

# Robust Differential Evolution for Solving Numerical Optimization Problems

Chun-Ling Lin<sup>1</sup>, Sheng-Ta Hsieh<sup>2,\*</sup>, Huang-Lyu Wu<sup>2</sup>, Tse Su<sup>2</sup>

<sup>1</sup> Department of Electrical Engineering, Ming Chi University of Technology New Taipei City, Taiwan, R.O.C.

<sup>2</sup> Department of Communication Engineering, Oriental Institute of Technology, New Taipei City, Taiwan, R.O.C.

\* fo013@mail.oit.edu.tw

**Abstract**—In this paper, the robust mutation strategy for differential evolution (DE) is proposed for enhancing its solution searching abilities. Also, the elitist crossover is involved to produce potential vectors. In the experiments, fifteen CEC 2005 test functions, which include uni-modal and multi-modal functions, are adopted for testing the proposed method and compare its performance with three DE variants. From the results, it can be observed that the proposed method performs better than other DE approaches on most test functions.

**Keywords**- differential evolution; elitist crossover; optimization; robust mutation; vector.

## I. INTRODUCTION

In 1995, the concept of original differential evolution (DE) was first proposed by Storn and Price [1]-[4]. It's a vector-based evolutionary algorithm with simple concept and high efficiency. In recent years, more and more DE variants were proposed and have been applied in various filed for solving real-world optimization problems.

There are more and more DE variants were proposed in last two decades. In 2006, Brest *et al.* proposed self-adaptive differential evolution (jDE) [5], which focused on adjust control parameters  $F$  and  $Cr$  correspond to each individual.

In 2009, Huang and Suganthan proposed a DE with multi-mutation named SaDE [6]. It combined four mutation strategies "DE/rand/1", "DE/best/1", "DE/current-to-best/1" and "DE/current-to-rand/1". Also, it involved self-adjusted parameters to generate potential solutions according to their previous experiences.

In 2011, Mallipeddi and Suganthan proposed a DE variant named EPSDE [7]. In EPSDE, each vector will be assigned one of mutation strategies. The parameters  $F$  and  $Cr$  will be randomly selected from the pools. If the target vector is better than the trial vector, a randomly selected mutation strategy will be assigned for the target vector and the parameters will also be re-selected in next generation. In the same year, Wang *et al.* proposed CoDE[8], which involved three mutation strategies, after evaluate the three trail vectors' fitness values, the best one is then kept.

In 2012, the EIDE was proposed by Zou and Gao [9] which is aiming to parameters adjustment. Thus, in EIDE, the scale

factor  $F$  is set as uniform distribution and the crossover rate  $Cr$  was increase linearly.

Although there are many DE variants were proposed, solving optimization problems with capable rapid convergence are still an active area of research. In this paper, an improved mutation strategy named robust mutation is proposed for enhancing DE's solution searching ability. Also, the elitist crossover is involved to speedup convergence and to perform deeply search.

This paper is organized as follows. The basic concept of differential evolution is described in section II. The detail of proposed method is introduced in section III. The experiments results are in section IV. Finally, the conclusions are described in section V.

## II. DIFFERENTIAL EVOLUTION

Differential evolution (DE) is arguably one of the most powerful stochastic real-parameter optimization algorithms in the mainstream. Its main characteristic include have a few parameters, simply structure and fast convergence, etc. The members of population in DE are called parameter vectors. Vectors' movements are according to trial vectors. The six common mutation strategies are listed as follows.

1. DE/rand/1

$$V_{i,G} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) \quad (1)$$

2. DE/best/1

$$V_{i,G} = X_{best,G} + F(X_{r1,G} - X_{r2,G}) \quad (2)$$

3. DE/current to best/1

$$V_{i,G} = X_{i,G} + F(X_{best,G} - X_{i,G}) + F(X_{r1,G} - X_{r2,G}) \quad (3)$$

4. DE/current to rand/1

$$V_{i,G} = X_{i,G} + F(X_{r1,G} - X_{i,G}) + F(X_{r2,G} - X_{r3,G}) \quad (4)$$

5. DE/ rand/2

$$V_{i,G} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) + F(X_{r4,G} - X_{r5,G}) \quad (5)$$

6. DE/best/2

$$V_{i,G} = X_{best,G} + F(X_{r1,G} - X_{r2,G}) + F(X_{r3,G} - X_{r4,G}) \quad (6)$$

where  $i, r_1, r_2, r_3, r_4$  and  $r_5$  denote current vector and five random selected vectors of population, respectively, and  $i \neq r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5$ . The  $G$  represents current generation,  $X_{best}$  denotes the best vector of the population, and  $F$  denotes scale factor and  $F \in (0, 1]$ . The  $X$  is the target vector and  $V$  is the donor vector.

Unlike traditional GA [10], the crossover in DE will be performed after mutation. Crossover in DE will be activated by crossover rate  $Cr$ . It can prevent convergent prematurely. Regardless the change of the mutation mechanism, DE algorithms adopt uniform crossover. After mutation mechanism, the crossover is then performed by selecting current vector and donor vector to produce trial vectors by following equation.

$$U_{i,G} = \begin{cases} V_{i,G}, & \text{if } (rand_{i,j}[0,1] \leq Cr \text{ or } j = j_{rand}) \\ X_{i,G}, & \text{otherwise} \end{cases} \quad (7)$$

where  $Cr \in [0,1]$  denotes crossover rate,  $j$  denotes dimension. If the trial vector ( $U$ ) obtained by crossover-stage is better than target vector ( $X$ ). It will be kept for next iteration. Otherwise, the trail vector will be updated and replaced in next iteration. The procedures that mentioned above will keep going until the termination condition is reached. The selection in DE is performed by following equation.

$$X_{i,G+1} = \begin{cases} U_{i,G}, & \text{if } f(U_{i,G}) \leq f(X_{i,G}) \\ X_{i,G}, & \text{otherwise} \end{cases} \quad (8)$$

### III. PROPOSED METHOD

In order to enhancing DE's solution searching ability, the robust mutation is proposed. It can generate useful moving vectors to guide vectors for finding better solutions. Also, the elitist crossover is involved to produce better vectors and they will guide others vector to perform deeply search.

#### A. Robust Mutation

In general, mutation of DE is to generate moving vector for guiding vectors toward to potential solution space. In fact, after several iterations, vectors will get closer around local optimum. If all vectors only toward to around the global best vector, diversity between vectors will keep decrease. In order to solve this situation, a new mutation strategy is proposed, named DE/current to focus/1. Thus, vectors will not only toward to the best vector's location but also other better vectors. The vector update equation is listed as follows.

$$V_{i,G} = X_{i,G} + F \cdot (X_{robust,G} - X_{i,G}) + F \cdot (X_{r_1,G} - X_{r_2,G}) \quad (9)$$

where  $X_{robust,G}$  represents one of top  $R\%$  vectors of populations and  $X_{i,G} \neq X_{robust,G}$ . The  $X_{r_1,G}$  and  $X_{r_2,G}$  are two random selected vectors, and their ate between  $[0, 2]$ . The parameter  $R$  is generated as follows.

$$R = 30 * (1 - \frac{gen}{max\_gen}) + 5 \quad (10)$$

where  $gen$  is current generation,  $max\_gen$  is maximum generations. Thus, the value  $R$  will keep decreased linearly, from 30 to 5, while solution searching is in progress. Thus, the vector will not just get closer around the global best vector. It can increase diversity between vectors.

#### B. Elitist Crossover

For the basic DE, the crossover is to randomly select donor vector and target vector for information exchange. In fact, the trial vector, which combine a part of donor vector and target vector, may useless for finding potential solutions. In this paper, the elitist crossover is proposed. For each generation, randomly select one of top 30% target vectors to perform crossover with donor vector for increase vectors' exploration ability and speedup convergence.

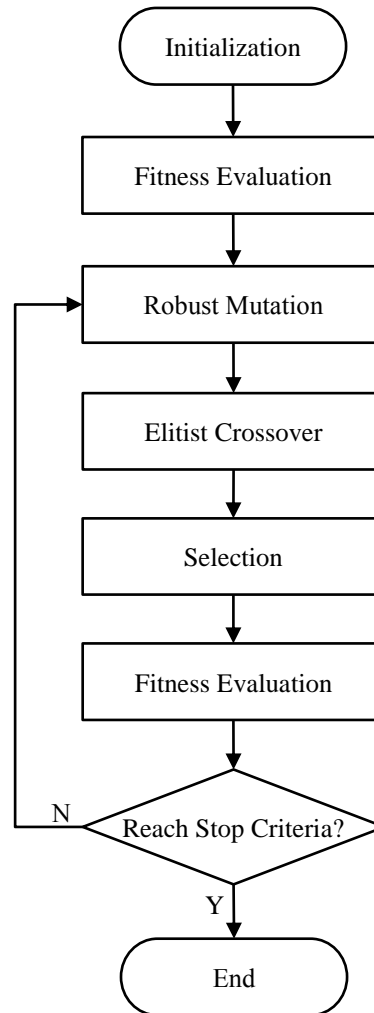


Fig. 1 Flowchart of Proposed Method

#### C. Parameter Adjustment

In the proposed method, the scale factor  $F_i$  of each particle is independently generated in each generation according to following equation

$$F_i = (\text{round}(\frac{i-1}{NP*0.1}) + 1) \cdot 0.1 \quad (11)$$

where  $F_i$  denotes scale factor of  $i$ th vector.

Similar, in order to adjust crossover rate according current solution search status, the  $Cr$  will be generated as follows.

$$Cr_i = \left( fix \left( \frac{i-1}{NP*0.1} \right) + 1 \right) \cdot 0.05 + 0.4 \quad (12)$$

where  $Cr_i$  represents crossover rate of  $i$ th vector. Thus, for the proposed, the crossover rate is not a fixed value.

The flowchart of proposed is presented in Fig 1 and the procedure of proposed method is listed as follows.

Step 1: Initialization.

Step 2: Fitness Evaluation.

Step 3: Robust mutation by (9).

Step 4: Elitist Crossover.

Step 5: Selection by (7)

Step 6: Fitness Evaluations.

Step 7: Repeat step 3 to step 6, until meet the stop criteria.

#### IV. EXPERIMENT RESULTS

##### A. Test functions

In order to test the performance of proposed method and compare it with other variants of DE, 15 test functions of CEC 2005 [14] are adopted for experiments. In test functions, functions  $f_1 \sim f_5$  are uni-modal, functions  $f_6 \sim f_{14}$  are multi-modal, and function  $f_{15}$  is hybrid composition. All the test functions are set as 30 dimensions. In order to easier compare the performance between optimizers, the error value  $e$  between the real global optimum  $f^*$  and function value  $f$  found by optimizer will be presented. The error value can be calculated as follows.

$$e = f - f^* \quad (13)$$

The initialization range and search range of 15 test functions are listed in Table I. In the experiments, the dimensions  $D$  of all test functions are set as 30.

TABLE I. INITIAL RANGE AND SEARCH RANGE OF TEST FUNCTIONS

Functions	Initial Range	Search Range
$f_1 \sim f_4$	$[-100, 100]^D$	$[-100, 100]^D$
$f_5$	$[-100, 100]^D$	$[-100, 100]^D$
$f_6$	$[-100, 100]^D$	$[-100, 100]^D$
$f_7$	$[0, 600]^D$	No Boundary
$f_8$	$[-32, 32]^D$	$[-32, 32]^D$
$f_9 \sim f_{10}$	$[-5, 5]^D$	$[-5, 5]^D$
$f_{11}$	$[-0.5, 0.5]^D$	$[-0.5, 0.5]^D$
$f_{12}$	$[-\pi, \pi]^D$	$[-\pi, \pi]^D$
$f_{13}$	$[-5, 5]^D$	$[-5, 5]^D$
$f_{14}$	$[-100, 100]^D$	$[-100, 100]^D$
$f_{15}$	$[-5, 5]^D$	$[-5, 5]^D$

##### B. Parameters Settings

In the experiments, three related works of DE, which include jDE[5], CoDE[8], and EIDE[9], are conducted to compare with the proposed method. The parameters of these methods are according to their original settings. Including the proposed method, each DE approach is executed for 25 independent runs, and the maximum fitness evaluations (FEs) are set as 300,000. All the DE variants were implemented using MATLAB 2011b. The experiments were executed on Intel(R) Xeon(R) CPU E5-2650 2.00GHz with 128GB RAM on Windows 7 64-bits Professional.

TABLE II. EXPERIMENT RESULTS

Methods	Results	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$
jDE	Mean	<b>0.0000E+00</b>	6.1018E+00	1.5784E+06	6.5517E+01	<b>3.1298E+02</b>
	Std.	<b>0.0000E+00</b>	2.9782E+00	6.4185E+05	4.8732E+01	<b>3.3554E+02</b>
CoDE	Mean	2.6744E+04	4.4547E+04	2.7560E+08	4.8559E+04	2.4804E+04
	Std.	6.2854E+03	1.3075E+04	1.7585E+08	1.2227E+04	3.0979E+03
EIDE	Mean	<b>0.0000E+00</b>	4.5191E+01	3.3370E+07	1.8098E+03	2.3794E+03
	Std.	<b>0.0000E+00</b>	3.4751E+01	7.4222E+06	1.1452E+03	4.8786E+02
Proposed Method	Mean	<b>0.0000E+00</b>	<b>5.2393E-28</b>	<b>4.9637E+04</b>	<b>4.8631E-13</b>	1.6949E+03
	Std.	<b>0.0000E+00</b>	<b>2.3789E-28</b>	<b>2.7941E+04</b>	<b>1.1761E-12</b>	3.5892E+02
Methods	Results	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$
jDE	Mean	2.7097E+01	<b>5.3224E-03</b>	2.0941E+01	8.4170E-01	1.4328E+02
	Std.	2.6187E+01	<b>5.2001E-03</b>	4.4383E-02	1.9437E+00	6.6111E+00
CoDE	Mean	1.4728E+10	4.7973E+03	2.0907E+01	2.3301E+02	3.9830E+02
	Std.	6.8900E+09	4.7043E+02	5.5381E-02	2.5974E+01	4.9051E+01
EIDE	Mean	1.2887E+01	4.0907E+02	2.0932E+01	<b>0.0000E+00</b>	1.9068E+02
	Std.	2.2120E+01	1.5947E+02	4.5294E-02	<b>0.0000E+00</b>	1.2057E+01
Proposed Method	Mean	<b>4.0341E+00</b>	2.0760E-02	<b>2.0334E+01</b>	2.0594E+01	<b>3.0247E+01</b>
	Std.	<b>3.0236E+00</b>	1.6557E-02	<b>3.8117E-02</b>	6.6087E+00	<b>6.5090E+00</b>
Methods	Results	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$
jDE	Mean	3.3735E+01	1.4614E+05	5.1188E+00	1.3281E+01	3.3200E+02
	Std.	9.6663E-01	1.5386E+04	3.7732E-01	9.3654E-02	4.8485E+01
CoDE	Mean	3.4283E+01	6.3185E+05	7.1252E+01	1.3215E+01	7.8519E+02
	Std.	1.2698E+00	1.7214E+05	5.1617E+01	1.3759E-01	7.7281E+01
EIDE	Mean	3.1218E+01	8.6138E+04	3.1715E+00	1.3178E+01	<b>1.2229E+02</b>
	Std.	1.0205E+00	1.1131E+04	2.8892E-01	1.2802E-01	<b>4.5471E+01</b>
Proposed Method	Mean	<b>1.6055E+01</b>	<b>1.4771E+03</b>	<b>2.7115E+00</b>	<b>1.2218E+01</b>	3.3532E+02
	Std.	<b>1.0605E+00</b>	<b>1.6809E+03</b>	<b>5.7500E-01</b>	<b>1.1965E-01</b>	4.8159E+01

##### C. Experimental Results

In experiments, four DE variants are executed for 25 independent runs. The experiment results are listed in Table II which presents the mean and standard deviation of error value  $e$  of 15 test functions. The best results among the four DE variants are shown in bold.

For function 1, jDE, CoDE and the proposed method can find the optimal solution. Besides, only the EIDE can find the optimal solution on function 9. The jDE performs better on function 5 and 7. The EIDE performs better on function 15. For the rest functions, the proposed method performs the best on function 2, 3, 4, 6, 8, 10, 11, 12, 13 and 14 among four DE approaches.

## V. CONCLUSIONS

In this paper, a robust mutation strategy is proposed to guide vectors toward to potential solution space efficiently. It can improve vectors' solution searching ability. Also, the elitist crossover can produce better trail vector for speedup convergence. Fifteen test functions of CEC 2005 are adopted for experiments through a reasonable average and fitness evaluations. From the results, it can be observed that the proposed method performs better than other three DE variants on most test functions.

## ACKNOWLEDGMENT

This work was supported in part by ministry of science and technology of Taiwan, R.O.C. under Grant MOST 103-2221-E-161-002.

## REFERENCES

- [1] R. Storn and K. V. Price, Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces, ICSI, Berkeley, CA, Tech. Rep. TR-95-012. [Online]. Available: <http://http.icsi.berkeley.edu/~storn/litera.html>
- [2] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.
- [3] K. Price, R. Storn, and J. Lampinen, *Differential Evolution—A Practical Approach to Global Optimization*. Berlin, Germany: Springer-Verlag, 2005.
- [4] R. Storn and K. V. Price, "Minimizing the real functions of the ICEC 1996 contest by differential evolution," in *Proc. IEEE Int. Conf. Evol. Comput.*, Nagoya, Japan, 1996, pp. 842–844.
- [5] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Selfadapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec. 2006.
- [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] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 1679–1696, Mar. 2011.
- [8] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. On Evolutionary Computation*, vol. 15, no. 1, pp. 55–66, 2011.
- [9] D. Zou and L. Gao, "An efficient improved differential evolution algorithm," in *Proc. 2012 31<sup>st</sup> Chinese Control Conference (CCC)*, Hefei, 2012.
- [10] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.
- [11] J. T. Tsai, T. K. Liu and J. H. Chou, "Hybrid Taguchi-genetic algorithm for global numerical optimization," *IEEE Trans. On Evolutionary Computation*, vol. 8, issue 4, pp. 365–377, 2004.
- [12] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*. New York: Wiley, 1997.
- [13] C. O. Imoru, "The power mean and the logarithmic mean," *Int. J. Math. Math. Sci.*, vol. 5, no. 2, pp. 337–343, 1982.
- [14] <http://www.ntu.edu.sg/home/EPNSuga>