Firefly Algorithm with Elite-*k* Neighborhood Attraction Model

Jing Wang
School of Software and Internet of things Engineering
Jiangxi University of Finance and Economics
Nanchang, China
wangjing@jxufe.edu.cn

Abstract. Firefly algorithm is a efficient meta-heuristic optimization algorithm which has been a hot topic in intelligent algorithms research. However, in the standard firefly algorithm, the fireflies will be attracted by all the other brighter fireflies, and there is a lot of attraction that does not affect, but will increase the computational time of the algorithm. In addition, all the best firefly information in the search process has not been recorded, which may lead the algorithm to be inefficient. To over these problems, this paper proposed en elite-k neighborhood attraction firefly algorithm (EkFA), which can not only reduce the no effective attractions between the fireflies but also can make full use of the best firefly's information to guide other nearby fireflies to movement. Thirteen well-known benchmark functions are used to verify the performance of our proposed method. The experimental results show that the accuracy and efficiency of the proposed algorithm are significantly better than those of other FA variants.

Keywords: firefly algorithm; meta-heuristic algorithm; **elite-k Neighborhood Attraction**; global optimization

1 Introduction

The firefly algorithm is a swarm-based optimization algorithm which is proposed by Yang in 2008[1]. The firefly algorithm searches for the brighter companion around the region by simulating the firefly luminescence behavior in nature, moving towards the better position in the region, and finally gathering to the brightest and optimal position, so as to realize the function of searching the optimal solution. Compared with other typical intelligent algorithms such as particle swarm optimization algorithm and genetic algorithm, the existing simulation results show that the firefly algorithm has high convergence speed and revenge accuracy. And the firefly algorithm's parameters are simple and effective, so it has been a hot topic in intelligent algorithms research in a short time, and has been widely used in many fields, such as scheduling problem [2, 3], wireless sensor networks[4, 5], stock forecasting[6], mechanical design optimization problem [7] and so on.

Great progress has been made in this field, however, firefly algorithm still has some flaws. In the standard firefly algorithm, each firefly will be attracted to all the remaining fireflies, and this behavior will increase the number of invalid objective function evaluations. When faced with high-dimensional problems or objective functions are complex, the complexity of the algorithm will be dramatically increased. To overcome these problems, Wang [8] proposes a neighborhood attraction FA (NaFA), In NaFA, each firefly is attracted by other brighter fireflies selected from a predefined neighborhood, but the predefined neighborhood is a virtual structure, and not the real structure of the fireflies. The essence of the algorithm is to reduce the number of times that fireflies are attracted by other remaining fireflies. On the basis of existing literature data, we carried out studies in an effort to a new FA variant, namely elite-k neighborhood attraction FA(EkFA). EkFA can not only reduce the no effective attractions between the fireflies, but also can make full use of the best firefly's information to guide other nearby fireflies to movement.

This paper is divided into 5 sections as follows. Section 1 is the introduction. Section 2 gives the background of the problem which includes stand FA and its variant. The main idea of our proposed EkFA is described in Section 3. Experimental results and analysis are presented in Section 4. Finally, the work is concluded in Section 5.

2 A Brief Review of Firefly Algorithm

In firefly algorithm, the location of each firefly represents a solution to the problem to be solved. The brightness of the firefly depends on the objective function value of the problem to be solved. The better the objective function is, the stronger the brightness of the firefly. As the iterative process progresses, the weak fireflies in the population are moving closer to their own fireflies, and most of the fireflies will gather near the brightest fireflies, and the brightest fireflies are the optimal solution to the problem.

There are four very important concepts in the firefly algorithm: light intensity, attractiveness, distance and movement.

Light intensity: The light intensity $I(r)$is defined by Yang[9]:

$$I(r) = I_0 e^{-\gamma r^2} \tag{1}$$

where I_0 is the initial brightness. The parameter $\gamma$is the light absorption coefficient, r is the distance between two fireflies.

Attractiveness: The attractiveness of a firefly is monotonically decreasing as the distance increases, and the attractiveness is as follows[9]:

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

Where β_0 is the attractiveness at r = 0.The light absorption coefficientγ will determine the variation of attractiveness$\beta$and$\gamma \in [0, \infty]$.For most practical implementations, Yang suggest that $\gamma = 1$ and $\beta_0 = 1$.

Distance: For two firefliesx_i and fireflies x_j, r is defined by Yang[9]:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{d=1}^{D} (x_{id} - x_{jd})^2} \tag{3}$$

Movement: The light intensity of the weak firefly will move to another brighter firefly, assuming that a firefly x_j is more brighter than firefly x_i, the position update equation given by the following formula[9]:

$$x_i(t+1) = x_i(t) + \beta_0 e^{-\gamma r^2}\left(x_j(t) - x_i(t)\right) + \alpha\epsilon_i \tag{4}$$

where t is the iterations.The third term of the right is a random disturbance term which containsα and ϵ_i, $\alpha \in [0, 1]$ is the step factor, $\epsilon_i \in [0.5, 0.5]$ is a random number vector obtained by Gaussian distribution or Levy flight[10].

The framework of stand FA is listed as follow, FEs is the number of evaluations, MAXits is the maximum number of evaluations, and PS is the population size.

Algorithm1: Framework of FA

```
Randomly initialize N fireflies (solutions) as an initial population {X_i | i = 1, 2, ..., N};
Calculate the fitness v of each firefly X_i;
FEs = 0 and PS = N;
while FEs ≤ MAXFEs
    for i= 1 to N
        for j = 1 to N
            if f(X_j) < f(X_i)
                Move X_i towards X_j according to Eq. 4;
                Calculate the fitness value of the new X_i;
                FEs++;
            end if
        end for
    end for
end while
```

Fister et al.[11] proposed a memetic self-adaptive FA (MFA), where values of control parameters are changed during the run. Experimental results show that MFA were very promising and showed a potential that this algorithm could successfully be applied in near future to the other combinatorial optimization.

MFA makes the dynamic change of the step factor with the evolutionary iteration. It is redefined as follows:

$$\alpha(t+1) = \left(\frac{1}{9000}\right)^{\frac{1}{t}}\alpha(t), \tag{5}$$

where t represents the current iteration. In the above-mentioned MFA[11], Fister also changes the fireflies's movement strategy, which can be defined by the following equation:

$$x_{id}(t+1) = x_{id}(t) + \beta\left(x_{jd}(t) - x_{id}(t)\right) + \alpha(t) s_d \epsilon_i, \tag{6}$$

where

$$\beta = \beta_{min} + (\beta_0 - \beta_{min})\, e^{-\gamma r^2}, \tag{7}$$

$$s_d = x_d^{max} - x_d^{min}, \tag{8}$$

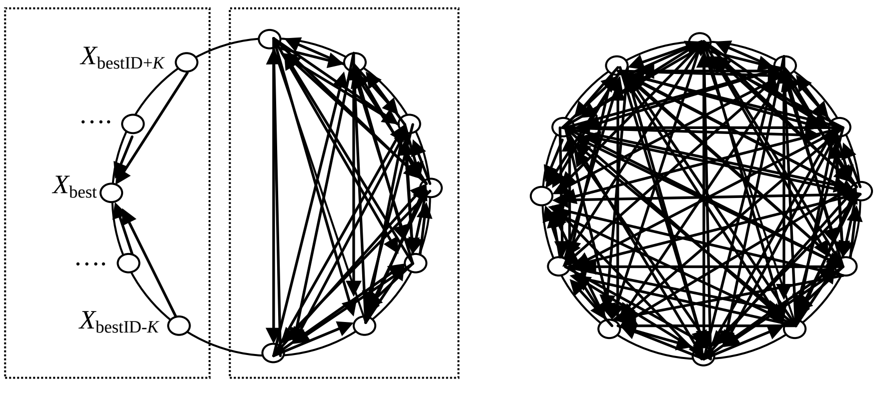
x_{id} denotes the d-dimensional variable of the i-th firefly, β_{min} is usually set to 0.2 that is the minimum value in β, s_d is the length of the domain of the initialization variable. x_d^{max} and x_d^{min} are the maximum and minimum boundaries of the variable, respectively. In this paper, our proposed strategies are based on MFA.

3 Our Proposed Firefly Algorithm

3.1 Elite Neighborhood Attraction Model

Firefly algorithm is a simple and efficient meta-heuristic algorithm. However, it also has two unfavorable factors. First, as shown in Algorithm 1, there are two inner cycles and one outer loop in the firefly algorithm. The number of outer cycles is the maximum number of iterations *MAXits*, and the number of internal cycles is the population number *N*. Assume that the time spent on the evaluation of the objective function is T, the complexity of the firefly algorithm is $O(N^2 * MAXits * \mathrm{T})$. In the algorithm, the main cost of the firefly algorithm is to spend on the evaluation of the objective function, no matter how far the distance between the fireflies, the fireflies will be attracted by other brighter fireflies. The attraction will cause the fireflies to move, and the movement will cause the algorithm to evaluate the objective function again. Although the attraction will increase the diversity of the algorithm, however it also will increase the complexity of algorithm. Second, the algorithm does not use the history optimal value of the individual, nor does it use the global optimal value. It also may affect the accuracy and convergence speed of the firefly algorithm.

Therefore, based on this phenomenon, this paper proposes a new FA variant with elite-k neighborhood attraction method, we called it EkFA. We select the best fireflies and then move the K other fireflies closest to the best fireflies, rather than moving to other brighter fireflies in each generation. For a more clear description of elite-k neighborhood attraction method, Fig. 1 shows the structure of elite-k neighborhood attraction and full attraction. Assume that all N fireflies are organized into circular topological structures according to their indices. Fig. 1(a) and Fig. 1(b) show the state of elite-k neighborhood attraction and full attraction when N is 12, respectively. It is easy to conclude that the time complexity of the algorithm is $O(N^2 * MAXits * T)$ in full attraction, and the time complexity of the algorithm in elite-k neighborhood attraction is $O((N^2 - K^2) * MAXits * T)$, and in each generation there are K^2 individuals that are moving according to the global optimal value. In this paper, *K* is set to 10.



(a) Eite-*k* attraction (b) Full attraction

Fig. 1. Eite-*k* attraction vs. Full attraction, where *N*=12

3.2 Framework of Our Proposed Firefly Algorithm

The main idea of our proposed method can be briefly described as follow: First, find the brightest firefly at present, then determine whether the distance between firefly and the brightest firefly is less than k. If less than, let the firefly move directly to the brightest firefly; otherwise, execute the normal firefly attraction operation.

Experiments results from the literature[11] show that MFA is an efficient FA variant. So our proposed EkFA algorithm not only uses the elite-k neighborhood attraction model, but also incorporates the advanced technology of MFA, it is improved based on MFA. The EkFA algorithm framework is shown in Algorithm 2, FEs is the number of evaluations, MAXits is the maximum number of evaluations, and PS is the population size.

Algorithm2: Framework of EkFA

```
Randomly initialize N fireflies (solutions) as an initial population {X_i | i = 1, 2, ..., N};
Calculate the fitness v of each firefly X_i;
FEs = 0 and PS = N;
while FEs ≤ MAXits
    Find the best individual's ID;
    Update the step factor α according to Eq. 5;
    Update the attractiveness β according to Eq. 7
    for i= 1 to N
        if (abs(i - bestID) < K)
            Move X_i towards X_best according to Eq. 6;
        else
            for j = 1 to N
                if f(X_j) < f(X_i)
                    Move X_i towards X_j according to Eq. 6;
                    Calculate the fitness value of the new X_i;
                    FEs++;
                end if
            end for
        end else if
    end for
end while
```

4 Experimental Study

4.1 Test problems

In this experiment, 13 well-known benchmark functions[12] were used to test the performance of the algorithm. In these functions, f_1-f_5 are unimodal functions, f_6 is a step function with a minimum value, and discontinuous, f_7 is a quadratic function with noise, and f_8-f_{13} are multimodal functions with many local minimums. All functions are minimization problems, and their descriptions are shown in Table 1.

Table1: Benchmark functions used in the experiments, where *D* is the problem dimension

Name	Function	Search Range	Global Optimum
Sphere	$f_1(x) = \sum_{i=1}^{D} x^2$	[-100, 100]	0
Schwefel 2.22	$f_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} x_i$	[-10, 10]	0
Schwefel 1.2	$f_3(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	[-100, 100]	0
Schwefel 2.21	$f_4(x) = \max\{\lvert x_i \rvert, 1 \le i \le D\}$	[-100, 100]	0
Rosenbrock	$f_5(x) = \sum_{i=1}^{D} [100(x_{i+1} - x_i^2)^2 + (1 - x_i^2)^2]$	[-30, 30]	0
Step	$f_6(x) = \sum_{i=1}^{D} \lfloor x_i + 0.5 \rfloor$	[-100, 100]	0
Quartic with noise	$f_7(x) = \sum_{i=1}^{D} i x_i^4 + random[0, 1)$	[-1.28, 1.28]	0
Schwefel 2.26	$f_8(x) = \sum_{i=1}^{D} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right) + 418.9829D$	[-500, 500]	0
Rastrigin	$f_9(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos 2\pi x_i + 10]$	[-5.12, 5.12]	0
Ackley	$f_{10}(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	[-32, 32]	0
Griewank	$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600, 600]	0
Penalized 1	$f_{12}(x) = \frac{\pi}{D}\left\{\sum_{i=1}^{D-1}(y_i - 1)^2[1 + \sin(\pi y_{i+1})] + (y_D - 1)^2 + 10\sin^2(\pi y_1)\right\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4), y_i = 1 + (x_i + 1)/4$ $u(x_i, a, k, m) = \begin{cases} u(x_i, a, k, m), & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	[-50, 50]	0
Penalized 2	$f_{13}(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2[1 + \sin^2(2\pi x_D)]\right\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	[-50, 50]	0

4.2 Experimental results

In order to ensure the fairness of the algorithm, the following parameters of all algorithms are set to the same, and list as follow:

- Population size: 30
- Max iterations: 5.0E+05
- Run times: 30
- Problem dimension:30

The computational results of EkFA are recorded in Table 1. As shown in the table1, except f_3, f_5 and f_8 fall into the local optimal, most of the other functions find the approximate optimal solution. Especially the f_6 and f_{11}, they finds the global optimal solution every time.

Table 2: Computational results of EkFA

Function	Worst	Best	Mean	Std Dev
f_1	1.41E-127	7.02E-128	1.09E-127	2.86E-128
f_2	1.85E-64	1.35E-64	1.60E-64	2.14E-65
f_3	5.39E+01	2.75E-02	1.83E+01	2.07E+01
f_4	2.27E-64	1.45E-64	1.76E-64	3.29E-65
f_5	9.91E+01	2.31E+01	4.06E+01	3.28E+01
f_6	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_7	1.48E-01	4.21E-02	9.88E-02	4.58E-02
f_8	-5.66E+03	-7.02E+03	-6.16E+03	5.66E+02
f_9	5.37E+01	2.69E+01	3.90E+01	1.16E+01
f_{10}	2.18E-14	1.47E-14	1.75E-14	3.89E-15
f_{11}	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{12}	1.57E-32	1.57E-32	1.57E-32	0.00E+00
f_{13}	5.29E-32	2.34E-32	3.72E-32	1.50E-32

In 2010, Yang[9] suggested $\gamma = \frac{1}{\Gamma^m}, m \geq 1$, Γ is the given length scale for a given problem. So EkFA and the two versions of FA are compared in the following experiment. Table 3 records the computational results between the two versions FA and EkFA. “Mean” represents the average optimal value for 30 runs, and “Std Dev” is the standard deviation. As shown in the table 3, when $\gamma = 1.0$, the stand FA is hard to get the global optimal solution in most benchmark functions, but when γ change to $1/\Gamma^2$, the performance of stand FA improved significantly. f_1, f_4, f_7 and f_{11} get the approximate optimal solution, and the quality of the rest functions is also improved. EkFA is much better than the two versions of FA except f_3, f_5 and f_7, FA($\gamma = 1/\Gamma^2$) is better than EkFA on f_5 and f_7, but their optimal values are very close. In other functions, the optimal value of EkFA is significantly improved compared with the two versions of FA.

Table 3: Computational results between the two versions FA and EkFA

Function	FA($\gamma = 1.0$)		FA($\gamma = 1/\Gamma^2$)		EkFA	
	Mean	Std dev	Mean	Std dev	Mean	Std dev
f_1	6.67E+04	1.83E+04	5.14E-02	1.36E-02	**1.09E-127**	**2.86E-128**
f_2	5.19E+02	1.42E+02	1.07E+00	2.65E-01	**1.60E-64**	**2.14E-65**
f_3	2.43E+05	4.85E+04	**1.26E-01**	**1.86E-01**	1.83E+01	2.07E+01
f_4	8.35E+01	3.16E+01	9.98E−02	2.34E−02	**1.76E-64**	**3.29E-65**
f_5	2.69E+08	6.21E+07	**3.41E+01**	**6.23E+00**	4.06E+01	3.28E+01
f_6	7.69E+04	3.38E+03	5.24E+03	1.08E+03	**0.00E+00**	**0.00E+00**
f_7	5.16E+01	2.46E+01	**7.55E−02**	**1.42E−02**	9.88E-02	4.58E-02
f_8	1.10E+04	3.77E+03	9.16E+03	1.78E+03	**-6.16E+03**	**5.66E+02**
f_9	3.33E+02	6.28E+01	4.95E+01	2.39E+01	**3.90E+01**	**1.16E+01**
f_{10}	2.03E+01	2.23E−01	1.21E+01	1.96E+00	**1.75E-14**	**3.89E-15**
f_{11}	6.54E+02	1.69E+02	2.13E−02	1.47E−02	**0.00E+00**	**0.00E+00**
f_{12}	7.16E+08	1.82E+08	6.24E+00	4.62E+00	**1.57E-32**	**0.00E+00**
f_{13}	1.31E+09	4.76E+08	5.11E+01	1.28E+01	**3.72E-32**	**1.50E-32**

4.3 Comparison of EkFA with other FA variants

In order to more fully test the performance of the EkFA algorithm, we have done a comparative analysis with the several recently proposed FA variants. The details of all the parameters of the involved algorithm set as shown in Table 4.

Table4: The parameters of the algorithms

	α	α_{min}	$\alpha(0)$	γ	β	β_{min}
MFA[11]	-	-	0.5	$1/\Gamma^2$	β_0 =1.0	0.2
WSSFA[13]	-	0.04	-	1.0	β_0 =1.0	-
VSSFA[14]	-	-	-	1.0	β_0 =1.0	-
NaFA[8]	-	-	0.5	$1/\Gamma^2$	β_0 =1.0	0.2
EKFA	-	-	0.5	$1/\Gamma^2$	β_0 =1.0	0.2

The comparison results between VSSFA, WSSFA, MFA, NaFA and EkFA are recorded in table 5. Because we did not implement the code for the NaFA, the comparison data of NaFA came from [8, 15]. All the adjustment parameters of the algorithms are set as shown in table 4, and the run parameters are set to the same as shown in section 4.2. As shown in table5, VSSFA and WSSFA seem difficult to find the global optimal solution in many functions. MFA is better than VSSFA and WSSFA, especially for f_6, MFA achieves global optimal solutions. Both NaFA and EkFA obtain global optimal values on functions f_6 and f_{11}. MFA achieves 4 optimal solutions(f_5, f_6, f_7, f_8), NaFA achieves 4 optimal solutions(f_3, f_6, f_9, f_{11}), while EkFA achieves 8 optimal solutions($f_{1-2}, f_4, f_6, f_{10-13}$), the EkFA algorithm is significantly better than the other four algorithms.

Table 5: Computational results of Mean best fitness value by VSSFA, WSSFA, MFA, NaFA and EkFA

Function	VSSFA Mean	WSSFA Mean	MFA Mean	NaFA Mean	EkFA Mean
f_1	5.84E+04	6.34E+04	1.56E-06	4.43E-29	**1.09E-127**
f_2	1.13E+02	1.35E+02	1.85E-03	2.98E-15	**1.60E-64**
f_3	1.16E+05	1.10E+05	5.89E-05	**2.60E-28**	1.83E+01
f_4	8.18E+01	7.59E+01	1.73E-03	3.43E-15	**1.76E-64**
f_5	2.16E+08	2.49E+08	**2.29E+01**	2.39E+01	4.06E+01
f_6	5.48E+04	6.18E+04	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_7	4.43E+01	3.24E-01	**1.30E-01**	2.91E-02	9.88E-02
f_8	1.07E+04	1.06E+04	**4.94E+03**	6.86E+03	-6.16E+03
f_9	3.12E+02	3.61E+02	6.47E+01	**2.09E+01**	3.90E+01
f_{10}	2.03E+01	2.05E+01	4.23E-04	3.02E-14	**1.75E-14**
f_{11}	5.47E+02	6.09E+02	9.86E-03	**0.00E+00**	**0.00E+00**
f_{12}	3.99E+08	6.18E+08	5.04E-08	1.36E-31	**1.57E-32**
f_{13}	8.12E+08	9.13E+08	6.06E-07	2.13E-30	**3.72E-32**

To further compare the performance of EkFA and other FA variant algorithms, we conducted Friedman[16] test on four variants of FA. Table 6 lists the mean ranks of EkFA and three other FA variants. The best rank (minimum rank) is in bold in table. As shown in Table 6, the mean ranks values for EkFA, MFA, VSSFA and WSSFA are 1.19, 1.81, 3.31 and 3.69, respectively. So we can conclude that EkFA is the best among the 4 FA variants.

Table 6: Mean rank achieved by the Friedman test for the four FA variants. The best rank is indicated in bold.

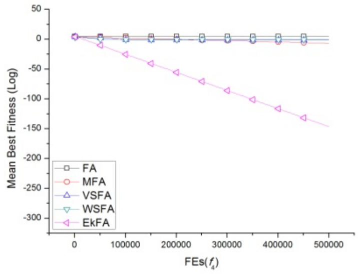
Algorithm	Mean rank
EkFA	**1.19**
MFA	1.81
VSSFA	3.31
WSSFA	3.69

Table 7 summarized the data collected during the experiment of Wilcoxon test[17] results between EkFA and other FA variants. From the literature [17], we can know that the smaller the p-values, the smaller the likelihood of the hypothesis. That is to say that the difference is more significant. As can be seen from the data in Table 7, Wilcoxon test results for most functions are much smaller than the pre-set value of 0.05, therefore, the improved EkFA algorithm has higher performance than other algorithms.

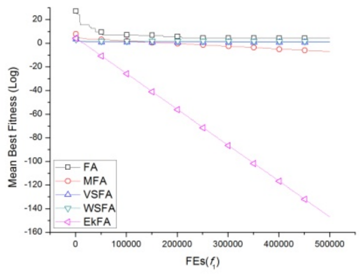
Table 7: Wilcoxon test between EkFA and the other FA variants on the test suite. The p -values below 0.05 are indicated in bold.

EkFA *vs.*	*p* -values
FA	**0.0014737808438751452**
MFA	0.11666446478102338
VSSFA	**0.0014737808438751452**
WSSFA	**0.0014737808438751452**

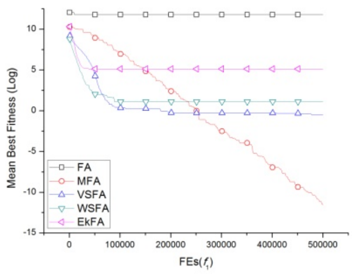
Figure 2 gives the convergence curves between VSSFA, WSSFA, MFA and EkFA. In order to increase the sensitivity of numerical differences, we make a logarithmic conversion to the fitness values of some functions (f1-f4, f7, f9-f10, f12, f13). As described in Figure 1, EkFA is significantly better than other four algorithms both at convergence speed and the solution's quality on the most functions only except f3.



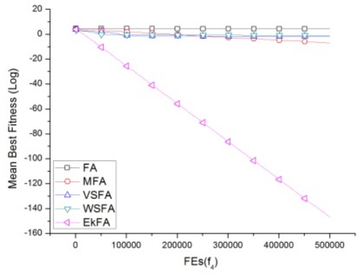
(a) Sphere (f_1)



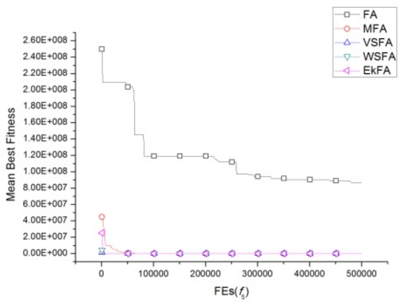
(b) Schwefel 2.22 (f_2)



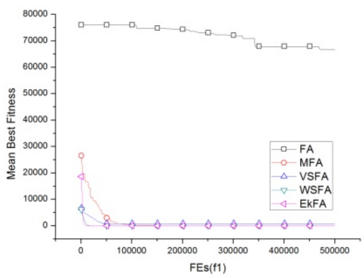
(c) Schwefel 1.2 (f_3)



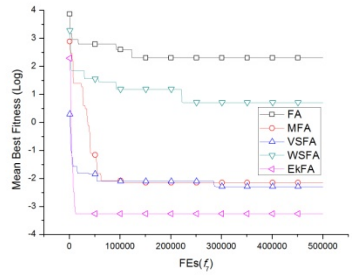
(d) Schwefel 2.21 (f_4)



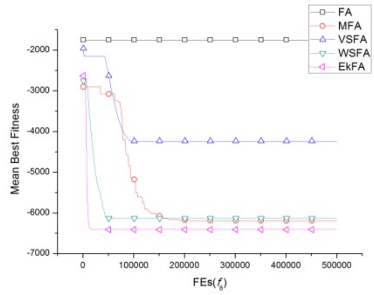
(e) Rosenbrock (f_5)



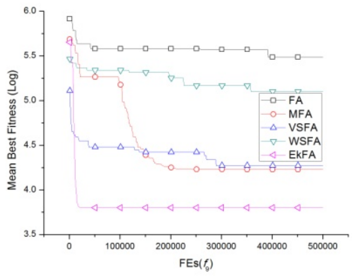
(f) Step (f_6)



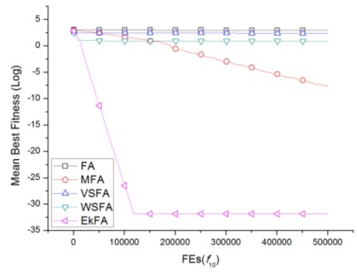
(g) Quartic with noise (f_7)



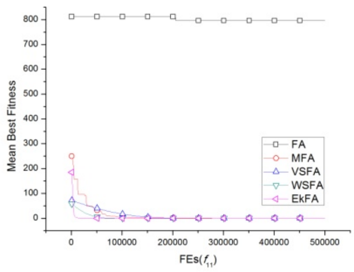
(h) Schwefel 2.26 (f_8)



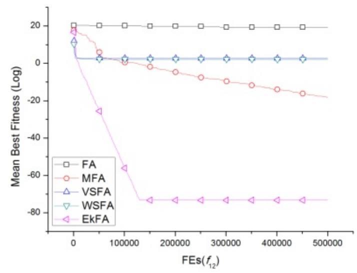
(i) Rastrigin (f_9)



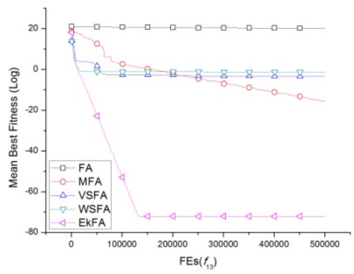
(j) Ackley (f_{10})



(k) Griewank (f_{11})



(l) Penalized 1 (f_{12})



(m) Penalized 2 (f_{13})

Fig. 2 The convergence curves of VSSFA, WSSFA, MFA, EkFA

5 Conclusions

In the standard FA, the fireflies will be attracted by all other bright fireflies, but lots of attractions are without effect, and it will increase the computational time of the algorithm. In addition, the best firefly information in all the search process have not been recorded, which may cause the algorithm inefficiency. This paper presents an elite-k neighborhood attraction firefly algorithm; Elite-k neighborhood attraction has two important roles:

- Reduce the number of attraction.
- Make full use of elite information to guide the movement of fireflies near the elite.

Thirteen well-known benchmark functions are used to verify the performance of our proposed method. The results of the experiment indicate that the accuracy and efficiency of EkFA are significantly better than the standard FA, VSSFA, WSSFA, MFA and NaFA.

In our proposed EkFA, k plays an important role. In this paper, when the population size is 30, the parameter k is set to 10, and a good solution can be obtained. In the future work, k will be further studied.

Acknowledgments. The authors would like to thank anonymous reviewers for their detailed and constructive comments that help us to increase the quality of this work. This work was supported by the National Natural Science Foundation of Jiangxi Province (No.20151BAB207023) and Sciece and Technology project of Jiangxi Provincial Department of Education (No. GJJ150448).

References

1. Yang, X.-S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press (2008)
2. SundarRajan, R., Vasudevan, V., Mithya, S.: Workflow scheduling in cloud computing environment using firefly algorithm. In: 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), pp. 955-960. (Year)
3. Marichelvam, M.K., Prabaharan, T., Yang, X.S.: A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems. IEEE transactions on evolutionary computation 18, 301-305 (2014)
4. Manshahia, M.S., Dave, M., Singh, S.B.: Firefly algorithm based clustering technique for Wireless Sensor Networks. In: 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 1273-1276. (Year)
5. Lalwani, P., Ganguli, I., Banka, H.: FARW: Firefly algorithm for Routing in wireless sensor networks. In: 2016 3rd International Conference on Recent Advances in Information Technology (RAIT), pp. 248-252. (Year)
6. Kazem, A., Sharifi, E., Hussain, F.K., Saberi, M., Hussain, O.K.: Support vector regression with chaos-based firefly algorithm for stock market price forecasting. Applied soft computing 13, 947-958 (2013)

7. Baykasoğlu, A., Ozsoydan, F.B.: Adaptive firefly algorithm with chaos for mechanical design optimization problems. Applied Soft Computing 36, 152-164 (2015)
8. Wang, H., Wang, W., Zhou, X., Sun, H., Zhao, J., Yu, X., Cui, Z.: Firefly Algorithm with Neighborhood Attraction. Information Sciences (2016)
9. Yang, X.-S.: Engineering Optimization: An Introduction with Metaheuristic Applications. Wiley Publishing (2010)
10. Yang, X.-S.: Firefly Algorithm, Lévy Flights and Global Optimization. 209-218 (2010)
11. Fister Jr, I., Yang, X.-S., Fister, I., Brest, J.: Memetic firefly algorithm for combinatorial optimization. arXiv preprint arXiv:1204.5165 (2012)
12. Lin, Y., Wang, L., Zhong, Y., Zhang, C.: Control scaling factor of cuckoo search algorithm using learning automata. International Journal of Computing Science and Mathematics 7, 476-484 (2016)
13. Yu, S., Su, S., Lu, Q., Huang, L.: A novel wise step strategy for firefly algorithm. International Journal of Computer Mathematics 91, 2507-2513 (2014)
14. Yu, S., Zhu, S., Ma, Y., Mao, D.: A variable step size firefly algorithm for numerical optimization. Applied Mathematics and Computation 263, 214-220 (2015)
15. Wang, H., Zhou, X., Sun, H., Yu, X., Zhao, J., Zhang, H., Cui, L.: Firefly algorithm with adaptive control parameters. Soft Computing (2016)
16. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180, 2044-2064 (2010)
17. Molina, D., Lozano, M., Herrera, F.: A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization. Journal of Heuristics 15, 617-644 (2009)