An Improved Enhanced Fireworks Algorithm Based on Adaptive Explosion Amplitude and Lévy Flight

Qiang Qu, Mei-li Qi, Ru-bing Gong, and Xue-bo Chen

***Abstract*—To solve the problems of the waste of resources caused by the small explosion amplitude (even close to 0) of the best firework in the conventional fireworks algorithm (FWA), and the relatively weak local search ability caused by the minimal explosion amplitude check method in the enhanced fireworks algorithm (EFWA), this paper proposes two improved strategies for FWA. First, based on the heuristic information of the distance between the best firework and other fireworks, an adaptive explosion amplitude strategy is proposed to search the local area accurately at the final phase of the FWA. Second, the highly random Lévy flight strategy is adopted instead of the Gaussian sparks strategy to generate mutation sparks in the EFWA to enhance the diversity of local search. Simulation results on 12 standard benchmark functions and their shifted functions indicate that the proposed algorithm improves the optimization precision and obtains better performance in high-dimensional complex optimization problems compared with the EFWA, the hybrid fireworks algorithm with differential evolution operator (FWADE), and the particle swarm optimization algorithm (PSO).**

***Index Terms*—enhanced fireworks algorithm, fireworks algorithm, adaptive explosion amplitude, Lévy flight**

I. INTRODUCTION

GLOBAL optimization problems are common in engineering and other fields. Owing to the complexity of search space and the existence of multiple local optimal values, global optimization problems are often difficult to solve, especially for high-dimensional optimization problems. Therefore, in the past 10 years, researchers have proposed a series of metaheuristic algorithms based on physical phenomena, natural laws or biological populations to solve the global optimization problems and have achieved good results [1-3].

Manuscript received November 1, 2019; revised August 17, 2020. This work was supported in part by National Natural Science Foundation of China (Grant No. 71571091 and 71371092) .

Qiang Qu is with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China (e-mail:askdqq@163.com).

Mei-li Qi is with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China (e-mail: 1017057418@qq.com).

Ru-bing Gong is with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China (e-mail: 979242214@qq.com).

Xue-bo Chen is with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China (e-mail: xuebochen@126.com).

In the fireworks algorithm (FWA), a novel metaheuristic algorithm proposed by Tan and Zhu in 2010 [4], a firework is regarded as a solution to optimization problems around the feasible range and the process of fireworks exploding can be regarded as the process of finding the optimal solution. After the explosion, the fireworks or sparks are obtained and evaluated. The best individual in the population is kept and the rest of the individuals are selected by certain strategies for the next generation. The algorithm stops when the optimal solution is found [5]. The explosion amplitude and the number of sparks are the two key parameters that determine the performance of FWA, the values of which are taken according to the following rules: fireworks with better fitness produce more sparks in a smaller explosion amplitude range, while fireworks with worse fitness produce fewer sparks in a larger explosion amplitude range. Gaussian mutation is also used to ensure diversity of the sparks [6].

Many scholars have put forward algorithms to improve upon FWA. Zheng et al. summarized the shortcomings of the FWA and proposed the enhanced fireworks algorithm (EFWA) through five strategies: minimal explosion amplitude check strategy, explosion sparks strategy, Gaussian sparks strategy, mapping strategy, and selection strategy [7]. By combining the cultural algorithm and FWA, Gao et al. proposed the cultural firework (CF) algorithm to design a digital filter and improved the convergence speed [8]. Zhang et al. used three other strategies to improve the performance of FWA. The first one was, in order to avoid premature convergence, a new Gaussian mutation operator that was used to enhance information interaction between sparks. The second strategy was, in order to enhance information sharing among populations, a migration operator that was applied in biogeography-based optimization to generate explosion sparks. The third was a new selection strategy for retaining better fireworks to the next generation with a higher probability [9]. By combining the grey wolf optimizer with FWA, a hybrid algorithm, FWA-GWO, was developed in [10]. Based on the differential evolution (DE), a hybrid FWA with differential evolution operator (FWADE) was proposed in [11]. Zhao et al. utilized the development from the previous best firework to the current best firework to guide the evolution of fireworks and proposed a best firework updating information guided adaptive fireworks algorithm [12]. Based on dynamic search and tournament selection, Han et al. proposed an improved FWA [13].

Although these algorithms improve the performance of fireworks algorithms to some extent, all of them adopt either the explosion amplitude strategy used in the FWA or the minimal explosion amplitude check strategy used in the EFWA. However, there are some shortcomings in the explosion amplitude strategies adopted in FWA and EFWA. In the FWA, the explosion amplitude of a firework with best fitness usually is very small, even close to zero, so the explosion sparks appear at nearly the same position as the optimal firework, causing a waste of resources. Therefore, the EFWA introduces the minimal explosion amplitude check strategy and obtains a better result. However, the increase or decrease of explosion amplitude in the EFWA only depends on the number of iterations and the fitness value of fireworks and does not take into account the heuristic information in the convergence process of the EFWA. As a result, the performance of the EFWA is sensitive to a pre-defined number of iterations. It is also difficult to reduce the explosion amplitude to a very small value in the last stage of the EFWA, which results in a weak local search ability.

To solve these problems, we use the distance between the current firework and the optimal firework, the minimum explosion radius, and maximum explosion radius to construct an adaptive dynamic explosion radius updating strategy. This strategy not only solves the problem that some explosion amplitude can be close to zero in FWA, but also balances the global search and local search capabilities of the FWA. Furthermore, to improve the diversity of population and global search ability, we replace the Gaussian process used in FWA and EFWA with Lévy flight with strong randomness to generate mutation sparks in the improved algorithm. Finally, the simulation results on 12 benchmark functions and their shifted functions show that the proposed algorithm improves optimization precision and convergence ability in comparison with EFWA, FWADE, and particle swarm optimization (PSO).

The rest of the article is structured as follows. Section II introduces the principle of the EFWA and analyzes the defects of the EFWA. Section III describes the principle of the proposed improved enhanced fireworks algorithm. Section IV gives the simulation results. Section V summarizes the main findings of this study.

II. Principle of the EFWA

A. Explosion Sparks Strategy

The explosion sparks strategy simulates the process of fireworks explosion in the EFWA. To ensure diversity of population and balance the global and local searching abilities, the EFWA incorporates an automatic procedure, that is, fireworks with better fitness have smaller explosion amplitudes and generate more explosion sparks than those with less fitness. Suppose that the number of fireworks is N and the number of dimensions is d , the number of explosion sparks s_i and the explosion amplitude A_i for each firework x_i can be defined as:

$$s_i = M \cdot \frac{f_{max} - f\left(\boldsymbol{x}_i\right) + \theta}{\sum_{i=1}^{N}\left(f_{max} - f\left(\boldsymbol{x}_i\right)\right) + \theta} \tag{1}$$

$$A_i = A \cdot \frac{f\left(\boldsymbol{x}_i\right) - f_{min} + \theta}{\sum_{i=1}^{N}\left(f\left(\boldsymbol{x}_i\right) - f_{min}\right) + \theta} \tag{2}$$

where M is a constant to control the number of explosion sparks and A is also a constant to control the explosion amplitude, $f\left(\boldsymbol{x}_i\right)$ is the fitness of firework x_i , $f_{max} = \max\left(f\left(\boldsymbol{x}_i\right)\right)$ and $f_{min} = \min\left(f\left(\boldsymbol{x}_i\right)\right)$ are the fitness of the worst firework and the best firework, respectively, in the current population for minimization problems. θ is the minimum constant of the computer.

To avoid the problem of wasting resources caused by the explosion amplitude of the best firework approaching zero, the EFWA algorithm introduces the minimal explosion amplitude check strategy to limit the minimum of the explosion amplitude, which can be written as:

$$A_i^k = \begin{cases} A_{\min}^k & A_i^k < A_{\min}^k \\ A_i^k & A_i^k \geq A_{\min}^k \end{cases} \tag{3}$$

$$A_{\min}^k = A_{init} - \frac{A_{init} - A_{final}}{t_{\max}}\sqrt{\left(2t_{\max} - t\right)t} \tag{4}$$

where A_{init} and A_{final} are the initial and final explosion amplitude, respectively. $t_{\max}$ is the maximum number of function evaluations, and t is the current number of function evaluations.

Once we obtain the explosion amplitude from (2) and (3), the new sparks can be generated by:

$$\Delta_i^k = A_i \cdot rand\left(-1,1\right) \tag{5}$$

$$x_i^k = x_i^k + \Delta_i^k \tag{6}$$

where x_i^k is the value of the ith firework in dimension k , and $rand\left(-1,1\right)$ is a random vector in [-1, +1].

B. Mapping Strategy

When a new spark position exceeds the search range, the new spark will be re-mapped to the feasible space by:

$$x_i^k = x_{\min}^k + rand(0,1) \cdot \left(x_{\max}^k - x_{\min}^k\right) \tag{7}$$

where $x_{\min}^k$ and $x_{\max}^k$ stand for the lower and upper boundary of the solution space in dimension k , respectively.

C. Gaussian Sparks Strategy

To maintain the diversity of population, the Gaussian sparks strategy is adopted to generate mutation sparks. Therefore, $\hat{m}$ fireworks can be selected to generate Gaussian sparks as:

$$x_i^k = x_i^k + \left(x_{best}^k - x_i^k\right) \cdot g \tag{8}$$

where $g \sim N\left(0,1\right)$, $i = (1, 2, \cdots, \hat{m})$, and x_{best}^k is the position of the current best firework in dimension k .

D. Selection Strategy

Since the selection strategy in FWA is based on distance, it is necessary to calculate the distance between any two fireworks or sparks. Although the selection operator in FWA can increase diversity of population, it leads to more calculations. To reduce the amount of calculation, the elitism random selection (ERP) method is applied in EFWA, that is, the best individual with best fitness is selected first and the others are selected randomly.

III. Improved Enhanced Fireworks Algorithm

A. Adaptive Explosion Amplitude Strategy

From (2), it can be obtained that a firework with better fitness owns smaller explosion amplitude, while a firework with less fitness owns larger explosion amplitude, by which the FWA achieves the balance of exploration and exploitation. Although this idea seems reasonable, the explosion amplitude of the firework with the best fitness will usually be very small. If the explosion amplitude is close to zero, the explosion sparks will be located at nearly the same position as the firework itself. As a result, the location of the best fireworks may not be updated until another firework finds a better location.

By introducing the minimum radius detection strategy shown as (3), the EFWA solves this problem to a certain extent. However, this detection strategy also has two drawbacks: First, at the final phase of EFWA, small explosion amplitude is needed to perform accurate search in the local area. However, the explosion amplitude reduction method shown as (4) in EFWA makes it difficult to reduce the amplitude to a very small value, which will weaken local search ability. Second, from (4), it can be known that the reduction or amplification of the lower bound of explosion amplitude depends only on the iterations number. However, the heuristic information in the optimization process is not used.

Therefore, we propose to use distance between the best firework and current firework to dynamically update the explosion amplitude strategy in order to balance the globe and local search capabilities. The adaptive explosion strategy can be described as:

$$A_i = A_{\min} + \left(A_{\max} - A_{\min}\right) d_i \tag{9}$$

$$d_i = \frac{\left\| x_i - x_{best} \right\|}{d_{\max}} \tag{10}$$

where x_i is the position of the ith firework, x_{best} is the position of the current best firework, and $d_{\max}$ is the maximum distance between the current best firework and any other fireworks. Constants $A_{\max}$ and $A_{\min}$ represent the maximum and minimum value of explosion amplitude, respectively.

The explosion amplitude can be dynamically adjusted by (9) and (10). When the position of the firework is far away from the current best firework, d_i will increase, which will generate big explosion amplitude to improve global search ability. On the contrary, when the position of the firework is close to the current best firework, d_i will decrease, which will generate small explosion amplitude to improve local search ability. When the firework is the current best firework, $d_i = 0$ and $A_i = A_{\min}$. Thus, the improved enhanced fireworks algorithm performs careful search by the smallest explosion amplitude near the optimal value to improve local search ability and convergence accuracy.

The values of $A_{\max}$ and $A_{\min}$ can influence the performance of the improved EFWA. If $A_{\max}$ is too small, the search amplitude at the early phase of algorithm will be small, which causes the algorithm convergence too fast and falls into local minima. If $A_{\min}$ is set too large, the search amplitude at the final phase will be large, which directly affects the local search ability of the algorithm. Through experiments, it can be suggested that:

$$A_{\max} = 0.02 \cdot \left(X_{\max} - X_{\min} \right) \tag{11}$$

$$A_{\min} = 0.005 \cdot \left(X_{\max} - X_{\min} \right) \tag{12}$$

where $X_{\max}$ and $X_{\min}$ refer to the lower and upper bounds of the search space, respectively.

B. Lévy Flight Strategy

Through Gaussian sparks strategy, FWA is easy to find the optimal value at the origin of coordinate. Nevertheless, the performance of FWA will be poor when the function is shifted. To overcome the disadvantage of Gaussian sparks, EFWA uses the position of the current global best individual to generate Gaussian sparks. To further improve the diversity of population and the optimal performance for the shifted function, the Lévy flight function with more randomness is adopted instead of Gaussian distribution used in EFWA to generate the mutation sparks in the proposed algorithm.

The Lévy flight with strong randomness comes from Paul Lévy, a French mathematician [14]. The Lévy flight is a random walk method that combines short distance search and occasional long-distance walk. Its step length obeys Lévy distribution and its direction obeys uniform distribution. In addition to the cuckoo search (CS) algorithm, the Lévy flight strategy has been successfully applied to many swarm intelligence algorithms [15-16]. It realizes the diversity of population and has a better jump ability to avoid local minima. The simplified form of Lévy distribution can be described as:

$$L(s) \sim |s|^{-1-\beta}, \quad 0 < \beta \le 2 \tag{13}$$

where s is random step length of the Lévy flight. Mantegna's algorithm is often used to simulate the random walk behavior in Lévy flight [17]. Random step length can be defined as:

$$s = \frac{\mu}{|v|^{1/\beta}} \tag{14}$$

where μ and v obey normal distribution, i.e.,

Algorithm: ALEFWA

Initialize N fireworks: $\boldsymbol{x}_i$, $i = 1, 2, ..., N$
for $i = 1$ to N do
  Calculate the fitness of each firework $f(\boldsymbol{x}_i)$
  Calculate the number of explosion sparks s_i for each firework by (1)
  Calculate the explosion amplitude A_i for each firework by (9) and (10)
end for

//For each individual, generate s_i sparks within amplitude A_i
for $i = 1$ to N do
  $\hat{\boldsymbol{x}}_i = \boldsymbol{x}_i$
  for j=1 to s_i do
    for each dimension of $\hat{\boldsymbol{x}}_i$ do
      if (round(rand(0,1)) == 1) $\hat{x}_i^k += A_i \cdot rand(-1,1)$
      if ($\hat{x}_i^k$ is out of scope) execute mapping operation by (7)
    end for
    Incorporate $\hat{\boldsymbol{x}}_i$ into the population of explosive sparks
    Calculate fitness of $\hat{\boldsymbol{x}}_i$
  end for
end for

//Generate mutation sparks by the Lévy flight strategy
  Set β = 1.5, calculate σ_μ by (17)
  for $i = 1$ to $\hat{m}$ do
    $\hat{\boldsymbol{x}}_i$ = randomly select firework
    for each dimension of $\hat{\boldsymbol{x}}_i$
      if (round(rand(0,1)) == 1)
        Calculate $u = \text{randn}(\text{size}(\hat{\boldsymbol{x}}_i)) \cdot \sigma_\mu$ and $v = \text{randn}(\text{size}(\hat{\boldsymbol{x}}_i))$
        Calculate Lévy distribution $l = u / abs(v)\hat{}(1/\beta)$
        Calculate Lévy explosion sparks $\hat{x}_k^i = \hat{x}_k^i + (\hat{x}_{best}^i - \hat{x}_k^i) \cdot l$
      end if
      if ($\hat{x}_k^i$ is out of scope) execute mapping operation by (7)
    end for
    Incorporate $\hat{\boldsymbol{x}}_i$ into the population of explosive sparks
    Calculate fitness of $\hat{\boldsymbol{x}}_i$
end for

//Selection
Keep the best individual and randomly select $(N-1)$ individuals for next generation

---

Fig. 1. Pseudocode of ALEFWA

$$\mu \sim N(0, \sigma_\mu^2) \tag{15}$$

$$v \sim N(0, \sigma_v^2) \tag{16}$$

where

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma\left[(1+\beta)/2\right]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta} \tag{17}$$

$$\sigma_v = 1 \tag{18}$$

and β usually equals 1.5.

The Lévy explosion sparks can be calculated by:

$$x_k^i = x_k^i + (x_{best}^i - x_k^i) \cdot l \tag{19}$$

where x_k^i denotes the ith element of x_k, and l is the random number in Lévy distribution.

C. ALEFWA Algorithm

The EFWA based on adaptive explosion amplitude strategy and Lévy flight strategy (ALEFWA) is shown in Fig. 1.

IV. EXPERIMENTS

A. Parameter Setting and Simulation Environment

The numerical efficiency of the ALEFWA developed in this article was tested by 12 benchmark functions used in [5]. The same parameters as [5] are used, i.e., N =5, M =50, A =40, a =0.04, b =0.8, $\hat{m}$ =5 for ALEFWA, EFWA [7] and FWADE [11]. Three more parameters in ALEFWA, β , $A_{\max}$ and $A_{\min}$ can take values as follows: β =1.5, $A_{\max}$ and $A_{\min}$ are set as (11) and (12). The scaling factor and crossover rate in FWADE are configured with $F = 0.5$ and $C_r = 0.9$. Furthermore, PSO uses a linearly decreasing weight strategy. The parameters of PSO are as follows: maximum inertia factor $w_{\max} = 0.9$, minimum inertia factor $w_{\min} = 0.2$, learning factors $c_1 = c_2 = 2$, and maximum speed $V_{\max} = 6$. The experimental platform is MATLAB R2012b, running on 64-bit Windows 7 with a 2.5 GHz Intel Core i7-3820QM and 6GB RAM.

TABLE I
STANDARD BENCHMARK FUNCTIONS

No.	Function Name	Search Range	Optimal Positions	Fitness at the Optimal Position	Dimension
F1	Sphere	[±100]	0.0^D	0	30
F2	Schwefel's Problem 1.2	[±100]	0.0^D	0	30
F3	Generalized Rosenbrock	[±30]	1.0^D	0	30
F4	Ackley	[±32]	0.0^D	0	30
F5	Generalized Griewank	[±600]	0.0^D	0	30
F6	Generalized Rastrigin	[±5.12]	0.0^D	0	30
F7	Penalized function P16	[±50]	1.0^D	0	30
F8	Six-hump Camel Back	[±5]	(-0.09,0.71) (0.09,-0.71)	-1.032	2
F9	Goldsein-Price	[±2]	(0,-1)	3	2
F10	Schaffer F6	[±100]	0.0^D	0	2
F11	Axis Parallel Hyper Ellipsoid	[±5.12]	0.0^D	0	30
F12	Rotated Hyper Ellipsoid	[±65.5]	0.0^D	0	30

TABLE II
SHIFTED INDEX (SI) AND SHIFTED VALUE (SV)

SI	1	2	3	4	5	6
SV	$0.05\times(\Delta^k/2)$ [a]	$0.1\times(\Delta^k/2)$	$0.2\times(\Delta^k/2)$	$0.3\times(\Delta^k/2)$	$0.5\times(\Delta^k/2)$	$0.7\times(\Delta^k/2)$

[a] $\Delta^k = x^k_{\max} - x^k_{\min}$

TABLE III
STATISTICAL RESULTS OF FUNCTIONS 1-6

SI	Alg.	F1		F2		F3		F4		F5		F6	
		Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
0	PSO	3.21E-2	2.1E-3	3.21E+0	5.3E-1	7.64E-1	2.3E-2	2.75E+0	7.0E-1	9.80E-1	6.3E-2	1.21E+2	6.5E+1
	EFWA	5.57E-2	3.2E-3	4.28E-1	2.1E-1	3.03E+0	3.1E-1	4.14E+0	8.3E-1	1.67E-1	5.6E-2	1.14E+2	5.6E+1
	FWADE	2.38E-2	5.1E-3	1.92E-1	5.3E-2	6.55E-1	8.5E-1	3.76E+0	2.6E+0	5.61E-2	2.3E-2	2.13E+1	5.6E+0
	ALEFWA	2.38E-2	2.3E-3	1.92E-1	3.4E-2	5.34E-1	3.4E-1	1.21E-1	6.7E-2	3.43E-2	7.8E-3	1.23E+0	4.8E-1
1	PSO	3.32E-2	2.5E-2	3.51E+0	5.6E-1	8.62E-1	6.7E-2	3.54E+0	5.2E-1	8.62E-2	3.2E-2	1.46E+2	3.6E+1
	EFWA	5.94E-2	3.5E-2	3.40E-1	2.3E-2	3.31E+0	5.6E-1	7.51E+0	1.1E+0	1.19E-1	6.2E-2	1.34E+2	6.5E+1
	FWADE	2.59E-2	2.3E-2	2.12E-1	3.2E-2	7.87E-1	4.6E-1	4.97E+0	1.3E+0	3.87E-2	2.6E-2	1.59E+1	9.8E+0
	ALEFWA	2.32E-2	2.5E-2	1.93E-1	2.1E-2	7.02E-1	3.5E-1	4.50E-1	7.6E-2	3.45E-2	9.8E-3	2.31E+0	5.6E-1
2	PSO	3.69E-2	2.3E-2	3.81E+0	2.3E-1	8.96E-1	1.2E-2	3.78E+0	6.2E-1	8.60E-2	4.2E-2	1.23E+2	6.5E+1
	EFWA	9.42E-2	4.5E-2	2.74E+0	7.5E-2	3.68E+0	9.8E-1	7.93E+0	2.6E+0	9.93E-2	1.2E-2	1.08E+2	7.8E+1
	FWADE	2.97E-2	8.6E-3	2.65E-1	5.6E-2	8.19E-1	6.5E-1	5.28E+0	2.3E+0	5.98E-2	4.2E-2	1.11E+1	5.6E+0
	ALEFWA	2.67E-2	6.7E-3	2.03E-1	3.4E-2	7.45E-1	2.3E-2	5.61E-1	7.8E-2	4.56E-2	3.2E-2	3.51E+0	6.7E-1
3	PSO	4.12E-2	2.5E-2	3.91E+0	5.3E-1	9.21E-1	2.3E-2	4.51E+0	6.2E-1	8.9E-2	2.4E-2	8.60E+1	3.6E+0
	EFWA	8.00E-2	4.3E-2	2.57E+1	3.6E-2	4.35E+0	4.7E-1	9.03E+0	2.6E+0	1.31E-1	4.6E-2	9.66E+1	6.5E+0
	FWADE	3.08E-2	2.4E-2	2.16E-1	2.3E-3	9.44E-1	6.5E-1	5.99E+0	1.3E+0	5.46E-2	1.2E-2	1.30E+1	4.6E+0
	ALEFWA	2.78E-2	2.1E-2	2.23E-1	2.3E-3	8.65E-1	1.7E-1	5.80E-1	6.5E-2	4.82E-2	6.3E-2	4.34E+0	7.8E-1
4	PSO	3.98E-2	2.7E-2	3.61E+0	5.6E-2	1.02E+0	2.3E-1	5.23E+0	1.5E+0	7.80E-1	2.3E-2	8.46E+1	6.4E+0
	EFWA	8.47E-2	3.5E-2	4.31E+1	3.2E-2	5.17E+0	8.9E-1	1.12E+1	4.6E+0	1.68E-1	6.3E-2	7.46E+1	7.8E+0
	FWADE	2.90E-2	1.2E-2	2.27E-1	5.3E-2	9.54E-1	4.5E-1	6.52E+0	1.3E+0	7.07E-2	4.3E-3	1.60E+1	2.3E+0
	ALEFWA	3.03E-2	1.1E-2	2.32E-1	4.3E-2	8.34E-1	3.6E-2	6.40E-1	5.6E-2	5.63E-2	1.4E-2	5.62E+0	8.9E-1
5	PSO	4.01E-2	3.2E-2	4.81E+0	2.3E-1	1.53E+0	8.9E-2	1.11E+1	4.2E+0	9.80E-1	4.3E-3	1.02E+2	5.6E+1
	EFWA	9.41E-2	5.2E-2	2.99E+0	3.2E-2	5.30E+0	5.6E-1	1.15E+1	5.6E+0	1.05E-1	1.2E-2	1.21E+2	3.6E+1
	FWADE	3.20E-2	2.3E-2	2.82E-1	1.2E-2	1.43E+0	7.8E-2	7.03E+0	1.5E+0	8.85E-2	4.2E-2	1.10E+1	6.3E+0
	ALEFWA	3.03E-2	2.1E-2	2.34E-1	2.1E-2	1.12E+0	2.3E-1	6.90E-1	3.2E-2	7.82E-2	6.7E-3	5.64E+0	1.0E+0
6	PSO	4.02E-2	3.5E-2	4.91E+0	2.3E-1	1.43E+0	5.6E-1	1.34E+1	3.5E+0	9.80E-1	4.5E-2	1.45E+2	1.2E+1
	EFWA	7.95E-2	4.2E-2	3.13E-1	5.6E-2	5.53E+0	1.6E+0	1.25E+1	5.6E+0	2.55E-1	5.6E-2	1.75E+2	4.6E+1
	FWADE	2.81E-2	3.2E-2	1.44E-1	3.4E-2	2.91E+0	1.5E+0	7.32E+0	2.2E+0	2.23E-1	1.2E-2	9.99E+0	5.6E+0
	ALEFWA	2.98E-2	3.4E-2	2.34E-1	2.3E-2	1.19E+0	6.7E-2	9.20E-1	1.1E-2	2.34E-2	3.4E-3	5.67E+0	1.2E+0

Table I lists the numbers, names, search range, optimal positions, fitness at the optimal position, and dimensions of 12 standard functions. These functions consist of unimodal and multimodal functions. The functions F1-F5 are unimodal because they have only one global optimum, and allow evaluating the exploitation capability of optimization algorithms. The functions F6-F12 are multimodal. They have many local optima, and the number of local optima increases exponentially with increasing the problem size. Therefore, they are highly useful in evaluating the exploration capability of optimization algorithms. To test the performance of ALEFWA for those functions whose optimums are not located at the original point, a number of shifted values are added to the benchmark functions. The shifted index (SI) and shifted value (SV) are shown in Table II, where $x^k_{\max}$ and $x^k_{\min}$ represent the maximum and minimum boundaries for search range, respectively.

B. Analysis of Simulation Results

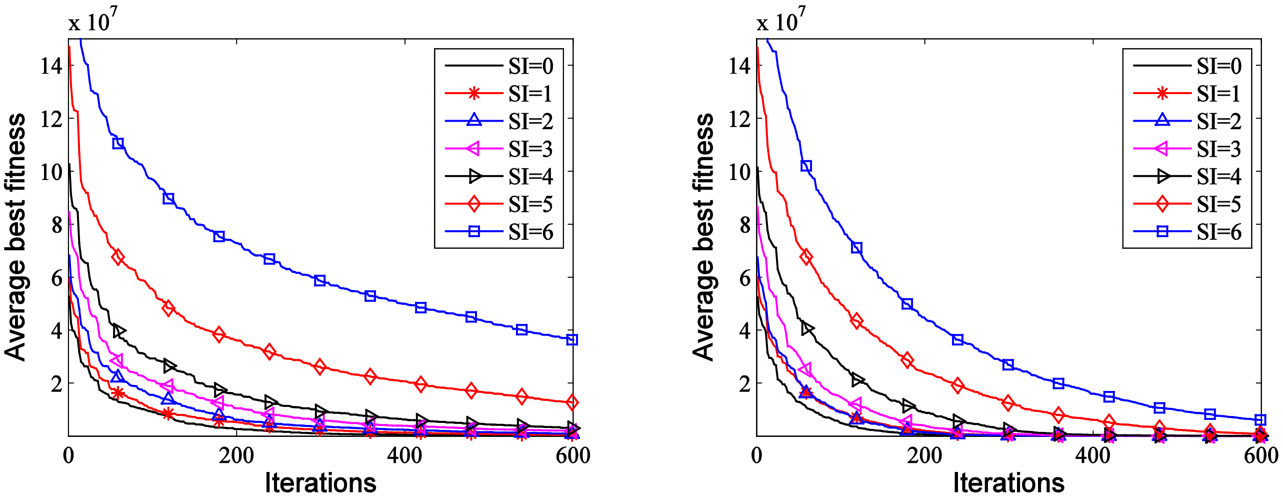
The selected 12 benchmark functions and 6 shifted benchmark functions were evaluated 5000 times on the PSO, EFWA, FWADE, and ALEFWA. Each function was run 30 times and the mean value and standard deviation of the best fitness so far were taken. The final results are presented in Tables III and IV.

From these tables, one can draw several conclusions. PSO usually obtains better results than EFWA for small SI. As SI increases, the performance of PSO decreases rapidly, and the performance of EFWA, FWADE, and ALEFWA deteriorate relatively slower than that of PSO. The average best fitness of

ALEFWA remains almost unaffected even if the optimum of the function is shifted to the edge of the search space. As a result, EFWA, FWADE, and ALEFWA obtain better results than PSO for large SI. It is worth noting that ALEFWA is always significantly better than EFWA, FWADE, and PSO, regardless of SI.

TABLE IV
AVERAGE BEST FITNESS OF FUNCTIONS 7-12

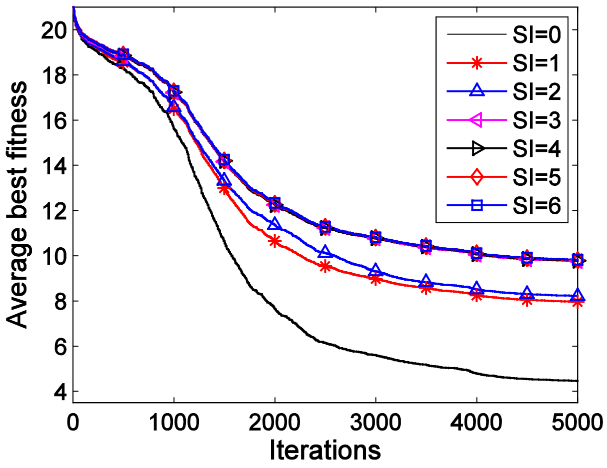
SI	Alg.	F7		F8		F9		F10		F11		F12	
		Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
0	PSO	6.61E-4	2.3E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	3.89E-3	4.9E-3	1.75E-3	2.3E-4	2.31E-1	5.6E-2
	EFWA	3.13E-3	5.6E-4	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	3.89E-3	4.9 E-3	2.75E-3	1.2E-4	4.45E-1	7.8E-2
	FWADE	6.41E-4	7.9E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	2.91E-3	4.6 E-3	9.35E-4	6.3E-5	1.47E-1	3.2E-2
	ALEFWA	5.43E-4	3.4E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	5.62E-4	3.6E-4	3.45E-4	4.3E-5	4.75E-2	2.4E-3
1	PSO	4.61E-4	4.6E-5	-1.03E+0	0.0E+0	3.00E+0	2.10E-3	2.91E-3	4.6E-3	1.21E-3	4.6E-4	2.31E-1	3.2E-2
	EFWA	0.00E+0	0.0E+0	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	4.38E-3	2.3 E-3	2.73E-3	6.3E-4	5.40E-1	5.2E-2
	FWADE	6.12E-4	5.6E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	3.89E-3	4.9 E-3	3.17E-4	3.6E-5	1.06E-1	2.3E-2
	ALEFWA	6,45E-4	6.3E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	6.52E-4	3.2E-4	4.35E-4	4.2E-5	5.67E-2	3.4E-3
2	PSO	0.00E+0	0.0E+0	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	5.83E-3	6.9E-3	1.12E-3	3.6E-4	3.61E-1	3.6E-2
	EFWA	1.62E-3	2.3E-4	-1.03E+0	0.0E+0	3.00E+0	3.2E-3	2.92E-3	1.6 E-3	2.97E-3	4.6E-4	4.36E-1	6.3E-2
	FWADE	8.17E-4	5.6E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	3.89E-3	4.9 E-3	1.05E-3	6.3E-4	1.09E-1	3.6E-2
	ALEFWA	6.78E-4	5.9E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	6.83E-4	4.5E-4	5.67E-4	5.6E-5	6.32E-2	5.6E-3
3	PSO	9.21E-4	6.5E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	4.37E-3	3.6E-3	2.32E-3	1.0E-3	3.56E-1	6.3E-2
	EFWA	1.43E-2	4.6E-3	-1.03E+0	0.0E+0	3.00E+0	3.2E-3	2.92E-3	4.6 E-3	3.11E-3	1.1E-3	4.66E-1	2.3E-2
	FWADE	8.46E-4	6.9E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	2.92E-3	4.6 E-3	7.84E-4	6.3E-5	1.49E-1	5.3E-2
	ALEFWA	7.82E-4	7.8E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	7.23E-4	6.3E-4	6.34E-4	6.7E-5	7.45E-2	7.3E-3
4	PSO	1.29E-3	5.6E-4	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	5.83E-3	4.9E-3	5.23E-3	8.9E-4	6.21E-1	1.2E-1
	EFWA	1.39E-2	6.3E-3	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	3.89E-3	4.9 E-3	3.13E-3	5.6E-4	4.07E-1	1.6E-2
	FWADE	8.07E-4	6.4E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	2.92E-3	4.6 E-3	9.04E-4	1.3E-4	1.69E-1	2.3E-2
	ALEFWA	7.98E-4	5.6E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	8.42E-4	7.8E-4	8.79E-4	7.8E-5	7.62E-2	6.7E-3
5	PSO	1.35E-3	5.7E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	4.86E-3	5.0E-3	6.31E-3	1.2E-3	7.81E-1	3.6E-2
	EFWA	2.61E-3	6.3E-4	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	5.34E-3	2.3 E-3	5.37E-3	1.02E-3	4.52E-1	4.3E-2
	FWADE	9.87E-4	4.5E-5	-1.03E+0	0.0E+0	3.00E+0	1.1E-3	2.43E-3	4.3 E-3	5.15E-3	1.02E-4	1.74E-1	4.2E-2
	ALEFWA	8.92E-4	3.4E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	9.22E-4	6.7E-4	9.82E-4	6.5E-5	8.47E-2	4.5E-3
6	PSO	4.67E-3	5.6E-4	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	4.86E-3	5.0E-3	5.61E-3	2.3E-3	6.31E-1	2.3E-2
	EFWA	2.41E-3	6.3E-4	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	5.40E-3	4.8 E-3	2.71E-3	1.03E-3	4.12E-1	6.3E-2
	FWADE	1.88E-3	8.6E-5	-1.03E+0	0.0E+0	3.00E+0	2.1E-3	4.38E-3	3.2 E-3	7.48E-3	2.3E-4	8.78E-2	6.3E-3
	ALEFWA	9,98E-4	3.4E-5	-1.03E+0	0.0E+0	3.00E+0	0.0E+0	5.38E-3	4.2 E-3	9.98E-4	3.4E-5	9.47E-2	8.9E-3



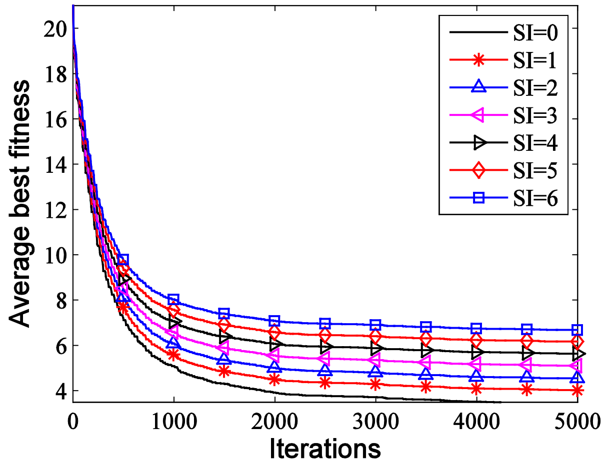
(a) EFWA

(b) ALEFWA

Fig. 2. Convergence curves of the EFWA and ALEFWA for F2



(a) EFWA



(b) ALEFWA

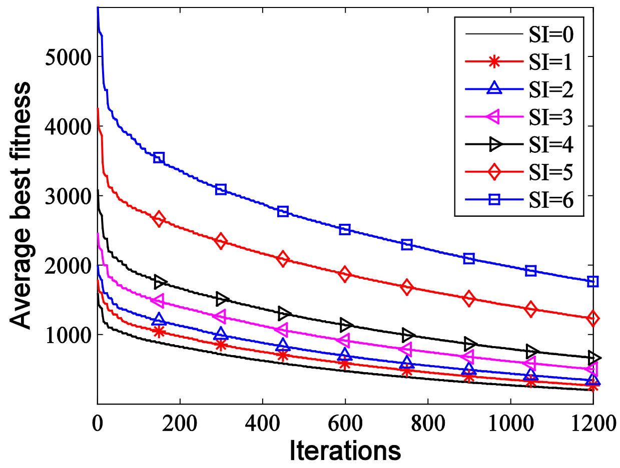
Fig. 3. Convergence curves of the EFWA and ALEFWA for F4

The convergence curves of EFWA and ALEFWA on functions F2, F4, and F5 are shown in Figs. 2–4, which show the convergence speeds of ALEFWA and EFWA for different SI. It can be seen from these three figures that the convergence speeds of EFWA and ALEFWA decline as the shifted value of the bench function increases, while the convergence speed and optimization accuracy of ALEFWA are better than those of EFWA. For function F4, the convergence performance of EFWA decreases sharply with the increase of the shifted value of bench function, while the convergence speed and accuracy of ALEFWA remain relatively stable.

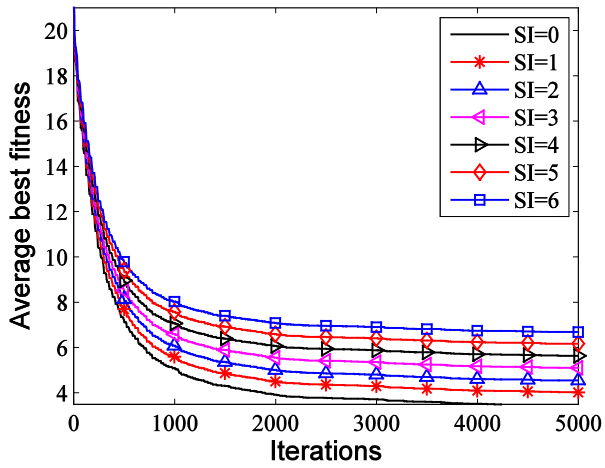
ALEFWA has high optimization precision and local search ability, probably because that at the final phase of the algorithm, its explosion amplitude reduction mechanism can reduce the amplitude to a very small value to perform accurate search in the local area, and the Lévy flight strategy gives ALEFWA better jumping ability to enhances the diversity of population.

Fig. 5 shows that the convergence curves of PSO, EFWA, FWADE, and ALEFWA on the 12 benchmark functions (SI = 4) averaged over 30 independent runs.

As shown in Fig. 5, the convergence performance of ALEFWA is better than that of PSO, EFWA, and FWADE. The main reason for the improvement of convergence performance in ALEFWA is because the search process in ALEFWA considers the heuristic information in the optimization process and adjusts the explosion amplitude dynamically, which can balance the local search and global search abilities.



(a) EFWA

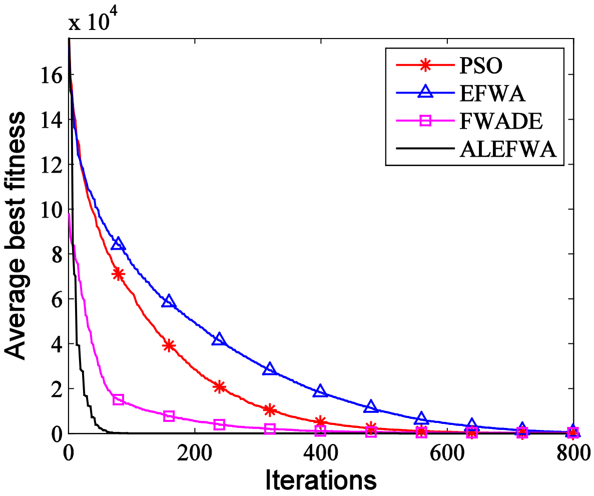


(b) ALEFWA

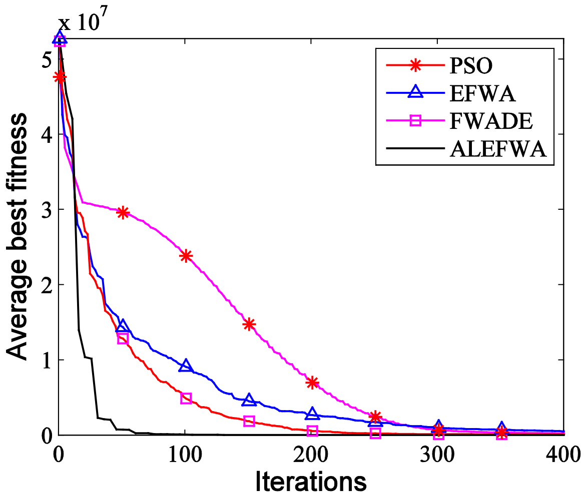
Fig.4. Convergence curves of the EFWA and ALEFWA for F5

V. Conclusion

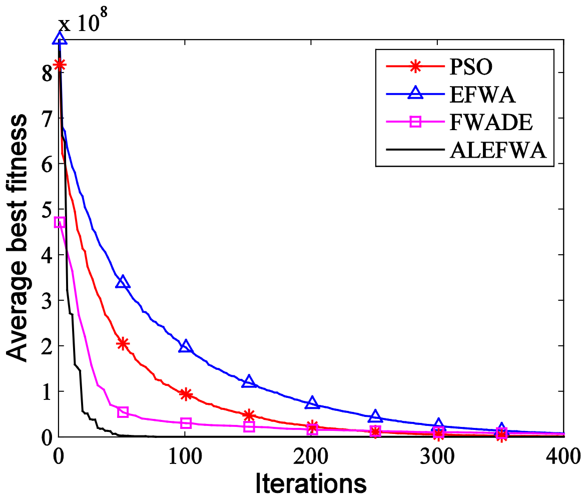
A novel adaptive explosion amplitude strategy is proposed to balance the local search and global search abilities of the EFWA, which can eliminate the waste of resources in FWA through the heuristic information in the optimization process. Furthermore, the Lévy flight strategy is introduced to calculate the positions of mutation sparks in order to enhance the diversity of population. Results of simulations on 12 benchmark functions and their shifted functions show that ALEFWA not only improves optimization precision, but it also achieves stable results for the shifted benchmark functions. ALEFWA also gains better convergence performance in solving high-dimensional complex optimization problems.



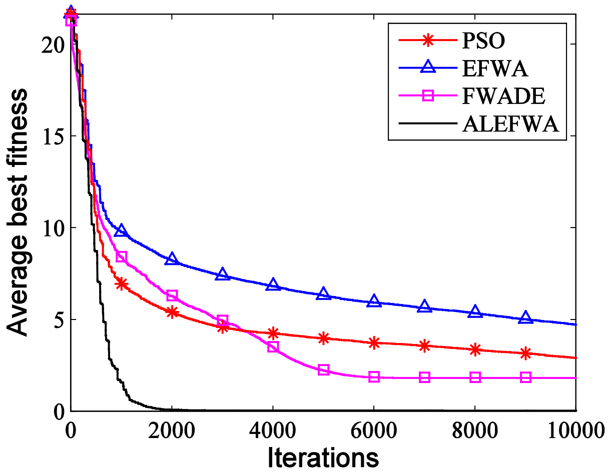
(a) Function F1



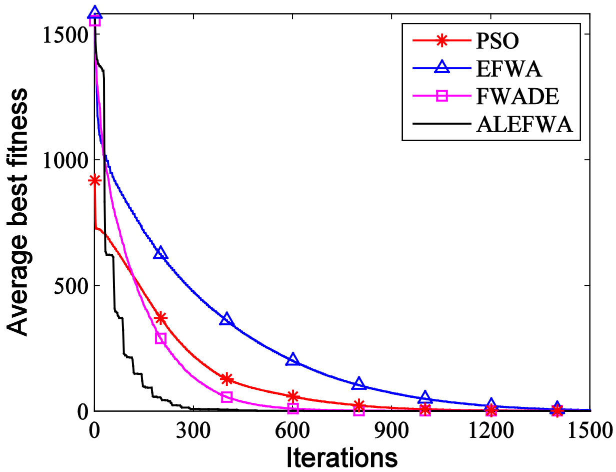
(b) Function F2



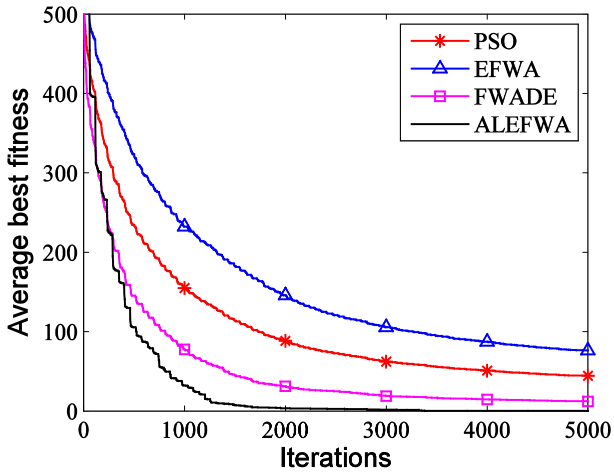
(c) Function F3



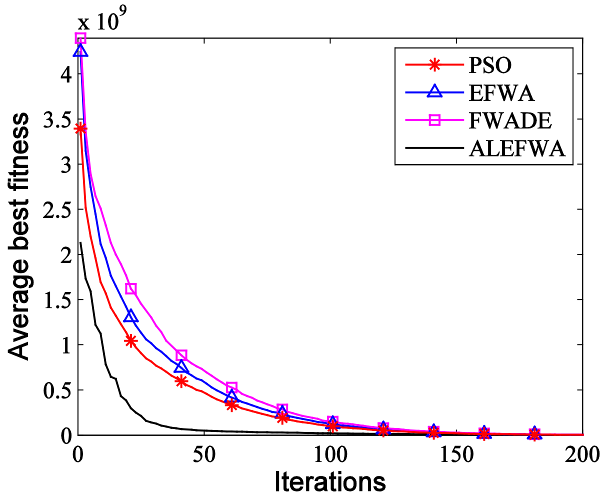
(d) Function F4



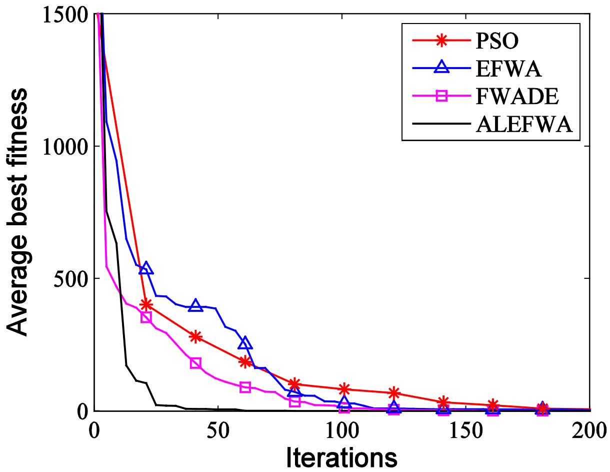
(e) Function F5



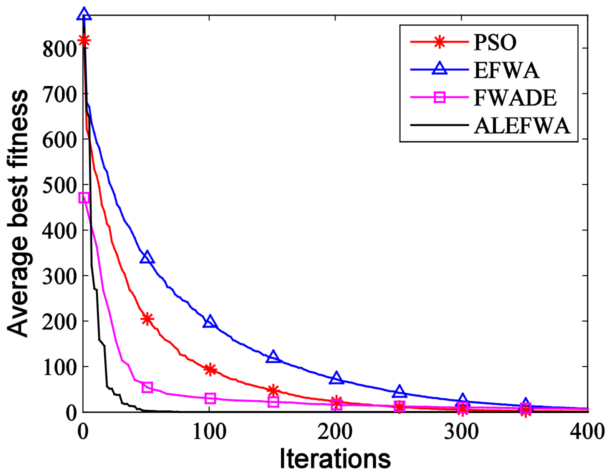
(f) Function F6



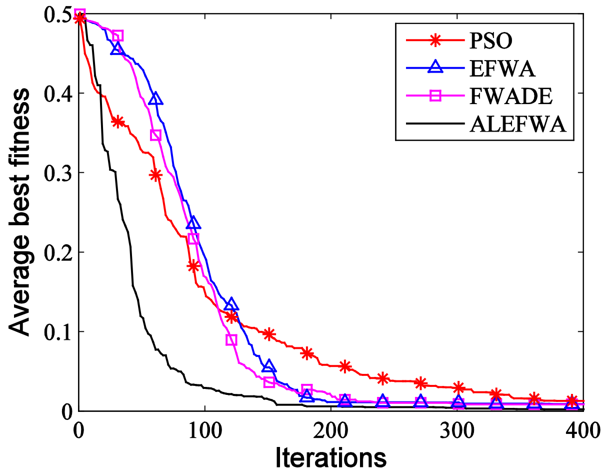
(g) Function F7



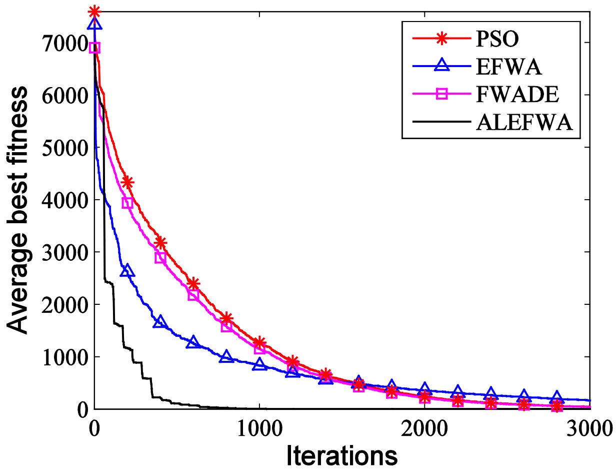
(h) Function F8



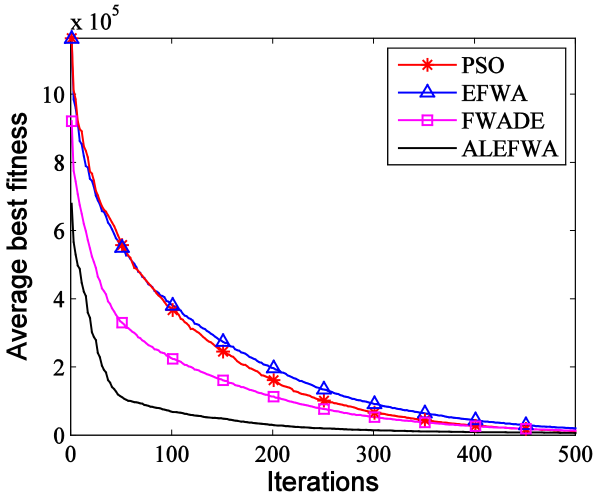
(i) Function F9



(j) Function F10



(k) Function F11



(l) Function F12

Fig. 5. Results of simulation of ALEFWA, FWADE, EFWA, and PSO

REFERENCES

[1] J. S. Wang, and J. D. Song, “A hybrid algorithm based on gravitational search and particle swarm optimization algorithm to solve function optimization problems,” *Engineering Letters*, vol. 25, no. 1, pp. 22-29, 2017.

[2] X. X. Ma, and J. S. Wang, “An improved flower pollination algorithm to solve function optimization problem,” *IAENG International Journal of Computer Science*, vol. 45, no. 3, pp. 364-370, 2018.

[3] Y. F. Sun, J. S. Wang, and X. X. Ma, “An improved krill herd algorithm based on meme grouping strategy of shuffled frog leaping algorithm,” *IAENG International Journal of Computer Science*, vol. 46, no. 2, pp. 156-162, 2019.

[4] Y. Tan, and Y. C. Zhu, “Fireworks algorithm for optimization,” *International Conference in Swarm Intelligence*, Beijing, pp. 355-364, 2010.

[5] Y. Xu. Yu, B. P. Zhao, T. H. Ma, and P. Wei, “A self-adaptive fireworks algorithm for classification problems,” *IEEE Access*, to be published.

[6] Y. Tan, C. Yu, S. Zheng, and K. Ding, “Introduction to fireworks algorithm,” *International Journal of Swarm Intelligence Research*, vol. 4, no. 4, pp. 39-70, 2013.

[7] S. Zheng, A. Janecek, and Y.Tan, “Enhanced fireworks algorithm,” *IEEE Congress on Evolutionary Computation*, Mexico, pp. 2069-2077, 2013.

[8] H. Gao, and M. Diao, “Cultural firework algorithm and its application for digital filters design,” *International Journal of Modelling Identification & Control*, vol. 14, no. 4, pp. 324-331, 2011.

[9] B. Zhang, Y. J. Zheng, M. X. Zhang, and S. Y. Chen, “Fireworks algorithm with enhanced fireworks interaction,” *IEEE/ACM Transactions on Computational Biology & Bioinformatics*, vol. 14, no.1, pp. 42-55, 2017.

[10] B. Juan, Luis Rodríguez, C. Oscar, M. Patricia, and V. Fevrier, “A new hybridization approach between the fireworks algorithm and grey wolf optimizer algorithm,” *Journal of Optimization*, vol. 2018, pp. 1-18, 2018.

[11] J. Guo, W. Liu, M. Liu, and S. Zheng, “Hybrid fireworks algorithm with differential evolution operator,” *International Journal of Intelligent Information and Database Systems*, vol. 12, no. 1/2, pp. 47-64, 2019.

[12] H. Zhao, C. Zhang, and J. Ning, “A best firework updating information guided adaptive fireworks algorithm,” *Neural Computing and Applications*, vol. 31, no.1, pp. 79-99, 2017.

[13] X. Han, L. Zheng, and L. Wang, “Fireworks algorithm based on dynamic search and tournament selection,” *International Journal of Computers and Applications*, to be published.

[14] G. M. Viswanathan, V. Afanasyev, S. V. Buldyrev, E. J. Murphy, and H. E. Stanley, “Lévy flight search patterns of wandering albatrosses,” *Nature*, vol.381, no. 6581, pp. 413-415, 1996.

[15] A. Layeb, “A novel quantum inspired cuckoo search for knapsack problems,” *International Journal of Bio-Inspired Computation*, vol. no.5, pp. 297-305, 2011.

[16] Q. X. Wang, and X. B. Guo, “Particle swarm optimization algorithm based on Lévy flight,” *Application Research of Computers*, vol. 33, no. 9, pp. 2588-2591, 2016.

[17] R. N. Mantegna, and H. E. Stanley, “Stochastic process with ultraslow convergence to a Gaussian: the truncated Lévy flight,” *Physical Review Letters*, vol.73, no. 22, pp. 4959, 1995.

Qiang Qu received an M.S. degree in control science and engineering from University of Science and Technology Liaoning, China in 2002, and a Ph.D. degree in signal and information processing from Dalian University of Technology, China in 2010. He is currently a professor and Master's supervisor in the School of Electronic and Information Engineering, University of Science and Technology Liaoning. His main research interests include modeling of complex industry processes, intelligent control, and swarm intelligence.

Mei-li Qi received a B.S. degree in electronic information engineering from University of Science and Technology Liaoning, China in 2018. She is working toward an M.S. degree in control science and engineering from University of Science and Technology Liaoning, China. Her main research interests include swarm intelligence and intelligent control.

Ru-bing Gong received a B.S. degree in electronic information engineering from University of Science and Technology Liaoning, China in 2018. He is working toward an M.S. degree in control science and engineering from University of Science and Technology Liaoning, China. His main research interests include swarm intelligence and intelligent control.

Xue-bo Chen received an M.S. degree from University of Science and Technology Liaoning, China in 1986, and received a Ph.D. degree in technical science from University of Belgrade, Yugoslavia in 1994. He is currently a professor and Ph.D. supervisor in University of Science and Technology Liaoning. His research interests include pattern recognition, large complex systems, and intelligent control.