

# Exponentially Decreased Dimension Number Strategy Based Dynamic Search Fireworks Algorithm for Solving CEC2015 Competition Problems

Shaoqiu Zheng, Chao Yu, Junzhi Li and Ying Tan

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

**Abstract**—Fireworks algorithm (FWA) is one swarm intelligence algorithm proposed in 2010, which takes the inspiration from the firework explosion process. Compared with other meta-heuristic algorithms, FWA presents a cooperative explosive search manner. In the explosive search manner, the explosion amplitudes, explosion sparks' numbers and explosion dimension selection methods play the key roles for its successful implementation. In this paper, the performance analyses of the different explosion dimension number strategies in FWA and its variants are presented at first, then the exponentially decreased explosion dimension number strategy is introduced for the most recent dynamic search fireworks algorithm (dynFWA), called ed-dynFWA, to enhance its local search ability. To validate the performance of ed-dynFWA, it is used to participate in the CEC 2015 competition for solving learning based optimization problems.

## I. INTRODUCTION

Fireworks algorithm (FWA) is one swarm intelligence algorithms designed by Tan and Zhu in 2010 [1]. Compared with other swarm intelligence algorithms, such as particle swarm optimization and ant colony optimization, FWA presents a novel search mechanism with inspiration derived from the observation of fireworks explosion in the night sky. In fact, the fireworks' explosion process can be used as the search strategy for the global optimum of an optimization problem. Fig. 1 presents the comparison between the firework explosion and the local search operation for the global optimum. For an optimization problem, if one local region is thought as promising, then the fireworks could be set off to this region and the explosion sparks will be generated to perform the local search. After the evaluation of the fireworks' fitness according to the optimization problem, if one evaluated solution is with good fitness, then the region that the firework locates is marked as one promising region and will be searched thoroughly in the next iteration, *i.e.* more explosion sparks will be generated. Here, it is necessary to point out that the successful implementation of FWA depends on that the local region of the fireworks should have similar properties, which means “the closeness implies a metric” [2].

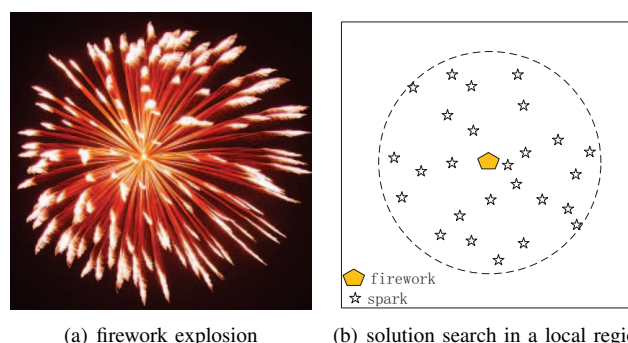


Fig. 1. Comparison between the firework explosion and the solution search in a local region for optimization problems.

The fireworks explosion process is a kind of sampling. Through sampling in the potential region, the algorithm tries to estimate the properties of the local regions. Moreover, the cooperative strategy among the different sampling regions enables the locations with better fitness to have higher sampling probabilities while those with worse fitness will have smaller sampling probabilities. After the explosion process, the Gaussian sparks are also introduced to increase the sampling diversity. Then, the candidates from the set which includes fireworks, explosion sparks and Gaussian sparks are selected as fireworks for the next iteration. The FWA continues the search till the stop criterions are met.

So far, a number of researchers have promoted the developments of FWA in three aspects: theoretical analysis [3], algorithmic developments [4]–[17] and applications [10], [18]–[28]. For algorithmic developments, they can be classified into single objective algorithm developments [4]–[9], some hybrid works [10]–[14], multi-objective developments [15], [16] and parallel implementations [17]. For applications, FWA has been used for digital filters design [10], non-negative matrix factorization calculation [18]–[20], pattern recognition [21], spam detection [22], network reconfiguration [23], [24], swarm robots [25], truss mass minimization [26], selective harmonic elimination problem [27], clustering [28], and so on. The

successful applications in practical problems indicate that FWA is one promising algorithm, which is worthy of further research.

**Synopsis.** Section II presents a detailed introduction of FWA and dynFWA. In Section III, the performance analyses of different dimension numbers in the explosive search process of dynFWA is firstly introduced. Based on the analyses, the exponentially decreased dimension number strategy for explosive search manner is introduced, and thus forms the ed-dynFWA. Then, the ed-dynFWA is used to participate in the CEC2015 competition and results on the learn based problems are recorded in Section IV. Finally, the conclusion is drawn in Section V.

## II. THE CONVENTIONAL FWA AND DYNFWA

This section will present detailed descriptions of conventional FWA [1] and dynFWA [8].

Without loss of generality, the fireworks algorithms are assumed to search the global minimal solution  $\mathbf{x}$  of the optimization problem in the form:

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x}), \quad (1)$$

where  $f: \mathbb{R}^N \rightarrow \mathbb{R}$  is a minimization problem, and  $\Omega$  is the feasible region, and the dimension of  $\mathbf{x}$  is  $D$ .

### A. The Conventional FWA

In FWA, at the beginning,  $N$  fireworks are initialized, and the fitness  $f(X_i)$  of the fireworks' locations ( $X_i$ ) are evaluated. For each firework, the explosion amplitude  $A_i$  and explosion sparks number  $S_i$  are calculated as follows.

$$A_i = \hat{A} \times \frac{f(X_i) - \min_i(f(X_i)) + \varepsilon}{\sum_{i=1}^N (f(X_i) - \min_i(f(X_i)) + \varepsilon)}, \quad (2)$$

$$S_i = \hat{M} \times \frac{\max_i(f(X_i)) - f(X_i) + \varepsilon}{\sum_{i=1}^N (\max_i(f(X_i)) - f(X_i)) + \varepsilon}, \quad (3)$$

where  $i = 1, 2, \dots, N$ .  $\hat{A}$  and  $\hat{M}$  are two constants to control the explosion amplitudes and explosion sparks number.  $\varepsilon$  is a machine epsilon.

In addition, the overwhelming effects of good/bad fireworks are bounded using following equation:

$$S_i = \begin{cases} \text{round}(a \cdot \hat{M}) & \text{if } S_i < a \cdot \hat{M}, \\ \text{round}(b \cdot \hat{M}) & \text{if } S_i > b \cdot \hat{M}, \\ \text{round}(S_i) & \text{otherwise.} \end{cases} \quad (4)$$

where  $a$  and  $b$  are two constant values to control the lower and upper bound for  $S_i$ .

Then, for each firework  $X_i, i = 1, 2, \dots, N$ , the explosion sparks are generated to perform the local search within the explosion amplitude, respectively. Alg.1 gives the process of generating one explosion spark for firework  $X_i$ .

To increase the diversity of explosion sparks swarm, FWA also introduces the Gaussian sparks. Alg.2 gives the process of generating one Gaussian spark for firework  $X_i$ .

Then, the candidate with minimal fitness among the fireworks, explosion sparks, and Gaussian sparks is selected as

---

### Algorithm 1 Generating one "explosion spark" in FWA

---

- 1: Initialize the position of one explosion spark:  $\tilde{X}_i = X_i$
  - 2: Calculate the explosion dimension number:  
 $n = \text{round}(D \cdot \text{rand}(0, 1))$
  - 3: Generate the selected dimensions set, which contains  $n$  dimensions.
  - 4: Determinate the offset displacement:  
 $\Delta X = A_i \times \text{rand}(-1, 1)$
  - 5: **for** each selected dimension of  $\tilde{X}_i^k$  **do**
  - 6:    $\hat{X}_i^k = \tilde{X}_i^k + \Delta X$
  - 7:   **if**  $\hat{X}_i^k$  is out of the search range **then**
  - 8:     map  $\hat{X}_i^k$  to the feasible range.
  - 9:   **end if**
  - 10: **end for**
- 

---

### Algorithm 2 – Generating one "Gaussian spark" in FWA

---

- 1: Initialize the position of one Gaussian spark:  $\tilde{X}_i = X_i$
  - 2: Calculate the explosion dimension number:  
 $n = \text{round}(D \cdot \text{rand}(0, 1))$
  - 3: Generate the selected dimensions set, which contains  $n$  dimensions.
  - 4: Determinate the Gaussian displacement:  
 $e = \text{Gaussian}(1, 1)$
  - 5: **for** each selected dimension of  $\tilde{X}_i^k$  **do**
  - 6:    $\hat{X}_i^k = \tilde{X}_i^k \times e$
  - 7:   **if**  $\hat{X}_i^k$  is out of the search range **then**
  - 8:     map  $\hat{X}_i^k$  to the feasible range.
  - 9:   **end if**
  - 10: **end for**
- 

one firework to the next iteration firstly, while the rest of fireworks are selected randomly with probability according to their crowdness [1]. Then, the selected fireworks continue the process of generating explosion sparks and Gaussian sparks till the terminal criterion is met.

### B. The dynFWA

In dynFWA, the fireworks are classified into two groups, the core firework (CF) and non-core fireworks (non-CFs). The definitions of CF and non-CFs are as follows:

**Definition:** CF/non-CF, among the fireworks in each iteration, the firework with best fitness is denoted as CF and the rest of fireworks except for the CF are denoted as non-CFs.

Compared with non-CFs, the CF is with smaller explosion amplitude and will perform the local search while non-CFs are with bigger explosion amplitudes which are responsible for global search. Moreover, the biggest difference between the two is that the CF has very high probability to generate the best candidate which will be selected to the next iteration as firework. Thus, the dynamic strategy is taken on CF, rather than non-CFs.

In dynFWA, for the non-CFs, their explosion amplitudes  $A_i$  in each iteration  $t$  are still calculated by Eq. (2), while for the

CF, the  $A_{CF}(t)$  is updated as follows.

$$A_{CF}(t) = \begin{cases} C_a \times A_{CF}(t-1) & \text{if } f(X_{CF}(t)) < f(X_{CF}(t-1)), \\ C_r \times A_{CF}(t-1) & \text{otherwise.} \end{cases} \quad (5)$$

where  $C_a$  and  $C_r$  are two parameters to control the amplification and reduction of explosion amplitudes. Moreover,  $A_{CF}(1)$  is usually initiated with a constant value.

From Eq.(5), it can be seen that if the best fitness found by the fireworks has been updated, then  $A_{CF}$  will be amplified in the next iteration, as  $C_a > 1$ , otherwise, the  $A_{CF}$  will be reduced as  $C_r < 1$ .

For an optimization problem, the objective is to find the global optimum as soon as possible, which contains two principles: *i*) the fireworks swarm should make sure that they move towards to global optimum, *i.e.* their best fitness needs be updated consecutively, *ii*) the fireworks swarm had better use large explosion amplitude to make it move as fast as possible to the global optimum. For dynFWA, the amplification of the CF's explosion amplitude makes it run faster, and the reduction of explosion amplitude will increase the probability that the firework can find a better position. More details can be found in [8].

In addition, dynFWA also removes the Gaussian sparks from FWA and only maintains the explosive search manner for the optimization.

---

**Algorithm 3** – Generating one “explosion sparks” in dynFWA

---

- 1: Initialize the position of one explosion spark:  $\hat{X}_i = X_i$
  - 2: Set  $z^k = \text{floor}(\text{rand}(0,1) + \beta)$ ,  $k = 1, 2, \dots, D$ ,  $\beta = 0.5$
  - 3: **for** each dimension of  $\hat{X}_i^k$ , where  $z^k == 1$  **do**
  - 4:    $\hat{X}_i^k = \hat{X}_i^k + A_i \times \text{rand}(-1, 1)$
  - 5:   **if**  $\hat{X}_i^k$  out of bounds **then**
  - 6:     map  $\hat{X}_i^k$  to the feasible range.
  - 7:   **end if**
  - 8: **end for**
- 

The explosion sparks generating process for dynFWA can be found in Alg.3 while the framework of dynFWA is shown in Alg. 4.

---

**Algorithm 4** – The Framework of dynFWA

---

- 1: Initialize the fireworks  $X_i$  and evaluate the fitness  $f(X_i)$
  - 2: **while** stop criteria is not met **do**
  - 3:   Calculate the numbers of explosion sparks,  $S_i$
  - 4:   Calculate the explosion amplitudes  $A_i$  (cf. Eq. 2, Eq.5)
  - 5:   **for** each firework  $X_i$  **do**
  - 6:     Generate the explosion sparks (cf. Alg 3)
  - 7:     Evaluate the fitness of explosion sparks
  - 8:   **end for**
  - 9:   Select fireworks for the next iteration
  - 10: **end while**
- 

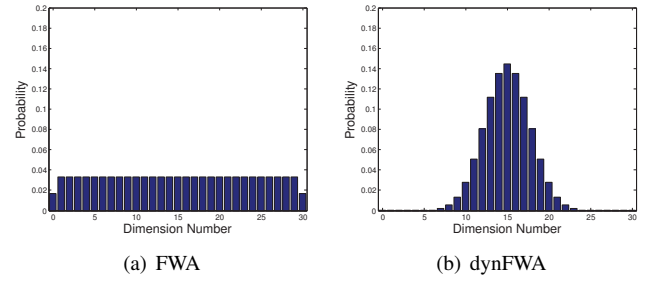


Fig. 2. Comparison between FWA and dynFWA explosion dimension number probability density function ( $D = 30$ ).

### III. EXPONENTIALLY DECREASED DYNAMIC DIMENSION NUMBER STRATEGY IN DYNFWA

In this section, we present the performance analysis of dimension number in the process of generating the explosion sparks, and then we propose the exponentially decreased dimension number strategy for dynFWA.

#### A. Performance Analysis of Dimension Number in Explosion Process

For FWA, the key feature compared with other swarm intelligence algorithms lies in the explosive search manner. The explosion amplitude, the explosion sparks number and the manner how the explosion sparks are generated are the parameters controlling the explosive sampling process, thus finally determinate the performance of FWA.

Compare the process of generating explosion sparks between FWA (cf Alg. 1) with dynFWA (cf Alg. 3), it can be seen that the explosion sparks are generated in two different manners. In FWA, the algorithm firstly determinate the total number of dimensions to perform the explosive mutation (cf. Alg.1 in line 2), which is under a uniform distribution in general (as when the dimension number is 0 or 30, the probability is smaller compared with the rest of dimension numbers, it is not a strict uniform distribution). For dynFWA, each dimension has the 0.5 probability to perform the explosive mutation (cf. Alg3 in line 3 and 4). So, the number of dimensions that will perform the explosion mutation is under a binomial distribution. The comparison of the two explosion dimension number probability density functions can be found in Fig. 2.

In pervious work of EFWA in [5], the experimental results are unintentionally based on the dimension selection method of conventional FWA, which varies the number of selected dimensions with uniform distribution. However, the pseudo algorithm presented in [5] is that each dimension have the probability of 0.5 to take the explosion mutation, and finally the number of dimensions is under the binomial distribution. In [29], experiments are designed to compare the performance of these two dimension number selection methods. Experimental results on the 12 basic functions suggest that EFWA has the better local search ability when it takes the uniform distributed dimension number strategy. In summary, the distribution of the explosion dimension numbers is also one important factor that will influence the performance of fireworks algorithms.

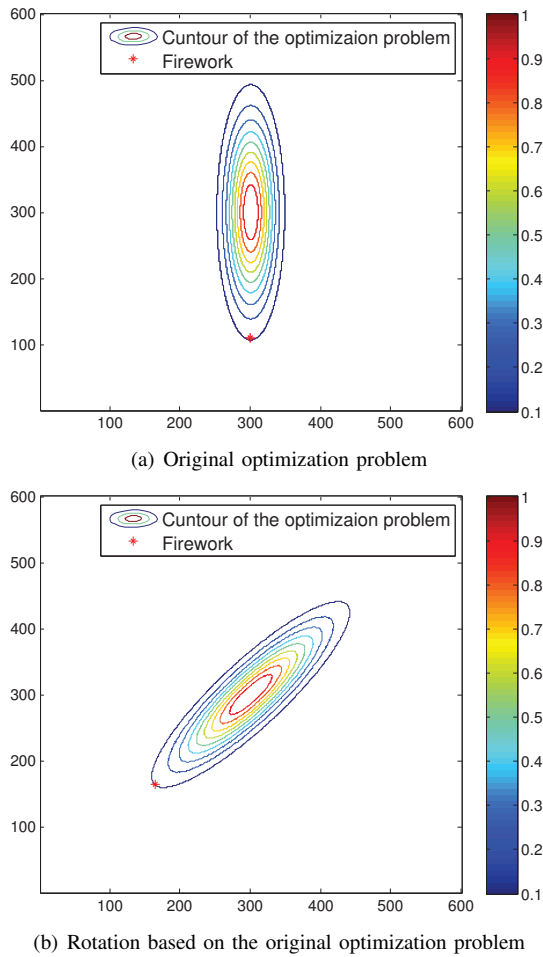


Fig. 3. Contour of one 2-dimension optimization problem

In the following paragraphs, we are going to give an example of how the dimension number of explosive sparks influence the algorithm's performance.

Assume a 2-dimensional optimization problem shown in Fig. 3(a), the character for this optimization problem is that the global optimum locates in a very narrow region, and the direction of this narrow region is parallel with one of the axes. For this optimization problem, if one firework now locates at the edge of this local region, see Fig. 3(a), then if we set the explosion dimension number to one, it will have high probability to generate explosion sparks with good fitness in this narrow region compared with when the explosion dimension number is set to two. However, if we rotate this optimization problem with an angle, the situation will be different. The two dimensional explosive search strategy will gains better performance compared with one dimensional explosion, see Fig. 3(b).

Readers may feel confused about this two optimization problems' comparison, as we just add the rotation operation to the original optimization problem, even without any shifting of the global optimal position. Here, we should clarify that it is due to the fact that in FWA and its variants, the explosive mutation in each dimension is designed to be independent, *i.e.*

the explosion mutation value is calculated in each dimension independently (cf. line 4 in Alg.3).

In fact, for a local region of a high dimensional optimization problem, usually the dimension number that greatly influence the performance is limited compared with the total dimension number of the optimization problem. Thus, for FWA, to get a better local search ability, the best dimension number for explosion is supposed to be influenced by the property of the local region that the fireworks locate. The ideal strategy in FWA is to use the explosion sparks to evaluate the property of this local region and to determine optimal explosion dimension number, and even what are the dimensions.

### B. Exponentially Decreased Dimension Number Strategy

To investigate the influences of the different dimension numbers of explosion to the performance of dynFWA, the following criterion is introduced.

$$\gamma_i = \frac{TS_i}{TA_i}, \quad (6)$$

where  $TS_i$  denotes the times that the explosion sparks are generated with  $i$  explosion dimensions and have better fitness than the firework while  $TA_i$  denotes the total times that the explosion sparks are generated with  $i$  explosion dimensions. Here, we only consider the CF for calculation of  $\gamma_i$ .

Fig. 4 gives the statistical results on two functions of CEC 2013 competition problems,  $f_1$  and  $f_{28}$  [30]. Here,  $f_1$  is an uni-modal problem while  $f_{28}$  is a multi-modal problem. Results suggest that the smaller dimension numbers will have higher probability to generate the explosion sparks with better fitness. The similar observation can be found among the rest of functions from  $f_2$  to  $f_{27}$ .

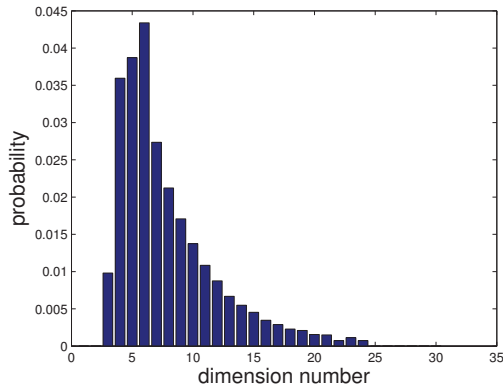
In fact, the best strategy is to take an adaptive dimension number strategy during the optimization process. For an optimization problem, when the firework locates in the wide region, the bigger dimension number of explosion strategy seems to gain more diversity, while it comes to the narrow region, the smaller dimension number of explosion strategy may gain higher improvement.

According to Alg. 3, it can be seen that in dynFWA, each dimension has the  $\beta = 0.5$  probability to be selected, which is a constant value. However, according to the statistical results from Fig 4, it seems that smaller explosion dimension number may have better performance.

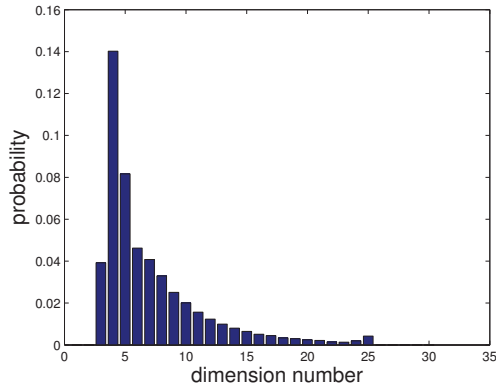
For simplicity, to participate in the CEC 2015 competition, we just take the exponentially decreased dimension number of explosion, *i.e.* the parameter  $\beta$  in Alg. 3 is a exponentially decreased value rather than a constant parameter. The parameter  $\beta$  updates its value as follows.

$$\beta = \beta * C_{ed} \quad (7)$$

where  $C_{ed}$  is a parameter to control the expectation of the selected dimension number, and  $\beta$  starts with 0.5 and updates its value every other  $I$  iterations. Then the expectation of explosion dimension number is  $\beta D$ . The final algorithm



(a)  $f_1$ , dimension is 30



(b)  $f_{28}$ , dimension is 30

Fig. 4. The probability that the different explosion dimensions make significant improvements.

that dynamic search fireworks algorithm with exponentially decreased dimension numbers is called ed-dynFWA.

#### IV. EXPERIMENTS

##### A. Experimental Setup

Here, we use ed-dynFWA to participate in the CEC 2015 competition. The details of CEC 2015 competition problems can be found in Tab. I [31].

Compared with the previous CEC competitions, the CEC 2015 competition allows participants to specify the different parameters for each optimization function.

For ed-dynFWA, it includes a number of parameters, only part of them are picked to tune for the competition problems. Tab. II gives a brief summary of the parameters we are going to tune for better performance in ed-dynFWA. For the rest of parameters, they are set identical to dynFWA [8].

For each optimisation problem, the ed-dynFWA is performed 51 runs and the final results are recorded with  $D \times 10000$  function evaluations are presented. The experimental platform is MATLAB 2011b (Windows 7; Intel Core i7-2600 CPU @ 3.7 GHz; 8 GB RAM).

TABLE I  
DETAILS OF CEC2015 COMPETITION PROBLEMS (UNI.=UNIMODAL, M.=MULTIMODAL, R.=ROTATED, S.=SHIFTED, C.=COMPOSITION) [31].

Func.	Type	Details	Optimum
$f_1$	Uni.	R High Conditioned Elliptic	100
$f_2$		R Bent Cigar	200
$f_3$	Simple M.	S. R. Ackley	300
$f_4$		S. R. Rastrigin	400
$f_5$		S. R. Schwefel	500
$f_6$	H. Function	H. Function 1 (N=3)	600
$f_7$		H. Function 1 (N=4)	700
$f_8$		H. Function 1 (N=5)	800
$f_9$	C. Functions	C. Function 1 (N=3)	900
$f_{10}$		C. Function 1 (N=3)	1000
$f_{11}$		C. Function 1 (N=3)	1100
$f_{12}$		C. Function 1 (N=3)	1200
$f_{13}$		C. Function 1 (N=3)	1300
$f_{14}$		C. Function 1 (N=3)	1400
$f_{15}$		C. Function 1 (N=3)	1500

TABLE II  
PARAMETERS IN ED-DYNFWA

Parameter	Details
$\hat{M}$	explosion sparks number
$C_r$	CF's reduction factor
$C_a$	CF's amplification factor
$C_{ed}$	exponentially reduction factor for dimension selected probability
$I$	iteration interval for update the dimension number selected probability

##### B. Experiments Results

At first, the parameters for each optimization problem is tuned. Here, the 30-dimensional problems are used to tune the parameters and then the tuned parameters are used for the 10-dimensional, 50-dimensional and 100-dimensional optimization problems.

Tab. III lists the parameters tuned for the 30-dimensional problems.

TABLE III  
PARAMETERS TUNED IN ED-DYNFWA FOR 30-DIMENSIONAL FUNCTIONS

Func.	$\hat{M}$	$C_r$	$C_a$	$C_{ed}$	$I$
$f_1$	120	0.93	1.1	0.95	70
$f_2$	150	0.87	1.2	0.95	30
$f_3$	180	0.93	1.2	0.95	70
$f_4$	150	0.87	1.1	0.95	50
$f_5$	150	0.87	1.3	0.97	30
$f_6$	120	0.9	1.1	0.95	70
$f_7$	180	0.93	1.3	0.97	50
$f_8$	120	0.87	1.2	0.97	30
$f_9$	180	0.93	1.3	0.95	50
$f_{10}$	120	0.87	1.1	0.97	30
$f_{11}$	150	0.93	1.2	0.95	30
$f_{12}$	180	0.93	1.1	0.99	70
$f_{13}$	180	0.93	1.2	0.95	30
$f_{14}$	120	0.87	1.3	0.95	70
$f_{15}$	120	0.87	1.1	0.95	30

The experimental results on the benchmark functions can be found in Tab. IV, Tab. V, Tab. VI, Tab. VII.

From the experimental results in all the dimensional problems, we can see that the results of  $f_3$ ,  $f_7$ ,  $f_{13}$ ,  $f_{15}$  are

very stable and with quite small standard deviations. For the performance, it can be seen that ed-dynFWA almost gains the optimum on  $f_{13}$ .

In addition, for the 10-dimensional problems, the optimized results are close to the optimum on functions  $f_3, f_4, f_7, f_9, f_{12}$  and  $f_{15}$  and it also achieves very promising results on functions  $f_3, f_4, f_7, f_9, f_{12}, f_{15}$  for 30-dimensional problems. For the 50-dimensional problems and the 100-dimensional problems, promising results are also returned in  $f_3, f_7, f_9, f_{12}, f_{15}$ .

Compare the performance in different dimensional optimization problems, it can be seen that with the increasing of dimensions, the results returned by ed-dynFWA get worse, due to the ‘‘curse of dimensionality’’.

TABLE IV  
EXPERIMENTAL RESULTS ON 10-DIMENSIONAL FUNCTIONS

Func.	Best	Worst	Median	Mean	Std.
$f_1$	5.26E+02	2.43E+05	6.39E+04	8.35E+04	6.44E+04
$f_2$	2.02E+02	3.29E+04	2.93E+03	7.06E+03	8.74E+03
$f_3$	3.20E+02	3.20E+02	3.20E+02	3.20E+02	1.30E-05
$f_4$	4.05E+02	4.37E+02	4.16E+02	4.18E+02	8.53E+00
$f_5$	5.12E+02	1.50E+03	9.04E+02	9.24E+02	2.42E+02
$f_6$	6.96E+02	7.94E+03	1.39E+03	2.05E+03	1.58E+03
$f_7$	7.00E+02	7.02E+02	7.01E+02	7.01E+02	3.30E-01
$f_8$	8.37E+02	1.94E+04	3.35E+03	5.15E+03	5.10E+03
$f_9$	1.00E+03	1.00E+03	1.00E+03	1.00E+03	3.36E-01
$f_{10}$	1.15E+03	1.33E+06	1.44E+03	2.76E+04	1.86E+05
$f_{11}$	1.10E+03	1.40E+03	1.40E+03	1.28E+03	1.47E+02
$f_{12}$	1.30E+03	1.32E+03	1.31E+03	1.31E+03	2.20E+00
$f_{13}$	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.24E-02
$f_{14}$	1.50E+03	1.01E+04	8.23E+03	8.24E+03	1.88E+03
$f_{15}$	1.60E+03	1.60E+03	1.60E+03	1.60E+03	0.00E+00

TABLE V  
EXPERIMENTAL RESULTS ON 30-DIMENSIONAL FUNCTIONS

Func.	Best	Worst	Median	Mean	Std.
$f_1$	3.24E+05	2.17E+06	9.30E+05	1.02E+06	4.61E+05
$f_2$	2.26E+02	1.44E+04	2.43E+03	4.10E+03	4.11E+03
$f_3$	3.20E+02	3.20E+02	3.20E+02	3.20E+02	3.19E-06
$f_4$	4.52E+02	6.32E+02	5.17E+02	5.20E+02	3.19E+01
$f_5$	1.87E+03	4.62E+03	3.50E+03	3.52E+03	7.06E+02
$f_6$	7.90E+03	2.88E+05	8.81E+04	9.98E+04	6.55E+04
$f_7$	7.08E+02	7.84E+02	7.13E+02	7.16E+02	1.41E+01
$f_8$	1.58E+04	4.90E+05	1.04E+05	1.38E+05	1.12E+05
$f_9$	1.01E+03	1.29E+03	1.01E+03	1.03E+03	6.09E+01
$f_{10}$	4.28E+03	1.52E+05	1.89E+04	3.12E+04	3.17E+04
$f_{11}$	1.40E+03	2.09E+03	1.81E+03	1.82E+03	1.11E+02
$f_{12}$	1.31E+03	1.40E+03	1.32E+03	1.32E+03	2.06E+01
$f_{13}$	1.30E+03	1.30E+03	1.30E+03	1.30E+03	2.44E-03
$f_{14}$	3.52E+04	4.79E+04	4.60E+04	4.49E+04	3.61E+03
$f_{15}$	1.60E+03	1.60E+03	1.60E+03	1.60E+03	0.00E+00

Moreover, the computation complexity of the algorithm is listed in Tab. VIII, we can see that with the increasing of the dimension, the optimization problems are supposed to have higher complexity, thus the runtime goes higher.

## V. CONCLUSION

In this paper, we have presented comprehensive analyses on the influence of different explosion dimension numbers

TABLE VI  
EXPERIMENTAL RESULTS ON 50-DIMENSIONAL FUNCTIONS

Func.	Best	Worst	Median	Mean	Std.
$f_1$	7.52E+05	5.42E+06	2.89E+06	2.83E+06	1.09E+06
$f_2$	2.07E+02	4.53E+04	4.47E+03	7.57E+03	9.58E+03
$f_3$	3.20E+02	3.20E+02	3.20E+02	3.20E+02	3.36E-06
$f_4$	5.57E+02	8.44E+02	6.63E+02	6.72E+02	5.46E+01
$f_5$	4.04E+03	7.77E+03	5.88E+03	5.83E+03	8.96E+02
$f_6$	1.80E+05	1.36E+06	3.89E+05	4.92E+05	2.66E+05
$f_7$	7.18E+02	7.95E+02	7.53E+02	7.59E+02	1.97E+01
$f_8$	1.23E+05	1.24E+06	5.72E+05	5.93E+05	2.64E+05
$f_9$	1.00E+03	1.00E+03	1.00E+03	1.00E+03	3.31E-01
$f_{10}$	6.65E+03	2.25E+05	5.12E+04	5.99E+04	4.01E+04
$f_{11}$	2.00E+03	2.59E+03	2.26E+03	2.26E+03	1.28E+02
$f_{12}$	1.32E+03	1.40E+03	1.40E+03	1.37E+03	3.86E+01
$f_{13}$	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.96E-02
$f_{14}$	5.41E+04	7.39E+04	5.41E+04	5.49E+04	3.85E+03
$f_{15}$	1.60E+03	1.60E+03	1.60E+03	1.60E+03	0.00E+00

TABLE VII  
EXPERIMENTAL RESULTS ON 100-DIMENSIONAL FUNCTIONS

Func.	Best	Worst	Median	Mean	Std.
$f_1$	2.37E+06	1.17E+07	5.69E+06	5.88E+06	1.83E+06
$f_2$	2.01E+02	1.84E+04	3.33E+03	4.73E+03	4.70E+03
$f_3$	3.20E+02	3.20E+02	3.20E+02	3.20E+02	6.00E-07
$f_4$	9.28E+02	1.55E+03	1.16E+03	1.15E+03	1.07E+02
$f_5$	1.06E+04	1.66E+04	1.40E+04	1.39E+04	1.41E+03
$f_6$	7.43E+05	2.94E+06	1.79E+06	1.77E+06	5.21E+05
$f_7$	7.42E+02	8.80E+02	8.25E+02	8.29E+02	3.40E+01
$f_8$	5.31E+05	3.68E+06	1.72E+06	1.77E+06	7.59E+05
$f_9$	1.01E+03	1.82E+03	1.01E+03	1.04E+03	1.43E+02
$f_{10}$	6.63E+03	5.34E+04	1.83E+04	2.15E+04	1.12E+04
$f_{11}$	3.35E+03	4.34E+03	3.71E+03	3.72E+03	1.95E+02
$f_{12}$	1.32E+03	1.40E+03	1.33E+03	1.36E+03	3.90E+01
$f_{13}$	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.59E-01
$f_{14}$	1.10E+05	1.63E+05	1.10E+05	1.11E+05	7.35E+03
$f_{15}$	1.61E+03	1.62E+03	1.61E+03	1.61E+03	2.63E+00

to the performance of FWA and its variants, and based on the analyses, the exponentially decreased dimension number strategy is introduced to enhance the performance of dynFWA (ed-dynFWA).

Then we take the ed-dynFWA to participate in the CEC 2015 competition, and the results on 15 functions with dimension number set to 10, 30, 50 and 100 are recorded. Optimization results on the benchmark suite suggest that ed-dynFWA is one promising algorithm that can work well to alleviate these problems. However, there are still many functions that ed-dynFWA can not find the global optimum, which needs further research.

TABLE VIII  
COMPUTATION COMPLEXITY COMPARISON

Dim	$T_0$ (s)	$T_1$ (s)	$T_2$ (s)	$\frac{T_2-T_1}{T_0}$
10	0.1370	2.0444	2.0555	0.0810
30	0.1370	2.3222	2.6669	2.5161
50	0.1370	2.8126	3.4365	4.5540
100	0.1370	4.6102	6.0284	10.3518

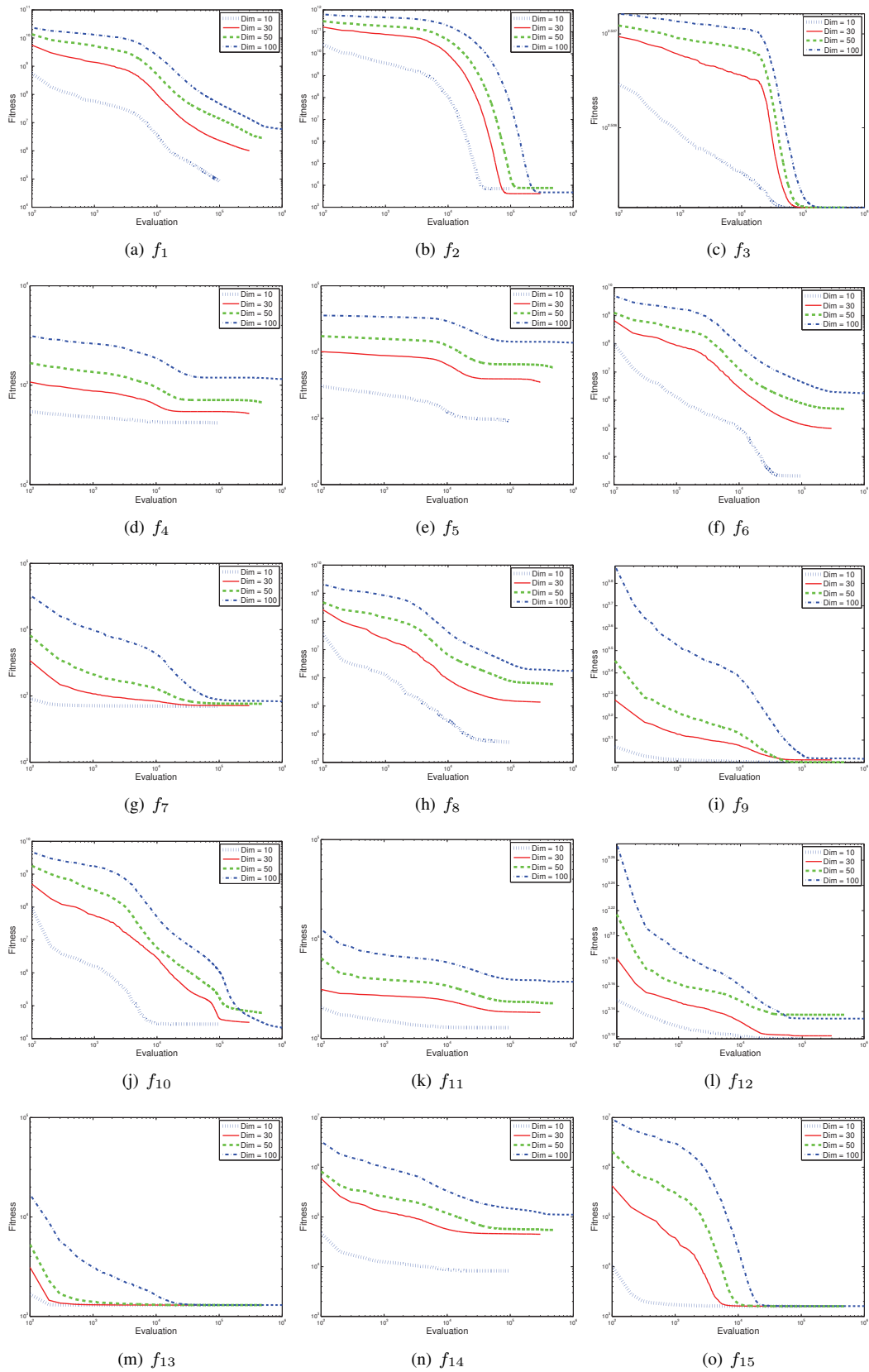


Fig. 5. Convergence curves of ed-dynFWA for CEC2015 competitions

## ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of China (NSFC) under grant no. 61375119 and 61170057, and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China with grant no. 2015CB352300. Prof. Ying Tan is the corresponding author.

## REFERENCES

- [1] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," *Advances in Swarm Intelligence*, pp. 355–364, 2010.
- [2] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, *The elements of statistical learning*. Springer, 2009, vol. 2, no. 1.
- [3] J. Liu, S. Zheng, and Y. Tan, "Analysis on global convergence and time complexity of fireworks algorithm," in *Evolutionary Computation (CEC)*, 2014 IEEE Congress on. IEEE, 2014, pp. 3207–3213.
- [4] Y. Pei, S. Zheng, Y. Tan, and T. Hideyuki, "An empirical study on influence of approximation approaches on enhancing fireworks algorithm," in *Proceedings of the 2012 IEEE Congress on System, Man and Cybernetics*. IEEE, 2012, pp. 1322–1327.
- [5] S. Zheng, A. Janecek, and Y. Tan, "Enhanced fireworks algorithm," in *Evolutionary Computation (CEC)*, 2013 IEEE Congress on. IEEE, 2013, pp. 2069–2077.
- [6] 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.
- [7] B. Zhang, M. Zhang, and Y.-J. Zheng, "Improving enhanced fireworks algorithm with new gaussian explosion and population selection strategies," in *Advances in Swarm Intelligence*. Springer, 2014, pp. 53–63.
- [8] 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.
- [9] J. Li, S. Zheng, and Y. Tan, "Adaptive fireworks algorithm," in *Evolutionary Computation (CEC)*, 2014 IEEE Congress on, July 2014, pp. 3214–3221.
- [10] H. Gao and M. Diao, "Cultural firework algorithm and its application for digital filters design," *International Journal of Modelling, Identification and Control*, vol. 14, no. 4, pp. 324–331, 2011.
- [11] Y. Zheng, X. Xu, and H. Ling, "A hybrid fireworks optimization method with differential evolution," *Neurocomputing*, 2012.
- [12] C. Yu, J. Li, and Y. Tan, "Improve enhanced fireworks algorithm with differential mutation," in *Systems, Man and Cybernetics (SMC)*, 2014 IEEE International Conference on. IEEE, 2014, pp. 264–269.
- [13] B. Zhang, M.-X. Zhang, and Y.-J. Zheng, "A hybrid biogeography-based optimization and fireworks algorithm," in *Evolutionary Computation (CEC)*, 2014 IEEE Congress on. IEEE, 2014, pp. 3200–3206.
- [14] N. BACANIN, M. TUBA, and M. BEKO, "Hybridized fireworks algorithm for global optimization."
- [15] Y.-J. Zheng, Q. Song, and S.-Y. Chen, "Multiobjective fireworks optimization for variable-rate fertilization in oil crop production," *Applied Soft Computing*, vol. 13, no. 11, pp. 4253–4263, 2013.
- [16] L. Liu, S. Zheng, and Y. Tan, "S-metric based multi-objective fireworks algorithm," in *Evolutionary Computation (CEC)*, 2014 IEEE Congress on. IEEE, 2015.
- [17] K. Ding, S. Zheng, and Y. Tan, "A gpu-based parallel fireworks algorithm for optimization," in *Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference*, ser. GECCO 2013. New York, NY, USA: ACM, 2013, pp. 9–16.
- [18] A. Janecek and Y. Tan, "Using population based algorithms for initializing nonnegative matrix factorization," in *Proceedings of the second international conference on Advances in swarm intelligence*, ser. ICSI'11. Springer-Verlag, 2011, pp. 307–316.
- [19] —, "Iterative improvement of the multiplicative update nmf algorithm using nature-inspired optimization," in *Natural Computation (ICNC)*, 2011 Seventh International Conference on, vol. 3. IEEE, 2011, pp. 1668–1672.
- [20] —, "Swarm intelligence for non-negative matrix factorization," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 2, no. 4, pp. 12–34, 2011.
- [21] S. Zheng and Y. Tan, "A unified distance measure scheme for orientation coding in identification," in *Information Science and Technology*, 2013 IEEE Congress on. IEEE, 2013, pp. 979–985.
- [22] W. He, G. Mi, and Y. Tan, "Parameter optimization of local-concentration model for spam detection by using fireworks algorithm," in *Advances in Swarm Intelligence*. Springer, 2013, pp. 439–450.
- [23] A. M. Imran, M. Kowsalya, and D. Kothari, "A novel integration technique for optimal network reconfiguration and distributed generation placement in power distribution networks," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 461–472, 2014.
- [24] A. Mohamed Imran and M. Kowsalya, "A new power system reconfiguration scheme for power loss minimization and voltage profile enhancement using fireworks algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 62, pp. 312–322, 2014.
- [25] Z. Zheng and Y. Tan, "Group explosion strategy for searching multiple targets using swarm robotic," in *Evolutionary Computation (CEC)*, 2013 IEEE Congress on. IEEE, 2013, pp. 821–828.
- [26] N. Pholdee and S. Bureerat, "Comparative performance of meta-heuristic algorithms for mass minimisation of trusses with dynamic constraints," *Advances in Engineering Software*, vol. 75, pp. 1–13, 2014.
- [27] R. Rajaram, K. Palanisamy, S. Ramasamy, and P. Ramanathan, "Selective harmonic elimination in pwm inverter using fire fly and fire works algorithm," 2014.
- [28] X. Yang and Y. Tan, "Sample index based encoding for clustering using evolutionary computation," in *Advances in Swarm Intelligence*. Springer, 2014, pp. 489–498.
- [29] S. Zheng, A. Janecek, and Y. Tan. (2013) Announcements of efwa. [Online]. Available: <http://www.cil.pku.edu.cn/research/fwa/activities/2014/announcements-of-efwa/index.html>
- [30] 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.
- [31] B. Qu, J. Liang, P. Suganthan, and Q. Chen, "Problem definitions and evaluation criteria for the cec 2015 competition on single objective multi-niche optimization," 2014.