Improving Enhanced Fireworks Algorithm with New Gaussian Explosion and Population Selection Strategies*

Bei Zhang, Minxia Zhang, and Yu-Jun Zheng

College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China
zhangbei-zjut@outlook.com,zmx@zjut.edu.cn,yujun.zheng@computer.org

Abstract. Fireworks algorithm (FWA) is a relatively new metaheuristic in swarm intelligence and EFWA is an enhanced version of FWA. This paper presents a new improved method, named IEFWA, which modifies EFWA in two aspects: a new Gaussian explosion operator that enables new sparks to learn from more exemplars in the population and thus improves solution diversity and avoids being trapped in local optima, and a new population selection strategy that enables high-quality solutions to have high probabilities of entering the next generation without incurring high computational cost. Numerical experiments show that the IEFWA algorithm outperforms EFWA on a set of benchmark function optimization problems.

Keywords: global optimization, fireworks algorithm (FWA), Gaussian explosion, population selection.

1 Introduction

Initially proposed by Tan and Zhu [1], fireworks algorithm (FWA) is a relatively new nature-inspired optimization method mimicking the explosion process of fireworks for optimization problems. In FWA, a solution to the problem is analogous to a firework or a spark, and an explosion is analogous to a stochastic search in the solution space around the firework. By explosion, fireworks with better fitness tend to generate more sparks within smaller explosion ranges in order to intensify local search, while fireworks with worse fitness generate fewer sparks within larger explosion ranges to facilitate global search, as illustrated in Fig. 1 [1]. Numerical experiments on a set of benchmark functions show that, FWA has more rapid convergence speed than some typical particle swarm optimization (PSO) algorithms such as [2] and [3].

Since its proposal, FWA has attracted much attention. Zheng et al. [4] developed a new hybrid FWA by combining it with differential evolution (DE) [5], which selects new fireworks for the next generation from highly ranked solutions

* Supported by grants from National Natural Science Foundation (No. 61105073) and Zhejiang Provincial Natural Science Foundation (No. LY14F030011) of China.

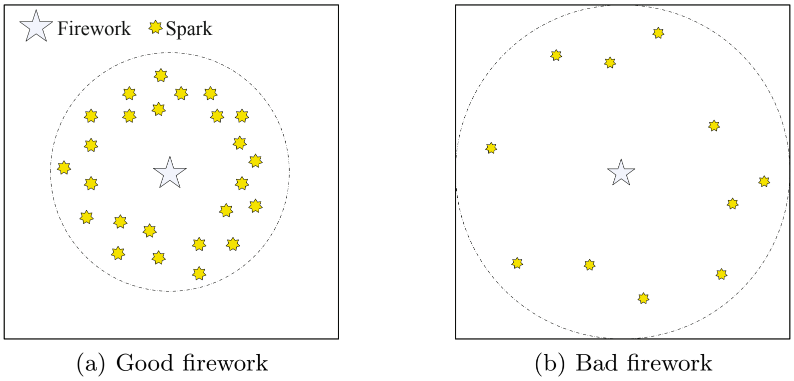


Fig. 1. Illustration of fireworks explosions in FWA

and updates these newly generated solutions with the DE mutation, crossover, and selection operators. Pei et al. [6] studied the influence of approximation model, sampling method, and sampling number on the acceleration performance of FWA, and improved the algorithm by using an elite strategy for enhancing the search capability. The hybrid algorithm proposed by Zhang et al. [7] introduces the migration operator of biogeography-based optimization (BBO) [8] to FWA, which can effectively enhance information sharing among the population, and thus improves solution diversity and avoids premature convergence.

In [9] Ding et al. studied the parallel implementation of FWA on GPUs. In [10] Zheng et al. presented a multiobjective version of FWA, which has shown the great success for variable-rate fertilization in oil crop production. FWA has also been successfully applied to many other practical problems [11,12,13,14].

An important improvement to FWA is the enhanced FWA (EFWA) proposed by Zheng et al. [15], which tackles the limitations of the original FWA by developing new explosion operators, new strategy for selecting population for the next generation, new mapping strategy for sparks out of the search space, and new parameter mechanisms. Consequently, EFWA outperforms FWA in terms of convergence capabilities, meanwhile reducing the runtime significantly.

In this paper, we further improve EFWA in two aspects, i.e., the Gaussian explosion operator and the population selection strategy. We develop a new Gaussian explosion operator that enables new sparks to learn from more exemplars in the population, and thus improves solution diversity and avoids being trapped in local optima. Moreover, we propose a new population selection strategy that enables high-quality solutions to have high probabilities of entering the next generation, without incurring high computational cost. Numerical experiments show that the proposed algorithm, named IEFWA, outperforms EFWA on a set of benchmark function optimization problems.

In the rest of the paper, Section 2 introduces FWA and EFWA, Section 3 describes IEFWA in detail, Section 4 presents the experiments, and Section 5 concludes.

2 FWA and EFWA

FWA is a metaheuristic optimization method inspired by the phenomenon of fireworks explosion, where a solution to the problem is analogous to a firework or a spark. The key principle of FWA is that fitter fireworks can explode more sparks within a smaller area, while worse fireworks generate fewer sparks within a larger amplitude. The basic framework of FWA is as follows:

1. Randomly initialize a certain number of locations to set off fireworks.
2. For each firework, perform a *regular explosion* operation to generate a set of sparks.
3. Select a small number of fireworks, on each of which perform a *Gaussian explosion* operation to generate a few number of sparks.
4. Choose individuals from the current generation of fireworks and sparks to enter into the next generation.
5. Repeat Step 2-4 until the termination condition is satisfied.

For regular explosion of FWA, the number of sparks s_i and the explosion amplitude A_i of the ith firework X_i are respectively calculated as follows:

$$s_i = M_e \cdot \frac{f_{\max} - f(X_i) + \epsilon}{\sum_{j=1}^{n}(f_{\max} - f(X_j)) + \epsilon} \tag{1}$$

$$A_i = \widehat{A} \cdot \frac{f(X_i) - f_{\min} + \epsilon}{\sum_{j=1}^{n}(f(X_j) - f_{\min}) + \epsilon} \tag{2}$$

where n is the size of population, $f(X_i)$ is the fitness value of X_i, $f_{\max}$ and $f_{\min}$ are respectively the maximum and minimum fitness values among the n fireworks, M_e and $\widehat{A}$ are two parameters for respectively controlling the total number of sparks and the maximum explosion amplitude, and ϵ is a small constant to avoid zero-division-error.

To avoid overwhelming effects of splendid fireworks, the number of sparks is further bounded as follows (where $s_{\min}$ and $s_{\max}$ are control parameters and *round* rounds a number to its closest integer):

$$s_i = \begin{cases} s_{\min} & \text{if } s_i < s_{\min} \\ s_{\max} & \text{if } s_i > s_{\max} \\ round(s_i) & \text{else} \end{cases} \tag{3}$$

According to Eq. (2), the explosion amplitude may be too small for very good fireworks. EFWA tackles this issue by setting at each dimension d a lower limit of explosion amplitude $A_{\min}^{d}$, which decreases with the number of generations (or function evaluations) t as follows:

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

where A_{init} and A_{final} are respectively the initial and final minimum explosion amplitude, and $t_{\max}$ is the maximum number of generations (or function evaluations).

When performing a regular explosion on X_i, FWA computes an offset displacement as $rand(-1,1) \cdot A_i$, which is added to z random dimensions of X_i. EFWA modifies the explosion operator by computing a different displacement value for each dimension d:

$$X_j^d = X_i^d + rand(-1,1) \cdot A_i^d \tag{5}$$

Gaussian explosion of FWA is mainly used for improving solution diversity. At each generation, FWA randomly chooses a small number M_g of fireworks, and for each firework obtains a Gaussian spark by computing its location as:

$$X_j^d = X_i^d \cdot N(1,1) \tag{6}$$

where $N(1,1)$ is a Gaussian random number with mean 1 and standard deviation 1. However, such a mechanism often causes many sparks to be very close to the origin of the search space, and fireworks already close to the origin cannot escape from this location. To tackle this, EFWA uses the following Gaussian explosion that makes the new spark learn from the best individual X_{best} found so far:

$$X_j^d = X_i^d + (X_{\text{best}}^d - X_i^d) \cdot N(0,1) \tag{7}$$

Algorithm 1 and Algorithm 2 respectively describes the regular explosion and Gaussian explosion procedures used in EFWA, where X_i denotes the firework to be exploded, and D is the dimension of the problem.

Algorithm 1. The regular explosion used in EFWA.

```
1 Calculate $s_i$ and $A_i$ for firework $X_i$;
2 for $k = 1$ to $s_i$ do
3     Initialize a spark $X_j = X_i$;
4     for $d = 1$ to $D$ do
5         if $rand(0,1) < 0.5$ then
6             $X_j^d = X_i^d + rand(-1,1) \cdot A_i^d$;
7             if $X_j^d$ is out of the search range then
8                 Randomly set $X_j^d$ in the search range;
9     Add $X_j$ as a new spark.
```

Algorithm 2. The Gaussian explosion in EFWA.

```
1 Initialize a spark $X_j = X_i$;
2 for $d = 1$ to $D$ do
3     if $rand(0,1) < 0.5$ then
4         $X_j^d = X_i^d + (X_{\text{best}}^d - X_i^d) \cdot N(0,1)$;
5         if $X_j^d$ is out of the search range then
6             Randomly set $X_j^d$ in the search range;
7     Return $X_j$ as a new Gaussian spark.
```

At each generation, the best individual among all the sparks and fireworks is always chosen to the next generation. In FWA, the other $(n-1)$ fireworks are selected according to the probabilities proportional to their distances to other individuals, which is computationally expensive. Therefore, EFWA employs a very simple strategy that randomly selects the remaining $(n-1)$ fireworks.

3 An Improved EFWA

The proposed IEFWA intends to improve EFWA in two aspects: the Gaussian explosion operator and the population selection strategy.

3.1 A New Gaussian Explosion Operator

As shown in Eq. (6), the Gaussian explosion of EFWA makes the new spark learn from the best individual X_{best} found so far. This is similar to the mechanism of learning from the global best in PSO [2]. Thus EFWA also partly suffers the problem of premature convergence as PSO: If the X_{best} is just a local optimum or very close to it, the Gaussian sparks will be heavily attracted by the local optimum; if there is no new best solution found for a certain number of generations, more and more sparks will converge to the local optimum, and the algorithm is easily trapped.

To tackle this issue, we develop a new Gaussian explosion operator that enables new Gaussian sparks to learn from not only the current best but also other exemplars in the population. Our tactic is very simple: When exploding a firework X_i, at each dimension d, we first randomly choose two individuals from the current population, and then select the one with higher fitness value, denoted by X_{lbest}, as the exemplar that replaces X_{best} in Eq. (7):

$$X_j^d = X_i^d + (X_{\text{lbest}}^d - X_i^d) \cdot N(0,1) \tag{8}$$

In this way, every Gaussian spark has a chance to learn from different exemplars at different dimensions, and thus the solution diversity can be increased greatly. The combination of the regular explosion of EFWA and the new Gaussian explosion of IEFWA can balance the exploration and exploitation much more effectively. Algorithm 3 presents the new Gaussian explosion procedure.

Algorithm 3. The Gaussian explosion in IEFWA.

```
1  Initialize a spark $X_j = X_i$;
2  for $d = 1$ to $D$ do
3     if $rand(0,1) < 0.5$ then
4        Randomly choose two individuals from the population;
5        Set $X_{\text{lbest}}$ to the better one between them;
6        $X_j^d = X_i^d + (X_{\text{lbest}}^d - X_i^d) \cdot N(0,1)$;
7        if $X_j^d$ is out of the search range then
8           Randomly set $X_j^d$ in the search range;
9     Return $X_j$ as a new Gaussian spark.
```

3.2 A New Population Selection Strategy

When selecting individuals to the next generation, the original FWA uses a distance-based selection strategy, which is effective in terms of population diversity but incurs high computational cost. EFWA employs a random selection strategy that is computationally efficient but may lose some high-quality individuals which may ultimately lead to the global optimum.

In general, we want that high-fitness individuals have high selection probability, while low-fitness individuals still have chances of entering into the next generation. Here we employ the pairwise comparison method used in evolutionary programming (EP) [16] to determine whether an individual should survive to the next generation. That is, for each individual, we randomly choose q opponents from the current generation, and conduct q pairwise comparisons between the test individual and the opponents. If the individual is fitter, it receives a "win". Finally, among all the individuals in the current generation, n individuals that have the most wins enter into the next generation.

EP has a fixed population size and uses a fixed q value of about 10∼20. However, in FWA the total number of fireworks and sparks often varies from generation to generation, and thus we set q to a random value in the range [1,np], where np is the total number of individuals in the current generation. Algorithm 4 presents the population selection procedure used in IEFWA.

Algorithm 4. The population selection in IEFWA.

```
1 Let $q = rand(1, np)$;
2 for each $X_i$ in the current generation do
3     Let $Wins(X_i) = 0$;
4     Randomly choose $q$ opponents from the population;
5     for each opponent $X'_i$ do
6         if $f(X_i) > f(X'_i)$ then
7             $Wins(X_i) \leftarrow Wins(X_i) + 1$;
8 Sort $np$ individuals in decreasing order of $Wins(X_i)$;
9 Return the first $n$ individuals.
```

4 Computational Experiment

4.1 Experimental Settings

We test the performance of the proposed IEFWA on a set of 18 benchmark functions denoted as f_1–f_{18}, which are summarized in Table 1. Here f_1–f_{13} include unimodal and simple multimodal functions taking from [17], and f_{14}–f_{18} are shifted and rotated (SR) functions taking from [18]. All the functions are high-dimensional problems, and in this paper we use 30-D problems.

To evaluate our new strategies, besides the proposed IEFWA that uses both the new Gaussian explosion operator and the new population selection strategy, we also implement another version that uses only the new Gaussian explosion

Table 1. A summary of the benchmark functions used in the paper (for f_{14}–f_{18}, $\boldsymbol{M}$ is the rotation matrix and $\boldsymbol{o}_i$ is the shifted global optimum [18])

Name	Function	Range
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$
Schwefel 2.22	$f_2(x) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$	$[-10, 10]^D$
Schwefel 1.2	$f_3(x) = \sum_{i=1}^{D} \big(\sum_{j=1}^{i} x_j\big)^2$	$[-100, 100]^D$
Schwefel 2.21	$f_4(x) = \max_i \{\lvert x_i\rvert, 1 \le i \le D\}$	$[-100, 100]^D$
Rosenbrock	$f_5(x) = \sum_{i=2}^{D-1} (100(x_i^2 - x_{i-1})^2 + (x_i - 1)^2)$	$[-30, 30]^D$
Step	$f_6(x) = \sum_{i=1}^{D} \big(\lfloor x_i + 0.5\rfloor\big)^2$	$[-100, 100]^D$
Quartic	$f_7(x) = \sum_{i=1}^{D} i x_i^4 + rand[0, 1)$	$[-1.28, 1.28]^D$
Schwefel	$f_8(x) = 418.9829 \times D - \sum_{i=1}^{D} x_i \sin(\lvert x_i\rvert^{\frac{1}{2}})$	$[-500, 500]^D$
Rastrigin	$f_9(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^D$
Ackley	$f_{10}(x) = -20\exp\Bigg(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\Bigg) - \exp\big(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\big) + 20 + e$	$[-32, 32]^D$
Griewank	$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^D$
Penalized1	$f_{12}(x) = \frac{\pi}{30}\big(10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2(1 + 10\sin^2(\pi y_{i+1})) + (y_D - 1)^2\big) + \sum_{i=1}^{D-1} u(x_i, 10, 100, 4)$	$[-50, 50]^D$
Penalized2	$f_{13}(x) = 0.1\big(\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2(1 + \sin^2(3\pi x_{i+1})) + (x_D - 1)^2(1 + \sin^2(2\pi x_D))\big) + \sum_{i=1}^{D-1} u(x_i, 5, 100, 4)$ where $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]^D$
SR Bent Cigar	$f_{14}(\boldsymbol{x}) = 200 + y_1^2 + 10^6 \sum_{i=2}^{D} y_i^2, \quad \boldsymbol{y} = \boldsymbol{M}(\boldsymbol{x} - \boldsymbol{o}_1)$	$[-100, 100]^D$
SR Discus	$f_{15}(\boldsymbol{x}) = 300 + 10^6 y_1^2 + \sum_{i=2}^{D} y_i^2, \quad \boldsymbol{y} = \boldsymbol{M}(\boldsymbol{x} - \boldsymbol{o}_2)$	$[-100, 100]^D$
SR Ackley	$f_{16}(\boldsymbol{x}) = 500 + f_{10}(\boldsymbol{y}), \quad \boldsymbol{y} = \boldsymbol{M}(\boldsymbol{x} - \boldsymbol{o}_3)$	$[-100, 100]^D$
SR Griewank	$f_{17}(\boldsymbol{x}) = 700 + f_{11}(\boldsymbol{y}), \quad \boldsymbol{y} = \boldsymbol{M}\big(\frac{600(\boldsymbol{x} - \boldsymbol{o}_4)}{100}\big)$	$[-100, 100]^D$
SR Rastrigin	$f_{18}(\boldsymbol{x}) = 900 + f_9(\boldsymbol{y}), \quad \boldsymbol{y} = \boldsymbol{M}\big(\frac{5.12(\boldsymbol{x} - \boldsymbol{o}_5)}{100}\big)$	$[-100, 100]^D$

operator, denoted as IEFWA1. For the sake of fair comparison, the maximum number of function evaluations (NFE) is set to 200,000 for every problem.

We have compared EFWA, IEFWA1, and IEFWA on the 18 test problems, using $n = 5$, $M_e = 50$, $M_g = 5$, $\widehat{A} = 40$, $s_{\max} = 40$, $s_{\min} = 2$, $A_{init} = 0.02(X_{\max} - X_{\min})$ and $A_{final} = 0.001(X_{\max} - X_{\min})$, as suggested in [1] and [15]. The experiments are conducted on a computer of Intel Core i5-2520M processor and 4GB DDR3 memory. Each algorithm has been run 60 times (with different random seeds) on each problem.

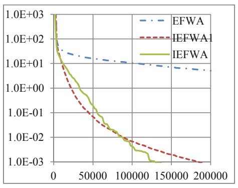
4.2 Experimental Results

Table 2 presents the mean and standard deviation of the best fitness values obtained by the three algorithms averaged over 60 runs. The bold values indicate the best results among the three algorithms. We have also conducted paired *t*-tests between IEFWA and the other two algorithms, and mark $^+$ before the mean values in columns 2 and 4 if IEFWA has statistically significant performance improvement over the corresponding algorithms (at 95% confidence level).

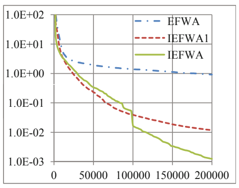
Table 2. The experimental results of the three algorithms

ID	FWA		IEFWA1		IEFWA	
	mean	std	mean	std	mean	std
f_1	$^+$5.01E+00	(9.01E−01)	$^+$7.48E−04	(3.51E−04)	**5.32E−05**	(5.10E−05)
f_2	$^+$9.17E−01	(1.49E−01)	$^+$1.18E−02	(2.35E−03)	**1.25E−03**	(8.58E−04)
f_3	$^+$6.23E+01	(1.45E+01)	$^+$3.79E−02	(1.23E−02)	**1.98E−03**	(1.63E−03)
f_4	$^+$9.55E−01	(6.79E−02)	**2.01E−01**	(4.29E−02)	3.29E−01	(3.30E−01)
f_5	$^+$2.18E+02	(2.57E+02)	$^+$1.00E+02	(1.57E+02)	**5.34E+01**	(3.29E+01)
f_6	$^+$3.87E+00	(9.99E−01)	**0.00E+00**	(0.00E+00)	**0.00E+00**	(0.00E+00)
f_7	$^+$1.83E−02	(1.36E−02)	$^+$1.75E−02	(1.18E−02)	**8.22E−03**	(5.68E−03)
f_8	5.30E+03	(8.56E+02)	**5.12E+03**	(7.47E+02)	5.20E+03	(4.37E+02)
f_9	$^+$1.27E+02	(2.02E+01)	**1.83E+01**	(5.95E+00)	9.55E+01	(1.90E+01)
f_{10}	$^+$1.16E+00	(1.72E−01)	$^+$1.88E−02	(5.88E−01)	**1.80E−03**	(1.21E−03)
f_{11}	$^+$1.03E+00	(2.21E−02)	$^+$2.48E−02	(2.20E−02)	**9.90E−03**	(8.87E−03)
f_{12}	$^+$1.12E+01	(2.77E+00)	$^+$1.42E+00	(1.59E+00)	**9.35E−01**	(9.43E−01)
f_{13}	$^+$8.18E−01	(2.39E−01)	$^+$9.67E−05	(1.33E−04)	**2.76E−05**	(5.98E−05)
f_{14}	$^+$6.86E+06	(1.27E+06)	$^+$8.87E+03	(1.09E+04)	**5.90E+03**	(1.25E+04)
f_{15}	$^+$3.99E+02	(2.76E+01)	$^+$3.53E+02	(2.25E+01)	**3.35E+02**	(1.77E+01)
f_{16}	$^+$5.21E+02	(7.56E−02)	$^+$5.20E+02	(4.78E−03)	**5.20E+02**	(1.82E−03)
f_{17}	$^+$9.58E+06	(1.60E+06)	$^+$1.02E+04	(3.64E+03)	**3.07E+03**	(2.05E+03)
f_{18}	$^+$1.06E+03	(4.80E+01)	1.02E+03	(5.00E+01)	**1.01E+03**	(3.12E+01)

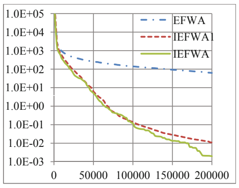
As we can see from the results, EFWA never obtains the best mean value on any of the 18 problems, and the two IEFWA versions both achieve considerable performance improvement over EFWA. In terms of statistical tests, except that on f_8 there is no significant difference between EFWA and IEFWA, on the



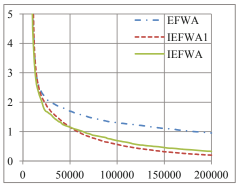
(a) f_1



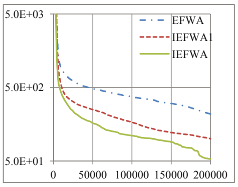
(b) f_2



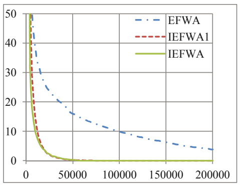
(c) f_3



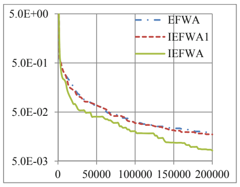
(d) f_4



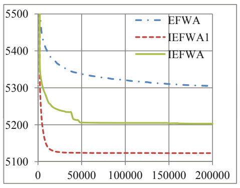
(e) f_5



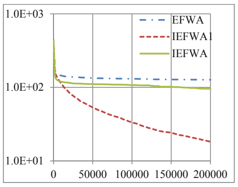
(f) f_6



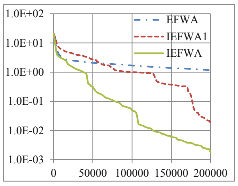
(g) f_7



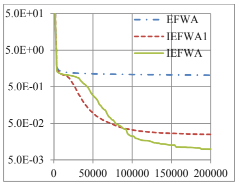
(h) f_8



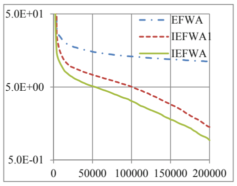
(i) f_9



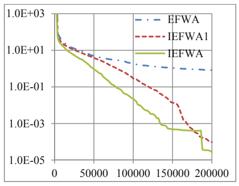
(j) f_{10}



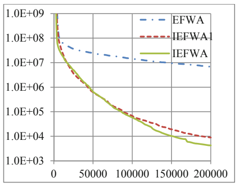
(k) f_{11}



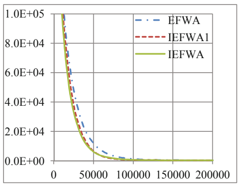
(l) f_{12}



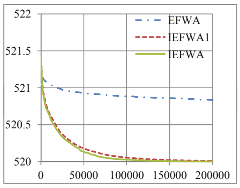
(m) f_{13}



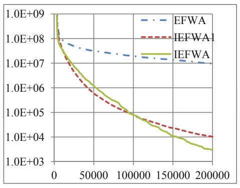
(n) f_{14}



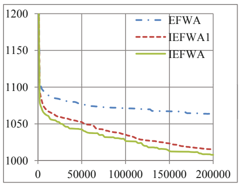
(o) f_{15}



(p) f_{16}



(q) f_{17}



(r) f_{18}

Fig. 2. Convergence curves of the comparative algorithms on the test problems

remaining 17 problems the performance of IEFWA is always significantly better than EFWA. This shows that our new Gaussian explosion operator is effective in improving the search ability of the algorithm.

By comparing IEFWA1 and IEFWA, the former obtains the best mean values on 4 problems, and the latter does so on 15 problems (they both reach the optimum on f_6). Statistical test results show that IEFWA has significant performance improvement over IEFWA1 on 13 problems. In particular, IEFWA always obtains the best mean values on the 5 shifted and rotated problems (f_{14}–f_{18}), and has significant performance improvement over IEFWA1 on 4 problems. This also demonstrates the effectiveness of our comparison-based population selection strategy, especially on complex test problems.

Fig. 2(a)–(t) respectively present the convergence curves of the algorithms on the 18 test problems. As we can see, the two IEFWA versions converges much better than EFWA on most of the problems, except that on f_7 the curves of EFWA and IEFWA1 overlap to a great extent, (but IEFWA converges faster and reaches a better result). On f_{10} EFWA and IEFWA both converge faster than IEFWA1 in the early stage but the curve of EFWA soon becomes very flat and overtaken by IEFWA1, while IEFWA still keeps a fast convergence speed at later stages and reaches a much better result. This is because the Ackley function has multiple local optima, where EFWA is easily trapped, but the two IEFWA versions are capable of jumping out of the local optima. On most other problems, the IEFWA versions not only converge faster than EFWA, but also have their curves falling for long periods where EFWA has been already trapped.

On the other hand, the convergence curves of IEFWA1 and IEFWA have similar shapes on many problems, but IEFWA often converges much faster than IEFWA1. This also demonstrates that the new population selection strategy contributes to the increase of the convergence speed.

In summary, the experimental results show that IEFWA has obvious advantages in convergence speed and solution accuracy, which demonstrates that our Gaussian explosion operator can greatly improve the solution diversity and thus effectively avoid premature convergence, and the new population selection strategy can efficiently accelerate the search. The combination of the two strategies provides a much better balance of exploration and exploitation than EFWA.

5 Conclusion

EFWA is a major improvement of the original FWA. The proposed IEFWA further uses a new Gaussian explosion operator that provides a more comprehensive learning mechanism to increase solution diversity, and employs a new population selection strategy to accelerate the search. Computational experiments show that IEFWA outperforms EFWA in both convergence speed and solution accuracy on a set of well-known benchmark functions. Ongoing work includes testing the new strategies for constrained and/or multiobjective optimization problems, and hybridizing the proposed IEFWA with other heuristics.

References

1. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) ICSI 2010, Part I. LNCS, vol. 6145, pp. 355–364. Springer, Heidelberg (2010)
2. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceeding of the IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
3. Tan, Y., Xiao, Z.: Clonal particle swarm optimization and its applications. In: Proceeding of the IEEE Congress on Evolutionary Computation, pp. 2303–2309 (2007)
4. Zheng, Y.J., Xu, X.L., Ling, H.F., Chen, S.Y.: A hybrid fireworks optimization method with differential evolution operators. Neurocomputing (2014), doi:10.1016/j.neucom.2012.08.075
5. Storn, R., Price, K.: Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11(4), 341–369 (1997)
6. Pei, Y., Zheng, S., Tan, Y., et al.: An empirical study on influence of approximation approaches on enhancing fireworks algorithm. In: Proceeding of the 2012 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1322–1327 (2012)
7. Zhang, B., Zhang, M.X., Zheng, Y.J.: A hybrid biogeography-based optimization and fireworks algorithm. In: Proceeding of the IEEE Congress on Evolutionary Computation (2014)
8. Simon, D.: Biogeography-based optimization. IEEE Trans. Evol. Comput. 12(6), 702–713 (2008)
9. Ding, K., Zheng, S., Tan, Y.: A GPU-based parallel fireworks algorithm for optimization. In: Proceeding of the 15th Annual Conference on Genetic and Evolutionary Computation Conference, pp. 9–16 (2013)
10. Zheng, Y.J., Song, Q., Chen, S.Y.: Multiobjective fireworks optimization for variable-rate fertilization in oil crop production. Applied Soft Computing 13(11), 4253–4263 (2013)
11. Gao, H., Diao, M.: Cultural firework algorithm and its application for digital filters design. International Journal of Modelling, Identification and Control 14(4), 324–331 (2011)
12. Janecek, A., Tan, Y.: Iterative improvement of the multiplicative update nmf algorithm using nature-inspired optimization. In: Proceeding of the 7th International Conference on Natural Computation, vol. 3, pp. 1668–1672 (2011)
13. Janecek, A., Tan, Y.: Using population based algorithms for initializing nonnegative matrix factorization. In: Tan, Y., Shi, Y., Chai, Y., Wang, G. (eds.) ICSI 2011, Part II. LNCS, vol. 6729, pp. 307–316. Springer, Heidelberg (2011)
14. He, W., Mi, G., Tan, Y.: Parameter optimization of localconcentration model for spam detection by using fireworks algorithm. In: Tan, Y., Shi, Y., Mo, H. (eds.) ICSI 2013, Part I. LNCS, vol. 7928, pp. 439–450. Springer, Heidelberg (2013)
15. Zheng, S., Janecek, A., Tan, Y.: Enhanced fireworks algorithm. In: Proceeding of the IEEE Congress on Evolutionary Computation, pp. 2069–2077 (2013)
16. Back, T., Schwefel, H.P.: An overview of evolutionary algorithms for parameter optimization. Evolutionary Computation 1(1), 1–23 (1993)
17. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Trans. Evol. Comput. 3(2), 82–102 (1999)
18. Liang, J.J., Qu, B.Y., Suganthan, P.N.: Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Tech. Rep. 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China (2014)