

# Randomly attracted firefly algorithm with neighborhood search and dynamic parameter adjustment mechanism

Hui Wang<sup>1,2</sup>  · Zhihua Cui<sup>3</sup> · Hui Sun<sup>1,2</sup> · Shahryar Rahnamayan<sup>4</sup> · Xin-She Yang<sup>5</sup>

© Springer-Verlag Berlin Heidelberg 2016

**Abstract** Firefly algorithm (FA) is a new swarm intelligence optimization algorithm, which has shown an effective performance on many optimization problems. However, it may suffer from premature convergence when solving complex optimization problems. In this paper, we propose a new FA variant, called NSRaFA, which employs a random attraction model and three neighborhood search strategies to obtain a trade-off between exploration and exploitation abilities.

Moreover, a dynamic parameter adjustment mechanism is used to automatically adjust the control parameters. Experiments are conducted on a set of well-known benchmark functions. Results show that our approach achieves much better solutions than the standard FA and five other recently proposed FA variants.

**Keywords** Firefly algorithm (FA) · Random attraction · Neighborhood search · Dynamic parameter adjustment mechanism · Global optimization

Communicated by V. Loia.

✉ Hui Wang  
huiwang@whu.edu.cn

Zhihua Cui  
zhihuacui@gmail.com

Hui Sun  
sun\_hui2006@163.com

Shahryar Rahnamayan  
shahryar.rahnamayan@uoit.ca

Xin-She Yang  
x.yang@mdx.ac.uk

<sup>1</sup> School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China

<sup>2</sup> Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing, Nanchang 330099, China

<sup>3</sup> School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China

<sup>4</sup> Department of Electrical, Computer, and Software Engineering, University of Ontario Institute of Technology (UOIT), 2000 Simcoe Street North, Oshawa, ON L1H 7K4, Canada

<sup>5</sup> School of Science and Technology, Middlesex University, Hendon Campus, London NW44BT, UK

## 1 Introduction

Nowadays, most new algorithms are bio-inspired, because they have been developed by drawing inspiration from nature. Swarm intelligence algorithms are a special class of bio-inspired algorithms, which have been developed by the collective behaviors of nature (Fister et al. 2013). In the past decades, swarm intelligence algorithms have become popular. Good examples are ant colony optimization (ACO) (Dorigo et al. 1996), particle swarm optimization (PSO) (Kennedy and Eberhart 1995), bat algorithm (Yang 2010), cuckoo search (CS) (Yang and Deb 2009), and firefly algorithm (FA) (Fister et al. 2013; Yang 2008).

Firefly algorithm (FA) is a recently proposed swarm intelligence algorithm inspired by the idealized behavior of the flashing characteristics of fireflies. Previous studies show that FA outperforms genetic algorithm (GA) and PSO on some benchmark functions (Yang 2010). Because of FA's simple concept, easy implementation, and its effectiveness, it has been successfully applied to various optimization fields, such as complex networks (Amiri et al. 2013), unit commitment (Chandrasekaran et al. 2013), energy conservation (Coelho and Mariani 2013), structural optimiza-

tion (Gandomi et al. 2013; Miguel et al. 2013), image compression (Hornig 2012), stock forecasting (Kazem et al. 2013), economic dispatch (Liang et al. 2015; Yang et al. 2012), and so on.

In FA, the fitness function for a given problem is associated with the light intensity. The brighter the firefly, the better is the firefly. That means a brighter firefly has a better fitness value. The search process of FA depends on the attraction among fireflies. The standard FA defines a full attraction model, in which each firefly is attracted by all other brighter fireflies in the swarm. If fireflies are attracted by too many other fireflies, all fireflies will become similar. As a result, FA shows slow convergence rate during the search process. Moreover, FA with a fully attracted model has high computational time complexity. For a population with  $N$  candidate solutions, most nature-inspired algorithms only conduct  $N$  operations in each generation, while FA executes  $\frac{N(N-1)}{2}$  operations.

In the standard FA and its most variants, a firefly moves toward other brighter fireflies by attraction. However, if a firefly is brighter than another one, the brighter one will not be conducted any search. It seems that the FA mainly updates the search on worse fireflies (candidate solutions). That may slow down the convergence speed.

To address the above two issues, we propose a new FA called the randomly attracted FA with neighborhood search (NSRaFA). The new approach employs three strategies: (1) a dynamic parameter adjustment mechanism; (2) a random attraction model; and (3) three neighborhood search operators. The first strategy aims to automatically adjust the control parameters  $\alpha$  and  $\beta$ , and avoid manual parameter settings. The second one is helpful to accelerate the convergence speed and reduce the computational time complexity. The last one defines a new neighborhood search operation for brighter fireflies (better candidate solutions). When a firefly is brighter than another one, the neighborhood search is conducted on the brighter one to provide more chances of finding better solutions.

The rest of the paper is organized as follows. Section 2 presents the standard FA and its brief review. Section 3 describes our proposed approach. Section 4 presents the experimental results and discussions. Finally, the work is concluded in Sect. 5.

## 2 Related work

### 2.1 Firefly algorithm

The original FA was developed by Yang (2008). It is inspired by the social behavior of fireflies. Most fireflies produce short and rhythmic flashes to attract mating partners and poten-

---

### Algorithm 1: The Standard FA

---

```

1 Randomly generate  $N$  fireflies (solutions) as an initial population
   $\{X_i | i = 1, 2, \dots, N\}$ ;
2 Calculate the fitness value of each firefly;
3  $FES = N$ ;
4 while  $FES \leq MAX\_FES$  do
5   for  $i = 1$  to  $N$  do
6     for  $j = 1$  to  $N$  do
7       if  $f(X_j) < f(X_i)$  then
8         Move firefly  $X_i$  towards  $X_j$  according to Eq. (3);
9         Calculate the fitness value of the new solution;
10         $FES++$ ;
11      end
12    end
13  end
14 end
```

---

tial prey. To describe the FA, the following three rules are used (Yang 2010):

- All fireflies are unisex. So, one firefly will be attracted to other fireflies regardless of their sex.
- Attractiveness is proportional to their brightness. Thus, for any two flashing fireflies, the less bright one will move toward the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly.
- The brightness of a firefly is affected or determined by the landscape of the objective function. For a minimization problem, the brightness can be reciprocal of objective function. It means that a brighter firefly has a smaller objective function value.

In FA, the attractiveness of a firefly is determined by its light intensity. According to Yang (2010), the attractiveness can be calculated as follows:

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

where  $\beta_0$  is the attractiveness at  $r = 0$ , and  $r$  is the distance between two fireflies. The parameter  $\gamma$  is the light absorption coefficient, which is usually set to 1.

For two fireflies  $X_i$  and  $X_j$ , their distance  $r_{ij}$  can be defined by

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

where  $D$  is the problem dimension.

The movement of a firefly  $X_i$ , which is attracted to another brighter firefly  $X_j$ , is determined by

$$x_{id}(t+1) = x_{id}(t) + \beta_0 e^{-\gamma r_{ij}^2} (x_{jd}(t) - x_{id}(t)) + \alpha \epsilon, \quad (3)$$

where  $x_{id}$  and  $x_{jd}$  are the  $d$ th dimension value of firefly  $X_i$  and  $X_j$ , respectively. In addition,  $\epsilon = (\text{rand} - 1/2)$  and  $\text{rand}$  is a random variable that is uniformly distributed in the range  $[0, 1]$ ,  $\alpha \in [0, 1]$  is the step parameter, and  $t = 1, 2, \dots$  indicates the iteration number.

The main steps of the standard FA are described in Algorithm 1, where  $N$  is the swarm size,  $f(\cdot)$  is the fitness evaluation function, FEs is the number of fitness evaluations, and MAX\_FEs is the maximum number of fitness evaluations. In this paper, we only consider minimization problems. Thus,  $f(X_j) < f(X_i)$  means that firefly  $X_j$  is better than firefly  $X_i$  in terms of fitness value.

## 2.2 Some FA variants

Since FA was developed, it has become a popular optimizer and has widely been applied in practical or benchmark problems (Duang and Luo 2015; Fister et al. 2013). In the past several years, some new FA variants have been proposed. A brief overview of these variants is presented as follows.

### 2.2.1 Memetic FA (MFA)

To improve the performance of FA, Fister et al. (2012) designed a memetic FA (MFA) to solve combinatorial optimization problems. In MFA, a new movement equation is proposed as follows.

$$x_{id}(t+1) = x_{id}(t) + \beta(x_{jd}(t) - x_{id}(t)) + \alpha(t)s_d\epsilon, \quad (4)$$

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

$$\alpha(t+1) = \left(\frac{1}{9000}\right)^{\frac{1}{t}} \alpha(t), \quad (6)$$

$$s_d = x_d^{\max} - x_d^{\min}, \quad (7)$$

where  $\beta_{\min}$  is a minimum value of  $\beta$ ,  $s_d$  is the scale of each design variable, and  $[x_d^{\min}, x_d^{\max}]$  is the boundary constraint for the  $d$ th variable. Although MFA is designed for combinatorial optimization problems, our experiments show that it works well on continuous problems.

### 2.2.2 FA with chaos (CFA)

In Gandomi et al. (2013), Gandomi et al. introduced chaos into FA to increase its global search ability for robust global optimization. The CFA designed three groups of experiments: (1) the parameter  $\gamma$  is replaced with the different chaotic maps; (2) the parameter  $\beta$  is generated by different chaotic maps; (3) both  $\gamma$  and  $\beta$  are updated by different chaotic maps.

There are 12 different chaotic maps including Chebyshev, Circle, Gauss/Mouse, Intermittency, Iterative, Liebovitch,

Logistic, Piecewise, Sine, Singer, Sinusoidal, and Tent. Simulation results show that tuning of the  $\beta$  is more effective than tuning  $\gamma$ . The Gauss map may be the best choice for updating the parameter  $\gamma$ . There is no improvement when both  $\gamma$  and  $\beta$  are replaced with the chaotic maps. In Fister et al. (2015), a comprehensive overview of FA is presented with chaotic maps.

### 2.2.3 Wise step strategy FA (WSSFA)

In Yu et al. (2014), a wise step strategy for FA is proposed to set the step for each firefly. In WSSFA, each firefly in the swarm has an independent step parameter  $\alpha_i$ , which considers the absolute distance of firefly's previous best and the global best positions. The  $\alpha_i$  for each firefly is defined by

$$\alpha_i(t+1) = \alpha_i(t) - (\alpha_i(t) - \alpha_{\min})e^{-\frac{|\text{gbest} - \text{pbest}_i| r}{G_{\max}}}, \quad (8)$$

where  $\alpha_i$  is the step parameter for the  $i$ th firefly in the swarm,  $\alpha_{\min}$  is the minimum step in the range  $[0, 1]$ ,  $\text{pbest}_i$  is the previous best firefly of  $X_i$ ,  $\text{gbest}$  is the global best firefly found so far and  $G_{\max}$  is the maximum number of generations.

### 2.2.4 Variable step size FA (VSSFA)

In the standard FA, the step parameter  $\alpha$  is static. It may not be helpful to the search. Generally, a large  $\alpha$  is suitable for exploring new search space, while a small  $\alpha$  is good for exploitation. In Yu et al. (2015), a new FA called variable step size firefly algorithm (VSSFA) is proposed. In VSSFA, the parameter  $\alpha$  employs a dynamic adjusting method in the following form:

$$\alpha(t) = \frac{0.4}{1 + e^{\frac{t - G_{\max}}{200}}}. \quad (9)$$

The reported results show that VSSFA and WSSFA achieved better solutions than the standard FA on some test functions. However, most of these functions are only two dimensional. Our experiments show that the performance of VSSFA and WSSFA may be seriously affected by its problem dimension.

### 2.2.5 Other approaches

To enhance the performance of FA, Farahani et al. (2011) proposed a Gaussian distributed FA (GDFA), in which the Gaussian distribution is used to move all fireflies to the global best in each iteration. Computational results on five benchmark functions show that GDFA outperforms particle swarm optimization (PSO) Kennedy and Eberhart (1995) and the standard FA. Tilahun and Ong (2012) modified the random movement of the brighter firefly by generating random

directions to determine the best direction. In addition, the assignment of attractiveness was modified in such a way that the effect of the objective function was magnified. Results indicated that the modified FA achieved better solutions than the standard FA.

Quaternion is a concept in mathematics, which can extend complex numbers (Chen et al. 2015). In Fister et al. (2013), quaternion is used for the representation of individuals in FA so as to enhance the performance of the firefly algorithm and to avoid possible stagnation. Wang et al. (2014) proposed a modified FA based on light intensity differences. The new algorithm considers the variation trend of light intensity differences. The light intensity differences vary with the movements of fireflies. Thus, the parameter settings are dynamically adjusted for different problems.

### 2.2.6 Applications of FA

Senthilnath et al. (2011) presented a performance study of FA on clustering. Simulation results show that FA can be efficiently used for clustering. Yang et al. (2012) applied FA to solve non-convex economic dispatch (ED) problems with value loading effect. To verify the efficiency and applicability of the FA, four ED problems are utilized in the experiments. Results show that FA is considered to be a promising alternative algorithm for the ED problems in practical power systems. The vector quantization (VQ) was a powerful technique in the applications of digital image compression. Significant energy savings can be achieved by optimizing chiller operation and design in heating, ventilation, and cooling (HVAC) systems. Coelho and Mariani (2013) designed an improved FA (IFA) based on Gaussian distribution to the optimal chiller loading design. Results show that the proposed IFA outperforms several optimization methods of the literature in terms of minimum energy consumption. Kazem et al. (2013) designed a hybrid FA to predict the price of the stock. The new approach employs two strategies: chaotic FA and support vector regression (SVR) (Gu et al. 2015a, b). Results show that the new approach is better than artificial neural networks (ANNs), GA-based SVR (GA-SVR), and, adaptive neuro-fuzzy inference systems.

Software testing is an important, but challenging task in the software life cycle. How to optimize the software testing process is a difficult task to solve, and the generation of the independent test paths remains unsatisfactory. Srivatsava et al. (2013) used a modified FA to generate optimal test paths. Results show that the test paths generated are critical and optimal.

In Long et al. (2015), a heart disease diagnosis system was proposed, in which rough sets-based attribute reduction using chaotic FA is investigated to find the optimal reduction. Sahu et al. (2015) presented a novel hybrid FA and pattern search (hFA-PS) for automatic generation control (AGC)

of multi area power systems. Results show that hFA-PS is able to handle nonlinearity and physical constraints in the system model. In Mahapatra et al. (2014), the hFA-PS was successfully applied to design a static synchronous series compensator (SSSC)-based power oscillation damping controller. In Kougianos and Mohanty (2015), FA is applied to optimize the cost of leakage delay product (LDP) under various resource constraints.

Studying the evolutionary community structure in complex networks is crucial for uncovering the links between structures and functions of a given community. Most community detection algorithms employ single optimization criteria. Amiri et al. (2013) considers community detection process as a multi-objective optimization problem (MOP) for investigating the community structures in complex networks. To tackle this problem, a new multi-objective enhanced FA is proposed. Experimental results show that the proposed approach provides useful paradigm for discovering overlapping community structures robustly. Recommender systems (RS) is a new technique, which can provide uses with required information (Ma et al. 2015). In Shomalnasab et al. (2014), FA was used for the optimal similarity in collaborative filtering. Results show that FA can improve the accuracy of recommendation on some real data sets.

Image segmentation is an important operation for image processing (Li et al. 2015; Zheng et al. 2015). FA was used to optimize Otsu's method. Simulation results show the efficiency of FA (Hassanzadeh et al. 2011). Classification is a hot research direction in the area of machine learning (Wen et al. 2015). In Saraç and Özel (2013), FA was applied to select a subset of features for Web page classification. Experiments on some WebKB and conference data sets show the effectiveness of FA.

With the rapid development of a smart society, different sensor networks have been designed (Gopinadh and Singh 2015; Xie and Wang 2014). Among these networks, underwater sensor networks (UWSNs) are a new technique (Shen et al. 2015). Xu and Liu (2013) proposed a variant of the firefly algorithm, called multi-population FA (MPFA) for correlated data routing in UWSNs. Results showed that MPFA achieved better performance than some existing protocols. Cloud computing is a very hot research topic in information technique (Fu et al. 2015; Ren et al. 2015; Xia et al. 2015b). Florence and Shanthi (2014) used FA to maximize the usage rate of resource in cloud servers.

Recently, some discrete and binary FA variants have been proposed. Poursalehi et al. (2013) developed a discrete FA to implement the loading pattern optimization of nuclear reactor core. Sayadi et al. (2013) proposed another discrete FA, in which a firefly's position is defined in terms of changes of probabilities that will be in one state or the other. Chandrasekaran et al. (2013) presented a new biologically-inspired binary real coded FA to solve the unit commitment

problem (UCP) by considering the system and generating unit constraints.

Furthermore, Marichelvam et al. (2014) designed a discrete FA for the multi-objective hybrid flowshop scheduling problems. Makespan and mean flow time were the considered objective functions. Results showed that the proposed approach outperformed many other algorithms in the literature. Rahmani and MirHassani (2014) proposed a hybrid evolutionary firefly-genetic algorithm for capacitated facility location problem. Results on some randomly generated problems consisting of 2000 locations and 2000 customers were reported. Steganalysis is an important technique in information security, which can detect the hidden messages (Xia et al. 2014, a). In Chhikara and Singh (2015), a discrete FA was used to improve the performance of blind image steganalysis.

### 3 Proposed approach

In this section, we present a new FA variant, called the randomly attracted FA with neighborhood search (NSRaFA). The NSRaFA employs three strategies: a dynamic parameter adjustment mechanism, a random attraction model, and three neighborhood search operators.

#### 3.1 Dynamic parameter adjustment mechanism

For the movement attraction, there are some different updating equations. Fister et al. (2012) proposed a modified movement equation (see Eq. 4), in which the  $\alpha$  is multiplied by the scale of the designed variables. Gandomi et al. (2013) suggested that the parameter  $\alpha$  should ideally be related to the actual scale of the designed variables. Our experimental results show that the Eq. (4) is suitable for solving continuous optimization problems. Therefore, we employ the Eq. 4 as the movement equation in our approach.

However, both parameters  $\alpha$  and  $\beta$  are very important for the performance of the FAs. Different parameter settings may seriously affect the accuracy of the final solutions. The search of FA is determined by the attractions among fireflies in the swarm. As the iterations continue, fireflies gradually approach the converged states because of the attractions. When FA is finally convergent,  $X_i(t+1) = X_i(t)$  and  $X_i(t) = X_j(t)$  are satisfied as  $t \rightarrow \infty$ . According to Eq. (3), we can get

$$\begin{aligned} X_i(t+1) - X_i(t) &= 0 \\ \Rightarrow \beta_0 e^{-\gamma r_{ij}^2} (X_j(t) - X_i(t)) + \alpha \left(\text{rand} - \frac{1}{2}\right) &= 0 \\ \Rightarrow 0 + \alpha \left(\text{rand} - \frac{1}{2}\right) &= 0 \\ \Rightarrow \alpha &= 0. \end{aligned} \quad (10)$$

The above analysis demonstrates that  $\alpha$  should be equal to 0, when FA converges to an optimal solution. To satisfy Eq. (10), a dynamic parameter setting method is proposed as follows.

$$\alpha(t+1) = 0.99\alpha(t), \quad (11)$$

where the initial  $\alpha(0)$  is set to 0.5 according to empirical studies. This idea of cooling down  $\alpha$  is based on the original idea in Yang (2008).

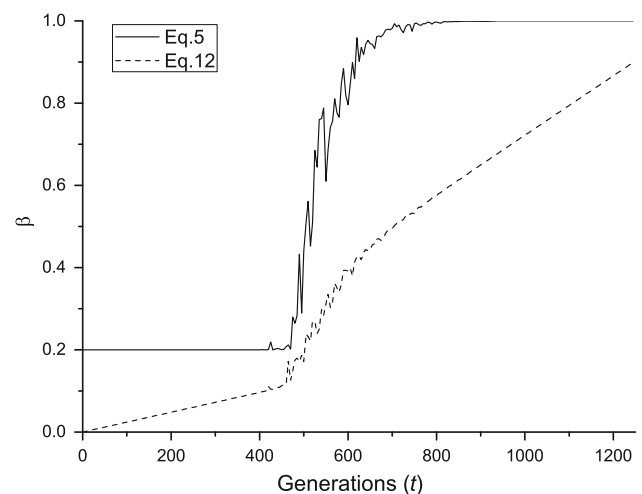
However, there is a potential a problem for Eq. (5) (please see Fig. 1). At the beginning stage, the attractiveness  $\beta$  is fixed to  $\beta_{\min} = 0.2$ , because  $r$  is very large and  $e^{-\gamma r_{ij}^2}$  tends to be 0 ( $\gamma = 1.0$ ). With increase of generations,  $\beta$  quickly increases to  $\beta_0 = 1.0$ . As shown, the changes of  $\beta$  are too fast between  $t=500$  and  $t = 700$ . That may not be beneficial for the search. To slow down the increase of  $\beta$ , we propose a modified strategy based on Eq. (5). The new  $\beta$  is defined by

$$\beta = \left( \beta_{\min} + (\beta_{\max} - \beta_{\min}) e^{-\gamma r_{ij}^2} \right) \frac{t}{G_{\max}}, \quad (12)$$

where  $\beta_{\min}$  and  $\beta_{\max}$  are the minimum and maximum values of  $\beta$ , respectively. In fact, it is difficult to select the best values for  $\beta_{\min}$  and  $\beta_{\max}$ . Based on our empirical studies, they are set to 0.3 and 0.9, respectively. As shown in Fig. 1, the new strategy (Eq. 12) can efficiently slow down the increasing speed of  $\beta$ .

#### 3.2 Random attraction model

In the standard FA, each firefly is attracted by all other brighter fireflies in the swarm. This attraction mechanism is referred to as the full attraction model. Assume that there



**Fig. 1** The changes of the attractiveness  $\beta$  under different parameter strategies (Eqs. 5 and 12)

are  $N$  fireflies in the swarm. To analyze how many attractions are there among fireflies in each generation, we rank all fireflies in the swarm. Then, the first firefly is the best one, and the  $N$ th firefly is the worst one. It means that the first and the  $N$ th fireflies are attracted by 0 and  $N - 1$  fireflies, respectively. Therefore, the total number of attractions ( $T_{\text{attraction}}$ ) for all fireflies is calculated as follows.

$$T_{\text{attraction}} = 0 + 1 + \dots + N - 1 = \frac{N(N - 1)}{2}. \quad (13)$$

It can be concluded that the average number of attractions for each firefly is  $\frac{T_{\text{attraction}}}{N} = \frac{N-1}{2}$ .

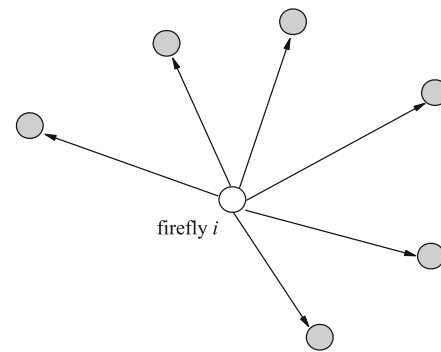
Although the full attraction model can provide many chances of searching candidate solutions (fireflies), it may result in oscillations under certain conditions. Finally, the standard FA may show a slow convergence speed and hardly achieves promising solutions under certain conditions such as a fixed  $\alpha$ . Furthermore, too many attractions dramatically increase the computational time complexity. For a given problem, we assume that  $O(f)$  is the computational time complexity of its fitness evaluation function  $f(\cdot)$ . The computational time complexity of the standard FA is  $O(G_{\text{max}} \cdot N^2 \cdot f)$ . Compared with other swarm intelligence algorithms, such as PSO, its time complexity is only  $O(G_{\text{max}} \cdot N \cdot f)$ . Therefore, the standard FA has a higher complexity. However, as  $N$  is usually not large, this may not be the main issue. The way of attraction can be more important.

In our previous work (Wang et al. 2016), we proposed a random attraction model to reduce the computational time complexity of FA. In the fully attracted model, each firefly  $i$  is compared to the rest of the  $N - 1$  fireflies, and the firefly  $i$  may conduct the attraction movements many times. In our random attraction model, each firefly  $i$  is only compared to another randomly selected firefly  $j$ , and the firefly  $i$  carries out the attraction movement once at most. Consequently, the random attraction model has much less attraction than the full attraction model. Let us consider an extreme case that each firefly (except the brightest one) moves to another randomly selected firefly. Thus, the maximum number of attractions ( $T'_{\text{attraction}}$ ) is

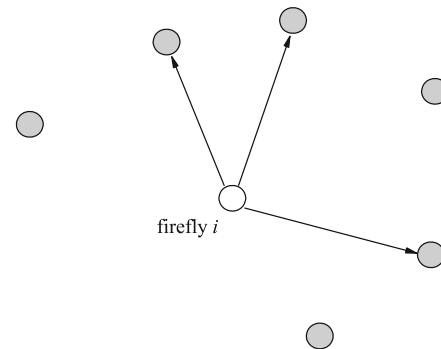
$$T'_{\text{attraction}} = 0 + 1 + \dots + 1 = N - 1. \quad (14)$$

It is obvious that  $T_{\text{attraction}}$  is much larger than  $T'_{\text{attraction}}$ .

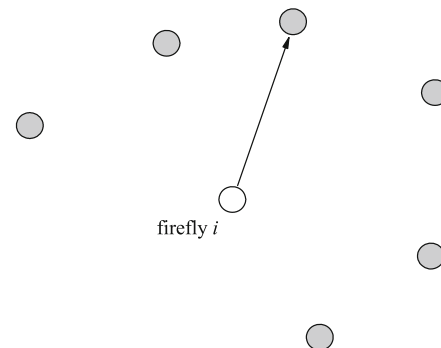
To clearly compare the full attraction model and random attraction model, Fig. 2 shows three examples for the attractions of a firefly in the swarm, where  $N = 7$ . As seen, Fig. 2a gives an extreme case for the full attraction model, in which a firefly is attracted by all other six fireflies. In Fig. 2b, each firefly is attracted by three other fireflies on average. Figure 2c presents an example for the random attraction model, where each firefly is attracted by one other firefly at most.



(a) An extreme case for the full attraction model.



(b) Average attractions for the full attraction model.



(c) An example for random attraction model.

Fig. 2 Full attraction model versus random attraction model for  $N = 7$

### 3.3 Neighborhood search

As mentioned before, a firefly moves to other brighter fireflies by attraction. If the current firefly is brighter than another one, the current one will not be conducted any major search, though it can be perturbed by a local random walk. It means that the FA mainly carries out the search operations on replacing some worse fireflies (candidate solutions). This may slow down the convergence speed. If we also carry out some search operations on better fireflies, it may improve the exploitation ability and accelerate the convergence speed.

**Algorithm 2:** The proposed NSRaFA

```

1 Randomly generate  $N$  fireflies (solutions) as an initial population
   $\{X_i | i = 1, 2, \dots, N\}$ ;
2 Calculate the fitness value of each firefly;
3  $FES = N$ ;
4 Initialize  $pbest$  and  $gbest$ ;
5 while  $FES \leq MAX\_FES$  do
6   for  $i = 1$  to  $N$  do
7     /*Random attraction model */
8     Randomly select a firefly  $X_j$  from the swarm, and  $i \neq j$ ;
9     if  $f(X_j) < f(X_i)$  then
10      /*Dynamic parameter adjustment mechanism */
11      Update the parameters  $\alpha$  and  $\beta$  according to Eqs. (11) and (12),
12      respectively;
13      Move firefly  $X_i$  towards  $X_j$  according to Eq. (4);
14      Calculate the fitness value of the new solution;
15       $FES++$ ;
16    end
17  else
18    /*Neighborhood search */
19    Generate three trial solutions  $X_i^1$ ,  $X_i^2$ , and  $X_i^3$  according to
20    Eqs. (15), (16), and (17), respectively;
21    Calculate the fitness values of  $X_i^1$ ,  $X_i^2$ , and  $X_i^3$ ;
22     $FES = FES + 3$ ;
23    Select the best solution among  $X_i$ ,  $X_i^1$ ,  $X_i^2$ , and  $X_i^3$  as the new  $X_i$ ;
24  end
25  Update  $pbest_i$  and  $gbest$ ;
26 end

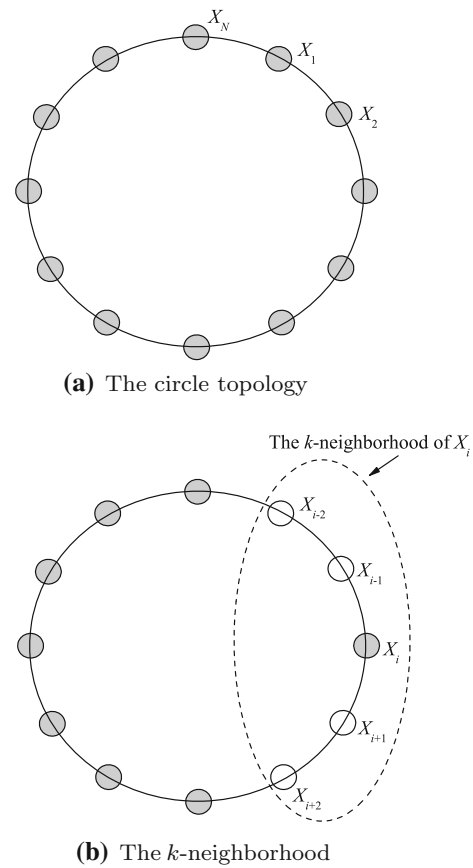
```

To address the above issue, we design three neighborhood search strategies for the standard FA. When the current firefly is brighter than another one, the brighter one will carry out the neighborhood search to provide more chances of finding more accurate candidate solutions. The neighborhood search consists of three strategies: one local and two global neighborhood search operators.

Assume that all  $N$  fireflies in the swarm are organized in a circle topology according to their indices. For example,  $X_N$  and  $X_2$  are two immediate neighbors of  $X_1$  (Das et al. 2009). Figure 3a presents an example for the circle topology, where the population has 12 fireflies. Based on the circle topology, Das et al. (2009) proposed a concept of  $k$ -neighborhood. For each firefly  $X_i$ , its  $k$ -neighborhood consisting of  $2k + 1$  fireflies  $X_{i-k}, \dots, X_i, \dots, X_{i+k}$ , where  $k$  is an integer  $0 \leq k \leq \frac{N-1}{2}$ . Figure 3b shows an example for the two-neighborhood, where there are five fireflies in the neighborhood of  $X_i$ .

The concept for  $k$ -neighborhood has been successfully used to improve the performance of differential evolution (DE) and PSO (Das et al. 2009; Wang et al. 2013, 2011). In Das et al. (2009), a neighborhood mutation operator is proposed to balance the exploration and exploitation of DE. Wang et al. (2013, 2011) presented two global and one local search operators to enhance the performance of PSO. In this paper, the neighborhood search strategies used in PSO (Wang et al. 2013, 2011) are employed to conduct the search of those better fireflies.

For each firefly, its neighborhood may have better candidate solutions. To improve the ability of exploitation, a local



**Fig. 3** The circle topology and  $k$ -neighborhood, where  $N = 12$  and  $k = 2$

neighborhood search operator is proposed as follows.

$$X_i^1 = r_1 \cdot X_i + r_2 \cdot pbest_i + r_3 \cdot (X_{i1} - X_{i2}), \tag{15}$$

where  $X_{i1}$  and  $X_{i2}$  are two fireflies randomly selected from the  $k$ -neighborhood radius of  $X_i$  ( $i1 \neq i2 \neq i$ ),  $pbest_i$  is the previous best of the  $i$ th firefly,  $r_1, r_2$  and  $r_3$  are three uniform random numbers with the range  $(0, 1)$ , and  $r_1 + r_2 + r_3 = 1$ .

Similar to the local neighborhood search operator, a global neighborhood search operator is proposed to enhance the ability of exploration.

$$X_i^2 = r_4 \cdot X_i + r_5 \cdot gbest + r_6 \cdot (X_{i3} - X_{i4}), \tag{16}$$

where  $X_{i3}$  and  $X_{i4}$  are two fireflies randomly selected from the whole population ( $i3 \neq i4 \neq i$ ),  $gbest$  is the global best firefly found so far,  $r_4, r_5$ , and  $r_6$  are three uniform random numbers with the range  $(0, 1)$ , and  $r_4 + r_5 + r_6 = 1$ .

In the second neighborhood search operator, A mutation operator of the Cauchy type is conducted. It is expected that the long tail of the Cauchy distribution may help trapped fireflies jump out of the local minima.

$$X_i^3 = X_i + \text{cauchy}(), \tag{17}$$

**Table 1** The 12 benchmark functions used in the experiments.

Name	Function	Search range	Global optimum
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	[-100, 100]	0
Schwefel 2.22	$f_2(x) = \sum_{i=1}^D  x_i  + \prod_{i=1}^D x_i$	[-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	$f_4(x) = \max\{ x_i , 1 \leq i \leq D\}$	[-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	$f_7(x) = \sum_{i=1}^D i \cdot x_i^4 + \text{random}[0, 1)$	[-1.28, 1.28]	0
Schwefel 2.26	$f_8(x) = \sum_{i=1}^D -x_i \cdot \sin(\sqrt{ x_i })$	[-500, 500]	-12569.5
Rastrigin	$f_9(x) = \sum_{i=1}^D [x_i^2 - 10 \cos 2\pi x_i + 10]$	[-5.12, 5.12]	0
Ackley	$f_{10}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)) + 20 + e$	[-32, 32]	0
Griewank	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^D (x_i)^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600, 600]	0
Penalized	$f_{12}(x) = \frac{\pi}{D} \{ \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + \sin(\pi y_{i+1})] + (y_D - 1)^2 + (10 \sin^2(\pi y_1)) + \sum_{i=1}^D u(x_i, 10, 100, 4), y_i = 1 + \frac{x_i + 1}{4}$	[-50, 50]	0
	$u(x_i, a, k, m) = \begin{cases} u(x_i, a, k, m), & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$		

where  $\text{cauchy}()$  is a random number drawn from the Cauchy distribution with a unity scale factor.

In our approach, when firefly  $X_i$  is brighter than firefly  $X_j$ , the firefly  $X_j$  will move toward the firefly  $X_i$ , and the firefly  $X_i$  will be conducted on the above three neighborhood search operators. During the neighborhood search, three trial solutions  $X_i^1$ ,  $X_i^2$ , and  $X_i^3$  are generated by Eqs. (15), (16), and (17), respectively. Then, the best solution among  $X_i$ ,  $X_i^1$ ,  $X_i^2$  and  $X_i^3$  is selected as the new  $X_i$ .

### 3.4 The framework of NSRaFA

The main steps of NSRaFA are described in Algorithm 2, where  $N$  is the population size, FEs is the number of fitness evaluations and MAX\_FEs is the maximum number of function evaluations.

## 4 Experimental verifications

### 4.1 Test problems

To verify the performance of our approach, 12 benchmark functions are utilized in the following experiments (Brest et al. 2006; Fister et al. 2015; Wang et al. 2013). Among these functions,  $f_1$ – $f_5$  are unimodal functions.  $f_6$  is a step function which has one minimum.  $f_7$  is a noisy quartic function.  $f_8$ – $f_{12}$  are multimodal functions with many local minima. All these problems are to be minimized, and the dimensional size  $D$  is set to 30. A brief description of these problems is listed in Table 1.

**Table 2** FA variants used for the comparison

Algorithm	Year	References
The standard FA	2010	Yang (2010)
Variable step size FA (VSSFA)	2015	Yu et al. (2015)
Wise step strategy FA (WSSFA)	2014	Yu et al. (2014)
Memetic FA (MFA)	2012	Fister et al. (2012)
FA with chaos (CFA)	2013	Gandomi et al. (2013)
FA with random attraction (RaFA)	2015	Wang et al. (2016)
Our approach NSRaFA	2015	–

### 4.2 FA variants used and their parameter settings

In this section, we compare the performance of NSRaFA with the standard FA and five other recently proposed FA variants. The involved algorithms are listed in Table 2.

To have a fair comparison, the same population size  $N$  and MAX\_FEs are used for all algorithms. The  $N$  should be small because of the double-loop attraction model in FA. The  $N$  and MAX\_FEs are set to 20 and 5.0E+05, respectively. By the suggestions of Yu et al. (2014, 2015), both  $\beta_0$  and  $\gamma$  are set to 1 for VSSFA and WSSFA. For the standard FA and RaFA,  $\alpha = 0.2$ . The parameter  $\alpha_{\min}$  is set to 0.04 for WSSFA (Yu et al. 2014). In MFA, the  $\beta_{\min}$  is equal to 0.2 (Fister et al. 2012). For CFA, the Gauss map is used for updating the parameter  $\beta$  (Gandomi et al. 2013). For NSRaFA, the initial



**Table 3** Mean best fitness values achieved by the standard FA, VSSFA, WSSFA, MFA, CFA, RaFA, and NSRaFA

Function	FA Mean	VSSFA Mean	WSSFA Mean	MFA Mean	CFA Mean	RaFA Mean	NSRaFA Mean
$f_1$	6.67E+04	5.84E+04	6.34E+04	1.56E-05	3.27E-06	<b>5.36E-184</b>	4.11E-110
$f_2$	5.19E+02	1.13E+02	1.35E+02	1.85E-03	8.06E-04	8.76E-05	<b>1.35E-55</b>
$f_3$	2.43E+05	1.16E+05	1.10E+05	5.89E-05	1.24E-05	4.91E+02	<b>1.59E-109</b>
$f_4$	8.35E+01	8.18E+01	7.59E+01	1.73E-03	8.98E-04	2.43E+00	<b>1.88E-55</b>
$f_5$	2.69E+08	2.16E+08	2.49E+08	2.29E+01	<b>2.06E+01</b>	2.92E+01	2.85E+01
$f_6$	7.69E+04	5.48E+04	6.18E+04	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
$f_7$	5.16E+01	4.43E+01	3.24E-01	1.30E-01	9.03E-02	5.47E-02	<b>4.41E-16</b>
$f_8$	-1563.4	-1854.6	-2012.8	-7634.35	-8207.62	<b>-12066.3</b>	-11963.6
$f_9$	3.33E+02	3.12E+02	3.61E+02	6.47E+01	5.27E+01	2.69E+01	<b>0.00E+00</b>
$f_{10}$	2.03E+01	2.03E+01	2.05E+01	4.23E-04	4.02E-04	3.61E-14	<b>5.89E-16</b>
$f_{11}$	6.54E+02	5.47E+02	6.09E+02	9.86E-03	7.91E-06	<b>0.00E+00</b>	<b>0.00E+00</b>
$f_{12}$	7.16E+08	3.99E+08	6.18E+08	5.04E-08	8.28E-09	4.50E-05	<b>1.57E-32</b>
$w/t/l$	12/0/0	12/0/0	12/0/0	10/1/1	10/1/1	8/2/2	-

The best results among the seven algorithms are shown in bold

$\alpha$ ,  $\beta_{\min}$ , and  $\beta_{\max}$  are set to 0.5, 0.3, and 0.9, respectively. All the experiments are conducted 30 times, and the mean results are reported.

### 4.3 Comparison of the quality of the final solutions

Table 3 presents the experimental results achieved by the standard FA, VSSFA, WSSFA, MFA, CFA, RaFA, and NSRaFA, where “Mean” represents the mean best fitness value. The comparison results between NSRaFA and other algorithms are summarized as  $w/t/l$ , which means that NSRaFA wins in  $w$  functions, ties in  $t$  functions and loses in  $l$  functions, compared with its competitors. As seen, the standard FA, VSSFA, and WSSFA could hardly achieve promising solutions on all test functions, and the proposed NSRaFA obtains much better solutions than them. Recent studies show that both VSSFA and WSSFA work well on some low-dimensional benchmark functions (Yu et al. 2014, 2015). It seems that the problem dimension size seriously affects their performance.

Compared with the standard FA, MFA and CFA can find promising solutions on most test functions. In MFA, the parameters  $\alpha$  and  $\beta$  are dynamically adjusted. In CFA, the  $\beta$  value is updated by a chaotic map function. It demonstrates that the performance of FA greatly depends on its parameter settings. NSRaFA performs better than MFA and CFA on ten functions, while they achieve better results than NSRaFA on  $f_5$ . For  $f_6$ , MFA, CFA, RaFA, and NSRaFA can converge to the global optimum.

From the comparison of NSRaFA with RaFA, both of them employ the random attraction model. It can be seen that NSRaFA and RaFA achieve better results than the other five FA variants. In fact, the RaFA is a hybridization of MFA, the random attraction model and a Cauchy mutation opera-

tor (Wang et al. 2016). NSRaFA outperforms RaFA on eight functions, while RaFA finds better solutions on two functions. For  $f_6$  and  $f_{11}$ , they obtain the same results.

Table 4 gives the mean computational time of the compared FA variants. All algorithms are run on an Intel Core i7-4510U CPU 2.60 GHz with 8.0 GB Memory in the Windows 7 Operating System. From the total average time, both MFA and RaFA almost have the same cost. WSSFA and VSSFA cost much more time than the standard FA, CFA, MFA, and RaFA. It seems that NSRaFA costs the most computational time. The main reason is that the neighborhood search used in NSRaFA costs too much computational time during the search. This is confirmed by calculating the computational time of NSRaFA without neighborhood search. If we decrease the frequency of conducting the neighborhood search, we may reduce the computational cost of NSRaFA and maintain its good performance at the same time.

Figure 4 lists the convergence graphs of NSRaFA and the other six FA variants on all test functions. As shown, NSRaFA converges faster than other algorithms on most functions. Especially on  $f_7$ ,  $f_9$ , and  $f_{12}$ , NSRaFA achieves promising solutions at the beginning of the search stage. For  $f_1$ , RaFA shows a faster convergence rate than NSRaFA.  $f_1$  is a Sphere function, which is unimodal. Too many neighborhood search operations affect the convergence rate of NSRaFA. As mentioned before, RaFA combines MFA, the random attraction model, and a Cauchy mutation operator. Our experiments confirm that RaFA without Cauchy mutation converges faster than the full RaFA. For  $f_8$ , RaFA is slightly faster than NSRaFA. On this function, there are many deep local minima being far from the global optimum. It seems that only the Cauchy mutation is beneficial for escaping from such deep local minima, while the other

**Table 4** Mean computational time (in seconds) achieved by the standard FA, VSSFA, WSSFA, MFA, CFA, RaFA, NSRaFA, and NSRaFA without neighborhood search

Function	FA	VSSFA	WSSFA	MFA	CFA	RaFA	NSRaFA	NSRaFA without neighborhood search
	Mean	Mean	Mean	Mean	Mean	Mean	Mean	Mean
$f_1$	0.53	0.61	1.24	0.53	0.58	0.72	1.69	0.58
$f_2$	0.98	1.07	1.68	0.98	1.01	1.06	2.17	1.05
$f_3$	1.42	1.45	2.14	1.43	1.49	1.37	2.59	1.08
$f_4$	0.78	0.83	1.53	0.81	0.87	0.73	2.05	0.43
$f_5$	0.63	0.67	1.39	0.58	0.60	0.53	1.98	0.26
$f_6$	0.62	0.63	1.36	0.39	0.42	0.47	2.14	0.24
$f_7$	1.78	1.86	2.49	1.81	1.87	2.06	2.97	1.89
$f_8$	1.62	1.65	2.31	1.64	1.71	1.49	2.85	1.30
$f_9$	1.26	1.29	1.92	1.26	1.30	1.15	2.64	0.83
$f_{10}$	1.40	1.41	2.11	1.40	1.39	1.30	2.70	1.03
$f_{11}$	1.56	1.52	2.26	1.51	1.56	1.45	2.78	1.05
$f_{12}$	2.83	2.87	3.53	2.01	2.12	1.97	3.31	1.62
Total average	1.28	1.32	2.00	1.20	1.24	1.19	2.49	0.95

two neighborhood search operators do not work. Though NSRaFA and RaFA can find the global optimum, RaFA is slightly faster than NSRaFA.

To compare the performance of multiple algorithms on the benchmark set, Friedman test is conducted (García et al. 2010). Table 5 presents the mean rankings of the standard FA, VSSFA, WSSFA, MFA, CFA, RaFA, and NSRaFA. The best rank (with the lowest mean rank value) is shown in bold. From the results, the performance of the seven FA variants can be sorted by the mean rank into the following order: NSRaFA, RaFA, CFA, MFA, VSSFA, WSSFA, and the standard FA. The best mean rank is obtained by the proposed NSRaFA. It demonstrates that NSRaFA outperforms the other six FA variants.

#### 4.4 Comparison of the robustness

The above experiment compares the quality of the final solutions achieved by the seven FA variants. In the following experiment, we investigate the robustness of these algorithms. A threshold value of the objective function is selected for each test function. The detailed threshold values are listed in the second column of Table 6. The stopping criterion is to find a fitness value smaller than the predefined threshold before reaching the maximum number of fitness evaluations (MAX\_FEs).

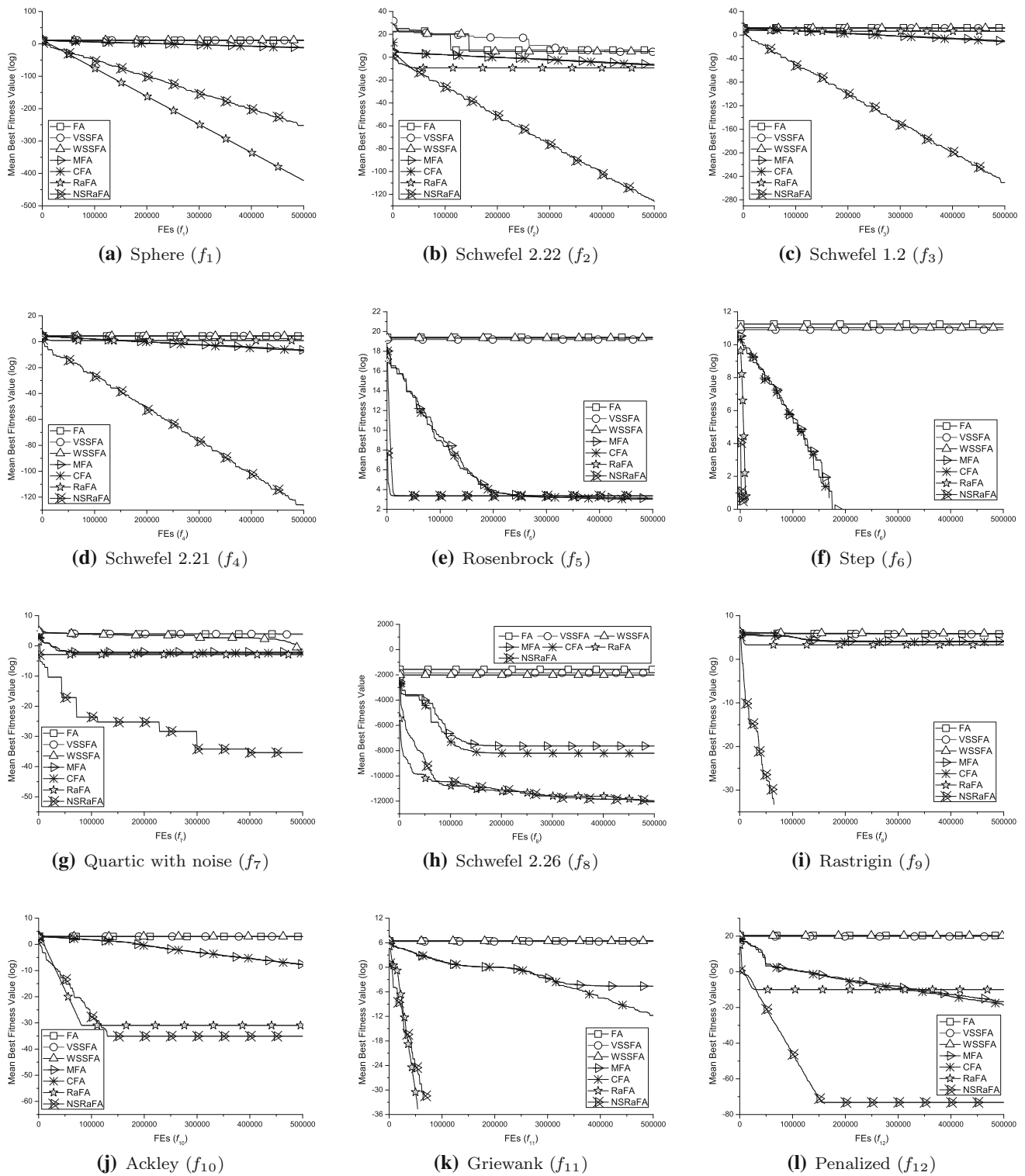
We compare the robustness of each algorithm by measuring the successful running rate (SR). The SR is defined by

$$SR = \frac{n_{suc\_run}}{n\_run}, \quad (18)$$

where  $n\_run$  is the total number of runs and  $n_{suc\_run}$  is the number of successful runs. A successful run means that the algorithm successfully converges to the threshold value within the predefined MAX\_FEs.

In this experiment, the parameters of the standard FA, VSSFA, WSSFA, MFA, CFA, RaFA, and NSRaFA employ the same settings as described in Sect. 4.2. Table 6 presents the results of SR for each algorithm. The highest SR among the seven algorithms is shown in bold. From the results, the standard FA, VSSFA, and WSSFA fail to solve all test functions. Both MFA and CFA achieve 100% SR on only one function ( $f_6$ ). MFA successfully solves  $f_3$  in 2 out of 30 runs, while CFA obtains 25 successful runs. For the rest of the ten functions, they cannot converge to the threshold values in all runs. RaFA achieves 100% SR on five functions, but it also fails to solve another five functions ( $f_3$ – $f_5$ ,  $f_7$ , and  $f_9$ ). For the rest  $f_2$  and  $f_{12}$ , RaFA can successfully solve them in 7 and 20 runs, respectively. NSRaFA obtains 100% SR on all functions except for  $f_5$ . On this function, all algorithms fall into the local minima.

From the total average SR, the standard FA, VSSFA, and WSSFA achieve the lowest SR (0%), because they fail to solve all functions. MFA and CFA fail to solve most functions, and they obtain 8.89 and 15.28%, respectively. By combining MFA and other strategies, RaFA achieves a significant improvement on SR (49.17%). The proposed NSRaFA successfully solves most functions and obtains the highest SR (91.67%). However, it is worth pointing out that the SR will largely depend on the criterion used for determining what a successful run is. Different criteria may result in different rates.



**Fig. 4** The convergence curves of the standard FA, VSSFA, WSSFA, MFA, CFA, RaFA, and NSRaFA on all test functions

### 4.5 Effects of different strategies

The proposed NSRaFA employs three strategies: a dynamic parameter adjustment mechanism, a random attraction model, and three neighborhood search operators. To investigate the

effects of these three strategies, we compare the performance of FA with different strategies. This is helpful to verify the effectiveness of these strategies separately. The involved algorithms are listed as follows.

**Table 5** Mean ranks achieved by Friedman test for the seven FA variants

Algorithm	Mean rank
NSRaFA	<b>1.50</b>
RaFA	2.50
CFA	2.54
MFA	3.46
VSSFA	5.38
WSSFA	5.83
FA	6.79

The best rank (with the lowest mean rank value) is shown in bold

- FA + dynamic parameter adjustment method.
- FA + dynamic parameter adjustment method + random attraction model.
- FA + dynamic parameter adjustment method + neighborhood search.
- The proposed NSRaFA (FA + three strategies).

In the experiments, all algorithms use the same parameter settings as described in Sect. 4.2. Table 7 presents the mean best fitness values achieved by FA with different strategies. From the results of the standard FA (see Table 3) and FA + dynamic parameter adjustment method, the dynamic parameter adjustment method is helpful to achieve significant

**Table 6** Results for successful running rate (SR) under a predefined accuracy level (threshold)

Function	Threshold	FA SR (%)	VSSFA SR (%)	WSSFA SR (%)	MFA SR (%)	CFA SR (%)	RaFA SR (%)	NSRaFA SR (%)
$f_1$	1.00E-10	0	0	0	0	0	<b>100</b>	<b>100</b>
$f_2$	1.00E-05	0	0	0	0	0	23.33	<b>100</b>
$f_3$	1.00E-05	0	0	0	6.67	83.33	0	<b>100</b>
$f_4$	1.00E-05	0	0	0	0	0	0	<b>100</b>
$f_5$	1.00E-05	0	0	0	0	0	0	0
$f_6$	1.00E-10	0	0	0	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>
$f_7$	1.00E-02	0	0	0	0	0	0	<b>100</b>
$f_8$	-1.00E+04	0	0	0	0	0	<b>100</b>	<b>100</b>
$f_9$	1.00E-05	0	0	0	0	0	0	<b>100</b>
$f_{10}$	1.00E-10	0	0	0	0	0	<b>100</b>	<b>100</b>
$f_{11}$	1.00E-10	0	0	0	0	0	<b>100</b>	<b>100</b>
$f_{12}$	1.00E-10	0	0	0	0	0	66.67	<b>100</b>
Total average		0	0	0	8.89	15.28	49.17	<b>91.67</b>

The best results among the seven algorithms are shown in bold

**Table 7** Mean best fitness values achieved by FA with different strategies

Function	FA + dynamic parameter adjustment method Mean	FA + dynamic parameter adjustment method + random attraction model Mean	FA + dynamic parameter adjustment method + neighborhood search Mean	NSRaFA (FA + three strategies) Mean
$f_1$	5.69E-07	<b>3.41E-213</b>	1.63E-07	4.11E-110
$f_2$	3.79E-04	1.01E-54	5.26E-05	<b>1.35E-55</b>
$f_3$	1.21E-06	2.96E+03	8.58E-06	<b>1.59E-109</b>
$f_4$	2.87E-04	2.03E-01	1.75E-04	<b>1.88E-55</b>
$f_5$	<b>2.31E+01</b>	2.91E+01	2.87E+01	2.85E+01
$f_6$	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
$f_7$	8.66E-02	4.32E-02	1.38E-06	<b>4.41E-16</b>
$f_8$	-6509.1	-6627.5	-11366.2	<b>-11963.6</b>
$f_9$	3.89E+01	3.83E+01	2.51E-07	<b>0.00E+00</b>
$f_{10}$	1.93E-04	1.82E-14	3.88E-05	<b>5.89E-16</b>
$f_{11}$	1.69E-06	0.00E+00	4.79E-08	<b>0.00E+00</b>
$f_{12}$	1.86E-09	3.56E-32	5.36E-04	<b>1.57E-32</b>
$w/t/l$	10/1/1	9/1/2	11/1/0	-

The best results among the comparisons are shown in bold

improvements on all test functions. The random attraction model improves the performance of FA + dynamic parameter adjustment method on nine functions. Especially for  $f_1$ – $f_2$  and  $f_{10}$ – $f_{12}$ , FA + dynamic parameter adjustment method + random attraction model achieves much better results than FA + dynamic parameter adjustment method. The neighborhood search improves the performance of FA + dynamic parameter adjustment method on eight functions. Especially for  $f_7$ – $f_9$ , FA + dynamic parameter adjustment method + neighborhood search performs much better than FA + dynamic parameter adjustment method. It seems that the random attraction model is not suitable for  $f_3$  and  $f_4$ , while the neighborhood search obtains slight improvements.

From the above analysis, the random attraction model or the neighborhood search can improve the performance of FA + dynamic parameter adjustment method on most test functions. When combining FA with three strategies, NSRaFA (FA + three strategies) achieves better results than FA + dynamic parameter adjustment method on ten functions. For FA + dynamic parameter adjustment method + random attraction model, introducing the neighborhood search significantly improves the quality of final solutions on  $f_3$ ,  $f_4$ , and  $f_7$ – $f_{10}$ . The random attraction model helps FA + dynamic parameter adjustment method + neighborhood search to obtain significant improvements on  $f_2$ – $f_4$ , and  $f_7$ – $f_{12}$ .

The above results demonstrate that FA with the proposed one or more strategies can significantly improve the performance of the standard FA. Each strategy has its own effects and plays an important role in the search process. By hybridization of FA and the proposed strategies, NSRaFA achieves a superior performance.

#### 4.6 Effects of population size

In this section, we investigate the effects of population size ( $N$ ) on the performance of NSRaFA. In the experiments,  $N$  is set to 20, 30, and 40, respectively. For other parameters, we use the same settings as described in Sect. 4.3.

Table 8 presents the mean best fitness values achieved by NSRaFA under different population size. From the results, NSRaFA with a small  $N$  achieves better solutions than NSRaFA with a large one. The population size seriously affects the accuracy of the final solutions for unimodal functions, such as  $f_1$ – $f_4$  and  $f_7$ . For multimodal functions, NSRaFA with different  $N$  values obtain similar performance.

## 5 Conclusion

This paper presents an improved FA variant called randomly attracted FA with neighborhood search (NSRaFA). The new

**Table 8** Results achieved by NSRaFA under different population sizes

Function	$N = 20$	$N = 30$	$N = 40$
$f_1$	<b>4.11E–110</b>	5.97E–110	3.81E–56
$f_2$	<b>1.35E–55</b>	7.67E–38	2.98E–29
$f_3$	<b>1.59E–109</b>	3.25E–72	2.11E–56
$f_4$	<b>1.88E–55</b>	1.17E–37	6.76E–29
$f_5$	2.85E+01	<b>2.84E+01</b>	2.85E+01
$f_6$	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
$f_7$	<b>4.41E–16</b>	6.29E–13	2.62E–10
$f_8$	<b>–11963.6</b>	–11814.4	–11224.7
$f_9$	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
$f_{10}$	<b>5.89E–16</b>	<b>5.89E–16</b>	<b>5.89E–16</b>
$f_{11}$	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
$f_{12}$	<b>1.57E–32</b>	<b>1.57E–32</b>	1.89E–32

approach employs three strategies: (1) a dynamic parameter adjustment mechanism; (2) a random attraction model; and (3) three neighborhood search operators. The first strategy aims to automatically adjust the control parameters  $\alpha$  and  $\beta$ , and avoid manual parameter settings. The second one focuses on accelerating the convergence rate and reducing the computational time complexity. The last strategy defines a new neighborhood search operation for brighter fireflies (better solutions). Searching the neighborhoods of these brighter fireflies is helpful to find better candidate solutions.

Experiments are conducted on 12 well-known benchmark functions to verify the performance of our approach NSRaFA. Results show that NSRaFA achieves much better solutions than the standard FA, VSSFA, WSSFA, MFA, CFA, and RaFA on the majority of test functions. The comparison of the robustness demonstrates that NSRaFA is the most robust algorithm among all seven FA variants.

The NSRaFA is a hybrid of FA and three proposed strategies. Results show that each strategy has different effects and it plays a significant role for finding good candidate solutions. Moreover, FA with more strategies can achieve better performance than FA without these. For example, FA with two strategies (the random attraction model or the neighborhood search) is better than FA with the dynamic parameter adjustment method. It also confirms the effectiveness of our proposed strategies. Moreover, it seems that the population size seriously affects the accuracy of final solutions only for unimodal functions.

The parameters  $\alpha$  and  $\beta$  can seriously affect the performance of FA. To avoid manual settings, we designed a dynamic parameter adjustment mechanism, which does not consider any search experience of fireflies. By introducing these search experiences, we may construct a self-adaptive parameter strategy to guide the search. This will be investigated in the future work.

**Acknowledgments** This work is supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions, Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, the Humanity and Social Science Foundation of Ministry of Education of China (No. 13YJCZH174), the National Natural Science Foundation of China (Nos. 61305150, 61261039, and 61461032), and the Natural Science Foundation of Jiangxi Province (No. 20142BAB217020).

### Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

## References

- Amiri B, Hossain L, Crawford JW, Wigand RT (2013) Community detection in complex networks: multi-objective enhanced firefly algorithm. *Knowl Based Syst* 46:1–11
- Brest J, Greiner S, Bošković B, Mernik M, Žumer V, (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans Evolut Comput* 10(6):646–657
- Chandrasekaran K, Simon SP, Padhy NP (2013) Binary real coded firefly algorithm for solving unit commitment problem. *Inf Sci* 249:67–84
- Chen BJ, Shu HZ, Coatrieux G, Chen G, Sun XM, Coatrieux JL (2015) Color image analysis by quaternion-type moments. *J Math Imaging Vis* 51(1):124–144
- Chhikara RR, Singh L (2015) An improved discrete firefly and t-Test based algorithm for blind image steganalysis. In: The 6th international conference on intelligent systems, modelling and simulation (ISMS), pp 58–63
- Coelho LS, Mariani VC (2013) Improved firefly algorithm approach applied to chiller loading for energy conservation. *Energy Build* 59:273–278
- Das S, Abraham A, Chakraborty U, Konar A (2009) Differential evolution using a neighborhood-based mutation operator. *IEEE Trans Evol Comput* 13(3):526–553
- Dorigo M, Maniezzo V, Colonna A (1996) The ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern Part B Cybern* 26:29–41
- Duang H, Luo Q (2015) New progresses in swarm intelligence-based computation. *Int J Bio-Inspir Comput* 7(1):26–35
- Farahani SM, Abshouri AA, Nasiri B, Meybodi MR (2011) A Gaussian firefly algorithm. *Int J Mach Learn Comput* 1(5):448–453
- Fister Jr I, Yang XS, Fister I, Brest J (2012) Memetic firefly algorithm for combinatorial optimization. In: *Bioinspired optimization methods and their applications (BIOMA 2012)*, pp 1–14
- Fister I Jr, Fister I, Yang XS, Brest J (2013) A comprehensive review of firefly algorithms. *Swarm Evolut Comput* 13:34–46
- Fister I, Yang XS, Brest J, Fister I Jr (2013) Modified firefly algorithm using quaternion representation. *Exp Syst Appl* 40(18):7220–7230
- Fister I Jr, Yang XS, Fister I, Brest J, Fister D (2013) A brief review of nature-inspired algorithms for optimization. *Elektrotehnički Vestnik* 80(3):1–7
- Fister I Jr, Perc M, Kamal SM, Fister I (2015) A review of chaos-based firefly algorithms: perspectives and research challenges. *Appl Math Comput* 252:155–165
- Fister I Jr, Yang XS, Brest J, Fister D, Fister I (2015) Analysis of randomisation methods in swarm intelligence. *Int J Bio-Inspir Comput* 7(1):36–49
- Florence AP, Shanthi V (2014) A load balancing model using firefly algorithm in cloud computing. *J Comput Sci* 10(7):1156–1165
- Fu ZJ, Sun XM, Liu Q, Zhou L, Shu JG (2015) Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. *IEICE Trans Commun* E98–B(1):190–200
- Gandomi AH, Yang XS, Alavi AH (2013) Mixed variable structural optimization using firefly algorithm. *Comput Struct* 89(23–24):2325–2336
- Gandomi AH, Yang XS, Talatahari S, Alavi AR (2013) Firefly algorithm with chaos. *Commun Nonlinear Sci Numer Simul* 18(1):89–98
- García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. *Inf Sci* 180(20):2044–2064
- Gopinadh V, Singh A (2015) Swarm intelligence approaches for cover scheduling problem in wireless sensor networks. *Int J Bio-Inspir Comput* 7(1):50–61
- Gu B, Sheng VS, Tay KY, Romano W, Li S (2015) Incremental support vector learning for ordinal regression. *IEEE Trans Neural Netw Learn Syst* 26(7):1403–1416
- Gu B, Sheng VS, Wang ZJ, Ho D, Osman S, Li S (2015) Incremental learning for  $\nu$ -support vector regression. *Neural Netw* 67:140–150
- Hassanzadeh T, Vojodi H, Moghadam AME (2011) An image segmentation approach based on maximum variance intra-cluster method and firefly algorithm. In: *The 7th international conference on natural computation (ICNC)*, pp 1817–1821
- Hong MH (2012) Vector quantization using the firefly algorithm for image compression. *Exp Syst Appl* 39(1):1078–1091
- Kazem A, Sharifi E, Hussain F, Saberi M, Hussain OK (2013) Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Appl Soft Comput* 13(2):947–958
- Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: *Proceedings of IEEE international conference on neural networks*, pp 1942–1948
- Kougianos E, Mohanty SP (2015) A nature-inspired firefly algorithm based approach for nanoscale leakage optimal RTL structure. *Integr VLSI J* 51:46–60
- Li J, Li XL, Yang B, Sun XM (2015) Segmentation-based image copy-move forgery detection scheme. *IEEE Trans Inf Forensics Secur* 10(3):507–518
- Liang RH, Wang JC, Chen YT, Tseng WT (2015) An enhanced firefly algorithm to multi-objective optimal active/reactive power dispatch with uncertainties consideration. *Int J Electr Power Energy Syst* 64:1088–1097
- Long NC, Meesad P, Unger H (2015) A highly accurate firefly based algorithm for heart disease prediction. *Exp Syst Appl* 42(21):8221–8231
- Ma TH, Zhou JJ, Tang ML, Tian Y, Al-dhelaan A, Al-rodhann M, Lee S (2015) Social network and tag sources based augmenting collaborative recommender system. *IEICE Trans Inf Syst* E98–D(4):902–910
- Mahapatra S, Panda S, Swain SC (2014) A hybrid firefly algorithm and pattern search technique for SSSC based power oscillation damping controller design. *Ain Shams Eng J* 5:1177–1188
- Marichelvam MK, Prabaharan T, Yang XS (2014) A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems. *IEEE Trans Evol Comput* 18(2):301–305
- Miguel LFF, Lopez RH, Miguel LFF (2013) Multimodal size, shape, and topology optimisation of truss structures using the firefly algorithm. *Adv Eng Softw* 56:23–37
- Poursalehi N, Zolfaghari A, Minuchehr A (2013) Multi-objective loading pattern enhancement of PWR based on the discrete firefly algorithm. *Ann Nucl Energy* 57:151–163
- Rahmani A, MirHassani SA (2014) A hybrid firefly-genetic algorithm for the capacitated facility location problem. *Inf Sci* 283:70–78

- Ren YJ, Shen J, Wang J, Han J, Lee S (2015) Mutual verifiable provable data auditing in public cloud storage. *J Int Technol* 16(2):317–323
- Sahu RK, Panda S, Padhan S (2015) A hybrid firefly algorithm and pattern search technique for automatic generation control of multi area power systems. *Int J Electr Power Energy Syst* 64:9–23
- Saraç E, Özel SA (2013) Web page classification using firefly optimization. In: *IEEE international symposium on innovations in intelligent systems and applications (INISTA)*, pp 1–5
- Sayadi MK, Hafezalkotob A, Naini S (2013) Firefly-inspired algorithm for discrete optimization problems: an application to manufacturing cell formation. *J Manuf Syst* 32(1):78–84
- Senthilnath J, Omkar SN, Mani V (2011) Clustering using firefly algorithm: performance study. *Swarm Evolut Comput* 1(3):164–171
- Shen J, Tan HW, Wang J, Wang JW, Lee S (2015) A novel routing protocol providing good transmission reliability in underwater sensor networks. *J Int Technol* 16(1):171–178
- Shomalnasab F, Sadeghzadeh M, Esmaeilpour M (2014) An optimal similarity measure for collaborative filtering using firefly algorithm. *J Adv Comput Res* 5(3):101–111
- Srivatsava PR, Mallikarjun B, Yang XS (2013) Optimal test sequence generation using firefly algorithm. *Swarm Evolut Comput* 8:44–53
- Tilahun SL, Ong HC (2012) Modified firefly algorithm. *J Appl Math* 2012:1–12. doi:[10.1155/2012/467631](https://doi.org/10.1155/2012/467631)
- Wang H, Wu ZJ, Rahnamayan S, Li CH, Zeng SY, Jiang DZ (2011) Particle swarm optimization with simple and efficient neighbourhood search strategies. *Int J Innov Comput Appl* 3(2):7–104
- Wang H, Rahnamayan S, Sun H, Omran MGH (2013) Gaussian barebones differential evolution. *IEEE Trans Cybern* 43(2):634–647
- Wang H, Sun H, Li CH, Rahnamayan S, Pan JS (2013) Diversity enhanced particle swarm optimization with neighborhood search. *Inf Sci* 223:119–135
- Wang B, Li DX, Jiang JP, Liao YH (2014) A modified firefly algorithm based on light intensity difference. *J Comb Optim* 31(3):1045–1060
- Wang H, Wang WJ, Sun H, Rahnamayan S (2016) Firefly algorithm with random attraction. *Int J Bio-Inspir Comput* 8(1):33–41
- Wen XZ, Shao L, Xue Y, Fang W (2015) A rapid learning algorithm for vehicle classification. *Inf Sci* 295:395–406
- Xia ZH, Wang XH, Sun XM, Liu QS, Xiong NX (2014) Steganalysis of LSB matching using differences between nonadjacent pixels. *Multimed Tools Appl*. doi:[10.1007/s11042-014-2381-8](https://doi.org/10.1007/s11042-014-2381-8)
- Xia ZH, Wang XH, Sun XM, Wang Q (2015) A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. *IEEE Trans Parallel Distrib Syst*. doi:[10.1109/TPDS.2015.2401003](https://doi.org/10.1109/TPDS.2015.2401003)
- Xia ZH, Wang XH, Sun XM, Wang BW (2014) Steganalysis of least significant bit matching using multi-order differences. *Secur Commun Netw* 7(8):1283–1291
- Xie SD, Wang YX (2014) Construction of tree network with limited delivery latency in homogeneous wireless sensor networks. *Wirel Pers Commun* 78(1):231–246
- Xu M, Liu GZ (2013) A multipopulation firefly algorithm for correlated data routing in underwater wireless sensor networks. *Int J Distrib Sens Netw*. doi:[10.1155/2013/865154](https://doi.org/10.1155/2013/865154)
- Yang XS, Deb S (2009) Cuckoo search via Levy flights. In: *World congress on nature and biologically inspired computing (NaBIC 2009)*, pp 210–214
- Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: *Nature inspired cooperative strategies for optimization (NICSO 2010)*, Springer, Berlin, pp 65–74
- Yang XS (2008) *Nature-inspired metaheuristic algorithms*. Luniver Press, London
- Yang XS (2010) *Engineering optimization: an introduction with metaheuristic applications*. Wiley, New York
- Yang XS, Hosseini SSS, Gandomi AH (2012) Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. *Appl Soft Comput* 12(3):1180–1186
- Yu SH, Su SB, Lu QP, Huang L (2014) A novel wise step strategy for firefly algorithm. *Int J Comput Math* 91(12):2507–2513
- Yu SH, Zhu SL, Ma Y, Mao DM (2015) A variable step size firefly algorithm for numerical optimization. *Appl Math Comput* 263:214–220
- Zheng YH, Jeon B, Xu DH, Wu QMJ, Zhang H (2015) Image segmentation by generalized hierarchical fuzzy C-means algorithm. *J Intell Fuzzy Syst* 28(2):961–973