ICIC Express Letters ICIC International ©2016 ISSN 1881-803X
Volume **10**, Number **6**, June **2016** pp. 1279–1284

UNCONSTRAINED FUNCTION OPTIMIZATION AND PARAMETERS PERFORMANCE ANALYSIS BASED ON FIREWORKS ALGORITHM

Yufeng Sun[1] and Jiesheng Wang[1,2,*]

[1]School of Electronic and Information Engineering
[2]National Financial Security and System Equipment Engineering Research Center
University of Science and Technology Liaoning
No. 185, Qianshan Road, Anshan 114051, P. R. China
syf95411@126.com; *Corresponding author: wang_jiesheng@126.com

Received November 2015; accepted February 2016

Abstract. *Fireworks algorithm (FA) is a new parallel diffuse optimization algorithm to simulate the fireworks explosion phenomenon, which realizes the balance between global exploration and local searching by means of adjusting the explosion mode of fireworks bombs. Through generating a certain number of fireworks shells in the function searching scope, the bombing operation is carried out for each fireworks bomb, which realizes the random search on the certain neighborhood scope of the original fireworks (burst points) by the explosion Mars. At the same time, FA with different parameters is carried out of the simulation experiments to verify the influence of different parameters on the function optimization performance. Simulation results show that the initial seed number of fireworks algorithm has influence on the solution accuracy and optimization ability.*
Keywords: Fireworks algorithm, Function optimization, Performance comparison

1. **Introduction.** The nature of function optimization problem is to find the optimal solution of an objective function through iterative [1]. The function features are usually described as continuous, discrete, linear, non-linear, convex function, etc. In that the constraint function optimization problem can be converted into unconstrained problem by using the designed special operators and penalty functions to make solution always feasible, the unconstrained function optimization problem is the main research focus. The swarm intelligent optimization algorithms [2] are a kind of random search algorithm to simulate the biological population evolution and evolution, which solve the complex global optimization problems through individual cooperation and competition between species, and are applied in many fields, such as multi-objective optimization, data mining, network routing, signal processing, and pattern recognition. The typical swarm intelligence optimization algorithms include ant colony optimization (ACO) algorithm [3], genetic algorithm (GA) [4], particle swarm optimization (PSO) algorithm [5], and artificial bee colony (ABC) algorithm [6].

Fireworks algorithm (FA) is a new swarm intelligence algorithm proposed by Tan and Zhu in 2010 [7], which has excellent optimization performance and arouses widespread concern in the world [8]. FA has been applied for solving many practical optimization problems [9,10]. Janecek and Tan [9] used FA together with particle swarm optimization (PSO), genetic algorithms (GA), differential evolution (DE), and fish school search for improving the initialization of non-negative matrix factorization (NMF). Bureerat [10] compared twelve different optimization algorithms on 35 benchmark functions with different dimensions ranging from 2 to 30. However, on the other hand, the research on the performance analysis of algorithm parameters of FA has not been carried out. In this paper, an unconstrained function optimization problem is solved by FA and the parameters performance analysis is carried out. The paper is organized as follows. In Section 2, the fireworks algorithm is introduced. Then, the simulation results are described in Section 3. Finally, the conclusion is illustrated in the last part.

2. Fireworks Algorithm. Fireworks algorithm (FA) is a new parallel diffuse optimization algorithm to simulate the fireworks explosion phenomenon, which realizes the balance between global exploration and local searching by means of adjusting the explosion mode of fireworks bombs. Its algorithm procedure is shown in Figure 1. Through generating a certain number of fireworks shells in the function searching scope, the bombing operation is carried out for each fireworks bomb, which realizes the random search on the certain neighborhood scope of the original fireworks (burst points) by the explosion Mars.

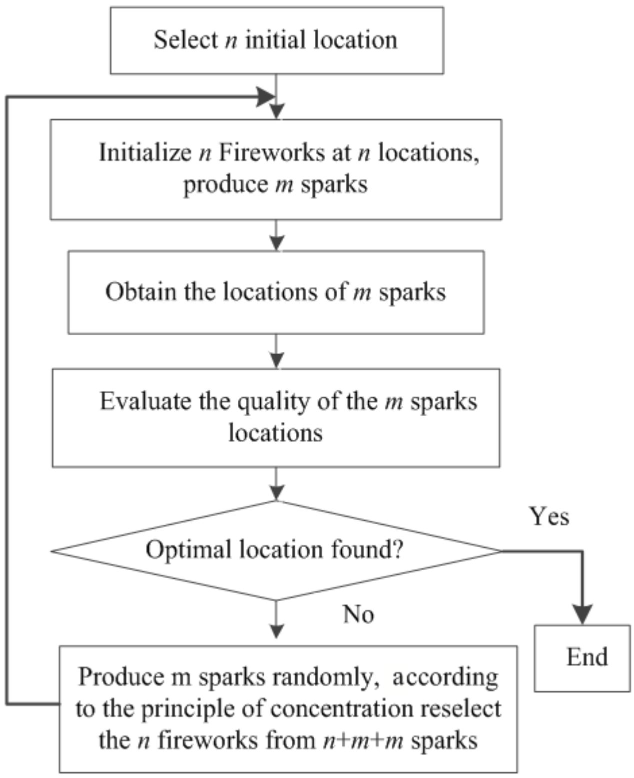


FIGURE 1. Algorithm procedure of FA

2.1. Description of FA. FA is to solve a kind of optimization problem $\min f(x)$, $x_{\min} < x < x_{\max}$, where $x = x_1, x_2, \ldots, x_d$ represents a potential solution. So the number of sparks produced by each firework x_i is described as follows.

$$s_i = m \cdot \frac{y_{\max} - f(x_i) + \varepsilon}{\sum_{i=1}^{n}(y_{\max} - f(x_i)) + \varepsilon} \tag{1}$$

where m is to control the total number of sparks produced by n fireworks.

$$y_{\max} = \max(f(x_i)) \ \ (i = 1, 2, \ldots, n) \tag{2}$$

where $y_{\max}$ is the maximum of objective functions under the worst case in n fireworks.

In order to avoid worse case under bad firework explosion, its scope s_i is defined as:

$$\hat{s}_i = \begin{cases} round(a \cdot m) & \text{if } s_i < am \\ round(b \cdot m) & \text{if } s_i > bm \\ round(s_i) & \text{otherwise } a < b < 1 \end{cases} \tag{3}$$

where a and b are fixed constant parameters.

The explosion amplitude of each firework is defined as follows:

$$A_i = A \cdot \frac{f(x_i) - y_{\min} + \varepsilon}{\sum_{i=1}^{n}(f(x_i) - y_{\min}) + \varepsilon} \tag{4}$$

where A represents the maximum explosion amplitude, $y_{\min} = \min(f(x_i))$ $(i = 1, 2, \ldots, n)$ is the minimum objective function value in n fireworks.

During the explosion process, spark may be affected by any direction (dimension). In FA, the number of arbitrary affected directions is defined as:

$$z = round(d \cdot rand(0, 1)) \tag{5}$$

where, d is the dimension of position x, $rand(0, 1)$ is a uniform distribution on the interval $[0, 1]$.

2.2. **Determination of spark locations.** Spark location of firework x_i can be obtained by Algorithm 1. By imitating the explosion process, the position $\hat{x}_j$ of a spark is produced. Then, if the obtained position is beyond the potential space, it is changed into the potential space by Algorithm 1.

Algorithm 1: Obtain the spark location
Initialize position $x_j = x_i$;
$z = round(d \cdot rand(0, 1))$;
Randomly select $\hat{x}_j$ with Z dimension and calculate shift $h = A_i \cdot rand(-1, 1)$;
For each dimension $x_k^j \in \{\textit{pre-selected z dimension} - x_j\}$
Set $x_k^j = x_k^j + h$. If $x_k^j < x_k^{\min}$ *or* $x_k^j > x_k^{\max}$
Convert x_k^j to the potential space $x_k^j = x_k^{\min} + x_k^j \bullet (x_k^{\max} - x_k^{\min})$;
for each dimension $x_k^j \in \{\textit{pre-selected z of } x_j\}$
$x_k^j = x_k^j + h$;
$\hat{x}_k^j = \hat{x}_k^j + h$
if $x_k^j < x_k^{\min}$ *or* $x_k^j > x_k^{\max}$ *then*
map x_k^j *to the potential space*;
end if
end for

In order to keep the spark diversity, a Gauss explosion method shown in Algorithm 2 is adopted to produce sparks. $\hat{m}$ sparks are produced in each Gauss explosion.

Algorithm 2: Obtain a certain spark position
Initialize the spark position $\hat{x}_j = x_i$;
z $= round(d \cdot round(0, 1))$;
Randomly select $\hat{x}_j$ with z dimension;
Calculate the coefficient of Gauss explosion g $= Gaussian(1, 1)$;
for each dimension $x_k^j \in \{\textit{pre-selected z of } x_j\}$
$\hat{x}_k^j = \hat{x}_k^j \cdot$ g;
if $x_k^j < x_k^{\min}$ *or* $x_k^j > x_k^{\max}$ *then*
map x_k^j *to the potential space*;
end if
end for

2.3. **Selection of explosion positions.** At the beginning of each explosion, n location should be chosen to realize the fireworks explosion. In FA, the best location x^* according to the best objective function $f(x^*)$ is retained for the next explosion. Since then, the selection of $n-1$ position is based on the distance with other positions to keep the diversity of sparks. In general, the distance between position x_i and other positions is calculated as follows:

$$R(x_i) = \sum_{j\in k} d(x_i, x_j) - \sum_{j\in k} x_i - x_j \tag{6}$$

where, k is the current position set of all fireworks and sparks.

A selection probability of position x_i is defined as follows:

$$p(x_i) = \frac{R(x_i)}{\sum_{j\in k} R(x_j)} \tag{7}$$

Algorithm 3: Construction of FA
Initialize n positions of fireworks randomly;
while Stop criterion is false
  Detonate n fireworks in n positions respectively;
  for Each firework x_i
    Calculate the spark number $\hat{s}_i$ produced by those fireworks by Equation (3);
    Obtain the position of spark s_i of firework x_i based on Algorithm 1;
  end for
  for $k = 1 : \hat{m}$
    Select a firework x_i randomly;
    Produce a certain spark of the above firework based on Algorithm 2;
    Save the best position to the next explosion;
    Based on the given probability by Equation (7), select $n-1$ position randomly from two sparks and the current firework.
end while

In FA, each generation carries out about $n+m+\hat{m}$ function estimations. If the optimum of a certain function can be found in generation T, the complexity of FA is $o*(n+m+\hat{m})$.

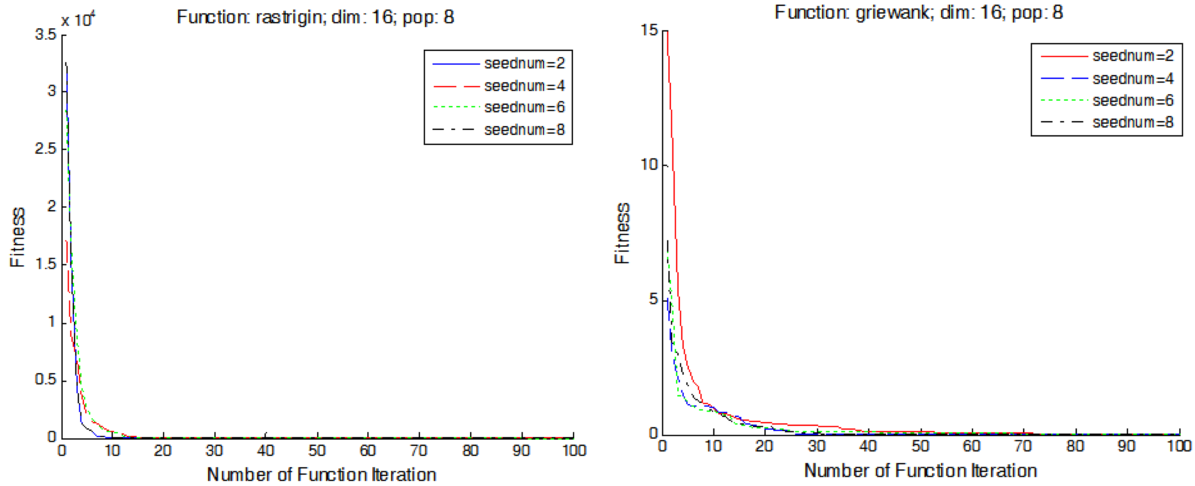
3. **Simulation Results.** In order to discuss the performance influence of FA parameters, two benchmark functions shown in Table 1 are adopted to carry on the simulation experiment. The feasible range of these unconstrained function minimization problems is set as $[-100, 100]$ and the dimension D is 30.

Table 1. Four benchmark functions used in the simulation experiments

Function	Expression
Rastrigin	$F_1 = \sum_{i=1}^{D-1} \left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right)$
Griewank	$F_3 = 1 + \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right)$

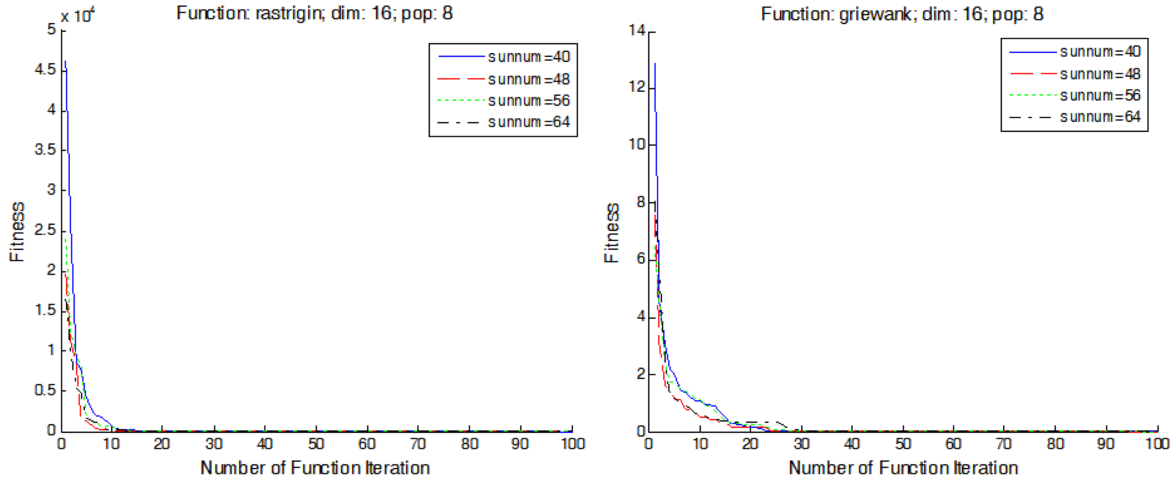
When the number of FA seeds is set as 2, 4, 6 and 8, respectively, the optimized curves of various functions are shown in Figures 2(a) and 2(b). It can be seen from Figure 2 that the convergence of all function is worse under 2 seeds and 4 seeds. The smaller the number of seeds is, the worse the convergence performance of the FA is. However, when the number of seeds reaches a certain value, the convergence does not change with its increase.

When the subalgebra number of FAs is set as 40, 48, 56 and 64, respectively, the optimized curves of various functions are shown in Figures 3(a) and 3(b). It can be seen form Figure 3 that the subalgebra number of FAs has little effect on the optimization performance despite of Ackley function.



(a) Rastrigin function (b) Griewank function

Figure 2. Curves under different number of FA seeds for two benchmark functions



(a) Rastrigin function (b) Griewank function

Figure 3. Curves under different subalgebra number of FAs for two benchmark functions

4. **Conclusions and Future Work.** Fireworks algorithm is adopted to solve the unconstrained function optimization problem and the parameters performance analysis is carried out. Simulation results show the validity of the proposed method. In future, this method could be extended to deal with the other optimization problems.

Acknowledgment. This work is partially supported by the National Key Technologies R & D Program of China (Grant No. 2014BAF05B01), the Project by National Natural Science Foundation of China (Grant No. 21576127), the Program for Liaoning Excellent Talents in University (Grant No. LR2014008), the Project by Liaoning Provincial Natural Science Foundation of China (Grant No. 2014020177), the Program for Research Special Foundation of University of Science and Technology of Liaoning (Grant No. 2015TD04) and the Opening Project of National Financial Security and System Equipment Engineering Research Center (Grant No. USTLKEC201401).

REFERENCES

[1] Y. Ren and Y. Wu, An efficient algorithm for high-dimensional function optimization, *Soft Computing*, vol.17, no.6, pp.995-1004, 2013.
[2] Z. Yuan, M. A. M. de Oca, M. Birattari and T. Stützle, Continuous optimization algorithms for tuning real and integer parameters of swarm intelligence algorithms, *Swarm Intelligence*, vol.6, no.1, pp.49-75, 2012.
[3] B. C. Mohan and R. Baskaran, A survey: Ant colony optimization based recent research and implementation on several engineering domain, *Expert Systems with Applications*, vol.39, no.4, pp.4618-4627, 2012.
[4] E. Vallada and R. Ruiz, A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times, *European Journal of Operational Research*, vol.211, no.3, pp.612-622, 2011.
[5] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *Proc. of IEEE International Conference on Neural Networks*, Piscataway, New Jersey, pp.1942-1948, 1995.
[6] D. Karaboga and B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, vol.39, no.3, pp.459-471, 2007.
[7] Y. Tan and Y. C. Zhu, Fireworks algorithms for optimization, *LNCS*, vol.6145, pp.355-364, 2010.
[8] H. Gao and M. Diao, Cultural firework algorithm and its application for digital filters design, *International Journal of Modelling, Identification and Control*, vol.14, no.4, pp.324-331, 2011.
[9] A. Janecek and Y. Tan, Iterative improvement of the multiplicative update NMF algorithm using nature-inspired optimization, *The 7th International Conference on Natural Computation*, pp.1668-1672, 2011.
[10] S. Bureerat, Hybrid population-based incremental learning using real codes, *Lecture Notes in Computer Science*, pp.379-391, 2011.