

# Enhancing firefly algorithm using generalized opposition-based learning

Shuhao Yu<sup>1,2</sup> · Shenglong Zhu<sup>3</sup> · Yan Ma<sup>1</sup> ·  
Demei Mao<sup>1</sup>

Received: 21 August 2014 / Accepted: 13 April 2015 / Published online: 25 April 2015  
© Springer-Verlag Wien 2015

**Abstract** Firefly algorithm has been shown to yield good performance for solving various optimization problems. However, under some conditions, FA may converge prematurely and thus may be trapped in local optima due to loss of population diversity. To overcome this defect, inspired by the concept of opposition-based learning, a strategy to increase the performance of firefly algorithm is proposed. The idea is to replace the worst firefly with a new constructed firefly. This new constructed firefly is created by taken some elements from the opposition number of the worst firefly or the position of the brightest firefly. After this operation, the worst firefly is forced to escape from the normal path and can help it to escape from local optima. Experiments on 16 standard benchmark functions show that our method can improve accuracy of the basic firefly algorithm.

**Keywords** Firefly algorithm · Premature convergence · Local optimum · Opposition-based learning

**Mathematics Subject Classification** 68Qxx · 68Txx · 68Uxx · 68Wxx · 65Yxx

## 1 Introduction

In recent years, many nature-inspired algorithms have been developed as powerful techniques to solve the optimization problems. The main reason is that it take too

---

✉ Demei Mao  
demeimao@126.com

<sup>1</sup> School of Information Engineering, West of Anhui University, Lu'an 237012, China

<sup>2</sup> Institute of Computer Network Systems, Hefei University of Technology, Hefei 230009, China

<sup>3</sup> Anhui Electric Power Science and Research Institute, Hefei 230022, China

much time to solve optimization problems with traditional methods, and that they cannot be solved efficiently [1].

Firefly algorithm (FA) is a robust stochastic optimization and population-based technique that developed by Yang [2]. It is inspired by the flashing lights of fireflies in nature. Previous studies show that the FA can perform superiorly, compared with genetic algorithm and particle swarm optimization [2,3], and it is applicable for mixed variable and engineering optimization[4–10]. In [11], it presented FA to minimize the real power losses and to improve the voltage profile. In [12], it studied the selection method of multi-parameter geomagnetic matching area and adopted the FA. In [13] they used FA to optimize the electrical discharge machining parameters and in [14] they used a modified FA to optimize planning of distributed generators in distribution networks.

Though FA is a successful evolutionary-based optimization algorithm, the results from experiments running FA have shown that it can either be mostly stuck into the local minimum (premature convergence) or the results do not improved furthermore (stagnation) [15]. FA suffers from premature convergence because it tends to be trapped in the local optima at the early stage which results in a low optimization precision or even failure. In addition, exploration and exploitation capabilities are two key aspects in the design of efficient. The FA should have strong balanced exploration and exploitation capabilities to enhance the searching capability of the basic FA. The cause of it may due to the initial adjusting parameters as other evolutionary-based optimization algorithms[16].

To overcome the defect of the basic FA, many researchers have tried to modify the basic FA in order to achieve a better accuracy and higher convergence speed. Such as fuzzy FA [17], self-adaptive step FA [18], chaotic FA [19], jumper FA [20], Lévy-flight FA [21], etc.

In this paper, inspired by the concept of opposition-based learning, we propose a modified FA to enhance the performance of the basic FA. The strategy used here is to replace the worst firefly with a new constructed firefly. The new firefly is created by taken some elements from the best firefly or the opposition number of the worst firefly. Numerical results show that the proposed algorithm is valid and can achieve better solution quality.

The remainder of this paper is organized as follows: Sect. 2 briefly presents the standard firefly algorithm and the concept of opposition-based learning. Section 3 explains the proposed firefly algorithm. Section 4 provides the experimental settings and discusses the experimental results. Section 5 concludes the paper.

## 2 Related work

In this section, we discuss the mechanism of the basic FA. Next, the state-of-art FA variants are reviewed. The symbols used in this paper are summarized in Table 1.

### 2.1 Basic firefly algorithm

Firefly algorithm is one of the powerful optimization algorithms. Most of the optimization problems, such as NP-hard problems, which have very high computational

**Table 1** Mathematical symbols used in this paper

Symbol	Meaning
$r_{ij}$	The distance between firefly $i$ and $j$
$I_i$	The light intensity
$d$	The dimension of the object function
$n$	The number of fireflies
$X$	The position of a firefly
$x'$	The opposition number of $X$
$\alpha$	The step value of firefly algorithm
$\beta$	The attractiveness of a firefly
$\gamma$	The light absorption coefficient
$p$	The probability of the worst firefly to learn from the brightest firefly
$X_{best}$	The position of the brightest firefly
$X_{worst}$	The position of the worst firefly
$X_{max}$	The maximum range of the object function
$X_{min}$	The minimum range of the object function

complexity and solving them make computational cost of algorithm increased [22]. The FA mimics the flashing behavior of fireflies. The FA uses the following three idealized rules. First, all fireflies are unisex which means that one firefly will be attracted to other fireflies regardless of their sex. Secondly, the degree of the attractiveness of a firefly is proportion to its brightness, thus for any two flashing fireflies, the less bright one will move towards the brighter one and the more brightness. If there is no brighter one than a particular firefly, it will move randomly. Finally, the brightness of a firefly is somehow related with the analytical form of the cost function. For a maximization problem, the brightness of each firefly is proportional to the value of the cost function. In case of a minimization problem, the brightness of each firefly is inversely proportional to the value of objective function [23].

The basic steps of the FA are summarized as the pseudo code shown in Fig. 1 which consists of the three rules discussed above.

Here,  $d$  is the dimension of the object function to be optimized,  $n$  is the number of fireflies,  $\gamma$  is the light absorption coefficient,  $I_i$  is the light intensity and the distance  $r$  between any two fireflies  $i$  and  $j$  locate at  $X_i$  and  $X_j$  can be evaluated in the Cartesian framework as follows.

$$r_{ij} = Distance(X_i, X_j) = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \tag{1}$$

As light intensity decreases as the distance from the source increases, the variations of light intensity should be monotonically decreasing functions. In most applications, it can be approximated using the following form.

$$I(r) = I_0 e^{-\gamma r^2} \tag{2}$$

### Firefly Algorithm

---

```

begin
  Objective function  $f(X)$ ,  $X=(x_1, \dots, x_d)^T$ 
  Generate initial population of fireflies  $X_i(i=1, \dots, n)$ 
  Brightness  $I_i$  at  $X_i$  is determined by  $f(X_i)$ 
  Define light absorption coefficient  $\gamma$ 
  while ( $t < \text{MaxGeneration}$ )
    for  $i=1:n$  all  $n$  fireflies
      for  $j=1:n$  all  $n$  fireflies
        if ( $I_j > I_i$ )
          Move firefly  $i$  towards  $j$  in  $d$ -dimension
        end if
        Attractiveness varies with distance via  $\exp[-\gamma r^2]$ 
        Evaluate new solutions and update brightness
      end for  $j$ 
    end for  $i$ 
    Rank the fireflies and find the current best
  end while
  Post process results and visualization
end

```

---

**Fig. 1** Pseudocode of the basic firefly algorithm

Where  $I_0$  is the initial light intensity. As a firefly's attractiveness is proportional to the light intensity, we can define the attractiveness  $\beta$  as follows.

$$\beta(r) = \beta_0 e^{-\gamma r^2} \quad (3)$$

Here,  $\beta_0$  is a constant and describes the attractiveness at  $r = 0$ . Now the movement of a firefly  $i$  is attracted to another more attractive firefly  $j$  is determined by Eq. (4).

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha (\text{rnd}() - 0.5) \quad (4)$$

The third term is randomization with step  $\alpha$  being the randomization parameter. The value of  $\text{rnd}()$  is a random number generator uniformly distributed range from 0 to 1.

## 2.2 State-of-the-art FA variants

Substantial amount of researches are performed to improve the FA's performance. Among these works, parameters adaptation strategy has become one of the promising approaches. In [19], they used different chaotic maps to replace the parameters of the basic FA, and revealed the improvement of the chaotic FA due to the application of deterministic chaotic signals in place of constant values. Yu et al. [18] proposed a self-adaptive step firefly algorithm. Its core idea was to set the step of each firefly varying with the iteration according to each firefly's historical information and current situation. Bidar and Kanan [22] used fuzzy logic as a tool for parameters tuning which considering algorithm progress trend toward the best solution.

Another area of research is to explore the FA’s learning strategies. Yang [21] presented a new firefly algorithm by combining Lévy flights with the search strategy. In [17] they proposed a fuzzy-based version of the basic FA to increase the exploration and improve the global search of the basic FA. Khajezadeh et al. [24] presented a new hybrid algorithm to combine the global exploration ability of the FA to converge rapidly to a near optimum solution. Roy et al. [6] used self-adaptation of the algorithm control parameter values by learning from fireflies’ previous experiences in generating quality solutions and studied on an automatic motion planning problem of nonholonomic car-like system.

### 2.3 Opposition-based learning

Opposition-based learning (OBL) was first proposed by Tizhoosh [25] and was successfully applied to several problems [26–29]. OBL has been hybridized with differential evolution, reinforcement learning and back propagation learning. The main idea behind OBL is the simultaneously consideration of a candidate solution and its corresponding opposite solution. Let  $x \in [a, b]$  is a real number, the opposition number of  $x$  is denoted as  $x'$  and defined as:

$$x' = a + b - x \tag{5}$$

The above definition can be extended to higher dimensions. Let  $P(x_1, x_2, \dots, x_n)$  be an  $n$ -dimensional vector, where  $x_i \in [a_i, b_i]$  and  $i = 1, 2, \dots, n$ . The opposite vector of  $P$  is defined by  $P' = (x'_1, x'_2, \dots, x'_n)$ , where  $x'_i = a_i + b_i - x_i$ .

## 3 The proposed algorithm

### 3.1 Enhancing FA using opposition-based learning

Population premature convergence around a local optimum is a common problem for population-based algorithms. During the FA searching process, a firefly will move towards the brighter one and the more brightness. At the beginning of the algorithm, diversity is highest when all fireflies have been randomly generated. After some iteration, the population diversity decreases due to sample generating bias or selection pressure and therefore increasing the difficulty of escaping a local optima [30,31]. When premature convergence occurs, the FA’s search ability of exploration is reduced and it will have a low possibility to explore new search areas.

To overcome this shortage, inspired by the concept of opposition-based learning, we propose a modified FA to enhance the performance of the basic FA. The strategy used here is to replace the worst firefly with a new constructed firefly which can increase the diversity of fireflies and help a firefly to escape from local optima. By adding this new constructed firefly, the regular searching route is broken and the population diversity is enhanced. The approach is formulated as follows.

$$X_{worst}(t + 1) = \begin{cases} X_{best}(t) & \text{if } \text{rnd}() < p \\ X_{\max} + X_{\min} - X_{worst}(t) & \text{otherwise} \end{cases} \tag{6}$$

where *worst* is the index of the worst firefly, *best* is the index of the brightest firefly,  $X_{best}(t)$  and  $X_{worst}(t)$  are the position of the brightest firefly and the worst one at  $t$ th iteration respectively,  $rnd()$  is a uniform random number distributed range from 0 to 1. Here,  $p \in [0, 1]$  is the probability of the worst firefly to learn from the brightest firefly. Under such operation, the worst firefly is forced to escape from the normal path. In addition to the above approach which takes some element from the brightest firefly.

In Eq. (6),  $X_{worst}(t + 1)$  is the worst firefly's position at  $t + 1$  iteration. It is determined by the opposition number of itself or the position of the brightest firefly as mentioned. Therefore, we introduce  $p$  to determine which way should compute. The bigger the value of  $p$  selected, the greater probability determined by the brightest firefly. In order to study the impact of the selected value of  $p$ , we adopt three different values for it in our experiment.

### 3.2 Procedure of proposed firefly algorithm

The implementation procedure of the proposed firefly algorithm can be described as follows:

- Step 1: Create the initial population of fireflies,  $\{X_1, X_2, \dots, X_n\}$ ;
- Step 2: Calculate the light intensity for each firefly member,  $\{I_1, I_2, \dots, I_n\}$ ;
- Step 3: Find and replace the worst firefly by Eq. (6)
- Step 4: Move a firefly  $i$  towards other brighter fireflies, the position is updated by Eq. (4);
- Step 5: Update the solution set;
- Step 6: Terminate if a meeting criterion is fulfilled otherwise go to Step 2.

## 4 Experimental and results

In order to make a fair comparison of the basic FA and the proposed FA, we used a test suite of 16 standard benchmark functions and the same settings.

### 4.1 Benchmark functions

The test suite is composed of a diverse set of problems of different dimensions including unimodal and multimodal functions. The benchmark functions have been listed in Table 2.

### 4.2 Settings for the experiments

The modified FA and the basic FA were tested on above sixteen standard benchmark functions. All the programs were run in Matlab 2010b with 2 GB of RAM under Windows XP. In each case study, it adopted 100 independent runs for each of the algorithm and this may eliminate stochastic unconformity. As it suggested by Yang

**Table 2** Benchmark functions

Functions	Formulations	Limits
1	$\min f(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$	$[-1, 1]$
2	$\min f(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	$[-3, 3]$
3	$\min f(x) = ( x_1  - 5)^2 + ( x_2  + 5)^2$	$[-10, 10]$
4	$\min f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	$[-5, 5]$
5	$\min f(x) = 100(x_2 - x_1)^2 + (6.4(x_2 - 0.5)^2 - x_1 - 0.6)^2$	$[-5, 5]$
6	$\min f(x) = 100(x_2 - x_1^2) + (x_1 - 1)^2$	$[-2.048, 2.048]$
7	$\min f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	$[-10, 10]$
8	$\min f(x) = (1 + (x_1 + x_2 + 1))^2 \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)^2 \left(30 + (2x_1 - 3x_2)^2 \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right)$	$[-2, 2]$
9	$\min f(x) = 0.5 + \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{1 + 0.001(x_1^2 + x_2^2)^2}$	$[-10, 10]$
10	$\max f(x) = \cos(x_1)\cos(x_2)e^{-((x_1 - \pi)^2 - (x_2 - \pi)^2)}$	$[-20, 20]$
11	$\max f(x) = 280 - \frac{x_1^4 - 16x_1^2 + 5x_1}{2} + \frac{x_2^4 - 16x_2^2 + x_2^2}{2}$	$[-5, 5]$
12	$\max f(x) = \ln \left\{ \left[ (\sin(\cos x_1 + \cos x_2))^2 - (\cos(\sin x_1 + \sin x_2))^2 \right]^2 + x_1 \right\} - 0.1((x_1 - 1)^2 + (x_2 - 1)^2)$	$[-10, 10]$
13	$\max f(x) = 660 - (x_1^2 - 11)^2 - (x_1 + x_2^2 - 7)^2$	$[-6, 6]$
14	$\min f(x) = \sum_{i=1}^n (x_i - 1)^2$	$[-2, 2]$
15	$\min f(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-10, 10]$
16	$\min f(x) = \sum_{i=1}^n x_i e^{-\sum_{i=1}^n x_i^2}$	$[-10, 10]$

**Table 3** Key parameter values setting

Parameter	Value
$\alpha$	0.2
$\beta_0$	1.0
$\gamma$	1.0
$p$ of FA1	0.25
$p$ of FA2	0.50
$p$ of FA3	0.75
$n$ (number of fireflies)	30
$d$ (dimension of function 14–16)	20
$t$ (the maximum of iteration)	1000

**Table 4** Results of objective functions in 100 runs

Function	Best value			Worst value			Standard deviation					
	Basic FA	FA1	FA2	FA3	Basic FA	FA1	FA2	FA3	Basic FA	FA1	FA2	FA3
	1	-2.0000E+00	-2.0000E+00	-2.0000E+00	-2.0000E+00	-1.9993E+00	-1.9992E+00	-1.9993E+00	-1.9993E+00	1.6621E-04	1.2229E-04	1.6883E-04
2	-1.0316E+00	-1.0316E+00	-1.0316E+00	-1.0316E+00	-2.1546E-01	-1.0316E+00	-1.0316E+00	-6.6359E-01	8.1616E-02	5.5184E-06	6.4805E-06	3.6803E-02
3	2.5000E+01	2.5000E+01	2.5000E+01	2.5000E+01	2.5002E+01	2.5001E+01	2.5002E+01	2.5002E+01	4.5472E-04	3.6141E-04	3.1257E-04	3.8348E-04
4	3.9789E-01	3.9789E-01	3.9789E-01	3.9789E-01	3.9790E-01	3.9790E-01	3.9790E-01	3.9789E-01	2.4592E-06	1.7245E-06	1.9808E-06	1.6748E-06
5	7.5497E-07	1.6384E-07	8.0709E-07	4.8956E-08	4.0820E-04	2.1044E-04	2.1755E-04	2.5267E-04	6.1139E-05	4.5176E-05	4.6951E-05	5.2361E-05
6	1.3969E-07	4.8956E-08	1.1322E-07	5.0341E-08	6.6770E-05	4.5574E-05	7.1502E-05	6.0515E-05	1.1191E-05	1.0395E-05	1.2574E-05	1.3118E-05
7	2.3499E-08	1.1920E-08	2.6533E-08	1.4189E-10	1.6307E-05	2.0913E-05	1.4386E-05	1.2322E-05	3.1127E-06	3.8161E-06	3.1484E-06	2.7088E-06
8	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00	3.0002E+01	3.0032E+00	3.0030E+00	3.0020E+00	2.7001E+00	4.6058E-04	3.8481E-04	3.5488E-04
9	9.0998E-07	2.4911E-09	4.9851E-07	8.4104E-08	3.7224E-02	9.7159E-03	3.7224E-02	3.7224E-02	5.4590E-03	3.2814E-03	4.5826E-03	7.7330E-03
10	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00	2.4463E-05	8.7430E-12	7.6439E-23	7.6438E-23	4.3282E-01	4.9427E-01	4.6879E-01	4.2921E-01
11	3.5833E+02	3.5833E+02	3.5833E+02	3.5833E+02	3.4424E+02	3.5833E+02	3.5833E+02	3.5833E+02	1.4088E+00	9.7513E-06	7.0113E-06	7.1627E-06
12	2.2051E+00	2.2051E+00	2.2051E+00	2.2051E+00	1.4938E+00	1.6587E+00	1.4839E+00	1.4839E+00	2.1664E-01	2.2296E-01	2.2595E-01	2.3924E-01
13	6.6000E+02	6.6000E+02	6.6000E+02	6.6000E+02	6.6000E+02	6.6000E+02	6.6000E+02	6.6000E+02	2.5492E-05	1.1414E-05	1.4449E-05	1.5109E-05
14	1.5318E-08	6.9358E-09	2.0105E-08	1.9419E-08	9.5035E-08	2.9650E-08	1.0655E-07	1.0342E-07	1.5312E-08	4.2024E-09	1.4846E-08	1.5038E-08
15	8.9546E+00	7.9597E+00	7.9597E+00	5.9698E+00	4.1788E+01	3.4824E+01	3.3829E+01	3.0844E+01	7.4180E+00	6.0715E+00	5.7849E+00	5.3692E+00
16	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00



[2], the number of fireflies was 30 and for function 14–16 the dimension was 20. All key parameter values are listed in Table 3.

Three different values of  $p$  (0.25, 0.5 and 0.75) are tested for the proposed firefly algorithm which named FA1, FA2 and FA3 respectively.

### 4.3 Experimental results

Table 4 shows the comparison results of best value, worst value and standard deviation on sixteen benchmark functions which the best results are highlighted in bold. The goal of experimental work is to show how our approach can improve the results of basic FA and how good these results are when compared with the basic FA. The experiment results show that the proposed FA (FA1:  $p = 0.25$ , FA2:  $p = 0.5$ , FA3:  $p = 0.75$ ) can obtain the best solution on all functions compared with the basic FA. It can be observed from Table 4 that the worst value obtained by the FA1 is very good. So, it performed well in terms of stability. This can be seen from standard deviation values.

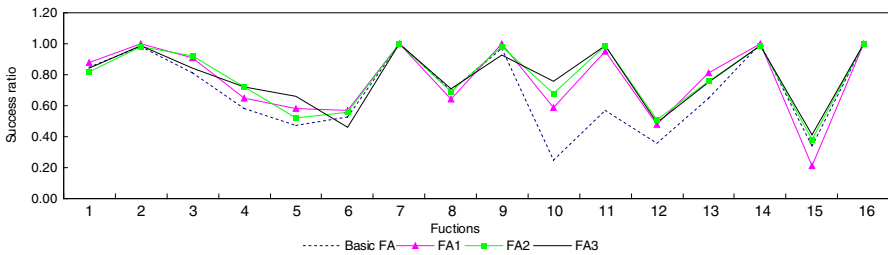


Fig. 2 Success ratios from all the test functions

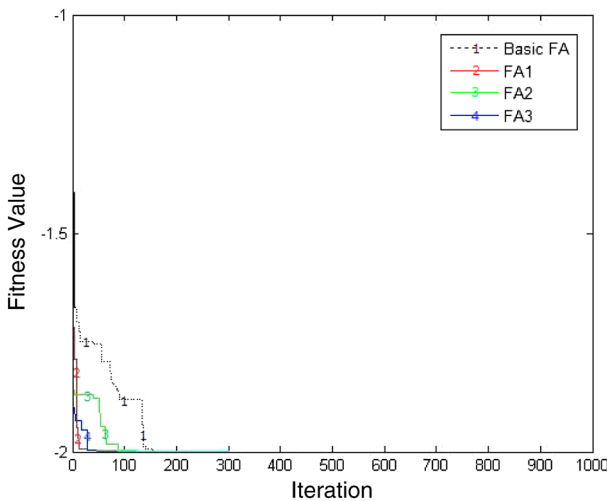


Fig. 3 Comparison of curve graph for  $f_1$

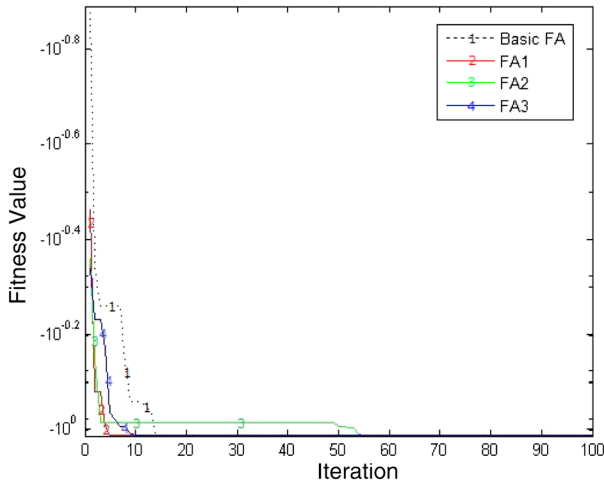


Fig. 4 Comparison of curve graph for  $f_2$

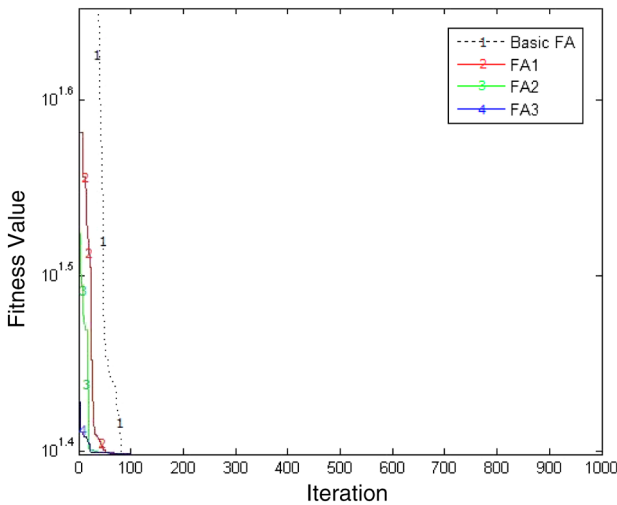


Fig. 5 Comparison of curve graph for  $f_3$

On the whole, the proposed FA, especially FA1 is much better than the basic FA. Some significant improvements are achieved from this method.

When algorithms are compared for a given set of test suite, we can use the success ratio to show whether a solution has better quality than the solution produced by another method for the same problem instance. Therefore, the success ratio is an important measure in optimization problems and it determines the success probability of an algorithm. Here, the success ratio is defined as  $SR = N_{successful} / N_{all}$ , where  $N_{successful}$  is the number of trials which found the solution is successful and  $N_{all}$  is the number of all trials. In our experiments,  $N_{all} = 100$ . Figure 2 gives the success

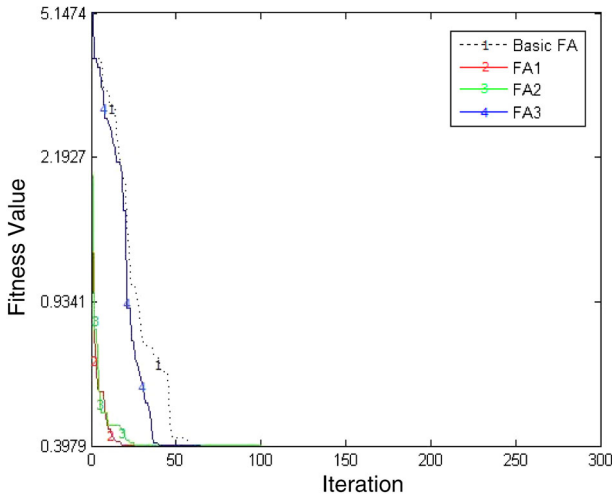


Fig. 6 Comparison of curve graph for  $f_4$

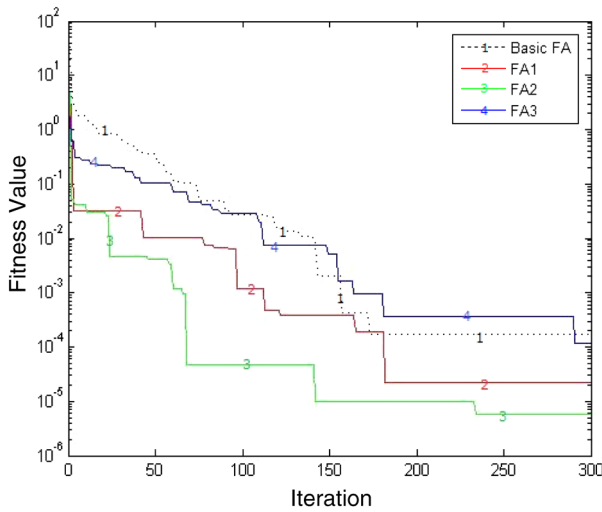
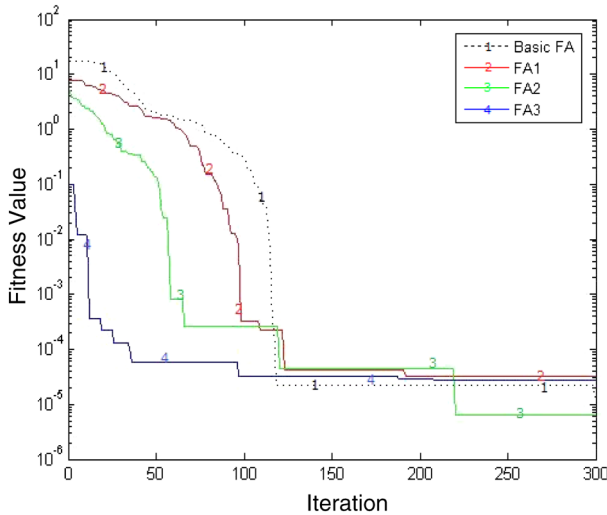


Fig. 7 Comparison of curve graph for  $f_6$

ratios of the basic FA and the proposed FAs for all the test functions. The results show the significant improvements obtained by the proposed FA.

We compare the fitness value obtained during the iterations between the basic FA and our proposed FA. Limited by space, Figs. 3, 4, 5, 6, 7, 8 measure the difference of the basic FA and the proposed FA with diversity control for  $f_1 - f_4$ ,  $f_6$  and  $f_7$ . In order to see more clearly, we adopt the different iterations in the compared graph.



**Fig. 8** Comparison of curve graph for  $f_7$

## 5 Conclusion

Based on the generalized opposition-based learning, a proposed FA search process is presented. The strategy used in this paper is to alter the position of the worst firefly. This new constructed firefly is created by taken some elements from the opposition number of the worst firefly or the position of the brightest firefly. After this operation, the worst firefly is forced to escape from the normal path. It increases the diversity of fireflies and can help algorithm to escape from local optimal. Experiments on 16 standard benchmark functions show that our method has some significant improvements. The proposed algorithm can achieve better performance both in solution quality and convergent property. How to apply the method in practice such as job-shop scheduling problem (JSP), vehicle routing problem (VRP), is our future work.

**Acknowledgments** This research is financially supported by the National Natural Science Foundation of China (No. 71131002) and the Universities Natural Science Foundation of Anhui Province (No. KJ2011A268, No. KJ2012Z429). The authors of the paper express great acknowledgement for these supports.

## References

1. Haklı H, Uğuz H (2014) A novel particle swarm optimization algorithm with Levy flight. *Appl Soft Comput* 23:333–345
2. Yang XS (2009) Firefly algorithms for multimodal optimization, in *stochastic algorithms: foundations and applications*. Springer, New York, pp 169–178
3. Yang XS (2010) *Engineering optimization: an introduction with metaheuristic applications*. Wiley, New York

4. Ram G, Mandal D, Kar R, Ghoshal SP (2014) Optimized hyper beamforming of receiving linear antenna arrays using firefly algorithm. *Int J Microw Wirel Technol* 6:181–194
5. Marichelvam MK, Prabaharan T, Yang XS (2014) A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems. *IEEE Trans Evolut Comput* 18:301–305
6. Roy G, Rakshit P, Konar A, Bhattacharya S, Kim E (2013) Adaptive firefly algorithm for nonholonomic motion planning of car-like system. *IEEE Congress Evolut Comput* 2162–2169
7. Sanaei P, Akbari R, Zeighami V, Shams S (2013) Using firefly algorithm to solve resource constrained project scheduling problem. In: *Proceedings of Seventh International Conference on Bio-Inspired Computing Theories and Applications (Bic-Ta 2012)*, vol 1. pp 417–428
8. Herbadji O, Nadhir K, Slimani L, Bouktir T (2013) Optimal power flow with emission controlled using firefly algorithm. In: *2013 5th International Conference on Modeling, Simulation and Applied Optimization (Icmsao)*
9. Sulaiman MH, Daniyal H, Mustafa MW (2012) Modified firefly algorithm in solving economic dispatch problems with practical constraints. In: *IEEE International Conference on Power and Energy (Pecon)*, pp 157–161
10. Poursalehi N, Zolfaghari A, Minuchehr A, Moghaddam HK (2013) Continuous firefly algorithm applied to PWR core pattern enhancement. *Nucl Eng Design* 258:107–115
11. Kannan G, Subramanian DP, Shankar RU (2015) Reactive power optimization using firefly algorithm. In: *Power Electronics and Renewable Energy Systems*, Springer, pp 83–90
12. Ma Y, Zhao Y, Wu L, He Y, Yang XS (2015) Navigability analysis of magnetic map with projecting pursuit-based selection method by using firefly algorithm. *Neurocomputing*
13. Raja SB, Pramod CS, Krishna KV, Raganathan A, Vinesh S (2015) Optimization of electrical discharge machining parameters on hardened die steel using firefly algorithm. *Eng Comput* 31:1–9
14. Abdelaziz Y, Hegazy YG, El-Khattam W, Othman MM (2015) Optimal planning of distributed generators in distribution networks using modified firefly method. *Electric Power Compon Syst* 43:320–333
15. Yazdani D, Nasiri B, Sepas-Moghaddam A, Meybodi MR (2013) A novel multi-swarm algorithm for optimization in dynamic environments based on particle swarm optimization. *Appl Soft Comput* 13:2144–2158
16. Olamaei J, Moradi M, Kaboodi T (2013) A new adaptive modified firefly algorithm to solve optimal capacitor placement problem. In: *Electrical Power Distribution Networks (EPDC), 2013 18th Conference*, pp 1–6
17. Hassanzadeh T, Kanan HR (2014) Fuzzy FA: a modified firefly algorithm. *Appl Artif Intell* 28:47–65
18. Yu SH, Yang SL, Su SB (2013) Self-adaptive step firefly algorithm. *J Appl Math*
19. Gandomi H, Yang XS, Talatahari S, Alavi AH (2013) Firefly algorithm with chaos. *Commun Nonlinear Sci Numer Simul* 18:89–98
20. Bidar M, Kanan HR (2013) Jumper firefly algorithm. In: *Proceedings of the 3rd International Conference on Computer and Knowledge Engineering (Ickce)*, pp 267–271
21. Yang XS (2010) Firefly algorithm, levy flights and global optimization. *Res Develop Int Syst* 209–218
22. Bidar M, Kanan HR (2013) Modified firefly algorithm using fuzzy tuned parameters. In: *2013 13th Iranian Conference on Fuzzy Systems (Ifsc)*
23. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimisation. *Int J BioInspired Comput* 2:78–84
24. Khajezadeh M, Taha MR, Eslami M (2013) A new hybrid firefly algorithm for foundation optimization. *Nat Acad Sci Lett India* 36:279–288
25. Tizhoosh HR (2006) Opposition-based learning: a new scheme for machine intelligence. In: *International Conference on Computational Intelligence for Modelling, Control and Automation Jointly with International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, vol 1. pp 695–701
26. Chen CH, Lin CM (2012) Enhance performance of particle swarm optimization by altering the worst personal best particle. In: *2012 Conference on Technologies and Applications of Artificial Intelligence (Taai)*, pp 56–61
27. Rahnamayan S, Tizhoosh HR, Salama MMA (2006) Opposition-based differential evolution algorithms. *IEEE Congress Evolut Comput* 1–6:1995–2002
28. Muthukumar R, Thanushkodi K (2014) Opposition based differential evolution algorithm for capacitor placement on radial distribution system. *J Elect Eng Technol* 9:45–51
29. Wang H, Wu ZJ, Rahnamayan S, Liu Y, Ventresca M (2011) Enhancing particle swarm optimization using generalized opposition-based learning. *Inform Sci* 181:4699–4714

30. Ventresca M, Tizhoosh HR (2008) A diversity maintaining population-based incremental learning algorithm. *Inform Sci* 178:4038–4056
31. Cheng S (2013) Population diversity in particle swarm optimization: definition, observation, control, and application. University of Liverpool, England