Neural Computing and Applications (2019) 31:1445–1453
https://doi.org/10.1007/s00521-018-3449-6

CrossMark

A new and efficient firefly algorithm for numerical optimization problems

Xiuqin Pan[1] · Limiao Xue[1] · Ruixiang Li[1]

Received: 4 January 2018 / Accepted: 16 March 2018 / Published online: 23 March 2018
© The Natural Computing Applications Forum 2018

RETRACTED ARTICLE

Abstract

Firefly algorithm (FA) is an excellent global optimizer based on swarm intelligence. Some recent studies show that FA was used to optimize various engineering problems. However, there are some drawbacks for FA, such as slow convergence rate and low precision solutions. To tackles these issues, a new and efficient FA (namely NEFA) is proposed. In NEFA, three modified strategies are employed. First, a new attraction model is used to determine the number of attracted fireflies. Second, a new search operator is designed for some better fireflies. Third, the step factor is dynamically updated during the iterations. Experiment verification is carried out on ten famous benchmark functions. Experimental results demonstrate that our new approach NEFA is superior to three other different versions of FA.

Keywords Firefly algorithm · Convergence speed · Attraction · Adaptive parameter

1 Introduction

In real world, many practical engineering problems can be formulated to optimization problem over continuous or discrete search space. A general unconstrained optimization problem can be defined as follows:

$$\min f(X) \tag{1}$$

where $X = [x_1, x_2, \ldots, x_D]$ is a potential solution in a D-dimensional search space.

With increasing demand and environmental changes, many optimization problems have become complex and difficult, such as nonlinear, multimodal, discrete, strong constraints, large-scale and many-objective. To solve those complex problems, more efficient optimization algorithms are needed. In the past decades, some new iterative optimization techniques have been designed based on Darwinian evolutionary theory "survival of the fittest," such as particle swarm optimization (PSO) [1, 2], ant colony optimization (ACO) [3], firefly algorithm (FA) [4], artificial bee colony (ABC) [5, 6], cuckoo search (CS) [7, 8] and bat algorithm (BA) [9, 10]. Among these optimization algorithms, FA is a popular optimizer, which mimics the mating attraction behaviors among fireflies. A recent study showed that FA was used to optimize various problems [11].

In the standard FA, a brighter (better) firefly can attract other all darker (worse) fireflies. Then, those worse fireflies can move to other better positions. At each iteration, each firefly moves to other all better ones. Thus, there are many attractions among fireflies. Too many attractions will lead to slow convergence rate and low accuracy of solutions [12]. In our approach, a modified attraction model is used to determine the number of attractions. In addition, the performance of FA is seriously affected by its step factor α [13]. To tackle this issue, an adaptive parameter strategy is employed. To validate the performance of our approach NEFA, some simulation studies are performed on a set of test functions. Experimental results demonstrate our NEFA is superior to three other different versions of FA.

The rest of paper is organized as follows. FA and its recent progress are reviewed in Sect. 2. Our proposed NEFA is given in Sect. 3. Computational results on the

✉ Xiuqin Pan
amycun@163.com

Limiao Xue
xuelimiao09@163.com

Ruixiang Li
ruixiang0822@163.com

1 School of Information Engineering, Minzu University of China, Beijing 100081, China

benchmark set are presented in Sect. 4. Finally, our work is concluded in Sect. 5.

2 Related work

2.1 Firefly algorithm

Firefly algorithm (FA) was firstly developed by Yang [14], which is inspired by the flashing light of fireflies in the summer sky. The flashing light can attract mating partners or potential prey. Based on the attraction behavior, Yang [14] built the original FA.

In FA, there is a set to initial solutions consisting of the initial population. Each firefly is regarded as a potential solution in the search space. Assume that N is the population size, and X_i is the ith solution in the population, where $i = 1, 2,\ldots,N$.

The light intensity (I) usually decreases with the increase in distance. According to the literature [14], the light intensity can be defined as follows [14]:

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

where I_0 is the initial light intensity and γ is called light absorption coefficient. The attractiveness β is defined as follows [14]:

$$\beta = \beta_0 \mathrm{e}^{-\gamma r^2} \tag{3}$$

where β_0 is a constant value and it is usually equal to 1.0.

For any two fireflies X_i and X_j, their distance can be calculated by [14]

$$r_{ij} = \|X_i - X_j\| = \sqrt{\sum_{d=1}^{D} (x_{id} - x_{jd})^2} \tag{4}$$

where x_{id} and x_{jd} are the dth component of X_i and X_j, respectively.

When X_j is brighter (better) than X_i, X_i is attracted to X_j. It means that X_i will move to X_j because of the attraction. The movement of fireflies is defined as follows [14]:

$$x_{id} = x_{id} + \beta \cdot (x_{jd} - x_{id}) + \alpha \cdot (rand - 0.5) \tag{5}$$

where α is called step factor and *rand* is a random value uniformly generated in the range [0,1].

2.2 Literature review

In recent years, many researchers have paid attention to FA. Different FA variants were proposed to solve various benchmark and practical problems. In this section, we present a brief literature review of this work.

Fister et al. [15] proposed a memetic FA (namely MFA), which uses two new parameter methods. First, the step factor α is dynamically changed. Second, the attractiveness β is constrained in a box range. In [16], MFA is combined with three neighborhood search strategies to obtain better performance. Wang et al. [13] proposed an adaptive FA (called ApFA), which uses an adaptive parameter method to set the step factor α. Results demonstrate that ApFA is better than MFA and FA. Tighzert et al. [17] proposed several new compact FA (cFA) variants to reduce the computational cost. Simulation results confirm that cFAs are very competitive. Cheung et al. [18] presented a non-homogeneous FA and analyzed the trajectory of a single firefly during the search. Yelghi and Köse [19] presented a tidal force FA for global minimum optimization problems, in which the tidal force formula is used for exploitation. Tilahun et al. [20] reviewed some recently published FA variants on continuous optimization problems and gave some possible future works for FA.

Zouache et al. [21] combined Quantum FA and PSO for solving 0-1 simple knapsack problem and multidimensional knapsack problem. Simulation results show that the proposed algorithm outperforms some existing methods. Wang et al. [22] used a hybrid multiobjective FA (HMOFA) to solve big data optimization problems. Results show the effectiveness of HMOFA. He and Huang [23] presented a modified firefly algorithm to seek the optimal multilevel threshold values of color image. To improve the performance, the search idea of PSO is introduced to enhance the movement of fireflies. Lieu et al. [24] designed an adaptive hybrid evolutionary FA (AHEFA) to optimize the truss structure. Simulation experiments on six test examples show that AHEFA can achieve promising performance.

3 Proposed approach

To overcome the drawbacks of FA, this paper proposes a new and efficient FA (namely NEFA). In NEFA, three modified strategies are employed. First, a new attraction model is used to determine the number of attracted fireflies. Second, a new search operator is designed for some better fireflies. Third, the step factor is dynamically updated during the iterations.

3.1 Modified attraction model

The attraction model is important to the performance of FA. The standard FA employs a full attraction model, in which each firefly can be attracted to other all brighter fireflies. Thus, there are many attractions among fireflies at each iteration. Too many attractions may lead to the

RETRACTED ARTICLE

RETRACTED ARTICLE

oscillation of search and slow convergence rate. To tackle this problem, several new attraction models were proposed. In [25], Wang et al. designed a random attraction model. In [12], Wang et al. presented another model called neighborhood attraction. These improved attraction models obtained better performance than the full attraction model in the standard FA. In this paper, we propose a modified attraction model, which can be described in the following steps.

(1) For each firefly (solution) X_i, M different solutions $\{X_{r1}, X_{r2},\ldots,X_{rM}\}$ are randomly selected from the current population, where $i \neq r1 \neq r2 \neq \ldots \neq rM$.
(2) X_i is compared with the selected M solutions. If X_{rj} is better than X_i, then X_i will move to X_{rj}, where $j = 1, 2,\ldots, M$.

In our design, M is much smaller than the population size N. So, the number of attractions in our approach is much less than the standard FA. Figure 1 presents the full attraction model in the standard FA. As seen, there are 10 fireflies in the population, and firefly i may be attracted to other 9 fireflies. Figure 2 shows the modified attraction model in our approach. It can be seen that three blue fireflies are randomly selected from the population, and $M = 3$. Then, firefly i is attracted to other 3 fireflies at most.

3.2 New search strategy

In the standard FA, if the current firefly is better than the compared firefly, the current firefly will move randomly. It is known that random movement is not beneficial for the search. To tackle this problem, a new search strategy is employed for brighter fireflies.

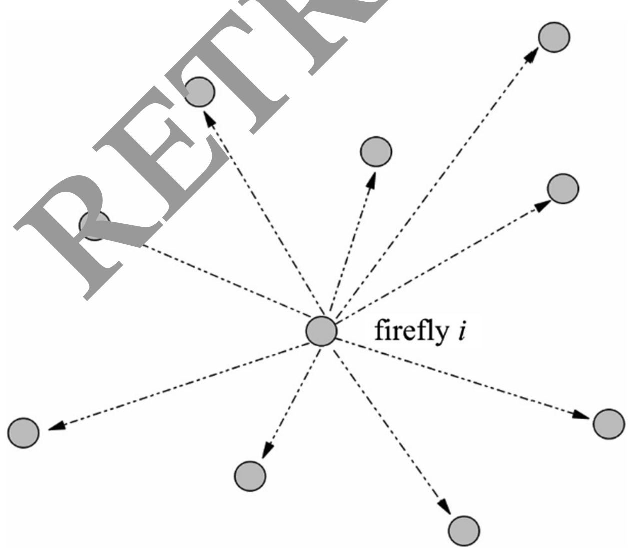

Fig. 1 Full attraction model in the standard FA

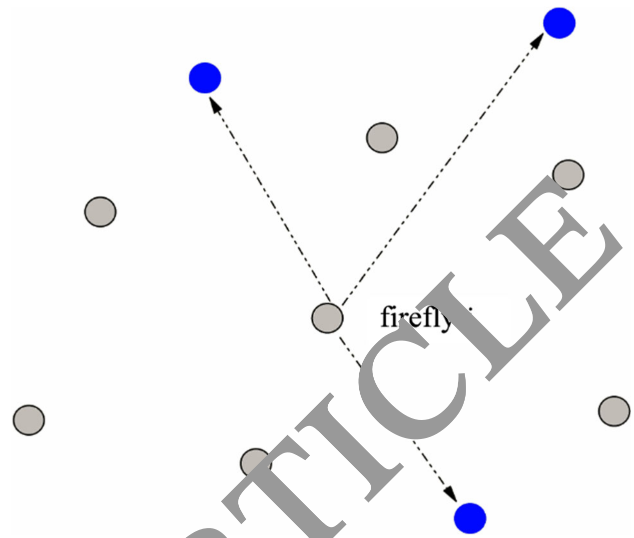

Fig. 2 Modified attraction model in NEFA

If X_j is brighter (better) than X_i, X_i is attracted to X_j; otherwise, X_i is conducted on the following search strategy:

$$x_{id}^{*} = x_{id} + \varphi \cdot (x_{id} - x_{hd}) \tag{6}$$

where x_{hd} is the dth component of X_h, X_h is randomly selected from the population, and φ is a random value uniformly generated in the range [− 1, 1]. The idea of Eq. (6) is inspired by the solution updating model of ABC.

We also use a greedy method to select the solution between X_i an X_i^* as follows:

$$X_i = \begin{cases} X_i^*, & \text{if } f\left(X_i^*\right) < f(X_i) \\ X_i, & \text{otherwise} \end{cases}. \tag{7}$$

3.3 Adaptive parameter strategy

Like PSO, the performance of FA is sensitive to its control parameters. Different parameter settings may result in different performance. In the literature [13], Wang et al. analyzed the relationship between the step factor α and convergence. When FA is convergent, the parameter α should satisfy the following condition [13]:

$$\lim_{t \to \infty} \alpha = 0. \tag{8}$$

Based on Eq. (8), an adaptive parameter method was designed to adjust the parameter α as follows [13]:

$$\alpha(t+1) = \left(1 - \frac{t}{T_{\max}}\right) \cdot \alpha(t) \tag{9}$$

where t is the index of iterations, $T_{\max}$ is the maximum number of iterations, and $\alpha(t)$ is the value of α at the tth iteration.

RETRACTED ARTICLE

4 Experimental verification

4.1 Test functions

This paper proposes a new FA variant called NEFA. To validate the performance of NEFA, a set of ten benchmark functions are tested. These functions were used to test the performance of optimization algorithms [26–28]. For these functions, their mathematical definitions, search range and global optimum are listed in Table 1. All test functions are minimization problems, and their minimal values are given in the third column of Table 1.

4.2 Effects of the parameter *M*

In Sect. 3.1, NEFA employs a modified attraction model. In this model, each firefly is attracted to M randomly selected fireflies. In the full attraction model, each firefly is attracted to $N - 1$ fireflies. In general, M is much less than N. So, the modified attraction model can reduce the computational complexity. However, the parameter M can seriously affect the performance of FA. When M is equal to $N - 1$, the proposed attraction model is equal to the full attraction model. Therefore, it is worth investigating the effects of the parameter M.

In this section, the parameter M is set to different values. Then, we use NEFA with different M to test the benchmark set. Finally, we can select the best choice of M. In the experiment, the parameter M is set to 3, 6 and 10, respectively. The population size N is equal to 20, and the dimensional size D is set to 30 [13]. The maximum number of fitness evaluations (Max_FEs) is set to 5.0E+05. The initial $\alpha(0)$, β_0 and γ are set to 0.5, 1.0 and $1/\Gamma^2$, respectively, where Γ is the length of search range. For example, the search range of function f_1 is [− 100,100], and the length of the search range is 200. Then, Γ is equal to 200 for this function.

Table 2 presents the computational results of NEFA with different M values, where "Man" is the mean best fitness value over 30 runs. For each test function, the best result among different M values is shown in bold. From the results, M = 3 achieves better results than other M values on f_2. For M = 6, it can find better solutions than M = 3 and 10 on f_3, f_5, f_7, f_8 and f_{10}. M = 10 obtains better solutions than other M values on f_1 and f_4. All three M values achieve the same results on f_6 and f_9. In order to clearly observe the effects of the parameter M, Figs. 3, 4, 5, 6 and 7 display the convergence curves of NEFA with different M values. Based on the above analysis, M = 6 is regarded as the best choice on the benchmark set. Therefore, M = 6 is used in the following experiment.

4.3 Comparison of NEFA with other FA variants

In this section, NEFA is compared with three other FA variants with D = 10 and 30. For testing the effectiveness and superiority of the proposed NEFA, the same conditions are used to compare with other existing optimization approaches such as FA [4], ApFA [13] and MFA [15].

Table 1 Test functions used in the experiments

Functions	Search range	Min
$f_1(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]$	0
$f_2(x) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$	$[-10, 10]$	0
$f_3(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j\right)^2$	$[-100, 100]$	0
$f_4(x) = \max_i\{\lvert x_i\rvert, 1 \le i \le D\}$	$[-100, 100]$	0
$f_5(x) = \sum_{i=1}^{D-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	$[-30, 30]$	0
$f_6(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5\rfloor)^2$	$[-100, 100]$	0
$f_7(x) = \sum_{i=1}^{D} i x_i^4 + rand[0, 1)$	$[-1.28, 1.28]$	0
$f_8(x) = \sum_{i=1}^{D} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	$[-500, 500]$	$-418.98 \cdot D$
$f_9(x) = \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	$[-5.12, 5.12]$	0
$f_{10}(x) = -20 \cdot \exp\left(-0.2 \cdot \sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]$	0

Table 2 Computational results for NEFA with different M values, where the best results are shown in bold

Functions	M = 3 Mean	M = 6 Mean	M = 10 Mean
f_1	2.63E−42	1.67E−92	**7.39E−125**
f_2	**2.14E−288**	5.15E−114	3.65E−90
f_3	8.14E−08	**7.18E−12**	1.15E−07
f_4	6.48E−03	1.43E−04	**6.75E−07**
f_5	2.74E+01	**2.47E+01**	2.53E+01
f_6	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_7	7.81E−03	**5.30E−03**	1.15E−02
f_8	7.27E+03	**6.49E+03**	6.53E+03
f_9	3.48E+01	**2.89E+01**	4.58E+01
f_{10}	**4.14E−15**	**4.14E−15**	**4.14E−15**

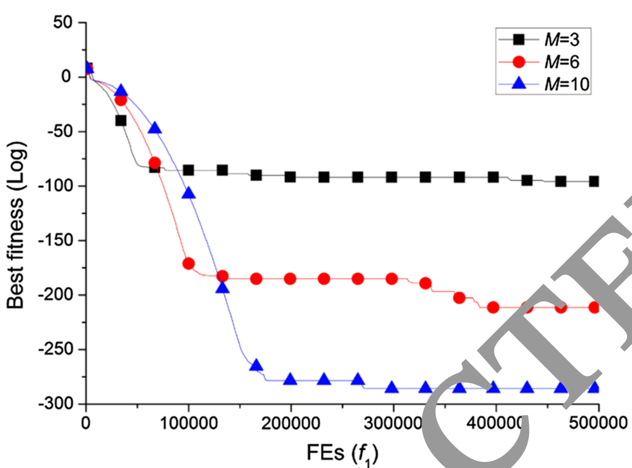

Fig. 3 Convergence curves of NEFA with different M values on function f_1

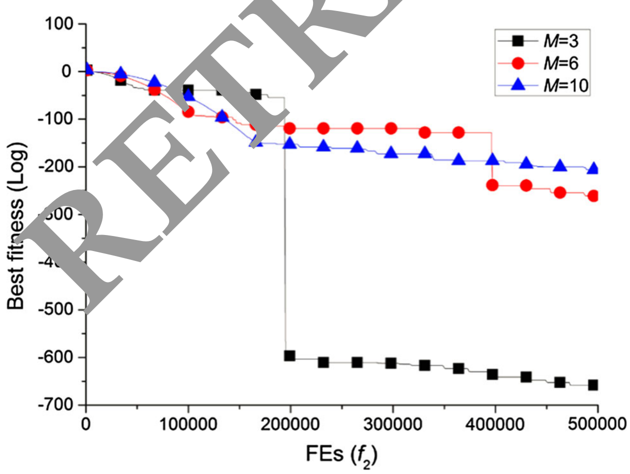

Fig. 4 Convergence curves of NEFA with different M values on function f_2

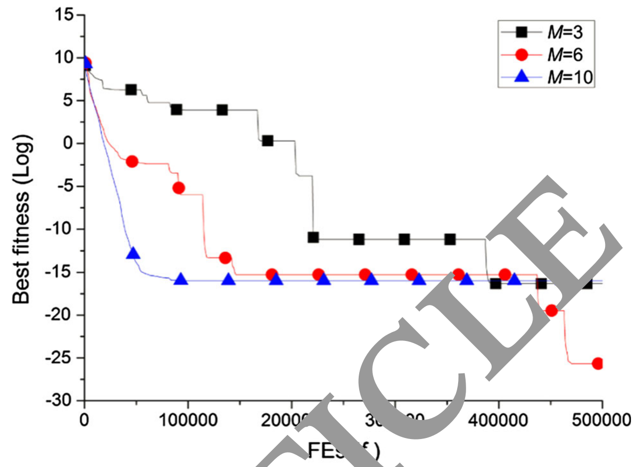

Fig. 5 Convergence curves of NEFA with different M values on function f_3

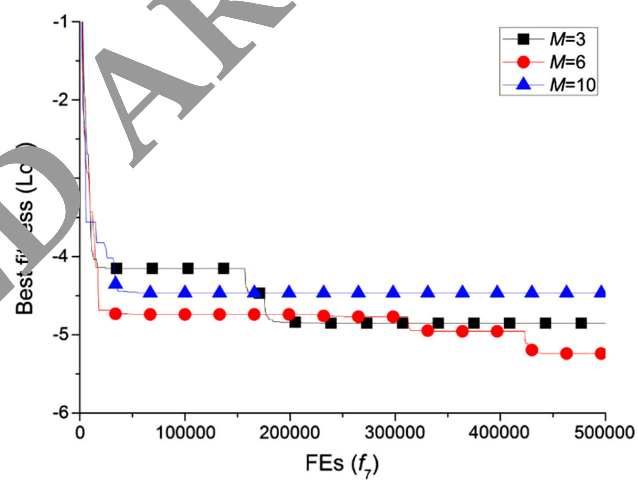

Fig. 6 Convergence curves of NEFA with different M values on function f_7

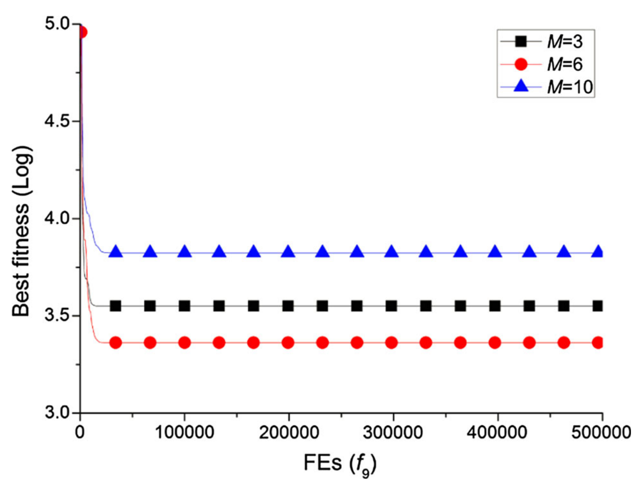

Fig. 7 Convergence curves of NEFA with different M values on function f_9

RETRACTED ARTICLE

RETRACTED ARTICLE

For each function, each algorithm is run 30 trials. To have a fair comparison, all algorithms use the same parameters. For all FA variants, the parameters N and γ are equal to 20 and $1/\Gamma^2$, respectively. For MFA, the parameter α is dynamically adjusted, and the parameters β_0 and β_{min} are set according to the literature [15]. For ApFA and NEFA, the parameter α use the same updating method, and the β_0 is set to 1.0 [13]. The parameter M used in NEFA is set to 6. For $D = 10$, the $MaxFEs$ is equal to 1.5E+05. When D increases to 30, the $MaxFEs$ is set to 5.0E+05.

Table 3 presents the computational results of four FA variants for $D = 10$, where "Man" is the mean best fitness value. From the results, NEFA outperforms FA on all test functions. For most test functions, NEFA and ApFA achieve much better solutions than FA. Compared to MFA, NEFA shows worse performance on functions f_5 and f_8. For function f_6, NEFA, ApFA and MFA can converge to the global optimum. For the rest of 7 functions, NEFA performs better than MFA. Especially for f_1–f_4 and f_{10}, NEFA obtains much better solutions than MFA. Both NEFA and ApFA use the same parameter strategy to control the step factor α. NEFA is superior to ApFA on six functions f_1–f_3, f_5, f_8 and f_{10}, while ApFA is better than NEFA on 3 functions f_4, f_7 and f_9. From the above analysis, NEFA can find more accurate solutions than ApFA, MFA and FA on most test functions.

Figures 8, 9, 10 and 11 list the convergence curves of FA, MFA, ApFA and NEFA on functions f_1–f_4 ($D = 10$). As shown, NEFA converges much faster than ApFA, MFA and FA on three functions f_1–f_3. For function f_4, ApFA is faster than NEFA, MFA and FA, but NEFA shows faster convergence than ApFA at the initial search stage.

Table 4 presents the computational results of four FA variants for $D = 30$, where "Man" is the mean best fitness value. Similar to $D = 10$, NEFA is superior to FA on all

Table 3 Computational results for $D = 10$, where the best results are shown in bold

Functions	FA Mean	MFA Mean	ApFA Mean	NEFA Mean
f_1	2.87E−03	1.33E−04	4.33E−54	**2.47E−160**
f_2	1.14E−01	3.01E−03	8.53E−27	**2.29E−101**
f_3	5.36E−03	1.31E−04	1.52E−13	**2.37E−36**
f_4	3.85E−02	9.48E−03	**3.05E−19**	1.16E−08
f_5	5.05E+00	**2.41E−01**	6.45E+00	2.83E+00
f_6	1.52E+03	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_7	4.79E−03	3.84E−04	**5.95E−04**	1.47E−03
f_8	2.63E+03	**1.28E+03**	1.94E+03	1.54E+03
f_9	1.67E+01	1.59E+01	**5.97E+00**	9.95E+00
f_{10}	1.28E+01	6.54E−03	7.69E−15	**5.89E−16**

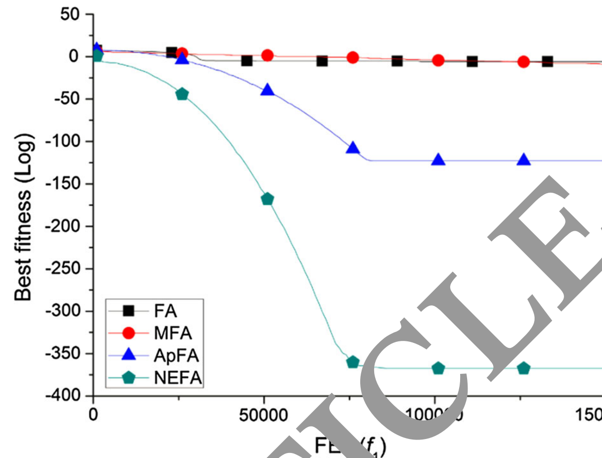

Fig. 8 Convergence curves of FA, MFA, ApFA and NEFA on function f_1 ($D = 10$)

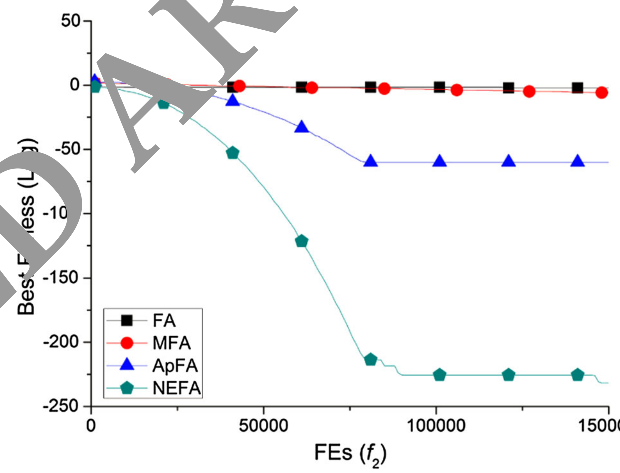

Fig. 9 Convergence curves of FA, MFA, ApFA and NEFA on function f_2 ($D = 10$)

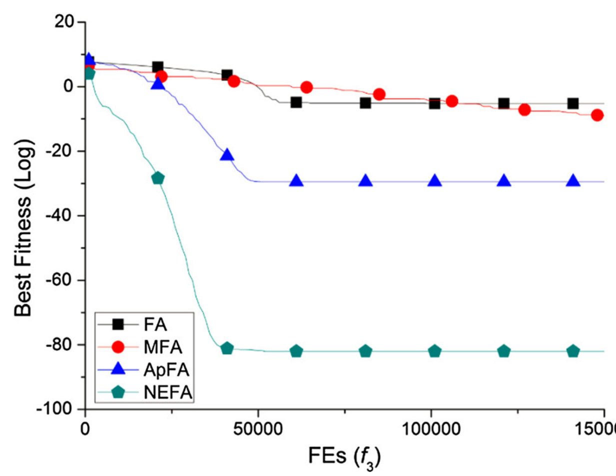

Fig. 10 Convergence curves of FA, MFA, ApFA and NEFA on function f_3 ($D = 10$)

RETRACTED ARTICLE

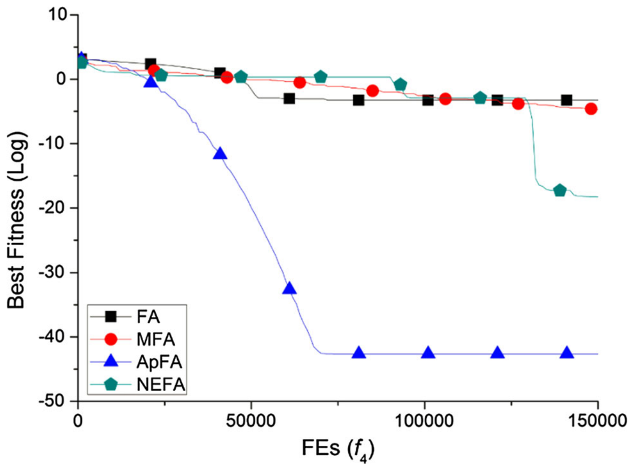

Fig. 11 Convergence curves of FA, MFA, ApFA and NEFA on function f_4 (D = 10)

Table 4 Computational results for D = 30, where the best results are shown in bold

Functions	FA Mean	MFA Mean	ApFA Mean	NEFA Mean
f_1	5.37E−02	6.56E−07	2.34E−46	**1.67E−92**
f_2	9.75E−01	3.64E−04	1.51E−15	**5.15E−114**
f_3	1.93E−01	2.53E−06	1.86E+01	**7.18E[illegible]12**
f_4	1.22E−01	3.32E−04	**2.32E−08**	1.4[illegible]04
f_5	3.47E+01	**2.17E+01**	2.85E+01	2.47E+0[illegible]
f_6	3.62E+03	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_7	8.07E−02	8.32E−02	**1.84E[illegible]3**	5.3[illegible]03
f_8	8.08E+03	**4.86E+03**	6.8[illegible]E+03	6.49E+03
f_9	4.71E+01	4.58E+01	**2.57[illegible]+01**	2.89E+01
f_{10}	1.29E+01	1.92E−04	1.48E−14	**4.14E−15**

test cases, and NEFA, ApFA and MFA can converge to the global optimum on function f_6. MFA performs better than NEFA, ApFA and FA on f_5 and f_8, while NEFA is better than MFA on functions f_1–f_4, f_7, f_9 and f_{10}. Compared to ApFA, NEFA achieves better solutions on six functions f_1–f_3, f_5, f_8 and f_{10}, while ApFA is better than NEFA on 3 functions f_4, f_7 and f_9. From Tables 3 and 4, NEFA outperforms ApFA, MFA and FA on most functions with D = 10 and 30.

Figures 12, 13, 14 and 15 give the convergence curves of FA, MFA, ApFA and NEFA on functions f_1–f_4 (D = 30). From the results, NEFA converges much faster than ApFA, MFA and FA on three functions f_1–f_3. For the rest of function f_4, ApFA is much faster than NEFA, MFA and FA, while NEFA is faster than ApFA, MFA and FA at the beginning stage. As the iteration grows, MFA converges faster than NEFA and FA. At the last search stage, NEFA shows faster convergence speed than ApFA and FA.

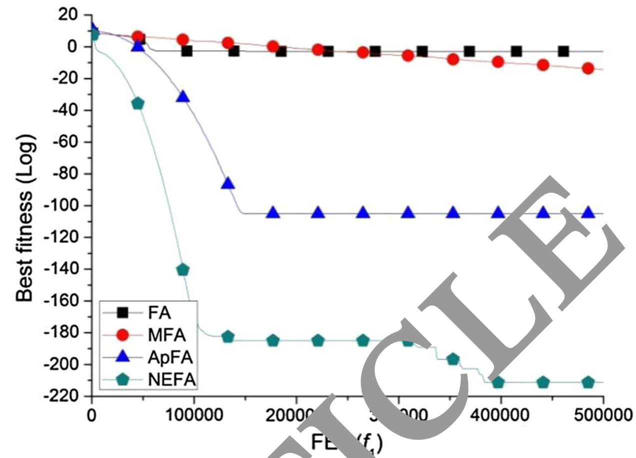

Fig. 12 Convergence curves of FA, MFA, ApFA and NEFA on function f_1 (D = 30)

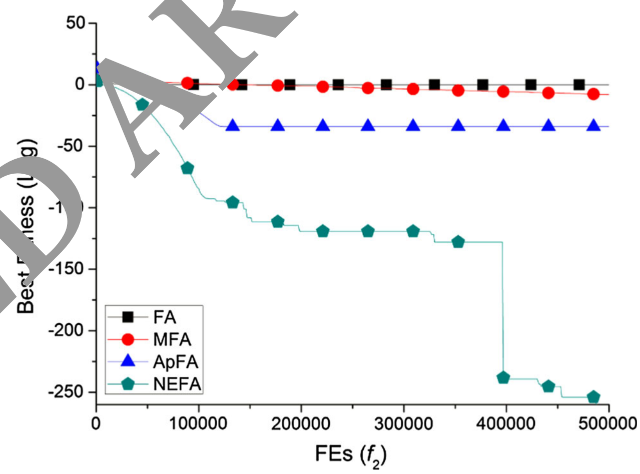

Fig. 13 Convergence curves of FA, MFA, ApFA and NEFA on function f_2 (D = 30)

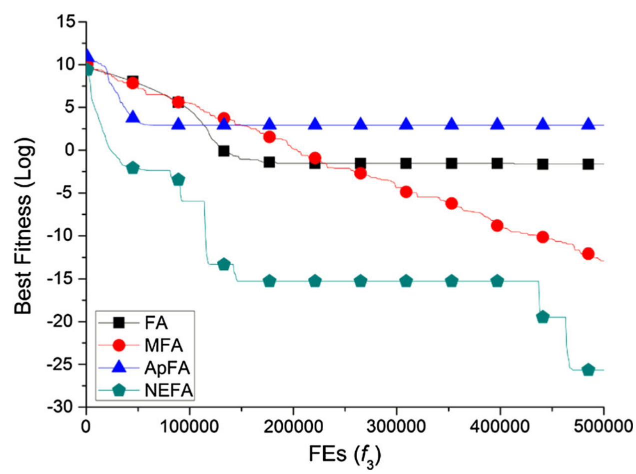

Fig. 14 Convergence curves of FA, MFA, ApFA and NEFA on function f_3 (D = 30)

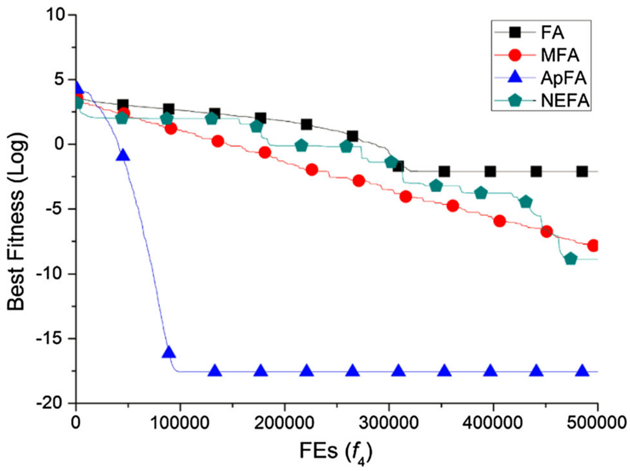

Fig. 15 Convergence curves of FA, MFA, ApFA and NEFA on function f_4 (D = 30)

5 Conclusions

In order to address the drawbacks of FA, this paper proposes a new and efficient FA (called NEFA), which employs three modified strategies. First, a modified attraction model is used to determine the number of attractions among fireflies. This is helpful to reduce the computational complexity and accelerate the convergence rate. Second, a new search strategy is applied to search better solutions in the attraction. It aims to strengthen the local search and provide more accurate solutions. Third, the step factor α is dynamically updated in order to avoid manually setting the parameter value. To validate the performance of the new approach NEFA, ten benchmark functions with different dimensions (10 and 30) are tested in the experiments.

Computational results show that NEFA with different M values can obtain different optimization performance. According to the experimental analysis, M = 6 is a good setting for the used benchmark set. Compared to ApFA, MFA and FA, NEFA can find better solutions on most test functions for D = 10 and 30.

Although we investigate the parameter M and obtain a good setting choice, M = 6 is a compromising setting. How to set the parameter M may need other strategies. For example, we may use a dynamical method to adjust the parameter M. This will be studied in our future work.

Acknowledgements This work is supported by the project of the First-Class University and the First-Class Discipline (No. 10301-017004011501), and the National Natural Science Foundation of China.

Compliance with ethical standards

Conflict of interest We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

1. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, pp 1942–1948
2. Wang H, Sun H, Li C, Rahnamayan S, Pan JS (2013) Diversity enhanced particle swarm optimization with neighborhood search. Inf Sci 223:119–135
3. Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern Part B (Cybern) 26(1):29–41
4. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimization. Int J Bio-Inspired Comput 2(2):78–84
5. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Erciyes University, engineering Faculty, Computer Engineering Department
6. Wang H, Wu ZJ, Rahnamayan S, Sun H, Liu Y, Pan JS (2014) Multi-strategy ensemble artificial bee colony algorithm. Inf Sci 279:587–603
7. Yang XS, Deb S (2010) Engineering optimisation by cuckoo search. Int J Math Model Numer Optim 1(4):330–343
8. Zhang MQ, Wang H, Cui ZH, Chen JJ (2017) Hybrid multi-objective cuckoo search with dynamical local search. Memet Comput. https://doi.org/10.1007/s12293-017-0237-2
9. Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: González JR, Pelta DA, Cruz C, Terrazas G, Krasnogor N (eds) Nature inspired cooperative strategies for optimization (NICSO 2010). Studies in computational intelligence, vol 284. Springer, Berlin
10. Cai XJ, Wang H, Cui ZH, Cai JH, Xue Y, Wang L (2017) Bat algorithm with triangle-flipping strategy for numerical optimization. Int J Mach Learn Cybern. https://doi.org/10.1007/s13042-017-0739-8
11. Fister JI, Fister I, Yang XS, Brest J (2013) A comprehensive review of firefly algorithms. Swarm Evol Comput 13:34–46
12. Wang H, Wang WJ, Zhou XY, Sun H, Zhao J, Yu X, Cui ZH (2017) Firefly algorithm with neighborhood attraction. Inf Sci 382–383:374–387
13. Wang H, Zhou XY, Sun H, Yu X, Zhao J, Zhang H, Cui LZ (2017) Firefly algorithm with adaptive control parameters. Soft Comput 21(17):5091–5102
14. Yang XS (2008) Nature-inspired metaheuristic algorithms. Luniver Press, Beckington
15. Fister JI, Yang XS, Fister I, Brest J (2012) Memetic firefly algorithm for combinatorial optimization. In: Bioinspired optimization methods and their applications (BIOMA), pp 1–14
16. Wang H, Cui ZH, Sun H, Rahnamayan S, Yang XS (2017) Randomly attracted firefly algorithm with neighborhood search and dynamic parameter adjustment mechanism. Soft Comput 21(18):5325–5339
17. Tighzert L, Fonlupt C, Mendil B (2017) A set of new compact firefly algorithms. Swarm Evol Comput. https://doi.org/10.1016/j.swevo.2017.12.006

RETRACTED ARTICLE

18. Cheung NJ, Ding XM, Shen HB (2016) A non-homogeneous firefly algorithm and its convergence analysis. J Optim Theory Appl 170(2):616–628
19. Yelghi A, Köse C (2018) A modified firefly algorithm for global minimum optimization. Appl Soft Comput 62:29–44
20. Tilahun SL, Ngnotchouye JMT, Hamadneh NN (2017) Continuous versions of firefly algorithm: a review. Artif Intell Rev. https://doi.org/10.1007/s10462-017-9568-0
21. Zouache D, Nouioua F, Moussaoui A (2016) Quantum-inspired firefly algorithm with particle swarm optimization for discrete optimization problems. Soft Comput 20(7):2781–2799
22. Wang H, Wang WJ, Cui LZ, Sun H, Zhao J, Wang Y, Xue Y (2017) A hybrid multi-objective firefly algorithm for big data optimization. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2017.06.029
23. He L, Huang S (2017) Modified firefly algorithm based multilevel thresholding for color image segmentation. Neurocomputing 240:152–174
24. Lieu QX, Doand DTT, Lee J (2018) An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. Comput Struct 195:99–112
25. Wang H, Wang WJ, Sun H, Rahnamayan S (2016) Firefly algorithm with random attraction. Int J Bio-Inspired Comput 8(1):33–41
26. Wang H, Rahnamayan S, Sun H, Omran MGH (2013) Gaussian bare-bones differential evolution. IEEE Trans Cybern 43(2):634–647
27. Zhou XY, Wang H, Wang MW, Wan JY (2017) Enhancing the modified artificial bee colony algorithm with neighborhood search. Soft Comput 21(10):2733–2743
28. Wang H, Wu ZJ, Rahnamayan S, Liu Y, Ventresca M (2011) Enhancing particle swarm optimization using generalized opposition-based learning. Inf Sci 181(20):4699–4714

RETRACTED ARTICLE