Volume 4, Issue 6, June 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Comparative Study on the Different Process of Mutation of Differential Evolution

Ankur Mondal
Department of Computer Science And Engineering,Guru Nanak Institute of Technology
India

Sharbari Basu
Department of Computer Science and Engineering,Guru Nanak Institute of Technology
India

Santanu Kumar Sen
Department of Computer Science and Engineering,Guru Nanak Institute of Technology
India

Abstract- Differential evolution (DE) is arguably one of the most powerful stochastic real-parameter optimization algorithms in current use. DE operates through similar computational steps as employed by a standard evolutionary algorithm (EA). Over the last few decades, a number of Differential Evolution (DE) algorithms have been proposed with excellent performance on mathematical benchmarks. However, like any other optimization problems, the efficiency of Differential evolution is highly dependent on the process of mutation which contributes the donor vector. While testing the algorithm using different benchmark functions, convergence of the result varies with the variety of the test functions as well as with the different process of mutation. In this paper, a comparative study has been made on different process of mutation in DE Family of Storn and Price applied on different benchmark functions.

Keywords- DE, Mutation, binomial crossover, optimization, benchmark function

I. INTRODUCTION

Problems demanding globally optimal solutions are ubiquitous, yet many are intractable when they involve constrained functions having many local optima and interacting, mixed-type va-riables. The differential evolution (DE)[1],[2],[3],[4],[5],[6] algorithm emerged as a very comp-etitive form of evolutionary computing more than a decade ago. The first written article on DE appeared as a technical report by R. Storn and K. V. Price in 1995[4]. Packed with illustrations, algorithm, new insights, and practical advice, this volume explores DE in both principle and practice. It is a valuable resource for professionals needing a proven optimizer and for students wanting an evolutionary perspective on global numerical optimization.

II. DIFFERENTIAL EVOLUTIONS: BASIC CONCEPTS AND FORMULATION IN CONTINUOUS REAL SPACE

Scientists and engineers from all disciplines often have to deal with the classical problem of search and optimization. While optimizing performance of a system, we aim at finding out such a set of values of the system parameters for which the overall performance of the system will be the best under some given conditions. Two journal articles [3], [7] describing the algorithm in sufficient details followed immediately in quick succession. Usually, the parameters governing the system performance are represented in a vector like vector $\vec{X} = [x_1, x_2, x_3, ..., x_D]^T$. For real parameter optimization, as the name implies, each parameter x_1 is a real number. To measure how far the "best" performance we have achieved, an objective function (or fitness function) is designed for the system. The task of optimization is basically a search for such the parameter vector $\vec{X}$.

III. INITIALIZATION OF THE PARAMETER VECTORS

DE searches for a global optimum point in a D-dimensional real parameter space vector D. It begins with a randomly initiated population of NP D dimensional real-valued parameter vectors. Each vector, also known as *genome/chromosome*[1], forms a candidate solution to the multidimensional optimization problem. We shall denote subsequent generations in DE by G = 0, 1...,G_{max}. Since the parameter vectors are likely to be changed over different generations[1], we may adopt the following notation for representing the ith vector of the population at the current generation:

$$\vec{X}_{i,G} = [x_{1,i,G},, x_{2,i.G}, x_{3,i,G}, ..., x_{D,i,G}]. \quad 1)$$

For each parameter of the problem, there may be a certain range within which the value of the parameter should be restricted, often because parameters are related to physical components or measures that have natural bounds (for example if one parameter is a length or mass, it cannot be negative). The initial population (at G = 0) should cover this range as much as possible by uniformly randomizing individuals within the search space constrained by the prescribed minimum and maximum bounds: $\vec{X}_{\min} = \{x_{1,\min}, x_{2,\min}, ..., x_{D,\min}\}$ and vector $\vec{X}_{\max} = \{ x_{1,\max}, x_{2,\max}, ..., x_{D,\max} \}$. Hence we may initialize the jth component of the ith vector

$$x_{j,i,0} = x_{j,\min} + rand_{i,j}[0, 1] \cdot (x_{j,\max} - x_{j,\min}) \quad (2)$$

where $rand_{i,j}[0, 1]$ is a uniformly distributed random number lying between 0 and 1 (actually $0 \leq rand_{i,j}[0,1]$) and is instantiated independently for each component of the i-th vector.

IV. MUTATION WITH DIFFERENCE VECTORS

Biologically, "mutation" means a sudden change in the gene characteristics of a chromosome. In the context of the evolutionary computing paradigm, however, mutation is also seen as a change or perturbation with a random element. In DE-literature, a parent vector from the current generation is called target vector, a mutant vector obtained through the differential mutation operation is known as donor vector and finally an offspring formed by recombining the donor with the target vector is called trial vector. In one of the simplest forms of DE-mutation, to create the donor vector for each ith target vector from the current population, three other distinct parameter vectors, say $\vec{X}_{r1^i}$, $\vec{X}_{r2^i}$, $\vec{X}_{r3^i}$, are sampled randomly from the current population. The indices $r1^i$, $r2^i$, and $r3^i$ are mutually exclusive integers randomly chosen from the range [1, NP], which are also different from the base vector index i. These indices are randomly generated once for each mutant vector. Now the difference of any two of these three vectors is scaled by a scalar number F (that typically lies in the interval [0.4, 1])[1,4] and the scaled difference is added to the third one whence we obtain the donor vector $\vec{V}_{i,G}$. We can express the process as

$$\vec{V}_{i,G} = \vec{X}_{r1^i,G} + F\,(\vec{X}_{r2^i,G} - \vec{X}_{r3^i,G}) \quad (3)$$

V. CROSSOVER

To enhance the potential diversity of the population, a crossover operation comes into play after generating the donor vector through mutation. The donor vector exchanges its components with the target $\vec{X}_{i,G}$ under this operation to form the trial vector $\vec{U}_{i,G} = [u_{1,i,G}, u_{2,i,G}, u_{3,i,G}, ..., u_{D,i,G}]$. The DE family of algorithms can use two kinds of crossover methods—exponential (or two-point modulo) and binomial (or uniform)[8]. In exponential crossover, we first choose an integer n randomly among the numbers [1,D]. This integer acts as a starting point in the target vector, from where the crossover or exchange of components with the donor vector starts. We also choose another integer L from the interval [1,D]. L denotes the number of components the donor vector actually contributes to the target vector. After choosing n and L the trial vector is obtained as

$$u_{j,i} = v_{j,i,G} \quad \text{for } j = \langle n \rangle_D \;\; \langle n+1 \rangle_D, \ldots, \langle n+L-1 \rangle_D \quad (4)$$
$$x_{j,i,G} \text{ for all other } j \in [1,D]$$

VI. SELECTION

To keep the population size constant over subsequent generations, the next step of the algorithm calls for selection to determine whether the target or the trial vector survives to the next generation, i.e., at G = G + 1. The selection operation is described as

$$\vec{X}_{i,G+1} = \vec{U}_{i,G} \quad \text{if } f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G}) \quad (5)$$
$$= \vec{X}_{i,G} \quad \text{if } f(\vec{U}_{i,G}) > f(\vec{X}_{i,G})$$

where f ($\vec{X}$) is the objective function to be minimized. Therefore, if the new trial vector yields an equal or lower value of the objective function, it replaces the corresponding target vector in the next generation; otherwise the target is retained in the population. Hence, the population either gets better (with respect to the minimization of the objective function) or remains the same in fitness status, but never deteriorates. Note that in (5) the target vector is replaced by the trial vector even if both yield the same value of the objective function—a feature that enables DE-vectors to move over flat fitness landscapes with generations. Note that throughout this paper, we shall use the terms objective function value and fitness interchangeably. But, always for minimization problems, a lower objective function value will correspond to higher fitness.

VII. EXPERIMENTAL COMPARISON OF DIFFERENT MUTATIVE PROCESSES

In a recently published article [9], Neri and Tirronen reviewed a number of DE-variants for single-objective optimization problems and also made an experimental comparison of these variants on a set of numerical benchmarks. Actually it is the

process of mutation that demarcates one DE scheme from another. The mutation scheme in (3) uses a randomly selected $\vec{X}_{r1}$ and only one weighted difference vector $F .(\vec{X}_{r2} - \vec{X}_{r3})$ to perturb it. Hence, in literature, the particular mutation scheme given by is referred to as DE/rand/1. When used in conjunction with binomial crossover, the procedure is called DE/rand/1/bin. We can now have an idea of how the different DE schemes are named. The general convention used above is DE/x/y/z, where DE stands for "differential evolution," x represents a string denoting the base vector to be perturbed, y is the number of difference vectors considered for perturbation of x, and z stands for the type of crossover being used (exp:exponential;bin: binomial). The other four different mutation schemes, suggested by Storn and Price[10],[11] are summarized as

DE/best/1:

$$\vec{V}\, i,G = \vec{X}_{best,G} + F \cdot (\vec{X}_{r1^i,G} - \vec{X}_{r2^i,G}) \quad (6)$$

DE/target-to-best/1:

$$\vec{V}\, i,G = \vec{X}_{i,G} + F \cdot (\vec{X}_{best,G} - \vec{X}_{i,G}) + F \cdot (\vec{X}_{r1^i,G} - \vec{X}_{r2^i,G}) \quad (7)$$

DE/best/2:

$$\vec{V}\, i,G = \vec{X}_{best,G} + F \cdot (\vec{X}_{r1^i,G} - \vec{X}_{r2^i,G}) + F \cdot (\vec{X}_{r3^i,G} - \vec{X}_{r4^i,G}) \quad (8)$$

DE/rand/2:

$$\vec{V}\, i,G = \vec{X}_{r1^i,G} + F \cdot (\vec{X}_{r2^i,G} - \vec{X}_{r3^i,G}) + F \cdot (\vec{X}_{r4^i,G} - \vec{X}_{r5^i,G}) \quad (9)$$

The indices ri1, ri2, ri3, ri4, and ri5 are mutually exclusive integers randomly chosen from the range [1, NP]
The comparison is made on different test functions of CEC2005, CEC2008, CEC2010[12],[13],14]

Table1. Standard Table (Test functions of CEC2005,CEC2008,CEC2010)

Test Functions	Dimension	search domain	minima
sphere	100	[-5.12,5.12]^D	0
Rosenbrock	100	[-5,10]^D	0
Griewank	100	[-600, 600]^D	0
Rastrigin	100	[-5.12,5.12]^D	0
Michalewicz	100	[0,3.14]^D	at n=2, f(x)=-1.8013. at n=5, f(x*) = --4.687658. at n=10, f(x*) = -9.66015.
Schwefel	100	[-500,500]^D	0
Dixon and Price	100	[-10,10]^D	0
Rotated_Rosenbrock	100	[-5,10]^D	0
Rotated_Discus	100	[-5.12,5.12]^D	0

Experimental Results: We applied different process of mutation of DE Family of Storn and Price to a set of benchmark optimization problems and get a comparative table. The initial population was generated uniformly at random in the range, as specified in Table I.
Throughout this paper, we have used F=0.6, CR=0.4, Population Size=40.

Table 2. Experimental results, averaged over 30 independent runs.

			f(min)			
Test Functions	**Dimension**	**search domain**	**/*DE/rand/2*/**	**/*DE/best/1*/**	**/*DE/best/2*/**	**/*DE/target_to_best/1*/**
sphere	100	[-5.12,5.12]^D	0	0.003	0.0033145	0
Rosenbrock	100	[-5,10]^D	0.006336	0.04	27.22132	0
Griewank	100	[-600, 600]^D	0	0.000197	0.000885	0.000004
Rastrigin	100	[-5.12,5.12]^D	0.000003	0.99	1.015953	0
Michalewicz	100	[0,3.14]^D	0.6	19.1482	35.36521	9.12314
Schwefel	100	[-500,500]^D	50	57.3214	-2.11224	-3.926627
Dixon and Price	100	[-10,10]^D	0.000078	0.34	0.07	0
Rotated_Rosenbrock	100	[-5,10]^D	0.000793	1.783498	0.089783	0
Rotated_Discus	100	[-5.12,5.12]^D	0	0.00056	0.008	0

From the above comparative table, we can conclude, that sphere test function works best in mutation process DE/rand/2 and DE/target_to_best/1.

Rosenbrock test function works best in mutation process DE/target_to_best/1.

Griewank test function works best in mutation process DE/rand/2.

Rastrigin test function works best in mutation process DE/target_to_best/1.

Michalewicz test function is not a very good option for DE, though it gives more or less optimized result in mutation process DE/rand/2.

Schwefel test function is not a very good option for DE.

Dixon and Price test function works best in mutation process DE/target_to_best/1.

Rotated_Rosenbrock test function works best in mutation process DE/target_to_best/1.

Rotated_Discus test function works best in mutation process DE/rand/2 and DE/target_to_best/1.

Thus we can conclude that every benchmark function does not give the most optimized result on every mutation processes. Each test function has its own properties that when applied on proper mutation process gives the best result. Thus without running over here and there we can select the proper process of mutation which the above comparative table can provide.

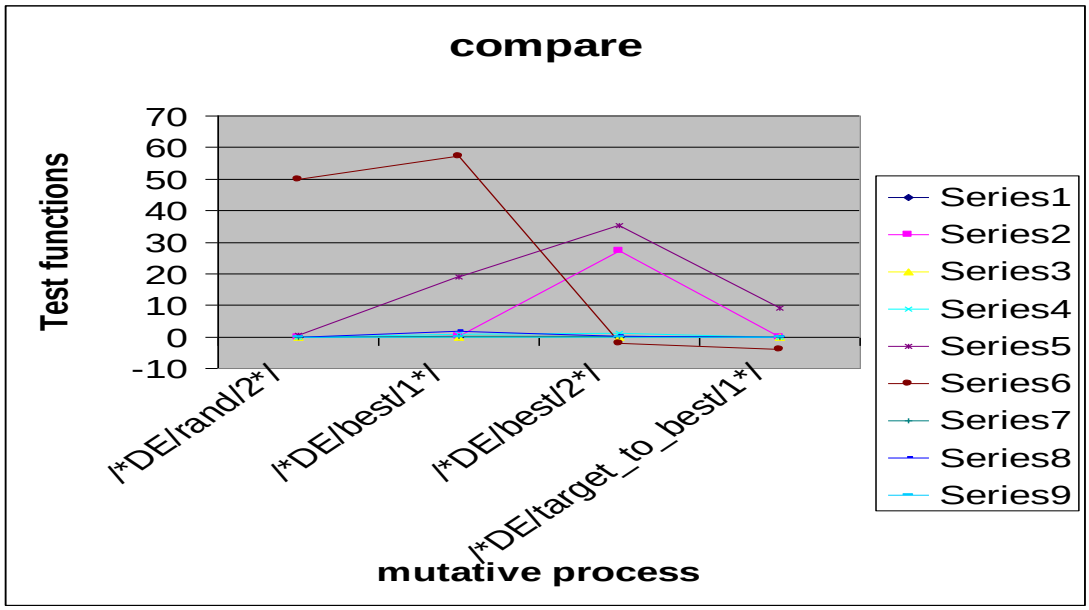

Fig. 1. Graphical Representation of the comparative study of different mutative processes

index		
		S
sphere	eries 1	
		S
Rosenbrock	eries 2	
		S
Griewank	eries 3	
		S
Rastrigin	eries 4	
		S
Michalewicz	eries 5	
		S
Schwefel	eries 6	
		S
Dixon and Price	eries 7	
		S
Rotated_Rosenbrock	eries 8	
		S
Rotated_Discus	eries 9	

VIII. ALGORITHM 1. PSEUDO-CODE FOR THE DE ALGORITHM WITH BINOMIAL CROSSOVER

Step 1: Read values of the control parameters of DE: scale factor F, crossover rate Cr, and the population size NP from user.

Step 2: Set the generation number $G = 0$ and randomly initialize a population of NP individuals $PG = \{ \vec{X}_{1,G} ....., \vec{X}_{NP,G} \}$ with $\vec{X}_{i,G} = [x_{1,i,G},, x_{2,i,G}, x_{3,i,G}, ..., x_{D,i,G}]$ and each individual uniformly distributed in the range $[\vec{X}_{min} - \vec{X}_{max}]$, where $\vec{X}_{min} = \{ x_{1,min},, x_{2,min}, x_{3,min}, ..., x_{D,min}\}$ and $\vec{X}_{max} = \{ x_{1,max},, x_{2,max}, x_{3,max}, ..., x_{D,max}\}$ with $i = [1, 2, ...,NP]$.

Step 3. WHILE the stopping criterion is not satisfied DO

FOR $i = 1$ to NP //do for each individual sequentially

Step 2.1 *Mutation Step*

Generate a donor vector $\vec{X}_{i,G} = \{ v_{1,i,G}, ..., \}$ $\{ v_{D,i,G} \}$ corresponding to the ith target vector $\vec{X}_{i,G}$ via the differential mutation scheme of DE as:

$$\vec{V}\,i,G = \vec{X}_{best,G} + F \cdot (\vec{X}_{r1^{i},G} - \vec{X}_{r2^{i},G})$$

Or

$$\vec{V}\,i,G = \vec{X}_{i,G} + F \cdot (\vec{X}_{best,G} - \vec{X}_{i,G}) + F \cdot (\vec{X}_{r1^{i},G} - \vec{X}_{r2^{i},G})$$

Or

$$\vec{V}\,i,G = \vec{X}_{best,G} + F \cdot (\vec{X}_{r1^{i},G} - \vec{X}_{r2^{i},G}) + F \cdot (\vec{X}_{r3^{i},G} - \vec{X}_{r4^{i},G})$$

Or

$$\vec{V}\,i,G = \vec{X}_{r1^{i},G} + F \cdot (\vec{X}_{r2^{i},G} - \vec{X}_{r3^{i},G}) + F \cdot (\vec{X}_{r4^{i},G} - \vec{X}_{r5^{i},G})$$

Step 2.2 *Crossover Step*

Generate a trial vector $\vec{U}_{i,G} = \{u_{1,i,G,}.....u_{D,i,G}\}$
for the *i*th target vector $\vec{X}_{i,G}$ through
binomial crossover in the following way:
$u_{j,i,G} = v_{j,i,G}$, if (*randi,j*[0, 1] ≤ *Cr* or *j* = *jrand*)
$= x_{j,i,G}$, otherwise,
Step 2.3 *Selection Step*
Evaluate the trial vector $\vec{U}_{i,G}$
IF $f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G})$, THEN $\vec{X}_{i,G+1} = \vec{U}_{i,G}$
ELSE $\vec{X}_{i,G+1} = \vec{X}_{i,G}$.
END IF
END FOR
Step 2.4 Increase the Generation Count
G = *G* + 1
END WHILE

IX. CONCLUSION

This paper attempted to provide an overall picture of the comparative study on mutation research on and with DE. Starting with a comprehensive introduction to the basic steps of the DE algorithm, it discussed the different schemes of parameter control and adaptation for DE and then overviewed several promising variants of the conventional DE. Next it provided an extensive review of the modifications of DE for tackling constrained, multi-objective, uncertain, and large-scale optimization problems.

REFERENCES

[1] Das and Suganthan: Differential Evolution: A Survey of the State-of-the-Art , IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 15, NO. 1, FEBRUARY 2011

[2] K. V. Price, "Differential evolution vs. the functions of the 2nd ICEO," in Proc. IEEE Int. Conf. Evol. Comput., Apr. 1997, pp. 153–157.

[3] K. V. Price and R. Storn, "Differential evolution: A simple evolution strategy for fast optimization," Dr. Dobb's J., vol. 22, no. 4, pp. 18–24, 1997.

[4] R. Storn and K. V. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization overcontinuousspaces,"ICSI,USA,Tech.Rep.TR950121995[Online].Available:http://icsi.berkeley.edu/~storn/litera.html

[5] 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., 1996, pp. 842–844.

[6] R. Storn, "On the usage of differential evolution for function optimization," in Proc. North Am. Fuzzy Inform. Process. Soc., 1996, pp. 519–523.

[7] R. Storn and K. Price, "Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces," J. Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.

[8] K. Price, R. Storn, and J. Lampinen, Differential Evolution—A Practical Approach to Global Optimization. Berlin, Germany: Springer, 2005.

[9] F. Neri and V. Tirronen, "Recent advances in differential evolution: A review and experimental analysis," Artif. Intell. Rev., vol. 33, no. 1, pp. 61–106, Feb. 2010.

[10] K. Price, R. Storn, and J. Lampinen, Differential Evolution—A Practical Approach to Global Optimization. Berlin, Germany: Springer, 2005.

[11] K. V. Price, "An introduction to differential evolution," in New Ideas in Optimization, D. Corne, M. Dorigo, and V. Glover, Eds. London, U.K.: McGraw-Hill, 1999, pp. 79–108.

[12] Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb3, Y. -P. Chen, A. Auger, S. Tiwari, Technical Report, Nanyang Technological University, Singapore And KanGAL Report Number 2005005 (Kanpur Genetic Algorithms Laboratory, IIT Kanpur) May 2005

[13] Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization K. Tang1, X. Yao1, 2, P. N. Suganthan, C. MacNish, Y. P. Chen, C. M. Chen, Z. Yang

[14] Benchmark Functions for the CEC'2010 Special Session and Competition on Large-Scale Global Optimization Ke Tang , Xiaodong Li , P. N. Suganthan , Zhenyu Yang , and Thomas Weise

