Optimal Choice of Parameters for Fireworks Algorithm

Vijay Kumar\textsuperscript{a}, Jitender Kumar Chhabra\textsuperscript{b}, Dinesh Kumar\textsuperscript{c}

\textsuperscript{a}Computer Science and Engineering Department, Thapar University, Patiala, Punjab, India
\textsuperscript{b}Computer Engineering Department, National Institute of Technology, Kurukshetra, Haryana, India
\textsuperscript{c}Computer Science and Engineering Department, GJUS&T, Hisar, Haryana, India

Abstract

The Fireworks Algorithm (FWA) is a relatively novel metaheuristic algorithm based on simulating the flare-up process of fireworks exploding and illuminating the night sky. The effects of major parameters on FWA were investigated on the unimodal and multimodal benchmark functions. The parameters such as number of fireworks, explosion amplitude, number of explosion sparks and number of iterations, have significant impact on the performance of FWA. The inappropriate selection of these parameters may lead to wrong results. The characteristics of FWA's parameters are depicted and general recommendation for deciding the optimal parameter values is given in this paper.

Keywords: Fireworks algorithm; Optimization; Metaheuristic; Parameter Selection.

1. Introduction

In the recent years, metaheuristic algorithms have been used for solving the real-life problems. Most of metaheuristic techniques are inspired by some intelligent colony activities in nature. The well-known metaheuristic techniques are Particle Swarm Optimization, Ant Colony Optimization, Gravitational Search Algorithm, Fireworks Algorithm etc. Tan and Zhu proposed a novel metaheuristic algorithm named as Fireworks Algorithm\textsuperscript{1} (FWA). It mimics the explosion phenomenon in searching nearby, the space of the solutions by generating sparks around the solutions. It has been applied on a variety of real world problems.

* Corresponding author. Tel.: +91-9466255106.
E-mail address: vijaykumarchahar@gmail.com
Parameter setting is the foremost problem of any metaheuristic technique. The FWA suffers from the same problem. The vital parameters of FWA are explosion amplitude coefficient, spark control parameter, and number of fireworks. These parameters greatly affect the performance of FWA such as exploration, exploitation, and convergence rate. Hence, the improper choice of these parameters may lead to wrong results. The main contribution of this paper is to investigate the effects of above-mentioned parameters on the six benchmark test functions.

The remainder of this paper is structured as follows. Section 2 presents the fireworks algorithm. Section 3 discusses the experimental setup for performance evaluation of FWA. The experimental results and discussions are presented in Section 4. Finally, the concluding remarks are made in Section 5.

2. Fireworks Algorithm

Fireworks algorithm (FWA) is a relatively novel optimization algorithm that simulates the explosion process of fireworks. FWA chooses some important positions as fireworks. They produce sparks to explore the local region. This process continues until at least one spark reaches the optimum value. The effectiveness of FWA lies in the suitable scheme for selecting the locations and superior design of explosion process. A good explosion generates numerous sparks and these sparks have an explosion center, whereas the bad explosion generates few sparks which are scattered in a particular place. The pseudo code of the FWA is described in Figure 1.

The FWA was mathematically modeled as follows. Suppose a system with \( n \) fireworks. The number of sparks and the explosion amplitude of \( i^{th} \) firework (i.e., \( x_i, \forall i = 1, 2, ..., n \)) are defined as:

\[
S_i = m \cdot \frac{f_{\text{high}} - f(x_i) + \epsilon}{\sum_{j=1}^{n} (f_{\text{high}} - f(x_j)) + \epsilon}
\]

\[
A_i = \hat{A} \cdot \frac{f(x_i) - f_{\text{low}} + \epsilon}{\sum_{j=1}^{n} (f(x_j) - f_{\text{low}}) + \epsilon}
\]

where \( m \) is a control parameter for sparks generation, \( \hat{A} \) is the highest explosion amplitude, \( n \) is the population size, and \( \epsilon \) is the small constant. \( f_{\text{high}} \) and \( f_{\text{low}} \) are the highest and lowest objective values among the \( n \) fireworks respectively.

To evade the overpowering effects of the fabulous fireworks, \( S_i \) is bounded as follows:

\[
S_i = \begin{cases} 
S_{\text{low}} & \text{if } S_i < X_{\text{low}} \\
S_{\text{high}} & \text{if } S_i > X_{\text{high}} \\
S_i & \text{otherwise}
\end{cases}
\]

For \( d \)-dimensional problem, the location of each spark \( x_j \) generated by \( x_i \) can be obtained by randomly selecting \( z^j \) directions such that \( z^j < d \), and for each dimension \( k \) setting the component \( x_j^k \) (\( 1 \leq j \leq S_i, 1 \leq k \leq d \)). There are two ways for setting \( x_j^k \). For most of the sparks, a displacement operator is added to \( x_j^k \) as:

\[
x_j^k = x_j^k + A_i \cdot \text{rand}(-1,1)
\]

For a few specific sparks, Gaussian distribution is applied to \( x_j^k \) such that

\[
x_j^k = x_j^k + \text{Gaussian}(1,1)
\]

According to the above mentioned ways, the newly obtained location of sparks goes outside the search space, it is mapped back to the search space as follows:
At each iteration, select the best location among all the sparks and keep it for the next generation. Thereafter, \( n - 1 \) fireworks are randomly selected with some probabilities, which is given by following equation:

\[
R_p(x_i) = \sum_{j \in K} \text{dist}(x_i, x_j) = \sum_{j \in K} \| x_i - x_j \|
\]

where \( K \) is the set of all the current locations of both fireworks and sparks. The selection probability of a location \( x_i \) is defined as follows:

\[
p(x_i) = \frac{R_p(x_i)}{\sum_{j \in K} R_p(x_j)}
\]

The main strength of FWA is the ability of focusing the search process on different area by using two types of spark generation methods.

**Fireworks Algorithm**

```
begin
Randomly initialize a swarm \( X \) of \( n \) fireworks
Initialize algorithm parameters \( \hat{A}, X_{\text{min}}, X_{\text{max}} \) and \( m \)
while ( \( t < \text{MaxGeneration} \) or \( \text{stop criterion} \) )
    Let \( W \) be the empty set of sparks
    for each fireworks \( x_i \in X \)
        Compute \( S_i \) for \( x_i \) by Eqs. 1 and 3
        Compute \( A_i \) for \( x_i \) by Eq. 2
        for \( j=1 \) to \( S_i \)
            Yield a spark \( x_j \)
            Let \( zl = \text{round} (d \cdot \text{rand} (0,1)) \)
            for \( k =1 \) to \( zl \)
                Compute \( x_j^k \) by Eqs. 4 and 6
            end for
        end for
    end for
Randomly select a set \( P \) of \( m \) fireworks from \( X \)
for each fireworks \( x_i \in P \)
    Yield a spark \( x_j \)
    Let \( zl = \text{round} (d \cdot \text{rand} (0,1)) \)
    for \( k =1 \) to \( zl \)
        Compute \( x_j^k \) by Eqs. 5 and 6
    end for
end for
W = W \cup x_j
end for
W = W \cup X
Let gbest be the best location among \( W \) and Set \( X = \text{gbest} \)
Add to \( X \) other \( n-1 \) locations selected from \( W \) based on Eq. 7
end while
Return the best firework.
end
```

Fig. 1. Pseudo code of the Fireworks Algorithm (FWA).
3. Experimental Setup

The experiments were performed to analyze the performance of FWA by varying the parameter settings. The benchmark test functions are mentioned in Section 3.1.

3.1. Benchmark Test Functions

Six benchmark test functions were used to explore the effect of control parameters on FWA. The details of benchmark functions are mentioned in Table 1. In these table, \( d \) is the dimension of the function. The objective of these benchmark test functions is to find global lowest value.

<table>
<thead>
<tr>
<th>Function name</th>
<th>Function Type</th>
<th>Benchmark Test Function</th>
<th>Search Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>Unimodal</td>
<td>( F_1(Y) = \sum_{i=1}^{d} y^2_i )</td>
<td>([0, 100]^d)</td>
</tr>
<tr>
<td>Schwefel</td>
<td>Unimodal</td>
<td>( F_2(Y) = \sum_{i=1}^{d} \left( \sum_{j=1}^{d} y_j \right)^2 )</td>
<td>([0, 100]^d)</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>Unimodal</td>
<td>( F_3(Y) = \sum_{i=1}^{d} \left[ 100(y_{i+1} - y_i^2)^2 + (y_i - 1)^2 \right] )</td>
<td>([-30, 30]^d)</td>
</tr>
<tr>
<td>Griewank</td>
<td>Multimodal</td>
<td>( F_4(Y) = -\frac{1}{4000} \sum_{i=1}^{d} y_i^2 - \prod_{i=1}^{d} \cos \left( \frac{y_i}{\sqrt{d}} \right) + 1 )</td>
<td>([-600, 600]^d)</td>
</tr>
<tr>
<td>Ackley</td>
<td>Multimodal</td>
<td>( F_5(Y) = -20 \exp \left( -0.2 \sqrt{\frac{1}{d} \sum_{i=1}^{d} y_i^2} \right) - \exp \left( \frac{1}{d} \sum_{i=1}^{d} \cos (2 \pi y_i) \right) + 20 + e )</td>
<td>([-32.32, 32.32]^d)</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>Multimodal</td>
<td>( F_6(Y) = \sum_{i=1}^{d} y_i^2 - 10 \cos (2 \pi y_i) + 10 )</td>
<td>([-5.12, 5.12]^d)</td>
</tr>
</tbody>
</table>

3.2. Parameter Settings for FWA

The explosion operator has two main parameters. The first parameter is the number of fireworks \( n \). The \( n \) is varied to different values from 5 to 50 with interval of 5. The second parameter is \( m \), which is used to limit the total number of sparks. It is varied from 10 to 100 with interval of 10. The another important parameter is the explosion amplitude \( \hat{A} \), which was allowed to vary from 10 to 100 with interval of 10. The ranges and increments of parameters used are depicted in Table 3. The maximum number of generation was taken as 5000.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>[5,50]</td>
<td>5</td>
</tr>
<tr>
<td>( m )</td>
<td>[10,80]</td>
<td>10</td>
</tr>
<tr>
<td>( \hat{A} )</td>
<td>[10,100]</td>
<td>10</td>
</tr>
</tbody>
</table>
4. Experimental Results and Discussions

The influence of number of fireworks, control parameter, and explosion amplitude is analyzed on the performance of benchmark test functions.

4.1. Influence of \( n \)

Figure 2 shows the effect of number of fireworks \( (n) \) on the benchmark test functions. The value of \( n \) is varied from 5 to 50. While other parameters are set as \( A = 40 \) and \( m = 40 \). The results reveal that the small value of \( n \) offers best fitness value. The optimal choice of \( n \) is 5.

![Fig. 2. Effect of \( n \) on the performance of FWA.](image)

4.2. Influence of \( m \)

Figure 3 shows the average fitness function value of benchmark test functions. The value of \( m \) is varied from 10 to 80 when \( A = 40 \) and \( n = 5 \). It is observed that small value of \( m \) generally lead to good results. The optimal choice of \( m \) is 10.
4.3. Influence of $\hat{A}$

Figure 4 demonstrates the effect of $\hat{A}$ when $m=10$ and $n=5$ for benchmark test functions. The value of $\hat{A}$ is varied from 10 to 100 with interval of 10. It is observed that the optimal choice of $\hat{A}$ is 90.
Fig. 4. Effect of $\hat{A}$ on the performance of FWA.

5. Conclusion

The fireworks algorithm has been used in a wide variety of optimization problems. The FWA contains some parameter tuning that significantly influence the performance. In this paper, the effect of parameters on performance of the FWA is observed. The optimal choice of number of fireworks is 5 for both unimodal and multimodal benchmark test functions. For benchmark test functions, it is better to choose the value of $m$ is equal to 10. The value of $\hat{A}$ is equal to 90 is a relatively better choice.

References