*algorithms*

MDPI

*Article*

# An Improved Fireworks Algorithm Based on Grouping Strategy of the Shuffled Frog Leaping Algorithm to Solve Function Optimization Problems

**Yu-Feng Sun [1], Jie-Sheng Wang [1,2,*] and Jiang-Di Song [1]**

[1] School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114044, China; syf95411@126.com (Y.-F.S.); sjd2011@163.com (J.-D.S.)

[2] National Financial Security and System Equipment Engineering Research Center, University of Science and Technology Liaoning, Anshan 114044, China

* Correspondence: wang_jiesheng@126.com or wangjiesheng@ustl.edu.cn; Tel.: +86-412-2538246

Academic Editor: Paul M. Goggans

Received: 10 October 2015; Accepted: 21 December 2015; Published: 1 April 2016

**Abstract:** The 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. By introducing the grouping strategy of the shuffled frog leaping algorithm (SFLA), an improved FA-SFLA hybrid algorithm is put forward, which can effectively make the FA jump out of the local optimum and accelerate the global search ability. The simulation results show that the hybrid algorithm greatly improves the accuracy and convergence velocity for solving the function optimization problems.

**Keywords:** fireworks algorithm; shuffled frog leaping algorithm; grouping strategy; function optimization

## 1. Introduction

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

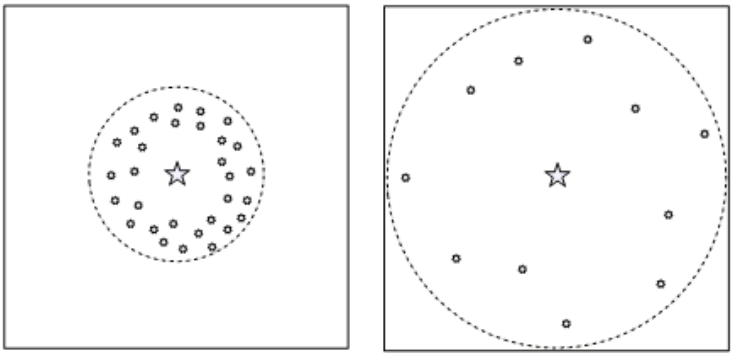
The fireworks algorithm (FA) is a new swarm intelligence algorithm that was proposed by Tan in 2010 [7], which has excellent optimization performance and aroused widespread concern in the world [8]. A hybrid fireworks-differential evolution (FWA-DE) algorithm was proposed by introducing the mutation operator, crossover operator and selection operator in the DE algorithm [9]. An enhanced fireworks algorithm (EFWA) was put forward by improving the minimum radius detection, mapping rules and spark selection strategy of the explosion sparks [10]. An adaptive firework algorithm (AFWA) was proposed to carry out the self-tuning for the blast radius [11]. That is to say, the distance between the optimal individual and the discussed individual is set as the next blast radius of the

best individual. Through this kind of adaptive step size adjustment, the improved FWA shows good optimization performance. A dynamic search firework algorithm (DFWA) was proposed, which divides the fireworks population into core fireworks with optimal fitness value and non-core fireworks. By doing so, the former carries out the local search and the latter proceeds with the global search effectively [12]. In this paper, by balancing the global exploration and local search capability of FA through adjusting the explosion of fireworks, an improved fireworks algorithm based on grouping strategy of SFLA is proposed. The simulation experiments are carried out by comparing it with the SFLA and FA. The simulation results show that the hybrid algorithm can make the FA jump out of local optimum effectively and achieve the global convergence quickly. The paper is organized as follows. In Section 2, the fireworks algorithm is introduced. The FA-SFLA hybrid algorithm is introduced in Section 3. The simulation experiments and results analysis are introduced in details in Section 4. Finally, the conclusion illustrates the final part.

## 2. Fireworks Algorithm

### *2.1. Basic Principle of FA*

The 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.



(**a**) (**b**)

**Figure 1.** Two kinds of different phenomenon after fireworks explode. (**a**) Good fireworks explode; (**b**) bad fireworks explode.

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. It can been seen form Figure 1 that good quality fireworks explosions will produce a lot of Mars concentrated in the center of the explosion and poor quality fireworks after the explosions will only produce less and messy Mars in space. Seen from the perspective of the algorithm searching, when producing a good fireworks explosion, Mars should be located near the optimal position, which is conducive to using more Mars for searching the local area of fireworks. A bad fireworks explosion means that the optimal position may be a great distance from the fireworks, so its search space will be larger.

When fireworks explode, the produced Mars are spread in the air and are filled with the local neighborhoods. In order to obtain the point $x$ in the target function $f(x) = y$, fire and explosion can be made in a potential searching area continually until Mars hits the point $x$ or very close to the point $x$. In order to make each fireworks explode, $N$ locations of the explosion are selected firstly. Then $N$ fireworks are detonated. According to the results after the explosion of fireworks, the position is

roughly estimated until the best position is obtained. Otherwise, $N$ different explosion locations are selected and fireworks explosions are produced again. The flowchart of FA is shown in Figure 2.

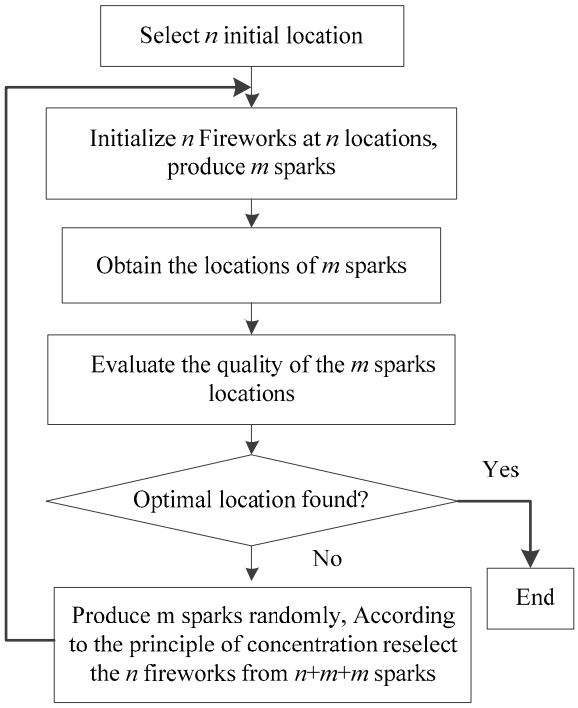



**Figure 2.** Flowchart of fireworks algorithm operation.

## *2.2. Design of FA*

### 2.2.1. Description of FA

FA is to solve a kind of optimization problem $\min f(x)$, $x_{\min} < x < x_{\max}$, where $x = x_1, x_2, \cdots, x_d$ represents a potential solution. $f(x)$ is the target function, $x_{\min}$ and $x_{\max}$ represent the scope of the solution space. Thus, the number of sparks produced by each firework $x_i$ is described as follows:

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

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

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

where $y_{\max}$ is the maximum of objective functions under the worst case in $n$ fireworks, and $\varepsilon$ is the minimum computer constant.

In order to avoid worse cases under bad firework explosions, its scope $s_i$ is defined as:

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

where $a$ and $b$ are fixed constant parameters.

The explosion amplitude of each firework is defined as follows:

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

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

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

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

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

2.2.2. 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 $x_j = x_i$;
$z = round(d \times rand(0, 1))$;
Randomly select $\hat{x}_j$ with $z$ dimension and calculate shift $h = A_i rand(-1, 1)$;
For each dimension $x_k^j \in \{\ pre - selected \quad z \quad \text{dim}esion - x_j\ \}$
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 $x_k^j = x_k^{\min} + x_k^j(x_k^{\max} - x_k^{\min})$;
For each dimension $x_k^j \in \{pre - selected \quad z \quad of \quad x_j\}$
    $x_k^j = x_k^j + h$
    $\hat{x}_k^j = \hat{x}_k^j + h$
    $if\ x_k^j < x_k^{\min}$ 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 $\hat{x}_j = x_i$;
$z = round(d \times rand(0, 1))$;
Randomly select $\hat{x}_j$ with $z$ dimension;
Calculate the coefficient of Gauss explosion g $= Gaussian(1, 1)$;
For each dimension $x_k^j \in \{\ pre - selected \quad z \quad of \quad x_j\ \}$
    $\hat{x}_k^j = \hat{x}_k^j \cdot g$;
    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

2.2.3.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. From 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 for
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 $o(n + m + \hat{m})$.

Algorithm 3 summarizes the framework of FA. In each of the explosions, according to Algorithms 1 and 2, two kinds of sparks are produced, respectively. For the first kind of spark, the number of sparks and the amplitude of the explosion depend on the quality of the fireworks. In contrast, other sparks are generated by using the Gauss explosion process in a local Gauss space of the fireworks for searching. After obtaining two kinds of spark positions, $n$ locations for the next explosion are selected.

## 3. FA-SFLA Hybrid Algorithm

### *3.1. Principle of Shuffled Frog Leaping Algorithm*

In 2003, the shuffled frog leaping algorithm (SFLA) was proposed to simulate the foraging process of frogs based on the transferring information according to the thought of ethnic groups [13]. The memetic algorithm (MA) is a swarm intelligence algorithm proposed by Moscato in 1989 to solve the optimization problem through a heuristic search [14]. SFLA is a hybrid swarm optimization algorithm combining the advantage of MA and PSO algorithm to obtain the balance between the global exploration and local exploitation [15,16]. The flowchart of the SFLA is described in Figure 3.

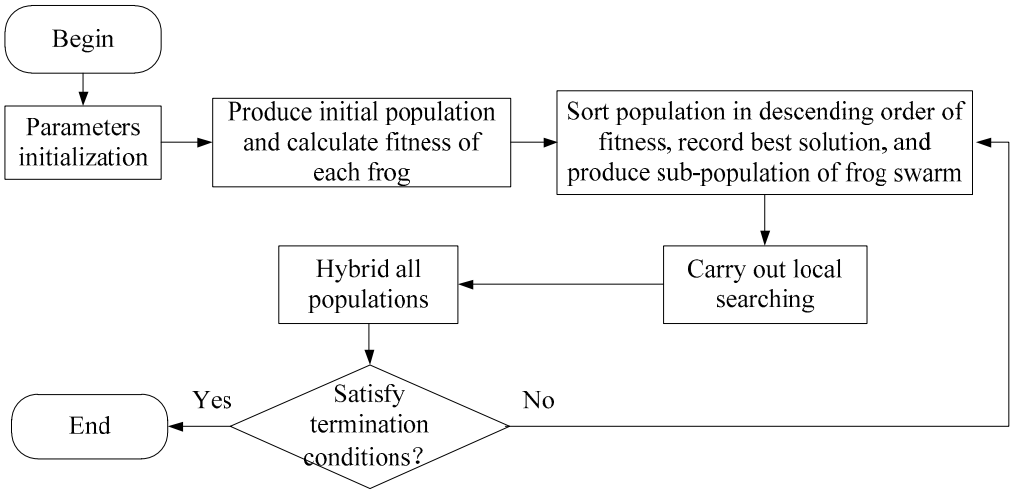



**Figure 3.** Flowchart of the shuffled frog leaping algorithm.

For $D$ dimensional problems, a frog $i$ can be expressed as $F_i = (f_{i1}, f_{i2}, \cdots, f_{iD})$. Firstly, the frog population is initialized as $S$. Then, the fitness of each frog is calculated and sorted in the descending order. Next, the entire population is divided into $m$ memeplexes, each containing $n$ frogs (*i.e.*, $S = m \times n$), in such a way that the first frog belongs to the first memeplex, the second frog goes to the second memeplex, the $m$th frog goes to the $m$th memeplex, and the $(m+1)$th frog goes back to the first memeplex, *etc*. Let $M^k$ be the set of frogs in the $k$th memeplex, and this dividing process can be described by the following expression:

$$M^k = \left\{ X_{k+m(l-1)} \in P \middle| 1 \leqslant l \leqslant n \right\}, 1 \leqslant k \leqslant m \tag{8}$$

The frog with the best and worst fitness in each subgroup are named $F_b$ and $F_w$, respectively. $F_g$ is the frog with the best fitness in the population. For each iteration in the evolution of the subgroup, the position of $F_w$ is updated based on the following equation:

$$C_i = rand() \times (F_b - F_w) \tag{9}$$

$$F'_w = F_w + C_i (\,||\, C_i \,||\, \leqslant C_{\max}) \tag{10}$$

where $rand()$ is a random number between 0 and 1 and $C_{\max}$ is the maximum allowed change of the frog's position in one jump.

If the new frog $F'_w$ is better the original frog $F_w$, it replaces the worst frog. Otherwise, $F_b$ is replaced by $F_g$, and the local search is carried out again according to Formulas (9) and (10). If no improvement is obtained in this case, the worst frog is deleted and a new frog is randomly generated to replace the worst frog $F_w$. The local search continues for a predefined number of memetic evolutionary steps $C_{\max}$ within each memeplex, and then the whole population is mixed together in the shuffling process. The local evolution and global shuffling continue until convergence iteration number $G_{\max}$ is reached.

### *3.2. FA-SFLA Hybrid Algorithm*

Although FA has a good ability to balance between global search and local search, the simulation experiments show that the running time it needs to reach its optimal value is relatively long. On the other hand, SFLA combines the calculus genetics of the mimetic algorithm (MA) and the group foraging behavior of the particle swarm optimization (PSO), which has a strong local search ability and stability through the sub cooperative search heuristic. Therefore, an improved fireworks algorithm based on the grouping strategy of SFLA is proposed based on the fireworks algorithm and the shuffled

frog leaping algorithm. The procedure of the proposed fireworks algorithm based on shuffled frog leaping algorithm (FA-SFLA) is shown in Figure 4.

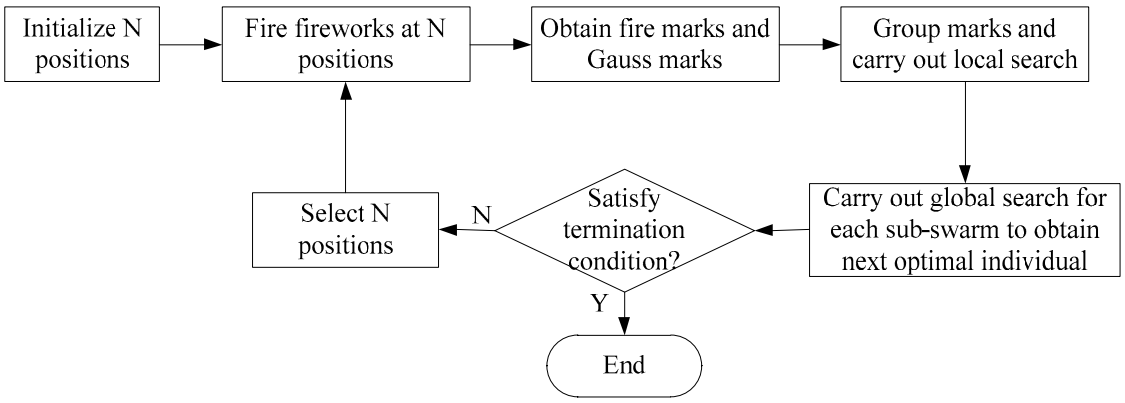



**Figure 4.** Flowchart of FA-SFLA algorithm.

The proposed FA-SFLA algorithm selects a certain number of individuals as the next generation of fireworks from the explosion spark, Gaussian mutation sparks and fireworks. The selection operator of sun-generation fireworks is based on the calculation of the original FA. Considering the number of the optimized spark generations located between maximum and minimum values, the grouping strategy of SFLA is introduced. Set the maximum value as $P_b$, the minimum value as $P_s$ and the optimal value as $P_g$. The candidate individuals are grouped to loop for the local search. According to the initial spark selection rule, the updating module is described as follows:

$$P_g = P_s - c \cdot r \cdot (P_b - P_s) \tag{11}$$

where $r$ is a random number located in the scope $[0, 1]$, and the coefficient factor $c$ is a constant and $c \in [1, 2]$. By introducing the group searching mechanism of SFLA, the search area of the FA is increased to a certain extent, so the proposed FA-SFLA can quickly jump out of local minima, explore towards the global optimum, and thus greatly improve the efficiency to obtain the global optimum.

## 4. Simulation Results and Analysis

In order to discuss the performance influence of FA-SFLA, four benchmark functions (Sphere, Griewank, Rosenbrock and Rrastrigin) shown in Table 1 are adopted to carry out the simulation experiments with FA and SFLA. The algorithm convergence, robustness, efficiency and performance of the proposed improved FA are compared with FA and SFLA on two performance indexes (average cost function and corresponding standard deviation) after simulation 30 times. The initialization of algorithm parameters are shown in Table 2. The simulation results are shown in Table 3, and the simulation curves under FA and FA-SFLA are shown in Figure 5a–d.

From the simulation results, it can be seen that the operation time of the proposed FA-SFLA hybrid algorithm was less than FA and SFLA. On the other hand, the optimal solution is less than or equal to the optimal solution of SFLA. This indicates that the FA-SFLA has feasibility and high efficiency for optimal seeking. By comparing the simulation results between FA and FA-SFLA, for the first three kinds of single peak function, the convergence velocity of FA-SFLA algorithm is relatively large. Except for the Griewank function, three other functions have the delay phenomena in later iterations but did not affect the polymorphic Rastrigin function converging to the optimum. The convergence velocity of the sphere function in FA and FA-SFLA is almost simultaneous, and Rosenbrock function with the use of the FA-SFLA algorithm exists apparent hysteretic convergence phenomena.

**Table 1.** Testing functions.

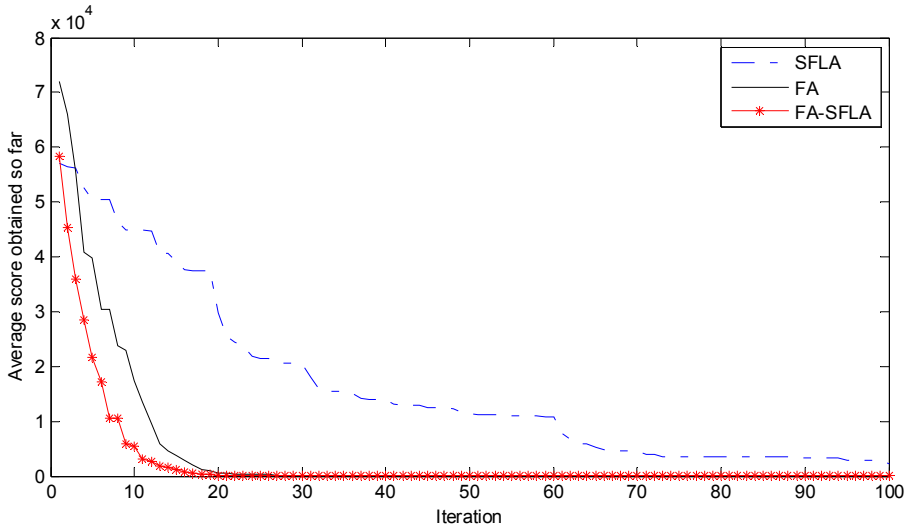
| Function Name | Expression | Scope | Dimension |
|---|---|---|---|
| F1 Sphere | $\sum_{i=1}^{n} x_i$ | $[-100,100]^n$ | 30 |
| F11 Griewank | $\sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ | $[-32,32]^n$ | 30 |
| F5 Rosenbrock | $\sum_{i=1}^{n-1} \left[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right]$ | $[-30,30]^n$ | 30 |
| F9 Rrastrigin | $\sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$ | $[-5.12,5.12]^n$ | 30 |

**Table 2.** Parameter set-up of three optimization algorithms.

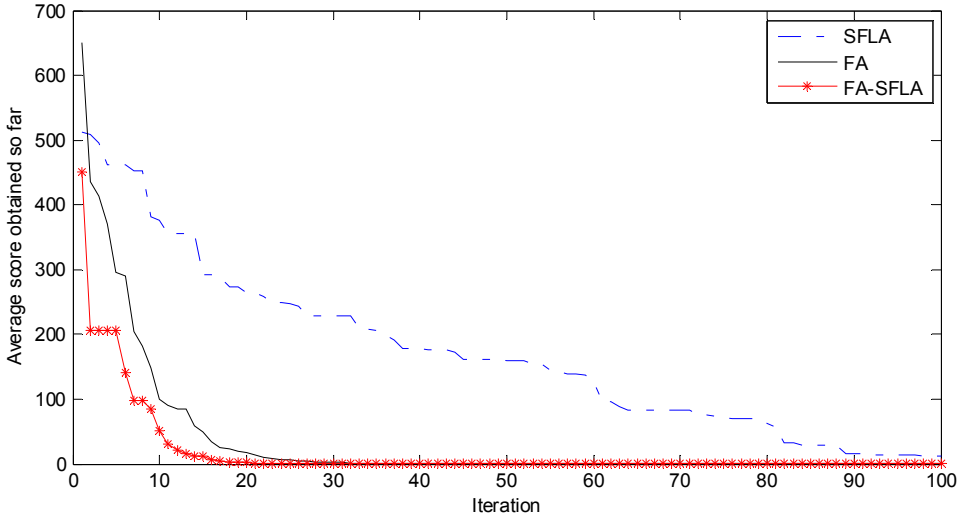
| Parameters Set-Up of FA and FA-SFLA | | Parameters Set-Up of SFLA | |
|---|---|---|---|
| **Parameter** | **Value** | **Parameter** | **Value** |
| Number of fireworks | 8 | Number of population grouping | 8 |
| Number of offspring | 64 | Number of frogs in each group contains a number of frogs | 12 |
| Dimension of optimized problem | 30 | Iteration number in inner group | 15 |
| Number of maximum iterations | 100 | Maximum step-length | 100 |
| Number of maximum evaluations | 10,000 | Amount of total population evolution | 10,000 |
| Coefficient factor | 1.5 | Dimension of optimized problem | 30 |

**Table 3.** Simulation results under three optimization algorithms.

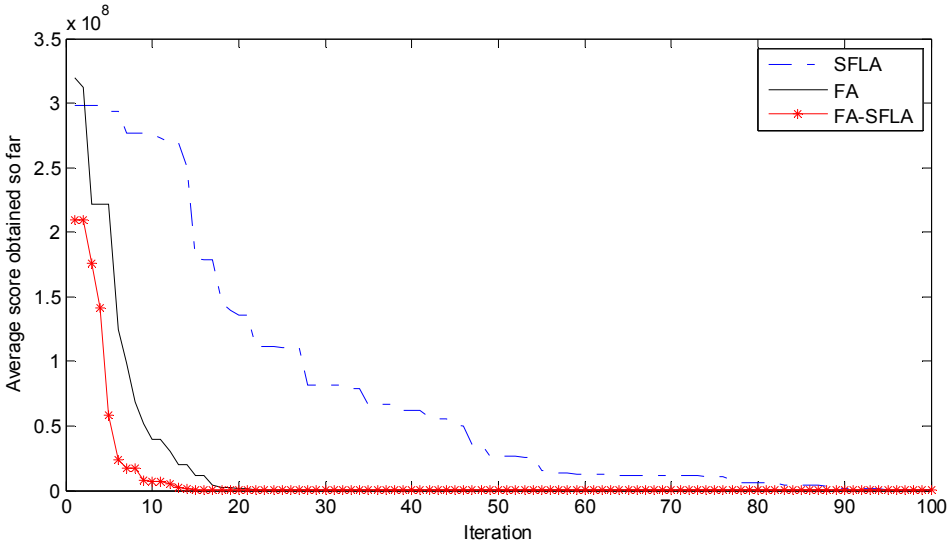
| Function | Algorithms | Average Cost Function | Standard Deviation |
|---|---|---|---|
| Sphere | FA | $2.62 \times 10^{-18}$ | $8.75 \times 10^{-19}$ |
| | FA-SFLA | $1.38 \times 10^{-30}$ | $5.21 \times 10^{-30}$ |
| | SFLA | $8.25 \times 10^{-14}$ | $6.73 \times 10^{-14}$ |
| Griewank | FA | 0.006437 | 0.008332 |
| | FA-SFLA | 0.000352 | 0.002783 |
| | SFLA | 0.02414 | 0.03564 |
| Rosenbrock | FA | 35.65643 | 15.53476 |
| | FA-SFLA | 28.54633 | 0.84352 |
| | SFLA | 58.23524 | 46.23514 |
| Rrastrigin | FA | 3.84563 | 2.45623 |
| | FA-SFLA | 0 | 0 |
| | SFLA | 45.43652 | 13.32451 |




(**a**)

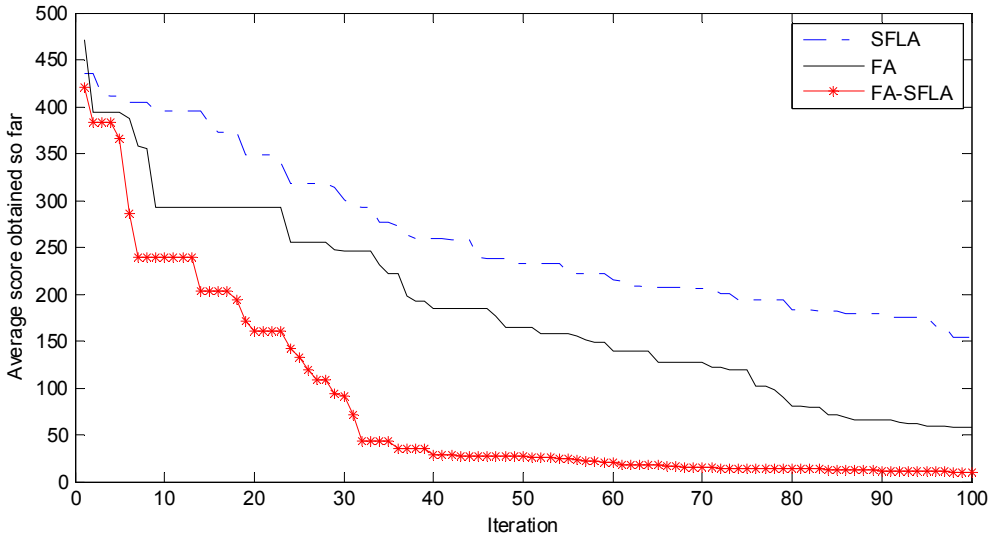



(**b**)




(**c**)

**Figure 5.** *Cont.*




(**d**)

**Figure 5.** Simulation comparison curves of FA-SFLA. (**a**) Sphere function; (**b**) Griewank function; (**c**) Rosenbrock function; (**d**) Rastrigin function.

## 5. Conclusions

The fireworks algorithm (FA) is a new parallel diffuse optimization algorithm to simulate the fireworks explosion phenomenon by means of adjusting the explosion mode of fireworks bombs. In order to realize the balance between global exploration and local searching, an improved fireworks algorithm based on the grouping strategy of the shuffled frog leaping algorithm (SFLA) is proposed in this paper. The simulation experiments are carried out on four benchmark functions (Sphere, Griewank, Rosenbrock and Rrastrigin). The algorithm convergence, robustness, efficiency and performance of the proposed improved FA are compared with FA and SFLA on two performance indices (average cost function and corresponding standard deviation) after simulation 30 times. These data should be compared with FA and SFLA. The simulation results show that the proposed method can effectively make the FA jump out of the local optimum and accelerate the global search ability. The simulation results show that the hybrid algorithm greatly improves the accuracy and convergence velocity for solving the function optimization problems.

**Acknowledgments:** This work is partially supported by the National Key Technologies R & D Program of China (Grant No. 2014BAF05B01), 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 the National Financial Security and System Equipment Engineering Research Center (Grant No. USTLKEC201401).

**Author Contributions:** Yu-Feng Sun participated in the draft writing and critical revision. Jie-Sheng Wang participated in the concept, design, interpretation and commented on the manuscript. Jiang-Di Song participated in the data collection, analysis and algorithm simulation of this paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Ren, Y.; Wu, Y. An efficient algorithm for high-dimensional function optimization. *Soft Comput.* **2013**, *17*, 995–1004. [CrossRef]
2. Yuan, Z.; Marco, A.; de Oca, M.; Birattari, M.; Stützle, T. Continuous optimization algorithms for tuning real and integer parameters of swarm intelligence algorithms. *Swarm Intell.* **2012**, *6*, 49–75. [CrossRef]

3. Chandra Mohan, B.; Baskaran, R. A survey: Ant colony optimization based recent research and implementation on several engineering domain. *Expert Syst. Appl.* **2012**, *39*, 4618–4627. [CrossRef]
4. Vallada, E.; Ruiz, R. A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. *Eur. J. Oper. Res.* **2011**, *211*, 612–622. [CrossRef]
5. Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ, USA, 27 November–1 December 1995; pp. 1942–1948.
6. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [CrossRef]
7. Tan, Y.; Zhu, Y.C. Fireworks algorithms for optimization. *LNCS* **2010**, *6145*, 355–364.
8. Gao, H.; Diao, M. Cultural firework algorithm and its application for digital filters design. *Int. J. Model. Identif. Control* **2011**, *14*, 324–331. [CrossRef]
9. Zheng, Y.J.; Xu, X.L.; Ling, H.F. A hybrid fireworks optimization method with differential evolution operators. *Neurocomputing* **2012**, *148*, 75–80. [CrossRef]
10. Zheng, S.Q.; Janecek, A.; Tan, Y. Enhanced fireworks algorithm. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013; pp. 2069–2077.
11. Li, J.Z.; Zheng, S.Q.; Tan, Y. Adaptive fireworks algorithm. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation, Beijing, China, 6–11 July 2014; pp. 3214–3221.
12. Zheng, S.Q.; Janecek, A. Dynamic search in fireworks algorithm. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation, Beijing, China, 6–11 July 2014; pp. 3222–3229.
13. Eusuff, M.M.; Lansey, K.E. Optimization of water distribution network design using shuffled frog leapind algorithm. *J. Water Resour. Plan. Manag.* **2003**, *129*, 210–225. [CrossRef]
14. Moscato, P. *On Evolution Search Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms*; Caltech Concurrent Computation Program Report 826; California Institute of Technology: Pasadena, CA, USA, 1989; pp. 16–20.
15. Elbeltagi, E.; Hezagy, T.; Grierson, D. Comparison among five evolutionary-based optimization algorithms. *Adv. Eng. Inform.* **2005**, *19*, 43–53. [CrossRef]
16. Rahimi-Vahed, A.; Mirzaei, A.H. A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem. *Comput. Ind. Eng.* **2007**, *53*, 642–666. [CrossRef]