Multi-Scale Collaborative Fireworks Algorithm

Yifeng Li
Key laboratory of Machine Perception (MOE)
Department of Machine Intelligence
School of Electronics Engineering and Computer Science
Peking University
Beijing, China
liyifeng@pku.edu.cn

Ying Tan
Key laboratory of Machine Perception (MOE)
Department of Machine Intelligence
School of Electronics Engineering and Computer Science
Peking University
Beijing, China
ytan@pku.edu.cn

***Abstract*—Fireworks Algorithm (FWA) is a special swarm intelligent optimization algorithm, which controls multiple subgroups of the population to search collaboratively. Instead of assigning fireworks to different local areas, we propose the multi-scale collaborative firework algorithm (MSCFWA) which helps fireworks to search at coordinated scales. Since the collaboration of search scales is accomplished by restarting or adjusting fireworks whose local search are not making meaningful progress, fireworks in MSCFWA are able to exploit different local areas independently or cooperate in the same local area with different search scales. Experimental results show that the proposed strategy stably improved the overall optimization performance of fireworks algorithm on the benchmark functions of the CEC'13 competition significantly. It also shows outstanding efficiency compared with typical swarm intelligence optimization algorithms and evolutionary algorithms.**

***Index Terms*—Fireworks Algorithm, Multi-modal Optimization, Swarm Intelligence, Evolutionary Algorithm, Multi-Scale Optimization**

I. Introduction

The black-box optimization problem is an important and difficult research direction. Mathematical methods (like Gradient-Descent and Bayes Optimization [1]) can solve problems with convex or simple objective function with extra information (gradient or objective prior). But they usually failed to be adapted for multi-modal problems with high dimensions. Based on inspiration from natural phenomenons, many evolutionary algorithms (EAs, like GA [2], ES [3]) and swarm intelligent optimization algorithms (SIOAs, like PSO [4], ACO [5]) are proposed. They are able to obtain a relatively good result by controlling a group of agents searching in the feasible space with balanced exploitation and exploration.

The fireworks algorithm (FWA) is a new SIOA proposed in 2010 [6], inspired by the phenomenon of fireworks' explosion. Its efficiency has been proven in numerous real-world applications and has been continuously improved in recent years. The core features of fireworks algorithm compared with other EAs and SIOAs are the grouped local searches and their interaction. During the optimization, each firework maintains a basic independent local search by simply generating concentrated sparks around itself. While the global optimization efficiency is guaranteed by suitable coordination methods and resource (sparks) allocation strategies. For example, the original FWA was built on an intuitive idea that fireworks with better fitness generate more sparks in a closer range for exploitation, while fireworks with worse fitness generate sparks in a boarder range for exploration.

However, collaboration strategies have not achieved yet ideal optimization efficiency. Instead, some variants of FWA with more independent fireworks showed more significant performance because they have better exploitation ability in local search. An important reason is that almost all the implementations of FWA consider fireworks to be searching in different areas, which caused difficulties in their collaboration. On the one hand, the effect of collaboration must compensate for the loss of efficiency caused by resource dispersion. Because only a small part of individuals in each generation can be directly used for searching near the current optimal location. On the other hand, there is little information that fireworks could share, because different regions can have completely different characteristics in complex objective functions.

Therefore, we propose the multi-scale collaborative firework algorithm. Instead of assigning each firework to search a local area independently, we accurately restart or adjust fireworks with insufficient potential and make different fireworks tends to collaborate on searches at different scales. In this way, fireworks can explore different local areas by themselves or exploit the same one cooperatively. Both the global optimization ability and the local optimization efficiency can be enhanced.

In the following sections, we first introduce the background of fireworks algorithm in Section II including the principle of FWA, LoTFWA which is the foundation of the proposed algorithm and some other related works. The proposed strategy of multi-scale collaborative fireworks algorithm is described in Section III, we also provide a detailed analysis and discussion of the proposed strategy. Then, the proposed algorithm is compared with other optimization algorithms on benchmark functions from CEC 2013 test suite in Section IV. Finally, we discuss and analyze the proposed method in Section V and conclude the work in Section VI.

II. Background

A. Principle of Fireworks Algorithm

Fireworks algorithms (FWA) are inspired by the explosion phenomenon of real-world fireworks. In the night sky, each firework explodes and emits a group of sparks to illuminate

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

a local area. Then the whole sky can be lit up with sufficient fireworks. Based on the observation of this phenomenon, fireworks algorithm is designed based on the following two main ideas:

- **Local Search by Explosion.** The optimization of FWA is divided into several local searches, each centered on a firework. The local areas are explored by a simple operator called explosion, in which a certain number of individuals called sparks are generated around the firework.
- **Global Coordination of Local Searches.** The specific parameters of local searches are determined by global coordination. Usually, the range size of local search and the number of allocated sparks are the core of collaboration.

Algorithm 1 gives a basic description of FWA's framework.

Algorithm 1 Framework of Fireworks Algorithm

```
Initialize $N$ fireworks $X_N$ randomly in the search space.
while termination conditions are not satisfied do
    decide spark number $\lambda_i$ and explosion amplitude $A_i$ for each firework
    for each firework $X_i$ do
        generate $\lambda_i$ spark around it within amplitude $A_i$
    end for
    select $N$ fireworks for next generation from current population
end while
```

The essential feature of FWA is the collaboration method among several isomorphic subgroups that are able to search independently, which is rarely considered in other swarm intelligence optimization algorithms or evolutionary algorithms. The local optimization method in FWA is extremely simple compared with most swarm-based optimization algorithm. In most variants of FWA, the explosion operator simply generates sparks uniformly within a certain range around the firework.

B. Loser-out Tournament based Fireworks Algorithm

Loser-out Fireworks Algorithm is proposed in 2017 [7]. Compared with other variants of FWA, it has extremely outstanding global optimization performance while maintaining high simplicity. During the optimization of LoTFWA, each firework conducts dynamic local search independently and the communication only happens in the competition strategy called loser-out tournament, in which some bad fireworks are restarted in time so the overall optimization efficiency can be improved.

For each firework, the explosion amplitude A_i is decided by the dynamic amplitude strategy from dynFWA [8], which expand or shrink the explosion ranges according to whether the firework is improved or not. A guiding spark G_i is also generated in the direction improving the firework with a high probability according to the strategy from GFWA [9]. Both methods have been analyzed and proved to be effective for local optimization. The spark number of each firework λ_i is decided by the rank of their fitness values. However, according to experimental results, when all the fireworks get the same number of sparks, the algorithm achieved the best overall performance. The local search process is described in Algorithm 2.

Algorithm 2 Local Search in LoTFWA

```
for $j = 0 \to \lambda_i$ do
    compute random bias $\eta \sim \mathcal{U}([-1,1]^{dim})$
    get one explosion spark $S_{ij} = X_i^g + A_i \times \eta$
end for
evaluate $f(S_{ij})$
compute guiding spark $G_i$ and $f(G_i)$
$X_i^{g+1} = \arg\min\{f(X_i^g), f(S_{ij}), f(G_i)\}$
```

The global coordination strategy stops and randomly restarts local searches with insufficient potential. In this way, LoTFWA avoids wasting excessive resources on some fireworks and improves the overall efficiency. The loser-out tournament strategy estimates the potential of an improved firework as if it would have the same amount of improvement in every subsequent generation, which is quite tolerant so valuable local areas will hardly be abandoned mistakenly. The loser-out tournament strategy is described in Algorithm 3.

Algorithm 3 Loser-out Tournament Strategy

```
for each firework $X_i$ do
    if $f(X_i^g) < f(X_i^{g-1})$ then
        compute improvement $I_i = f(X_i^{g-1}) - f(X_i^g)$
        if $I_i \times (g_{max} - g) < f(X_i^g) - \min_j f(X_j^g)$ then
            restart $X_i$ at random location
            re-initialize $A_i$
        end if
    end if
end for
```

Here, g_{max} is the max number of generations. f is the objective function. LoTFWA has been proved to be highly effective and fundamental, so it is used as the basis of the proposed algorithm.

C. Related Works

The idea of letting different fireworks search on different scales was reflected in the principle idea of the original fireworks algorithm [6], in which fireworks' optimization scales (amplitudes) are linearly related to their fitness values. All the classic variants of FWA like the enhanced fireworks algorithm (EFWA [10]), the dynamic fireworks algorithm (dynFWA [8]) and dynFWA with exponentially decreased dimension (eddynFWA [11]) applied similar strategies but they are later proved not effective by the bare-bone fireworks algorithm (BBFWA [12]).

In the adaptive fireworks algorithm (AFWA [13]), amplitudes are decided by the information of the local sparks' location. Some recent variants of FWA like [14] and [15] proposed more complex but flexible explosion mechanisms, which are also able to adapt the search scale according to the local characteristics of the objective function. However, they did not study the collaboration on search scales.

A cooperative framework of fireworks algorithm (CoFFWA [16]) and the loser-out tournament fireworks algorithm (LoTFWA [7]) are specially designed to enhance the collaboration between fireworks. A few algorithms based on LoTFWA like [17] are also published recently. However, all those variants focus on avoiding multiple fireworks searching in the same local area, which is quite different from the idea of the proposed method.

The multi-scale method in general optimization algorithms usually refers to methods making different levels of smooth approximation of the objective functions (A better term for it would be multi-fidelity method). A great number of algorithms are built or improved based on this idea. FWA can be explained as a more flexible way of multi-scale optimization because each local optimization has its own search scale, that is, the explosion amplitude. To our knowledge, there is no published algorithm that groups the population and decides their search scales collaboratively as our proposed method does.

III. Proposed Method

A. Multi-Scale Collaborative Fireworks Algorithm

The loser-out tournament strategy greatly strengthens FWA's global exploration ability, but the local search of each firework is completely independent so it can't help with its local exploitation ability. In order to make different fireworks collaborate, we borrow the idea of multi-scale optimization to introduce more possible relationships between local searches compared with LoTFWA. During the optimization, fireworks could be exploring different local areas, or they can be searching the same area with different scales.

The basic principle is that a firework should be searching efficiently and independently when it's possible, otherwise it will be adjusted to help another firework's search on a larger scale. To diversify the scales, the fireworks are organized according to their fitness values. And when the local search of a firework is not making good progress, it would be adjusted with reference to the better firework.

The independent local optimization in the proposed algorithm is the same with LoTFWA and it should proceed undisturbed if possible. We apply the dynamic amplitude strategy to maximize the local exploitation efficiency. In other words, the firework's amplitude in the next generation is amplified by C_a or reduced by C_r when it is improved or not respectively ($0 < C_r < 1 < C_a$). The guiding spark is also generated in the direction from bad sparks to good sparks.

In each iteration, the fireworks generate sparks and adjust themselves according to the local optimization strategy described above. According to the fitness values of sparks, the progress of the local search can be evaluated. The fireworks are sorted according to their current fitness values and visited from best to worst. In the following two cases, independent local searches of non-optimal fireworks will be coordinated.

One case where the local optimization should be stopped is consistent with the restart mechanism of LoTFWA, that is, the improving rate of local optimization is not fast enough to exceed the current best individual. In order to restart the firework, we sample a random location from the search space and set the explosion amplitude as α times the distance to the previous firework, where $\alpha > 1$. This explosion amplitude will not be too large as in LoTFWA, and also help the restarted firework cooperates with the local optimization of the previous firework better.

Another case when the local optimization should be adjusted is when the firework has not improved for many iterations and also does not match the search progress of the previous firework severely. When a firework is so lucky to be located in a very good position, it might be hard to improve for several iterations. Then its explosion amplitude reduces rapidly and its search scale becomes so small that it would affect its progress rate of later iterations. And since the firework is not improved, we are not able to restart it by the former strategy.

So when a fireworks X_i^g is not improved for certain iterations (the new firework after the local search would be X_i^{g+1}, but since it is not improved, we know $X_i^g = X_i^{g+1}$), we examine whether it is better than the worst spark of the former firework. If$f(X_i^{g+1}) < \max_j\{f(S_{i-1,j})\}$, we adjust the amplitude and location towards the former firework as equation 1 and 2.

$$A_i^{g+1} = (1-\beta) \times A_i^g + \beta \times d(X_i^g, X_{i-1}^g) \tag{1}$$

$$X_i^{g+1} = X_i^g + \beta \times (X_{i-1}^{g+1} - X_i^{g+1}) \tag{2}$$

The parameter $\psi \in [0, 1]$ decide how fast the firework should adjust towards the previous one.

The proposed fireworks algorithm with multi-scale collaborative strategy is described in Algorithm 4. Here, g is the iteration number, which is limited to g_{max}. The subscript $i = 0, 1, ..., n-1$ indicate n fireworks. During the local search, e_spark_{ij} and e_fit_{ij} are explosion sparks and their fitness values. Here, we allocate the same number of sparks for each firework. g_spark_i and g_fit_i are guiding spark and its fitness value. X_i^{t+1} is selected as the best one of the firework's explosion sparks, guiding spark and itself. After explosion and mutation in each iteration, each firework is checked by the loser-out tournament strategy and multi-scale collaborative strategy. If they are not applied, the firework's explosion amplitude is updated according to the dynamic amplitude method.

B. Analysis of Multi-scale Collaborative Strategy

When the strategy is activated for firework X_2 and X_1 is the better firework, consider the relationship between the explosion area of X_2 and the location of X_1, there are two possible situations.

If X_1 is within the explosion area of X_2, then there is a huge probability that the scale of X_2 is too large compared with X_1. Because otherwise there is very likely a spark of X_2 will fall into the explosion area of X_1 and it is better than the

Algorithm 4 Multi-Scale Collaborative Fireworks Algorithm

```
initialize $g = 0, A_i = 0, \lambda_i = M$
randomly sample $X_i^0$, evaluate $f(X_i^0)$
while termination conditions are not satisfied do
  //Local Search by Explosion, Mutation and Selection
  for each firework $X_i^g$ do
    obtain $S_{ij}$ and $f(S_{ij})$, $j = 1, ..., \lambda_i$
    obtain $M_i$ and $f(M_i)$
    select $X_i^{g+1} = \arg\min\{f(X_i^g), f(S_{ij}), f(M_i)\}$
  end for
  sort $X_i^g$ according to fitness $f(X_i^g)$
  for each firework $X_i^g$ do
    if $f(X_i^{g+1}) < f(X_i^g)$ then
      //Loser-out Tournament Strategy
      $I_i = f(X_i^g) - f(X_i^{g+1})$
      if $i = 0$ or $I_i \times (g_{max} - g - 1) > f(X_i^{g+1}) - \min_j\{f(X_j^{g+1})\}$ then
        $A_i^{g+1} = C_a \times A_i^g$
      else
        randomly sample firework $X_i^{g+1}$, compute $f(X_i^{g+1})$
        $A_i^{g+1} = \alpha \times d(X_i^{g+1}, X_{i-1}^{g+1})$
      end if
    else
      //Multi-Scale Collaborative Strategy
      if $i = 0$ or $f(X_i^{g+1}) < \max_j\{f(S_{i-1,j})\}$ or $X_i$ improved in $K$ recent iterations then
        $A_i^{g+1} = C_r \times A_i^g$
      else
        compute $A_i^{g+1}$ according to equation (1)
        compute $X_i^{g+1}$ according to equation (2)
      end if
    end if
  end for
  $g = g + 1$
end while
```

worst spark of X_1. In such a situation, the proposed strategy shrinks the explosion amplitude of X_2 and lead it to search near X_1. This situation is shown on the left side of Fig. 1.

If X_1 is not within the explosion area of X_2, through the same analysis as above we know that the overlapped explosion area of X_1 and X_2 could only take a limited proportion of them. In most situations, X_1 and X_2 are searching in different local areas but the optimization of X_2 is not progress effectively compared with X_1. The proposed strategy will enlarge the search scale of X_2 and force it to explore around X_1. This situation is shown on the right side of Fig. 1. Only with a small probability, the explosion area of X_2 takes a very small part within the explosion area of X_1, our strategy also enlarges the scale of X_2 so it can assist the local search of X_1. Such a firework will most likely be stopped by our restart strategy since it is really hard for X_2 to catch up with X_1 with both worse fitness and smaller search scale.

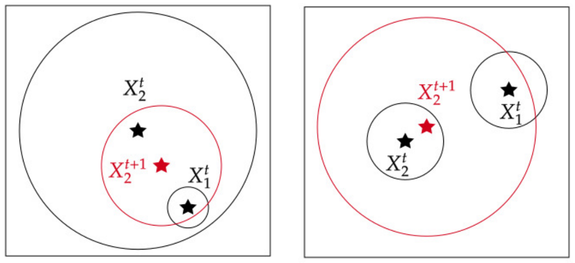

Fig. 1. The multi-scale collaborative strategy in different situations. In both subplots, firework X_2 is adjusted according to the better firework X_1.

C. Discussion

The proposed strategy is helpful for optimization in several ways.

First, the local exploitation ability of fireworks would not be hurt by the proposed algorithm. In most cases, each firework conducts efficient local optimization adapted to its local information. The loser-out tournament has been proven not able to cause little damage to local searches in [7]. The multi-scale collaborative strategy is only applied when the firework has not improved in K iterations, which means the firework seems not possible to improve at the current scale. The second condition $f(X_i^{g+1}) < \max_j\{f(S_{i-1,j})\}$ also implies the firework is not possible to exceed the previous firework because it has both worse fitness and smaller search scale. So, the adjusted fireworks have very limited potential.

Second, the new strategy makes the firework tends to spread closer with different scales. This is because when a firework is restarted or adjusted, it is ensured to cover the previous firework into its explosion area. And the adjusted firework moves towards the previous one. Although we did not force the worse spark to have a larger explosion amplitude, these strategies still help the explosion amplitudes to correlate with the fitness values because the amplitude of each firework is adjusted with reference to the better one. The moderate converge ability is very helpful for functions with objective functions with sufficient global landscape, which is more likely to be found in real optimization problems.

Finally, the proposed method can improve efficiency significantly when fireworks are located in the same area of a local optimal. When fireworks are too close to each other, the efficiency of algorithms with independent local search is hurt because they are searching in the same area while the information obtained can not be exchanged sufficiently to each other. However, with the proposed algorithm, fireworks usually have different optimization scales so they won't duplicate the local search even when they are located at the same position. If fireworks are in the same local area but could not cooperate, the worse one is very likely to trigger the restart strategy or multi-scale collaborative strategy. Then their search scale will be adjusted reasonably.

IV. Experimental Results

A. Benchmark Functions

The proposed multi-scale collaborative fireworks algorithm (MSCFWA) is tested on the benchmark functions from the CEC'13 competition. The details of the objective functions are available in [18]. We choose the CEC'13 competition test suite because LoTFWA is proposed and tuned on these functions. So it can be used as a competitive comparison for our method.

B. Parameter Settings

Since it is designed based on LoTFWA and mainly compared with LoTFWA, we set the main parameters which appear in both algorithms according to [7]. In this situation, the efficiency of LoTFWA is maximized, thus the improvement effect of the new algorithm can be shown by comparison. The parameters introduced in the proposed algorithm are determined by analysis and experiments.

Basic parameters from LoTFWA are kept in the proposed algorithm. It controls 5 fireworks and generates 300 sparks in explosion. The coefficient of dynamic amplitude are $C_r = 0.9$ and $C_a = 1.2$. The parameter in guiding spark mutation is also the same as in LoTFWA, that is, $\sigma = 0.2$.

There are three new parameters introduced in the proposed strategy, that is, K, α, and β.

Parameter K is the maximum number of consecutive failures for a stable local search. Due to the nature of the dynamic explosion amplitude, fireworks tend to repeat improving and non-improving during stable optimization. So K can not be too small. Larger K means better tolerance for local search's instability. We choose $K = 10$ to obtain a better performance according to experiments.

Parameter α decides the amplitudes of the restarted fireworks. When $\alpha \geq 1$, the restarted firework's explosion covers the referenced better firework. When α is larger, the global exploration ability is stronger. When α is closer to 1, the restarted searches are more focused on the previous firework, so the exploitation ability is stronger. We choose $\alpha = 1.2$ for better performance.

Parameter $\beta \in [0, 1]$ is the most important parameter for the proposed strategy, which controls the speed of firework collaborating towards the better firework. With a large β, the fireworks' optimization areas overlap quickly and exploitation ability is enhanced. While with a small β, fireworks could continue on its local optimization for more generations, thus the global exploration ability is ensured. We choose $\beta = 0.1$. On the one hand, this value corresponds to a better optimization performance in experiments. On the other hand, it is also consistent with the shrinking speed of amplitude when a firework fails to improve.

C. Experimental Results

The experiments are run on a Ubuntu 18.04 operating system with Intel(R) Xeon(R) CPU E5-2675 v3 @ 1.80GHz. The algorithm is implemented by Python 3.7 and Numpy package. The C files of benchmark functions from the competition are compiled for Python. Due to the limitation on the length of the article, here we only show the results on the 30-dimensional problems of the CEC'13 benchmark. Each function is tested for 50 repetitions and each run stops after 300000 times of evaluations. Since individual evaluation is considered to be the main time-consuming part in dealing with black-box optimization problems, the evaluation cost of each algorithm can be considered consistent. In actual computation, the time consumption of each algorithm is basically similar.

The Wilcoxon rank-sum tests are conducted to compare the performance of LoTFWA and the proposed multi-scale collaborative fireworks algorithm (MSCFWA). The results are listed in table I. The significant better results are highlighted (with confidence level at 95%).

Compared with LoTFWA, the proposed MSCFWA obtained significantly better results on 10 objective functions, including 3 uni-modal functions, 5 multi-modal functions, and 2 composition functions. Only on one test function, MSCFWA performed significantly worse than LoTFWA. This shows that the proposed algorithm stably and effectively improved the efficiency of fireworks algorithm on all types of objective functions.

We also provide comparison data with some classic swarm intelligent optimization algorithms and evolutionary algorithms, including ABC [19], SPSO2011 [20], DE [21] and CMA-ES [22]. In table II, we can see that MSCFWA has average rank of 2.42, which is the best in all five algorithms.

A box plot of the results of all those algorithms is provided at the end of the article.

V. Conclusion

We proposed a multi-scale collaborative fireworks algorithm to enhance the collaboration between fireworks by assigning different fireworks to different scales of optimization. We give a clear description of the proposed algorithm, with detailed analysis and discussion on how it works in different situations. Experimental results on the CEC'2013 single objective benchmark suite indicate that the proposed algorithm improves LoTFWA steadily on both uni-modal functions and multi-modal functions. It also has significant performance compared with typical evolutionary algorithms and swarm intelligent optimization algorithms.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (GrantNos.: 61673025, 61375119), and National Key R&D Program of China (Grant Nos.: 2018AAA0100300, 2018AAA0102301), and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China (Grant No. 2015CB352302).

References

[1] M. Pelikan, D. E. Goldberg, E. Cantú-Paz *et al.*, "Boa: The bayesian optimization algorithm," in *Proceedings of the genetic and evolutionary computation conference GECCO-99*, vol. 1, 1999, pp. 525–532.

[2] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing*, vol. 4, no. 2, pp. 65–85, 1994.

TABLE I
COMPARISON BETWEEN LoTFWA AND MSCFWA

	LoTFWA					MSCFWA					
F.	Mean	Min	Median	Max	Std.	Mean	Min	Median	Max	Std.	p
1	1.137E-12	2.274E-13	4.547E-13	1.387E-11	8.40E-13	**6.821E-13**	**2.274E-13**	**4.547E-13**	**1.387E-11**	**6.07E-13**	0
2	1.038E+06	2.920E+05	9.164E+05	1.950E+06	4.29E+05	9.756E+05	2.920E+05	9.164E+05	1.950E+06	4.93E+05	0.14
3	2.512E+07	7.258E+05	1.147E+07	1.283E+08	2.48E+07	**1.803E+07**	**7.258E+05**	**1.147E+07**	**1.283E+08**	**2.03E+07**	0.05
4	2.033E+03	7.013E+02	1.791E+03	4.744E+03	8.18E+02	1.888E+03	7.013E+02	1.791E+03	4.744E+03	7.09E+02	0.26
5	4.243E-03	3.133E-03	4.005E-03	6.226E-03	6.18E-04	**4.004E-03**	**3.133E-03**	**4.005E-03**	**6.226E-03**	**6.19E-04**	0.02
6	1.479E+01	5.614E+00	1.523E+01	4.384E+01	5.81E+00	1.515E+01	5.614E+00	1.523E+01	4.384E+01	5.89E+00	0.15
7	5.186E+01	1.695E+01	4.294E+01	6.988E+01	1.22E+01	**4.078E+01**	**1.695E+01**	**4.294E+01**	**6.988E+01**	**1.27E+01**	0
8	**2.085E+01**	**2.069E+01**	**2.088E+01**	**2.099E+01**	**5.93E-02**	2.088E+01	2.069E+01	2.088E+01	2.099E+01	5.14E-02	0.02
9	1.733E+01	1.237E+01	1.792E+01	2.210E+01	1.94E+00	1.695E+01	1.237E+01	1.792E+01	2.210E+01	1.82E+00	0.17
10	3.355E-02	2.389E-07	2.464E-02	8.381E-02	2.47E-02	3.494E-02	2.389E-07	2.464E-02	8.381E-02	2.31E-02	0.26
11	9.287E+01	4.996E+01	8.557E+01	1.254E+02	1.57E+01	**8.228E+01**	**4.996E+01**	**8.557E+01**	**1.254E+02**	**1.62E+01**	0
12	8.787E+01	4.179E+01	8.209E+01	1.244E+02	1.47E+01	**7.840E+01**	**4.179E+01**	**8.209E+01**	**1.244E+02**	**1.52E+01**	0
13	1.585E+02	8.731E+01	1.484E+02	1.887E+02	2.25E+01	**1.461E+02**	**8.731E+01**	**1.484E+02**	**1.887E+02**	**2.79E+01**	0.01
14	2.709E+03	2.076E+03	2.887E+03	3.476E+03	3.19E+02	2.761E+03	2.076E+03	2.887E+03	3.476E+03	3.24E+02	0.19
15	2.716E+03	2.046E+03	2.812E+03	3.433E+03	3.08E+02	2.749E+03	2.046E+03	2.812E+03	3.433E+03	3.22E+02	0.27
16	1.740E-01	6.079E-02	1.889E-01	2.824E-01	5.98E-02	1.870E-01	6.079E-02	1.889E-01	2.824E-01	7.19E-02	0.24
17	1.367E+02	8.973E+01	1.316E+02	1.771E+02	1.62E+01	1.344E+02	8.973E+01	1.316E+02	1.771E+02	2.01E+01	0.12
18	1.398E+02	8.926E+01	1.202E+02	1.805E+02	2.04E+01	1.366E+02	8.926E+01	1.202E+02	1.805E+02	1.78E+01	0.17
19	4.632E+00	2.742E+00	4.807E+00	6.991E+00	1.08E+00	5.028E+00	2.742E+00	4.807E+00	6.991E+00	1.14E+00	0.11
20	1.307E+01	1.105E+01	1.253E+01	1.451E+01	1.02E+00	**1.268E+01**	**1.105E+01**	**1.253E+01**	**1.451E+01**	**1.03E+00**	0.05
21	2.076E+02	1.001E+02	2.000E+02	3.000E+02	5.19E+01	2.184E+02	1.001E+02	2.000E+02	3.000E+02	3.83E+01	0.09
22	3.296E+03	1.948E+03	3.358E+03	4.545E+03	3.93E+02	3.396E+03	1.948E+03	3.358E+03	4.545E+03	4.62E+02	0.21
23	3.340E+03	2.453E+03	3.294E+03	4.231E+03	4.55E+02	3.417E+03	2.453E+03	3.294E+03	4.231E+03	4.07E+02	0.2
24	2.472E+02	2.251E+02	2.416E+02	2.573E+02	8.27E+00	**2.440E+02**	**2.251E+02**	**2.416E+02**	**2.573E+02**	**8.73E+00**	0.04
25	2.762E+02	2.115E+02	2.786E+02	2.900E+02	7.21E+00	2.782E+02	2.115E+02	2.786E+02	2.900E+02	6.35E+00	0.08
26	2.001E+02	2.000E+02	2.000E+02	2.001E+02	2.11E-02	**2.001E+02**	**2.000E+02**	**2.000E+02**	**2.001E+02**	**2.06E-02**	0.02
27	7.829E+02	6.470E+02	7.868E+02	9.494E+02	5.67E+01	7.954E+02	6.470E+02	7.868E+02	9.494E+02	5.23E+01	0.2
28	2.682E+02	1.000E+02	3.000E+02	3.000E+02	7.30E+01	2.800E+02	1.000E+02	3.000E+02	3.000E+02	5.99E+01	0.17

[3] I. Rechenberg, J. Zurada, R. Marks II, and C. Goldberg, "Evolution strategy, in computational intelligence: imitating life," *Computational intelligence imitating life. IEEE Press, Piscataway*, 1994.

[4] J. Kennedy *et al.*, "Encyclopedia of machine learning," *Particle swarm optimization*, pp. 760–766, 2010.

[5] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 53–66, 1997.

[6] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *International conference in swarm intelligence*. Springer, 2010, pp. 355–364.

[7] J. Li and Y. Tan, "Loser-out tournament-based fireworks algorithm for multimodal function optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 5, pp. 679–691, 2017.

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

[9] J. Li, S. Zheng, and Y. Tan, "The effect of information utilization: Introducing a novel guiding spark in the fireworks algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 153–166, 2016.

[10] S. Zheng, A. Janecek, and Y. Tan, "Enhanced fireworks algorithm," in *2013 IEEE congress on evolutionary computation*. IEEE, 2013, pp. 2069–2077.

[11] S. Zheng, C. Yu, J. Li, and Y. Tan, "Exponentially decreased dimension number strategy based dynamic search fireworks algorithm for solving cec2015 competition problems," in *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2015, pp. 1083–1090.

[12] J. Li and Y. Tan, "The bare bones fireworks algorithm: A minimalist global optimizer," *Applied Soft Computing*, p. S1568494617306609, 2017.

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

[14] J. Yu, Y. Tan, and H. Takagi, "Scouting strategy for biasing fireworks algorithm search to promising directions," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2018, pp. 99–100.

[15] J. Yu, H. Takagi, and Y. Tan, "Multi-layer explosion based fireworks algorithm," *J. Swarm Intel. Evol. Comput*, vol. 7, no. 173, p. 2, 2018.

[16] S. Zheng, J. Li, A. Janecek, and Y. Tan, "A cooperative framework for fireworks algorithm," *IEEE/ACM Transactions on Computational Biology & Bioinformatics*, vol. 14, no. 1, pp. 27–41, 2017.

[17] J. Zhang and W. Li, "Last-position elimination-based fireworks algorithm for function optimization," in *International Conference on Swarm Intelligence*. Springer, 2019, pp. 267–275.

[18] X. Li, K. Tang, M. N. Omidvar, Z. Yang, K. Qin, and H. China, "Benchmark functions for the cec 2013 special session and competition on large-scale global optimization," *gene*, vol. 7, no. 33, p. 8, 2013.

[19] S. Biswas, S. Kundu, D. Bose, S. Das, P. N. Suganthan, and B. K. Panigrahi, "Migrating forager population in a multi-population artificial bee colony algorithm with modified perturbation schemes," in *2013 IEEE Symposium on Swarm Intelligence (SIS)*. IEEE, 2013, pp. 248–255.

[20] M. Clerc, "Standard particle swarm optimisation from 2006 to 2011," *Particle Swarm Central*, vol. 253, 2011.

[21] R. Storn and K. Price, "Differential evolution–a simple and efficient

TABLE II
COMPARISON WITH CLASSIC ALGORITHMS

	ABC		SPSO2011		DE		CMA-ES		MSCFWA	
F.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
1	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.89E-03	4.65E-04	**0.00E+00**	**0.00E+00**	6.82E-13	6.07E-13
2	6.20E+06	1.62E+06	3.38E+05	1.67E+05	5.52E+04	2.70E+04	**0.00E+00**	**0.00E+00**	9.76E+05	4.93E+05
3	5.74E+08	3.89E+08	2.88E+08	5.24E+08	2.16E+06	5.19E+06	**1.41E+01**	**9.96E+01**	1.80E+07	2.03E+07
4	8.75E+04	1.17E+04	3.86E+04	6.70E+03	1.32E-01	1.02E-01	**0.00E+00**	**0.00E+00**	1.89E+03	7.09E+02
5	**0.00E+00**	**0.00E+00**	5.42E-04	4.91E-05	2.48E-03	8.16E-04	**0.00E+00**	**0.00E+00**	4.00E-03	6.19E-04
6	1.46E+01	4.39E+00	3.79E+01	2.83E+01	7.82E+00	1.65E+01	**7.82E-02**	**5.58E-01**	1.52E+01	5.89E+00
7	1.25E+02	1.15E+01	8.79E+01	2.11E+01	4.89E+01	2.37E+01	**1.91E+01**	**1.18E+01**	4.08E+01	1.27E+01
8	**2.09E+01**	**4.97E-02**	2.09E+01	5.89E-02	2.09E+01	5.65E-02	2.14E+01	1.35E-01	2.09E+01	5.14E-02
9	3.01E+01	2.02E+00	2.88E+01	4.43E+00	**1.59E+01**	**2.69E+00**	4.81E+01	2.48E+00	1.70E+01	1.82E+00
10	2.27E-01	6.75E-02	3.40E-01	1.48E-01	3.24E-02	1.97E-02	**1.78E-02**	**1.11E-02**	3.49E-02	2.31E-02
11	**0.00E+00**	0.00E+00	1.05E+02	2.74E+01	7.88E+01	2.51E+01	4.00E+02	2.49E+02	8.23E+01	1.62E+01
12	3.19E+02	5.23E+01	1.04E+02	3.54E+01	8.14E+01	3.00E+01	9.42E+02	2.33E+01	**7.84E+01**	**1.52E+01**
13	3.29E+02	3.91E+01	1.94E+02	3.86E+01	1.61E+02	3.50E+01	1.08E+03	6.28E+01	**1.46E+02**	**2.78E+01**
14	**3.58E-01**	**3.91E-01**	3.99E+03	6.19E+02	2.38E+03	1.42E+03	4.94E+03	3.66E+02	2.76E+03	3.24E+02
15	3.88E+03	3.41E+02	3.81E+03	6.94E+02	5.19E+03	5.16E+02	5.02E+03	2.61E+02	**2.75E+03**	**3.22E+02**
16	1.07E+00	1.96E-01	1.31E+00	3.59E-01	1.97E+00	2.59E-01	**5.42E-02**	**2.81E-02**	1.87E-01	7.19E-02
17	**3.04E+01**	**5.15E-03**	1.16E+02	2.02E+01	9.29E+01	1.57E+01	7.44E+02	1.96E+02	1.34E+02	2.01E+01
18	3.04E+02	3.52E+01	**1.21E+02**	**2.46E+01**	2.34E+02	2.56E+01	5.17E+02	3.52E+02	1.37E+02	1.78E+01
19	**2.62E-01**	**5.99E-02**	9.51E+00	4.42E+00	4.51E+00	1.30E+00	3.54E+00	9.12E-01	5.03E+00	1.14E+00
20	1.44E+01	4.60E-01	1.35E+01	1.11E+00	1.43E+01	1.19E+00	1.49E+01	3.96E-01	**1.27E+01**	**1.03E+00**
21	**1.65E+02**	**3.97E+01**	3.09E+02	6.80E+01	3.20E+02	8.55E+01	3.44E+02	7.64E+01	2.18E+02	3.83E+01
22	**2.41E+01**	**2.81E+01**	4.30E+03	7.67E+02	1.72E+03	7.06E+02	7.97E+03	2.19E+02	3.40E+03	4.62E+02
23	4.95E+03	5.13E+02	4.83E+03	8.23E+02	5.28E+03	6.14E+02	6.95E+03	3.27E+02	**3.42E+03**	**4.07E+02**
24	2.90E+02	4.42E+00	2.67E+02	1.25E+01	2.47E+02	1.54E+01	6.62E+02	7.20E+01	**2.44E+02**	**8.73E+00**
25	3.06E+02	6.49E+00	2.99E+02	1.05E+01	2.80E+02	1.57E+01	4.41E+02	4.00E+00	**2.78E+02**	**6.35E+00**
26	2.01E+02	1.93E-01	2.86E+02	8.24E+01	2.52E+02	6.83E+01	3.29E+02	8.24E+00	**2.00E+02**	**2.06E-02**
27	**4.16E+02**	**1.07E+02**	1.00E+03	1.12E+02	7.64E+02	1.00E+02	5.39E+02	7.64E+01	7.95E+02	5.23E+01
28	**2.58E+02**	**7.78E+01**	4.01E+02	4.76E+02	4.02E+02	3.90E+02	4.78E+03	3.79E+02	2.80E+02	5.99E+01
Average Rank	2.88		3.36		2.80		3.54		2.42	

heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[22] N. Hansen, "The cma evolution strategy: a comparing review," in *Towards a new evolutionary computation*. Springer, 2006, pp. 75–102.

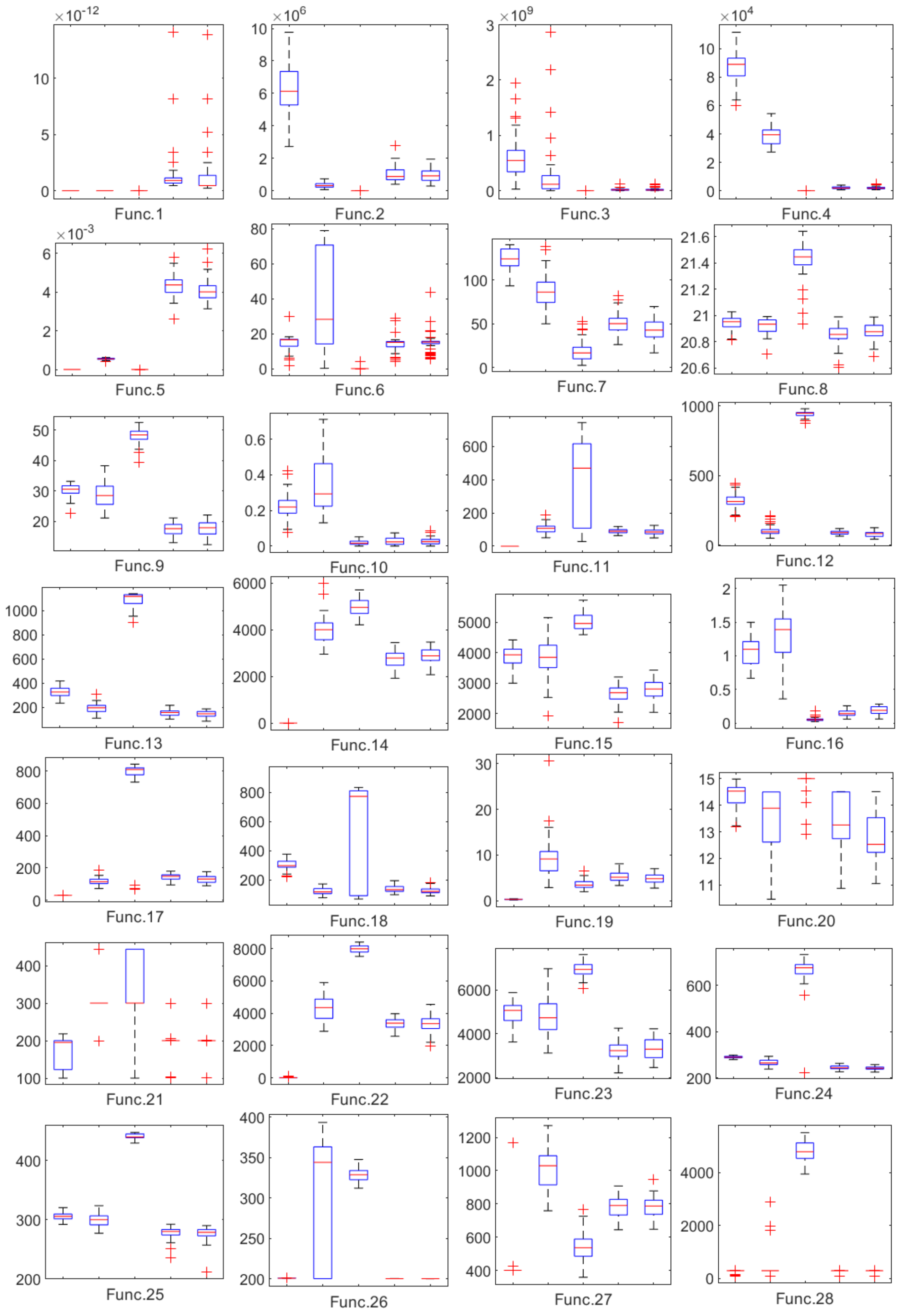

Fig. 2. Box-plots of the algorithms' results tested on CEC'13 benchmarks. (From left to right: ABC, SPSO2011, CMA-ES, LoTFWA and MSCFWA.)