

Hybrid Fireworks Algorithm with Dynamic Coefficients and Improved Differential Evolution[☆]

Lixian Li^{1, 2} and Jaewan Lee^{2*}

ABSTRACT

Fireworks Algorithm (FWA) is a new heuristic swarm intelligent algorithm inspired by the natural phenomenon of the fireworks explosion. Though it is an effective algorithm for solving optimization problems, FWA has a slow convergence rate and less information sharing between individuals. In this paper, we improve the FWA. Firstly, explosion operator and explosion amplitude are analyzed in detail. The coefficient of explosion amplitude and explosion operator change dynamically with iteration to balance the exploitation and exploration. The convergence performance of FWA is improved. Secondly, differential evolution and commensal learning (CDE) significantly increase the information sharing between individuals, and the diversity of fireworks is enhanced. Comprehensive experiment and comparison with CDE, FWA, and VACUFWA for the 13 benchmark functions show that the improved algorithm was highly competitive.

✉ keyword : Fireworks Algorithm; Dynamic Coefficient; Differential Evolution; Commensal Learning

1. Introduction

In the past twenty years, many stochastic swarm intelligence (SI) has been proposed. The field of SI has emerged many optimization algorithms [1-4]. Fireworks algorithm (FWA) is a novel evolutionary algorithm. In 2010, inspired by the natural phenomenon that the fireworks explode in the night sky, Tan [5] proposed the fireworks algorithm. As a high-efficiency SI algorithm, FWA has attracted much research interest and has been applied in engineering and science [6-10].

In FWA, fireworks are launched into potential search space and a shower of sparks fills the surrounding local space [11-13]. However, it does not take advantage of more information about fireworks and sparks in the whole population. The critical feature of FWA is the random detonation process. "good"

fireworks with small fitness value can produce a large number of explosive sparks in a smaller explosion amplitude. "bad" fireworks with great fitness can produce a small number of explosive sparks with a more extensive scope [14]. Therefore, the performance of FWA can be improved by dynamically changing the number of sparks and amplitude.

In the paper, the main contributions are as follows: First, the dynamic coefficient formula is created. Second DE is improved with Commensal learning (CDE). Then, we propose hybrid fireworks algorithm with dynamic coefficients and improved differential evolution (namely HDEFWA). The dynamic coefficient balances the global and local search. CDE avoids premature convergence of population, and the diversity of the population is improved. Experiments on 13 benchmark functions show that the performance of FWA can be significantly enhanced by using well-informed fireworks and sparks.

2. Related Research

2.1 Bare-Bones FWA

FWA is a new meta-heuristic algorithm based on swarm intelligence to solve complex optimization problems [5]. The

¹ School of Information Science and Technology, Jiujiang University, Jiujiang, 332005, P.R.China

² Department of Information and Communication Engineering, Kunsan National University, Kunsan, 54150, Korea

* Corresponding author: Jaewan Lee(jwlee@kunsan.ac.kr)

[Received 10 November 2020, Reviewed 1 December 2020(R2 4 February 2021), Accepted 16 February 2021]

☆ This research is supported by Institute of Information and Telecommunication Technology of Kunsan National University, South Korea

bare-bones FWA is described as Algorithm 1.

Algorithm 1 The bare-bones fireworks

```

1  Set the algorithm parameters  $p$ ,  $lowerInit$ ,  $upperInit$ ,
    $AC$ , and  $CE$ ;
2  Randomly initialize a swarm of  $p$  fireworks, and
   Calculate the fitness values;
3  While termination conditions  $\neq$  True do
4    For each firework  $y_i \in p$  do
5      Generate the explosion sparks  $N_i$  by
       firework as Eq. (1) ;
6      Generate the explosion amplitude  $A_i$  by
       firework as Eq. (2) ;
7      For  $j=1:p$  do
8        select a dimension of  $y_i$  Randomly,
        Carry out Gaussian mutation and
        Generate Gaussian sparks;
9      Calculate the location and fitness value
       of each spark and firework;
10     Select the spark with best fitness value
        and keep it for next generation;
11     Other  $p-1$  locations selected based on
        distance probabilities
12  End
    
```

There are two important factors in the process of the explosion [15, 16]. The number of sparks generated by each firework is defined as follows:

$$N_i = EC \cdot \frac{y_{\max} - f(x_i) + \delta}{\sum_{j=1}^p (y_{\max} - f(x_j)) + \delta} \quad (1)$$

The amplitude of a firework explosion is defined as follows:

$$A_i = AC \cdot \frac{f(x_i) - y_{\min} + \delta}{\sum_{j=1}^p (f(x_j) - y_{\min}) + \delta} \quad (2)$$

where AC and EC are parameters that are used to control the rang of explosion amplitude and sparks generated by the fireworks, $f(x_i)$ denotes a function of x_i , y_{\min} is the minimum fitness values, y_{\max} is the maximum fitness values. δ is the smallest amount for the machine which is utilized to avoid that the denominator is zero.

2.2 Bare-bones Differential Evolution

Differential Evolution (DE) is a heuristic search and a simple yet effective algorithm for global optimization [17]. It maintains a population with P individuals $Y_{1,N}, Y_{2,N}, \dots, Y_{P,N}$, every individual represents one candidate solutions. N represents the generation index and P represents the population [17]. The process of DE is as follows.

2.2.1 Mutation

DE selects an initial population $Y_{1,0}, Y_{2,0}, \dots, Y_{P,0}$ randomly in the search space. Then, in each generation, the optimal solution is searched by Mutation, crossover, and selection. In this paper, only the minimization of the objective function is considered without loss of consistency.

① DE/best/1

$$T_{j,N} = Y_{best,N} + C \cdot (Y_{j1,N} - Y_{j2,N}) \quad (3)$$

② DE/rand/1

$$T_{j,N} = Y_{j1,N} + C \cdot (Y_{j2,N} - Y_{j3,N}) \quad (4)$$

③ DE/rand/2

$$T_{j,N} = Y_{j1,N} + C \cdot (Y_{j2,N} - Y_{j3,N}) + C \cdot (Y_{j4,N} - Y_{j5,N}) \quad (5)$$

④ DE/current-to-best/1

$$T_{i,N} = Y_{j,N} + C \cdot (Y_{best,N} - Y_{j,N}) + C \cdot (Y_{j1,N} - Y_{j2,N}) \quad (6)$$

⑤ DE/current-to-rand/1

$$C_{j,N} = Y_{j,N} + rand \cdot (Y_{j1,N} - Y_{j,N}) + C \cdot (Y_{j2,N} - Y_{j3,N}) \quad (7)$$

⑥ DE/best/2

$$T_{j,N} = Y_{best,N} + C \cdot (Y_{j1,N} - Y_{j2,N}) + C \cdot (Y_{j3,N} - Y_{j4,N}) \quad (8)$$

where the indices $j1, j2, j3, j4$, and $j5$ are used to distinguish form j , which are mutually distinct from $\{1, 2, \dots, P\}$. C is the mutation factor which is a constant from $[0, 2]$.

$Y_{best,N}$ is the optimal individual which is in the population p at the N th generation. whereas, instead of a mutant vector $T_{i,N}$, the “DE/current-to-rand/1” directly produces a trial vector $C_{i,N}$ with no need for crossover operation.

2.2.2 Crossover

Binomial crossover operator incorporates successful solutions from the target vector $Y_{i,N}$ with mutant vector $T_{i,N}$ and generates a trial vector $C_{i,N}$.

where $j = 1, 2, \dots, D$, D is the parameter which controls the dimension of the mutant vector, j_{rand} denotes a random integer in $\{1, 2, \dots, D\}$, $rand_j$ denotes a standard normal distribution number which is in the j th dimension, and CR represents the crossover rate, which Values range from 0 to 1.

$$C_{i,j,N} = \begin{cases} T_{i,j,N}, & \text{if } rand_j \leq CR \text{ or } j = j_{rand} \\ Y_{i,j,N}, & \text{otherwise} \end{cases} \quad (9)$$

2.2.3 Selection

One-to-one compare is adopted by the selection operator. $Y_{i,N}$ and $C_{i,N}$ denote the target vector and the trial vector respectively, $Y_{i,N}$ is compared with $C_{i,N}$, the one with the better fitness value is admitted to the next generation [18].

$$Y_{i,N+1} = \begin{cases} C_{i,N}, & \text{if } f(Z_{i,G}) < f(X_{i,G}) \\ Y_{i,N}, & \text{otherwise} \end{cases} \quad (10)$$

3. Hybrid Fireworks Algorithm

In this section, we use commensal learning in differential evolution (CDE), and hybridize CDE in FWA with a dynamic coefficient method. HDEFWA enhances information sharing among individuals and balances the exploration and exploitation ability of FWA.

3.1 Commensal Learning

As known, firstly, the performance of DE is sensitive to parameter settings. Secondly, mutation, crossover, and selection of DE depend on a population to some extent. In

the search process, the parameter settings and mutation strategies are automatically changed in the existing methods [19]. Therefore, in this section, two-sided Gaussian distributions with different mean values are constructed respectively to update the scale factor C and the crossover probability CR . Parameter setting of “two-sided Gaussian distribution” [17] is used to update C and CR dynamically. Parameters are setting as Eq. (11-12).

$$\text{Lower} = \begin{cases} C_{lower} = N(0.5, 0.1) \\ CR_{lower} = N(0.1, 0.1) \end{cases} \quad (11)$$

$$\text{Upper} = \begin{cases} C_{upper} = N(0.8, 0.1) \\ CR_{upper} = N(0.9, 0.1) \end{cases} \quad (12)$$

where *Upper* indicates upper value of the parameter settings and *Lower* indicates lower value of the parameter settings. N indicates the Gaussian distribution.

Three mutation strategies with different characteristics are elaborately selected.

- ① DE/rand/1.
- ② DE/best/1.
- ③ DE/current-to-rand/1.

We combine the parameter setting of “two-sided Gaussian distribution” with the above three strategies, and six experimental vector schemes are generated, as follows:

- Ⓐ DE/best/1, Upper;
- Ⓑ DE/best/1, Lower;
- Ⓒ DE/rand/1, Upper;
- Ⓓ DE/rand/1, Lower;
- Ⓔ DE/target-to-rand/1, Upper;
- Ⓕ DE/target-to-rand/1, Lower.

where “DE/target-to-rand/1” is the modification of “DE/current-to-rand/1”. Parameter setting of “two-sided Gaussian distribution” combines “DE/target-to-rand/1” strategy and conducts binomial crossover. as shown Eq. (13).

$$T_{i,G} = Y_{i,G} + rand \cdot (Y_{i1,G} - Y_{i,G}) + C \cdot (Y_{i2,G} - Y_{i3,G}) \quad (13)$$

3.2 Dynamic Coefficient

In FWA, two essential factors affect the performance of fireworks, which are explosion amplitude and explosion sparks. Yet, the coefficient of the explosion sparks and explosion amplitude is fixed in the bare-bones FWA. In the real-world, dynamic adjustment of coefficient with the iteration may correspond to an actual process of a fireworks explosion. The large explosion amplitude is conducive to explore new search space in the initial phase of the fireworks explosion. While the smaller the amplitude and the more sparks of the blast, the more brilliant the fireworks, which is conducive to the local search in the middle and late stages of the fireworks explosion. We propose the approach on controlling the explosion sparks coefficient and amplitude coefficient as shown in Eq. (14) and Eq. (15), which are used to calculate the explosion sparks and explosion amplitude of each firework in Eq. (1) and Eq. (2) respectively.

$$EC(t) = 40 * \left(e^{0.69 * \left(\frac{t - \max}{\max} \right)} \right) \quad (14)$$

$$AC(t) = \frac{40}{1 + e^{(0.015 * (t - \max))}} \quad (15)$$

where \max represents the maximum of evaluation times, and t denotes the current evaluation times.

3.3 The Framework of HDEFWA

In the bare-bones FWA, The global and local searches are performed by controlling amplitude and sparks when fireworks explode. However, this mechanism is not flexible enough in terms of the diversity and searchability of the population. First, to better reflect the diversity of fireworks explosions, the amplitude and the sparks of the explosions should change with the number of evaluations. Second, it should share more information about other individuals in the population. As far as swarm intelligence is concerned, individuals do not have adequate access to information about the population.

Algorithm 2. The HDEFWA

```

1  Set the algorithm parameters  $p$ ,  $lowerInit$ ,  $upperInit$ ,
    $AC$ , and  $CE$ ;
2  Randomly initialize a swarm of  $p$  fireworks, and
   Calculate the fitness values;
3  While termination conditions != True do
4    For each firework  $y_i \in p$  do
5      Generate the explosion sparks  $N_i$  by
       firework according to Eq. (14) and Eq. (1);
6      Generate the explosion amplitude  $A_i$  by
       firework according to Eq. (15) and Eq. (2);
7      For  $j=1:p$  do
8        select a dimension of  $y_i$  Randomly,
       Carry out Gaussian mutation and
       Generate Gaussian sparks;
9      Calculate the location and fitness value of
       each spark and firework;
10     Select the spark that has the most fitness
        value and keep it as a candidate for the
        next generation
11     other p-1 locations selected as candidates for
        the next generation based on distance
        probabilities
12     Using CDE to optimize the candidate set ,
        evaluate each location and its fitness value,
        and keep these as the next generation
13  End

```

Because of the above analysis, CDE and dynamic coefficients are used to improve FWA. The coefficient of the explosion amplitude and explosion spark is adjusted dynamically with the evaluation times. In each generation, CDE and dynamic coefficients are used to search for new feasible solutions, which avoid premature of the FWA, and the diversity of the FWA is further improved. The framework of the HDEFWA is as algorithm 2. The algorithm set parameters and initialize fireworks (line 1 to line 2). Next, set termination conditions and begin the main loop (line 3). First, each firework explodes according to its amplitude and number of sparks, the explosion sparks are generated (line 4 to line 6). Second, Gaussian sparks are generated (line 7 to line 8). Subsequently, obtain the position and fitness values of the two types of sparks (line 9). Select the best spark from both explosions to represent the population of the fireworks for the next generation candidate (line 10). Randomly select p-1 sparks from both explosions sparks (line 11). The selected P sparks are improved by mutation, crossover, and selection of the CDE, then generate the next generation (line 12). Check the termination condition (line 13).

(Table 1) Comparison of CDE, FWA, VACUFWA, and HDEFWA for 13 functions (D=30, EavTimes=10000)

Fun	CDE	FWA	VACUFWA	HDEFWA
	Mean(standard deviation)	Mean(standard deviation)	Mean(standard deviation)	Mean(standard deviation)
f1	4.05E+01(1.02E+01)-	7.70E-32(4.12E-31)-	1.08E-38(5.54E-38)-	3.41E-42(1.77E-41)
f2	2.62E+00(3.49E-01)-	2.73E-17(9.50E-17)-	2.83E-21(1.11E-20)+	7.98E-25(4.04E-24)
f3	1.27E+04(3.55E+03)-	6.35E-15(3.41E-14)-	2.97E-25(1.60E-24)-	5.36E-37(1.83E-36)
f4	2.56E+01(4.23E+00)-	2.75E-12(1.47E-11)-	5.26E-17(1.38E-16)-	3.40E-18(9.91E-18)
f5	3.05E+03(8.54E+02)-	2.83E+01(3.14E+00)-	2.22E+01(1.09E+01)+	2.38E+01(9.46E+00)
f6	4.49E+01(1.18E+01)-	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
f7	1.50E-01(4.37E-02)-	3.82E-03(4.12E-03)-	4.35E-03(4.02E-03)-	3.02E-03(3.15E-03)
f8	4.57E+03(3.70E+02)-	6.40E+03(6.91E+02)-	5.18E+03(6.86E+02)-	4.40E+03(8.12E+02)
f9	1.14E+02(9.54E+00)-	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
f10	3.18E+00(1.72E-01)-	3.55E-16(1.07E-15)-	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
f11	1.35E+00(9.52E-02)-	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
f12	2.53E+00(6.97E-01)-	6.63E-02(6.93E-02)-	3.60E-02(1.68E-02)-	2.60E-02(3.59E-02)
f13	7.52E+00(2.76E+00)-	6.79E-01(5.28E-01)-	3.97E-01(1.38E-01)-	3.54E-01(1.28E-01)
-/+≈	13/0/0	10/0/3	8/1/4	--

(Table 2) Comparison of CDE, FWA, VACUFWA, and HDEFWA for 13 functions (D=100, EavTimes=10000)

Fun	CDE	FWA	VACUFWA	HDEFWA
	Mean(standard deviation)	Mean(standard deviation)	Mean(standard deviation)	Mean(standard deviation)
f1	1.14E+04(1.78E+03)-	2.16E-32(8.54E-32)-	4.16E-28(2.00E-27)-	1.39E-37(7.39E-37)
f2	1.07E+02(7.95E+00)-	1.78E-17(9.08E-17)-	6.82E-15(3.67E-14)-	4.16E-21(1.19E-20)
f3	1.88E+05(3.60E+04)-	2.27E-15(1.22E-14)-	9.47E-13(5.10E-12)-	5.28E-23(2.66E-22)
f4	5.94E+01(4.76E+00)-	3.14E-10(18.82-10)-	2.96E-09(1.49E-08)-	2.82E-13(1.43E-12)
f5	6.58E+06(2.20E+06)-	9.88E+01(2.47E+01)-	9.28E+01(2.16E+01)-	9.09E+01(1.96E+01)
f6	1.13E+04(1.57E+03)-	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
f7	1.01E+01(4.37E_00)-	3.30E-03(3.57E-03)+	7.79E-03(6.84E-03)-	4.31E-03(3.93E-03)
f8	2.78E+04(4.67E+02)-	3.01E+04(1.78E+03)-	2.43E+04(1.73E+03)-	2.15E+04(1.59E+03)
f9	8.18E+02(2.70E+01)-	0.00E+00(0.00E+00)≈	2.96E-16(1.59E-15)-	0.00E+00(0.00E+00)
f10	1.21E+01(5.89E-01)-	1.54E-15(3.00E-15)-	1.09E-14(5.47E-14)-	1.18E-16(6.38E-16)
f11	1.05E+02(2.01E+01)-	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
f12	1.35E+06(8.61E+05)-	1.13E-01(1.97E-01)-	5.01E-02(2.22E-02)-	4.32E-02(1.54E-02)
f13	1.01E+07(3.54E+06)-	2.24E-00(1.66E-00)-	1.73E+00(9.73E-01)-	1.66E+00(6.27E-01)
-/+≈	13/0/0	9/1/3	11/0/2	--

4. Experimental Simulation and Analysis

4.1 Experimental Parameters Setting and Test Function

13 benchmark functions are selected to test the

performance of the HDEFWA. $f1-f5$ are unimodal functions and $f6-f13$ are basic multimodal functions in the benchmark functions, which are described in detail in reference [20].

In the experimental simulation, the dimension size of the population is 30. The maximum number of evaluations for the function is set to 10000, i.e., the termination criterion of algorithms. The HDEFWA and relevant algorithms run

independently 30 times on each test function. The experimental hardware environment contains 3.00GHz Intel-Core i7-9700 Processor and 16GB RAM. The software environment includes Windows 10 operating system and Matlab R2018a.

So far, researchers have not agreed on the size of the population. The population P is determined by each generation of fireworks in FWA. The population P of CDE is the same as HDEFWA, which are all set to 5. Each algorithm is run 30 times on the benchmark functions, standard deviation and the mean of the algorithm are recorded in the experiment. To evaluate objectively and impartially, the Wilcoxon rank-sum test and the Friedman test [18] are used to analyze the experimental results.

4.2 Experimental Results and Comparison

To illustrate the performance of the HDEFWA, which is compared with CDE, FWA, and VACUFWA [21], Friedman test result of CDE, FWA, VACUFWA, and HDEFWA is 3.85, 2.77, 2.04, and 1.35. The HDEFWA gets the best result. Wilcoxon rank-sum test is shown in Table 1. The symbol“ \approx ”, “-”, and “+” indicates that the other three algorithms are respectively similar to, inferior to, and superior to HDEFWA in performance. As you can observe from the last rows of Table 1 and Table 2, in the four algorithms for the 13 test functions, HDEFWA is the best one in general. HDEFWA is superior to CDE, FWA, and HDEFWA, respectively, for almost all test functions, but only inferior to VACUFWA on f5 when the dimension is 30 and inferior to FWA when the dimension is 100.

To observe the convergence speed of each algorithm more intuitively, Figure 1 shows the convergence characteristics of 9 representative functions. The convergence curves of the CDE, FWA, VACUFWA, and VACUFWA are painted by four lines with different styles. To make the images more apparent, the ordinate represents the mean of the objective function of 30 experiments in the form of $\log_{10}(f)$. The abscissa in the figure represents the evaluation times of the function.

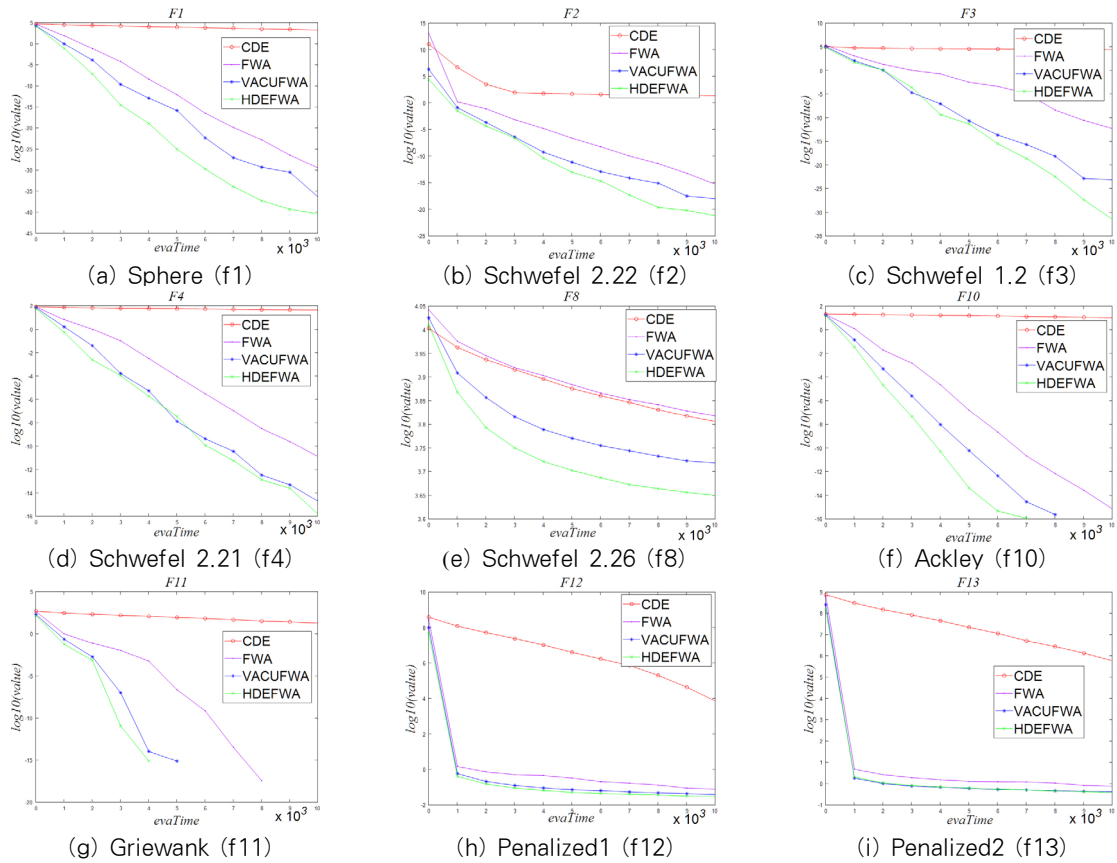
As shown in Figure 1, by comparing the convergence curves of CDE, FWA, VACUFWA, and HDEFWA, the following results can be obtained: Firstly, the convergence

property of FWA is significantly better than that of CDE, and the convergence property of HDEFWA is best in the four algorithms. Secondly, in the early phase of evolution, for 9 functions, the convergence rate of the HDEFWA is similar to VACUFWA and FWA; in the middle and late stage of evolution, for the function f1, f2, f3, f8, f10, and f11, dynamic coefficient of HDEFWA cause that the amplitude of explosion to decrease to generate more sparks in a small explosion range. The convergence speed of HDEFWA is faster than the FWA and VACUFWA. Thirdly, for the function f4, f12, and f13, the convergence rate of HDEFWA is similar to that of VACUFWA and FWA. HDEFWA has a better diversity property.

4.3 The Effectiveness of the Two Components in HDEFWA

As above-mentioned, two new methods, i.e., dynamic coefficients and differential evolution with commensal learning are used to improve the HDEFWA. To investigate the impact of these two components, we conducted a set of experiments on 13 benchmark functions. Therefore, we verify the performance of the four algorithms which are FWA, FWA_CDE, FWA_DC, and HDEFWA. FWA_CDE is bare-bones FWA combining with differential evolution and commensal learning. FWA_DC is bare-bones FWA with the dynamic coefficient. The parameters setting of FWA_DE and FWA_DC are the same as that of FWA and HDEFWA.

As shown in Table 3, the HDEFWA gets the best results out of the four algorithms for the 13 test functions. The performance of HDEFWA is improved by CDE and DC, among the 13 test functions, 10 test functions have significantly better solution quality than the bare-bones FWA. The statistical results show that HDEFWA surpasses FWA_DE on 9 test functions, and surpasses FWA_DC on 6 test functions. Combing with differential evolution and commensal learning, individuals can make more effective use of beneficial information in the population, thus HDEFWA produces better quality solutions. Dynamic coefficient enables firework to balance the exploration and exploitation ability of HDEFWA with the number of evaluation.



(Figure 1) The convergence curves of the standard CDE, FWA, VACUFWA, and HDEFWA on 12 test functions

(Table 3) Experimental results of FWA, FWA_CDE, FWA_DC, and HDEFWA for 13 functions (D=30, EvalTimes=10000)

Fun	FWA Mean(standard deviation)	FWA_CDE Mean(standard deviation)	FWA_DC Mean(standard deviation)	HDEFWA Mean(standard deviation)
f1	7.70E-32(4.12E-31)-	7.03E-30(3.78E-29)-	4.27E-39(2.30E-38)-	3.41E-42(1.77E-41)
f2	2.73E-17(9.50E-17)-	2.00E-18(5.86E-18)-	2.65E-25(8.04E-25)+	7.98E-25(4.04E-24)
f3	6.35E-15(3.41E-14)-	3.25E-19(1.75E-18)-	2.36E-31(1.17E-30)-	5.36E-37(1.83E-36)
f4	2.75E-12(1.47E-11)-	2.69E-12(1.41E-11)-	2.68E-18(5.64E-18)+	3.40E-18(9.91E-18)
f5	2.83E+01(3.14E+00)-	1.99E+01(1.13E+01)+	2.11E+01(1.13E+01)+	2.38E+01(9.46E+00)
f6	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
f7	3.82E-03(4.12E-03)-	5.17E-03(4.37E-03)-	3.35E-03(4.15E-03)-	3.02E-03(3.15E-03)
f8	6.40E+03(6.91E+02)-	4.72E+03(7.43E+02)-	4.82E+03(7.22E+02)-	4.40E+03(8.12E+02)
f9	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
f10	3.55E-16(1.07E-15)-	2.37E-16(8.86E-16)-	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
f11	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)≈	0.00E+00(0.00E+00)
f12	6.63E-02(6.93E-02)-	32.88E-02(2.04E-02)-	4.10E-02(1.97E-02)-	2.60E-02(3.59E-02)
f13	6.79E-01(5.28E-01)-	3.92E-01(2.11E-01)-	4.44E-01(2.06E-01)-	3.54E-01(1.28E-01)
-/+≈	10/0/3	9/1/3	6/3/4	--

4.4 Complexity Analysis

The runtime complexity of a classic FWA is $O(E \cdot P_s \cdot (S \cdot D + SN))$. Where E is the maximum number of evaluations for the function, and P_s is the number of fireworks in each iteration, S is the sparks generated by a firework in the explosion, and D is the dimension of the fireworks population, and SN is the total number of sparks generated by currently fireworks. The runtime complexity of classic DE is $O(G_{max} \cdot NP \cdot D)$ [22], where G_{max} is the maximum number of generations, and P_s is the number of population, and D is the dimension of the problem. In HDEFWA, G_{max} is set to the same to E , so the total complexity of HDEFW is $O(E \cdot (P_s \cdot (S \cdot D + SN) + NP \cdot DP))$, where the number of population NP is set to the same value as P_s , and DP is also the dimension of the fireworks population. Hence the runtime complexity of HDEFWA does not increase.

5. Conclusions

HDEFWA has the smallest mean among the four algorithms and a greater rate of convergence than FWA and VACUFWA. The possible reason is that CDE and dynamic coefficients improve FWA. This indicates that the contribution of CDE operators enhances the diversity of FWA search and increases the information sharing between fireworks and sparks. Furthermore, the dynamic coefficient balances the local exploitation and global exploration. Compared with CDE, FWA, and VACUFWA based on 13 benchmark test functions, the superior performance of HDEFWA is verified. Our ongoing work is implementing the experiment of FWA on the spark platform. In the future, we will also seek the potential hybridization strategies of FWA and exhibit more different behaviors.

References

- [1] Gandomi A H, Yang X S., "Chaotic bat algorithm" *Journal of Computational Science*, 5(2): 224-232, 2014. <https://doi.org/10.1016/j.jocs.2013.10.002>
- [2] Gandomi A H, Yang X S, Talatahari S, et al. "Firefly algorithm with chaos," *Communications in Nonlinear Science and Numerical Simulation*, 18(1): 89-98, 2013. <https://doi.org/10.1016/j.cnsns.2012.06.009>
- [3] Gandomi A H, Yang X S, Alavi A H., "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with computers*, 29(1): 17-35. 2013. <http://dx.doi.org/10.1007/s00366-012-0308-4>
- [4] Liu J, Lampinen J., "A fuzzy adaptive differential evolution algorithm," *Soft Computing*, 9(6): 448-462, 2005. <https://doi.org/10.1007/s00500-004-0363-x>
- [5] Tan Y, Zhu Y., "Fireworks algorithm for optimization," *International conference in swarm intelligence*, Springer, Berlin, Heidelberg, 355-364, 2010. https://doi.org/10.1007/978-3-642-13495-1_44
- [6] Imran, A.M., Kowsalya, M., Kothari, D.P., "A novel integration technique for optimal network reconfiguration and distributed generation placement in power distribution networks," *International Journal of Electrical Power & Energy Systems*, 63, 461-472, 2014. <https://doi.org/10.1016/j.ijepes.2014.06.011>
- [7] Luo H, Xu W, Tan Y., "A discrete fireworks algorithm for solving large-scale travel salesman problem," 2018 *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1-8, 2018. <https://doi.org/10.1109/CEC.2018.8477992>
- [8] Lana I, Del Ser J, Vélez M., "A novel Fireworks Algorithm with wind inertia dynamics and its application to traffic forecasting," 2017 *IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 706-713, 2017. <https://doi.org/10.1109/CEC.2017.7969379>
- [9] Zhou X, Zhao Q, Zhang D.. "Discrete Fireworks Algorithm for Welding Robot Path Planning," *Journal of Physics: Conference Series*. IOP Publishing, 1267(1): 012003, 2019. <https://doi.org/10.1088/1742-6596/1267/1/012003>
- [10] Tuba E, Strumberger I, Bacanin N, et al., "Bare bones fireworks algorithm for feature selection and SVM optimization," 2019 *IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 2207-2214, 2019. <https://doi.org/10.1109/CEC.2019.8790033>
- [11] S. Zheng, A. Janecek, and Y. Tan, "Enhanced Fireworks Algorithm," 2013 *IEEE Congress on*, pp.2069-2077, 2013. <https://doi.org/10.1109/CEC.2013.6557813>
- [12] Xue Y, Zhao B, Ma T, et al., "A self-adaptive fireworks

- algorithm for classification problems,” *IEEE Access*, 6: 44406-44416, 2018.
<https://doi.org/10.1109/ACCESS.2018.2858441>
- [13] Li J, Tan Y., “A Comprehensive Review of the Fireworks Algorithm,” *ACM Computing Surveys (CSUR)*, 52(6): 1-28, 2019.
<https://doi.org/10.1145/3362788>
- [14] Barraza J, Melin P, Valdez F, et al., “Iterative fireworks algorithm with fuzzy coefficients,” 2017 *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, 1-6, 2017.
<https://doi.org/10.1109/FUZZ-IEEE.2017.8015524>
- [15] Zheng Y J, Xu X L, Ling H F, et al., “A hybrid fireworks optimization method with differential evolution operators,” *Neurocomputing*, 148: 75-82, 2015.
<https://doi.org/10.1016/j.neucom.2012.08.075>
- [16] Gong C., “Opposition-based adaptive fireworks algorithm,” *Algorithms*, 9(3): 43, 2016.
<https://doi.org/10.3390/a9030043>
- [17] Price K V., *Differential evolution[M]//Handbook of Optimization*, Springer, Berlin, Heidelberg, 187-214, 2013. https://doi.org/10.1007/978-3-642-30504-7_8
- [18] Peng H, Wu Z, Deng C., “Enhancing differential evolution with commensal learning and uniform local search,” *Chinese Journal of Electronics*, 26(4): 725-733, 2017.
<https://doi.org/10.1049/cje.2016.11.010>
- [19] Zhang J, Sanderson A C., “JADE: adaptive differential evolution with optional external archive,” *IEEE Transactions on evolutionary computation*, 13(5): 945-958, 2009.
<https://doi.org/10.1109/TEVC.2009.2014613>
- [20] X. J. Wang, H. Peng, C. S. Deng et al., “Firefly algorithm based on uniform local search and variable step size” . *Journal of Computer Applications*, vol. 38, no. 3, pp. 715-721, 2018.
http://en.cnki.com.cn/Article_en/CJFDTotal-JSYY201803020.htm
- [21] Lixian Li, Jaewan Lee. “The Variable Amplitude Coefficient Fireworks Algorithm with Uniform Local Search Operator,” *Journal of Internet Computing and Services*, vol. 21, no. 3, pp. 21-28, 2020.
<https://doi.org/10.7472/jksii.2020.21.3.21>
- [22] S. Das, A. Abraham, U. K. Chakraborty and A. Konar, “Differential Evolution Using a Neighborhood-Based Mutation Operator,” in *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526-553, June 2009. <https://doi.org/10.1109/TEVC.2008.2009457>

● 저 자 소 개 ●



Lixian Li

2005 B.S. in Computer Science and Technology, Zhengzhou University, Zhengzhou, P.R.China
 2010 M.S. in Computer Technology, Huazhong University of Science and Technology, Wuhan, P.R.China
 2005.07~2018.08 Lecturer at School of Information Science and Technology, Jiujiang University, Jiujiang, P.R.China
 2018.09~Present pursuing Ph.D. in Kunsan National University, Kunsan, Korea
 Research Interests: Intelligent Operation, Data Mining
 E-mail: llxkansan@kunsan.ac.kr



Jaewan Lee

1984 B.S. in Computer Engineering, Chung-Ang University, Seoul, Korea
 1987 M.S. in Computer Engineering, Chung-Ang University, Seoul, Korea
 1992 Ph.D. in Computer Engineering, Chung-Ang University, Seoul, Korea
 1995.03~present, Professor at the Department of Information and Communication Engineering, Kunsan National University
 Research Interests: distributed systems, cloud computing, data mining, and computer networks
 E-mail: jwlee@kunsan.ac.kr