Enhancing Fireworks Algorithm in Local Adaptation and Global Collaboration

Yifeng Li and Ying Tan(✉)

Key laboratory of Machine Perception (MOE), Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
{liyifeng,ytan}@pku.edu.cn

Abstract. In order to improve the performance of fireworks algorithm, this paper carries out a comprehensive enhancement for its framework. Locally, the basic explosion operator is replaced by an efficient adaptation method in CMA-ES. Globally, the explosion range of all fireworks is effectively collaborated by search space partition. On the one hand, the proposed algorithm can quickly adapt to local landscape and improve the local exploitation efficiency significantly. On the other hand, it can collaborate the search ranges of multiple fireworks to form a seamless and non-overlapping partition of the search space, thereby ensuring the global search ability. Since the proposed framework evaluates one batch of a fixed large number of solutions in each iteration, it also achieves better computational efficiency in modern parallel hardware. The proposed algorithm is tested on the CEC 2020 benchmark functions with three different dimensions. The experimental results prove that those strategies improve fireworks algorithm significantly.

Keywords: Fireworks algorithm · Swarm intelligence · Optimization · Collaboration

1 Introduction

Modern optimization problem has changed drastically in recent years. On the one hand, more and more difficult objective functions have emerged in practical applications, which are usually multi-modal and high-dimensional. On the other hand, modern computing technologies, especially parallel technology, put forward new directions for the development of optimization algorithms.

The fireworks algorithm (FWA [12]) is a family of algorithms inspired by the phenomenon of firework explosion, which is very promising for solving such kind of problems effectively. During the optimization, each firework search a

This work is supported by the National Natural Science Foundation of China (Grant No. 62076010), and partially supported by Science and Technology Innovation 2030 - "New Generation Artificial Intelligence" Major Project (Grant Nos.: 2018AAA0102301 and 2018AAA0100302).

Y. Tan and Y. Shi (Eds.): ICSI 2021, LNCS 12689, pp. 451–465, 2021.
https://doi.org/10.1007/978-3-030-78743-1_41

local area by explosion. All fireworks collaborate their strategies for overall efficiency. Many variants of FWA have achieved competitive performance in standard benchmarks, such as EFWA [19], AFWA [7], dynFWA [18], GFWA [8] and LoTFWA [6]. FWA has also solved real-world problems like image processing [15], engineering [4] and resource scheduling [10].

However, in most variants of FWA, the efficiency of explosion operator based on uniform sampling is very limited. Their collaborative strategies also have little effect on the independent search of fireworks. At the same time, the parallel efficiency is obviously weakened by operators like mutation, because they requires an additional evaluation of a small batch of mutation sparks.

In this article, new strategies are proposed for both local explosion and global collaboration. For each firework, the basic explosion method is replaced by an adaptation strategy for a Gaussian distribution, which is able to fit the local landscape and target the extreme very fast. For global optimization, a collaboration method based on search space partition is proposed to arrange the search areas of fireworks, thus greatly reduce the probability of overlapping or omission. The restart strategy is also improved. Since all the evaluations in each iteration are done in one large batch, the proposed algorithm is also well adapted to large-scale parallel computing hardware.

The paper is organized as follows. It starts by introducing backgrounds in Sect. 2. In Sect. 3, the proposed strategies are described in detail. In Sect. 4, the proposed algorithm is evaluated and compared on benchmark problems. Finally, the proposed algorithm is discussed and analyzed in Sect. 4 and concluded in Sect. 5.

2 Backgrounds

2.1 Problem Definition

In this paper, we consider the general bound-constrained optimization problem which targets to find the optimal solution x^*:

$$x^* = \arg\min_{x \in S} f(\mathbf{x}) \tag{1}$$

where $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is an unknown objective function (also called fitness function). $S = \left\{x \in \mathbb{R}^d : lb_i < x_i < ub_i\right\}$ is the feasible space of f.

Optimization algorithms (or optimizers) are applied to approximate the optimal x^* or its value $f(x^*)$ by iterating the process of ask and tell. Since we consider complex objective functions with high time cost, the termination condition is a specific number of evaluations. At the same time, in order to maximize the computational efficiency, we hope the algorithm always provide a fixed number of solutions in each batch. The actual size of batch should be determined by the parallel computing devices.

2.2 Fireworks Algorithms

Fireworks algorithm is a novel optimization framework that adopts multiple collaborative isomorphic subgroups. Among all the novel implementations of fireworks algorithm, LoTFWA [12] has achieved the most significant global optimization performance with extremely simple mechanisms. In LoTFWA, each firework optimize its local area by an uniform explosion within dynamic amplitudes. A guided mutation spark is generated for each firework to accelerate its local exploitation. Then, some unpromising fireworks are detected and restarted to avoid waste of resources.

There are two major weaknesses in LoTFWA which are improved in the proposed algorithm. First, the local search efficiency of the explosion operator and mutation operator is limited by a basic uniform trust region scheme. Second, the collaboration method is too weak because the restart mechanism is rarely triggered and it can only save limited resources rather than guide fireworks to cooperate.

2.3 Related Works

A great number of Evolutionary Algorithms (EAs) and Swarm Intelligence Optimization Algorithms (SIOAs) have been proposed for similar optimization problems, but their ideas are fundamentally different from the proposed algorithm.

The idea of adopting multiple sub-populations for optimization is implemented in a large number of recent research of EAs and SIOAs. In most cases, different sub-populations evolve under different strategies in order to combine their advantages and obtain efficient hybrid algorithm. For example, EBOwithCMAR [5] uses three sub-populations which apply Effective Butterfly Optimizer or Covariance Matrix Adapted Retreat method respectively, and achieved outstanding performance in the competition of CEC2017 [16]. Some optimizers use the same algorithm with different parameters in sub-populations. For example, BIPOP-CMA-ES [1] adopts multi-restart populations with different sizes. In IMODE [11], the winner of CEC2020 competition [17], different sub-populations with dynamic sizes evolve under different DE parameters. There are also many algorithms like SHADE [13] that utilizes archive strategy to collect an elite population in order to enhance the optimization efficiency.

The essential difference between those methods and the proposed algorithm is that we analytically defined the ranges of sub-populations according to the principle of search space partition. And the sub-populations are diversified and cooperated in different local areas instead of different strategies.

3 Proposed Strategies

The proposed algorithm is improved in both local exploitation and global collaboration. Locally, CMA-ES [3] is introduced to accelerate the optimization of each firework. Globally, the explosion distributions are collaborated to form a seamless and non-overlapping partition of the search space. The framework of the proposed algorithm is described in Algorithm 1.

Algorithm 1. Framework of Proposed Algorithm

```
Initialize each firework X_i
while termination conditions are not satisfied do
  // 1. Adaptation
  for each firework X_i do
    generate λ_i sparks by explosion
  end for
  Gather and estimate all sparks
  for each firework X_i do
    update states of X_i
  end for
  // 2. Restart
  Examine and restart fireworks
  // 3. Collaboration
  for each pair of fireworks X_i and X_j do
    Determine their collaborative search boundaries
  end for
  for each firework X_i do
    Fit search boundary towards the collaboration result
  end for
end while
```

3.1 Adaptation

In order to enhance the local optimization efficiency, the uniform explosion is replaced by a self-adaptive Gaussian distribution. With strategies introduced from CMA-ES [2], it is able to estimate the local fitness landscape and generate more effective sparks.

In the g-th generation, the k-th explosion spark $\mathbf{x}_k^{(g+1)}$ is generated from a Gaussian distribution:

$$\mathbf{x}_k^{(g+1)} \sim \mathbf{m}^{(g)} + \sigma^{(g)} \times \mathcal{N}(\mathbf{0}, C^{(g)}) \tag{2}$$

where $\mathbf{m}$ and C is the mean and co-variance matrix. $\sigma^{(g)}$ is the overall step size. In the proposed algorithm, each firework generate the same number of λ sparks.

After evaluation of all sparks $\mathbf{x}^{(g+1)}$, the explosion distribution is adapted according to the strategies in CMA-ES. The complete adaptation algorithm is provided in supplementary A. And a detailed explanation and parameter setting of CMA-ES can be found in [2].

3.2 Restart

Since the adaptation accelerates local optimization significantly, several conditions are proposed to ensure timely restart of fireworks that are not promising to improve the global optimal.

Three restart conditions are determined by the search status of the firework individual:

1. **Low Value Variance:** var $\left[f(x_{1:\lambda}^{(g+1)})\right] \leq \epsilon_v$
2. **Low Position Variance:** $\sigma^{(g+1)} \times \left\|\mathbf{C}^{(g+1)}\right\| \leq \epsilon_p$
3. **Not improving:** Not improved for $I_{max\_not\_improve}$ iterations.

One more restart condition is determined by the relationship between fireworks:

1. **Covered by Better:** More than 85% of the firework's sparks are covered by a better firework's explosion range.

3.3 Collaboration

The explosion boundary of a firework X with parameters $(\mathbf{m}, C, \sigma)$ is defined as:

$$\left\{\mathbf{x} \middle| \left\|\mathbf{C}^{-\frac{1}{2}}(\frac{\mathbf{x}-\mathbf{m}}{\sigma})\right\| = E\left\|\mathcal{N}(\mathbf{0},\mathbf{I})\right\|\right\} \tag{3}$$

Obviously, it is a elliptical shell and covers the majority of X's explosion sparks. The proposed strategy is designed according to two core ideas:

1. The explosion scopes tends to form a segmentation within the global optimization area.
2. The better fireworks tend to search independently, and the worse fireworks tend to search collaboratively.

The first idea is helpful to avoid overlapping or omission of search scopes, so the overall efficiency of fireworks can be improved in collaboration. The second idea ensures the local optimization of leading fireworks will not be severely affected by collaboration. Based on these ideas, the proposed algorithm conducts collaboration by the following steps:

a) Compare Fireworks. A fuzzy comparison between each pair of fireworks is introduced to estimate their relative optimization progress, which is described in Algorithm 2.

b) Compute Dividing Points. The dividing point for each pair of fireworks is obtained, which specifies where the search range of both fireworks are divided. Figure 1 gives examples of the collaboration method in 4 possible situations.

The following steps are conducted to calculate the dividing point:

1. Calculate the distance d_{ij} between X_i and X_j
2. Calculate the radius $r_{ij} = |X_i A_i|$ and $r_{ji} = |X_j A_j|$ on line $X_i X_j$
3. Determine the situation (See Fig. 1) according to r_{ij}, r_{ji} and d_{ij}
4. Calculate the position of A_i and A_j
5. If the optimization of X_i is ahead of X_j, A_i is the dividing point. If X_j is ahead of X_i, A_j is the dividing point. Otherwise, the midpoint B of $A_i A_j$ is the dividing point.

Algorithm 2. Fuzzy Comparison of Fireworks

Require: Fireworks $\mathbf{X}_i$ and $\mathbf{X}_j$ with sparks $\mathbf{x}_{i,1:\lambda}^{(g+1)}$ and $\mathbf{x}_{j,1:\lambda}^{(g+1)}$ (if not restarted)
  if Both $\mathbf{X}_i$ and $\mathbf{X}_j$ are just restarted **then**
    return $\mathbf{X}_i$ and $\mathbf{X}_j$ are similar
  end if
  if $\mathbf{X}_i$ is restarted **then**
    return $\mathbf{X}_j$ is ahead of $\mathbf{X}_i$
  end if
  ... #vice versa
  if $\min \mathbf{x}_{i,1:\lambda}^{(g+1)} > \max \mathbf{x}_{j,1:\lambda}^{(g+1)}$ **then**
    return $\mathbf{X}_j$ is ahead of $\mathbf{X}_i$
  end if
  ... #vice versa
  return X_i and X_j are similar

Before fitting the boundary to obtained dividing points of X_i, two additional operations are required. First, only the closest N (the dimension of objective function) dividing points are kept, so the collaboration is conducted locally. Second, the distance of X_i to its dividing points with X_j is clipped within $[0.5r_{ij}, 2r_{ij}]$, so there won't be too drastic changes after collaboration.

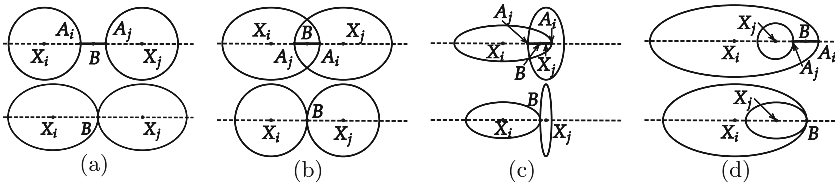


Fig. 1. Four cases of collaboration between two fireworks. A_i and A_j are the closer intersections of line X_iX_j with their boundaries. The actual dividing point could be any point on A_iA_j. The second row shows the collaboration results when taking the midpoint B of A_iA_j as dividing point.

c) Fit Dividing Points. The boundary of firework $X(\mathbf{m}, C, \sigma)$ is adapted to fit its dividing points. For each dividing point P_k, a new covariance matrix C_k is calculated. On the direction of XP_k, P_k lies right on the boundary. On the conjugate directions, the radius of boundary is not changed. The mathematical calculation for fitting a single split point is given in the Appendix B. The mean of all adapted covariance matrix $\frac{1}{K}\sum_{k=1}^{K} C_k$ is taken as the overall collaborated results of X.

Algorithm 3 outlines the framework of the proposed collaboration strategy:

Algorithm 3. Framework of Fireworks Collaboration

Require: n fireworks X_i and parameters $(\mathbf{m}_i, \mathbf{C}_i, \sigma_i)$ in N dimensional feasible space
Ensure: Collaborated parameters of fireworks
  for each pair of fireworks X_i and X_j **do**
    Compare the progress of X_i and X_j
    Calculate $d_{ij} = |X_i X_j|$, expected sample distance r_{ij} and r_{ji} on $X_i X_j$
    Calculate the dividing point P_{ij} $(= P_{ji})$
  end for
  for each firework X_i **do**
    Gather $K = \min(N, n-1)$ closest dividing points $P_{i,j_{1:K}}$
    Clip the length of $X_i P_{ij_k}$ within $[0.5 r_{ij_k}, 2 r_{ij_k}]$
    for $k \leftarrow 1 : K$ **do**
      Fit P_{ij_k} on the boundary of X_i and obtain $\mathbf{C}_{ij_k}$
    end for
    $\mathbf{C}_i \leftarrow \frac{1}{K}\sum_{k=1}^{K} \mathbf{C}_{ij_k}$
  end for

3.4 Experiments

The performance of proposed algorithm is tested on objective functions from the CEC 2020 benchmark test suit [17]. According to the settings of the bound-constrained single-objective optimization competition, each function is tested for 30 repetitions with 10, 15, 20 dimensions. The termination condition is a maximum of 1,000,000, 3,000,000 or 10,000,000 evaluations for 10, 15 or 20 dimensions, respectively.

For the generalization ability of the proposed strategies, there is few additional parameters introduced. In the restart conditions, ϵ_v and ϵ_p are both $1E{-}6$, and the maximum number of unimproved iteration $I_{max\_unimprove}$ is 20. Its basic settings are the same as LoTFWA, which includes 5 fireworks and 300 sparks in each iteration. The parameters of local adaption is also set to be the same as CMA-ES. As we can see, there is no parameter selection according to the target problems.

In order to prove the effectiveness of our proposed strategies, the proposed algorithm is compared with two baselines. LoTFWA is the most efficient one of the main variants of the firework algorithm. CMAFWA is a compromise between LoTFWA and the proposed algorithm, whose fireworks use the local search strategy of CMA-ES but are collaborated by the loser-out tournament strategies from LoTFWA.

The statistical test results of the three algorithms are shown in the Table 1, Table 2 and Table 4 for 10, 15 and 20 respectively. Their fitness curves are shown in supplementary C.

As can be seen from the experimental results, the proposed algorithm outperforms LoTFWA significantly on all objective functions. CMAFWA improves on uni-modal (1), basic functions (2, 3, 4) and hybrid functions (5, 6, 7) compared with LoTFWA, but becomes worse in complex functions (8, 9, 10) due to ineffective collaboration. The proposed algorithm is overall better than CMAFWA,

Table 1. Wilcoxon signed-rank test on 10D problems. ($\alpha = 0.05$. Statistical test is conducted against the proposed method. '+' means the proposed method is significantly better. '−' means the proposed method is significantly worse. '=' means the two algorithm performs similarly.)

F.	LoTFWA			CMAFWA			Proposed	
	Mean	Std		Mean	Std		Mean	Std
1	9.123E+05	2.579E+05	+	0.000E+00	0.000E+00	=	0.000E+00	0.000E−00
2	3.947E+02	2.131E+02	+	2.558E+01	3.445E+01	−	1.427E+02	1.606E+02
3	3.629E+01	7.296E+00	+	1.147E+01	6.350E−01	−	1.295E+01	3.144E+00
4	3.456E+00	6.189E−01	+	6.900E−01	1.427E−01	+	0.000E+00	0.000E+00
5	8.592E+03	7.521E+03	+	2.227E+02	1.166E+02	+	3.110E+01	1.643E+01
6	1.367E+02	4.205E+01	+	8.365E−01	3.917E−01	=	7.231E−01	4.031E−01
7	4.532E+03	4.710E+03	+	7.656E+00	2.204E+01	=	5.257E+00	6.804E+00
8	9.747E+01	2.689E+01	+	2.240E+02	3.200E+01	+	9.507E+01	1.845E+01
9	3.022E+02	8.871E+01	+	3.305E+02	9.224E−01	+	1.616E+02	8.019E+01
10	4.140E+02	2.155E+01	+	4.313E+02	2.004E+01	+	3.978E+02	1.015E−01
Rank	2.60			2.00			1.20	

Table 2. Wilcoxon signed-rank test on 15D problems. ($\alpha = 0.05$)

F.	LoTFWA			CMAFWA			Proposed	
	Mean	Std		Mean	Std		Mean	Std
1	1.444e+06	2.522e+05	+	0.000e+00	0.000e+00	=	0.000e+00	0.000e−00
2	1.095e+03	3.681e+02	+	8.932e+01	5.050e+01	−	9.387e+02	4.298e+02
3	5.343e+01	5.794e+00	+	1.652e+01	2.682e−01	−	3.617e+01	1.679e+01
4	7.060e+00	1.108e+00	+	7.677e−01	1.449e−01	+	6.257e−01	3.998e−01
5	1.060e+05	8.527e+04	+	2.967e+02	1.062e+02	+	1.217e+02	3.851e+01
6	3.769e+02	9.960e+01	+	7.248e−01	1.722e−01	−	4.624e+01	2.497e+01
7	5.507e+04	3.426e+04	+	1.982e+00	9.683e−01	−	3.834e+01	4.806e+01
8	1.103e+02	5.193e−01	+	2.298e+02	1.095e+01	+	1.000e+02	1.666e−07
9	3.099e+02	1.458e+02	+	3.905e+02	2.175e−01	+	1.583e+02	7.640e+01
10	4.385e+02	7.625e+01	+	5.228e+02	8.684e+01	+	4.000e+02	4.451e−07
Rank	2.60			1.80			1.40	

especially on composition functions (8, 9 and 10) and problems that have relatively simple local landscape (4, 5). But it failed to improve on problems with a large number of local areas with insignificant overall trend (2, 3) or related hybrid problems when the dimension becomes large. The most possible reason could be that the limited number of fireworks are not able to form an effective partition of the huge search space when dimension grows. On the other hand, linking the ranges of limited local search sometimes might leads to inefficient local exploitation (Table 3).

Table 3. Wilcoxon signed-rank test on 20D problems. ($\alpha = 0.05$)

F.	LoTFWA			CMAFWA			Proposed	
	Mean	Std		Mean	Std		Mean	Std
1	1.798e+06	4.388e+05	+	0.000e+00	0.000e+00	=	0.000e+00	0.000e-00
2	1.454e+03	3.986e+02	+	5.772e+01	1.833e+01	−	4.299e+02	1.681e+02
3	6.751e+01	1.126e+01	+	2.316e+01	5.063e−01	−	6.181e+01	2.962e+01
4	1.017e+01	1.274e+00	+	1.985e+00	1.113e−01	+	1.867e+00	6.521e-01
5	2.564e+05	1.808e+05	+	7.899e+02	1.829e+02	+	1.891e+02	4.939e+01
6	5.239e+02	1.968e+02	+	1.947e+00	2.190e+01	−	1.594e+02	5.865e+01
7	9.663e+04	6.730e+04	+	5.185e+00	3.104e+00	−	1.005e+02	4.884e+01
8	1.001e+02	1.945e+01	=	2.694e+02	1.855e+01	+	1.000e+02	2.272e-07
9	4.446e+02	1.994e+01	+	4.005e+02	1.476e+00	+	2.112e+02	9.651e+01
10	4.255e+02	1.695e+01	+	4.063e+02	4.547e−13	+	4.024e+02	5.840e+00
Rank	2.70			1.50			1.40	

The proposed algorithm is also compared with LoTFWA, IPOP-CMA-ES [9] and LSHADE [14], which have been the most famous EA or SIOA in recent years, in the Table 1 on CEC 2020 benchmark test suits with 20 dimensions. The proposed algorithm outperforms LoTFWA and IPOP-CMA-ES in all problems. It achieved better performance on composition functions but is not as good as LSHADE on basic functions and hybrid functions of CEC 2020 test suits.

Table 4. Comparison with classic algorithms on 20D problems of CEC 2020

F.	LoTFWA		IPOP-CMA-ES		LSHADE		Proposed	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
1	1.80e+06	4.39e+05	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e−00
2	1.45e+03	3.99e+02	2.16e+03	2.41e+01	2.39e+00	1.38e+00	4.30e+02	1.68e+02
3	6.75e+01	1.13e+01	5.43e+01	7.97e+00	2.08e+01	5.23e−01	6.18e+01	2.96e+01
4	1.02e+01	1.27e+00	2.32e+00	2.78e−01	4.70e−01	4.66e−02	1.87e+00	6.52e−01
5	2.56e+05	1.81e+05	1.23e+03	2.83e+02	5.51e+01	6.01e+01	1.89e+02	4.94e+01
6	5.24e+02	1.97e+02	4.91e+02	2.19e+00	3.48e−01	8.05e−02	1.59e+02	5.87e+01
7	9.66e+04	6.73e+04	7.18e+02	2.10e+02	8.13e−01	1.33e−01	1.00e+02	4.88e+01
8	1.00e+02	1.95e+01	2.48e+03	1.85e+02	1.00e+02	1.00e−03	1.00e+02	2.27e−07
9	4.45e+02	1.99e+01	4.32e+02	1.48e+00	4.03e+02	1.06e+00	2.11e+02	9.65e+01
10	4.26e+02	1.70e+01	4.30e+02	4.55e−01	4.14e+02	1.47e−02	4.02e+02	5.84e+00
Rank	3.7		3.0		1.2		1.7	

There are also some highly efficient variants of classic algorithms in the CEC 2020 bound-constrained single-objective competition. The proposed algorithm has certain advantages in the competition, especially on composition functions.

But it is still insufficient to compete with the best algorithms. We do not compare with these algorithms for several reasons: a) Many of them improve their performance based on too delicate strategies and tricks, like applying additional optimizer in certain stage of optimization. b) Almost all of them applied dynamic population size, which violates our assumption on parallel computing devices. c) Most of them are designed and fine-tuned for the specific problems of CEC 2020 instead of general problems.

4 Discussions

The collaboration strategy plays an important role for the proposed algorithm in two ways.

Globally, the collaboration strategy tends to link the explosion ranges of separated fireworks. Therefore, fireworks naturally fill their vacancy even when searching in the same direction. It also help the poor fireworks to expand their search ranges and get closer to the better fireworks.

Locally, the collaboration strategy helps to avoid overlapping explosion ranges of different fireworks. Even when multiple fireworks fall into a same convex area, they tends to form a segmentation of the local area and search together, instead of overlapping and conduct similar searches independently.

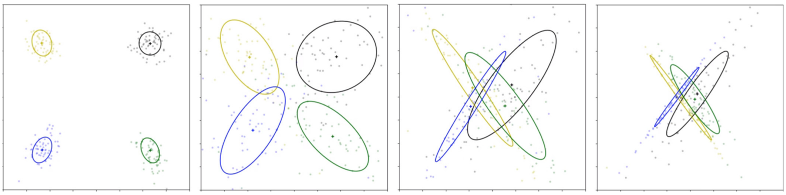


Fig. 2. A simple example of the collaboration of 4 fireworks.

Figure 2 gives a simple example of the collaboration of 4 fireworks in a single-modal problem. In the early stage of optimization, the explosion ranges expand and connect each other quickly. While in the later stage, four explosion ranges collaboratively search around the optimal, just like a single Gaussian distribution with larger population size.

5 Conclusion

This paper proposed a novel fireworks algorithm which is enhanced in both local adaptation and global collaboration. The uniform explosion method is replaced by a self-adaptive Gaussian distribution with strategy introduced from CMA-ES. The fireworks are effectively collaborated by the idea of search space partition.

The experimental results show that the proposed algorithm has better performance compared to former FWA variants.

The proposed algorithm is developed based on a theoretical thinking of firework algorithm. There are still plenty of details could be improved. For example, the balancing between local adaptation and global collaboration is extremely valuable for an in-depth study.

Appendix

A. Covariance Matrix Adaption

The local adaption of firework is done independently by the strategies from CMA-ES.

For firework X with parameters $(\mathbf{m}^{(g)}, C^{(g)}, \sigma^{(g)})$ and explosion sparks $\mathbf{x}_{1:\lambda}^{(g+1)}$. First, a recombination weight $\mathbf{w}$ is applied to μ best sparks for updating the mean:

$$\mathbf{m}^{(g+1)} = \mathbf{m}^{(g)} + c_m \sum_{i=1}^{\mu} w_i (\mathbf{x}_i^{(g+1)} - \mathbf{m}^{(g)}) \tag{4}$$

where c_m is the learning rate. $w_i \geq 0$ and $\sum w_i = 1$.

For the adaption of covariance matrix, a combined rank-μ update and rank-one update is applied in CMA-ES:

$$\begin{aligned} C^{(g+1)} = &(1 - c_1 - c_\mu \sum w_j) C^{(g)} \\ &+ c_\mu \sum_{i=1}^{\lambda} w_i \mathbf{y}_i^{(g+1)} (\mathbf{y}_i^{(g+1)})^T \\ &+ c_1 \mathbf{p}_c^{(g+1)} (\mathbf{p}_c^{(g+1)})^T \end{aligned} \tag{5}$$

where

- c_1 and c_μ are learning rates.
- $\mathbf{y}_i^{(g+1)} = (\mathbf{x}_i^{(g+1)} - \mathbf{m}^{(g)}) / \sigma^{(g)}$.
- $\mathbf{p}_c^{(g+1)}$ is the evolution path, which is intiallized as $\mathbf{0}$ and updated by Eq. 6

$$\mathbf{p}_c^{(g+1)} = (1 - c_c)\mathbf{p}_c^{(g)} + \sqrt{c_c(2 - c_c)\mu_{\text{eff}}} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}} \tag{6}$$

For the adaptation of scale σ, a conjugate evolution path $\mathbf{p}_\sigma^{(g)}$ is initialized as $\mathbf{0}$ and updated in each iteration:

$$\mathbf{p}_\sigma^{(g+1)} = (1 - c_\sigma)\mathbf{p}_\sigma^{(g)} + \sqrt{c_\sigma(2 - c_\sigma)\mu_{\text{eff}}} \mathbf{C}^{(g)^{-\frac{1}{2}}} \frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\sigma^{(g)}} \tag{7}$$

In general CMA-ES, the rank-1 update plays a significant role in rapidly moving towards a better local position at the initial stage of search. However, we'd like the firework to focus more around its initial position. According to experiments, the proposed algorithm performs much better without rank-1 update, that is:

$$C^{(g+1)} = (1 - c_\mu \sum w_j)C^{(g)} + c_\mu \sum_{i=1}^{\lambda} w_i \mathbf{y}_i^{(g+1)} (\mathbf{y}_i^{(g+1)})^T \tag{8}$$

Finally, the scale $\sigma^{(g+1)}$ is updated by comparing the length of $\|\mathbf{p}_\sigma^{(g+1)}\|$ with its expected length $E\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|$:

$$\ln \sigma^{(g+1)} = \ln \sigma^{(g)} + \frac{c_\sigma}{d_\sigma} \left(\frac{\left\|\mathbf{p}_\sigma^{(g+1)}\right\|}{E\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} \right) \tag{9}$$

B. Fitting Single Dividing Point

Here, we fit a diving point $\mathbf{x}$ on the boundary (defined in Eq. 3) of a Gaussian distribution $\mathcal{N}(\mathbf{m}, C)$ and overall sample scale σ.

First, a linear transformation $f(\mathbf{x}) = C^{-\frac{1}{2}}(\mathbf{x} - \mathbf{m})/\sigma$ is conducted to the entire space, so the normal distribution is transformed to $\mathcal{N}(\mathbf{0}, I)$. Assume diving point $\mathbf{x}$ is projected to $\mathbf{z}$.

In the transformed space, the boundary should only be changed on the direction of $\mathbf{z}$. So we can assume the adapted covariance matrix C_x in the transformed space is $aI + b\mathbf{z}\mathbf{z}^T$.

Extend $\mathbf{z}$ into a set of linear bases $B = \{\mathbf{z}, \mathbf{z}_1, ..., \mathbf{z}_{N-1}\}$. Assume all $\mathbf{z}_i$ is on the boundary of $\mathcal{N}(\mathbf{0}, I)$. Since the sample distance on the conjugate directions of $\mathbf{z}$ should not be changed for C and C_x, they are also on the boundary of $\mathcal{N}(\mathbf{0}, aI + b\mathbf{z}\mathbf{z}^T)$. So we have:

$$\left\|(aI + b\mathbf{z}\mathbf{z}^T)^{-\frac{1}{2}}\mathbf{y}\right\| = E\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|, \forall \mathbf{y} \in B \tag{10}$$

Let $d = E\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|$, it is equivalent to:

$$\mathbf{y}^T(aI + b\mathbf{z}\mathbf{z}^T)^{-1}\mathbf{y} = d^2, \forall \mathbf{y} \in B \tag{11}$$

According to the Woodbury Matrix Identity (Eq. 13), the equations can be solved to obtain a and b.

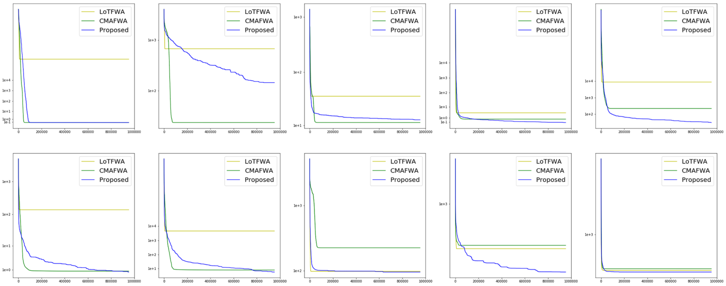
$$(A + UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1} + VA^{-1}U\right)^{-1} VA^{-1} \tag{12}$$

Finally, the adapted matrix in original space is :

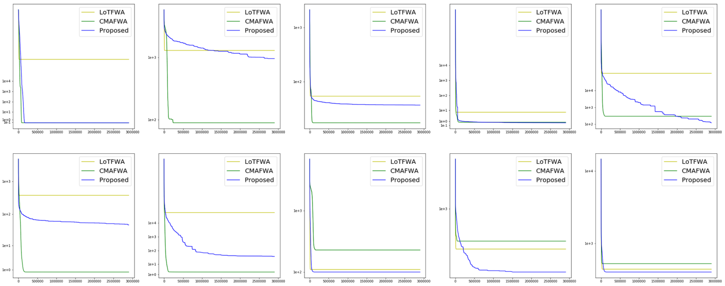
$$\begin{aligned} C_x &= C^{\frac{1}{2}}\left(aI + b\mathbf{z}\mathbf{z}^T\right)C^{\frac{1}{2}} \\ &= aC + b(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T/\sigma^2 \end{aligned} \tag{13}$$

C. Fitness Curves

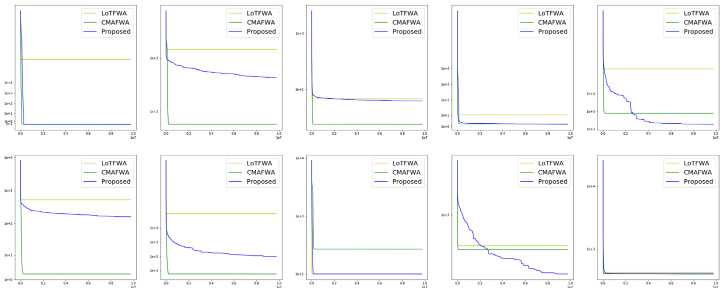
Here we provide the fitness curves of LoTFWA, CMAFWA and the proposed algorithm on the CEC 2020 benchmark problems (Fig. 3).



(a) 10D



(b) 15D



(c) 20D

Fig. 3. Fitness curves on CEC2020 problems.

References

1. Hansen, N.: Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, pp. 2389–2396 (2009)

2. Hansen, N.: The CMA evolution strategy: a tutorial. arXiv preprint arXiv:1604.00772 (2016)
3. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol. Comput. **11**(1), 1–18 (2003)
4. Imran, A.M., Kowsalya, M.: A new power system reconfiguration scheme for power loss minimization and voltage profile enhancement using fireworks algorithm. Int. J. Electr. Power Energy Syst. **62**, 312–322 (2014)
5. Kumar, A., Misra, R.K., Singh, D.: Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase. In: 2017 IEEE Congress on Evolutionary Computation (CEC), pp. 1835–1842 (2017). https://doi.org/10.1109/CEC.2017.7969524
6. Li, J., Tan, Y.: Loser-out tournament-based fireworks algorithm for multimodal function optimization. IEEE Trans. Evol. Comput. **22**(5), 679–691 (2017)
7. Li, J., Zheng, S., Tan, Y.: Adaptive fireworks algorithm. In: 2014 IEEE Congress on evolutionary computation (CEC), pp. 3214–3221. IEEE (2014)
8. Li, J., Zheng, S., Tan, Y.: The effect of information utilization: introducing a novel guiding spark in the fireworks algorithm. IEEE Trans. Evol. Comput. **21**(1), 153–166 (2016)
9. Loshchilov, I.: CMA-ES with restarts for solving CEC 2013 benchmark problems. In: 2013 IEEE Congress on Evolutionary Computation, pp. 369–376. IEEE (2013)
10. Reddy, K.S., Panwar, L.K., Kumar, R., Panigrahi, B.K.: Distributed resource scheduling in smart grid with electric vehicle deployment using fireworks algorithm. J. Mod. Power Syst. Clean Energy **4**(2), 188–199 (2016). https://doi.org/10.1007/s40565-016-0195-6
11. Sallam, K.M., Elsayed, S.M., Chakrabortty, R.K., Ryan, M.J.: Improved multi-operator differential evolution algorithm for solving unconstrained problems. In: 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2020)
12. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) ICSI 2010. LNCS, vol. 6145, pp. 355–364. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13495-1_44
13. Tanabe, R., Fukunaga, A.: Success-history based parameter adaptation for differential evolution. In: 2013 IEEE Congress on Evolutionary Computation, pp. 71–78. IEEE (2013)
14. Tanabe, R., Fukunaga, A.S.: Improving the search performance of shade using linear population size reduction. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 1658–1665. IEEE (2014)
15. Tuba, M., Bacanin, N., Alihodzic, A.: Multilevel image thresholding by fireworks algorithm. In: 2015 25th International Conference Radioelektronika (RADIOELEKTRONIKA), pp. 326–330. IEEE (2015)
16. Wu, G., Mallipeddi, R., Suganthan, P.N.: Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical report (2017)
17. Yue, C., et al.: Problem definitions and evaluation criteria for the CEC 2020 special session and competition on single objective bound constrained numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China And Technical Report, Nanyang Technological University (2019)

18. Zheng, S., Janecek, A., Li, J., Tan, Y.: Dynamic search in fireworks algorithm. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 3222–3229. IEEE (2014)
19. Zheng, S., Janecek, A., Tan, Y.: Enhanced fireworks algorithm. In: 2013 IEEE Congress on Evolutionary Computation, pp. 2069–2077. IEEE (2013)