Information Sciences 382–383 (2017) 374–387

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Firefly algorithm with neighborhood attraction

Hui Wang[a,b,*], Wenjun Wang[c], Xinyu Zhou[d], Hui Sun[a,b], Jia Zhao[a,b], Xiang Yu[a,b], Zhihua Cui[e]

[a] *Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing, Nanchang Institute of Technology, Nanchang 330099, China*
[b] *School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China*
[c] *School of Business Administration, Nanchang Institute of Technology, Nanchang 330099, China*
[d] *College of Computer and Information Engineering, Jiangxi Normal University, Nanchang 330022, China*
[e] *School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China*

ARTICLE INFO

Article history:
Received 13 March 2016
Revised 8 October 2016
Accepted 14 December 2016
Available online 15 December 2016

Keywords:
Firefly algorithm (FA)
Attraction
Neighborhood attraction
Random attraction
Optimization

ABSTRACT

Firefly algorithm (FA) is a new optimization technique based on swarm intelligence. It simulates the social behavior of fireflies. The search pattern of FA is determined by the attractions among fireflies, whereby a less bright firefly moves toward a brighter firefly. In FA, each firefly can be attracted by all other brighter fireflies in the population. However, too many attractions may result in oscillations during the search process and high computational time complexity. To overcome these problems, we propose a new FA variant called FA with neighborhood attraction (NaFA). In NaFA, each firefly is attracted by other brighter fireflies selected from a predefined neighborhood rather than those from the entire population. Experiments are conducted using several well-known benchmark functions. The results show that the proposed strategy can efficiently improve the accuracy of solutions and reduce the computational time complexity.

1. Introduction

Many real-world problems can be converted into optimization problems, e.g., water distribution network design [5], economic dispatch [15], recurrent network design [23], mobile device localization [8], and dynamic resource reallocation [9]. It is becoming increasingly difficult for traditional optimization algorithms to meet current requirements. Hence, there is a growing need for more effective algorithms. In recent decades, some new optimization algorithms inspired by swarm intelligence have been proposed, e.g., particle swarm optimization (PSO) [26], ant colony optimization (ACO) [7], artificial bee colony algorithm (ABC) [24], and firefly algorithm (FA) [46].

FA is a new swarm intelligence algorithm originally developed by Yang in 2008 [46]. It simulates the social behavior of fireflies. Fireflies use flashing to attract mating partners. The movement of fireflies is determined by the resulting attraction, and the attractiveness is related to the intensity of the emitted light. Since the development of FA, it has been successfully applied to various optimization problems, such as structure design [16], stock forecasting [25], and production scheduling [32].

* Corresponding author.
E-mail addresses: huiwang@whu.edu.cn (H. Wang), wangwenjun881@126.com (W. Wang), xyzhou@jxnu.edu.cn (X. Zhou), sun_hui2006@163.com (H. Sun), zhaojia925@163.com (J. Zhao), xyuac@ust.hk (X. Yu), zhihuacui@gmail.com (Z. Cui).

http://dx.doi.org/10.1016/j.ins.2016.12.024
0020-0255/

Although FA has been shown to be an effective optimization technique in many optimization problems, it has some drawbacks. In FA, each firefly can be attracted by all other brighter fireflies. Too many attractions may result in oscillations during the search process and high computational time complexity. To overcome these problems, a new FA variant, namely neighborhood attraction FA (NaFA), is proposed in this paper. NaFA differs from the standard FA in that each firefly in NaFA is attracted only by some brighter fireflies selected from a predefined neighborhood. Thus, NaFA is expected to reduce the attractions among fireflies. The main contributions of this paper can be summarized as follows. First, a novel attraction model (i.e., neighborhood attraction) is proposed on the basis of the population topology. Second, the computational effort required by different attraction models is analyzed. Third, the proposed approach is shown to effectively reduce the oscillations during the search process and the computational time complexity.

The remainder of this paper is organized as follows. Section 2 presents and briefly reviews the standard FA. Section 3 describes the neighborhood attraction model. Section 4 presents and discusses the experimental results. Finally, Section 5 summarizes our findings and concludes the paper.

2. Firefly algorithm and its brief review

2.1. Firefly algorithm (FA)

Like PSO, FA is a population-based stochastic search algorithm. Each member (firefly) in the population represents a candidate solution in the search space. Fireflies move toward other positions and find potential candidate solutions. The attractiveness is determined by the intensity of the emitted light, which is usually measured by the fitness value.

Let X_i be the ith firefly in the population, where $i = 1, 2, \ldots, N$ and N is the population size. The attractiveness between two fireflies X_i and X_j can be calculated as follows [46].

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

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

where $d = 1, 2, \ldots, D$; D is the problem dimension; r_{ij} is the distance between X_i and X_j; and x_{id} and x_{jd} are the dth dimension of X_i and X_j, respectively. Further, the parameter β_0 denotes the attractiveness at the distance $r = 0$, and γ is the light absorption coefficient. Based on [46], Γ is set to $\frac{1}{\Gamma^2}$, where Γ is the length scale of the designed variables.

Each firefly X_i is compared with all other fireflies X_j, where $j = 1, 2, \ldots, N$ and $j \neq i$. If X_j is brighter (better) than X_i, X_i will be attracted to and move toward X_j. The movement of X_i can be defined by [46]

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

where ϵ_i is a random value uniformly distributed in the range $[-0.5, 0.5]$ and $\alpha \in [0, 1]$ is the step factor.

The main steps of the standard FA are described in Algorithm 1, where N is the population 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

Algorithm 1: Standard FA.

```
/* Population initialization                                    */
Randomly initialize the population and generate N fireflies(solutions) X_i, i = 1, 2, ..., N;
Compute the fitness value of each firefly;
FEs = N;
while FEs ≤ MAX_FEs do
    for i = 1 to N do
        for j = 1 to N do
            /* Movement through attraction                      */
            if f(X_j) < f(X_i) then
                Move X_i toward X_j according to Eq. 3;
                Compute the fitness value of the new X_i;
                FEs++;
            end
        end
    end
end
```

this paper, we consider only minimization problems. Thus, if $f(X_j) < f(X_i)$, X_j is better than X_i (firefly j is brighter than firefly i).

To explain the framework of FA, we summarize its main steps as follows.

Step 1: **Initialization** – Generate randomly N solutions ($X_i = [x_{id}, x_{i2}, \ldots, x_{id}]$) as an initial population according to the following expression:

$$x_{id} = low + rand(0,1)(up - low), \tag{4}$$

where $i = 1, 2, \ldots, N$, $d = 1, 2, \ldots, D$. low and up are the lower and upper bounds for the dimension d, respectively. $rand$(0, 1) is a random number in the range [0, 1]. Each firefly in the initial population is evaluated using the objective function.

Step 2: **Movement (attraction)** – For each solution X_i, we compare it with other all solutions X_j in the population, where $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, N$, and $i \neq j$. If the objective function value of X_j is better than X_i, X_i moves towards X_j and changes its position according to Eq. (3). Each solution is evaluated according to the updated position.

Step 3: **Stopping criteria** – If the stopping criteria is satisfied, then stop the algorithm; otherwise go to step 2.

2.2. Brief review of FA

In recent years, FA has emerged as a well-known optimization tool that has been used to solve various optimization problems [10]. In this section, a brief review of FA is presented.

In [12], Fister et al. presented a memetic FA (MFA), in which the attractiveness β is redefined and a dynamic parameter strategy is used to update the parameter α. The modifications of MFA can be described as follows.

$$x_{id}(t+1) = x_{id}(t) + \beta\left(x_{jd}(t) - x_{id}(t)\right) + \alpha(t)s_d\epsilon_i, \tag{5}$$

$$\beta = \beta_{min} + (\beta_0 - \beta_{min})e^{-\gamma r_{ij}^2}, \tag{6}$$

$$\alpha(t+1) = \left(\frac{1}{9000}\right)^{\frac{1}{t}}\alpha(t), \tag{7}$$

where $\beta_{\min}$ is the minimum value of β and s_d is the length scale of each designed variable. As shown in Eq. 6, the attractiveness β is limited in the range $[\beta_{\min}, \beta_0]$.

Like PSO, FA is sensitive to the control parameters. Therefore, some new FA variants have been proposed by introducing different parameter strategies. In [17], 12 different chaotic maps were used to update γ and β. In [48], Yu et al. proposed a variable step size FA (VSSFA), in which the parameter α is dynamically adjusted with increasing generations. A wise step strategy for FA (WSSFA) was presented in [47], whereby each firefly in the population has an independent parameter α_i, $i = 1, 2, \ldots, N$. During the search process, α_i is updated by considering the information of the previous best firefly ($pbest_i$) and the global best firefly ($gbest$).

Some other improvements for FA have been proposed. In mathematics, quaternions constitute a number system that extends the complex numbers [3]. In [11], Fister et al. used quaternions to represent individuals in FA. Their results showed that the proposed strategy is useful for avoiding premature convergence. Hassanzadeh and Kanan [21] presented a fuzzy FA (FFA) to enhance the exploration and improve the global search of FA. Verma et al. [39] designed opposition and dimensional based FA (ODFA), which applies opposition-based learning (OBL) and the dimensional-based strategy to FA. Their experiments on 11 benchmark functions showed that ODFA performs better than FA, PSO, ACO, and DE. Wang et al. [40] proposed a modified FA called FA with random attraction (RaFA), which employs a random attraction model and a Cauchy mutation operator.

Kazem et al. [25] presented a new FA for stock market price forecasting, which employs two new strategies, namely chaotic FA and support vector regression (SVR) [19]. Their simulation results showed that FA can achieve good performance. In [16], Gandomi et al. used FA to solve mixed structural optimization problems. Recommender systems (RS) can provide users with the required information [31]. Shomalnasab et al. [38] used FA to improve the recommendation accuracy. To implement image segmentation [27,49], Hassanzadeh et al. [22] used FA to optimize Otsu's method. Classification is an important research area in machine learning [28,42]. Saraç and Özel [35] designed an improved FA to select a subset of features for web page classification. Capacitated facility location problem (CFLP) is a well-known combinatorial optimization problem. Rahmani and MirHassani [33] combined FA and GA to solve this problem. Their results showed that the hybrid algorithm performs better than PSO and CPLEX (an optimization software package). To solve dynamic multidimensional knapsack problems, Baykasoğlu and Ozsoydan [1] proposed an effective FA that employs a diversity mechanism and an adaptive move procedure. Their simulation results showed that the proposed approach outperforms differential evolution (DE), GA, and the standard FA. Sensor networks have been widely used in various fields [37]. In [45], a multi-population FA was proposed for correlated data routing in underwater sensor networks (UWSNs). Cloud computing is a new computing method based on the Internet, which can share computational resources among computers [14,34,43]. In [13], FA was used to maximize the usage rate of resources in cloud servers. Wireless mesh networks (WMN) are communication networks that can be used in adverse environments [20]. In [36], FA was used to optimize the routing in WMN.

Many optimization problems are multi-objective optimization problems (MOPs) in the real world [2,29,30]. FA was originally used to solve single-objective optimization problems. Recently, FA has been applied to MOPs. Marichelvam et al. [32] designed a discrete FA for solving a multi-objective hybrid flow shop scheduling problem, in which makespan and mean flow time are the optimized objectives. Their computational results showed that the proposed algorithm outperforms genetic algorithm (GA), parallel GA (PGA), simulated annealing (SA), and ACO.

3. Proposed approach

3.1. Motivation

As mentioned earlier, the search behavior of FA is determined by the attractions among fireflies. Each firefly can be attracted by all other brighter fireflies. This is known as the full attraction model [40]. We have noted that there are too many attractions among fireflies in the full attraction model. To calculate the number of attractions at every generation, we sort all the fireflies in the population according to their fitness values. Assume that there are N fireflies in the population. The first firefly X_1 is the brightest (best) one, and the Nth firefly X_N is the least bright (worst) one. Thus, $X_1, X_2, \ldots, X_N$ are attracted by $0, 1, \ldots, N-1$ brighter fireflies, respectively. Therefore, the total number of attractions (T_a) at each generation is

$$T_a = 0 + 1 + \cdots + N - 1 = \frac{N(N-1)}{2}. \tag{8}$$

It can be concluded that the average number of attractions for each firefly is $\frac{T_a}{N} = \frac{N-1}{2}$. Although the attraction can enable the fireflies to find new candidate solutions, too many attractions may result in oscillations during the search process and high computational time complexity. Let $O(f)$ be the computational time complexity of the fitness evaluation function $f(\cdot)$. The computational time complexity of the standard FA is $O(G_{\max} \cdot N^2 \cdot f)$, where $G_{\max}$ is the maximum number of generations. In comparison, the computational time complexity of PSO is only $O(G_{\max} \cdot N \cdot f)$. However, as N is usually not large, this may not be the main issue. The manner of attraction could be more important.

To reduce the attractions, we proposed random attraction FA (RaFA) in our previous study [40]. In RaFA, at every generation, each firefly X_i is compared with another firefly X_j ($i \neq j$) randomly selected from the current population. If X_j is brighter than X_i, X_i will move toward X_j. Thus, the number of attractions for each firefly is not greater than 1. Although random attraction can effectively reduce the computational time complexity and accelerate the search, it may result in premature convergence. To overcome this problem, both the random attraction model and Cauchy mutation have been used in RaFA [40]. Cauchy mutation can potentially enhance the global search and prevent the algorithm from being trapped in local minima.

3.2. Neighborhood attraction

Too many attractions may result in oscillations during the search process, while too few attractions may lead to premature convergence. Therefore, the number of attractions is very important. To achieve a trade-off between full attraction and random attraction, this paper proposes a new FA variant called FA with neighborhood attraction (NaFA), which employs a neighborhood attraction (Na) model inspired by the k-neighborhood concept [6].

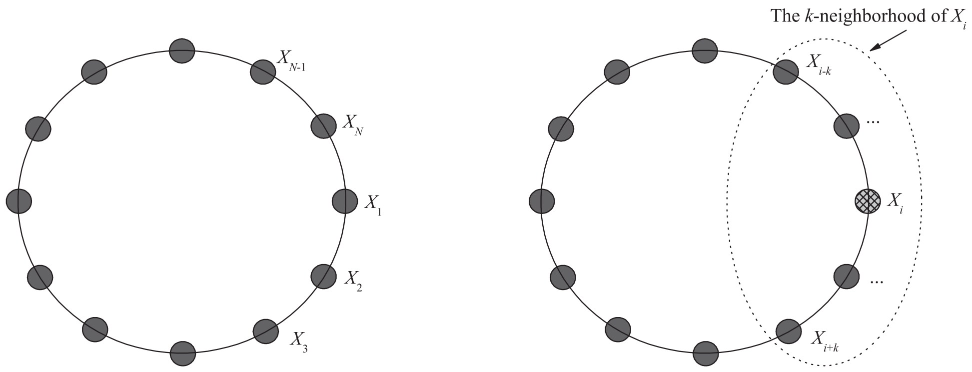
Assume that all N fireflies in the population are organized in a circle topology in terms of their indices. For instance, X_N connects with X_1, and X_1 connects with X_2. Then, X_1 has two immediate neighbors: X_2 and X_N. Fig. 1(a) shows an example of the circle topology, where the population size N is 12. Based on [6], the k-neighborhood of X_i consists of $2k+1$ fireflies $\{X_{i-k}, \ldots, X_i, \ldots, X_{i+k}\}$, where k is an integer and $1 \leq k \leq \frac{N-1}{2}$. To illustrate the k-neighborhood concept, Fig. 1(b) shows the k-neighborhood of X_i.

In our neighborhood attraction model, each firefly X_i is attracted by all other brighter fireflies selected from its k-neighborhood rather than those from the entire population. To compare the full attraction model used in the standard FA with the proposed neighborhood attraction model, Fig. 2 shows two examples for the attractions of X_i, where N is set to 12. In the full attraction model, X_i moves toward all other brighter fireflies selected from the entire population. In the neighborhood attraction model, X_i is only attracted by all other brighter fireflies chosen from its k-neighborhood. Compared to the entire population, the k-neighborhood is much smaller.

To analyze the number of attractions generated by the neighborhood attraction model, all the fireflies in the population are sorted initially. Then, they are organized in a circle topology. In the k-neighborhood of X_i, there are k fireflies $(X_{i+1}, \ldots, X_{i+k})$ that are brighter than X_i. Thus, each firefly X_i is attracted by k other brighter fireflies. Therefore, the total number of attractions (T_{na}) in the neighborhood attraction model is

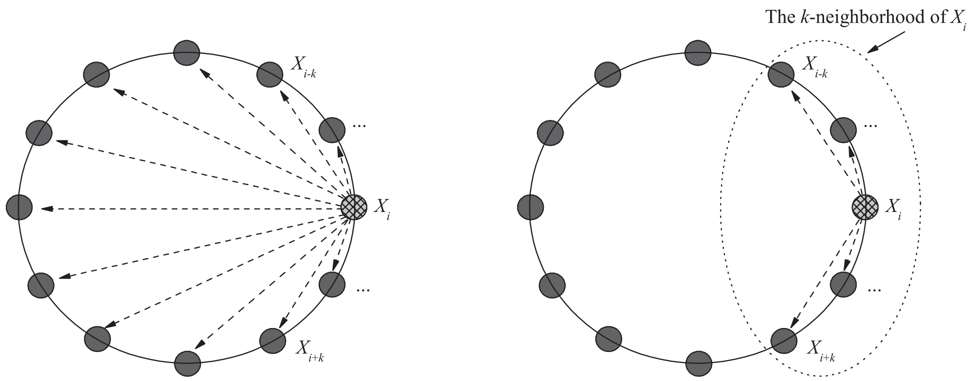
$$T_{ra} = k + k + \cdots + k = Nk. \tag{9}$$

It can be seen that $T_{na} = Nk \leq \frac{N(N-1)}{2}$. In our empirical studies, k is much smaller than $\frac{N-1}{2}$.



(a) Circle topology (b) k-neighborhood of X_i

Fig. 1. Circle topology and k-neighborhood of X_i, where $N = 12$.



(a) Full attraction model (b) Neighborhood attraction model

Fig. 2. Full attraction model vs. neighborhood attraction model, where $N = 12$.

3.3. Implementation

Besides the neighborhood attraction model, NaFA uses MFA as the parent algorithm. In fact, NaFA is a hybridization of neighborhood attraction model and MFA. The main steps of NaFA 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.

To explain the framework of NaFA, we summarize its main steps as follows.

Step 1: **Initialization** – Generate randomly N solutions ($X_i = [x_{id}, x_{i2}, \ldots, x_{id}]$) as an initial population according to Eq. (4). Each firefly in the initial population is evaluated using the objective function.

Step 2: **Parameter updating** – Calculate the parameter α according to Eq. (7).

Step 3: **Movement (neighborhood attraction)** – For each solution X_i, we compare it with other all solutions X_j in the k-neighborhood of X_i, where $i = 1, 2, \ldots, N$, $j = i - k, \ldots, i, \ldots, i + k$, and $i \neq j$. If $j < 1$ or $j > N$, then $j = (j + N)\%N$. Calculate the attractiveness β according to Eq. (6). If the objective function value of X_j is better than X_i, X_i moves towards X_j and changes its position according to Eq. (5). Each solution is evaluated according to the updated position.

Step 4: **Stopping criteria** – If the stopping criteria is satisfied, then stop the algorithm; otherwise go to step 2.

The computational time complexity of NaFA is $O(G_{\max} \cdot N \cdot k \cdot f)$, where k is much less than $\frac{N-1}{2}$. In our experiments, $k = 3$ is a good choice for NaFA. Therefore, the computational time complexity of NaFA is much lower than the standard FA and other FA variants under the full attraction model.

4. Experimental study

4.1. Benchmark functions

In the experiments, 13 well-known benchmark functions were used to evaluated the performance of our approach. These functions have been considered in previous studies [4,44]. Among these functions, f_1–f_5 are unimodal functions, f_6 is a step

Algorithm 2: Proposed NaFA.

/* Population initialization */
Randomly initialize the population and generate N fireflies(solutions) X_i, $i = 1, 2, \ldots, N$;
Compute the fitness value of each firefly;
FEs $= N$;
while $FEs \leq MAX\_FEs$ **do**
  Update the parameter α according to Eq. 7;
  for $i = 1$ *to* N **do**
    /* Neighborhood attraction model */
    for $j = i - k$ *to* $i + k$ **do**
      if $j \neq i$ **then**
        Set $j = (j + N)\%N$;
        Calculate the attractiveness β according toEq. 6;
        /* Movement through attraction */
        if $f(X_j) < f(X_i)$ **then**
          Move X_i toward X_j according to Eq. 5;
          Compute the fitness value of the new X_i;
          FEs++;
        end
      end
    end
  end
end

Table 1
Benchmark functions used in the experiments, where D is the problem dimension.

Name	Function	Search range	Global optimum
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]$	0
Schwefel 2.22	$f_2(x) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} x_i$	$[-10, 10]$	0
Schwefel 1.2	$f_3(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	$[-100, 100]$	0
Schwefel 2.21	$f_4(x) = \max\{\lvert x_i\rvert, 1 \leq i \leq D\}$	$[-100, 100]$	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	$f_7(x) = \sum_{i=1}^{D} i \cdot x_i^4 + random[0, 1)$	$[-1.28, 1.28]$	0
Schwefel 2.26	$f_8(x) = \sum_{i=1}^{D} -x_i \cdot \sin(\sqrt{\lvert x_i\rvert}) + 418.9829 \cdot D$	$[-500, 500]$	0
Rastrigin	$f_9(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos 2\pi x_i + 10]$	$[-5.12, 5.12]$	0
Ackley	$f_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	$[-32, 32]$	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 1	$f_{12}(x) = \frac{\pi}{D}\{\sum_{i=1}^{D-1}(y_i - 1)^2[1 + \sin(\pi y_{i+1})] + (y_D - 1)^2) + (10\sin^2(\pi y_1))\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4),\ y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} u(x_i, a, k, m), & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]$	0
Penalized 2	$f_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2[1 + \sin^2(2\pi x_D)]\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$	$[-50, 50]$	0

function that has one minimum and is discontinuous, f_7 is a noisy quartic function, and f_8–$f_1$3 are multimodal functions with many local minima. All the functions are minimization problems, and their descriptions are given in Table 1. In this paper, the problem dimension D is set to 30.

4.2. Investigations of the parameter k

The size of the k-neighborhood is an important factor in the performance of NaFA. The parameter k satisfies $1 \leq k \leq \frac{N-1}{2}$. For $k = 1$, each firefly is attracted by two other brighter fireflies at most. For $k = \frac{N-1}{2}$, the neighborhood attraction model is the same as the full attraction model. To investigate the effects of k on the performance of NaFA, different values of k were tested in the following experiments. This enabled us to determine a good choice of k.

Table 2
Mean best fitness values achieved by NaFA with different k values. The best results are indicated in bold.

Function	$k=1$ Mean	$k=2$ Mean	$k=3$ Mean	$k=4$ Mean	$k=5$ Mean	$k=7$ Mean	$k=9$ Mean
f_1	**8.44E−95**	1.62E−45	4.43E−29	6.91E−21	6.26E−16	2.40E−10	2.41E−07
f_2	**4.31E−48**	1.85E−23	2.98E−15	3.97E−11	1.15E−08	6.38E−06	2.50E−04
f_3	1.98E+02	1.33E−01	**2.60E−28**	3.85E−20	2.25E−15	8.22E−10	1.07E−06
f_4	7.57E−01	**2.09E−23**	3.43E−15	3.76E−11	1.05E−08	7.09E−06	2.68E−04
f_5	2.79E+01	6.71E+01	2.39E+01	5.73E+01	2.22E+01	2.18E+01	**2.13E+01**
f_6	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_7	4.21E−02	4.93E−02	2.91E−02	3.51E−02	**2.02E−02**	3.51E−02	2.70E−02
f_8	7.23E+03	6.50E+03	6.86E+03	6.36E+03	6.69E+03	**6.09E+03**	6.22E+03
f_9	3.02E+01	3.32E+01	**2.09E+01**	3.35E+01	3.62E+01	2.65E+01	3.91E+01
f_{10}	2.90E−14	**2.66E−14**	3.02E−14	1.96E−11	5.76E−09	3.69E−06	1.34E−04
f_{11}	1.85E−16	3.29E−03	**0.00E+00**	**0.00E+00**	4.93E−03	4.80E−10	5.74E−07
f_{12}	2.08E−03	**1.61E−32**	1.36E−31	2.18E−23	1.55E−18	7.37E−13	9.88E−10
f_{13}	3.65E−32	**1.92E−32**	2.13E−30	2.94E−22	2.18E−17	3.66E−03	1.33E−08

Table 3
Mean rank achieved by the Friedman test for NaFA with different k values. The best rank is indicated in bold.

NaFA	Mean rank
$k=3$	**2.81**
$k=2$	3.54
$k=4$	3.69
$k=1$	4.23
$k=7$	4.35
$k=5$	4.38
$k=9$	5.00

The population size N and MAX_FEs were set to 20 and 5.0E+05, respectively. The initial α, γ, β_{min}, and β_0 were set to 0.5, 1.0, 0.2, and 1.0, respectively. Since $N=20$, k should satisfy $1 \le k \le 9$. Then, k was set to 1, 2, 3, 4, 5, 7, and 9, respectively. For each value of k, NaFA was run 30 times, and the mean best fitness values were recorded.

Table 2 summarizes the computational results achieved by NaFA with different k values, where "Mean" denotes the best fitness value. The best results are indicated in bold. From the results, it can be concluded that the performance of NaFA depends on the value of k. A small k yields better results than a large k for most of the functions. For f_1 and f_2, $k=1$ is the best choice, but it does not work for f_3, f_4, and f_{12}. Although $k=2$ yields promising solutions for f_4, it falls into local minima for f_3 and f_{11}. Further, $k=3$ and $k=4$ yield reasonable solutions for f_1–f_4, f_6, and f_{10}–f_{13}. Moreover, $k=3$ outperforms $k=4$ for these eight functions, except f_6 and f_{11}, which can be easily solved by NaFA with all k.

To determine the best k, the Friedman test was conducted [18]. Table 3 lists the mean ranks achieved by NaFA with different k values. It can be seen that $k=3$ achieves the best rank. Thus, $k=3$ is the best choice for the test suite. Therefore, k was set to 3 in the following experiments.

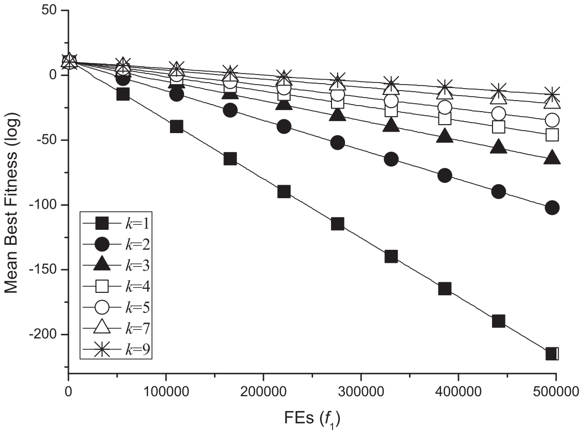
Fig. 3 shows some convergence graphs of NaFA with different k values for some selected functions. It can be seen that NaFa with $k=1$ converges faster than that with other k values in the initial stage during the search process. In the middle and last stages, $k=1$ does not work, i.e., for f_3, f_4, and f_9. Thus, a fixed k may not be a good choice. At different search stages, k should be dynamically adjusted to suit the current search.

4.3. Comparison of NaFA with other FA variants

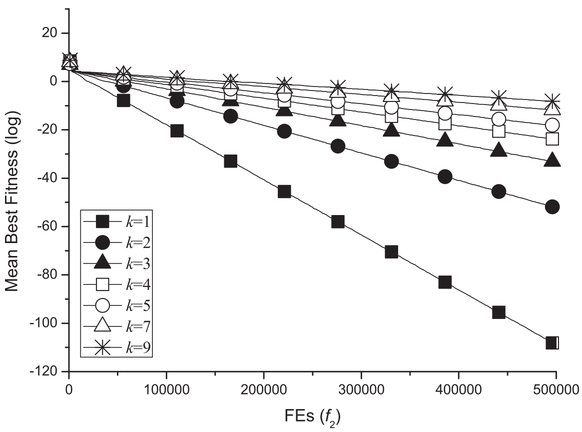
In this section, the performance of NaFA is compared with that of the standard FA and five other recently developed FA variants. The considered algorithms are listed below.

- Standard FA [46].
- Memetic FA (MFA) [12].
- FA with chaos (CFA) [17].
- Wise step strategy FA (WSSFA) [47].
- Variable step size FA (VSSFA) [48].
- FA with random attraction and Cauchy mutation (RaFA) [40].
- Our approach (NaFA).

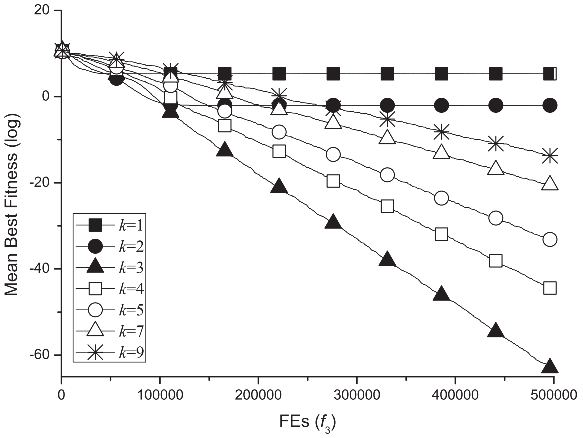
For all the algorithms, N and MAX_FEs were set to 20 and 5.0E+05, respectively. For the standard FA, the parameters α, β_0, and γ were set to 0.2, 1.0, and $\frac{1}{\Gamma^2}$, respectively. The parameter settings of WSSFA, VSSFA, CFA, and RaFA can be found



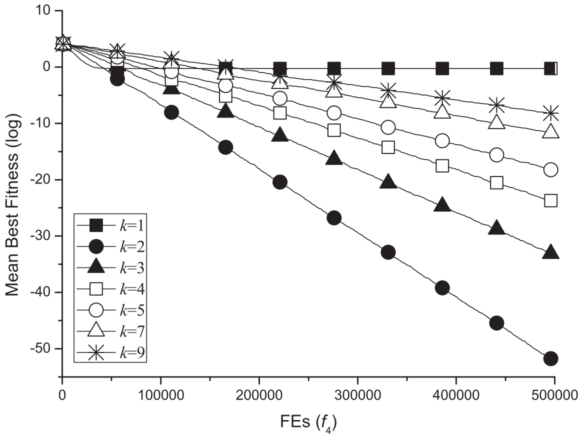
(a) Sphere (f_1)



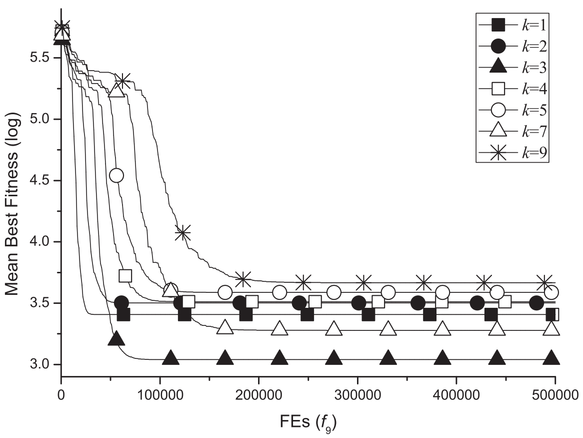
(b) Schwefel 2.22 (f_2)



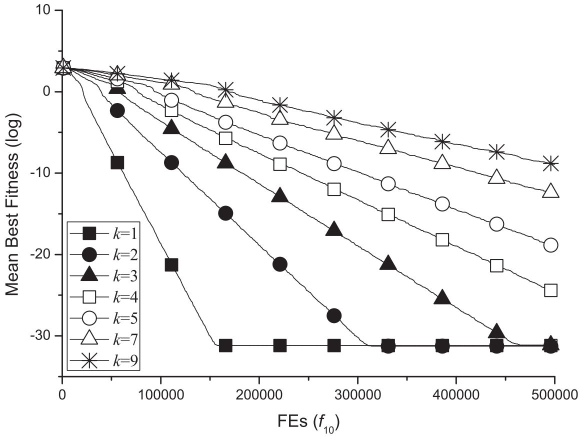
(c) Schwefel 1.2 (f_3)



(d) Schwefel 2.21 (f_4)



(e) Rastrigin (f_9)



(f) Ackley (f_{10})

Fig. 3. Convergence curves of NaFA with different k values on some selected functions.

Table 4
Mean best fitness values achieved by the standard FA, VSSFA, WSSFA, MFA, CFA, RaFA, and NaFA. The best results among the seven algorithms are indicated in bold.

Function	Standard FA Mean	VSSFA Mean	WSSFA Mean	MFA Mean	CFA Mean	RaFA Mean	NaFA Mean
f_1	5.67E−02	5.84E+04	6.34E+04	4.07E−06	3.27E−06	**5.36E−184**	4.43E−29
f_2	1.00E+00	1.13E+02	1.35E+02	9.16E−04	8.06E−04	8.76E−05	**2.98E−15**
f_3	1.23E−01	1.16E+05	1.10E+05	1.96E−05	1.24E−05	4.91E+02	**2.60E−28**
f_4	1.01E−01	8.18E+01	7.59E+01	8.69E−04	8.98E−04	2.43E+00	**3.43E−15**
f_5	8.42E+01	2.16E+08	2.49E+08	2.38E+01	**2.06E+01**	2.92E+01	2.39E+01
f_6	5.30E+03	5.48E+04	6.18E+04	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_7	6.74E−02	4.43E+01	3.24E−01	8.80E−02	9.03E−02	5.47E−02	**2.91E−02**
f_8	8.14E+03	1.07E+04	1.06E+04	6.09E+03	4.36E+03	**5.03E+02**	6.86E+03
f_9	4.49E+01	3.12E+02	3.61E+02	3.65E+01	5.27E+01	2.69E+01	**2.09E+01**
f_{10}	1.25E+01	2.03E+01	2.05E+01	4.49E−04	4.02E−04	3.61E−14	**3.02E−14**
f_{11}	2.94E−02	5.47E+02	6.09E+02	2.47E−03	7.91E−06	**0.00E+00**	**0.00E+00**
f_{12}	1.25E+01	3.99E+08	6.18E+08	1.02E−08	8.28E−09	4.50E−05	**1.36E−31**
f_{13}	5.28E+01	8.12E+08	9.13E+08	1.49E−07	1.69E−07	**8.25E−32**	2.13E−30
w/t/l	13/0/0	13/0/0	13/0/0	10/1/2	10/1/2	8/2/3	–

Table 5
Mean rank achieved by the Friedman test for the seven FA variants. The best rank is indicated in bold.

Algorithm	Mean rank
NaFA	**1.69**
RaFA	2.54
CFA	2.96
MFA	3.19
Standard FA	4.62
VSSFA	6.31
WSSFA	6.69

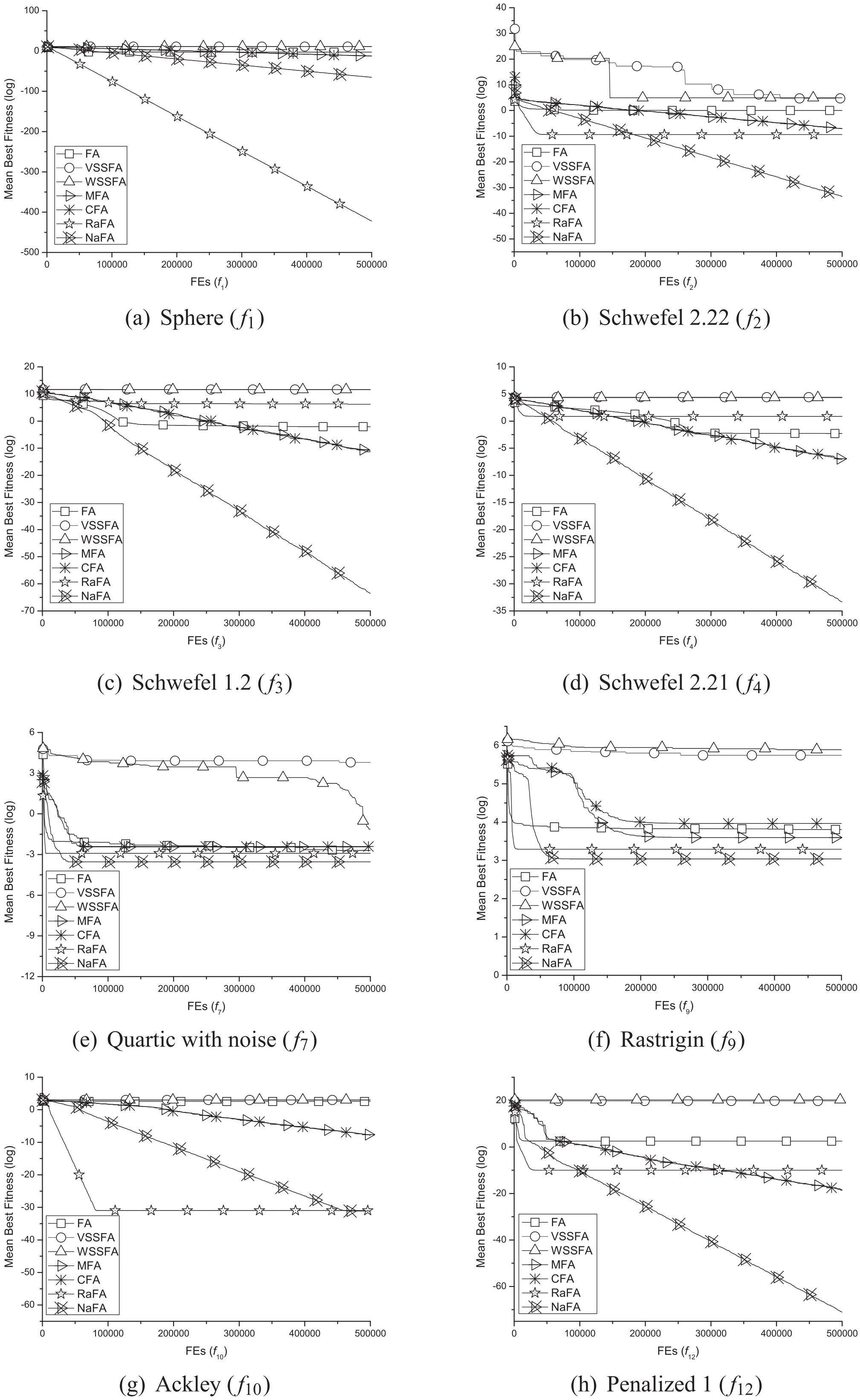
in the relevant literature [17,40,47,48]. For MFA and NaFA, the initial α, γ, β_{min}, and β_0 were set to 0.5, 1.0, 0.2, and 1.0, respectively. The parameter k used in NaFA was set to 3. Each algorithm was run 30 times per function, and the mean best fitness values were recorded.

Table 4 summarizes the computational results of the standard FA, VSSFA, WSSFA, MFA, CFA, RaFA, and NaFA, where “Mean” denotes the mean best fitness value. The comparison results of NaFA and the other six FA variants are indicated by $w/t/l$, which imply that NaFA wins in w functions, ties in t functions, and loses in l functions, compared with its competitors. The results show that VSSFA and WSSFA could barely achieve reasonable solutions for all the problems. However, VSSFA and WSSFA show good performance on some low-dimensional problems [47,48]. The dimension size seems to affect their performance significantly. The standard FA performs better than VSSFA and WSSFA, but it falls into local minima for all the functions. Compared to the three above-mentioned FA variants, NaFA achieves much better results. Our recent study has shown that the performance of FA is strongly dependent on its control parameters [41]. Thus, setting of the control parameters is a critical factor. NaFA outperformed MFA and CFA on 10 functions, while MFA and FA performed better than NaFA on 2 functions. For f_6, MFA, CFA, RaFA, and NaFA could converge to the global optimum. A comparison of RaFA and NaFA showed that RaFA outperformed NaFA for f_1 among the seven unimodal functions. The function f_1 is very simple and has no local minima. In RaFA, each firefly is attracted by one randomly selected brighter firefly at most. This is useful for accelerating the search for some simple unimodal functions. Among the multimodal functions, RaFA achieved much better solutions than NaFA for f_8. The main reason is that Cauchy mutation enables RaFA to escape from deep local minima. NaFA showed slightly better performance than RaFA for f_9 and f_{10}. Both of them could find the global optimum for f_{11}. For the penalized functions $f_1$2 and $f_1$3, RaFA fell into local minima for f_{12}, while NaFA achieved promising solutions for both functions.

Fig. 4 shows some convergence graphs of NaFA and the other six FA variants. As mentioned earlier, RaFA converges faster than the other algorithms in the initial stage of the search. In the middle and last stages, RaFA shows slow convergence for some complex functions. Both MFA and CFA achieved similar performance for most of the functions.

To compare all seven FA variants in terms of their performance on the test suite, the Friedman test was conducted [18]. Table 5 lists the mean ranks achieved by NaFA and the other six FAs. The best rank (with the smallest rank value) is indicated in bold. According to the mean rank values, all the FA variants are sorted in the following order: NaFA, RaFA, CFA, MFA, standard FA, VSSFA, and WSSFA. NaFA achieved the best rank. Thus, NaFA is the best algorithm among the seven FA variants.

Table 6 summarizes the results of Wilcoxon’s test between NaFA and the other FA variants. The p-values below 0.05 (the significance level) are indicated in bold. It can be seen that NaFA is significantly better than the standard FA, VSSFA, and



(a) Sphere (f_1)

(b) Schwefel 2.22 (f_2)

(c) Schwefel 1.2 (f_3)

(d) Schwefel 2.21 (f_4)

(e) Quartic with noise (f_7)

(f) Rastrigin (f_9)

(g) Ackley (f_{10})

(h) Penalized 1 (f_{12})

Fig. 4. Convergence graphs of the standard FA, VSSFA, WSSFA, MFA, CFA, RaFA, and NaFA for some selected functions.

Table 6
Wilcoxon test between NaFA and the other FA variants on the test suite. The p-values below 0.05 are indicated in bold.

NaFA vs.	p-values
Standard FA	**1.47E−03**
VSSFA	**1.47E−03**
WSSFA	**1.47E−03**
MFA	1.82E−01
CFA	1.82E−01
RaFA	9.12E−02

Table 7
Mean best fitness values achieved by MFA with different attraction models. The best results are indicated in bold.

Function	MFA + Ra		MFA + Na (NaFA)	
	Mean	Std Dev	Mean	Std Dev
f_1	**1.99E−193**	**0.00E+00**	4.43E−29	4.06E−30
f_2	3.84E−01	4.13E−01	**2.98E−15**	**2.80E−16**
f_3	2.25E+03	8.72E+02	**2.60E−28**	**2.37E−29**
f_4	3.01E+00	1.45E+00	**3.43E−15**	**2.89E−17**
f_5	1.74E+02	2.56E+02	**2.39E+01**	**8.96E−01**
f_6	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_7	4.49E−02	1.27E−02	**2.91E−02**	**1.56E−02**
f_8	7.03E+03	997.025	**6.86E+03**	**5.17E+02**
f_9	2.89E+01	9.95E−01	**2.09E+01**	**6.96E+00**
f_{10}	**2.55E−14**	**6.15E−15**	3.02E−14	8.94E−15
f_{11}	1.11E−16	1.11E−16	**0.00E+00**	**0.00E+00**
f_{12}	3.08E−02	5.33E−02	**1.36E−31**	**1.23E−32**
f_{13}	**7.84E−32**	**3.35E−32**	2.13E−30	4.62E−31

WSSFA. Although NaFA is not significantly better than MFA, CFA, and RaFA, it outperforms them in terms of their mean rank values.

4.4. Neighborhood attraction (Na) vs. random attraction (Ra)

Besides the full attraction model, there are two other attraction models: the random attraction (Ra) model and the proposed neighborhood attraction (Na) model. We have noted that the number of attractions plays an important role in the search process of FA. Each of Ra and Na has distinct characteristics. This section presents a comparative study of Ra and Na. Table 4 shows that NaFA outperforms RaFA on most of the test functions. However, RaFA not only employs the random attraction model but also uses Cauchy mutation. In fact, RaFA is a combination of MFA, random attraction, and Cauchy mutation. NaFA can be regarded as MFA + neighborhood attraction. For fair comparison, we replaced the neighborhood attraction model with the random attraction model in NaFA (MFA + random attraction).

In the experiment, both MFA + Ra and MFA + Na used the same parameter settings as those stated in Section 4.3. Table 7 summarizes the computational results of MFA with different attraction models, where “Mean” represents denotes the best fitness value. It can be seen that the random attraction model performs better than the neighborhood attraction model for f_1, f_4, and $f_1$3. Both of them obtain the same results for f_6. For the remaining functions, the neighborhood attraction model outperforms the random attraction model. Thus, the neighborhood attraction model is better than the random attraction model.

4.5. Different FA variants with neighborhood attraction

The above-mentioned experiments showed that our neighborhood attraction model can significantly improve the performance of MFA for most of the test functions. Besides MFA, there are several other FA variants. In this section, we extend the neighborhood attraction model to the standard FA and CFA. VSSFA and WSSFA could barely find reasonable solutions for the test suite; hence, we did not extend the neighborhood attraction model to them.

The considered algorithms are listed below.

- Standard FA [46].
- Standard FA + Na.
- FA with chaos (CFA) [17].
- CFA + Na.

Table 8
Mean best fitness values achieved by different FA variants with the neighborhood attraction model. The best comparison results are indicated in bold.

Function	Standard FA Mean	Standard FA + Na Mean	CFA Mean	CFA + Na Mean	MFA Mean	MFA + Na Mean
f_1	5.67E−02	**5.14E−02**	3.27E−06	**1.78E−32**	4.07E−06	**4.43E−29**
f_2	1.00E+00	**9.59E−01**	8.06E−04	**5.69E−17**	9.16E−04	**2.98E−15**
f_3	1.23E−01	**9.96E−02**	1.24E−05	**1.49E−31**	1.96E−05	**2.60E−28**
f_4	1.01E−01	**9.15E−02**	8.98E−04	**6.89E−17**	8.69E−04	**3.43E−15**
f_5	8.42E+01	**3.31E+01**	**2.06E+01**	2.47E+01	**2.38E+01**	2.39E+01
f_6	5.30E+03	**3.87E+03**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_7	6.74E−02	**3.61E−02**	9.03E−02	**4.04E−02**	8.80E−02	**2.91E−02**
f_8	8.14E+03	**7.59E+03**	**4.36E+03**	5.82E+03	**6.09E+03**	6.86E+03
f_9	4.49E+01	**3.90E+01**	5.27E+01	**3.58E+01**	3.65E+01	**2.09E+01**
f_{10}	1.25E+01	**1.10E+01**	4.02E−04	**4.68E−14**	4.49E−04	**3.02E−14**
f_{11}	2.94E−02	**7.20E−03**	7.91E−06	**0.00E+00**	2.47E−03	**0.00E+00**
f_{12}	1.25E+01	**1.87E+00**	8.28E−09	**1.86E−32**	1.02E−08	**1.36E−31**
f_{13}	5.28E+01	**2.71E+01**	1.69E−07	**3.32E−32**	1.49E−07	**2.13E−30**

- Memetic FA (MFA) [12].
- MFA + Na (NaFA).

These six algorithms used the same parameter settings as those described in Section 4.3. Table 8 summarizes the computational results of different FA variants with neighborhood attraction, where “Mean” represents denotes the best fitness value. For clear comparison, we included the results of MFA and NaFA. The results showed that although the neighborhood attraction model improves the performance of the standard FA for all the test functions, standard FA + Na cannot escape from local minima. The neighborhood attraction model could significantly improve the performance of CFA and MFA for f_1–f_4 and f_{10}–f_{13}, while it could not achieve any improvement for f_5 and f_8.

4.6. Discussion

To evaluate the performance of NaFA, we conducted four series of experiments: (1) investigation of neighborhood size, (2) comparison of the proposed approach with several recently developed FA variants, (3) performance comparison of different attraction models, and (4) extension of the proposed model to other FA variants.

In the neighborhood attraction model, each firefly is attracted by other brighter fireflies selected from its k-neighborhood. Thus, the neighborhood size plays an important role in balancing the number of attractions. This was investigated in the first experiment. The results showed that the performance of NaFA is significantly affected by the neighborhood size and that $k \in \{2, 3, 4\}$ yields better results than other values. Based on the Friedman test, $k = 3$ was selected as the best choice. In the second experiment, the performance of NaFA was compared with that of several recently developed FA variants. NaFA was found to outperform its competitors on most of the test functions. In the third experiment, the proposed neighborhood attraction model was compared with the random attraction model. For fair comparison, the two models were embedded into MFA. The results confirmed that the neighborhood attraction model is better than the random attraction model. In the last experiment, the proposed neighborhood attraction model was extended to FA, CFA, and MFA. The results showed that the neighborhood attraction model can also improve the performance of FA and CFA.

5. Conclusions

Attraction plays an important role in the search process of FA. It enables fireflies to move to other positions and find new candidate solutions. The standard FA employs a full attraction model, in which each firefly moves toward all other brighter fireflies in the entire population. Thus, there will be too many attractions, which may result in oscillations during the search process and high computational time complexity. To overcome these problems, this paper proposed a new FA variant called FA with neighborhood attraction (NaFA), which employs a neighborhood attraction (Na) model. In the new model, each firefly is attracted by all other brighter fireflies selected from its k-neighborhood rather than those from the entire population. To evaluate the performance of our approach, experiments were conducted on 13 well-known benchmark functions. The main findings can be summarized as follows.

- Too many or too few attractions may result in oscillations during the search process or premature convergence. The proposed neighborhood attraction model achieves a trade-off between full attraction and random attraction.
- The proposed approach can effectively reduce the computational time complexity and accelerate convergence.
- The number of attractions is strongly dependent on the size of the k-neighborhood. To choose the best k, we tested different neighborhood sizes. The results showed that $k = 3$ was the best setting for the test suite.
- The computational results showed that NaFA outperformed the standard FA, VSSFA, WSSFA, MFA, CFA, and RaFA on most of the test functions. Moreover, the neighborhood attraction model was shown to yield better solutions than the random attraction model.

- The proposed neighborhood attraction model was successfully embedded into other FA variants (FA, CFA, and MFA) to improve the accuracy of their solutions.

We observed that different search stages may need different neighborhood sizes. In the initial stage, $k = 1$ is a good choice. With increasing generations, k should be dynamically adjusted to suit the current search. The neighborhood model was constructed on the basis of a circle topology. Other topologies can be studied to obtain a better model. Owing to the neighborhood attraction model, the computational time complexity of FA is reduced. Thus, we can apply the new FA to some complex optimization problems. These will be investigated in the future.

Acknowledgment

This work was supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET), the National Natural Science Foundation of China (Nos. 61663028, 61663029 and 61261039), the Natural Science Foundation of Shanxi Province (No. 201601D011045), and the Science and Technology Plan Project of Jiangxi Provincial Education Department (Nos. KJLD13096 and GJJ151115).

References

[1] A. Baykasoğlu, F.B. Ozsoydan, An improved firefly algorithm for solving dynamic multidimensional knapsack problems, Expert Syst. Appl. 41 (2014) 3712–3725.
[2] J.Y. Chen, Q.Z. Lin, Z. Ji, Chaos-based multiobjective immune algorithm with a fine-grained selection mechanism, Soft comput. 15 (7) (2011) 1273–1288.
[3] B.J. Chen, H.Z. Shu, G. Coatrieux, G. Chen, X.M. Sun, J.L. Coatrieux, Color image analysis by quaternion-type moments, J. Math. Imaging Vis. 51 (1) (2015) 124–144.
[4] L.Z. Cui, G.H. Li, Q.Z. Lin, J.Y. Chen, N. Lu, Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations, Comput. Oper. Res. 67 (2016) 155–173.
[5] M.C. Cunha, J. Sousa, Water distribution network design optimization: simulated annealing approach, J. Water Resour. Plann. Manage. 125 (4) (1999) 215–221.
[6] S. Das, A. Abraham, U. Chakraborty, A. Konar, Differential evolution using a neighborhood-based mutation operator, IEEE Trans. Evol. Comput. 13 (3) (2009) 526–553.
[7] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: optimization by a colony of cooperating agents, IEEE transactions on systems, Man Cybern. Part B 26 (1996) 29–41.
[8] M. Ficco, C. Esposito, A. Napolitano, Calibrating indoor positioning systems with low efforts, IEEE Trans. Mob. Comput. 13 (4) (2014) 737–751.
[9] M. Ficco, C. Esposito, F. Palmieri, A. Castiglione, A coral-reefs and game theory-based approach for optimizing elastic cloud resource allocation, Future Gener. Comput. Syst. (2016). To be published, 10.1016/j.future.2016.05.025.
[10] J.I. Fister, I. Fister, X.S. Yang, J. Brest, A comprehensive review of firefly algorithms, Swarm Evol. Comput. 13 (2013a) 34–46.
[11] I. Fister, X.S. Yang, J. Brest, J.I. Fister, Modified firefly algorithm using quaternion representation, Expert Syst. Appl. 40 (18) (2013b) 7220–7230.
[12] J.I. Fister, X.S. Yang, I. Fister, J. Brest, Memetic firefly algorithm for combinatorial optimization, Bioinspired Optim. Methods Appl. (BIOMA) (2012) 1–14.
[13] A.P. Florence, V. Shanthi, A load balancing model using firefly algorithm in cloud computing, J. Comput. Sci. 10 (7) (2014) 1156–1165.
[14] Z.J. Fu, X.M. Sun, Q. Liu, L. Zhou, J.G. Shu, Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing, IEICE Trans. Commun. E98-B (1) (2015) 190–200.
[15] Z.L. Gaing, Particle swarm optimization to solving the economic dispatch considering the generator constraints, IEEE Trans. Power Syst. 18 (3) (2003) 1187–1195.
[16] A.H. Gandomi, X.S. Yang, A.H. Alavi, Mixed variable structural optimization using firefly algorithm, Comput. Struct. 89 (23–24) (2013a) 2325–2336.
[17] A.H. Gandomi, X.S. Yang, S. Talatahari, A.R. Alavi, Firefly algorithm with chaos, Commun. Nonlinear Sci. Numer. Simul. 18 (1) (2013b) 89–98.
[18] S. García, A. Fernández, J. Luengo, F. Herrera, 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) (2010) 2044–2064.
[19] B. Gu, V.S. Sheng, K.Y. Tay, W. Romano, S. Li, Incremental support vector learning for ordinal regression, IEEE Trans. Neural Netw. Learn. Syst. 26 (7) (2015) 1403–1416.
[20] P. Guo, J. Wang, B. Li, S. Lee, A variable threshold-value authentication architecture for wireless mesh networks, J. Internet Technol. 15 (6) (2014) 929–936.
[21] T. Hassanzadeh, H.R. Kanan, Fuzzy FA: a modified firefly algorithm, Appl. Artif. Intell. 28 (1) (2014) 47–65.
[22] T. Hassanzadeh, H. Vojodi, A.M.E. Moghadam, An image segmentation approach based on maximum variance intra-cluster method and firefly algorithm, in: The Seventh International Conference on Natural Computation (ICNC), 2011, pp. 1817–1821.
[23] C.F. Juang, A hybrid of genetic algorithm and particle swarm optimization for recurrent network design, IEEE Trans. Syst. Man Cybern. Part B 34 (2) (2004) 997–1006.
[24] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical report tr06, Computer Engineering Department, Erciyes University, Turkey, 2005.
[25] A. Kazem, E. Sharifi, F. Hussain, M. Saberi, O.K. Hussain, Support vector regression with chaos-based firefly algorithm for stock market price forecasting, Appl. Soft Comput. 13 (2) (2013) 947–958.
[26] J. Kennedy, R.C. Eberhart, Particle swarm optimization, Proc. IEEE Int. Conf. Neural Netw. (1995) 1942–1948.
[27] J. Li, X.L. Li, B. Yang, X.M. Sun, Segmentation-based image copy-move forgery detection scheme, IEEE Trans. Inf. Forensics Secur. 10 (3) (2015) 507–518.
[28] Z.P. Liang, J.T. Sun, Q.Z. Lin, Z.H. Du, J.Y. Chen, Z. Ming, A novel multiple rule sets data classification algorithm based on ant colony algorithm, Appl. Soft Comput. 38 (2016) 1000–1011.
[29] Q.Z. Lin, J.Y. Chen, A novel micro-population immune multiobjective optimization algorithm, Comput. Oper. Res. 40 (6) (2013) 1590–1601.
[30] Q.Z. Lin, Q.L. Zhu, P.Z. Huang, J.Y. Chen, Z. Ming, J.P. Yu, A novel hybrid multi-objective immune algorithm with adaptive differential evolution, Comput. Oper. Res. 65 (2015) 95–111.
[31] T.H. Ma, J.J. Zhou, M.L. Tang, Y. Tian, A. Al-dhelaan, M. Al-rodhann, S. Lee, Social network and tag sources based augmenting collaborative recommender system, IEICE Trans. Inf. Syst. E98-D (4) (2015) 902–910.
[32] M.K. Marichelvam, T. Prabaharan, X.S. Yang, A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems, IEEE Trans. Evol. Comput. 18 (2) (2014) 301–305.
[33] A. Rahmani, S.A. MirHassani, A hybrid firefly-genetic algorithm for the capacitated facility location problem, Inf. Sci. 283 (2014) 70–78.
[34] Y.J. Ren, J. Shen, J. Wang, J. Han, S. Lee, Mutual verifiable provable data auditing in public cloud storage, J. Internet Technol. 16 (2) (2015) 317–323.
[35] E. Saraç, S.A. Özel, Web page classification using firefly optimization, IEEE Int. Symp. Innovations Intell. Syst. Appl. (INISTA) (2013) 1–5.

[36] S. Sharma, S. Kumar, B. Singh, Routing in wireless mesh networks: three new nature inspired approaches, Wirel. Pers. Commun. 83 (4) (2015) 3157–3179.
[37] J. Shen, H.W. Tan, J. Wang, J.W. Wang, S. Lee, A novel routing protocol providing good transmission reliability in underwater sensor networks, J. Internet Technol. 16 (1) (2015) 171–178.
[38] F. Shomalnasab, M. Sadeghzadeh, M. Esmaeilpour, An optimal similarity measure for collaborative filtering using firefly algorithm, J. Adv. Comput. Res. 5 (3) (2014) 101–111.
[39] O.P. Verma, D. Aggarwal, T. Patodi, Opposition and dimensional based modified firefly algorithm, Expert Syst. Appl. 44 (2016) 168–176.
[40] H. Wang, W.J. Wang, H. Sun, S. Rahnamayan, Firefly algorithm with random attraction, Int. J. Bio Inspired Comput. 8 (1) (2016a) 33–41.
[41] H. Wang, X.Y. Zhou, H. Sun, X. Yu, J. Zhao, H. Zhang, L.Z. Cui, Firefly algorithm with adaptive control parameters, Soft comput. (2016b), doi:10.1007/s00500-016-2104-3.
[42] X.Z. Wen, L. Shao, Y. Xue, W. Fang, A rapid learning algorithm for vehicle classification, Inf. Sci. 295 (2015) 395–406.
[43] Z.H. Xia, X.H. Wang, X.M. Sun, Q. Wang, A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data, IEEE Trans. Parallel Distrib. Syst. 27 (2) (2015) 340–352.
[44] Z.X. Xu, A. Unveren, A. Acan, Probability collectives hybridised with differential evolution for global optimisation, Int. J. Bio Inspired Comput. 8 (3) (2016) 133–153.
[45] M. Xu, G.Z. Liu, A multipopulation firefly algorithm for correlated data routing in underwater wireless sensor networks, Int. J. Distrib. Sens. Netw. (2013), doi:10.1155/2013/865154.
[46] X.S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, UK, 2008.
[47] S.H. Yu, S.B. Su, Q.P. Lu, L. Huang, A novel wise step strategy for firefly algorithm, Int. J. Comput. Math. 91 (12) (2014) 2507–2513.
[48] S.H. Yu, S.L. Zhu, Y. Ma, D.M. Mao, A variable step size firefly algorithm for numerical optimization, Appl.Math. Comput. 263 (2015) 214–220.
[49] Y.H. Zheng, B. Jeon, D.H. Xu, Q.M.J. Wu, H. Zhang, Image segmentation by generalized hierarchical fuzzy c-means algorithm, J. Intell. Fuzzy Syst. 28 (2) (2015) 961–973.