A Hybrid Fireworks Optimization Method with Differential Evolution Operators

YuJun Zheng\(^a\),*, XinLi Xu\(^a\), HaiFeng Ling\(^b\)

\(^a\) College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou, 310023, China

\(^b\) Department of Mechanical Engineering, PLA University of Science & Technology, Nanjing 210007, China

Abstract

Fireworks algorithm (FA) is a relatively new swarm-based metaheuristic for global optimization. The algorithm is inspired by the phenomenon of fireworks display and has a promising performance on a number of benchmark functions. However, in the sense of swarm intelligence, the individuals including fireworks and sparks are not well-informed by the whole swarm. In this paper we develop an improved version of the FA by combining with differential evolution (DE) operators: mutation, crossover, and selection. At each iteration of the algorithm, most of the newly generated solutions are updated under the guidance of two different vectors that are randomly selected from highly ranked solutions, which increases the information sharing among the individual solutions to a great extent. Experimental results show that the DE operators can improve diversity and avoid prematurity effectively, and the hybrid method outperforms both the FA and the DE on the selected benchmark functions.

Key words: Fireworks algorithm (FA), Differential evolution (DE), Global optimization, Hybrid

1 Introduction

The complexity of real-world engineering optimization problems gives rise to various kinds of metaheuristics that use stochastic techniques to effectively explore the search space for a global optimum. In particular, metaheuristics

* Corresponding author. Tel.: +86-571-85290085.

Email address: yujun.zheng@computer.org (YuJun Zheng).
based on swarm intelligence (e.g., [1–7]), which simulates a population of simple individuals evolving their solutions by interacting with one another and with the environment, have shown promising performance on many difficult problems and have become a very active research area in recent years.

Fireworks algorithm (FA) is a relatively new global optimization method originally proposed by Tan and Zhu [7]. Inspired by the phenomenon of fireworks explosion, the algorithm selects in the search space a certain number of locations, each for exploding a firework to generate a set of sparks; The fireworks and sparks with good performance are chosen as the locations for the next generation’s fireworks, and the evolutionary process continues until a desired optimum is obtained, or the stopping criterion is met. Numerical experiments on a number of benchmark functions show that, the FA can converge to a global optimum with a much smaller number of function evaluations than that of particle swarm optimization (PSO) algorithms [1,8].

In the standard FA, the convergence speed is accelerated by “good” fireworks that generate more sparks within smaller explosion areas, and the search diversity is improved by “bad” fireworks that generate fewer sparks within larger explosion areas. However, to some extent, such a diversification mechanism is not very flexible and, in particular, it does not utilize more information about other quality solutions in the swarm. That is, in the sense of swarm intelligence, the individuals (fireworks and sparks) are not well-informed by the whole swarm.

Inspired by this observation, we develop an improved fireworks optimization method by combining with differential evolution (DE) operators: mutation, crossover, and selection [9]. At each iteration of the algorithm, these operators are applied to guide the generation of new solutions, which improves the diversity of the swarm and avoids being trapped in local optima too early. Experiments on selected benchmark functions show that the well-informed fireworks and sparks can improve the performance of the FA to a great extent.

The remainder of this paper is structured as follows: Section 2 briefly describes the FA algorithm and the DE algorithm, Section 3 presents the framework of our hybrid FA method, the performance of which is demonstrated by the experimental results in Section 4. Section 5 concludes with discussion.
2 Backgrounds

2.1 Fireworks Algorithm

In the FA presented in [7], the number of sparks and the amplitude of explosion for each firework $x_i$ is respectively defined as follows.

$$s_i = m \cdot \frac{f_{\text{max}} - f(x_i) + \epsilon}{\sum_{j=1}^{p}(f_{\text{max}} - f(x_j)) + \epsilon}$$  \hspace{1cm} (1)

$$A_i = \hat{A} \cdot \frac{f(x_i) - f_{\text{min}} + \epsilon}{\sum_{j=1}^{p}(f(x_j) - f_{\text{min}}) + \epsilon}$$  \hspace{1cm} (2)

where $m$ and $\hat{A}$ are control parameters, $p$ is the size of the swarm, $f_{\text{max}}$ and $f_{\text{min}}$ are respectively the maximum and minimum objective values among the $p$ fireworks, and $\epsilon$ is a small constant to avoid zero-division-error.

To avoid overwhelming effects of splendid fireworks, lower and upper bounds are defined for $s_i$ such that:

$$s_i = \begin{cases} 
  s_{\text{min}} & \text{if } s_i < s_{\text{min}} \\
  s_{\text{max}} & \text{else if } s_i > s_{\text{max}} \\
  s_i & \text{else}
\end{cases}$$  \hspace{1cm} (3)

For a $D$-dimensional problem, the location of each spark $x_j$ generated by $x_i$ can be obtained by randomly setting $z$ directions ($z < D$), and for each dimension $k$ setting the component $x_{ij}^k$ based on $x_i^k$ ($1 \leq j \leq s_i, 1 \leq k \leq z$). There are two ways for setting $x_{ij}^k$. For most sparks, a displacement $h_k = A_i \cdot \text{rand}(-1, 1)$ is added to $x_i^k$, i.e.:

$$x_{ij}^k = x_i^k + A_i \cdot \text{rand}(-1, 1)$$  \hspace{1cm} (4)

To keep the diversity, for a few specific sparks, an explosion coefficient based on Gaussian distribution is applied to $x_i^k$ such that:

$$x_{ij}^k = x_i^k \cdot \text{Gaussian}(1, 1)$$  \hspace{1cm} (5)

In both the ways, if the new location falls out of the search space, it is mapped to the search space as follows:

$$x_{ij}^k = x_{\text{min}}^k + |x_{ij}^k| \% (x_{\text{max}}^k - x_{\text{min}}^k)$$  \hspace{1cm} (6)
where % denotes the modulo operator for floating-point numbers, as defined in most computer languages.

At each iteration of the FA, among all the current sparks and fireworks, the best location is always selected as a firework of the next generation. After that, $p - 1$ fireworks are selected with probabilities proportional to their distance to other locations.

The general framework of the FA is described in Algorithm 1.

**Algorithm 1** The standard fireworks algorithm.

```
set the algorithm parameters $p$, $s_{\text{min}}$, $s_{\text{max}}$, $\hat{A}$, and $\hat{m}$;
randomly initialize a swarm $S$ of $p$ fireworks;

while (stop criteria is not met) do
  let $R$ be the empty set of sparks;
  foreach firework $x_i \in S$ do
    calculate $s_i$ for $x_i$ according to Equation (1) and (3);
    calculate $A_i$ for $x_i$ according to Equation (2);
    for $j = 1$ to $s_i$ do
      yield a spark $x_j$;
      let $z = \text{round}(D \cdot \text{rand}(0, 1))$;
      for $k = 1$ to $z$ do set $x_{jk}$ according to Equation (4) and (6);
      $R = R \cup \{x_j\}$;
  end
  randomly select a set $P$ of $\hat{m}$ fireworks from $S$;
  foreach firework $x_i \in P$ do
    yield a spark $x_j$;
    let $z = \text{round}(D \cdot \text{rand}(0, 1))$;
    for $k = 1$ to $z$ do set $x_{jk}$ according to Equation (5) and (6);
    $R = R \cup \{x_j\}$;
    $R = R \cup S$;
    let $gbest$ be the best location among $R$, and set $S = \{gbest\}$;
    Add to $S$ other $p - 1$ locations selected from $R$ based on distance probabilities;
  end
end
```

### 2.2 Difference Evolution

Introduced by Storn and Price [9], DE is an efficient evolutionary algorithm that simultaneously evolves a population of solution vectors. But unlike the genetic algorithm (GA) [10], DE uses floating-point vectors and does not employ some probability density functions for vector reproduction. Specifically, DE generates a mutant vector $\mathbf{v}$ for each vector $\mathbf{x}_i$ in the population by adding
the weighted difference between two randomly selected vectors to a third one:

\[ \mathbf{v}_i = \mathbf{x}_{r_1} + \gamma (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \]  

(7)

where random indexes \( r_1, r_2, r_3 \in \{1, 2, ..., p\} \) and coefficient \( \gamma > 0 \).

A trial vector \( \mathbf{u}_i \) is then generated by using the crossover operator which mixes the components of the mutant vector and the original one, where each \( j \)th component of \( \mathbf{u}_i \) is determined as follows:

\[
\mathbf{u}_{ij} = \begin{cases} 
\mathbf{v}_{ij} & \text{if } \text{rand}(0, 1) < c_r \text{ or } j = r(i) \\
\mathbf{x}_{ij} & \text{else}
\end{cases}
\]

(8)

where \( c_r \) is the crossover probability ranged in \((0, 1)\) and \( r(i) \) is a random integer within \((0, p]\) for each \( i \).

In the last step of each iteration, the selection operator chooses the better one for the next generation by comparing \( \mathbf{u}_i \) with \( \mathbf{x}_i \):

\[
\mathbf{x}_i = \begin{cases} 
\mathbf{u}_i & \text{if } f(\mathbf{u}_i) \leq f(\mathbf{x}_i) \\
\mathbf{x}_i & \text{else}
\end{cases}
\]

(9)

3 The Hybrid Fireworks Optimization Method

For a \( D \)-dimensional optimization, the fitness value of a solution is determined by values of all components, and a solution that has discovered the region corresponding to the global optimum in some dimensions may have a low fitness value because of the poor quality in the other dimensions [11]. Thus, many population-based optimization methods, including DE, comprehensive learning PSO [11], fully informed PSO [12], enable the individuals to make use of the beneficial information in the swarm more effectively to generate better quality solutions.

In the standard FA, after obtaining the set \( R \) of all fireworks and sparks, the locations for new fireworks are selected based on distance to other locations in \( R \) so as to keep diversity of the swarm. Here we introduce the DE operators to the FA to achieve this purpose.

In the hybrid algorithm, after obtaining the set \( R \) of locations, we first sort the locations in increasing order of fitness, and create a set \( S \) of candidate locations which contains the first (best) location in \( R \) and other \( p-1 \) locations randomly selected from the top \( 2p \) locations in \( R \). Then the standard DE process is applied to \( S \) except its best vector.
In detail, the following procedure is used to replace the last two lines in Algorithm 1:

1. sort \( R \) in increasing order of vector fitness;
2. let \( S = \{R[1]\}, R = R\backslash\{R[1]\} \);
3. truncate the length of \( R \) to \( 2p - 1 \);
4. randomly select \( p - 1 \) locations from \( R \) and add them to \( S \), where the selection probability of each \( x \in R \) is \( f(x)/\sum_{z \in R} f(z) \);
5. for \( i = 2 \) to \( p \), apply the DE mutation, crossover, and selection operators to \( S[i] \).

Generally speaking, the above DE procedure helps to improve the algorithm in the following aspects:

- For high quality (top \( 2p \)) vectors in \( R \), each of them has an opportunity to influence the new fireworks for the next generation, in terms of the roulette-wheel selection and DE selection operations.
- For candidate fireworks in \( S \), each of them has an opportunity to be informed by existing high quality vectors at each dimension, in terms of the DE mutation and crossover operations.
- In particular, the DE mutation operator makes the difference of two random vectors acts as a search direction for the third one [13]; In comparison with large-amplitude explosion and distance-based selection, the mutation operation is more effective in improving the probability of obtaining the global optimum, whilst requiring less computational cost.

4 Computational Experiments

We choose a set of well-known test functions as benchmark problems, the definitions of which are listed in the Appendix. All the functions are tested on 30, 50, and 100 dimensions respectively. The performance of the hybrid algorithm is evaluated in comparison with the standard FA and standard DE. The parameters of the DE are set as those in Ref. [9]. For the FA, we set \( p = 5 \), \( m = 50 \), \( s_{\text{min}} = 2 \), \( s_{\text{max}} = 40 \), \( \hat{m} = 5 \), and \( \hat{A} = \min_{1 \leq k \leq D} (x^k_{\text{max}} - x^k_{\text{min}})/5 \) (where \( [x^k_{\text{min}}, x^k_{\text{max}}] \) is the search range of the \( k \)th dimension of the benchmark function), as suggested in Ref. [7]. For the hybrid algorithm, since the DE operators are introduced for improving solution diversity, we decrease the values of three parameters such that \( m = 25 \), \( s_{\text{max}} = 20 \), and \( \hat{A} = \min_{1 \leq k \leq D} (x^k_{\text{max}} - x^k_{\text{min}})/7 \).

For the algorithms, the mean best and the success rate over 40 runs are used as performance measures. The maximum number of function evaluations (nfe) is set to 10000 for 30 dimensional functions, 20000 for 50 dimensional func-
Table 1

Detailed information of the benchmark functions used in the paper.

<table>
<thead>
<tr>
<th>Function</th>
<th>Search range</th>
<th>$x^*$</th>
<th>$f(x^*)$</th>
<th>$a_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>$[-100, 100]^D$</td>
<td>$[0,0,...,0]$</td>
<td>0</td>
<td>0.000001</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>$[-2.048, 2.048]^D$</td>
<td>$[1,1,...,1]$</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>Schwefel</td>
<td>$[-500, 500]^D$</td>
<td>$[420.96,420.96,\ldots,420.96]$</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>$[-5.12, 5.12]^D$</td>
<td>$[0,0,...,0]$</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>Griewank</td>
<td>$[-600, 600]^D$</td>
<td>$[0,0,...,0]$</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>Ackley</td>
<td>$[-32.768, 32.768]^D$</td>
<td>$[0,0,...,0]$</td>
<td>0</td>
<td>0.000001</td>
</tr>
</tbody>
</table>

The experimental results on 30, 50, and 100 dimensional functions are summarized in Table 2, Table 3, and Table 4 respectively, and the average convergence speeds of the algorithms are illustrated in Fig 1 ~ 3. As we can see, for all the benchmark problems, the performance of the hybrid algorithm is better (or at least not worse than) the standard FA and DE. In the sense of required accuracy, except for the 100-D Schwefel function, the hybrid algorithm successfully achieves the global optimum for all the other functions. In particular we observe that:

- For the Sphere, Rastrigin, and Griewank functions, where the FA has significant performance advantages over the DE, the introduction of DE operators does not degrade the performance of the hybrid algorithm.
- For the Rosenbrock, Schwefel, and Ackley functions, where the performance of the DE is not far from (or may be better than) that of the FA, the introduction of DE operators contributes to the performance improvement of the hybrid algorithm to a great extent.

Moreover, the advantage of the hybrid algorithm is more obvious on 100-D problems than on 50-D and 30-D problems, which shows that the hybrid algorithm is scalable and efficient to high-dimensional problems.

5 Conclusion

FA has been shown to be one of the best performing algorithms for global optimization problems. The paper improves the standard FA by introducing
### Table 2
The experimental results on 30-D benchmark problems.

<table>
<thead>
<tr>
<th>Function</th>
<th>mean best</th>
<th>FA</th>
<th>DE</th>
<th>Hybrid</th>
<th>success rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>0</td>
<td>1.077E-39</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Schwefel</td>
<td>4.827E+00</td>
<td>1.133E+01</td>
<td>4.699E-01</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>1.228E-11</td>
<td>1.985E-03</td>
<td>0</td>
<td>100</td>
<td>32.5</td>
</tr>
<tr>
<td>Griewank</td>
<td>4.441E-16</td>
<td>2.565E-15</td>
<td>4.441E-16</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

### Table 3
The experimental results on 50-D benchmark problems.

<table>
<thead>
<tr>
<th>Function</th>
<th>mean best</th>
<th>FA</th>
<th>DE</th>
<th>Hybrid</th>
<th>success rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>0</td>
<td>3.383E-36</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>3.080E-01</td>
<td>1.115E+00</td>
<td>1.750E-03</td>
<td>57.5</td>
<td>5</td>
</tr>
<tr>
<td>Schwefel</td>
<td>9.245E+01</td>
<td>1.293E+01</td>
<td>6.883E-01</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>1.541E-11</td>
<td>2.480E-03</td>
<td>0</td>
<td>100</td>
<td>25</td>
</tr>
<tr>
<td>Griewank</td>
<td>4.441E-16</td>
<td>4.752E-13</td>
<td>4.441E-16</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

### Table 4
The experimental results on 100-D benchmark problems.

<table>
<thead>
<tr>
<th>Function</th>
<th>mean best</th>
<th>FA</th>
<th>DE</th>
<th>Hybrid</th>
<th>success rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>7.474E-187</td>
<td>1.627E-26</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>3.044E+01</td>
<td>4.036E+00</td>
<td>5.270E-02</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Schwefel</td>
<td>2.062E+02</td>
<td>3.430E+01</td>
<td>4.100E-00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>0</td>
<td>3.526E-06</td>
<td>0</td>
<td>100</td>
<td>92.5</td>
</tr>
<tr>
<td>Griewank</td>
<td>4.380E-08</td>
<td>5.288E-03</td>
<td>0</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>Ackley</td>
<td>4.441E-16</td>
<td>8.089E-08</td>
<td>4.441E-16</td>
<td>100</td>
<td>70</td>
</tr>
</tbody>
</table>
DE operators for increasing the information sharing among the fireworks and sparks and diversifying the search process. Experimental results show that the hybrid method performs better than both the FA and the DE. We believe such an improvement can bring significant advantages to practical engineering applications. In future work, we will study more relationships between the swarm-based optimization methods and seek more hybridization strategies for potential improvement on those methods.

Fig. 1. The averaged convergence curves of the 30D test problems.
Fig. 2. The averaged convergence curves of the 50D test problems.

Acknowledgment

This work was supported by grants from National Natural Science Foundation (No. 61105073, 61103140, 61173096, 61020106009) of China. We are grateful to the anonymous reviewers of ICSI 2012 for their valuable suggestions for improving the paper.
Fig. 3. The averaged convergence curves of the 100D test problems.

Appendix: Definitions of the test functions

(1) Sphere function

\[ f_1(x) = \sum_{i=1}^{D} x_i^2 \]
(2) Rosenbrock’s function

\[ f_2(x) = \sum_{i=2}^{D-1} (100(x_i^2 - x_{i-1})^2 + (x_i - 1)^2) \]

(3) Schwefel’s function

\[ f_3(x) = 418.9829 \times D - \sum_{i=1}^{D} x_i \sin(|x_i|^{\frac{1}{2}}) \]

(4) Rastrigin’s function

\[ f_4(x) = \sum_{i=1}^{D} (x_i^2 - 10 \cos(2\pi x_i) + 10) \]

(5) Griewank’s function

\[ f_5(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \]

(6) Ackley’s function

\[ f_6(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^{D} \cos(2\pi x_i)\right) + 20 + e \]

References


