

A New Adaptive Firefly Algorithm for Solving Optimization Problems

Wenjun Wang¹, Hui Wang^{2,3}(✉), Jia Zhao^{2,3}, and Li Lv^{2,3}

¹ School of Business Administration, Nanchang Institute of Technology,
Nanchang 330099, China

² Jiangxi Province Key Laboratory of Water Information Cooperative Sensing
and Intelligent Processing, Nanchang Institute of Technology,
Nanchang 330099, China
huiwang@whu.edu.cn

³ School of Information Engineering, Nanchang Institute of Technology,
Nanchang 330099, China

Abstract. Firefly algorithm (FA) is an efficient optimization technique, which has been widely used to solve various engineering problems. However, FA is sensitive to its control parameters. Recently, a memetic FA (called MFA) is proposed to improve the sensitivity of FA. To further enhance the performance of MFA, this paper proposes a new method to adaptively adjust the step factor. Experiments on several benchmark problems show that our approach is superior to the standard FA, MFA, and some other improved FAs.

Keywords: Firefly algorithm · Dynamic · Adaptive parameter · Optimization

1 Introduction

Optimization problems exist in many engineering fields. With the growth of problem complexity, stronger optimization algorithms are required. In the past decades, several new optimization techniques have been proposed by the inspiration of swarm intelligence, such as artificial bee colony (ABC) [1–4], bat algorithm (BA) [5–7], firefly algorithm (FA) [8–10], cuckoo search (CS) [11, 12], fruit fly optimization (FFO) [13], and artificial plant optimization algorithm [14, 15]. FA is a new optimization technique originally invented by Prof. Yang [8]. In FA, every firefly moves toward new positions and find potential solutions by the attraction of other brighter fireflies. Some latest researches proved that FA is an efficient optimization tool.

However, the standard FA still has some drawbacks. For instance, FA is sensitive to the control parameters, and the convergence speed is slow. To tackle these issues, several improved strategies have been designed. Recently, a memetic FA (MFA) was proposed, in which two improved strategies are employed [16]. First, the step factor α is dynamically decreased based on an empirical model. Second, the attractiveness β is constrained into a predefined range [0.2, 1.0]. Moreover, the factor α is multiplied by the length of search interval. Gandomi et al. [17] presented a chaotic FA, in which different chaotic maps were used to adjust the step factor α and the light absorption coefficient γ . Besides Gandomi's work, some researchers also combined FA with chaos

to obtain a good performance. In [18], an adaptive selection method was used to choose the parameter α from a candidate set. In [19], Wang et al. investigated the relationships between convergence and the parameter α . Results show that α should tend to zero when FA converges to a solution. Based on this principle, Wang et al. designed two dynamic methods adjust α , in which α is gradually decreased based on different models as the generation increases [19, 20].

In this paper, we present a new adaptive firefly algorithm (AFA), which is an enhanced version of MFA. In AFA, we combine MFA with a new adaptive parameter strategy to dynamically adjust the step factor α . Thirteen famous test functions are used for performance verification. Results show that AFA is superior to the standard FA, MFA [16], chaotic FA (CFA) [17], and FA with random attraction (RaFA) [9].

2 Firefly Algorithm

For two fireflies X_i and X_j , their attractiveness β is defined as follows [21].

$$\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \quad (1)$$

where β_0 is the attractiveness at $r = 0$, γ is the light absorption coefficient, and r_{ij} is the distance between X_i and X_j . The distance r_{ij} is computed as follows [21].

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

where D is the problem size.

When X_j is brighter (better) than X_i , X_i will move towards X_j because of the attraction. In the standard FA, this movement is defined as follows [21].

$$x_{id}(t+1) = x_{id}(t) + \beta(r_{ij}) \cdot (x_{jd}(t) - x_{id}(t)) + \alpha \left(rand - \frac{1}{2} \right) \quad (3)$$

where x_{id} and x_{jd} are the d th dimensions of X_i and X_j , respectively, $\alpha \in [0, 1]$ is called step factor, and $rand$ is a random value within $[0, 1]$.

3 Proposed Approach

Recently, a memetic firefly algorithm (MFA) was designed to enhance the performance of FA [16]. The MFA made three improvements. Firstly, the step factor α is dynamically updated as follows.

$$\alpha(t+1) = \left(\frac{1}{9000} \right)^{\frac{1}{t}} \alpha(t) \quad (4)$$

where t indicates the generation index. We can find that the value of α decreases with the growth of t .

Secondly, the definitions of the attractiveness β is modified. The new β is calculated as follows.

$$\beta(r_{ij}) = \beta_{\min} + (\beta_0 - \beta_{\min})e^{-\gamma r_{ij}^2} \tag{5}$$

where β_{\min} is the minimum value of the attractiveness β . The β is constrained into the range $[\beta_{\min}, \beta_0]$. In [16], β_{\min} and β_0 are equal to 0.2, and 1.0, respectively.

Thirdly, the step factor α is multiplied by the length of the search range by the suggestions of [21]. Then, the new movement equation is modified as follows.

$$x_{id}(t+1) = x_{id}(t) + \beta(r_{ij}) \cdot (x_{jd}(t) - x_{id}(t)) + \alpha(t) \cdot s_d \cdot \left(rand - \frac{1}{2} \right) \tag{6}$$

where s_d is the length of the search interval of the d th dimension.

Based on MFA, we propose an enhanced version by employing a new adaptive parameter strategy to adjust the step factor α . In our approach AFA, Eq. (4) is modified as follows.

$$\alpha(t+1) = \left(1 - \frac{FEs}{MaxFEs} \right)^m \alpha(t) \tag{7}$$

where $m > 0$, FEs represents the number of fitness evaluations, $MaxFEs$ indicates the maximum value for the FEs , and t is the generation number. When $0 < m < 1$, a small

Algorithm 1: The Proposed AFA

```

1: Begin
2:   Randomly initialize all fireflies in the population;
3:    $FEs=N$ ;
4:   while  $FEs \leq MaxFEs$  do
5:     Update the parameter  $\alpha$  according to Eq. (7);
6:     for  $i=1$  to  $N$  do
7:       for  $j=1$  to  $i$  do
8:         if  $f(X_j) < f(X_i)$  then
9:           Move  $X_i$  towards  $X_j$  according to Eq. (6);
10:          Evaluate the new solution  $X_i$ ;
11:           $FEs++$ ;
12:         end if
13:       end for
14:     end for
15:   end while
16: End

```

Fig. 1. The framework of our approach AFA.

value is added to the weighed term of $\alpha(t)$. It avoids that $1 - FEs/MaxFEs = 0$. In our experiments, m is set to 0.5. In fact, Eq. (7) is a general version of our previous work [19].

The framework of AFA is given in Fig. 1. Compared to MFA, AFA only modifies the updating strategy of the step factor α . Therefore, AFA has the same complexity with MFA.

4 Experimental Study

4.1 Experimental Setup

In the experiment, thirteen benchmark functions are utilized for performance verification [22–26]. All test functions are minimization problems. Table 1 gives a brief description of these functions. More detailed descriptions of these functions can be found in [27–30].

Table 1. A brief descriptions of test functions.

Functions	Search interval	Global minimum
Sphere (f_1)	$[-100, 100]$	0
Schwefel 2.22 (f_2)	$[-10, 10]$	0
Schwefel 1.2 (f_3)	$[-100, 100]$	0
Schwefel 2.21 (f_4)	$[-100, 100]$	0
Rosenbrock (f_5)	$[-30, 30]$	0
Step (f_6)	$[-100, 100]$	0
Quartic with noise (f_7)	$[-1.28, 1.28]$	0
Schwefel 2.26 (f_8)	$[-500, 500]$	0
Rastrigin (f_9)	$[-5.12, 5.12]$	0
Ackley (f_{10})	$[-32, 32]$	0
Griewank (f_{11})	$[-600, 600]$	0
Penalized 1 (f_{12})	$[-50, 50]$	0
Penalized 2 (f_{13})	$[-50, 50]$	0

In the comparison, AFA is compared with four FAs. The related FAs are presented as follows.

- FA
- MFA [16]
- CFA [17]
- RaFA [9]
- Proposed AFA

The parameters N and $MaxFEs$ are equal to 20 and $5.0E + 05$, respectively. In the standard FA, the parameters α , β_0 , and γ are set to 0.2, 1.0, and $\gamma = 1/\Gamma^2$, respectively. For MFA, RaFA, and AFA, the initial α , β_{\min} , β_0 , and γ are set to 0.5, 0.2, 1.0, and 1.0,

respectively. The parameter m used in AFA is set to 0.5. Besides AFA, RaFA is also an improved version of MFA.

4.2 Results

Table 2 presents the results of AFA, FA, MFA, CFA, and RaFA on thirteen test functions, where “Mean” is the mean best fitness value over on 30 runs. From the results, AFA outperforms the standard FA on 11 functions, while AFA achieves worse solutions on two functions f_3 and f_7 . For function f_3 , AFA is trapped into local optima and can hardly obtain reasonable solutions. For function f_7 , the standard FA is a little better than AFA. Compared to MFA, the proposed adaptive parameter strategy helps AFA to achieve significant improvements, especially for f_1, f_2 , and $f_{10}-f_{13}$. CFA outperforms AFA on three functions f_3, f_5 and f_8 . For function f_6 , AFA, CFA, RaFA, and MFA find the same solution. RaFA is better than AFA on 6 functions, while AFA achieves more accurate solutions than RaFA on 6 functions.

Table 2. Results for different FAs.

Functions	FA mean	MFA mean	CFA mean	RaFA mean	AFA mean
f_1	5.14E-02	1.56E-05	3.27E-06	5.36E-184	5.36E-184
f_2	1.07E+00	1.85E-03	8.06E-04	8.76E-05	1.73E-07
f_3	1.26E-01	5.89E-05	1.24E-05	4.91E+02	8.57E+01
f_4	9.98E-02	1.73E-03	8.98E-04	2.43E+00	1.97E-04
f_5	3.41E+01	2.29E+01	2.06E+01	2.92E+01	2.69E+01
f_6	5.24E+03	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_7	7.55E-02	1.30E-01	9.03E-02	5.47E-02	7.87E-02
f_8	9.16E+03	4.94E+03	4.36E+03	5.03E+02	4.62E+03
f_9	4.95E+01	6.47E+01	5.27E+01	2.69E+01	4.08E+01
f_{10}	1.21E+01	4.23E-04	4.02E-04	3.61E-14	2.19E-14
f_{11}	2.13E-02	9.86E-03	7.91E-06	0.00E+00	7.78E-16
f_{12}	6.24E+00	5.04E-08	8.28E-09	4.50E-05	1.22E-25
f_{13}	5.11E+01	6.06E-07	1.69E-07	8.25E-32	8.36E-22

Figures 2 and 3 show the convergence graphs on some unimodal functions and multimodal functions, respectively. As seen, both RaFA and AFA converges faster than FA, MFA, and CFA.

To compare the optimization performance of the five FA variants on the whole test set, we calculate the mean rank values by the Friedman test. Table 3 gives the mean rank values of the five algorithms. The highest rank is marked in boldface. It is obvious that AFA obtains the highest rank. It demonstrates that AFA is the best one among AFA, FA, CFA, MFA, and RaFA.

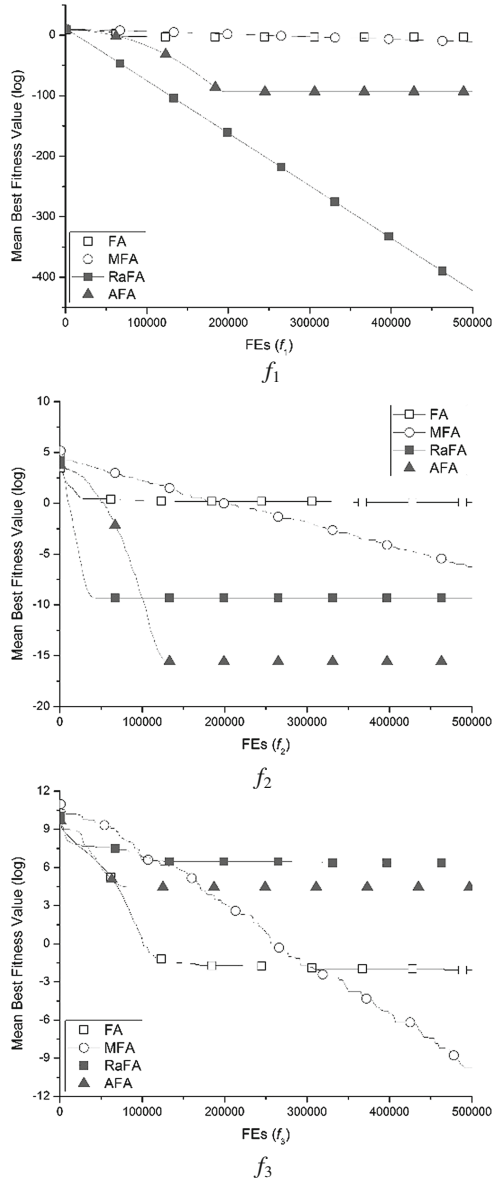


Fig. 2. The convergences processes of different FAs on some unimodal functions.

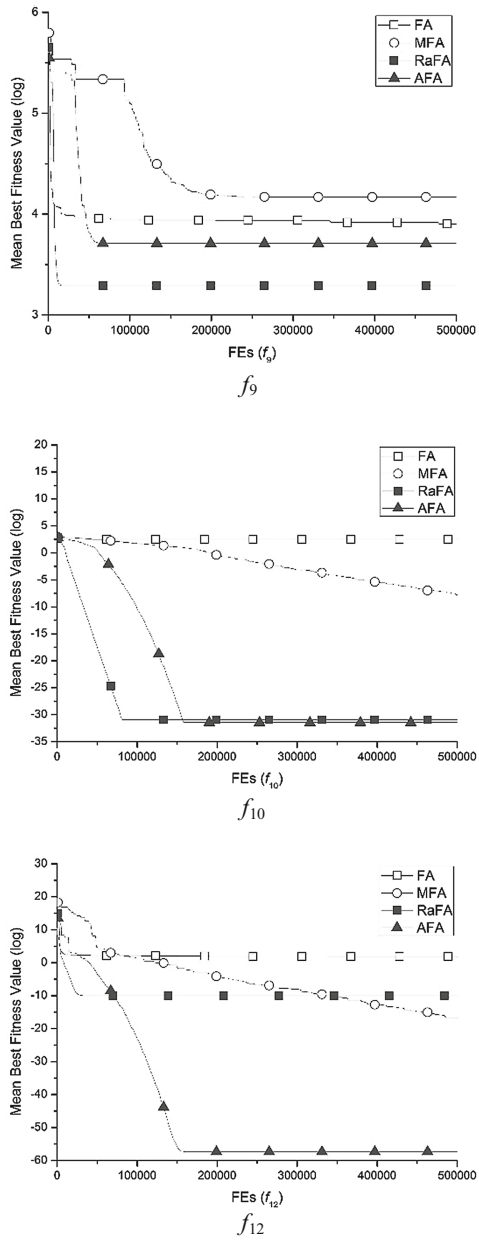


Fig. 3. The convergences processes of different FAs on some multimodal functions.

Table 3. Results for the Friedman test.

Algorithms	Rank
FA	4.38
MFA	3.58
CFA	2.58
RaFA	2.35
AFA	2.12

5 Conclusions

In this paper, an adaptive firefly algorithm (AFA) is proposed. It is an enhanced version of MFA. In AFA, a new parameter method is designed to adaptively change the step factor. In the experiment, thirteen test functions are used for performance verification. Simulation results show that AFA is superior to the standard FA, MFA, CFA, and RaFA. The adaptive parameter strategy is a general version of our previous work. In this paper, an empirical value is used. More investigations will be conducted in the future work.

Acknowledgement. This work is supported by the Science and Technology Plan Project of Jiangxi Provincial Education Department (No. GJJ161115), the National Natural Science Foundation of China (No. 61663028), the Distinguished Young Talents Plan of Jaingxi Province (No. 20171BCB23075), the Natural Science Foundation of Jiangxi Province (No. 20171BAB 202035), and the Open Research Fund of Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing (No. 2016WICSIP015).

References

1. Sun, H., Wang, K., Zhao, J., Yu, X.: Artificial bee colony algorithm with improved special centre. *Int. J. Comput. Sci. Math.* **7**(6), 548–553 (2016)
2. Yun, G.: A new multi-population-based artificial bee colony for numerical optimization. *Int. J. Comput. Sci. Math.* **7**(6), 509–515 (2016)
3. Lv, L., Wu, L.Y., Zhao, J., Wang, H., Wu, R.X., Fan, T.H., Hu, M., Xie, Z.F.: Improved multi-strategy artificial bee colony algorithm. *Int. J. Comput. Sci. Math.* **7**(5), 467–475 (2016)
4. Lu, Y., Li, R.X., Li, S.M.: Artificial bee colony with bidirectional search. *Int. J. Comput. Sci. Math.* **7**(6), 586–593 (2016)
5. Cai, X., Gao, X.Z., Xue, Y.: Improved bat algorithm with optimal forage strategy and random disturbance strategy. *Int. J. Bio-Inspired Comput.* **8**(4), 205–214 (2016)
6. Xue, F., Cai, Y., Cao, Y., Cui, Z., Li, F.: Optimal parameter settings for bat algorithm. *Int. J. Bio-Inspired Comput.* **7**(2), 125–128 (2015)
7. Cai, X., Wang, L., Kang, Q., Wu, Q.: Bat algorithm with Gaussian walk. *Int. J. Bio-Inspired Comput.* **6**(3), 166–174 (2014)
8. Yang, X.S.: *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, Beckington (2008)
9. Wang, H., Wang, W.J., Sun, H., Rahnamayan, S.: Firefly algorithm with random attraction. *Int. J. Bio-Inspired Comput.* **8**(1), 33–41 (2016)

10. Wang, H., Wang, W.J., Zhou, X.Y., Sun, H., Zhao, J., Yu, X., Cui, Z.: Firefly algorithm with neighborhood attraction. *Inf. Sci.* **382–383**, 374–387 (2017)
11. Cui, Z., Sun, B., Wang, G., Xue, Y.: A novel oriented cuckoo search algorithm to improve DV-Hop performance for cyber-physical systems. *J. Parallel Distrib. Comput.* **103**, 42–52 (2017)
12. Wang, G.G., Gandomi, A.H., Yang, X.S., Alavi, A.H.: A new hybrid method based on krill herd and cuckoo search for global optimization tasks. *Int. J. Bio-Inspired Comput.* **8**(5), 286–299 (2016)
13. Zhang, Y.W., Wu, J.T., Guo, X., Li, G.N.: Optimising web service composition based on differential fruit fly optimisation algorithm. *Int. J. Comput. Sci. Math.* **7**(1), 87–101 (2016)
14. Cui, Z., Fan, S., Zeng, J., Shi, Z.Z.: APOA with parabola model for directing orbits of chaotic systems. *Int. J. Bio-Inspired Comput.* **5**(1), 67–72 (2013)
15. Cui, Z., Fan, S., Zeng, J., Shi, Z.Z.: Artificial plant optimisation algorithm with three-period photosynthesis. *Int. J. Bio-Inspired Comput.* **5**(2), 133–139 (2013)
16. Fister Jr., I., Yang, X.S., Fister, I., Brest, J.: Memetic firefly algorithm for combinatorial optimization. arXiv preprint [arXiv:1204.5165](https://arxiv.org/abs/1204.5165) (2012)
17. Gandomi, A.H., Yang, X.S., Talatahari, S., Alavi, A.H.: Firefly algorithm with chaos. *Commun. Nonlinear Sci. Numer. Simul.* **18**(1), 89–98 (2013)
18. Roy, A.G., Rakshit, P., Konar, A., Bhattacharya, S., Kim, E., Nagar, A.K.: Adaptive firefly algorithm for nonholonomic motion planning of car-like system. In: *IEEE Congress on Evolutionary Computation (CEC 2013)*, pp. 2162–2169. IEEE, Cancun (2013)
19. Wang, H., Zhou, X.Y., Sun, H., Yu, X., Zhao, J., Zhang, H., Cui, L.Z.: Firefly algorithm with adaptive control parameters. *Soft. Comput.* 1–12 (2016). doi:[10.1007/s00500-016-2104-3](https://doi.org/10.1007/s00500-016-2104-3)
20. Wang, H., Cui, Z.H., Sun, H., Rahnamayan, S., Yang, X.S.: Randomly attracted firefly algorithm with neighborhood search and dynamic parameter adjustment mechanism. *Soft. Comput.* 1–15 (2016). doi:[10.1007/s00500-016-2116-z](https://doi.org/10.1007/s00500-016-2116-z)
21. Yang, X.S.: *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley, Hoboken (2010)
22. Wang, H., Wu, Z.J., Rahnamayan, S., Liu, Y., Ventresca, M.: Enhancing particle swarm optimization using generalized opposition-based learning. *Inf. Sci.* **181**(20), 4699–4714 (2011)
23. Wang, H., Rahnamayan, S., Sun, H., Omran, M.G.H.: Gaussian bare-bones differential evolution. *IEEE Trans. Cybern.* **43**(2), 634–647 (2013)
24. Guo, Z.L., Wang, S.W., Yue, X.Z., Yin, B.: Enhanced social emotional optimisation algorithm with elite multi-parent crossover. *Int. J. Comput. Sci. Math.* **7**(6), 568–574 (2016)
25. Wang, H., Liu, Y., Li, C.H., Zeng, S.Y.: A hybrid particle swarm algorithm with Cauchy mutation. In: *IEEE Swarm Intelligence Symposium (SIS 2007)*, pp. 356–360. IEEE, Honolulu (2007)
26. Yu, G.: An improved firefly algorithm based on probabilistic attraction. *Int. J. Comput. Sci. Math.* **7**(6), 530–536 (2016)
27. Wang, H., Sun, H., Li, C.H., Rahnamayan, S., Pan, J.S.: Diversity enhanced particle swarm optimization with neighborhood search. *Inf. Sci.* **223**, 119–135 (2013)
28. Wang, H., Wu, Z.J., Rahnamayan, S., Sun, H., Liu, Y., Pan, J.S.: Multi-strategy ensemble artificial bee colony algorithm. *Inf. Sci.* **279**, 587–603 (2014)
29. Zhou, X.Y., Wu, Z.J., Wang, H., Rahnamayan, S.: Gaussian bare-bones artificial bee colony algorithm. *Soft. Comput.* **20**(3), 907–924 (2016)
30. Zhou, X.Y., Wang, H., Wang, M.W., Wan, J.Y.: Enhancing the modified artificial bee colony algorithm with neighborhood search. *Soft. Comput.* **21**(10), 2733–2743 (2017)