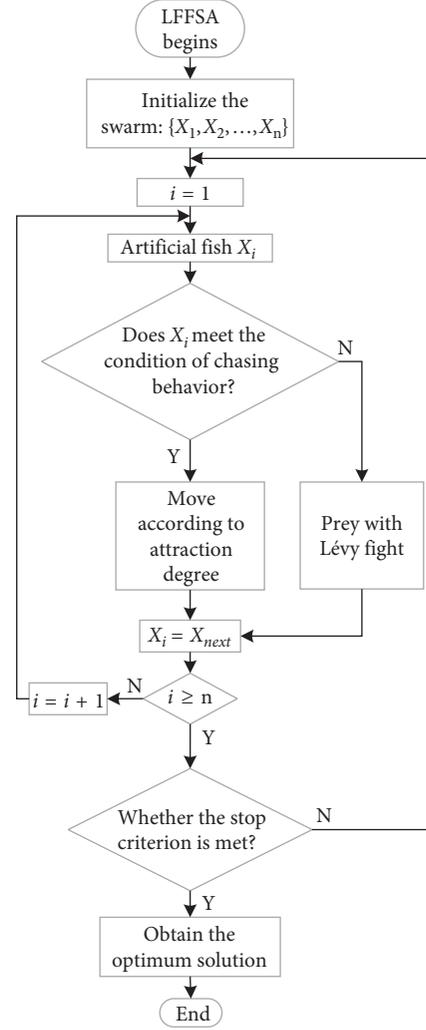


FIGURE 3: Flowchart of LFFSA.

The position-updating with preying behavior established by attraction degree can be formulated as

$$X_{\text{next}} = X_i + \text{beta}_{ij}(X_j - X_i) + \alpha(\text{rand} - 0.5), \quad (9)$$

where X_i is the current position of artificial fish i ; $\text{beta}_{ij}(X_j - X_i)$ is the attraction degree; α is the step factor which is a constant between 0 and 1; rand is a number chosen randomly in $[-1, 1]$; and $\alpha(\text{rand} - 0.5)$ is to avoid falling into the local optimum.

- (b) *Improvement 2*: inertia weight. A linear inertia weight is added into Equation (9) as

$$X_{\text{next}} = \omega_t X_i + \text{beta}_{ij}(X_j - X_i) + \alpha(\text{rand} - 0.5), \quad (10)$$

$$\omega_t = \omega_{\text{max}} - (\omega_{\text{max}} - \omega_{\text{min}}) \cdot \text{gen}/\text{Maxgen}, \quad (11)$$

where ω_t is the weight size inherited from the last position of an artificial fish; ω_{max} denotes the biggest weight; ω_{min} is the minimum weight; gen represents the current iteration; and Maxgen is the ultimate iteration.

```

(1) Initialization of  $\{X_1, X_2, \dots, X_n\}$ ;
(2) Evaluate fitness;
(3) while the stop criterion is not met do
(4)   while  $i < n$  do
(5)     if  $X_i$  meets the condition of chasing behavior
(6)       Move with attraction degree and get a new  $X_i$  based on Equation (13);
(7)     else
(8)       Prey with Lévy flight and get a new  $X_i$  based on Equations (11) and (14);
(9)     end if
(10)  end while
(11)  Update the “board”;
(12) end while
(13) Obtain the optimum solution
    
```

ALGORITHM 1: LFFSA.

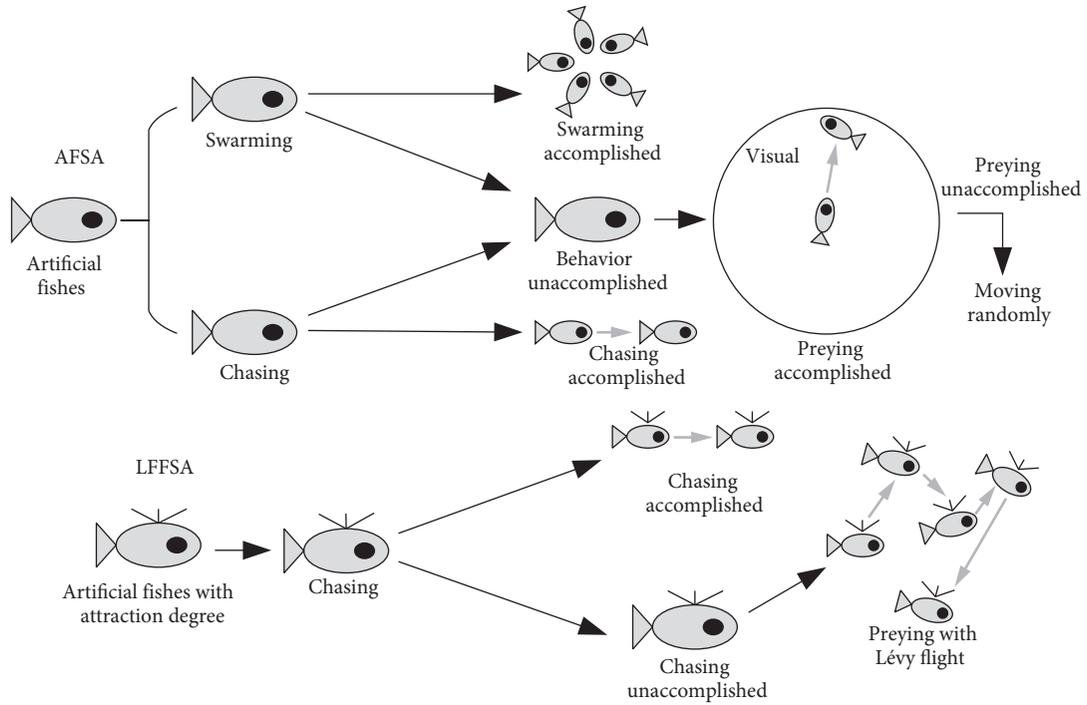


FIGURE 4: Mechanisms of AFSA and LFFSA.

Similarly, the position-updating with chasing behavior can be formulated as

$$X_{\text{next}} = \omega_i X_i + \text{beta}_{i,j} (X_{\text{max}} - X_i) + \alpha (\text{rand} - 0.5), \quad (12)$$

where X_{max} is the position with the highest food concentration in the view of artificial fish X_i and other parameters are defined in Equations (10) and (11).

(c) *Improvement 3: Lévy flight-based search strategy.* Lévy flight is involved in the definition of artificial fishes are executing preying behavior. The movement can be formulated as

$$X_{\text{next}} = X_i + \alpha \oplus L(\lambda), \quad (13)$$

$$L(\lambda) = \frac{\phi \times \mu}{|\nu|^{1/\beta}} (X_i - X_{\text{best}}), \quad (14)$$

where X_i is the current position of artificial fish i ; \oplus is the point to point multiplication; $L(\lambda)$ denotes a random vector generated by Lévy flight; X_{best} represents the best fish on the “board”; $\mu = t^{-\lambda}$, $1 < \lambda < 3$; and μ and ν have the standard normal distribution $\mu \sim N(0, \phi^2)$, $\nu \sim N(0, 1)$, respectively, where

$$\phi = \left\{ \frac{\Gamma(1 + \beta) \sin(\pi\beta/2)}{\Gamma[(1 + \beta)/2] \beta \cdot 2^{(\beta-1)/2}} \right\}^{1/\beta}, \quad (15)$$

where Γ is the standard Gamma function.

(d) *Improvement 4: nonlinear visual and step.* Visual and step change nonlinearly and dynamically in LFFSA. The updating equations are as follows:

$$\text{Visual} = \rho \cdot \text{Visual} + \text{Visual}_{\text{min}}, \quad (16)$$

$$\text{Step} = \rho \cdot \text{Step} + \text{Step}_{\min}, \quad (17)$$

$$\rho = \exp\left(-30 \times \left(\frac{\text{gen}}{\text{Maxgen}}\right)^s\right), \quad (18)$$

where Visual_{\min} , the minimum visual, takes 0.001; Step_{\min} , the minimum step, takes 0.0002; ρ is a nonlinear weight; and s represents an integer, $s > 1$. Here, $s = 3$. Other parameters are the same as those in Equation (11). The relationship between s and ρ is as shown in Figure 5.

Besides, define the biggest distance between two artificial fishes as

$$\max D = \sqrt{(x_{\max} - x_{\min})^2 \times D}, \quad (19)$$

where, x_{\max} and x_{\min} represent the upper bound and the lower bound of searching range respectively; D denotes the D -dimension searching space. The initial Visual is equal to $\max D$, and initial Step is equal to $\max D/8$. Then Visual and Step change dynamically according to Equations (16)–(18).

- (e) *Improvement 5*: decrease of time complexity. Swarming behavior in AFSA is excluded to decrease the algorithm complexity.

4. Numerical Simulation

4.1. Comparison of Convergence Accuracy. LFFSA is validated by numerical simulations. Fish swarm algorithm with firefly behavior (FFSA), differential evolution (DE) algorithm, self-adaptive differential evolution (jDE) algorithm, and the two basic algorithms (AFSA and FA) are compared. All the algorithms are coded in Matlab 2014b. The operating system is windows 7. Simulation hardware is a PC with 2.50 GHz Inter Core i5 and 2.00 GB Memory.

Parameters shown in Table 1 are determined by trial and error.

The following benchmarks are chosen carefully according to their features. Functions Sphere, Quartic, and Rosenbrock, etc., are simple unimodal problems. Functions Ackley, Rastrigin, and Schwefel, etc., are highly complex multimodal problems with many local minima. Schwefel function has a maximum value and other functions have minimum values. These benchmarks are listed in Table 2 [44, 45]. The solutions of 17 test functions obtained by different algorithms are compared. To compare the convergence speed and accuracy of the algorithms clearly and correctly, all functions are run 50 times for each algorithm, respectively. The results are averaged and plotted in Figure 6.

From Figure 6, LFFSA can avoid local optimum and have better convergence accuracy compared with the other algorithms. For AFSA and FA, the solutions of most functions are unsatisfactory; the DE cannot find ideal solutions of f3, f4, f5, f6, f10, f11, f12, f14, and f15; the jDE has good accuracy while solving some of those functions, e.g. f1, f2, f3, and f8; however, solutions of f4, f10, f11, and f15 obtained by jDE are not so precise; the LFFSA can obtain the ideal accuracy

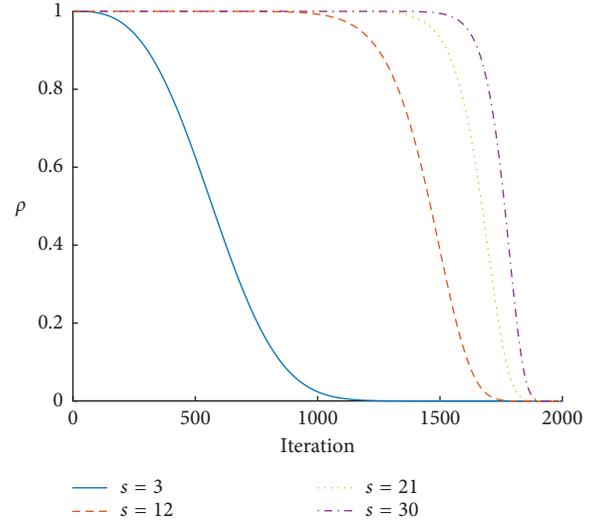


FIGURE 5: Value of ρ .

TABLE 1: Parameter settings.

Algorithms	Parameters	Values
FA, FFSA, LFFSA	β_0	1.0
FA, FFSA, LFFSA	γ	1.0
AFSA, FFSA, LFFSA	δ	0.618
AFSA, FFSA, LFFSA	Trynumber	5
DE, jDE	Scaling constant	0.5
DE, jDE	Crossover constant	0.9
All 6 algorithms	Population	50
All 6 algorithms	Maximum function evaluations (FEs)	2×10^5

for almost all functions, although it cannot achieve a high precision level like solutions of f2 obtained by jDE; the FFSA is slightly worse than LFFSA. The LFFSA outperforms jDE in 10 benchmark functions, while 2 functions are comparative, and 5 functions are worse.

LFFSA is better than AFSA because Lévy flight is able to restrict the movement step of AFSA to a very small area around the current position. Furthermore, the attraction degree guides the fish moving. Besides, LFFSA can quickly lead the fish individual to the close-by optimal point. Considering all the advantages discussed above, the optimum solution can be found successfully by using LFFSA, which outperforms the basic algorithms for all test functions, and outperforms jDE for several functions. To observe the searching capabilities of different algorithms directly, the average, median, best, and worst values obtained by different algorithms are listed in Table 3. Results indicate that LFFSA can find ideal solutions and have a better robustness.

4.2. Computational Complexity Analysis. Time complexity is also an important indicator in the analysis of algorithms. If an algorithm is composed of several parts, then its complexity is the sum of the complexities of these parts. The algorithm may consist of a loop executed many times, and each time is with a different complexity. Time complexity of

TABLE 2: Test functions.

No.	Test functions	Expression	Optimum value	Domain	D
f1	Sphere	$f(x) = \sum_{i=1}^D x_i^2$	0	$(-100, 100)^D$	30
f2	Quartic	$f(x) = \sum_{i=1}^D ix_i^4$	0	$(-1.28, 1.28)^D$	30
f3	Ackley	$f(x) = -20 \exp\left\{-0.2\sqrt{(1/D)\sum_{i=1}^D x_i^2} - \exp[(1/D)\sum_{i=1}^D \cos(2\pi x_i)]\right\} + 20 + e$	0	$(-32.768, 32.768)^D$	30
f4	Rosenbrock	$f(x) = \sum_{i=1}^{D-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	0	$(-2.048, 2.048)^D$	30
f5	Rastrigin1	$f(x) = \sum_{i=1}^D \{x_i^2 - 10 \cos(2\pi x_i) + 10\}$	0	$(-5.12, 5.12)^D$	30
f6	Rastrigin2	$f(x) = \sum_{i=1}^D \{y_i^2 - 10 \cos(2\pi y_i) + 10\}$ $y_i = \begin{cases} x_i, & x_i < (1/2), \\ \text{round}(2x_i)/2, & x_i > (1/2), \end{cases}$	0	$(-5.12, 5.12)^D$	30
f7	Schwefel	$f(x) = \sum_{i=1}^D \{x_i \cdot \sin \sqrt{ x_i }\}$	418.9829D	$(-500, 500)^D$	30
f8	Griewank	$f(x) = (1/4000)\sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i/\sqrt{i}) + 1$	0	$(-600, 600)^D$	30
f9	Quadric	$f(x) = \sum_{i=1}^D (\sum_{j=1}^i x_j)^2$	0	$(-100, 100)^D$	30
f10	Schaffer1	$f(x) = \sum_{i=1}^{D-1} \{((\sin^2 \sqrt{x_{i+1}^2 + x_i^2} - 0.5) / (0.001(x_{i+1}^2 + x_i^2) + 1)^2) + 0.5\}$	0	$(-100, 100)^D$	30
f11	Schaffer2	$f(x) = \sum_{i=1}^{D-1} \left\{ (\sin^2 \sqrt{\sum_{i=1}^D x_i^2} - 0.5) / (0.001(\sum_{i=1}^D x_i^2) + 1)^2 + 0.5 \right\}$	0	$(-100, 100)^D$	30
f12	Maxmod	$f(x) = \max(x_i)$	0	$(-10, 10)^D$	30
f13	Dixon and price	$(x_1 - 1)^2 + \sum_{i=1}^D i(2x_i^2 - x_{i-1})$	0	$(-10, 10)^D$	30
f14	Powell	$f(x) = \sum_{i=1}^{D/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^2 + 10(x_{4i-3} - x_{4i})^4]$	0	$(-4, 5)^D$	28
f15	Zakharov	$f(x) = \sum_{i=1}^D x_i^2 + (\sum_{i=1}^D 0.5ix_i)^2 + (\sum_{i=1}^D 0.5ix_i)^4$	0	$(-5, 10)^D$	30
f16	Sin1	$\sum_{i=1}^D x_i \sin(x_i) + 0.1x_i $	0	$(-10, 10)^D$	30
f17	Sin2	$f(x) = -\sum_{i=1}^D \sin(x_i) \sin^{20}(ix_i^2/\pi)$	-99.2784	$(0, \pi)^D$	100

the algorithm is used to estimate the efficiency of the algorithm. It is defined that the time complexity of the algorithm, or the running time, is $O(f(n))$ [46]. Define N as the population.

In the definition of time complexity, $O(N^2)$ and $O(N)$ are at different levels. If the time complexity of one algorithm is $O(N^2)$, the time complexity of the other one is $O(N)$, then the former algorithm is more complex. In the other case, if the time complexity of one algorithm is $O(N^2)$ while the time complexity of the other one is $O(N^2 + N)$, their complexities are both $O(N^2)$.

The time complexity analysis of AFSA is provided in Table 4.

From Table 4, the time complexity of AFSA is

$$O(\text{Maxgen} * (3 * N^2 + \text{trynumber} * N + 6 * N)). \quad (20)$$

Swarming behavior has N times of calculating congestion factor, 1 time of judging, and 1 time of moving. Therefore, time complexity of swarming behavior is $O(N^2 + 2 * N)$. Chasing behavior has N times of calculating

congestion factor, N times of searching, 1 time of judging, and 1 time of moving. Therefore, time complexity of chasing behavior is $O(2 * N^2 + 2 * N)$.

Time complexity analysis of LFFSA is listed in Table 5.

Due to the lack of swarming behavior, time complexity of LFFSA can be calculated as

$$O(\text{Maxgen} * (2 * N^2 + \text{Trynumber} * N + 4 * N)). \quad (21)$$

We can also obtain time complexity of FA:

$$O(\text{Maxgen} * (N^2 + N)). \quad (22)$$

A conclusion can be obtained that time complexities of the three algorithms are at the same level. Their computational complexities in the worst case are only the square of the training sample size.

4.3. Experimental Complexity Analysis. Time complexity is a rough estimate of time cost. The more accurate time cost of an algorithm can only be validated by running it on

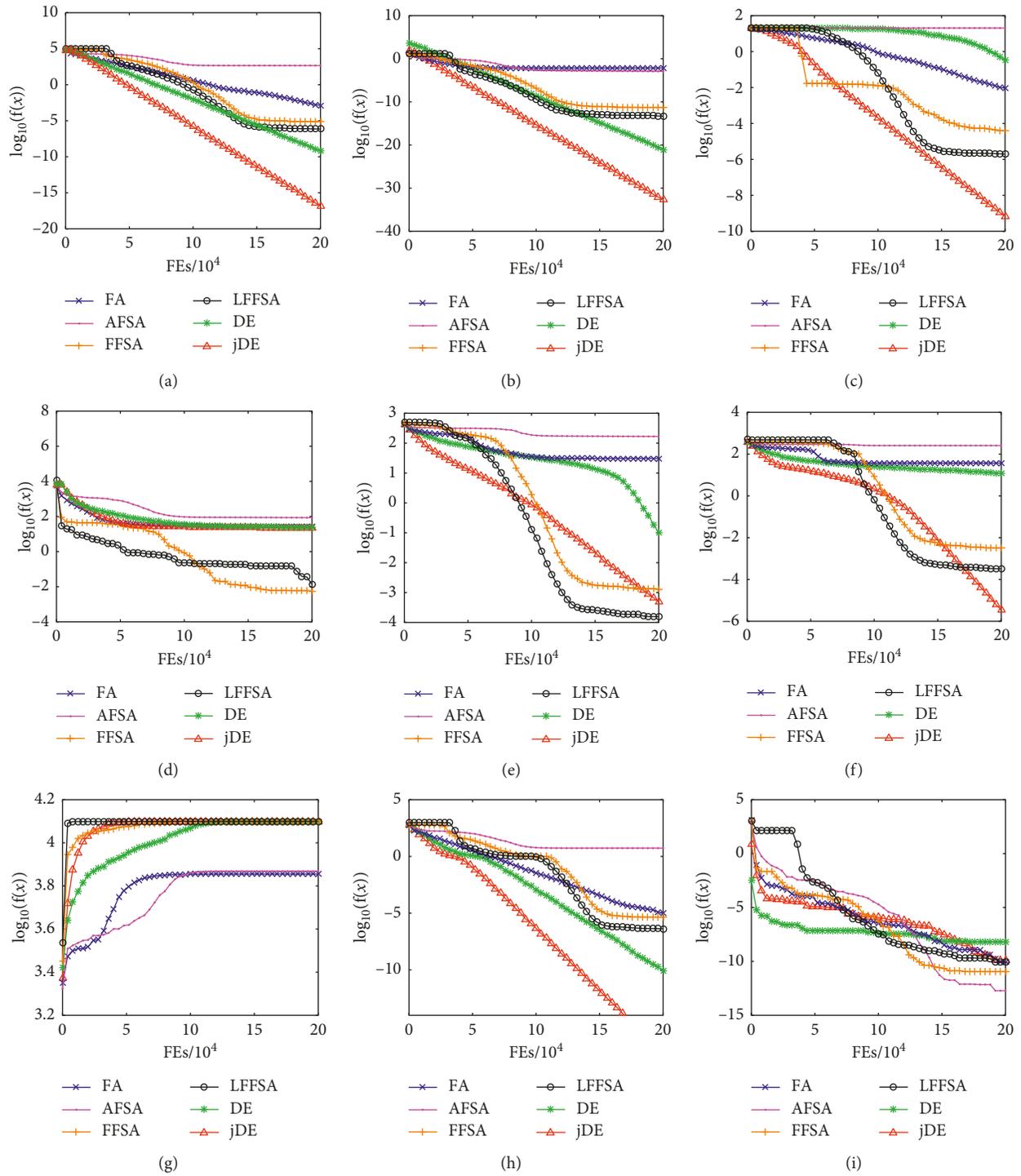


FIGURE 6: Continued.

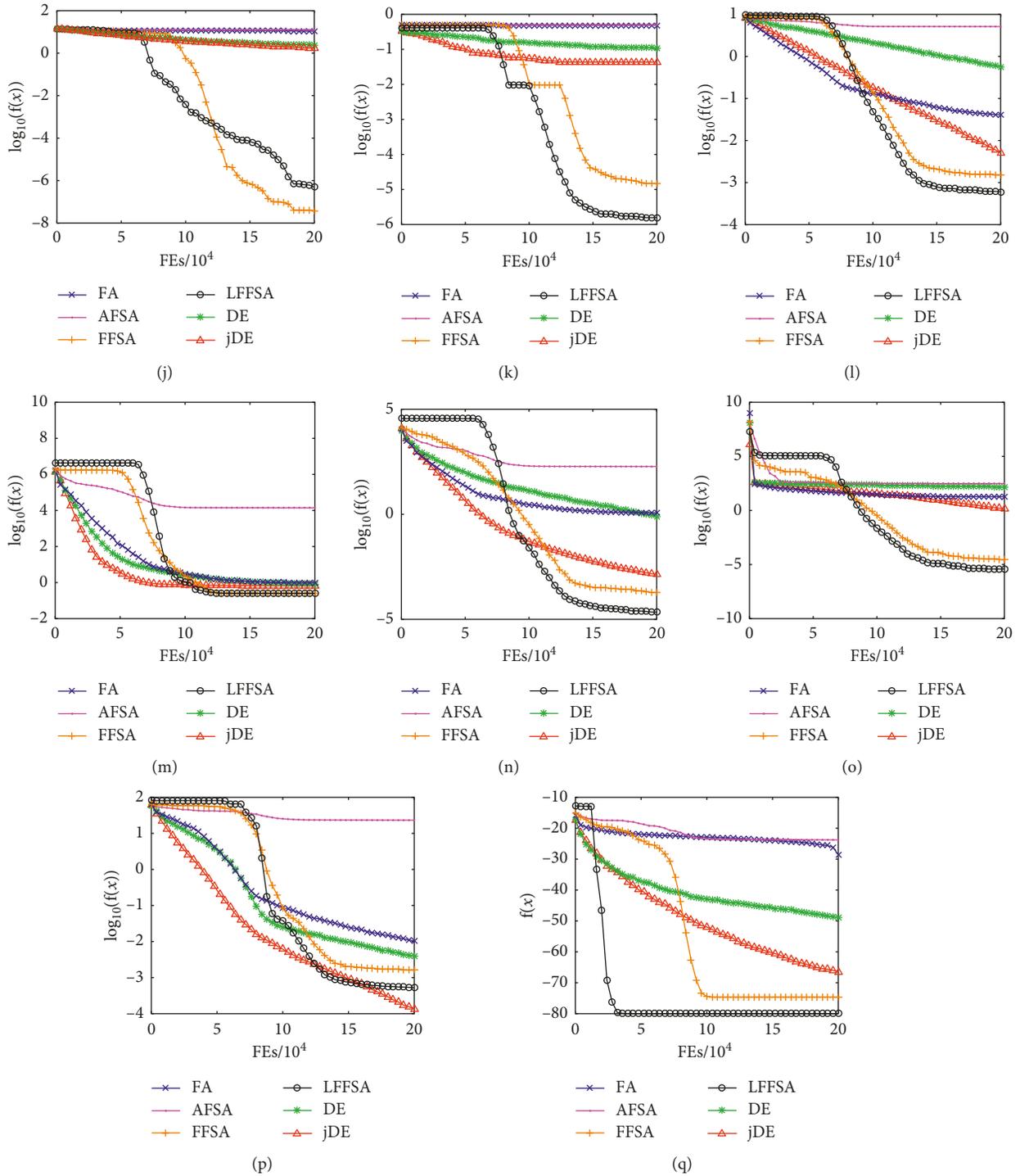


FIGURE 6: Iterative curves of test functions. (a) f1. (b) f2. (c) f3. (d) f4. (e) f5. (f) f6. (g) f7. (h) f8. (i) f9. (j) f10. (k) f11. (l) f12. (m) f13. (n) f14. (o) f15. (p) f16. (q) f7.

computer. Since different algorithms cannot reach the same convergence accuracy, the test with fixed convergence accuracy is not available. Therefore, the test with max function evaluations is conducted. Running time of each algorithm is counted by the explorer of MATLAB. Parameter settings of algorithms are the same in Section 4.1. Average running time of different algorithms is listed in Table 6. When function

evaluations are the same, running speed of LFFSA is faster than that of AFSA, while DE has the fastest running speed. Results are quite in accord with those obtained by computational complexity analysis. LFFSA and jDE are comparative in experimental complexity. Running time of FFSA is almost twice as much as that of LFFSA. The improvement of LFFSA decreases time complexity to some extent.

TABLE 3: Comparison of optimization results.

No.	Items	AFSA	Std.	FA	Std.	FFSA	Std.	LFFSA	Std.	DE	Std.	jDE	Std.
f1	Worst	3.014		-2.798		-5.132		-6.743		-8.544		-15.021	
	Best	2.402	0.181	-3.257	0.120	-5.233	0.071	-8.500	0.071	-10.371	0.132	-17.242	0.121
	Average	2.656		-3.048		-5.145		-7.278		-8.924		-16.326	
	Median	2.651		-3.102		-5.193		-6.325		-8.586		-16.706	
f2	Worst	-2.322		-1.9245		-11.178		-14.207		-20.506		-31.644	
	Best	-4.126	0.171	-2.813	0.135	-11.313	0.074	-15.585	0.134	-21.611	0.400	-33.250	0.535
	Average	-2.812		-2.278		-11.273		-14.812		-21.054		-32.687	
	Median	-2.562		-2.197		-11.271		-14.834		-20.970		-32.844	
f3	Worst	1.386		-1.776		-3.908		-6.121		-0.027		-8.934	
	Best	1.307	0.001	-2.415	0.124	-4.647	0.197	-6.938	0.138	-1.281	0.438	-9.471	0.167
	Average	1.587		-2.168		-4.147		-6.546		-0.049		-9.163	
	Median	1.586		-2.177		-4.225		-6.325		-0.035		-9.164	
f4	Worst	2.143		2.2733		-2.0348		-1.076		1.520		1.454	
	Best	1.565	0.087	1.043	0.097	-2.416	0.125	-3.405	0.376	1.412	0.003	1.363	0.025
	Average	1.946		1.476		-2.158		-1.946		1.385		1.312	
	Median	1.854		1.385		-2.235		-2.325		1.363		1.287	
f5	Worst	2.310		1.864		-2.846		-4.385		-0.579		0.898	
	Best	2.096	0.077	1.243	0.856	-2.982	0.044	-5.145	0.054	-1.591	0.323	-12.831	6.009
	Average	2.236		1.454		-2.946		-4.643		-0.999		-3.303	
	Median	2.136		1.285		-2.435		-4.325		-0.963		0.148	
f6	Worst	2.445		1.716		-2.382		-3.414		1.131		0.698	
	Best	2.318	0.038	1.255	0.133	-2.618	0.076	-3.606	0.048	1.012	0.042	-11.404	5.049
	Average	2.408		1.571		-2.486		-3.489		1.077		-5.466	
	Median	2.419		1.601		-2.462		-3.487		1.094		-7.632	
f7	Worst	3.815		3.7846		4.186		4.156		4.099		4.087	
	Best	3.945	0.018	3.978	0.031	4.099	$7.36e-6$	4.099	$4.34e-6$	4.099	$2.96e-9$	4.099	$4.03e-3$
	Average	3.813		3.848		4.099		4.099		4.099		4.097	
	Median	3.736		3.785		-4.099		-4.099		4.099		4.097	
f8	Worst	0.956		-0.960		-5.301		-6.271		-9.89		-Inf	
	Best	-0.644	0.133	-0.500	0.223	-5.444	0.051	-6.455	0.055	-10.255	0.114	-Inf	0
	Average	-0.735		-0.697		-5.357		-6.372		-10.071		-Inf	
	Median	0.736		0.685		-5.435		-6.325		-10.074		-Inf	
f9	Worst	-11.665		-9.76		-9.347		-8.695		-6.848		-6.131	
	Best	-14.433	1.057	-10.574	1.324	-12.194	0.843	-11.937	1.323	-10.680	1.124	-14.831	2.697
	Average	-12.786		-10.456		-10.764		-10.137		-8.178		-9.938	
	Median	1.136		1.085		-7.435		-6.325		-7.745		-9.663	
f10	Worst	0.957		0.974		-5.375		-6.274		0.673		0.280	
	Best	0.644	0.131	0.497	0.223	-5.448	0.049	-6.486	0.056	0.483	0.032	-0.022	0.110
	Average	0.747		0.649		-5.376		-6.348		0.526		0.230	
	Median	1.136		1.085		-7.435		-6.325		0.547		0.211	
f11	Worst	-0.301		-0.303		-4.751		-5.647		-0.896		-1.106	
	Best	-0.301	$2.72e-05$	-0.331	0.009	-5.011	0.071	-5.965	0.085	-1.107	0.103	-1.429	0.135
	Average	-0.301		-0.313		-4.838		-5.804		-0.975		-1.364	
	Median	-0.301		-0.311		-4.835		-5.804		-0.896		-1.402	
f12	Worst	0.779		-1.177		-2.756		-3.158		-0.136		-1.904	
	Best	0.658	0.037	-1.638	0.163	-2.892	0.035	-3.287	0.037	-0.339	0.052	-2.526	0.221
	Average	0.711		-1.388		-2.811		-3.221		-0.252		-2.291	
	Median	0.716		-1.376		-2.801		-3.216		0.248		-2.374	
f13	Worst	4.286		0.574		-0.602		-0.602		0.039		-0.176	
	Best	3.706	0.173	-0.175	0.297	-0.605	0.007	-0.605	0.005	-0.162	0.068	-0.176	$5.38e-6$
	Average	4.147		-0.013		-0.603		-0.603		-0.088		-0.176	
	Median	4.213		-0.135		-0.603		-6.603		-0.086		-0.176	
f14	Worst	2.404		0.405		-3.598		-4.514		0.276		-2.192	
	Best	2.173	0.084	-0.576	0.318	-4.001	0.117	-4.869	0.101	-0.369	0.211	-3.409	0.401
	Average	2.275		0.069		-3.773		-4.656		-0.117		-2.867	
	Median	2.246		0.133		-3.719		-4.651		-0.134		-2.924	

TABLE 3: Continued.

No.	Items	AFSA	Std.	FA	Std.	FFSA	Std.	LFFSA	Std.	DE	Std.	jDE	Std.
f15	Worst	2.588		1.505		-4.281		-4.975		2.241		0.635	
	Best	2.366	0.0705	-4.679	0.198	-5.792	0.117	-7.2474	0.241	1.972	0.077	-0.370	0.372
	Average	2.490		-4.561		-5.408		-6.3259		2.140		0.148	
	Median	2.506		-4.601		-5.456		-6.3259		2.136		0.218	
f16	Worst	1.438		-1.463		-2.762		-3.239		-2.354		-3.392	
	Best	1.268	0.058	-2.296	0.253	-2.838	0.026	-3.334	0.032	-2.482	0.042	-4.646	0.433
	Average	1.366		-1.987		-2.793		-3.282		-2.408		-3.868	
	Median	1.375		-2.016		-2.793		-3.284		-2.402		-3.751	
f17	Worst	-22.954		-25.673		-70.748		-79.645		-47.025		-63.031	
	Best	-24.885	0.643	-32.075	1.721	-77.054	1.908	-80.098	0.131	-51.440	1.403	-68.738	1.942
	Average	-23.772		-28.641		-74.598		-79.996		-48.982		-66.539	
	Median	-23.731		-28.759		-74.907		-80.001		-49.035		-66.559	

TABLE 4: Time complexity analysis of AFSA.

Procedure of AFSA	Time complexity
(1) Initialization of N artificial fishes	$O(N)$
(2) Initialization of 'board'	$O(N)$
(3) Swarming behavior	$O(N^2 + 2 * N)$
(4) Chasing behavior	$O(2 * N^2 + 2 * N)$
(5) Preying behavior	$O(\text{Trynumber} * N)$
(6) Judging of terminal condition	$O(1)$
(7) Information output of 'board'	$O(1)$

TABLE 5: Time complexity analysis of LFFSA.

Procedure of LFFSA	Time complexity
(1) Initialization of N artificial fishes	$O(N)$
(2) Initialization of 'board'	$O(N)$
(3) Chasing behavior	$O(2 * N^2 + 2 * N)$
(4) Preying behavior	$O(\text{trynumber} * N)$
(5) Judging of terminal condition	$O(1)$
(6) Information output of 'board'	$O(1)$

TABLE 6: Average running time of algorithms.

No.	Running time (s)					
	AFSA	FA	FFSA	LFFSA	DE	jDE
f1	10.23	7.47	18.5	7.53	4.64	6.08
f2	12.37	7.73	21.11	7.10	4.32	6.53
f3	10.87	7.80	20.73	8.96	4.12	8.25
f4	11.40	7.67	20.49	10.27	5.63	6.98
f5	10.53	7.66	19.47	8.03	3.28	5.96
f6	19.72	6.10	25.77	16.37	7.71	10.65
f7	10.53	7.43	18.77	8.56	3.27	6.56
f8	12.33	8.03	22.72	9.23	3.79	6.63
f9	20.83	7.20	21.36	7.71	4.01	6.84
f10	16.27	8.46	20.62	7.56	4.65	7.15
f11	9.26	8.95	13.74	6.35	3.63	6.67
f12	9.65	5.34	13.75	6.51	3.04	6.28
f13	9.12	5.26	14.36	10.41	4.13	7.15
f14	13.95	5.74	19.02	10.31	6.48	9.86
f15	9.45	5.39	13.07	7.36	4.62	7.74
f16	8.73	5.21	13.88	11.46	8.52	18.87
f17	14.53	7.28	18.41	5.96	3.32	6.44

4.4. *Parameter Analysis of LFFSA.* The effect of parameters on optimization is analyzed in this section. Taking Ackley function as example, Figure 7 shows the change of the objective function value in the case of varying parameters. Trynumber and β_0 are proportional to the optimization result. Trynumber can impact the time complexity of the algorithm, so the value should be appropriate not to affect the running speed. The best value of γ and δ is 2.5 and 1.2, respectively.

5. Conclusion

LFFSA is proposed to improve the capability of AFSA which integrates the merits of both AFSA and FA. Firstly, the searching characteristic of AFSA is studied by calculating the time complexity. Secondly, 17 benchmark test functions are used to verify LFFSA. Then time complexity of LFFSA is estimated. Numerical results demonstrate that LFFSA has a better performance in accuracy and speed of optimization to solve nonlinear optimization problems than the other test algorithms. However, the solution obtained by LFFSA can be more precise, and the way of modification could provide reference for those efficient algorithms, e.g. DE and GWO.

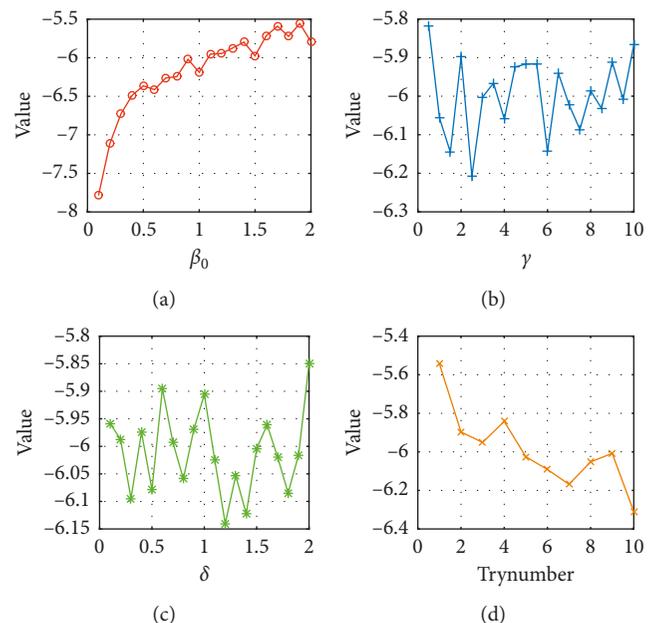


FIGURE 7: Test curves of parameters. (a) Test curve of β_0 . (b) Test curve of γ . (c) Test curve of δ . (d) Test curve of Trynumber.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by National Natural Science Foundation of China (61463028). The authors wish to thank Dr. Yanliang Cui for his fruitful comments and suggestions.

References

- [1] X. L. Li, Z. J. Shao, J. X. Qian et al., "An optimizing method based on autonomous animats: fish-swarm algorithm," *Systems Engineering-Theory and Practice*, vol. 22, pp. 188–200, 2002, in Chinese.
- [2] C. R. Wang, C. L. Zhou, and J. W. Ma, "An improved artificial fish-swarm algorithm and its application in feed-forward neural networks," in *Proceedings of the Fourth International Conference on Machine Learning & Cybernetics*, Guangzhou, China, August 2005.
- [3] H. C. Tsai and Y. H. Lin, "Modification of the fish swarm algorithm with particle swarm optimization formulation and communication behavior," *Applied Soft Computing*, vol. 11, no. 8, pp. 5367–5374, 2011.
- [4] X. Y. Luan, Z. P. Li, and T. Z. Liu, "A novel attribute reduction algorithm based on rough set and improved artificial fish swarm algorithm," *Neurocomputing*, vol. 174, pp. 522–529, 2016.
- [5] Y. Wu, X. Z. Gao, Z. Kai et al., "Knowledge-based artificial fish-swarm algorithm," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 188–200, 2011.
- [6] X. T. Hu, H. Q. Zhang, Z. C. Li et al., "A novel self-adaptation hybrid artificial fish-swarm algorithm," *IFAC Proceedings Volumes*, vol. 46, no. 5, pp. 583–588, 2013.
- [7] K. P. Kumar, B. Saravanan, and K. S. Swarup, "Optimization of renewable energy sources in a microgrid using artificial fish swarm algorithm," *Energy Procedia*, vol. 90, pp. 107–113, 2016.
- [8] Q. He, X. T. Hu, H. Ren et al., "A novel artificial fish swarm algorithm for solving large-scale reliability-redundancy application problem," *ISA Transactions*, vol. 59, pp. 105–113, 2015.
- [9] Z. Q. Zhang, K. P. Wang, L. X. Zhu et al., "A Pareto improved artificial fish swarm algorithm for solving a multi-objective fuzzy disassembly line balancing problem," *Expert Systems with Applications*, vol. 86, pp. 165–176, 2017.
- [10] P. K. Singhal, R. Naresh, and V. Sharma, "Binary fish swarm algorithm for profit-based unit commitment problem in competitive electricity market with ramp rate constraints," *Generation Transmission and Distribution IET*, vol. 9, no. 13, pp. 1697–1707, 2015.
- [11] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, pp. 83–96, Luniver Press, London, 2008.
- [12] S. M. Farahani, B. Nasiri, A. A. Abshouri et al., "An improved firefly algorithm with directed movement," in *Proceedings of IEEE International Conference on Computer Science & Information Technology*, Sichuan, China, June 2011.
- [13] L. D. S. Coelho, D. L. D. A. Bernert, and V. C. Mariani, "A chaotic firefly algorithm applied to reliability-redundancy optimization," *Evolutionary Computation*, vol. 30, pp. 517–521, 2011.
- [14] X. S. Yang and S. Deb, "Eagle strategy using Lévy walk and firefly algorithms for stochastic optimization," in *Studies in Computational Intelligence*, vol. 284, pp. 101–111, Springer, Berlin, Germany, 2010.
- [15] S. L. Tilahun and C. O. Hong, "Modified firefly algorithm," *Journal of Applied Mathematics*, vol. 12, pp. 2428–2439, 2012.
- [16] K. Jagatheesan, B. Anand, S. Samanta et al., "Design of a proportional-integral-derivative controller for an automatic generation control of multi-area power thermal systems using firefly algorithm," *IEEE/CAA Journal of Automatica Sinica*, pp. 1–14, 2016.
- [17] H. Su, Y. Cai, and Q. Du, "Firefly-algorithm-inspired framework with band selection and extreme learning machine for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 1, pp. 309–320, 2016.
- [18] S. Ranganathan, M. S. Kalavathi, and A. R. C. Christoper, "Self-adaptive firefly algorithm based multi-objectives for multi-type FACTS placement," *IET Generation Transmission and Distribution*, vol. 10, no. 11, pp. 188–200, 2016.
- [19] D. F. Teshome, C. H. Le, Y. W. Lin et al., "A modified firefly algorithm for photovoltaic maximum power point tracking control under partial shading," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 5, no. 2, pp. 661–671, 2017.
- [20] M. Alb, P. Alotto, C. Magele et al., "Firefly algorithm for finding optimal shapes of electromagnetic devices," *IEEE Transactions on Magnetics*, vol. 52, no. 3, pp. 1–4, 2016.
- [21] A. Mishra and V. N. K. Gundavarapu, "Line utilisation factor-based optimal allocation of IPFC and sizing using firefly algorithm for congestion management," *Generation Transmission and Distribution IET*, vol. 10, no. 1, pp. 115–122, 2016.
- [22] M. H. Hornig, "Vector quantization using the firefly algorithm for image compression," *Expert Systems with Applications*, vol. 39, no. 1, pp. 078–1091, 2012.
- [23] A. Kazem, E. Sharifi, F. K. Hussain et al., "Support vector regression with chaos-based firefly algorithm for stock market price forecasting," *Applied Soft Computing*, vol. 13, no. 2, pp. 947–958, 2013.
- [24] L. F. He and S. W. Huang, "Modified firefly algorithm based multilevel thresholding for color image segmentation," *Neurocomputing*, vol. 240, pp. 152–174, 2017.
- [25] A. H. Gandomi, X. S. Yang, and A. H. Alavi, "Mixed variable structural optimization using Firefly Algorithm," *Computers and Structures*, vol. 89, no. 23–24, pp. 2325–2336, 2011.
- [26] M. Alweshah and S. Abdullah, "Hybridizing firefly algorithms with a probabilistic neural network for solving classification problems," *Applied Soft Computing*, vol. 35, pp. 513–524, 2015.
- [27] R. M. Rizk-Allah, E. M. Zaki, A. A. El-Sawy et al., "Hybridizing ant colony optimization with firefly algorithm for unconstrained optimization problems," *Applied Mathematics and Computation*, vol. 224, pp. 473–483, 2013.
- [28] X. S. Yang, S. S. S. Hosseini, and A. H. Gandomi, "Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect," *Applied Soft Computing*, vol. 12, no. 3, pp. 1180–1186, 2012.
- [29] J. Senthilnath, S. N. Omkar, and V. Mani, "Clustering using firefly algorithm: performance study," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 164–171, 2011.

- [30] T. Kanimozhi and K. Latha, "An integrated approach to region based image retrieval using firefly algorithm and support vector machine," *Neurocomputing*, vol. 151, pp. 1099–1111, 2015.
- [31] A. Baykasoglu and F. B. Ozsoydan, "Adaptive firefly algorithm with chaos for mechanical design optimization problems," *Applied Soft Computing*, vol. 36, pp. 152–164, 2015.
- [32] I. Fister, M. Perc, S. M. Kamal et al., "A review of chaos-based firefly algorithms: perspectives and research challenges," *Applied Mathematics and Computation*, vol. 252, pp. 155–165, 2015.
- [33] I. Fister, I. Fister, X. S. Yang et al., "A comprehensive review of firefly algorithms," *Swarm and Evolutionary Computation*, vol. 13, pp. 34–46, 2013.
- [34] C. T. Brown, L. S. Liebovitch, and R. Glendon, "Lévy flights in Dobe Ju/'hoansi foraging patterns," *Human Ecology*, vol. 35, no. 1, pp. 129–138, 2007.
- [35] I. Pavlyukevich, "Lévy flights, non-local search and simulated annealing," *Mathematics*, vol. 226, no. 2, pp. 1830–1844, 2012.
- [36] I. Pavlyukevich, "Cooling down Lévy flights," *Journal of Physics A: Mathematical and Theoretical*, vol. 40, no. 41, pp. 12299–12313, 2007.
- [37] R. Jensi and G. W. Jiji, "An enhanced particle swarm optimization with Lévy flight for global optimization," *Applied Soft Computing*, vol. 43, pp. 248–261, 2016.
- [38] D. Tang, J. Yang, S. Dong et al., "A Lévy flight-based shuffled frog-leaping algorithm and its applications for continuous optimization problems," *Applied Soft Computing*, vol. 49, pp. 641–662, 2016.
- [39] M. Yahya and M. P. Saka, "Construction site layout planning using multi-objective artificial bee colony algorithm with Lévy flights," *Automation in Construction*, vol. 38, pp. 14–29, 2014.
- [40] C. Blum and A. Roli, *Hybrid Metaheuristics: An Introduction*, Springer, Berlin, Germany, 2008.
- [41] 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, 2007.
- [42] M. F. Shlesinger, G. M. Zaslavsky, and U. Frisch, *Lévy Flights and Related Topics in Physics*, Springer, Berlin, Heidelberg, Germany, 1995.
- [43] M. F. Shlesinger, "Mathematical physics: search research," *Nature*, vol. 443, no. 7109, pp. 281–282, 2006.
- [44] S. T. Hsieh, T. Y. Sun, C. C. Liu et al., "Efficient population utilization strategy for particle swarm optimizer," *IEEE Transactions on Systems Man & Cybernetics Part B*, vol. 39, no. 2, pp. 444–456, 2009.
- [45] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Proceedings of IEEE Swarm Intelligence Symposium*, vol. 107, pp. 120–127, Honolulu, Hawaii, April 2007.
- [46] U. Manber, *Introduction to Algorithms: A Creative Approach*, Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA, 1989.



Hindawi

Submit your manuscripts at
www.hindawi.com

