



Mini Review

Volume 3 Issue 2 - June 2018
DOI: 10.19080/RAEJ.2018.03.555607

Robot Autom Eng J

Copyright © All rights are reserved by Ali Wagdy Mohamed

Control Parameters in Differential Evolution (DE): A Short Review



Ali Khater Mohamed¹, Ali Wagdy Mohamed^{2*}

¹Department of Business Administration, College of sciences and humanities, Majmaah University, Saudi Arabia

²Department Operations Research, Institute of Statistical Studies and Research, Cairo University, Egypt

Submission: April 19, 2018; **Published:** June 05, 2018

***Corresponding author:** Ali Wagdy Mohamed, Operations Research Department, Institute of Statistical Studies and Research, Cairo University, Egypt, Tel: +966-556269723; Email: aliwagdy@gmail.com

Abstract

Differential Evolution (DE) is a population based stochastic search algorithm for optimization. DE has three main control parameters, Crossover (cr), Mutation factor (F) and Population size (NP). These control parameters play a vital and crucial rule in improving the performance of search process in DE. This paper introduces a brief review for control parameters in Differential evolution (DE).

Keywords: Differential evolution; Population size; Global optimization; Control parameters

Abbreviations: DE: Differential Evolution; Cr: Crossover; NP: Population Size; F: Mutation Factor

Introduction

Differential Evolution (DE) is a population-based heuristic algorithm proposed by Storn & Price [1] to solve global optimization problems with different characteristics over continuous space. Despite its simplicity, it proved a great performance in solving non-differentiable, non-continuous and multi-modal optimization problems [2]. DE has three main control parameters which are the crossover (CR), mutation factor (F) and population size (NP). The values of the control parameters affect significantly on the performance of DE. Therefore, the tuning of those control parameters is considered a challenging task. DE has a great performance in exploring the solution space and this is considered as the main advantage, on the other side, an obvious weak point is its poor performance in exploitation phase which may cause a stagnation and/or premature convergence.

The next section introduces differential evolution. Section 3 introduces a short review for control parameters in DE. And finally, the paper is concluded in section 4.

Differential Evolution

In simple DE, DE/rand/1/bin [1,2], an initial population of NP individuals $\vec{x}_j, j=1, 2, \dots, NP$, is generated at random according to a uniform distribution within lower and upper boundaries (x_j^L, x_j^U) . Individuals are evolved by the means of crossover and mutation to generate a trial vector. The trial vector competes with his parent in order to select the fittest to the next generation. The steps of DE are:

Initialization of a population

Initial population in DE, as the starting point for the process of optimization, is created by assigning a random chosen value for each decision variable in every vector, as indicated in equation (1).

$$x_{ij}^0 = L_j + rand_j * (U_j - L_j) \quad (1)$$

Where L_j, U_j : the lower and upper boundaries for x_j , $rand_j$: a random number uniform [0, 1].

Mutation

A mutant vector v_i^{G+1} is generated for each target vector x_i^G at generation G according to equation (2)

$$v_i^{G+1} = x_{r_1}^G + F * (x_{r_2}^G - x_{r_3}^G), \quad r_1 \neq r_2 \neq r_3 \neq i \quad (2)$$

Where r_1, r_2, r_3 are randomly chosen from the population. The mutation factor $F \in [0, 2]$. A new value for the component of mutant vector is generated using (1) if it violates the boundary constraints.

Recombination (crossover)

Crossover is the process of swapping information between the target and the mutated individuals using (3), to yield the trial vector u_i^{G+1} . Two types of crossover can be used, binomial crossover or exponential crossover.

$$u_{ij}^{G+1} = \begin{cases} v_{ij}^{G+1}, & \text{rand}(j) \leq CR \text{ or } j = \text{rand}(i), \\ x_{ij}^G, & \text{rand}(j) > CR \text{ and } j \neq \text{rand}(i), \end{cases} \quad (3)$$

where $j=1, 2, \dots, D$, $\text{rand}(j) \in [0,1]$ is the j th evaluation of a uniform random number. Crossover rate (CR) is between 0 and 1, r and (i) is a random index between 1 and D to ensures that u_i^{G+1} gets at least one element from v_i^{G+1} ; otherwise, the population remains without change.

In the exponential crossover, a starting index l and a number of components w are chosen randomly

from the ranges $\{l, D\}$ and $\{l, D-1\}$ respectively. The values of variables in locations l to $l+w$ from v_i^{G+1} and the remaining locations from the x_i^G are used to produce the trial vector u_i^{G+1} .

Selection

Greedy scheme for fast convergence of DE. The child u_i^{G+1} is compared with its parent x_i^G to select the better for the next generation according to the selection scheme in equation (4).

$$x_i^{G+1} = \begin{cases} u_i^{G+1}, & f(u_i^{G+1}) \leq f(x_i^G) \\ x_i^G, & \text{otherwise} \end{cases} \quad (4)$$

A detailed description of standard DE algorithm is given in Figure 1.

```

01. Begin
02. G=0
03. Create a random initial population  $\bar{x}_i^G \forall i, i=1, \dots, NP$ 
04. Evaluate  $f(\bar{x}_i^G) \forall i, i=1, \dots, NP$ 
05. For G=1 to Gmax Do
06. For i=1 to NP Do
07.     Select randomly  $r1 \neq r2 \neq r3 \neq i \in [1, NP]$ 
08.      $j_{rand} = \text{randint}(1, D)$ 
09. For j=1 to D Do
10.     If  $(\text{rand}, i, [0,1] < CR \text{ or } j = j_{rand})$  Then
11.          $u_{i,j}^G = x_{r1,j}^G + F \cdot (x_{r2,j}^G - x_{r3,j}^G)$ 
12.     Else
13.          $u_{i,j}^G = x_{i,j}^G$ 
14.     End If
15. End For
16. If  $(f(u_i^G) \leq f(\bar{x}_i^G))$  Then
17.      $\bar{x}_i^{G+1} = u_i^G$ 
18. Else
19.      $\bar{x}_i^{G+1} = \bar{x}_i^G$ 
20. End If
21. End For
22. G=G+1
23. End For
24. End

```

Figure 1: Pseudocode of standard DE algorithm.

Short Review

During the last two decades, the problem of finding the balance between the exploration and exploitation has attracted many researchers in order to improve the performance of DE by developing new mutation strategies or hybridizing promising mutation strategies.

Das et al. [3] proposed an improved variant of DE/target-to-best/1/bin based on the concept of population members' neighborhood. Zhang & Sanderson [4] proposed a new mutation strategy "DE/current-to-pbest" with an optional external archive that utilizes the historical data in order to progress towards the promising direction and called it JADE. Qin, Huang & Suganthan [5] proposed SaDE, in which a self-adaptive mechanism for trial

vector generation is presented, that is based on the idea of learning from the past experience in generating promising solutions. Mohamed et al. [6-8] proposed a novel mutation strategy which is based on the weighted difference between the best and the worst individual during a specific generation, the new mutation strategy is combined with the basic mutation DE/rand/1/bin with equal probability for selecting each of them. Li & Yin [9] used two mutation strategies based on the best and random vectors. Mohamed [10] proposed IDE, in which new triangular mutation rule that selects three random vectors and adding the difference vector between the best and worst to the better vector. The new mutation rule is combined with the basic mutation rule through a non-linear decreasing probability rule. And a restart mechanism to avoid the premature convergence is presented. Recently,

triangular mutation has been also used to solve IEEE CEC 2013 unconstrained problems [11], constrained non-linear integer and mixed-integer global optimization problems [12], IEEE CEC2006 constrained optimization problems [13], CEC 2010 large-scale optimization problems [14], and stochastic programming problems [15].

Extensive research was presented for controlling the parameters, as control parameters play a vital role in the evolution process. Brest et al. [16] presented a new self-adaptive technique for controlling the parameters. Noman & Iba [17] proposed an adaptive crossover based on local search and the length of the search was adjusted using hill-climbing. Peng et al. [18] proposed rJADE, in which a weighting strategy is added to JADE, with a "restart with knowledge transfer" method in order to benefit from the knowledge obtained from the previous failure. Montgomery & Chen [19] presented a complete analysis of how much the evolution process affected by the value of CR. Mallipeddi et al. [20] proposed a pool of values for each control parameter to select the appropriate value during the evolution process. Wang, Cai & Zhang [21] proposed a new method that randomly chooses from a pool that contains three strategies in order to generate the trial vector and three control parameter settings, they called it CoDE. Yong et al. [22] presented CoBiDE, in which a covariance matrix learning for the crossover operator and a bimodal distribution parameter to control the parameters are introduced. Draa, Bouzoubia & Boukhalfa [23] introduced a new sinusoidal formula in order to adjust the values of crossover and the scaling factor, they called it SinDE. A complete review could be found in [24,25].

DE mechanism depends on selecting three random individuals from the population to perform the mutation process. Therefore, the population size must be greater than the selected vectors. Large population size increases the diversity but consumes more resources (function calls), while small population size may cause stagnation or tripping in local optima. Thus, the choosing of the population size is considered a very critical aspect. From the literature, it has been found that researchers choose the population size in four different ways.

- i. Choosing the population size for each problem separately based on the experience or previous knowledge and keep it constant during all runs [26,27].
- ii. Relate the population size to the problem dimensionality [6,28,29].
- iii. Setting the population size fixed during all runs and independent of the dimension of the problems [22,30].
- iv. Allowing the population size to vary during the runs using adaptation rule [31-33]. A complete review of population size could be found in [34,35].

Conclusion

Control parameters plays a vital rule in the evolution process of the DE. Over the last decades, many EAs have been proposed

to solve optimization problems. However, all these algorithms including DE have the same shortcomings in solving optimization problems. One of them is the choice of the control parameters which are difficult to adjust for different problems with different characteristics. This paper introduced a brief review for a considerable number of research studies that have been proposed to enhance the performance of DE.

References

1. Storn R, Price K (1995) Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces. International Computer Science Institute Technical Report, Tech Rep TR-95-012.
2. Storn R, Price K (1997) Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J Global Optimization* 11(4): 341-359.
3. Das S, Abraham A, Chakraborty UK, Konar A (2009) Differential evolution using a neighborhood-based mutation operator. *IEEE Transactions on Evolutionary Computation* 13(3): 526-553.
4. Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation* 13(5): 945-958.
5. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation* 13(2): 398-417.
6. Mohamed AW, Sabry HZ (2012) Constrained optimization based on modified differential evolution algorithm. *Information Sciences* 194: 171-208.
7. Mohamed AW, Sabry HZ, Khorshid M (2012) An alternative differential evolution algorithm for global optimization. *J Adv Res* 3(2): 149-165.
8. Mohamed AW, Sabry HZ, Farhat A (2011) Advanced differential evolution algorithm for global numerical optimization. In: *Proceedings of the IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE'11)*, Penang, Malaysia, pp156-161.
9. Li X, Yin M (2014) Modified differential evolution with self-adaptive parameters method. *J Comb Optim* 31(2): 546-576.
10. Mohamed AW (2015) An improved differential evolution algorithm with triangular mutation for global numerical optimization. *Computers & Industrial Engineering* 85: 359-375.
11. Mohamed AW, Suganthan PN (2017) Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation. *Soft Comput* 22(10): 3215-3235.
12. Mohamed AW (2017) An efficient modified differential evolution algorithm for solving constrained non-linear integer and mixed-integer global optimization problems. *International Journal of Machine Learning and Cybernetics* 8(3): 989-1007.
13. Mohamed AW (2017) A novel differential evolution algorithm for solving constrained engineering optimization problems. *Journal of Intelligent Manufacturing* 29(3): 659-692.
14. Mohamed AW, Almazyad AS (2017) Differential Evolution with Novel Mutation and Adaptive Crossover Strategies for Solving Large Scale Global Optimization Problems. *Applied Computational Intelligence and Soft Computing* p. 18.
15. Mohamed AW (2017) Solving stochastic programming problems using new approach to Differential Evolution algorithm. *Egyptian Informatics Journal* 18(2): 75-86.
16. Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-Adapting Control Parameters in Differential Evolution: A Comparative

- Study on Numerical Benchmark Problems. IEEE Tran. E vol. Comput 10(6): 646-657.
17. Noman N, Iba H (2008) Accelerating differential evolution using an adaptive local search. IEEE Trans Evol Comput 12(1): 107-25.
 18. Peng F, Tang K, Chen G, Yao X (2009) Multi-start JADE with knowledge transfer for numerical optimization. In IEEE CEC, pp. 1889-1895.
 19. Montgomery J, Chen S (2010) An analysis of the operation of differential evolution at high and low crossover rates. IEEE Congress on Evolutionary Computation, Barcelona p. 1-8.
 20. Mallipeddi R, Suganthan PN, Pan QK, Tasgetiren MF (2011) Differential evolution algorithm with ensemble of parameters and mutation Strategies. Applied Soft Computing 11(2): 1679-1696.
 21. Wang Y, Cai Z, Zhang Q (2011) Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters. IEEE Tran Evol Comput 15(1): 55-66.
 22. Yong W, Han-Xiong L, Tingwen H, Long L (2014) Differential evolution based on covariance matrix learning and bimodal distribution parameter setting. Applied Soft Computing 18: 232-247.
 23. Draa A, Bouzoubia S, Boukhalfa I (2015) A sinusoidal differential evolution algorithm for numerical optimization. Appl Soft Comput 27: 99-126.
 24. Das S, Suganthan PN (2011) Differential evolution: A survey of the state-of-the-art. IEEE transactions on evolutionary computation 15(1): 4-31.
 25. Das S, Mullick SS, Suganthan PN (2016) Recent advances in differential evolution-an updated survey. Swarm and Evolutionary Computation 27: 1-30.
 26. Cheng JX, Zhang GX, Neri F (2013) Enhancing distributed Differential Evolution with multicultural migration for global numerical optimization. Inf Sci 247: 72-93.
 27. Gao WF, Pan Z, Gao J (2014) A new highly efficient Differential Evolution with self-adaptive strategy for multimodal optimization. IEEE Trans Cybern 44 (8): 1314-1327.
 28. Mallipeddi R, Suganthan PN (2008) Empirical study on the effect of population size on Differential Evolution algorithm. In: Proceedings of IEEE congress on Evolutionary Computation, Hong Kong, USA.
 29. Wang H, Wang WJ, Cui ZH, Sun H, Ranhnmayan S (2014) Heterogeneous Differential Evolution for numerical optimization. Scientific World Journal 318063.
 30. Gao WF, Yen GG, Liu SY (2015) A dual Differential Evolution with coevolution for constrained optimization. IEEE Trans Cybern 45(5): 1094-1107.
 31. Brest J, Maucec MS (2011) Self-adaptive Differential Evolution algorithm using population size reduction and three strategies. Soft Comput 15(11): 2157-2174.
 32. Zamuda A, Brest J (2015) Self-adaptive control parameters' randomization frequency and propagations in Differential Evolution. Swarm Evolut Comput 25: 72-99.
 33. Piotrowski AP (2017) Review of Differential Evolution population size. In Swarm and Evolutionary Computation 32: 1-24.
 34. Mohamed AW, Mohamed AK (2017) Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. Int J Mach Learn & Cyber p. 1-25.
 35. Mohamed AK, Mohamed AW, Elfeky EZ, Saleh M (2018) Enhancing AGDE Algorithm Using Population Size Reduction for Global Numerical Optimization. In: Hassanien A, Tolba M, Elhoseny M, Mostafa M (Eds.), The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018). AMLTA 2018. Advances in Intelligent Systems and Computing, vol 723.



This work is licensed under Creative Commons Attribution 4.0 License
DOI: [10.19080/RAEJ.2018.03.555607](https://doi.org/10.19080/RAEJ.2018.03.555607)

Your next submission with Juniper Publishers will reach you the below assets

- Quality Editorial service
- Swift Peer Review
- Reprints availability
- E-prints Service
- Manuscript Podcast for convenient understanding
- Global attainment for your research
- Manuscript accessibility in different formats
(Pdf, E-pub, Full Text, Audio)
- Unceasing customer service

Track the below URL for one-step submission

<https://juniperpublishers.com/online-submission.php>