Fireworks Algorithm Based on Opposition-Based Learning and Quantum Optimization Strategy

JIN Hongbin¹, LI Hao¹, MA Yanyan¹, FANG Xi²

¹ Air Force Early Warning Academy, Wuhan, China
² School of Science, Wuhan University of Technology, Wuhan, China

jhb760817@sina.com, afeu_li@163.com, ma163email@163.com, fangxi@whut.edu.cn

Abstract. Aiming at the bottleneck of optimization performance and slow convergence speed of fireworks algorithm, a new fireworks algorithm (FWA) is proposed by integrating Opposition-Based Learning and Quantum Optimization strategy (OQFWA). The new algorithm optimizes the original fireworks algorithm in the aspects of the selection of sparks and the local improvement of the optimal individuals, which effectively improves the convergence accuracy and speed of the algorithm. The simulation results of extremum optimization of typical test functions with different characteristics show that the fireworks algorithm, which integrates the Opposition-Based Learning and Quantum Optimization strategy, and has good optimization performance.

1. Introduction

Firework algorithm (FWA) originated from the simulation of the physical process of fireworks explosion, which was proposed by Tan in 2010 [1]. The algorithm follows the following principles: the smaller the fitness function value of fireworks is, the smaller the explosion amplitude is, at the same time, the more sparks are generated; on the contrary, the larger the fitness function value is, the larger the explosion amplitude is and the less sparks are generated. The unique search mechanism of fireworks algorithm makes it have strong ability of local search and global search self-regulation. Once proposed, it has been widely concerned and applied in image recognition, filter design, distribution network reconstruction and optimization and other fields. At the same time, many scholars put forward improved algorithms for specific problems. Among them, Zheng made a detailed analysis on the explosion operator, Gaussian mutation operator, selection strategy and mapping rule of fireworks algorithm, improved the existing defects and proposed Enhanced Fireworks Algorithm (EFWA) [2]. Then, Zheng calculated the explosion radius of fireworks dynamically according to the dynamic change of spark fitness value, and proposed the dynamic search fireworks algorithm (dynFWA) [3]. Li determined the explosion radius of fireworks according to the distance between the optimal individual and the specific individual, and proposed an Adaptive Fireworks Algorithm (AFWA) [4]. However, fireworks algorithm is still in the development stage, so further research is needed to improve the performance of the algorithm and expand the scope of engineering application.

Based on the above discussion, Opposition-Based Quantum Fireworks Algorithm (OQFWA) is proposed in this paper by integrating Opposition-Based Learning and Quantum Optimization strategy into the fireworks algorithm. The algorithm improves the original fireworks algorithm in the aspects of
spark selection and local improvement of the optimal individual, effectively improving the convergence accuracy and speed of the algorithm.

2. Basic Principle of Fireworks Algorithm
Similar to other bionic intelligent algorithms, fireworks algorithm is also an intelligent algorithm based on population optimization. In the iterative process of the algorithm, the explosive operator, Gaussian mutation operator, mapping rule and selection strategy are used to iterate until the termination condition is reached [5-7].

2.1. Explosive operator
In the fireworks algorithm, the explosion of the $i^{th}$ fireworks $X_i$ produces $S_i$ sparks. The formula for $S_i$ is as follows:

$$S_i = m \frac{Y_{worst} - f(X_i) + \xi}{\sum_{i=1}^{N} (Y_{worst} - f(X_i)) + \xi}$$  \hspace{1cm} (1)

where $m$ is a constant, which is used to adjust the number of sparks produced by explosion; Parameter $\xi$ is used to prevent the denominator from becoming zero. Function $f(X_i)$ represents the fitness value for $X_i$, $Y_{worst}$ is the current worst fitness value. At the same time, in order to prevent the fireworks with better adaptability from generating too many sparks or the fireworks with poor adaptability from generating too few sparks, the following formula shall be used for correction:

$$S_i = \begin{cases} \text{round}(am), & S_i < am \\ \text{round}(bm), & S_i > bm, a < b < 1 \\ \text{round}(S_i), & \text{otherwise} \end{cases}$$ \hspace{1cm} (2)

where $\text{round}(\bullet)$ is the rounding function, $a$ and $b$ are constants; $A_i$ is the explosion radius of $X_i$, calculated by the following formula:

$$A_i = \hat{A} \frac{f(X_i) - Y_{best} + \xi}{\sum_{i=1}^{N} (f(X_i) - Y_{best}) + \xi}$$ \hspace{1cm} (3)

where $\hat{A}$ is the set maximum explosion radius.

the Z-dimensional coordinates are randomly selected for updating from the D-dimensional objective function, then the k-dimensional coordinates of the $j^{th}$ spark produced by the explosion of fireworks $X_j$ are $X_j^k$, where $j=1,2,\ldots,S_i$ and $k=1,2,\ldots,Z$, and the updating formula is as follows:

$$X_j^k = X_j^k + A_i \text{rand}(-1,1)$$ \hspace{1cm} (4)

where $\text{rand}(-1,1)$ is a random number generated independently within the range of $[0,1]$.

2.2. Gaussian mutation operator
The Gaussian mutation operator is introduced in order to increase the diversity of population. If the algorithm is preset to generate $M$ Gaussian mutation sparks, then the update formula of the $k^{th}$ coordinate of the $j^{th}$ spark in $M$ $X_j^k$ is as follows:

$$X_j^k = X_j^k + \text{Gaussian}(1,1)$$ \hspace{1cm} (5)

$\text{Gaussian}(1,1)$ is the Gaussian mutation operator.

2.3. Mapping rule
When the explosion sparks and Gaussian mutation sparks are generated, the sparks may exceed the boundary of feasible region, so they need to be corrected by mapping rule. When the sparks are beyond the value range, they can be mapped to the new positions by the following formula:


\[ X_j^k = X_{\text{min}}^k + \left| X_{\text{max}}^k - X_{\text{min}}^k \right| \text{mod}(X_{\text{max}}^k - X_{\text{min}}^k) \]  

(6)

where \( X_{\text{max}}^k \) and \( X_{\text{min}}^k \) are the upper and lower bounds of the value range of the \( k \)th dimension of the \( i \)th spark.

### 2.4. Selection strategy

In order to transfer the excellent information from the population to the next generation, the algorithm adopts the elitist retention strategy, that is, after the algorithm produces the explosion sparks and the Gaussian mutation sparks, it compares the fitness values of all individuals, among which the individuals with the best fitness value are automatically retained to the next generation, and the rest are selected by the way of roulette. For an individual \( X_i \), the probability of being selected is:

\[ p(X_i) = \frac{R(X_i)}{\sum_{j=1}^{N} R(X_j)} \]  

(7)

where \( R(X_j) \) is the sum of the distance between individual \( X_i \) and other individuals, calculated by the following formula:

\[ R(X_i) = \sum_{j=1}^{N} d(X_i, X_j) = \sum_{j=1}^{N} |X_i - X_j| \]  

(8)

where \( d(X_i, X_j) \) is the Euclidean distance between \( X_i \) and \( X_j \). \( N \) is the total number of sparks generated by explosion operator and Gaussian mutation operator.

### 3. Opposition-Based Learning and Quantum Optimization strategy

#### 3.1. Opposition-Based Learning

Opposition-Based Learning (OBL) was first proposed by Tizhoosh in 2005. Its main idea is to select the best solution to use the current solution and its opposite solution. It is an effective method to enhance the optimization ability and accelerate the convergence of the optimization algorithm [8-11].

Opposite Number: Let \( x \in [a, b] \) be a real number. The opposite of \( x \) is defined by:

\[ \bar{x} = a + b - x \]  

(9)

Opposite point: Let \( X = (x_1, x_2, \ldots, x_D) \) be a point in a D-dimensional solution space, and \( x_i \in [a, b] \). Its opposite point \( \bar{X} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_D) \) is defined by:

\[ \bar{x}_i = a_i + b_i - x_i \]  

(10)

where \( i = 1, 2, \ldots, D \).

According to the definition of opposite point, the algorithm idea of OBL is as follows: Let \( P(x_1, x_2, \ldots, x_D) \) be a point in D-dimensional space, where \( x_i \in [a, b], i = 1, 2, \ldots, D \). Then \( P \) represents an individual in the population of the optimization algorithm. According to formula (10), the opposite point of \( P \) can be defined as \( P(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_D) \). Fitness function \( f(x) \) is used to evaluate the fitness of individuals. For \( i = 1, 2, \ldots, D \), if \( f(\bar{x}_i) \leq f(x_i) \) holds, then \( P \) will be replaced by its opposite point \( \bar{P} \), otherwise it will remain unchanged.

#### 3.2. Quantum Optimization

In the standard fireworks algorithm, the position of sparks are updated according to formula (4) and (5), while in the fireworks algorithm with quantum strategy, the positions of sparks are updated according to the following formula [12]:

\[ X_j^k = X_{\text{min}}^k + \left| X_{\text{max}}^k - X_{\text{min}}^k \right| \text{mod}(X_{\text{max}}^k - X_{\text{min}}^k) \]  

(6)
where $\varphi = \text{rand}(0,1)$ and $u = \text{rand}(0,1)$ are random numbers generated independently within the range of $[0,1]$; $P^g_i$ is the global best spark position; $P^t_i$ is a random point between $X^t_i$ and $P^g_i$; $\alpha$ is the quantum selected parameter, i.e. expansion or contraction coefficient: when $\text{rand} > 0.5$, $\alpha$ is between $[0,1]$, whereas $\alpha$ is between $[-1,0]$.

3.3. Opposition-Based Quantum Fireworks Algorithm

In conclusion, the Opposition-Based Learning and Quantum Optimization strategy are integrated into fireworks algorithm, and Opposition-Based Quantum Fireworks Algorithm (OQFWA) is proposed in this paper in order to improve the efficiency of the algorithm in solving the function optimization problem. The algorithm is designed as follows:

\begin{verbatim}
Begin
1 Initialize the position of fireworks, calculate the fitness value of each fireworks; calculate the number of sparks generated by each fireworks explosion according to formula (3), and calculate the explosion radius of each fireworks according to formula (1);
2 while $t = 1$ to $\text{maxIter}$ do
3   for $i = 1$ to $N$
4       Update the spark position according to (4);
5       Gaussian mutation sparks are generated according to (5);
6       The opposite points of all sparks are calculated according to (10), and the fitness values of all sparks and the opposite points are calculated. The sparks with better fitness values are selected from the original sparks and opposite sparks to form a new group
7       Update the positions of sparks according to (11);
8       Generate the next generation of fireworks according to the selection strategy;
9   end for
10  $t = t + 1$
11 end while
12 return the optima
\end{verbatim}

4. Simulation Analysis

Six typical test functions with different characteristics are selected to verify the performance of OQFWA, and the BEST, MEAN, WORST and STD are used as the evaluation criteria for the algorithms. FWA, OFWA and OQFWA algorithms were used to perform 100 independent operations under maximum number of iterations $T_{\text{max}} = 100$ on the six test functions. The expressions of six test functions and their parameters are shown in Table 1, and the parameter settings of the four algorithms are shown in Table 2.

Figure 1 shows the optimization results of the four algorithms, and table 3 shows the results obtained by the four algorithms. It can be seen that Compared with FWA and OFWA algorithm, OQFWA algorithm has better BEST, MEAN, WORAT value and standard deviation, which illustrates that the accuracy and stability of OQFWA algorithm have been improved. It can be seen from Figure 1 that OQFWA algorithm has better ability of continuous optimization and good pioneering. After continuous optimization, it is closer to the optimal solution and can converge earlier. In summary, the performance of the algorithm is greatly improved after integrating Opposition-Based Learning and Quantum Optimization strategy.
Table 1. Information of test functions.

| Functions          | Expressions                                                                 | Range       | f_{min} |
|--------------------|------------------------------------------------------------------------------|-------------|---------|
| Sumsquares         | \( F_i(x) = \sum_{i=1}^{d} x_i^2 \)                                         | \([-10,10]^{10}\) | 0       |
| Schaffer           | \( F_i(x) = 0.5 + \frac{\sin^2(x_i^2 + x_j^2) - 0.5}{[1 + 0.001(x_i^2 + x_j^2)]^2} \) | \([-100,100]^2\) | 0       |
| Griewank           | \( F_i(x) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \) | \([-600,600]^4\) | 0       |
| Rastrigin          | \( F_i(x) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)] \)               | \([-5.12,5.12]^{30}\) | 0       |
| Six hump camel     | \( F_i(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \) | \([-3,3]^2\) | -1.0316 |
| Ackley             | \( F_i(x) = -20\exp\left(-0.2\sqrt{\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d} \cos(2\pi x_i)\right) + 20 + e \) | \([-32.768,32.768]^4\) | 0       |

Table 2. Parameter setting of algorithms.

| Algorithms   | Parameter setting                                                                 |
|--------------|----------------------------------------------------------------------------------|
| FWA          | Number of fireworks \( N = 5 \), Number of Gaussian mutation sparks \( N_g = 5 \), Constant \( m = 50 \). Parameter \( a = 0.04 \), \( b = 0.8 \). Explosion radius \( \hat{A} = 40 \). |
| OFWA         | Number of fireworks \( N = 5 \), Number of Gaussian mutation sparks \( N_g = 5 \), Constant \( m = 50 \). Parameter \( a = 0.04 \), \( b = 0.8 \). Explosion radius \( \hat{A} = 40 \). |
| OQFWA        | Number of fireworks \( N = 5 \), Number of Gaussian mutation sparks \( N_g = 5 \), Constant \( m = 50 \). Parameter \( a = 0.04 \), \( b = 0.8 \). Explosion radius \( \hat{A} = 40 \). Parameter \( \alpha = \pm 0.7 \). |

Figure 1. 2-D versions of functions and average convergence curves
Table 3. Function optimization results.

| Functions             | Evaluation criterion | Algorithms |
|-----------------------|----------------------|------------|
|                       |                      | FWA        | OFWA | OQFWA         |
| Sphere                | BEST                 | 0.1947     | 2.2821e-34 | 9.5852e-37  |
|                       | MEAN                 | 0.4388     | 6.1172e-15 | 4.9134e-17  |
|                       | WORST                | 0.7845     | 2.6586e-13 | 4.9011e-15  |
|                       | STD                  | 0.1329     | 3.4070e-14 | 4.9010e-16  |
| Sumsquares            | BEST                 | 6.2224     | 0.0025      | 9.6200e-38  |
|                       | MEAN                 | 15.0340    | 0.0025      | 3.0812e-13  |
|                       | WORST                | 35.5387    | 0.0025      | 2.5806e-11  |
|                       | STD                  | 5.2068     | 1.3547e-05 | 2.6042e-12  |
| Schaffer              | BEST                 | 0.0025     | 0.0025      | 0.0025       |
|                       | MEAN                 | 0.0025     | 0.0025      | 0.0025       |
|                       | WORST                | 0.0025     | 0.0026      | 0.0026       |
|                       | STD                  | 2.1780e-07 | 1.9497e-05 | 1.7833e-05  |
| Rastrigin             | BEST                 | 60.1215    | 0           | 0            |
|                       | MEAN                 | 102.8152   | 2.5889e-09 | 2.9276e-10  |
|                       | WORST                | 149.2116   | 2.5843e-07 | 1.5868e-08  |
|                       | STD                  | 17.7331    | 2.5842e-08 | 1.8769e-09  |
| Six hump camel        | BEST                 | -1.0316    | -1.0316    | -1.0316     |
|                       | MEAN                 | -1.0316    | -1.0312    | -1.0314     |
|                       | WORST                | -1.0315    | -1.0286    | -1.0295     |
|                       | STD                  | 1.8930e-05 | 5.2308e-04 | 4.2281e-04  |
| Ackley                | BEST                 | 0.0063     | 8.8817e-16 | 8.8818e-16  |
|                       | MEAN                 | 0.0267     | 3.0195e-12 | 2.8209e-13  |
|                       | WORST                | 0.0632     | 1.6235e-10 | 1.5210e-11  |
|                       | STD                  | 0.0128     | 1.8689e-11 | 1.6778e-12  |

5. Conclusion
In this paper, Opposition-Based Quantum Fireworks Algorithm (OQFWA) is proposed by integrating the Opposition-Based Learning and Quantum Optimization strategy. The new algorithm optimizes the original fireworks algorithm in the aspects of the selection of sparks and the local improvement of the optimal individuals, which effectively improves the convergence accuracy and speed of the algorithm. Finally, four algorithms are used to optimize the typical test functions with different characteristics. The simulation results show that OQFWA algorithm can combine good search performance and refinement, and converge to the global optimal solution with a large probability. It has strong global search ability, fast search speed and good convergence, especially superiority in solving numerical optimization problems.

Acknowledgments
This paper was supported in part by the National Natural Science Foundation of China through grant 61502522, Equipment Pre-Research Field Foundation of China through grant JZX7Y20190253036101.
the Natural Science Foundation of Hubei Province through grant 2019CFC897, and Joint Foundation of Ministry of Education for Equipment Pre-research through grant 6141A02033703.

References
[1] TAN Y, ZHU Y C. Fireworks Algorithm for Optimization[C]//1st International Conference on Swarm Intelligence. Beijing:Springer, 2010: 355-364.
[2] ZHENG S Q, Janek A, TAN Y. Enhanced Fireworks Algorithm[C]//2013 IEEE Congress on Evolutionary Computation. Cancun:IEEE, 2013:2069-2077. This reference has two entries but the second one is not numbered (it uses the ‘Reference (no number)’ style.
[3] ZHENG S Q, Janecek A, LI J. Dynamic Search In Fireworks Algorithm[C]//2014 IEEE Congress on Evolutionary Computation. Beijing: IEEE, 2014: 3222-3229.
[4] LI J, ZHENG S Q, TAN Y. Adaptive Fireworks Algorithm[C]//2014 IEEE Congress on Evolutionary Computation. Beijing: IEEE, 2014: 3214-3221.
[5] LI Hao, BAI Peng, ZHANG Hui, JIN Hongbin, XUE Junjie. Backward Fireworks Algorithm and Application Research [J]. Journal of Xi’an Jiaotong University, 2015,49(11):82-88.
[6] TAN Y. Fireworks Algorithm: A Swarm Intelligence Optimization Method [M]. Germany: Springer, 2015: 20-20.
[7] TAN Ying, ZHENG Shaoqiu. Recent Advances in Fireworks Algorithm [J]. CAAI Transactions on Intelligent Systems, 2014,9(05):515-528.
[8] Tizhoosh H R. Opposition-based learning: A new scheme for machine intelligence[C]. International Conference on Computing Intelligence for Modeling, Control and Automatic. Piscataway: Inst of Elec and Elec Eng Computer Society, 2005: 695-701.
[9] Rahnamayan S, Tizhoosh H R,Salama M M A. Quasi-oppositional Differential Evolution[C]// IEEE Congress on Evolutionary Computation. Singapore: IEEE, 2007: 2229-2236.
[10] Ergezer M, Simon D, Du D W. Oppositional biogeography-based optimization [C]// IEEE International Conference on Systems, Man and Cybernetics. San Antonio: IEEE, 2009: 1009-1014.
[11] Jiang M, Luo Y P, Yang S Y. Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm [J]. Information Processing Letters, 2007, 102(2007): 8-16.
[12] XIAO Wenxian, LIU Zhen. Particle Swarm Optimization Based on Opposition-based Learning and Quantum Optimization[J]. Microelectronics & Computer, 2013,30(06):126-130.