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

Differential Evolution Strategy based on the Constraint of Fitness Values Classification

Zhihui Li[1], Zhigang Shang[1]
[1]School of Electrical Engineering
Zhengzhou University,
Zhengzhou, China
{lizhrain, zhigang_shang, liangjing}@zzu.edu.cn,

B. Y. Qu[2], J. J. Liang[1]
[2]School of Electric and Information Engineering
Zhongyuan University of Technology
Zhengzhou, China
qby1984@hotmail.com

***Abstract*—This paper presents a new Differential Evolution (DE) strategy, named as FCDE, based on the constraint of classification of fitness function values. To ensure the population could move to the better fitness landscape, the global fitness value distribution information of the objective function are used and all points in the population are classified into three class by their fitness values in each generation, so the points in each class choose their donor vector and differential vector from the points in adjacent senior class to form the trial vector. This strategy could speed up the convergence to global optimal as well as avoid falling into the local optimal. Another attractive character of FCDE is the control parameters in this DE variant are self-adaptive. This method is tested on the 30 benchmark functions of CEC2014 special session and competition on single objective real-parameter numerical optimization. The experimental results showed acceptable reliability of this strategy in high search dimension. This paper will participate in the competition on real parameter single objective optimization to compare with other algorithms.**

Keywords—Differential Evolution; Constraint optimization; Fitness Values; Classification

I. INTRODUCTION

Differential Evolution (DE) has been one of popular real coded optimization algorithm because of its simplicity but good performance in evolutionary computation. Now there are great deals of variants of the basic algorithm with improved performance since its inception in 1995, and recent advance of DE are comprehensively summarized by Neri and Tirronen [1] and by Das and Suganthan [2].

In the standard DE computation frame, there are two key questions to be considered. One is what evolution strategy be used, and the other is how to determine the control parameters. For the second question, there are many works have made good contribution and significative try [3,4,5]. We know that the suitable values of three control parameters, including the mutation scale factor F, the crossover ratio CR, and the population size NP, depend on the complexity, dimensionality and scale of the question, and are selected generally by numerical experiment trial, prior experience or self-adaptive. The self-adaptation approach to dynamically determine control parameters are introduced more and more, usually is applied to tune the control parameters F and CR[6-9]. One popular example is SaDE proposed by Qin [6]. In SaDE, both the trial vector generation strategies and their associated control parameters CR are gradually self-adapted by learning from their previous experiences of generating promising solutions, and the control parameters F is sampled by the presetting Gauss distribution. In many test functions and practice application, SaDE shows better performance than regular DE on difficult fitness landscapes.

For the first question, there are main kinds of mutation scheme in DE to get a possible better solution, including DE/rand/1, DE/best/1, DE/target-best/1, DE/rand/2, DE/best/2 and so on. In the Das' survey[2], the experiments performed by Mezura-Montes et al. indicate that DE/best/1/bin remained the most competitive scheme, regardless the characteristics of the problem to be solved, based on final accuracy and robustness of results. However, on multimodal and nonseparable functions, DE/rand/2/dir remained most competitive and slightly faster to converge to the global optimum. We can see that the fitness value of objective function, in these successful mutation schemes, is used indirectly or directly to form the trial vector. The idea is very strange that only depending on the differential vector between two random vectors, without regard for their fitness values, to get the best result.

One advantage of DE is that the differential vector between the two or more points in population provides more possible evolution direction. How far to go in this direction and whether it is credible both need the fitness values information. All kinds of control parameters self-adaptation DE should be able to 'learn' the current conditions of the fitness landscape. As we know, if the direction to the position with the current best fitness value always be chosen in evolution process, the diversity of population will be decreased and it is easy to fall into the local optimal. However, if the points choose some other points with better fitness value totally random to form differential vector, it will be ruleless or slower evolution. To keep the good balance between the randomness and regularity, this paper presents a new Differential Evolution (DE) strategy, named as FCDE, based on the constraint of classification of fitness function values. The points in each class with different fitness values choose their donor vector and differential vector from the points in adjacent senior class to form the trial vector.

The remainder of this paper is organized follows: Section II outlines the main scheme of DE and its main variants;

Section III describes the strategy of FCDE and its scheme in details; Section IV gives the test results of FCDE in the different CEC14 benchmark functions in 10D and 30D, compared with other DE variants; Finally, Section V concludes the paper with discussion of results and suggestions for future work.

II. SCHEME OF DE AND ITS VARIANTS

A. Scheme of Standard DE

For a real parameter single-objective optimization question, any possible solution x can be represented as a vector or point in D-dimension solution space. The DE uses the differential vector between two or more points to perturb one point in the same solution space to get the trial point. The original DE framework is remarkably simple, described as follow:

1) Population

$$P_{x,g} = (x_{i,g}), i = 0,1,\cdots NP-1, g = 0,1,\cdots, g_{\max}$$
$$x_{i,g} = (x_{j,i,g}), j = 0,1,\cdots D-1 \quad (1)$$

where NP denotes the number of solution vectors in population, g defines the generation counter, and D is the dimension of parameter space.

2) Population initialization

$$x_{j,i,0} = rand_j[0,1) * (b_{j,U} - b_{j,L}) + b_{j,U} \quad (2)$$

For a D-dimensional initialization vector X_i, bL and bU indicate the lower and upper bounds of the parameter $X_{j,i}$. The random number generator, $rand_j[0,1)$, returns an uniformly distributed random number from within the range [0,1).The subscript, j, indicates that a new random value is generated for each parameter.

3) Mutation with Difference Vectors

$$V_{i,g} = X_{r1,g} + F * (X_{r2,g} - X_{r3,g}) \quad (3)$$

The perturbation of a base vector $X_{r1,g}$ by using a difference vector between other two random vectors will generate a new mutation donor vector. In above equation, F is the mutation scale factor. The three vectors in mutation are all randomly chosen from population and each should be mutually exclusive.

4) Crossover for Diversity Enhancement

$$U_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} & if\,(rand_j[0,1) \le CR) \\ x_{j,i,g} & otherwise \end{cases} \quad (4)$$

where $U_{i,g}$ is the trial vector which mixes parameters of the donor vector $V_{i,g}$ and the so-called target vector $X_{i,g}$ in order to enhance the diversity of population. CR is the crossover ratio.

5) Selection

DE uses simple one-to-one survivor selection where the trial vector $U_{i,g}$ competes against the target vector $X_{i,g}$. The vector with the lowest objective function value survives into the next generation g+1.

$$X_{i,g+1} = \begin{cases} U_{i,g} & if\ f(U_{i,g}) \le f(X_{i,g}) \\ X_{i,g} & otherwise \end{cases} \quad (5)$$

B. Popular DE Variants

1) Mutation Strategy

The mutation scheme used in different DE-variants leads to the different performance in different fitness function. The notation to classify the various DE-variants is defined by DE/x/y/z where x denotes the base vector, y denotes the number of difference vectors used, and z representing the crossover method. Mostly used notations are listed as following:

$$\begin{aligned}
&\text{"DE/rand/1"}: V_{i,g} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) \\
&\text{"DE/best/1"}: V_{i,g} = X_{best,G} + F(X_{r1,G} - X_{r2,G}) \\
&\text{"DE/current-to-best/1"}: V_{i,g} = X_{i,G} + F(X_{best,G} - X_{i,G}) + F(X_{r1,G} - X_{r2,G}) \\
&\text{"DE/best/2"}: V_{i,g} = X_{best,G} + F(X_{r1,G} - X_{r2,G}) + F(X_{r3,G} - X_{r4,G}) \\
&\text{"DE/rand/2"}: V_{i,g} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) + F(X_{r4,G} - X_{r5,G})
\end{aligned} \quad (6)$$

2) Self-Adaptive DE

Here we summarize the main scheme use in SaDE as an example Self-Adaptive DE.

SaDE selects mutation strategies “DE/rand/1” and “DE/current to best/1” in Eq.(6) as candidates, and produces the trial vector based on the probability to decide which mutation strategy is selected. The probability is learned from those offspring successfully entering the next generation after some iteration.

In SaDE, mutation scale factor F in ith generation is self-adaptive sampled from a Gaussian distribution:

$$F_i = N_i(0.5, 0.3) \quad (7)$$

SaDE allocates a CR for each individuals in ith generation according to:

$$CR_i = N_i(CR_m, 0.1) \quad (8)$$

where CRm is learned from those offspring successfully entering the next generation after some iteration.

III. SCHEME OF FITNESS CLASSIFICATION CONSTRAINT DE

As mentioned above, the differential vector provides a possible evolution direction in the D dimension search space. A good direction will lead the population move to the optimal. There lies a dilemma when the point move to the optimal because we don’t know this optimal is local or global. So if the point moves to the one has the best fitness value, maybe it will fall into the local optimal; meanwhile when it moves to the one has better fitness value, the possible candidate direction maybe are nimiety and the time going to the global optimal will be much longer.

Without loss of generality, some points belong to the same class will have similar character while they belong to the different class will have different character. Similar with social rank, if the points in population are classified according to their fitness values, it could choose some points in senior class to generate evolution direction, regardless that points in same class even with better fitness values. So we propose a new Differential Evolution (DE) strategy, named as FCDE, based on the constraint of classification of fitness function values.

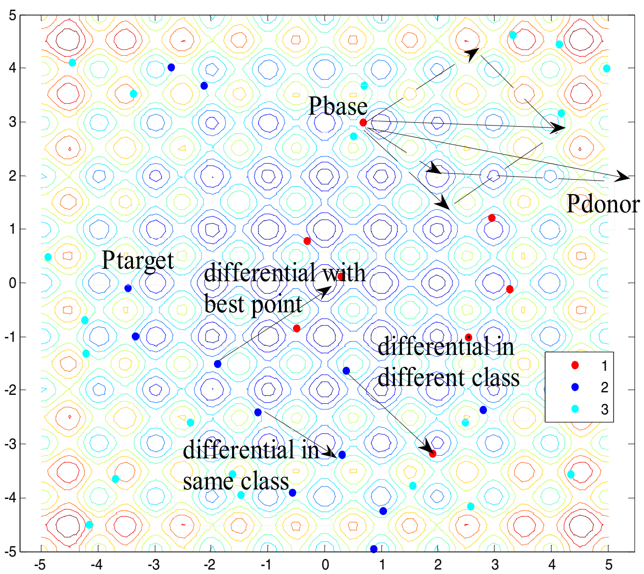

Fig. 1. Mutation strategy of FCDE

When the mutation direction is determined, the mutation scale factor F in ith generation is sampled from a Gaussian distribution like used in SaDE. Because the mutation strategy of FCDE will generate donor point with better fitness value with high possibility, so the crossover should be more. The crossover ratio CR is preset to 0.9.

The basic structure of FCDE algorithm is shown as follows:

Algorithm of FCDE

1: generate an initial population $P=[X_1,X_2,\cdots,X_{NP}]$, $X_i \in \Omega$ distributed uniformly.

2: evaluate Fitness value of objective function

$$f(X_i), i=1,2,\cdots,NP$$

3: classify all points in population into 3 classes by their fitness value. The lower class index is corresponding to the better fitness.

$$f_C(X_i)=k, k\in[1,2,3]$$

4: while stopping condition not reached do

$F=[0.5,0.3]$

$CR=0.9$

Generation: $g=g+1$

5: for i = 1 to NP do

6: get the corresponding class index num $f_C(i)$

7: generate a donor vector

$$V = X_{r\in fc(i)-1} + F\bullet(Xbest - X_{r\in fc(i)}) + F\bullet(X_{r\in fc(i)-1} - X_{r\in fc(i)}) + F\bullet(X_{r1\in fc(i)} - X_{r2\in fc(i)})$$

Where r means sample randomly point in given class.

If $f_C(i)-1=0$, $X_{r\in fc(i)-1}=X_{best}$

8: generate a trial vector $U_{i,g}$

9: evaluate objective function $f(U_{i,g})$

10: if $f(U_{i,g})\le f(X_{i,g})$ then

insert $U_{i,g}$ into new generation g+1

else

insert $X_{i,g}$ into new generation g+1

11: end for

12: update the f_C

13: end while

IV. EXPERIMENT RESULTS

PC Configuration:

System: Windows XP; CPU: 3.40GHz

RAM: 2.00 GB;

Language: Matlab 2012a

Algorithm: FCDE

Parameters Setting:

a) Parameters to be adjusted:

F, CR, NP,FEs

b) Corresponding dynamic ranges:

NP:10*Dimension of test Function

Max_FES: 10000*D

c) Guidelines on how to adjust the parameters

Mutation scale factor F in ith generation is self-adaptive sampled from a Gaussian distribution: $F_i = N_i(0.5, 0.3)$.

d) Actual parameter values used.

$CR = 0.9$

Experiments are progressed over a suite of 30 single objective real parameter optimization benchmark functions of CEC 2014[10]. For each function, the FCDE is run 51 times. The computation complexity is given in Table I and Tables II-III reports the results. The predefined tolerance value for the 30 test functions is 1e-8.

TABLE I. COMPUTATIONAL COMPLEXITY

	T0	T1	$\hat{T}2$	$(\hat{T}2-T1)/T0$
10D	0.1563	2.50E+00	4.07E+01	2.45E+02
30D		2.00E+00	2.59E+02	1.65E+03

TABLE II RESULTS FOR 10D

F	Best	Worst	Median	Mean	Std
1	0.00E+00	1.56E-02	0.00E+00	3.06E-04	2.18E-03
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4	0.00E+00	3.47E+01	3.47E+01	1.92E+01	1.66E+01
5	1.49+01	2.04+01	2.03E+01	2.02E+01	7.57E-01
6	2.60E-01	6.28E+00	3.13E+00	3.15E+00	1.45E+00
7	2.95E-02	1.06E+00	1.91E-01	2.18E-01	1.61E-01
8	2.98E+00	3.28E+01	1.49E+01	1.68E+01	6.94E+00
9	7.96E+00	5.07E+01	2.29E+01	2.21E+01	8.38E+00
10	3.12E-01	7.98E+02	2.39E+02	2.54E+02	1.93E+02
11	1.42E+02	1.39E+03	6.97E+02	7.22E+02	2.87E+02
12	1.28E-01	1.40E+00	7.04E-01	7.24E-01	3.29E-01
13	9.90E-02	8.01E-01	3.44E-01	3.61E-01	1.43E-01
14	1.07E-01	1.15E+00	3.08E-01	4.07E-01	2.75E-01
15	6.54E-01	2.95E+00	1.47E+00	1.62E+00	6.74E-01
16	2.12E+00	3.62E+00	3.14E+00	3.10E+00	3.30E-01
17	1.02E+01	7.12E+02	3.00E+02	3.04E+02	1.84E+02
18	2.21E+00	5.49E+01	2.05E+01	2.40E+01	1.49E+01
19	8.34E-02	7.24E+00	1.93E+00	2.56E+00	1.72E+00
20	1.03E-01	5.00E+01	1.08E+01	1.51E+01	1.32E+01
21	1.96E-01	5.46E+02	1.25E+02	1.54E+02	1.31E+02
22	9.66E-02	6.02E+01	2.12E+01	2.40E+01	9.74E+00
23	3.29E+02	3.29E+02	3.29E+02	3.29E+02	2.66E-13
24	1.20E+02	1.80E+02	1.34E+02	1.38E+02	1.37E+01
25	1.36E+02	2.03E+02	1.99E+02	1.84E+02	2.07E+01
26	1.00E+02	1.01E+02	1.00E+02	1.00E+02	1.59E-01
27	3.33E+00	4.00E+02	7.93E+00	2.55E+01	7.77E+01
28	3.57E+02	7.09E+02	4.85E+02	4.90E+02	9.61E+01
29	2.10E+02	3.12E+06	2.34E+02	2.06E+05	7.25E+05
30	4.60E+02	1.90E+03	7.40E+02	8.87E+02	3.48E+02

TABLE III RESULTS FOR 30D

F	Best	Worst	Median	Mean	Std
1	5.36E+03	2.65E+05	5.46E+04	6.54E+04	4.90E+04
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	1.70E-05	8.56E+02	4.03E-01	3.51E+01	1.26E+02
4	1.60E-06	1.62E+02	8.82E-01	1.62E+01	3.23E+01
5	2.07E+01	2.10E+01	2.09E+01	2.09E+01	7.23E-02
6	1.51E+01	2.89E+01	2.20E+01	2.20E+01	3.36E+00
7	0.00E+00	3.65E-01	1.72E-02	2.89E-02	5.41E-02
8	4.28E+01	1.47E+02	9.25E+01	9.44E+01	2.46E+01
9	7.26E+01	2.22E+02	1.31E+02	1.33E+02	2.89E+01
10	5.87E+02	3.51E+03	2.33E+03	2.24E+03	5.02E+02
11	2.30E+03	4.63E+03	3.33E+03	3.40E+03	5.99E+02
12	4.36E-01	2.73E+00	1.49E+00	1.56E+00	5.83E-01
13	3.52E-01	9.61E-01	5.83E-01	5.98E-01	1.16E-01
14	1.88E-01	1.31E+00	3.38E-01	4.60E-01	2.82E-01
15	3.69E+00	4.15E+01	1.41E+01	1.59E+01	7.64E+00
16	1.07E+01	1.29E+01	1.21E+01	1.21E+01	5.74E-01
17	4.91E+02	1.39E+04	3.01E+03	4.00E+03	3.17E+03
18	2.10E+01	3.58E+02	1.17E+02	1.21E+02	6.62E+01
19	6.47E+00	6.92E+01	1.09E+01	1.33E+01	1.16E+01
20	2.44E+01	5.50E+02	1.17E+02	1.45E+02	9.98E+01
21	1.43E+02	1.27E+04	1.06E+03	1.88E+03	2.37E+03
22	1.46E+02	1.02E+03	5.62E+02	5.44E+02	2.35E+02
23	3.15E+02	3.15E+02	3.15E+02	3.15E+02	1.67E-12
24	2.30E+02	2.66E+02	2.51E+02	2.50E+02	6.82E+00
25	2.03E+02	2.12E+02	2.05E+02	2.05E+02	2.29E+00
26	1.00E+02	1.01E+02	1.01E+02	1.01E+02	1.10E-01
27	4.00E+02	1.04E+03	4.49E+02	6.18E+02	2.32E+02
28	1.02E+03	2.71E+03	1.40E+03	1.50E+03	3.75E+02
29	5.33E+02	1.37E+07	7.92E+02	1.06E+06	3.28E+06
30	8.28E+02	4.73E+03	2.39E+03	2.53E+03	9.82E+02

The computational complexity of FCDE and its test results for benchmark function are listed in TABLE Ⅰ, Ⅱ andⅢ. As can be seen from the results, FCDE shows a good performance for most test function, no matter in 10D or 30D.

The convergence maps of a random run for function No.01~ No.30 are given in Fig.2. The results show that our approach is able to find a very good feasible solution quickly among those 30 test functions. It shows our mutation strategy with constraint of fitness classification could keep the good balance between the randomness and regularity and speed up the convergence to global optimal; meanwhile, avoid falling into the local optimal in 10D, in spite of not perform very well in high dimension situation.

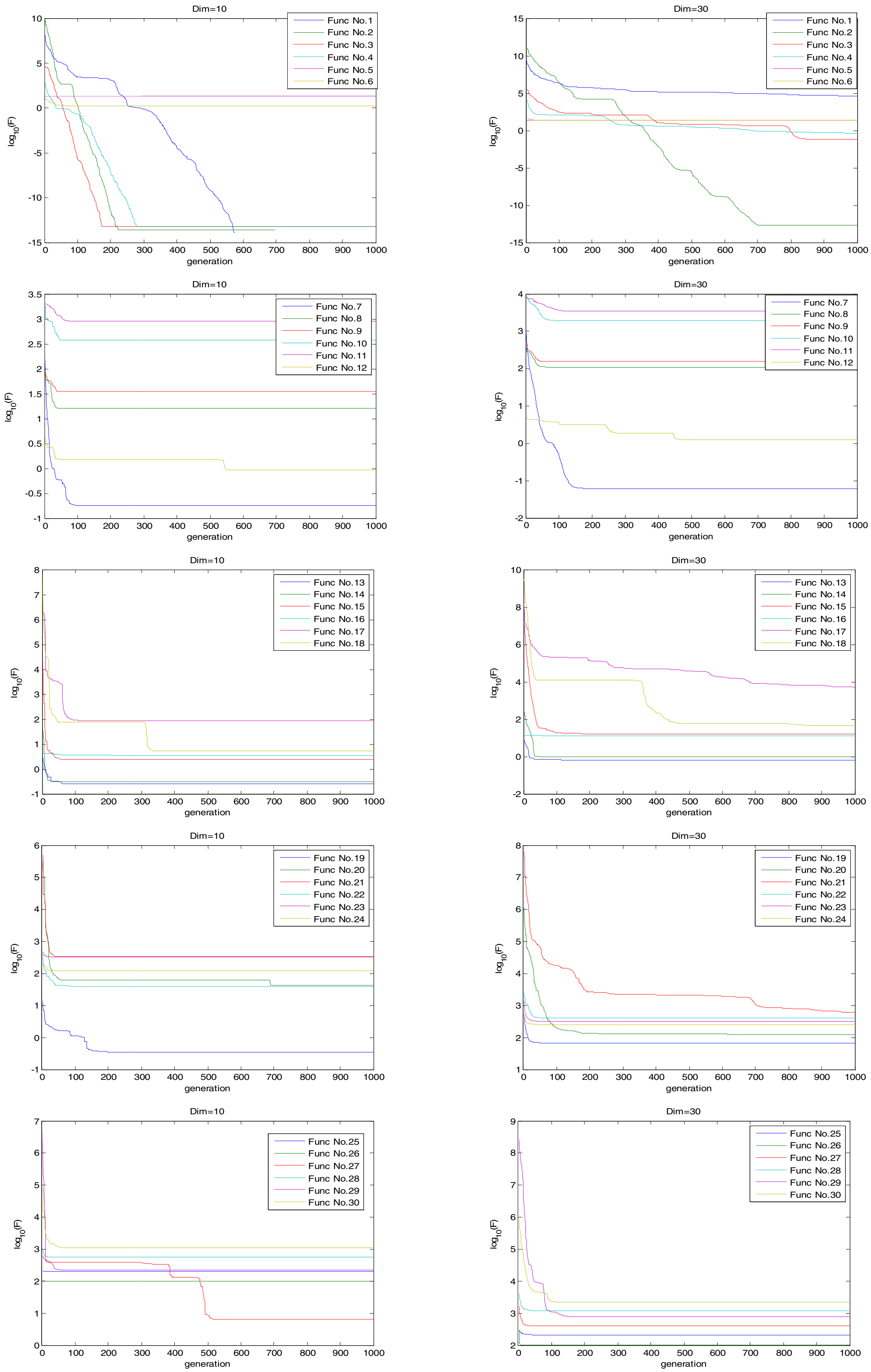

Fig. 2. Convergence Maps of FCDE for Functions No.1~ No.30

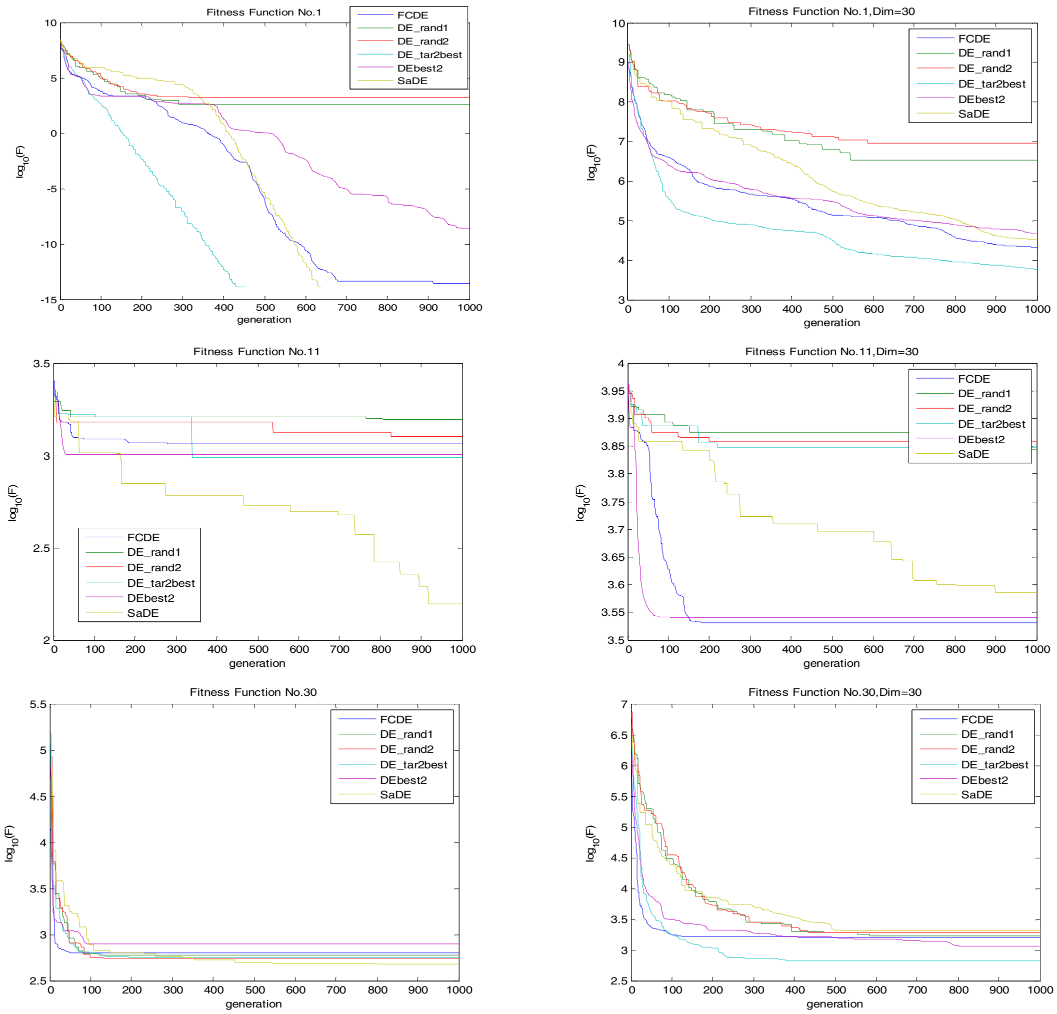

Fig. 3. Comparison of convergence maps between FCDE and other DE variants

The comparison of convergence maps between FCDE and other DE variants are given in Fig.3, in which a random run for function No.01, No.11, No.30 as a example. The better performance is generated by the FCDE in most functions and its convergence speed is acceptable even in 30D.

The test result also indicates for some complex functions such as Function 1,10 and 11, The FCDE is not good enough to find the global optimal point. It indicates although FCDE has good convergence performance but it need to be improved for global search ability.

V. Conclusion

In this paper, we propose a new DE mutation strategy to improve the performance, named as FCDE, based on the constraint of classification of fitness function values was proposed to solve single objective rained real parameter optimization problems. In this method, all control parameter are not need to be preset for specific case. According to the new mutation strategy, all points in the population are classified into three classes by their fitness values in each generation by considering the global fitness value distribution information of the objective function to ensure the population could move to the better fitness landscape. The test results of FCDE on 30 competition benchmark functions of CEC 2014 indicate it has good performance in balance between explo- rative and exploitative, especially in high search dimension.

In this paper, the class number of population classifying with their fitness value is set to 3. It is groundless, but just intuitive. In the future work, we will compare the effect with different class number to optimization result.

In this research, we don't considering the neighborhood information of every point. In the future work, we will combine the neighborhood information to generate the donor vector to improve its global explorative performance. In a word, FCDE is on its initial stage, there are lots of works to do. The following research will be focused on improving FCDE.

Acknowledgement

This paper is supported by National Natural Science Foundation of China (U1304602), Scientific and Technological Project of Henan Province (122300410264), National Natural Science Foundation of China (61305080), Scientific and Technological Project of Henan Province (132102210521), Postdoctoral Science Foundation of China (Grants 20100480859) and Specialized Research Fund for the Doctoral Program of Higher Education (20114101110005).

REFERENCES

[1] F. Neri and V. Tirronen, “Recent advances in differential evolution: a survey and experimental analysis,” Artificial Intelligence Review, vol. 33, pp. 61–106, 2010.

[2] S. Das and P. N. Suganthan, “Differential Evolution: A Survey of the State-of-the-Art,” IEEE. Trans. Evolutionary Computation, vol. 15, No.1, pp. 4–31, 2011.

[3] Uday K. Chakraborty(Ed.), Advances in Differential Evolution, Springer, 2008.

[4] Grant Dick,"The Utility of Scale Factor Adaptation in Differential Evolution", WCCI 2010 IEEE World Congress on Computational Intelligence, July, 2010, pp. 4355-4362.

[5] R. Storn, "Differential Evolution Research–Trends and Open Questions",U.K. Chakraborty (Ed.): Advances in Differential Evolution, SCI 143, pp. 1–31, 2008.

[6] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” IEEE Trans. Evol. Comput., vol. 13, no. 2, pp. 398–417,Apr. 2009.

[7] J. Brest, B. Boskovic, S. Greiner, V. Zumer, and M. S. Maucec, “ Performance comparison of self-adaptive and adaptive differential evolution algorithms,” Soft Comput., vol. 11, no. 7, pp. 617–629, 2007.

[8] Z. Yang, K. Tang, and X. Yao, “Self-adaptive differential evolution with neighborhood search,” in Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on, June 2008, pp. 1110–1116.

[9] Zhigang Shang; Zhihui Li; Liang, J.J.; Ben Niu, "Control parameters self-adaptation in differential evolution based on intrisic structure information," Evolutionary Computation (CEC), 2012 IEEE Congress on , vol., no., pp.1,6, 10-15 June 2012

[10] J. J. Liang, B-Y. Qu, and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization", Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, December 2013.