Enhancing Interaction in the Fireworks Algorithm by Dynamic Resource Allocation and Fitness-Based Crowdedness-Avoiding Strategy

Junzhi Li and Ying Tan
Key Laboratory of Machine Perception (Ministry of Education), Department of Machine Intelligence,
School of Electronics Engineering and Computer Science, Peking University, Beijing, 100871, P.R. China,
Email:{ljz,ytan}@pku.edu.cn

Abstract—**The fireworks algorithm (FWA) is a newly proposed nature-inspired swarm intelligence algorithm. In this paper, two novel mechanisms are proposed to enhance the exploration capability by means of interaction among fireworks in the FWA. The dynamic resource allocation allows the algorithm to allocate the resource (the number of sparks) adaptively according to the search results and fitness ranking of the fireworks. The fitness-based crowdedness-avoiding strategy is proposed to improve the diversity of the fireworks population by sharing the fitness information among the fireworks. Experimental results on a large variety of test functions indicate that the proposed measures significantly improve the performance of the FWA, especially on complex objective functions.**

I. Introduction

The fireworks algorithm (FWA) is a newly proposed swarm intelligence algorithm which has proven to be very useful in many applications, such as multilevel image thresholding [1], RFID network planning [2], multi-satellite control resource scheduling [3], constrained portfolio optimization [4], regional seismic waveform [5], modern web information retrieval [6], gamma-ray spectrum fitting for radioisotope identification [7], de novo motif prediction [8], thermal unit commitment [9], privacy preserving [10], etc.

The FWA searches for the optimal solution(s) in the feasible space by generating sparks around the fireworks and selecting fireworks from the sparks. Previous works include improving the explosion ampilitude [11]–[13], the mutation operator [14] and the selection operator [11], [15], [16].

As a swarm algorithm, the performance of the FWA relies also largely on the interaction or information interchange among these fireworks. Some research works have been conducted to enhance the interaction in the FWA [16]–[18].

A recent study [15] has revealed by detailed experimental analyses that the cooperation among fireworks is not efficient enough in the previous FWA versions and proposed a cooperative framework to improve the information inheritance and interchange of the fireworks. The cooperative framework greatly promoted the efficiency of the fireworks, especially of these non-core fireworks.

In this paper, two novel mechanisms are proposed to further enhance the interaction among fireworks so that the exploration capability of the FWA can be improved. Dynamic resource allocation is adopted for letting more promising fireworks have more resource, i.e., the number of sparks. Fitness-based crowdedness-avoiding strategy is adopted for reinitializing these hopeless fireworks. By these two mechanisms, the odds of finding the global optimal solution is greatly improved.

The remainder of this paper is organized as follows. The previous resource allocation mechanism in the FWA is analyzed and the new dynamic resource allocation is proposed in Section II. The distance based crowdedness-avoiding strategy is analyzed and the new fitness-based crowdedness-avoiding strategy is proposed in Section III. The complete framework of the new fireworks algorithm is described in Section IV. Experimental results and discussions are presented in Section V. Finally, Section VI concludes this paper.

II. Dynamic Resource Allocation

In most of the previous FWA versions [19]–[21], the number of explosion sparks for each firework is calculated by the following formula:

$$\lambda_i = \hat{\lambda} \cdot \frac{\max\limits_j(f(\mathbf{X}_j)) - f(\mathbf{X}_i)}{\sum\limits_k (\max\limits_j(f(\mathbf{X}_j)) - f(\mathbf{X}_k))}, \tag{1}$$

where $\mathbf{X_i}$ is the location of the ith firework, $\hat{\lambda}$ is a constant parameter which controls the total number of explosion sparks in one generation.

The idea behind this formula was to make fireworks with better fitness have more explosion sparks to search the local area more thoroughly.

However, a previous work [22] has already analyzed and pointed out some problems of this equation:

1) The number of explosion sparks for the fireworks are not stable. The fitness values fluctuate fiercely with different objective functions and different locations. As a result, there is no regularity in the number of explosion sparks according to Eq. (1).
2) The number of explosion sparks for the firework with the best fitness has no significant advantage over other fireworks. Suppose the worst fitness among these fireworks $\max\limits_j(f(\mathbf{X}_j))$ is very large, then the resource of the best firework is in the worst case only a quarter of the total resource.

It is concluded that, the number of explosion sparks for each firework should rather depend on the ranking of its fitness value than the fitness value itself.

In this paper, we adopt the power law distribution [23], which is simple and very common in nature and human society, to determine the number of sparks for each firework:

$$\lambda_r = \hat{\lambda} \cdot \frac{r^{-\alpha}}{\sum\limits_{i=1}^{\mu} i^{-\alpha}} \tag{2}$$

where r is the fitness ranking of this firework, μ is the total number of the fireworks, and $\alpha > 1$ is a parameter to control the shape of the distribution. The larger α is, the more explosion sparks good fireworks generate.

Based on this distribution, a dynamic resource allocation is proposed. Although the fireworks with better fitness deserve more resource, once they approach the local optimal point and cannot make further improvement, there is no need to give them more resource. In this case, the algorithm should spare their resource to other hopeful fireworks.

Algorithm 1 Dynamic Resource Allocation

Require: fireworks' number μ

for $i = 1$ **to** μ **do**
  if $f(\mathbf{X}_i^g) >= f(\mathbf{X}_i^{g-1})$ **then**
    $c_i \leftarrow c_i + 1$
  else
    $c_i \leftarrow 0$
  end if
  if $\lambda_i - 2^{c_i} > 1$ **then**
    $\lambda_i \leftarrow \lambda_i - 2^{c_i}$
    $spare \leftarrow 2^{c_i}$
  else
    $\lambda_i \leftarrow 1$
    $spare \leftarrow \lambda_i - 1$
  end if
  for $j = 1$ **to** μ **do**
    if $j <> i$ **then**
      $\lambda_j \leftarrow \lambda_j + spare/(\mu - 1)$
    end if
  end for
end for

In Algorithm 1, $\mathbf{X}_i^g$ is the position of the ith firework in generation g. c_i is the number of how many generations in a row the ith firework failed to find a better solution. The spared resource 2^{c_i} will be uniformly distributed to other fireworks.

If the rate of finding better solutions of a firework begin to suffer, the resource of this firework will gradually reduce to one. In this way, the algorithm won't waste much resource on hopeless fireworks.

Fig. 1 shows how the number of sparks for each firework changes according to dynamic resource allocation on a multimodal function (function 28 in Table I). Firework 3 was not initialized in a promising area, and thus its number of sparks was small. Then it was reinitialzed (cf. Section III) in a promising area, and its number of sparks gradually increased. Finally, when it arrived at the local minimal point, it ceased to improve its fitness, thus its number of sparks gradually decreased to one.

III. Fitness-Based Crowdedness-Avoiding Strategy

In the cooperative framework for the fireworks algorithm [15], a simple cooperative mechanism was proposed, which is called the crowdedness-avoiding strategy. If the distance between a certain firework and the best firework is smaller than ten times the amplitude of the best firework, this firework will be reinitialized. By such a mechanism, the algorithm can avoid having multiple fireworks searching the same area.

However, this mechanism is still not efficient enough. The explosion amplitude of the core firework is dynamic [24], which is large at early phases and small at late phases. So, at early phases the non-core fireworks have to be frequently reinitialized which cannot search for promising areas and at late phases they have no chance to be reinitialized which will be easily trapped in local minimal areas.

Here we introduce a new kind of crowdedness-avoiding strategy called fitness-based crowdedness-avoiding which uses the fitness information of the fireworks to improve the probability of finding the global optimum. The idea is very simple: if a certain firework cannot catch up with the best fitness among fireworks with the current speed of improvement, it will be reinitialized.

Algorithm 2 Fitness-Based Crowdedness-Avoiding

Require: maximal generation number g_{max}, fireworks number μ

for $i = 1$ **to** μ **do**
  if $f(\mathbf{X}_i^g) < f(\mathbf{X}_i^{g-1})$ **then**
    $imp(i) \leftarrow f(\mathbf{X}_i^{g-1}) - f(\mathbf{X}_i^g)$
  end if
  if $imp(i) \cdot (g_{max} - g) < f(\mathbf{X}_i^g) - \min\limits_j(f(\mathbf{X}_j^g))$ **then**
    reinitialize the ith firework.
  end if
end for

Usually, the improvement reduces with the search process. At early phases, the improvements of non-core fireworks are comparatively large, so that they will not be frequently reinitialized and can exploit deeper of a local area. While at late phases, the improvements of non-core fireworks are comparatively small, and the remaining generation is fewer, so they will begin to be reinitialized to search new areas if the current one is considered not promising by the algorithm.

In this way, the algorithm not only avoids searching the same area with multiple fireworks but also avoids searching unpromising areas.

Fig. 2 shows how the fitness of each firework changes using fitness-based crowdedness-avoiding strategy on a multimodal function (function 28 in Table I). Firework 3 was not initialized in a promising area, and its fitness was large (bad) at the

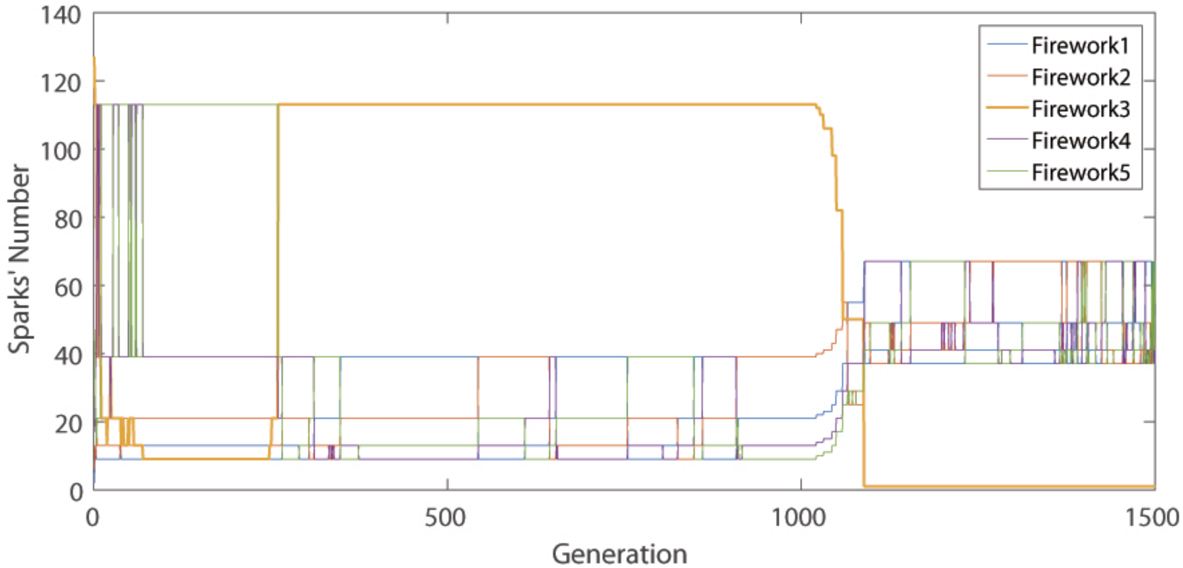

Fig. 1: Dynamic Resource Allocation

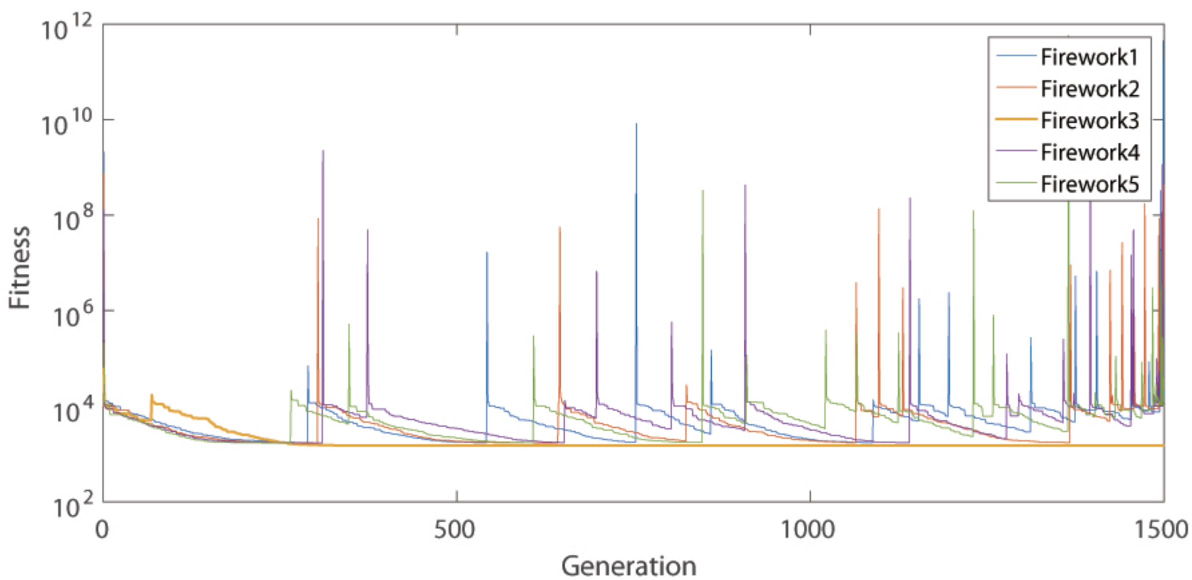

Fig. 2: Fitness-Based Crowdedness-Avoiding Strategy

beginning and more importantly, its fitness improved slowly. Thus the algorithm decided to reinitialize it. Fortunately, this time it was reinitialized in a promising area, and gradually achieved the best fitness among all the fireworks. Then other fireworks began to be reinitialized if they cannot catch up with its fitness value.

IV. Fireworks Algorithm with Dynamic Resource Allocation and Fitness-Based Crowdedness-Avoiding Strategy

Besides the two proposed mechanisms introduced above, there are several other operators adopted here to complete the algorithm.

In previous works [24] [15], only the CF's explosion amplitude is dynamically controlled. Here, all the explosion amplitudes of the fireworks are controlled in a dynamic way since they are all selected independently.

$$A_i^g = \begin{cases} A_i^1 & g = 1 \\ C_r A_i^{g-1} & f(\mathbf{X}_i^g) \geq f(\mathbf{X}_i^{g-1}) \\ C_a A_i^{g-1} & f(\mathbf{X}_i^g) < f(\mathbf{X}_i^{g-1}) \end{cases} \tag{3}$$

where A_i^g is the explosion amplitude of the ith. firework in generation g. In the first generation, the amplitude is preset to a constant number which is usually the diameter of the search space. After that, if in generation $g-1$, the algorithm found a better solution than the best in generation $g-2$, the amplitude will be multiplied by an amplification coefficient $C_a > 1$, otherwise it will be multiplied by a reduction coefficient $C_r < 1$. The best solution in generation $g-1$ is always selected into generation g as the new firework, so the right hand conditions in Eq. (3) indicate whether the best solution found has been improved.

The core idea of this dynamic explosion amplitude is described as follows: if in one generation no better solution is found, that means the explosion amplitude is too long (aggressive) and thus need to be reduced to increase the probability of finding a better solution, and otherwise it may be too short (conservative) and cannot make the largest progress and thus need to be amplified. By the dynamic control, the algorithm can keep the amplitude proper for the search. That is, the dynamic explosion amplitude is long in early phases to perform exploration, and is short in late phases to perform exploitation.

Algorithm 3 shows how the explosion sparks are generated for each firework, which is simplified compared to conventional versions [19], [20]. There was a dimension selection mechanism in the explosion operator, but it is eliminated here because it is not effective and it costs some extra time to generate random numbers.

Algorithm 3 Generating Explosion Sparks for the ith. Firework

Require: $\mathbf{X}_i$, A_i and λ_i

```
for j = 1 to λ_i do
    for each dimension k = 1, 2, ...d do
        sample η from U(−1, 1)
        s_ij^(k) ← X_i^(k) + η · A_i
    end for
end for
return all the s_ij
```

Algorithm 4 shows how the orienting mutation sparks are generated for each firework [14]. Note that only one orienting mutation spark is generated for each firework.

Algorithm 4 Generating the Orienting Mutation Spark for the ith firework

Require: $\mathbf{X}_i$, $\mathbf{s}_{ij}$, λ_i and σ

1. Sort the sparks by their fitness values $f(\mathbf{s}_{ij})$ in the ascending order.
2. $\boldsymbol{\Delta}_i \leftarrow \frac{1}{\sigma\lambda_i}(\sum_{j=1}^{\sigma\lambda_i} \mathbf{s}_{ij} - \sum_{j=\lambda_i-\sigma\lambda_i+1}^{\lambda_i} \mathbf{s}_{ij})$
3. $\mathbf{M}_i \leftarrow \mathbf{X}_i + \boldsymbol{\Delta}_i$
4. **return** $\mathbf{M}_i$

Algorithm 5 shows the complete fireworks algorithm proposed in this paper.

Algorithm 5 Fireworks Algorithm with Dynamic Resource Allocation and Fitness-Based Crowdedness-Avoiding Strategy

```
Randomly initialize μ fireworks in the search space.
Evaluate the fireworks' fitness.
repeat
    for i = 1 to μ do
        Calculate λ_i according to Eq.(2) and Algorithm 1 .
        Calculate A_i according to Eq.(3).
        Generate explosion sparks according to Algorithm 3.
        Generate orienting mutation sparks according to Algorithm 4.
        Evaluate all the fitness of the sparks.
        Select the best individual (including the ith. firework,
        its explosion sparks and mutation sparks) as the ith.
        firework of next generation.
    end for
    Perform the fitness-based crowdedness-avoiding strategy according to Algorithm 2.
until termination criteria is met.
return the position and the fitness of the best individual.
```

V. Experiments and Discussion

In this section, a set of experiments is conducted on CEC13 single objective optimization benchmark suite [25]. This benchmark suite includes unimodal functions, multimodal functions and composition functions, shown in Table I. In the following experiments, the dimensionality of these functions is $D = 30$. All the algorithms are run 51 times for each function and the maximal number of evaluations of each run is $10000D$.

TABLE I: 28 benchmark functions of IEEE CEC'2013

	No.	Name
Unimodal Functions	1	Sphere Function
	2	Rotated High Conditioned Elliptic Function
	3	Rotated Bent Cigar Function
	4	Rotated Discus Function
	5	Different Powers Function
Basic Multimodal Functions	6	Rotated Rosenbrocks Function
	7	Rotated Schaffers F7 Function
	8	Rotated Ackleys Function
	9	Rotated Ackleys Function
	10	Rotated Griewanks Function
	11	Rastrigins Function
	12	Rotated Rastrigins Function
	13	Non-Continuous Rotated Rastrigins Function
	14	Schwefel's Function
	15	Rotated Schwefel's Function
	16	Rotated Katsuura Function
	17	Lunacek Bi_Rastrigin Function
	18	Rotated Lunacek Bi_Rastrigin Function
	19	Expanded Griewanks plus Rosenbrocks Function
	20	Expanded Scaffers F6 Function
Composition Functions	21	Composition Function 1 (Rotated)
	22	Composition Function 2 (Unrotated)
	23	Composition Function 3 (Rotated)
	24	Composition Function 4 (Rotated)
	25	Composition Function 5 (Rotated)
	26	Composition Function 6 (Rotated)
	27	Composition Function 7 (Rotated)
	28	Composition Function 8 (Rotated)

Parameter setting: $\mu = 5, \hat{\lambda} = 200, \sigma = 0.2, \alpha = 1.5, C_a = 1.2, C_r = 0.9$.

Basically, the new algorithm follows the cooperative framework proposed in [15], so the cooperative framework fireworks algorithm (CoFFWA) is adopted here for a comparison. In [15], experimental results have indicated that the CoFFWA is a state-of-the-art fireworks algorithm, which outperforms the previous FWA versions on CEC13 benchmark suite, such as the dynFWA [24], the AFWA [21] and the EFWA [20].

In order to test whether or not the dynamic resource allocation is effective, the CoFFWA and the FWA with dynamic resource allocation (FWA-DRA) are firstly compared. Table II shows the mean errors and standard deviations of the CoFFWA and the FWA-DRA. A set of two-sided Wilcoxon rank sum tests are also conducted to validate if their medians are significantly different with confidence level at least 95%. The p values are also shown in Table II. The significantly better results are highlighted.

The FWA-DRA performs better than the CoFFWA on 18 functions, including 3 unimodal functions, 10 multimodal functions and 5 composition functions. The dynamic resource allocation is able to give the resource to the promising fireworks and improve the efficiency of the search.

Similarly, in order to test whether or not the fitness-based crowdedness-avoiding strategy is effective, the FWA-DRA and the FWA-DRA with fitness-based crowdedness-avoiding strategy (FWA-DRA-FBCAS) are compared. Table III shows the mean errors and standard deviations of the FWA-DRA and the FWA-DRA-FBCAS. A set of two-sided Wilcoxon rank sum tests is also conducted to validate if their medians are significantly different with confidence level at least 95%. The p values are also shown in Table II. The significantly better results are highlighted.

Although there is almost no difference between the FWA-DRA and the FWA-DRA-FBCAS on unimodal functions, it can be seen that the fitness-based crowdedness-avoiding strategy can greatly improve the probability of finding the global optimum on multimodal and composition functions.

The results of the FWA-DRA-FBCAS are also compared with some other typical heuristic algorithms: standard particle swarm optimization (SPSO) [26], artificial bee colony (ABC) [27], differential evolution (DE) [28] and covariance matrix adaptation evolutionary strategy (CMA-ES) [29]. The mean error results of ABC and DE are taken from competition session papers [30], [31] respectively. The mean error results of SPSO are not reported in the competition paper [32], but can be calculated from raw data [33]. The mean error results of CMA-ES are based on the code from [34] using default settings.

The mean errors and average rankings (AR) of the 6 algorithm are presented in Table IV. The minimal mean errors on each function and the minimal average ranking are highlighted.

The proposed algorithm outperforms other typical heuristic algorithms in terms of average ranking and the times when it gets the minimal mean error, which proves it a competitive and adaptive optimization algorithm. DE takes the second place. The performances of the CoFFWA and the ABC are comparable. As for the CMA-ES, although it performs extremely well on unimodal functions, it suffers from premature convergence on some multimodal and composition functions.

VI. Conclusion

In this paper, the drawbacks of previous sparks number calculation mechanism are analyzed and a new dynamic resource allocation is proposed to make sure the sparks are allocated to the promising fireworks. The distance-based crowdedness-avoiding strategy is also analyzed and a new fitness-based crowdedness-avoiding strategy is proposed to enhance the exploration capability by avoiding both searching overlapped areas and searching unpromising areas. Experimental results indicate that the fireworks algorithm with dynamic resource allocation and distance-based crowdedness-avoiding strategy is powerful in optimizing multimodal and composition functions, which not only outperforms the state-of-the-art cooperative framework fireworks algorithm but also beats some other typical heuristic algorithms on CEC2013 single objective benchmark suite.

Acknowledgment

Prof. Y. Tan is the corresponding author. This work was supported by the Natural Science Foundation of China (NSFC) under grant no. 61375119 and Supported by Beijing Natural Science Foundation (4162029), and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China under grant no. 2015CB352302.

References

[1] M. Tuba, N. Bacanin, and A. Alihodzic, "Multilevel image thresholding by fireworks algorithm," in *Radioelektronika (RADIOELEKTRONIKA), 2015 25th International Conference*. IEEE, 2015, pp. 326–330.

[2] M. Tuba, N. Bacanin, and M. Beko, "Fireworks algorithm for RFID network planning problem," in *Radioelektronika (RADIOELEKTRONIKA), 2015 25th International Conference*. IEEE, 2015, pp. 440–444.

[3] Z. Liu, Z. Feng, and L. Ke, "Fireworks algorithm for the multi-satellite control resource scheduling problem," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 1280–1286.

[4] N. Bacanin and M. Tuba, "Fireworks algorithm applied to constrained portfolio optimization problem," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 1242–1249.

[5] K. Ding, Y. Chen, Y. Wang, and Y. Tan, "Regional seismic waveform inversion using swarm intelligence algorithms," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 1235–1241.

[6] H. A. Bouarara, R. M. Hamou, A. Amine, and A. Rahmani, "A fireworks algorithm for modern web information retrieval with visual results mining," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 6, no. 3, pp. 1–23, 2015.

[7] M. Alamaniotis, C. K. Choi, and L. H. Tsoukalas, "Application of fireworks algorithm in gamma-ray spectrum fitting for radioisotope identification," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 6, no. 2, pp. 102–125, 2015.

[8] A. Lihu and t. Holban, "De novo motif prediction using the fireworks algorithm," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 6, no. 3, pp. 24–40, 2015.

[9] L. K. Panwar, S. Reddy, and R. Kumar, "Binary fireworks algorithm based thermal unit commitment," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 6, no. 2, pp. 87–101, 2015.

[10] A. Rahmani, A. Amine, R. M. Hamou, M. E. Rahmani, and H. A. Bouarara, "Privacy preserving through fireworks algorithm based model for image perturbation in big data," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 6, no. 3, pp. 41–58, 2015.

TABLE II: Comparison between CoFFWA and FWA-DRA

	COFFWA		FWA-DRA		
F.	Mean	Std.	Mean	Std.	p
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	NaN
2	8.80E+05	4.18E+05	**5.78E+05**	**2.02E+05**	3.97E-05
3	8.04E+07	8.88E+07	**1.21E+07**	**1.35E+07**	1.38E-10
4	2.01E+03	1.37E+03	**2.34E-01**	**2.10E-01**	3.30E-18
5	**7.41E-04**	**9.82E-05**	1.67E-03	1.99E-04	3.30E-18
6	2.47E+01	2.08E+01	**1.15E+01**	**7.33E+00**	3.05E-05
7	8.99E+01	1.78E+01	**7.10E+01**	**2.08E+01**	7.28E-03
8	2.09E+01	9.79E-02	2.09E+01	8.16E-02	6.73E-01
9	2.40E+01	4.04E+00	**1.73E+01**	**3.19E+00**	2.80E-12
10	4.10E-02	2.69E-02	**2.94E-02**	**1.44E-02**	2.22E-02
11	9.90E+01	2.36E+01	1.00E+02	2.47E+01	9.73E-01
12	1.40E+02	4.06E+01	**9.78E+01**	**2.17E+01**	4.73E-08
13	2.50E+02	5.93E+01	**1.94E+02**	**3.51E+01**	9.63E-07
14	**2.70E+03**	**4.95E+02**	3.16E+03	4.17E+02	4.85E-06
15	3.37E+03	5.01E+02	3.28E+03	4.46E+02	4.38E-01
16	4.56E-01	3.15E-01	**1.07E-01**	**5.06E-02**	2.70E-15
17	1.10E+02	2.16E+01	**8.11E+01**	**1.52E+01**	1.28E-09
18	1.80E+02	4.04E+01	**8.54E+01**	**1.97E+01**	1.90E-17
19	6.51E+00	2.08E+00	**4.29E+00**	**9.92E-01**	1.68E-08
20	1.32E+01	1.01E+00	1.31E+01	1.22E+00	6.47E-01
21	**2.06E+02**	**6.14E+01**	2.10E+02	3.61E+01	1.62E-02
22	**3.32E+03**	**6.31E+02**	3.79E+03	6.73E+02	5.27E-04
23	4.47E+03	7.90E+02	**3.97E+03**	**5.73E+02**	9.69E-04
24	2.68E+02	2.19E+01	**2.50E+02**	**1.71E+01**	5.10E-08
25	2.94E+02	1.28E+01	**2.84E+02**	**9.01E+00**	1.03E-05
26	2.13E+02	4.16E+01	**2.00E+02**	**1.63E-02**	7.73E-03
27	8.71E+02	2.10E+02	**8.00E+02**	**8.25E+01**	1.16E-05
28	2.84E+02	5.41E+01	2.76E+02	6.51E+01	3.44E-01

TABLE III: Comparison between FWA-DRA and FWA-DRA-FBCAS

	FWA-DRA		FWA-DRA-FBCAS		
F.	Mean	Std.	Mean	Std.	p
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	NaN
2	5.78E+05	2.02E+05	5.91E+05	2.20E+05	9.89E-01
3	1.21E+07	1.35E+07	1.66E+07	1.84E+07	2.01E-01
4	2.34E-01	2.10E-01	2.31E-01	2.07E-01	6.93E-01
5	1.67E-03	1.99E-04	1.69E-03	1.78E-04	7.76E-01
6	1.15E+01	7.33E+00	1.13E+01	5.72E+00	9.20E-01
7	7.10E+01	2.08E+01	**5.77E+01**	**1.15E+01**	1.44E-03
8	2.09E+01	8.16E-02	**2.09E+01**	**9.10E-02**	9.60E-03
9	1.73E+01	3.19E+00	**1.52E+01**	**2.55E+00**	5.27E-04
10	**2.94E-02**	**1.44E-02**	3.87E-02	2.07E-02	2.10E-02
11	1.00E+02	2.47E+01	**6.98E+01**	**1.17E+01**	4.63E-11
12	9.78E+01	2.17E+01	**7.39E+01**	**1.26E+01**	8.97E-09
13	1.94E+02	3.51E+01	**1.31E+02**	**2.46E+01**	2.45E-13
14	3.16E+03	4.17E+02	**2.57E+03**	**3.30E+02**	4.26E-10
15	3.28E+03	4.46E+02	**2.79E+03**	**3.65E+02**	9.96E-08
16	1.07E-01	5.06E-02	**6.72E-02**	**2.87E-02**	1.49E-06
17	8.11E+01	1.52E+01	**7.49E+01**	**1.23E+01**	4.19E-02
18	8.54E+01	1.97E+01	**7.78E+01**	**1.44E+01**	4.06E-02
19	4.29E+00	9.92E-01	**3.49E+00**	**7.50E-01**	6.83E-05
20	1.31E+01	1.22E+00	1.31E+01	1.24E+00	1.00E+00
21	2.10E+02	3.61E+01	1.90E+02	3.00E+01	1.10E-01
22	3.79E+03	6.73E+02	**3.04E+03**	**4.12E+02**	2.89E-08
23	3.97E+03	5.73E+02	**3.36E+03**	**3.86E+02**	3.52E-07
24	2.50E+02	1.71E+01	**2.38E+02**	**1.54E+01**	7.02E-05
25	2.84E+02	9.01E+00	**2.78E+02**	**1.40E+01**	1.38E-02
26	2.00E+02	1.63E-02	2.00E+02	1.94E-02	1.29E-01
27	8.00E+02	8.25E+01	**7.36E+02**	**1.04E+02**	2.13E-03
28	2.76E+02	6.51E+01	**2.33E+02**	**9.52E+01**	1.74E-02

TABLE IV: Comparison with other heuristic algorithms

F.	CMA-ES	SPSO	DE	ABC	CoFFWA	FWA-DRA-FBCA
1	**0.00E+00**	**0.00E+00**	1.89E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**
2	**0.00E+00**	3.38E+05	5.52E+04	6.20E+06	8.80E+05	5.91E+05
3	**1.41E+01**	2.88E+08	2.16E+06	5.74E+08	8.04E+07	1.66E+07
4	**0.00E+00**	3.86E+04	1.32E-01	8.75E+04	2.01E+03	2.31E-01
5	**0.00E+00**	5.42E-04	2.48E-03	0.00E+00	7.41E-04	1.69E-03
6	**7.82E-02**	3.79E+01	7.82E+00	1.46E+01	2.47E+01	1.13E+01
7	**1.91E+01**	8.79E+01	4.89E+01	1.25E+02	8.99E+01	5.77E+01
8	2.14E+01	2.09E+01	2.09E+01	2.09E+01	2.09E+01	**2.09E+01**
9	4.81E+01	2.88E+01	1.59E+01	3.01E+01	2.40E+01	**1.52E+01**
10	**1.78E-02**	3.40E-01	3.24E-02	2.27E-01	4.10E-02	3.87E-02
11	4.00E+02	1.05E+02	7.88E+01	**0.00E+00**	9.90E+01	6.98E+01
12	9.42E+02	1.04E+02	8.14E+01	3.19E+02	1.40E+02	**7.39E+01**
13	1.08E+03	1.94E+02	1.61E+02	3.29E+02	2.50E+02	**1.31E+02**
14	4.94E+03	3.99E+03	2.38E+03	**3.58E-01**	2.70E+03	2.57E+03
15	5.02E+03	3.81E+03	5.19E+03	3.88E+03	3.37E+03	**2.79E+03**
16	**5.42E-02**	1.31E+00	1.97E+00	1.07E+00	4.56E-01	6.72E-02
17	7.44E+02	1.16E+02	9.29E+01	**3.04E+01**	1.10E+02	7.49E+01
18	5.17E+02	1.21E+02	2.34E+02	3.04E+02	1.80E+02	**7.78E+01**
19	3.54E+00	9.51E+00	4.51E+00	**2.62E-01**	6.51E+00	3.49E+00
20	1.49E+01	1.35E+01	1.43E+01	1.44E+01	1.32E+01	1.31E+01
21	3.44E+02	3.09E+02	3.20E+02	**1.65E+02**	2.06E+02	1.90E+02
22	7.97E+03	4.30E+03	1.72E+03	**2.41E+01**	3.32E+03	3.04E+03
23	6.95E+03	4.83E+03	5.28E+03	4.95E+03	4.47E+03	**3.36E+03**
24	6.62E+02	2.67E+02	2.47E+02	2.90E+02	2.68E+02	**2.38E+02**
25	4.41E+02	2.99E+02	2.80E+02	3.06E+02	2.94E+02	**2.78E+02**
26	3.29E+02	2.86E+02	2.52E+02	2.01E+02	2.13E+02	**2.00E+02**
27	5.39E+02	1.00E+03	7.64E+02	**4.16E+02**	8.71E+02	7.36E+02
28	4.78E+03	4.01E+02	4.02E+02	2.58E+02	2.84E+02	**2.33E+02**
AR.	4.11	4.11	3.36	3.54	3.54	**2.00**

[11] S. Zheng, A. Janecek, and Y. Tan, "Enhanced fireworks algorithm," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, June 2013, pp. 2069–2077.

[12] S. Zheng, A. Janecek, J. Li, and Y. Tan, "Dynamic search in fireworks algorithm," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, July 2014, pp. 3222–3229.

[13] J. Li, S. Zheng, and Y. Tan, "Adaptive fireworks algorithm," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, July 2014, pp. 3214–3221.

[14] J. Li and Y. Tan, "Orienting mutation based fireworks algorithm," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 1265–1271.

[15] S. Zheng, J. Li, A. Janecek, and Y. Tan, "A cooperative framework for fireworks algorithm," *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[16] B. Zhang, Y. Zheng, M. Zhang, and S. Chen, "Fireworks algorithm with enhanced fireworks interaction," *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 2015.

[17] Y. Tan, "Cooperative fireworks algorithm," in *Fireworks Algorithm*. Springer, 2015, pp. 133–149.

[18] J. Chen, Q. Yang, J. Ni, Y. Xie, and S. Cheng, "An improved fireworks algorithm with landscape information for balancing exploration and exploitation," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*. IEEE, 2015, pp. 1272–1279.

[19] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *Advances in Swarm Intelligence*. Springer, 2010, pp. 355–364.

[20] S. Zheng, A. Janecek, and Y. Tan, "Enhanced fireworks algorithm," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 2069–2077.

[21] J. Li, S. Zheng, and Y. Tan, "Adaptive fireworks algorithm," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 3214–3221.

[22] J. Liu, S. Zheng, and Y. Tan, "The improvement on controlling exploration and exploitation of firework algorithm," in *Advances in swarm intelligence*. Springer, 2013, pp. 11–23.

[23] https://en.wikipedia.org/wiki/Power_law#Power-law_probability_distributions.

[24] S. Zheng, A. Janecek, J. Li, and Y. Tan, "Dynamic search in fireworks algorithm," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 3222–3229.

[25] J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, vol. 201212, 2013.

[26] M. Clerc, "Standard particle swarm optimization, from 2006 to 2011," *Particle Swarm Central*, 2011.

[27] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of global optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[28] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[29] N. Hansen and A. Ostermeier, "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation," in *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*. IEEE, 1996, pp. 312–317.

[30] M. El-Abd, "Testing a particle swarm optimization and artificial bee colony hybrid algorithm on the CEC13 benchmarks," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 2215–2220.

[31] N. Padhye, P. Mittal, and K. Deb, "Differential evolution: Performances and analyses," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 1960–1967.

[32] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, "Standard particle swarm optimisation 2011 at CEC-2013: A baseline for future PSO improvements," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 2337–2344.

[33] http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/Shared%20Documents/CEC2013/Results-of-22-papers.zip.

[34] https://www.lri.fr/∼hansen/purecmaes.m.