Hybridized Fireworks Algorithm for Global Optimization

NEBOJSA BACANIN  
Faculty of Computer Science  
Megatrend University  
Bulevar umetnosti 29, Belgrade  
SERBIA  
nbacanin@megatrend.edu.rs

MILAN TUBA  
Faculty of Computer Science  
Megatrend University  
Bulevar umetnosti 29, Belgrade  
SERBIA  
tuba@ieee.org

MARKO BEKO  
Department of Information Technology  
Universidade Lusófona  
Lisbon  
PORTUGAL  
beko.marko@gmail.com

Abstract: In this paper we introduce hybridized fireworks algorithm for global optimization problems. We replaced Gaussian search method from the original fireworks algorithm with the search equation adopted from the firefly algorithm. To test our approach, we implemented six standard bound-constrained benchmarks and performed comparative analysis with the basic fireworks algorithm, as well as with two other metaheuristics. We conducted two tests. In the experiments, all algorithms were tested with the same number of function evaluations. Preliminary results show that our algorithm has a potential when dealing with such problems and it is worth of further research.

Key–Words: Swarm intelligence, global optimization, hybridized algorithms, fireworks algorithm, firefly algorithm

1 Introduction

Global optimization problems have been solved by different methods and techniques. Traditional, deterministic approaches could not obtain satisfying results in a reasonable computational time when tackling large-scale tasks. Thus, as an alternative, a metaheuristics have emerged. Metaheuristics can roughly be divided into two main categories - those inspired, and those not inspired by the nature. Nature-inspired algorithms can be further divided into evolutionary algorithms (EA) and swarm intelligence. The most prominent representative of EA is genetic algorithm (GA).

Social behavior of swarms of ants, bees, birds, fish and worms was an inspiring source for the deviation of swarm intelligence [1]. Swarms are composed of relatively simple and unsophisticated individuals, but together, they exhibit intelligent and coordinated behavior that guides them to the desired objective with no central component that controls and manages the whole system. There are many papers in the literature that deal with the swarm intelligence [2]. Also, there are many algorithms that belong to this group.

Ant colony optimization (ACO) models ant’s ability to find the shortest path between the nest and the food source by deploying pheromones. It was one among the first members of swarm intelligence [3], but it was later extended and applied to wide variety of problems [4], [5], [6], [7], [8]. Particle swarm optimization (PSO) mimics the behavior of schools of fish and birds and many applications of PSO are known in the literature [9]. Artificial bee colony (ABC) algorithm performs intensification and diversification processes, which are cornerstones of swarm algorithms, by utilizing employed, onlooker and scout bees. It was successfully applied on constrained benchmark problems [10], on engineering problems [11], [12], as well as on many others [13], [14], [15], [16]. Cuckoo search (CS) algorithm, that was devised by Yang [17], emulates the search process of cuckoo birds by using simplification rules. It also has many applications in the literature [18]. Bat algorithm is relatively new bio-inspired metaheuristic, also proposed by Yang, and successfully applied to various problems [19], [20]. Seeker optimization algorithm (SOA) mimics the human search process by utilizing fuzzy reasoning and human memory. It was successfully adopted for global unconstrained optimization [21] and constrained tasks [22].

Due to the fact that each swarm intelligence algorithm has its own advantages and drawbacks, some approaches are merged together in hybrid implementations. Many hybrid approaches are known in the literature [10], [23], [24], [22].

In this paper, we show hybridized fireworks algorithm (FWA) for solving global optimization tasks. To enhance the search process, we replaced Gaussian search procedure from the basic FWA with the mechanism adopted from the firefly algorithm (FA). We tested our approach on standard benchmarks as proposed in [25], and compared testing results with those obtained by the basic FWA and other state-of-the-art
approaches. For the purpose of objective comparative analysis, all algorithms were tested with the same number of function evaluations.

After Introduction, we show both, original and hybridized FWA. Section 3 presents experimental results for six standard bound-constrained benchmarks with different number of function evaluations. Also, in this section, we give comparative analysis with other approaches. Finally, Section 4 concludes this paper.

2 Original and Hybridized Fireworks Algorithm

2.1 Background

Fireworks algorithm (FWA) is a novel swarm intelligence metaheuristic for solving optimization problems. It was devised in 2010 by Tan and Zhu [25]. FWA simulates the process of firework explosion.

According to the literature survey, there are only few FWA implementations. In its very first implementation, it was reported that FWA obtains better performance than standard PSO (SPSO) and Clonal PSO (CPSO) [25]. Enhanced fireworks algorithm (EFWA) was proposed in [26] and showed excellent performance. In [27], improved version on EFWA based on adaptive dynamic local search mechanism (dynFWA) was showed for solving CEC 2013 benchmark suite that includes 28 different benchmark functions. dynFWA approach outscored EFWA. Hybrid approach, which was devised in 2014, combines FWA and differential mutation (FWA-DM). It was successfully tested on CEC 2014 benchmark problems. One more approach which is worth mentioning is parallelized GPU-based Fireworks Algorithm (GPU-FWA), that efficiently utilizes graphical processing unit (GPU), for solving large-scale problems [28].

The main source of inspiration for the FWA was the process of setting off a firework. When a firework is set off, a shower of sparks fill the local space around it. According to the opinion of the FWA’s creators, the explosion process of a firework can be viewed as a search in the local space around a specific point where the firework is set off through the sparks generated in the explosion [25].

Firework explosion exhibits two characteristic behaviors. In well manufactured fireworks, numerous sparks are generated, and sparks tend to be in the center of the explosion [25]. On the other hand, fireworks of lower quality manifest different behavior. In such cases, quite few sparks are generated, and the sparks are scattered in the space.

These two behaviors have implications on the search algorithm. In the first case, when a firework is well manufactured, a firework is located in the promising area of the search space which may be close to the optimal solution. Thus, it is better to generate more sparks to search around the firework. Contrarily, the second case, when sparks are scattered, implies that the optimal problem’s solution may be far from the location of the firework, and that the search radius should be larger. In FWA, more sparks are generated and the explosion amplitude is smaller for a good firework, compared to a bad one [25].

2.2 Basic algorithm details

FWA is stochastic, iterative-based approach. In each generation of explosions, $n$ locations for the deployment of $n$ fireworks are selected. After that, $n$ other locations are selected from the current sparks and fireworks locations for the next iteration of execution.

Two crucial components of the FWA are number of sparks and the amplitude of explosion. To depict this, let’s suppose that the FWA is designed for numerical minimization problem:

$$
\min f(x), \ x = (x_1, x_2, x_3, ..., x_D) \in S,
$$

where $x$ represents a real vector with $D \geq 1$ components (parameters) and $S \in \mathbb{R}^D$ is hyper-rectangular search space with $D$ dimensions constrained by lower and upper bounds:

$$
lb_i \leq x_i \leq ub_i, \ i \in [1, D]
$$

Then, the number of sparks which are generated by each firework $x_i$ can be expressed as [25]:

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

where $m$ represents a parameter that controls the overall number of sparks that are generated by the $n$ fireworks, $y_{\max} = \max(f(x_i) \ (i = 1, 2, ..., n))$ is the worst firework in the population (with the greatest value of objective function in the case of minimization problems), and $\eta$ is a small constant which is used to avoid division-by-zero error.

According to experimental results, it was concluded that it is not good if $s_i$ is too big, and bounds on $s_i$ are defined according to [25]:

$$\hat{s}_i = \begin{cases} 
\text{round}(\alpha \cdot m) & \text{if } s_i < \alpha \cdot m \\
\text{round}(\beta \cdot m) & \text{if } s_i > \beta \cdot m, \ \alpha < \beta < 1,
\end{cases}
$$

where $s_i$ is the unbounded number of sparks. 

where $\alpha$ and $\beta$ are constant parameters.

Second important component of FWA is the amplitude of explosion. The amplitude of a well-designed fireworks is smaller than that in a bad one. This amplitude is defined with the following expression [25]:

$$A_i = \hat{A} \cdot \frac{f(x_i) - y_{\min} + \eta}{\sum_{i=1}^{n} (f(x_i) - y_{\min}) + \eta},$$

(5)

where $\hat{A}$ represents the highest value of the explosion amplitude, while $y_{\min} = \min(f(x_i)) (i = 1, 2, ..., n)$ denotes best fireworks in the population of $n$ fireworks (the one with the lowest value of objective function).

When an explosion occurs, $z$ random dimensions (directions) of spark are affected. The number of affected dimensions is obtained by using [25]:

$$z = \text{round} (d \cdot \chi),$$

(6)

where $d$ is number of dimensions of optimization problem (the location $x$), and $\chi$ is a random number uniformly distributed between 0 and 1.

In order to determine the location of a spark of the firework $x_i$, a spark location $\hat{x}_j$ is generated first. The whole process is shown in the pseudo-code below [25].

**Pseudo-code 1**

Determine the initial spark's location: $\hat{x}_j = x_i$

Select random $z$ dimensions of $\hat{x}_j$ by using Eq. (6)

Calculate the displacement: $h = A_i \cdot \sigma$

for each selected dimension $\hat{x}_j$ do

$$\hat{x}_j = \hat{x}_j + h$$

if $\hat{x}_j < x_k^{\min}$ or $\hat{x}_j > x_k^{\max}$

map $\hat{x}_j$ to the potential space

end if

end for

In each iteration, $n$ fireworks locations are selected. Moreover, the current best location $x^*$ is always kept and transferred to the next iteration. Thus, some kind of elitism is utilized. After that point, $n - 1$ locations are selected based on their distance to the other locations [25]. The distance between a position $x_i$ and other locations is calculated as [25]:

$$R(x_i) = \sum_{j \in K} d(x_i, x_j) = \sum_{j \in K} ||x_i - x_j||,$$

(7)

where $K$ is the set of all current locations of both fireworks and sparks.

The probability of selecting location $x_i$ is formulated as [25]:

$$p(x_i) = \frac{R(x_i)}{\sum_{j \in K} R(x_j)},$$

(8)

where $p(x_i)$ is the probability that the position $x_i$ will be selected.

### 2.3 Our hybridization

We hybridized basic FWA with the firefly algorithm (FA), which was inspired by the flashing properties of fireflies. We named our hybrid FWA firefly search (FWA-FS).

FA was first proposed by Yang for unconstrained optimization [29]. By observing FA, it is proved that its search mechanism is very powerful in a way that it improves the convergence speed towards the optimal region of the search space [14]. Many FA’s implementations for different kinds of problems are found in the literature survey [23], [30], [31].

Essential principle of FA is that each firefly goes towards the brighter firefly. With the increase of the distance from the lighting source, the light intensity decreases, and vice-versa. It is modeled by the:

$$I(r) = \frac{I_0}{1 + \gamma r^2},$$

(9)
where \( I(r) \) is the light intensity, \( r \) is distance, \( I_0 \) is the light intensity at the source, and \( \gamma \) is light absorption coefficient.

The attractiveness \( \beta \) of a firefly is proportional to its light intensity (brightness), and it can be shown by the following expression [23]:

\[
\beta(r) = \frac{\beta_0}{1 + \gamma r^2},
\]

where \( \beta_0 \) is the attractiveness at \( r = 0 \).

Search equation of FA is based upon attractiveness, and when firefly \( j \) is more attractive (brighter) than firefly \( i \), firefly \( i \) is moving towards \( j \) [23]:

\[
x_i(t) = x_i(t) + \beta_0 r^{-\gamma} (x_j - x_i) + \alpha (\text{rand} - 0.5),
\]

where \( \beta_0 \) is attractiveness at \( r = 0 \), \( \alpha \) is randomization parameter, \( \text{rand} \) is random number uniformly distributed between 0 and 1, and \( r_{i,j} \) is distance between fireflies \( i \) and \( j \). This distance is calculated using Cartesian distance form:

\[
r_{i,j} = ||x_i - x_j|| = \sqrt{\sum_{k=1}^{D} (x_{i,k} - x_{j,k})},
\]

In FWA-FS, a Gaussian explosion is not used. Instead of it, firefly search is triggered. New search mechanism is shown in the pseudo-code below.

**Pseudo-code 3**

1. Determine the initial spark’s location: \( \hat{x}_j = x_i \)
2. Select random \( z \) dimensions of \( \hat{x}_j \) by using Eq. (6)
3. Calculate the brightness of all sparks and find a spark that is brighter than \( \hat{x}_j \)
4. for each selected dimension \( \hat{x}^j_k \) of \( \hat{x}_j \) do
   1. \( \hat{x}^k_j = \hat{x}^j_k + \beta_0 r^{-\gamma} (\hat{x}^j_k - x^k_i) + \alpha (\text{rand} - 0.5) \)
   2. if \( \hat{x}^k_j < x_k^\text{min} \) or \( \hat{x}^k_j > x_k^\text{max} \) map \( \hat{x}^k_j \) to the potential space
5. end if
6. end for

In the presented pseudo-code, it is assumed that the spark \( \hat{x}_s \) is brighter than the spark \( \hat{x}_j \).

### 3 Experimental section

To validate our algorithm, we performed two tests on six standard bound-constrained benchmarks: Sphere, Rosenbrock, Rastrigin, Griewank, Schwefel and Ackley. Lower and upper parameters’ bounds for all functions in both tests are set to \(-100\) and \(100\), respectively. Also, dimensionality for all benchmarks is set to \(30\) \((D = 30)\). We measured algorithm performance by observing two indicators - statistical mean and standard deviation of problem solutions. The values are averaged over 20 independent algorithms’ runs, as in [25].

In the result tables (Table 1 and Table 2), standard deviation values are underlined, while best results for all test categories are mark bold. All results are rounded to seven digits.

We used the same basic FWA parameter settings as in [25]: \( n = 5 \), \( m = 50 \), \( \alpha = 0.04 \), \( \beta = 0.8 \), \( \hat{A} = 40 \) and \( \hat{m} = 5 \). FAs’s specific parameters are set as: attractiveness at \( r = 0 \) \( \beta_0 \) to 0.2, initial value of randomization parameter \( \alpha \) to 0.5 and light absorption coefficient \( \gamma \) to 1.0.

The initial value of randomization parameter \( \alpha \) is gradually decreasing from its initial value during the algorithm’s execution by the following expression:

\[
\alpha(t) = (1 - (1 - ((10^{-4}/9)^{1/FE}))) \cdot \alpha(t - 1),
\]

where \( t \) is the current iteration, and \( FE \) is maximum number of function evaluations. This parameter is being decreased in order to perform fine-tuning of the search process.

We compared our approach to the original FWA [25], clonal PSO (CPSO) and standard PSO (SPSO) [32]. We conducted two tests, as in [25]. In the first test, number of function evaluations depends on the tested function, and we tested Sphere with 500,000 evaluations, Rosenbrock with 600,000 evaluations, Rastrigin with 500,000 evaluations, Griewank with 200,000 evaluations, Schwefel with 600,000 evaluations and Ackley with 200,000 evaluations. The results are presented in Table 1.

As can be concluded from results shown in Table 1, both FWA and FWA-FS outperformed CPSO and SPSO in all tests. For Rosenbrock benchmark, our FWA-FS obtained better results than the original FWA. In this case, firefly search equation enhanced the convergence speed of the algorithm.

In the second test, we conducted 10,000 function evaluations for all benchmarks. Results are shown in Table 2.

As in the case of the first test, both, the original FWA and FWA-FS obtained much better results than SPSO and CSPO for all tests and found optimal solutions on most benchmark functions in less than 10,000 function evaluations. But, in 10,000 evaluations tests, the optimization accuracy of the CPSO and SPSO was
Table 1: Mean and standard deviation with function-specific evaluation number

<table>
<thead>
<tr>
<th>Prob.</th>
<th>CSPSO</th>
<th>SPSO</th>
<th>FWA</th>
<th>FWA-FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphe.</td>
<td>0.00000</td>
<td>1.90996</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>0.00000</td>
<td>2.59464</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>Rosen.</td>
<td>33.40319</td>
<td>410.522</td>
<td>9.56949</td>
<td>6.00284</td>
</tr>
<tr>
<td></td>
<td>42.51345</td>
<td>529.389</td>
<td>12.1283</td>
<td>11.5823</td>
</tr>
<tr>
<td>Rastr.</td>
<td>0.05304</td>
<td>167.256</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>0.37068</td>
<td>42.913</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>Griew.</td>
<td>0.63240</td>
<td>2.178</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>0.32764</td>
<td>42.913</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>Schw.</td>
<td>0.09509</td>
<td>0.33599</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>0.37661</td>
<td>0.7752</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>Ackl.</td>
<td>1.68364</td>
<td>12.3654</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>1.31786</td>
<td>1.2653</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

Table 2: Mean and standard deviation with 10,000 function evaluations

<table>
<thead>
<tr>
<th>Prob.</th>
<th>CSPSO</th>
<th>SPSO</th>
<th>FWA</th>
<th>FWA-FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphe.</td>
<td>11857.4</td>
<td>24919.0</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>3305.973</td>
<td>3383.2</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>Rosen.</td>
<td>2750997</td>
<td>557194</td>
<td>19.3833</td>
<td>16.9015</td>
</tr>
<tr>
<td></td>
<td>1741747</td>
<td>9604216</td>
<td>11.9437</td>
<td>12.7833</td>
</tr>
<tr>
<td>Rastr.</td>
<td>10940.1</td>
<td>24013.0</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>3663484</td>
<td>4246961</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>Griew.</td>
<td>3.45727</td>
<td>7.12597</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>0.011027</td>
<td>0.96578</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>Schw.</td>
<td>8775860</td>
<td>6743699</td>
<td>4.35373</td>
<td>4.11783</td>
</tr>
<tr>
<td></td>
<td>1217609</td>
<td>5977700</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>Ackl.</td>
<td>15.90766</td>
<td>18.4233</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td></td>
<td>1.19608</td>
<td>0.503372</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

not satisfying within 1,000 evaluations. Our FWA-FS showed better performance than the FWA in statistical mean indicator in Rosenbrock and Schwefel tests. However, original FWA obtained better results than FWA-FS for standard deviation indicator in Rosenbrock benchmark.

4 Conclusion

In this paper, we showed hybridized fireworks algorithm with firefly search (FWA-FS) for global optimization. We tested our approach on six standard bound-constrained benchmarks. The results of the research reported in this paper show that the FWA-FS metaheuristic has potential when tackling high-dimensionality global optimization problems.

We conducted two types of experiments and performed a comparative analysis with other state-of-the-art algorithms on the same benchmark set, as reported in [25]. Comparative analysis with the original FWA is especially important, because it shows that the firefly search method obtains better performance than Gaussian explosion which was applied in the basic FWA.

FWA-FS was implemented only for global optimization. There is a large scope of continuous and combinatorial optimization problems, with or without constraints, on which FWA-FS should be applied and it may be direction of the future research process.

Acknowledgements: This research is supported by Ministry of Education and Science of Republic of Serbia, Grant No. III-44006.

References:


