2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Fireworks Algorithm with Differential Mutation for Solving the CEC 2014 Competition Problems

Chao Yu, Lingchen Kelley, Shaoqiu Zheng, and Ying Tan

***Abstract*— The idea of fireworks algorithm (FWA) is inspired by the fireworks explosion in the sky at night. When a firework explodes, a shower of sparks appear around it. In this way, the adjacent area of the firework is searched. By controlling the amplitude of the explosion, the ability of local search for FWA is guaranteed. The way of fireworks algorithm searching the surrounding area can be further improved by differential mutation operator, forming an algorithm called FWA-DM. In this paper, the benchmark suite in the competition of congress of evolutionary computation (CEC) 2014 is used to test the performance of FWA-DM.**

I. INTRODUCTION

Fireworks algorithm (FWA) is put forward by Tan and Zhu in 2010 [1]. As a firework explodes, a shower of sparks appears around the firework while the adjacent area is illuminated. The explosion operator in FWA is to find global minimum values by searching the surrounded area of an individual. As a swarm intelligence algorithm, FWA is effective for solving non-linear and complex numerical optimization problems. Moreover, the applications for FWA are various. Janecek *et al.* applied FWA to non-negative matrix factorization and got good experimental results [2]–[4]. Gao *et al.* used FWA to design digital filters and proved that FWA worked better than particle swarm optimization (PSO) [5]. He W. *et al.* applied FWA to the fields of spam detection [6].

Differential evolution (DE) algorithm was proposed by Storn and Price [7]. DE algorithm can deal with high dimension and non-linear problems. When solving multi-modal and non-linear problems, DE algorithm performs steadily and converges quickly. As a simple and effective algorithm, DE algorithm is widely researched and applied to many fields. Brest *et al.* studied the self-adaptive parameters in DE algorithm on numeric benchmark problems [8]. Das and Suganthan presented the details of DE and researched the major variants, application and theory [9]. Mallipeddi *et al.* applied ensemble of parameters and mutation strategies to DE algorithm [10]. Still, there are many people focusing on DE algorithm [11]–[15].

In this paper, DE mutation operator is introduced to fireworks algorithm (FWA) so as to form a new algorithm called FWA-DM. FWA-DM is then used to find the optima of CEC 2014 benchmark functions [16].

The remainder of the paper is organized as follows. Section II shows the process of applying DE mutation operator to FWA. Section III represents the experimental results, including the results of mean values and computational time. The conclusion is given in Section IV.

II. FIREWORKS ALGORITHM WITH DIFFERENTIAL MUTATION

A. Differential Evolution

DE algorithm belongs to a class of evolutionary algorithms. As a global optimization algorithm, DE is simple and easy to implement. The core operator in DE is mutation operator. This operator scales the difference of two individuals in the same population, producing a mutant by adding the scaled difference to a third individual. The produced mutant is then applied to its parent individual and a trial vector is generated. Last, the trial vector is compared with the parent individual and the *better* one is kept for next generation [1]. DE algorithm keeps finding better objective values until the terminal condition is met. Since DE algorithm is easy to use, it is now widely applied to many fields [17]–[21].

The process of DE is similar to other evolutionary algorithms, including population initialization, fitness function evaluation and population iteration. Algorithm 1 shows DE with DE/rand/1/bin strategy.

In Algorithm 1, the core of the strategy lies in line 9, while lines 14 to 19 shows the selection operation. Parameter NP stands for the number of individuals in a population and parameter D is the dimension of the problem. Other parameters are CR and F, which represent crossover possibility and scale factor, respectively. $rand(0,1)$ generates random numbers from the region $(0,1)$ with uniform distribution. It can be seen from Algorithm 1 that DE is simple and easy to implement.

1) Mutation Operator: As the core operator of DE algorithm, differential evolution operator plays an important role in DE algorithm. DE/rand/1 is one of the differential evolution operators. In fact, there are more mutation operators in DE algorithm. The operators are stated in a form of DE/a/b, where DE represents the DE algorithm, a stands for the way to select basic vectors and b indicates the number of vectors that involved in the mutation operation.

Chao Yu, Shaoqiu Zheng and Ying Tan (Corresponding author) are with the Key Laboratory of Machine Perception and Intelligence (Peking University), Ministry of Education, Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing, China (e-mail:chaoyu, zhengshaoqiu, ytan@pku.edu.cn). Lingchen Kelley is with the School of Humanities, Chinese History Department, Tsinghua University. (e-mail:lingchenkelley@gmail.com).

This work was supported by the Natural Science Foundation of China with grant no. 61375119, 61170057 and 60875080.

[1]Without loss of generality, the optimization problem f is treated as a minimal problem.

Algorithm 1 The process of DE/rand/1/bin algorithm

randomly generates the initial population with NP individuals
evaluates the fitness values for all individuals
while terminal condition not met **do**
  for $i = 1 \rightarrow NP$ **do**
    randomly selects $r1 \neq r2 \neq r3 \neq i$
    randomly selects j_{rand} from [1, D] /* D stands for dimension */
    for $j = 1 \rightarrow D$ **do**
      if $rand(0,1) \leq CR$ or $j == j_{rand}$ **then**
        $U_i(j) = X_{r1}(j) + F \times (X_{r2}(j) - X_{r3}(j))$
      else
        $U_i(j) = X_i(j)$
      end if
    end for
  end for
  for $j = 1 \rightarrow D$ **do**
    evaluates the fitness values for U_i
    if U_i is better than X_i **then**
      $X_i = U_i$
    end if
  end for
end while

2) Crossover Operator: There are two forms of crossover operator in DE algorithm, including binomial crossover operator and exponential crossover operator. The crossover operator generates the trial vector U_i by dealing with the mutant vector V_i and the parent vector X_i. The crossover operation can be stated as follows.

$$U_i(j) = \begin{cases} V_i(j), if\ rand(0,1) \leq CR\ or\ j == j_{rand} \\ X_i(j), otherwise \end{cases} \tag{1}$$

In Eq. 1, $rand(0,1)$ generates random numbers between 0 and 1 with uniform distribution. Parameter CR is the crossover possibility and parameter j_{rand} is a randomly selected dimension number, which varies from 1 to D.

3) Selection Operator: After producing a children population with a differential mutate operator and a crossover operator, the individuals in the children population are compared with their corresponding parent individuals by selection operation. The ones with better fitness values are then selected for next generation. The selection operation can be described as follows.

$$X_i = \begin{cases} U_i, if(f(U_i) < f(X_i)) \\ X_i, otherwise \end{cases} \tag{2}$$

In Eq. 2, $f(X_i)$ stands for the fitness value of an individual X_i. It can be seen from Eq. 2 that the better one is always kept for next generation. Therefore, DE algorithm is a steady evolutionary algorithm.

B. Fireworks Algorithm

The idea of FWA was inspired by the fireworks explosion in the night sky. When a firework explodes, a shower of sparks appears around it. In this way, the adjacent area of the spark is illuminated. The process of fireworks explosion can be treated as a good way to search the area around a specific point. Hence, when FWA searches the area, there are two parameters that have to be determined. The first parameter is the number of explosion sparks and the second parameter is the amplitude of the explosion.

S_i denotes the number of sparks for a firework X_i.

$$S_i = \hat{S} * \frac{Y_{\max} - f(x_i) + \varepsilon}{\sum_{i=1}^{N} (Y_{\max} - f(x_i)) + \varepsilon} \tag{3}$$

In Eq. 3, $\hat{S}$ is a constant that stands for the total number of sparks. Parameter Y_{max} means the fitness value of the worst individual in the population. $f(x_i)$ is the fitness value for an individual x_i, while the last parameter ε is used to prevent the denominator from becoming zero.

A_i denotes the amplitude for the i^{th} individual.

$$A_i = \hat{A} * \frac{f(x_i) - Y_{\min} + \varepsilon}{\sum_{i=1}^{N} (f(x_i) - Y_{\min}) + \varepsilon} \tag{4}$$

In Eq. 4, $\hat{A}$ is a constant representing the sum of all the amplitudes. Parameter Y_{min} means the fitness value of the best individual in the population. The meaning of $f(x_i)$ and parameter ε are the same as aforementioned.

As pointed in [22], when an amplitude is too small, it leads to useless explosion since the new generated sparks are close and similar. Therefore, a new parameter is proposed to prevent the amplitudes from being too small. A_{min} denotes the minimum of the amplitude. There are two ways to set the parameter A_{min}, as linear and non-linear decreasing method, respectively. The value of A_{min} decreasing while the number of iteration increasing.

$$(Linear) A_{\min} = A_{init} - (A_{init} - A_{final}) * Iter/MaxEval \tag{5}$$

$$\begin{aligned} (Non-linear) A_{\min} &= A_{init} - (A_{init} - A_{final}) \\ &* \sqrt{(2 * MaxEval - Iter) * Iter/MaxEval} \end{aligned} \tag{6}$$

In both Eq. 5 and Eq. 6, parameters A_{init} and A_{final} represent the initial and final minimum amplitude boundaries of the explosions. Parameter $Iter$ is the number of current function evaluations and parameter $MaxEval$ represents the maximum number of function evaluation. In this paper, the non-linear form of A_{min} is used due to empirical experiments.

FWA simulates the way of fireworks explode and searches the around areas of individuals. DE mutation operator can be used in FWA and solves the function optimization problems.

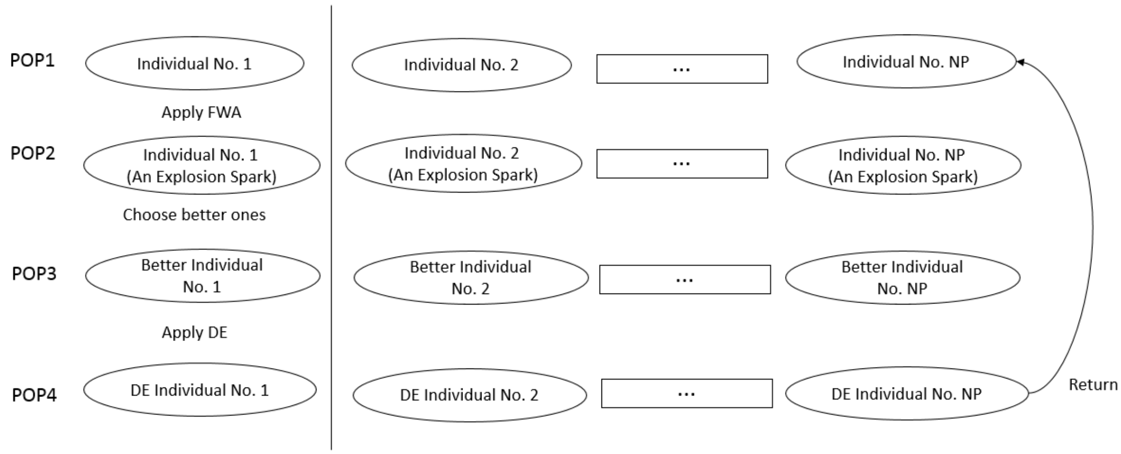

Fig. 1. The process of applying DE to FWA.

C. Apply DE to FWA

At first, NP individuals are initialized randomly with uniform distribution. The random seed is based on time. This population with NP individuals is marked as POP1. Secondly, a spark is produced around each individual within a certain amplitude. The amplitude is determined by FWA and greater than A_{min} at the same time. The explosion sparks form a population POP2. Thirdly, the individuals in POP1 are compared with the corresponding individuals in POP2 in pairs. The ones with better fitness values are kept and used to form a new population marked as POP3. Fourthly, the mutation and crossover operators in DE algorithm are applied to POP3 and a new population is generated as POP4. Finally, the selection operator is applied to POP4 and the selected individuals are used to form a new population POP1. The iteration continues until the maximum times of function evaluations are achieved or the objective function values are lower than 1e-8. In this way, DE mutation operator is applied to FWA.

The process of applying DE mutation operator to FWA is drawn in Fig.1. The first row represents a population POP1 with NP individuals. The second row shows the individuals that are generated by fireworks explosion and the individuals form a population POP2. After comparing the individuals in the first row with the second row in pairs, better ones are chosen and displayed in the third row as a population POP3. Then DE mutation operator is applied to a population POP3 and the population POP4 is produced in the last row. The better individuals between population POP3 and POP4 are selected for next iteration as a new population POP1. Moreover, the algorithm for FWA-DM is given in Algorithm 2.

Algorithm 2 The process of FWA-DM

randomly generate the initial population with NP individuals as POP1
evaluate the fitness values for all individuals
while terminal condition not met **do**
  for $i = 1 \to NP$ **do**
    apply FWA to POP1 and forms POP2
    choose the better ones from POP1 and POP2 and forms POP3
    randomly select $r1 \neq r2 \neq r3 \neq i$
    randomly select j_{rand} from [1, D]
    for $j = 1 \to D$ **do**
      if $rand(0,1) \leq CR$ or $j == j_{rand}$ **then**
        $U_i(j) = X_{r1}(j) + F \times (X_{r2}(j) - X_{r3}(j))$
      else
        $U_i(j) = X_i(j)$ /* U_i forms POP4 */
      end if
    end for
  end for
  for $j = 1 \to D$ **do**
    evaluate the fitness values for U_i
    if U_i is better than X_i **then**
      $X_i = U_i$ /* X_i return to POP1 */
    end if
  end for
end while

III. Experiments

A. Benchmark Functions

FWA-DM is used to find the global optimum values of 30 benchmark functions from the CEC'14 competition [16]. The details of the functions are given as follows.

Unimodal Functions:
1 Rotated High Conditioned Elliptic Function
2 Rotated Bent Cigar Function
3 Rotated Discus Function
Simple Multimodal Functions:
4 Shifted and Rotated Rosenbrocks Function
5 Shifted and Rotated Ackley's Function
6 Shifted and Rotated Weierstrass Function
7 Shifted and Rotated Griewank's Function
8 Shifted Rastrigin's Function

9 Shifted and Rotated Rastrigin's Function
10 Shifted Schwefel's Function
11 Shifted and Rotated Schwefel's Function
12 Shifted and Rotated Katsuura Function
13 Shifted and Rotated HappyCat Function
14 Shifted and Rotated HGBat Function
15 Shifted and Rotated Expanded Griewank's plus Rosenbrock's Functions
16 Shifted and Rotated Expanded Scaffer's F6 Function

Hybrid Functions:

17 Hybrid Function 1 (N=3)
18 Hybrid Function 2 (N=3)
19 Hybrid Function 3 (N=4)
20 Hybrid Function 4 (N=4)
21 Hybrid Function 5 (N=5)
22 Hybrid Function 6 (N=5)

Composition Functions:

23 Composition Function 1 (N=5)
24 Composition Function 2 (N=3)
25 Composition Function 3 (N=3)
26 Composition Function 4 (N=5)
27 Composition Function 5 (N=5)
28 Composition Function 6 (N=5)
29 Composition Function 7 (N=3)
30 Composition Function 8 (N=3)

B. Parameter Settings

Each experiment runs 51 times and during each run, the function is evaluated for 10000*D times. The actual used parameter values are set as follows. Parameters A_{init} and A_{final} are set as 0.2 and 0.001, respectively, as in [22]. In addition, parameters F and CR are set as 0.5 and 0.9, as in [7]. The population size is set as 5*D. It is obvious that the parameters are simple and easy to set.

C. Experimental Results

The experimental platform is Visual Studio 2012 and the program is running on a Windows 8 operating system. The experimental results for 10D, 30D, 50D and 100D are listed below.

The computational complexity of FWA-DM is given in Table V.

D. Discussion

The functions contain unimodal, simple multimodal, hybrid and composition functions. Hence, the experimental results do not skew to any kind of functions and are objective.

When calculating the functions with 10 dimension, four global optimum values were found when FWA-DM performed the best. Moreover, one of the mean objective function values was lower than 1e-8. The best experimental result came from calculating the F3, which was a unimodal function. For F23, FWA-DM fell into local optimum values, as the standard deviation was low.

When the dimension was 30, FWA-DM found the global optimum values on four functions when it performed the best. In addition, when applying FWA-DM to calculate F2, F3 and

TABLE I
RESULTS FOR 10D

Func.	Best	Worst	Median	Mean	Std.
1	2.48E-02	1.12E+05	2.33E+02	5.01E+03	1.67E+04
2	0.00E+00	6.84E-03	7.14E-22	1.34E-04	9.49E-04
3	0.00E+00	5.94E-08	5.38E-18	1.88E-09	9.02E-09
4	0.00E+00	4.73E+00	8.41E-02	1.41E+00	1.60E+00
5	2.00E+01	2.01E+01	2.00E+01	2.00E+01	4.17E-02
6	3.40E-03	2.48E+00	5.74E-01	7.06E-01	6.40E-01
7	1.72E-02	2.24E-01	8.61E-02	9.48E-02	4.92E-02
8	0.00E+00	3.98E+00	1.78E-15	2.54E-01	8.09E-01
9	1.99E+00	1.69E+01	5.97E+00	6.01E+00	2.45E+00
10	9.09E-13	6.95E+00	3.75E-01	1.59E+00	2.08E+00
11	3.98E+01	8.13E+02	3.60E+02	3.72E+02	1.53E+02
12	3.34E-06	2.84E-01	2.82E-02	4.25E-02	4.78E-02
13	3.39E-02	2.99E-01	1.04E-01	1.21E-01	7.18E-02
14	3.87E-02	5.52E-01	1.86E-01	2.14E-01	1.20E-01
15	3.21E-01	1.47E+00	7.43E-01	7.75E-01	2.63E-01
16	7.96E-01	2.82E+00	1.73E+00	1.76E+00	4.68E-01
17	2.63E+00	7.39E+02	2.26E+02	2.55E+02	1.77E+02
18	1.34E+00	8.68E+01	2.01E+01	2.52E+01	1.83E+01
19	3.66E-02	3.00E+00	1.11E+00	1.30E+00	7.75E-01
20	8.81E-01	6.52E+01	9.69E+00	1.34E+01	1.16E+01
21	4.01E-01	4.26E+02	5.90E+01	9.46E+01	9.80E+01
22	1.20E-01	1.57E+02	2.05E+01	3.41E+01	4.40E+01
23	3.29E+02	3.29E+02	3.29E+02	3.29E+02	5.59E-08
24	1.08E+02	2.08E+02	1.16E+02	1.27E+02	2.90E+01
25	1.20E+02	2.01E+02	2.00E+02	1.79E+02	2.76E+01
26	1.00E+02	1.00E+02	1.00E+02	1.00E+02	7.47E-02
27	1.91E+00	4.19E+02	3.49E+02	3.21E+02	1.21E+02
28	3.06E+02	4.13E+02	3.07E+02	3.47E+02	4.76E+01
29	2.02E+02	3.39E+02	2.07E+02	2.12E+02	2.08E+01
30	2.24E+02	7.08E+02	3.67E+02	3.94E+02	1.18E+02

TABLE II
RESULTS FOR 30D

Func.	Best	Worst	Median	Mean	Std.
1	3.70E+04	9.94E+05	2.24E+05	2.76E+05	1.82E+05
2	4.74E-19	1.13E-15	3.20E-17	1.08E-16	1.87E-16
3	2.35E-17	2.23E-15	2.48E-16	4.42E-16	4.74E-16
4	1.22E-03	7.44E+01	1.58E+01	2.04E+01	1.91E+01
5	2.04E+01	2.06E+01	2.05E+01	2.05E+01	5.36E-02
6	8.12E-02	2.09E+01	1.75E+01	1.29E+01	8.25E+00
7	0.00E+00	3.69E-02	7.40E-03	8.55E-03	9.81E-03
8	0.00E+00	2.69E-12	1.78E-15	1.13E-13	4.51E-13
9	3.32E+01	7.82E+01	5.62E+01	5.66E+01	1.08E+01
10	4.65E+00	1.54E+01	8.09E+00	8.53E+00	2.42E+00
11	2.00E+03	3.03E+03	2.65E+03	2.63E+03	2.48E+02
12	2.08E-01	5.21E-01	3.70E-01	3.71E-01	6.66E-02
13	2.88E-01	4.90E-01	4.00E-01	3.89E-01	5.51E-02
14	1.78E-01	7.40E-01	2.59E-01	2.69E-01	7.76E-02
15	5.64E+00	9.05E+00	7.33E+00	7.37E+00	8.46E-01
16	1.03E+01	1.14E+01	1.10E+01	1.10E+01	2.71E-01
17	1.08E+03	2.43E+04	3.30E+03	6.29E+03	5.95E+03
18	1.17E+01	1.60E+02	6.65E+01	7.67E+01	3.66E+01
19	3.88E+00	1.34E+01	1.03E+01	9.95E+00	1.93E+00
20	1.30E+01	1.40E+02	3.16E+01	4.28E+01	2.61E+01
21	1.09E+02	5.11E+03	4.48E+02	7.29E+02	9.49E+02
22	2.72E+01	3.70E+02	1.53E+02	1.46E+02	8.83E+01
23	3.14E+02	3.14E+02	3.14E+02	3.14E+02	9.28E-14
24	2.22E+02	2.38E+02	2.25E+02	2.26E+02	3.59E+00
25	2.00E+02	2.01E+02	2.01E+02	2.01E+02	1.98E-01
26	1.00E+02	1.01E+02	1.00E+02	1.00E+02	5.35E-02
27	3.13E+02	4.90E+02	4.01E+02	4.01E+02	3.06E+01
28	3.72E+02	4.31E+02	3.90E+02	3.93E+02	1.46E+01
29	2.05E+02	2.17E+02	2.11E+02	2.11E+02	2.90E+00
30	2.35E+02	1.08E+03	3.84E+02	4.51E+02	1.96E+02

TABLE III

RESULTS FOR 50D

Func.	Best	Worst	Median	Mean	Std.
1	2.15E+06	1.29E+07	5.76E+06	6.15E+06	2.39E+06
2	2.23E+00	2.10E+04	2.15E+03	4.83E+03	6.06E+03
3	3.09E+01	1.28E+02	7.61E+01	7.66E+01	2.46E+01
4	3.91E+01	1.00E+02	4.30E+01	5.05E+01	1.88E+01
5	2.06E+01	2.08E+01	2.07E+01	2.07E+01	3.61E-02
6	3.66E+01	4.81E+01	4.38E+01	4.39E+01	2.31E+00
7	7.22E-10	1.48E-02	9.06E-09	2.66E-03	4.25E-03
8	4.70E+00	1.59E+01	9.17E+00	9.17E+00	2.39E+00
9	1.17E+02	1.92E+02	1.47E+02	1.47E+02	1.57E+01
10	2.53E+02	8.90E+02	6.07E+02	6.14E+02	1.42E+02
11	4.27E+03	6.15E+03	5.49E+03	5.43E+03	4.02E+02
12	3.94E-01	7.92E-01	6.04E-01	6.02E-01	8.72E-02
13	3.86E-01	6.21E-01	4.82E-01	4.88E-01	4.63E-02
14	2.04E-01	7.99E-01	3.06E-01	3.30E-01	1.05E-01
15	1.74E+01	2.33E+01	2.06E+01	2.08E+01	1.45E+00
16	1.92E+01	2.06E+01	2.00E+01	2.00E+01	3.21E-01
17	3.14E+04	3.19E+05	9.46E+04	1.08E+05	6.03E+04
18	3.20E+02	1.70E+04	1.95E+03	3.31E+03	3.72E+03
19	1.07E+01	2.85E+01	2.09E+01	2.11E+01	3.17E+00
20	2.39E+02	5.70E+02	3.91E+02	4.00E+02	8.12E+01
21	7.78E+03	8.32E+04	2.09E+04	2.55E+04	1.58E+04
22	2.07E+02	8.29E+02	5.49E+02	5.45E+02	1.25E+02
23	3.37E+02	3.37E+02	3.37E+02	3.37E+02	1.31E-05
24	2.64E+02	2.77E+02	2.65E+02	2.66E+02	3.48E+00
25	2.02E+02	2.11E+02	2.04E+02	2.05E+02	2.26E+00
26	1.00E+02	1.01E+02	1.00E+02	1.00E+02	5.98E-02
27	1.27E+03	1.59E+03	1.46E+03	1.45E+03	7.27E+01
28	3.72E+02	4.92E+02	3.88E+02	4.01E+02	3.06E+01
29	2.18E+02	2.31E+02	2.22E+02	2.23E+02	2.67E+00
30	3.95E+02	1.84E+03	7.71E+02	8.36E+02	3.06E+02

TABLE IV

RESULTS FOR 100D

Func.	Best	Worst	Median	Mean	Std.
1	1.49E+08	3.23E+08	2.22E+08	2.28E+08	4.08E+07
2	1.25E+03	1.03E+05	9.31E+03	1.62E+04	1.81E+04
3	2.03E+04	3.57E+04	2.98E+04	2.95E+04	3.64E+03
4	9.74E+01	4.02E+02	1.52E+02	1.83E+02	1.00E+02
5	2.09E+01	2.11E+01	2.10E+01	2.10E+01	2.44E-02
6	1.09E+02	1.19E+02	1.14E+02	1.14E+02	2.54E+00
7	7.08E-02	2.05E-01	1.23E-01	1.29E-01	3.14E-02
8	9.52E+01	1.20E+02	1.09E+02	1.08E+02	5.40E+00
9	4.53E+02	6.36E+02	5.58E+02	5.52E+02	4.44E+01
10	4.97E+03	6.21E+03	5.66E+03	5.67E+03	3.04E+02
11	1.32E+04	1.58E+04	1.46E+04	1.46E+04	6.30E+02
12	1.08E+00	1.37E+00	1.23E+00	1.21E+00	7.78E-02
13	4.68E-01	6.36E-01	5.62E-01	5.61E-01	3.61E-02
14	1.47E-01	2.28E-01	1.89E-01	1.89E-01	2.06E-02
15	7.47E+01	9.91E+01	8.72E+01	8.74E+01	5.87E+00
16	4.24E+01	4.43E+01	4.35E+01	4.35E+01	3.75E-01
17	1.39E+07	4.17E+07	2.30E+07	2.31E+07	5.63E+06
18	3.93E+02	3.70E+04	2.29E+03	5.68E+03	8.70E+03
19	5.48E+01	6.97E+01	6.40E+01	6.34E+01	2.43E+00
20	4.12E+04	9.49E+04	6.98E+04	6.93E+04	1.10E+04
21	3.40E+06	1.55E+07	9.41E+06	9.57E+06	2.31E+06
22	1.23E+03	1.84E+03	1.54E+03	1.51E+03	1.34E+02
23	3.45E+02	3.47E+02	3.46E+02	3.46E+02	2.18E-01
24	3.58E+02	3.69E+02	3.64E+02	3.63E+02	2.85E+00
25	2.62E+02	3.28E+02	3.10E+02	3.03E+02	1.74E+01
26	1.01E+02	2.25E+02	2.12E+02	1.61E+02	5.88E+01
27	1.32E+03	3.44E+03	3.21E+03	3.14E+03	4.13E+02
28	1.07E+03	2.43E+03	1.54E+03	1.60E+03	3.42E+02
29	2.59E+02	2.85E+02	2.69E+02	2.70E+02	5.23E+00
30	9.18E+02	5.99E+03	1.84E+03	2.23E+03	1.16E+03

TABLE V

COMPUTATIONAL COMPLEXITY OF FWA-DM GIVEN FOR 10, 30, 50 AND 100 DIMENSIONAL F18

	T0	T1	$\hat{T}_2$	($\hat{T}_2$-T1)/T0
D=10	0.077	0.223	1.9142	21.96364
D=30	0.077	0.84	2.582	22.62338
D=50	0.077	1.992	3.7522	22.85974
D=100	0.077	6.926	8.6232	22.04156

F8, the objective function values were lower than 1e-8. For F23, the standard deviation was low, which means FWA-DM achieved local optimum values.

The experimental results on 50D functions shown worse performance than both 10D and 30D. The global optimum was found on F7 alone. In addition, no global optimum was found on 100D functions.

IV. CONCLUSION

In this paper, the performance of FWA-DM is tested on benchmark suite of CEC 2014. The mean objective function values of FWA-DM were lower than 1e-8 on three functions in dimension 30, one function in dimension 10 and one function in dimension 50. In addition, the convergence curves were given at the end of this paper.

FWA-DM is proposed by applied DE mutation operator to FWA. The potential of the parameters tuning is not used. Also, the experimental results can be further improved by introducing strategies. At last, it would be interesting to compare the results of the proposed algorithm with single FWA and DE in future.

REFERENCES

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

[2] A. Janecek and Y. Tan, "Iterative improvement of the multiplicative update nmf algorithm using nature-inspired optimization," in *Natural Computation (ICNC), 2011 Seventh International Conference on*, vol. 3, pp. 1668–1672, IEEE, 2011.

[3] A. Janecek and Y. Tan, "Swarm intelligence for non-negative matrix factorization," *International Journal of Swarm Intelligence Research (IJSIR)*, vol. 2, no. 4, pp. 12–34, 2011.

[4] A. Janecek and Y. Tan, "Using population based algorithms for initializing nonnegative matrix factorization," in *Advances in Swarm Intelligence*, pp. 307–316, Springer, 2011.

[5] 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.

[6] 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*, pp. 439–450, Springer, 2013.

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

[8] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 6, pp. 646–657, 2006.

[9] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 4–31, 2011.

[10] R. Mallipeddi, P. N. Suganthan, Q.-K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, no. 2, pp. 1679–1696, 2011.

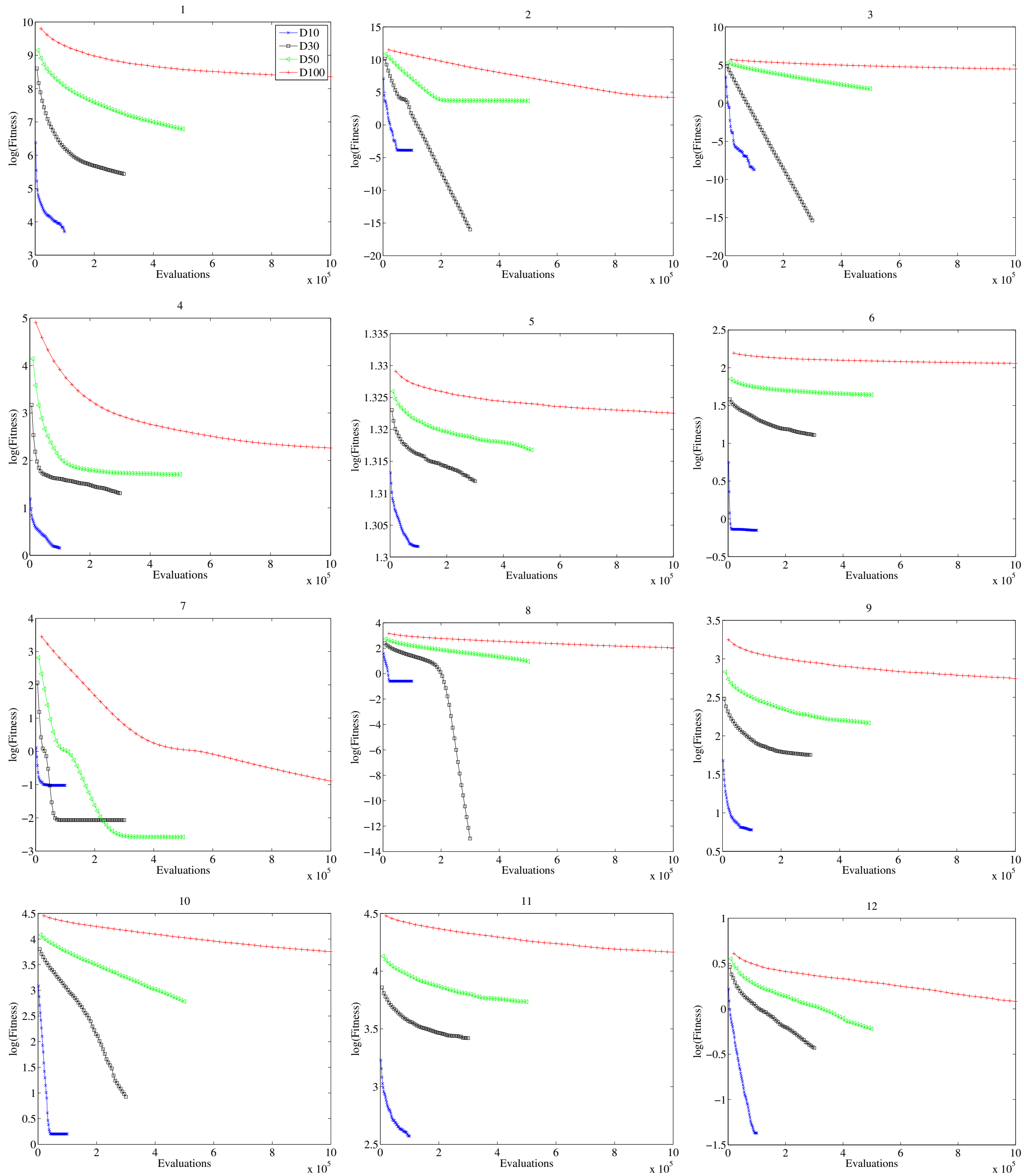

Fig. 2. Convergence curves for FWA-DM on function 1 to 12 in Dimension 10, 30, 50 and 100.

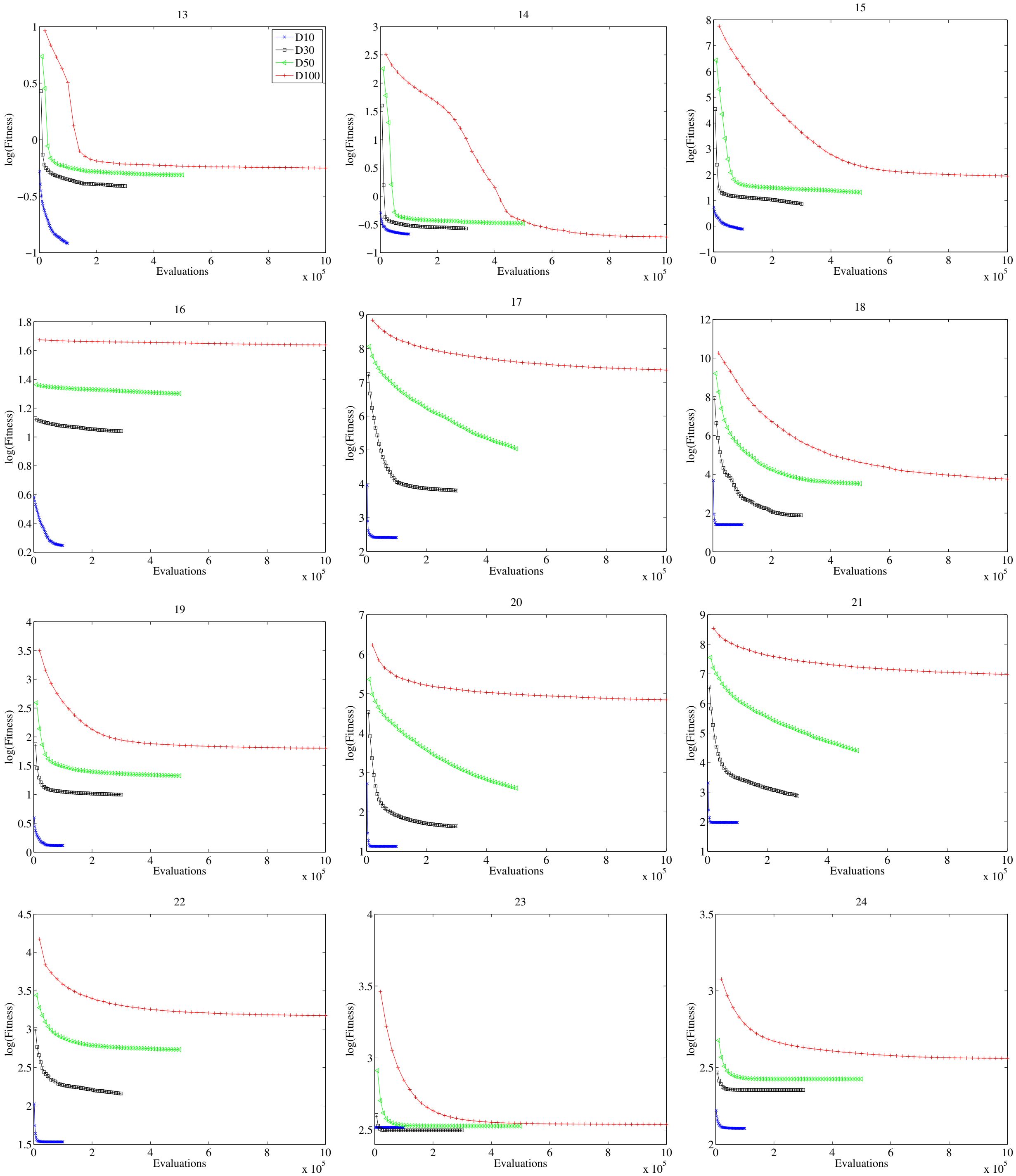

Fig. 3. Convergence curves for FWA-DM on function 13 to 24 in Dimension 10, 30, 50 and 100.

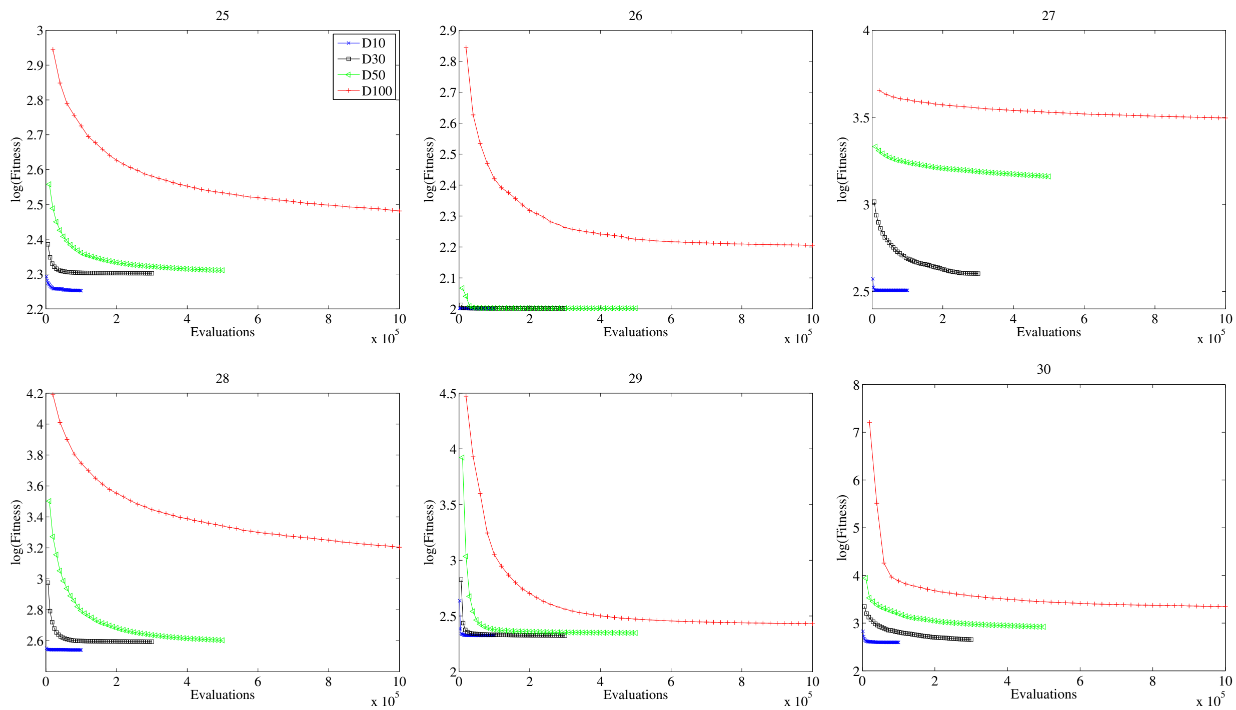

Fig. 4. Convergence curves for FWA-DM on function 25 to 30 in Dimension 10, 30, 50 and 100.

[11] J. Dominguez and G. T. Pulido, "A comparison on the search of particle swarm optimization and differential evolution on multi-objective optimization," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pp. 1978–1985, IEEE, 2011.

[12] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 55–66, 2011.

[13] S. Biswas, S. Kundu, S. Das, and A. V. Vasilakos, "Teaching and learning best differential evoltuion with self adaptation for real parameter optimization," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 1115–1122, IEEE, 2013.

[14] S. Biswas, S. Kundu, and S. Das, "An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution," *IEEE Transactions on Cybernetics*, 2014.

[15] S. Biswas, S. Kundu, D. Bose, S. Das, and P. Suganthan, "Synchronizing differential evolution with a modified affinity-based mutation framework," in *Differential Evolution (SDE), 2013 IEEE Symposium on*, pp. 61–68, IEEE, 2013.

[16] J. Liang, B. Qu, and P. Suganthan, "Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization," 2013.

[17] M. Fatih Tasgetiren, Q.-K. Pan, P. N. Suganthan, and O. Buyukdagli, "A variable iterated greedy algorithm with differential evolution for the no-idle permutation flowshop scheduling problem," *Computers & Operations Research*, vol. 40, pp. 1729–1743, 2013.

[18] A. Mandal, H. Zafar, S. Das, and A. V. Vasilakos, "A modified differential evolution algorithm for shaped beam linear array antenna design," *Progress In Electromagnetics Research*, vol. 125, pp. 439–457, 2012.

[19] F. Neri and E. Mininno, "Memetic compact differential evolution for cartesian robot control," *Computational Intelligence Magazine, IEEE*, vol. 5, no. 2, pp. 54–65, 2010.

[20] K. Rzadca and F. Seredynski, "Heterogeneous multiprocessor scheduling with differential evolution," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 3, pp. 2840–2847, IEEE, 2005.

[21] M. Secmen, M. F. Tasgetiren, and K. Karabulut, "Null control in linear antenna arrays with ensemble differential evolution," in *Differential Evolution (SDE), 2013 IEEE Symposium on*, pp. 92–98, IEEE, 2013.

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