Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover Strategies

Rammohan Mallipeddi and Ponnuthurai Nagaratnam Suganthan,

Nanyang Technological university, Singapore, 639798.
{mall0004, epnsugan}@ntu.edu.sg

Abstract. Differential Evolution (DE) has attracted much attention recently as an effective approach for solving numerical optimization problems. However, the performance of DE is sensitive to the choice of the mutation and crossover strategies and their associated control parameters. Thus, to obtain optimal performance, time consuming parameter tuning is necessary. Different mutation and crossover strategies with different parameter settings can be appropriate during different stages of the evolution. In this paper, we propose a DE with an ensemble of mutation and crossover strategies and their associated control parameters known as EPSDE. In EPSDE, a pool of distinct mutation and crossover strategies along with a pool of values for each control parameter coexists throughout the evolution process and competes to produce offspring. The performance of EPSDE is evaluated on a set of 25 bound-constrained problems designed for Conference on Evolutionary Computation (CEC) 2005 and is compared with state-of-the-art algorithm.

Keywords: Differential Evolution, Global optimization, Parameter adaptation, Ensemble, Mutation strategy adaptation.

1 Introduction

DE proposed by Storn and Price [1] is a fast and simple technique which performs well on a wide variety of problems. DE is a population based stochastic search technique, which is inherently parallel. DE has been successfully applied in diverse fields of engineering [2, 3, 4, 5, 6, 7, 8, 9, 10]. The performance [11, 12] of the DE algorithm is sensitive to the mutation strategy, crossover strategy and control parameters such as the population size (*NP*), crossover rate (*CR*) and the scale factor (*F*). The best settings for the control parameters can be different for different optimization problems and the same functions with different requirements for consumption time and accuracy. Therefore, to successfully solve a specific optimization problem, it is generally necessary to perform a time-consuming trial-and-error search for the most appropriate combination of strategies and their associated parameter values. However, such a trial-and-error search process suffers from high computational costs. The population of DE may evolve through different regions in the search space, within which different strategies [13] with different parameter settings may be more effective than others. Although different partial adaptation schemes have been proposed [14, 15, 13, 16, 17] to overcome the time consuming trial-and-error procedure, we demonstrate the superior performance of the ensemble strategy proposed [18].

The reminder of this paper is organized as follows: Section 2 presents a literature survey on different mutation and crossover strategies and parameter settings used in DE. Section 3 presents the proposed ensemble of mutation and crossover strategies and parameters in DE (EPSDE) algorithm. Section 4 presents the experimental results and discussions while Section 5 concludes the paper.

2 Literature Review

Differential Evolution (DE) algorithm is a floating-point encoded evolutionary algorithm for global optimization over continuous spaces [19]. The performance of DE becomes more sensitive to the strategy and the associated parameter values when the problem is complex [20]. Inappropriate choice of mutation and crossover strategies and the associated parameters may lead to premature convergence, stagnation or wastage of computational resources [21, 20, 22, 23, 16] . Initially it was thought that [19, 24] the control parameters of DE are not difficult to choose. But due to the complex interaction of control parameters with the DE's performance on hard optimization problems [14], choosing an appropriate mutation strategy and control parameters require some expertise. Since DE was proposed various empirical guidelines were suggested for choosing a mutation strategy and its associated control parameter settings.

In DE, the larger the population size (*NP*), the higher the probability of finding a global optimum. But, a larger population implies a slower convergence rate requiring a larger number of function evaluations. Initially $NP=10D$ was considered as a good choice [4] for DE to find a global optimum. However, to balance the speed and reliability different ranges of *NP* values such as $5D$ to $10D$ [19], $3D$ to $8D$ [20] and $2D$ to $40D$ [25] were suggested.

The classical DE proposed by Price and Storn uses DE/rand/1/bin, which is most widely used. Later, various DE mutation strategies were proposed [26, 4]. In [20, 19] it was stated that a 2 difference vector strategies such as DE/rand/2/bin are better than DE/rand/1/bin due to their ability to improve diversity by producing more trial vectors [22]. DE/best/1/bin and DE/rand-to-best/1/bin are faster on easier optimization problems, but become unreliable when solving highly multi-modal problems. To balance the exploration and exploitation abilities of DE, and adaptive mutation strategy with optional external archive (JADE) was proposed [27]. DE/current-to-rand/1 being a rotation-invariant strategy can solve rotated problems better than other strategies [28].

The main difference between binomial and exponential crossover is the fact that while in the binomial case the components inherited from the mutant vector are arbitrarily selected in the case of exponential crossover they form one or two compact subsequences [29].

The crossover rate (*CR*) controls which and how many components are mutated in each element of the current population. The crossover rate *CR* is a probability $0 \leq CR \leq 1$ of mixing between trial and target vectors. A large *CR* speeds up convergence [20, 4, 19]. In [19], it is said that $CR = 0.1$ is a good initial choice while $CR = 0.9$ or 1.0 can be tried to increase the convergence speed. For separable problems, CR from the range (0, 0.2) is the best while for multi-modal, parameter dependant problems CR in the range (0.9,1.0) is best [23, 25]. Based on these observations, CR with non-continuous ranges are also used [30].

Scaling factor, F is usually chosen in [0.5, 1] [4]. In [19], it is said that values of F smaller than 0.4 and greater than 1.0 are occasionally effective. The scale factor F, is strictly greater than zero. A larger F increases the probability of escaping from a local optimum [20, 25]. F must be above a certain critical value to avoid premature convergence to a sub-optimal solution [20, 25], but if F becomes too large, the number of function evaluations to find the optimum grows very quickly. In [20, 19], it is said that $F = 0.6$ or 0.5 would be a good initial choice while in [25] it is said that $F = 0.9$ would be a good initial choice. Typical values of F are 0.4 to 0.95 according to [25].

Even though the above guidelines are useful for choosing the individual parameters of DE to some extent, the performance of DE is more sensitive to the combination of the mutation strategy and its associated parameters. To improve the performance of DE various adaptation techniques have been proposed [14, 26, 31, 32, 33, 16, 34, 13, 35, 36, 37, 38].

3 Ensemble of Mutation and Crossover Strategies and Parameters in DE (EPSDE)

The effectiveness of conventional DE in solving a numerical optimization problem depends on the selected mutation and crossover strategy and its associated parameter values. However, different optimization problems require different mutation strategies with different parameter values depending on the nature of problem (uni-modal and multi-modal) and available computation resources. In addition, to solve a specific problem, different mutation strategies with different parameter settings may be better during different stages of the evolution than a single mutation strategy with unique parameter settings as in the conventional DE. Motivated by these observations, we propose an ensemble of mutation and crossover strategies and parameter values for DE (EPSDE) in which a pool of mutation strategies, along with a pool of values corresponding to each associated parameter competes to produce successful offspring population. The candidate pool of mutation and mutation strategies and parameters should be restrictive to avoid the unfavorable influences of less effective mutation strategies and parameters [13]. The mutation strategies or the parameters present in a pool should have diverse characteristics, so that they can exhibit distinct performance characteristics during different stages of the evolution, when dealing with a particular problem.

EPSDE consists of a pool of mutation and crossover strategies along with a pool of values for each of the associated control parameters. Each member in the initial population is randomly assigned with a mutation strategy and associated parameter values taken from the respective pools. The population members (target vectors) produce offspring (trial vectors) using the assigned mutation strategy and parameter values. If the generated trial vector produced is better than the target vector, the mutation strategy and parameter values are retained with trial vector which becomes the parent (target vector) in the next generation. The combination of the mutation strategy and the parameter values that produced a better offspring than the parent are stored. If the target vector is better than the trial vector, then the target vector is randomly reinitialized with a new mutation strategy and associated parameter values from the respective pools or from the successful combinations stored with equal probability. This leads to an increased probability of production of offspring by the better combination of mutation strategy and the associated control parameters in the future generations.

In this paper the EPSDE formed uses the following mutation and crossover strategies and parameter values:

1) Mutation strategies: JADE [27] and DE/current-to-rand/1 [28]
2) Crossover strategies: Binomial crossover and exponential crossover [**39**]

In the proposed EPSDE, the population size (NP = 50) is maintained constant throughout the evolution process. $F \in \{0.5, 0.9\}$ and $CR \in \{0.1, 0.5, 0.9\}$

Table 1: Ensemble of Parameters and Mutation and Crossover Strategies in DE (EPSDE)

Step 1 Set the generation number $G = 0$, and randomly initialize a population of NP individuals $P_G = \{X_{i,G}, ..., X_{NP,G}\}$ with $X_{i,G} = \{x_{i,G}^1, ..., x_{i,G}^D\}, i = 1, ..., NP$ uniformly distributed in the range [X_{min}, X_{max}], where $X_{min} = \{x_{min}^1, ..., x_{min}^D\}$ and $X_{max} = \{x_{max}^1, ..., x_{max}^D\}$

Step 2 Select a pool of mutation strategies and a pool of values for each associated parameters corresponding to each mutation strategy.

Step 3 Each population member is randomly assigned with one of the mutation strategy from the pool and the associated parameter values are chosen randomly from the corresponding pool of values.

Step 4 WHILE stopping criterion is not satisfied

DO

Step 4.1 ***Mutation step***

/*Generate a mutated vector $V_{i,G} = \{v_{i,G}^1, v_{i,G}^2, ..., v_{i,G}^D\}$ for each target vector $\mathbf{X}_{i,G}$ */

FOR i = 1 to NP

Generate a mutated vector $V_{i,G} = \{v_{i,G}^1, v_{i,G}^2, ..., v_{i,G}^D\}$ corresponding to the target vector $\mathbf{X}_{i,G}$ using the mutation strategy and F value associated with the target vector.

END FOR

Step 4.2 ***Crossover step***

/*Generate a trial vector $U_{i,G} = \{u_{i,G}^1, u_{i,G}^2, ..., u_{i,G}^D\}$ for each target vector $\mathbf{X}_{i,G}$ */

FOR i = 1 to NP

Generate a trial vector $U_{i,G} = \{u_{i,G}^1, u_{i,G}^2, ..., u_{i,G}^D\}$ corresponding to the target vector $\mathbf{X}_{i,G}$ using the crossover strategy and CR value associated with the target vector.

END FOR

Step 4.3 ***Selection step***

/* Selection by competition between target (parent) and trial (offspring) vectors */

FOR i =1 to NP

/* Evaluate the trial vector $U_{i,G}$ */

IF $f(U_{i,G}) \leq f(X_{i,G})$, THEN $X_{i,G+1} = U_{i,G}$, $f(X_{i,G+1}) \leq f(U_{i,G})$

IF $f(U_{i,G}) < f(X_{best,G})$, THEN $X_{best,G} = U_{i,G}$, $f(X_{best,G}) \leq f(U_{i,G})$

/* $X_{best,G}$ is the best individual in generation G */

ELSE $X_{i,G+1} = X_{i,G}$

END IF

END IF

END FOR

Step 4.4 ***Updating Step***

FOR i =1 to NP

IF $f(U_{i,G}) > f(X_{i,G})$ THEN

Randomly select a new mutation strategy and parameter values from the pools or from the stored successful combinations.

END IF

END FOR

Step 4.5 ***Increment the generation count*** $G = G + 1$

Step 5 END WHILE

4 Experimental Results

To evaluate the performance of the algorithms the test problems of CEC 2005 are used. The maximum numbers of function evaluations used are 100000 and 300000 for 10D and 30D problems respectively. The results of the algorithms are presented in Table II. In Table II, for a particular problem the results are highlighted if the performance of the algorithm is statistically significant.

Table 2. Results for 10D and 30D benchmark problems of CEC 2005

Fcn	10D				30D			
	JADE		EPSDE		JADE		EPSDE	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
f_1	0	0	0	0	0	0	0	0
f_2	0	0	0	0	8.59E-28	4.19E-28	3.37E-27	4.73E-27
f_3	1.11E-25	4.03E-26	6.96E-25	1.73E-26	**7.96E+03**	**3.88E+03**	7.74E+04	3.77E+04
f_4	0	0	0	0	2.45E-02	8.40E-02	**1.76E-12**	**2.97E-12**
f_5	2.42E-13	6.28E-13	**0**	**0**	7.53E+02	3.68E+02	**2.26E+02**	**2.61E+02**
f_6	8.13E-01	1.67E+00	**0**	**0**	1.03E+01	2.72E+01	**2.12E-20**	**1.13E-19**

f_7	**7.51E-03**	**6.61E-03**	3.94E-02	3.48E-02	1.56E-02	1.51E-02	**5.60E-03**	**6.11E-03**
f_8	2.02E+01	1.54E-01	2.04E+01	5.00E-03	2.08E+01	2.46E-01	2.08E+01	1.31E-01
f_9	0	0	0	0	0	0	0	0
f_{10}	4.39E+00	1.09E+00	**3.30E+00**	**9.59E-01**	**2.73E+01**	**5.70E+00**	4.71E+01	1.52E+01
f_{11}	4.42E+00	1.03E+00	**4.16E+00**	**3.21E+00**	2.68E+01	2.03E+00	2.86E+01	9.61E-01
f_{12}	8.89E+01	2.89E+02	**5.00E+00**	**7.07E+00**	4.92E+03	3.97E+03	1.32E+04	1.35E+04
f_{13}	**2.48E-01**	**5.43E-02**	4.13E-01	1.08E-01	1.67E+00	3.04E-02	**1.19E+00**	**1.24E-01**
f_{14}	**2.75E+00**	**3.00E-01**	3.08E+00	8.31E-02	1.24E+01	3.27E-01	1.25E+01	1.64E-01
f_{15}	1.16E+02	1.67E+02	**8.53E+01**	**1.49E+02**	3.20E+02	1.18E+02	**2.12E+02**	**1.98E+01**
f_{16}	1.21E+02	6.78E+00	**9.76E+01**	**4.40E+00**	1.45E+02	1.55E+02	**9.08E+01**	**2.98E+01**
f_{17}	1.31E+02	5.05E+00	**1.21E+02**	**6.28E+00**	1.34E+02	1.44E+02	**1.04E+02**	**7.27E+01**
f_{18}	7.55E+02	1.86E+02	**6.37E+02**	**3.03E+02**	9.05E+02	1.82E+00	**8.20E+02**	**3.35E+00**
f_{19}	7.40E+02	2.04E+02	**5.82E+02**	**3.11E+02**	8.98E+02	2.68E+01	**8.21E+02**	**3.35E+00**
f_{20}	7.18E+02	2.15E+02	**5.62E+02**	**2.96E+02**	9.05E+02	1.51E+00	**8.22E+02**	**4.17E+00**
f_{21}	5.30E+02	1.98E+02	**4.80E+02**	**6.10E+00**	5.10E+02	5.47E+01	**5.00E+02**	**6.64E-14**
f_{22}	**7.53E+02**	**1.94E+01**	7.70E+02	1.90E+01	8.85E+02	1.84E+01	8.85E+02	6.82E+01
f_{23}	2.15E+02	5.01E+01	2.13E+02	6.01E+01	5.50E+02	8.05E+01	**5.07E+02**	**7.26E+00**
f_{24}	2.00E+02	2.90E-14	2.00E+02	2.90E-14	**2.00E+02**	**2.90E-14**	2.13E+02	1.52E+00
f_{25}	5.22E+02	1.11E+02	**4.41E+02**	**2.05E+02**	2.11E+02	7.27E-01	2.13E+02	2.55E+00

From the results, it can be observed that EPSDE is significantly worse than, similar to and better than JADE in 4, 8 and 13 cases respectively in 10D problems. Similarly in 30D problems, EPSDE is significantly worse than, similar to and better than JADE in 3, 9 and 13 cases respectively.

5 Conclusions

The performance of DE depends on the selected mutation and crossover strategy and its associated parameter values. Different optimization problems require different mutation strategies with different parameter values depending on the nature of problem and available computation resources. For a problem at hand different mutation strategies with different parameter settings may be more effective during different stages of the evolution than a single mutation strategy with unique parameter settings as in the conventional DE. Based on these observations, we propose an ensemble of mutation and crossover strategies and parameter values in which a pool of mutation and crossover strategies, along with a pool of values corresponding to each associated parameter compete to produce offspring population. The performance of EPSDE is evaluated on set of benchmark problems and is favorably compared with the state-of-the-art DE methods in the literature.

References

1. R. Storn and K. Price, "Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces," Technical Report TR-95-012, ICSI, http://http.icsi.berkeley.edu/~storn/litera.html, 1995.
2. R. Joshi and A. C. Sanderson, Minimal representation multisensor fusion using differential evolution, IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans. **29** (1999), no. 1, 63-76.
3. T. Rogalsky, R. W. Derksen and S. Kocabiyik, Differential evolution in aerodynamic optimization, Proc. of 46th Annual Conference of Canadian Aeronautics and Space Institute, 1999, p.^pp. 29-36.
4. R. Storn, "On the usage of differential evolution for function optimization " Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS), IEEE, Berkeley, 1996, pp. 519-523.
5. M. K. Venu, R. Mallipeddi and P. N. Suganthan, Fiber bragg grating sensor array interrogation using differential evolution, Optoelectronics and Advanced Materials - Rapid Communications **2** (2008), no. 11, 682-685.
6. J. Ilonen, J. K. Kamarainen and J. Lampinen, Differential evolution training algorithm for feed-forward neural networks, Neural Processing Letters **17** (2003), no. 1, 93-105.
7. S. Das and A. Konar, Automatic image pixel clustering with an improved differential evolution Applied Soft Computing **9** (2009), no. 1, 226-236
8. U. Maulik and I. Saha, Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery, Pattern Recognition **42** (2009), 2135-2149.
9. R. Storn, "Differential evolution design of an iir-filter " IEEE International Conference on Evolutionary Computation IEEE, 1996, pp. 268-273.
10. M. Varadarajan and K. S. Swarup, Differential evolution approach for optimal reactive power dispatch, Applied Soft Computing **8** (2008), no. 4, 1549-1561.
11. J. Liu and J. Lampinen, On setting the control parameter of the differential evolution method, Proc. 8th Int, Conf. Soft Computing (MENDEL 2002), 2002, pp. 11-18.
12. S. Das and P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE Trans. on Evolutionary Computation.
13. A. K. Qin, V. L. Huang and P. N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Transactions on Evolutionary Computation **13** (2009), no. 2, 398-417.
14. J. Brest, S. Greiner, B. Boscovic, M. Mernik and V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, IEEE Transactions on Evolutionary Computation **10** (2006), no. 8, 646-657.
15. M. G. H. Omran, A. Salman and A. P. Engelbrecht, Self-adaptive differential evolution, Computational Intelligence and Security, PT 1, Proceedings Lecture Notes in Artificial Intelligence, 2005, p.^pp. 192-199.
16. D. Zaharie, Control of population diversity and adaptation in differential evolution algorithms, In: Proceedings of the 9th International Conference on Soft Computing, Brno 41-46 (2003).
17. J. Tvrdik, Adaptation in differential evolution: A numerical comparison, Applied Soft Computing **9** (2009), no. 3, 1149-1155
18. R. Mallipeddi, P. N. Suganthan, Q. K. Pan and M. F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Applied Soft Computing.
19. R. Storn and K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization **11** (1997), no. 4, 341-359.

20. R. Gämperle, S. D. Müller and P. Koumoutsakos, "A parameter study for differential evolution," Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, WSEAS Press, Interlaken, Switzerland, 2002, pp. 293-298.
21. S. Das, A. Konar and U. K. Chakraborty, Two improved differential evolution schemes for faster global search, Proceedings of the 2005 conference on Genetic and evolutionary computation (2005), 991-998.
22. J. Lampinen and I. Zelinka, On stagnation of the differential evolution algorithm, Proceedings of MENDEL 2000, 6th International Mendel Conference on Soft Computing (2000), 76-83.
23. K. V. Price, R. M. Storn and J. A. Lampinen (Editors), Differential evolution : A practical approach to global optimization, Springer, Berlin, 2005.
24. R. Storn and K. Price, Differential evolution : A simple evolution strategy for fast optimization, Dr. Dobb's Journal **22** (1997), no. 4, 18-24.
25. J. Rönkkönen, S. Kukkonen and K. V. Price, Real-parameter optimization with differential evolution, IEEE Congress on Evolutionary Computation, 2005, p.^pp. 506-513.
26. K. V. Price (Editor), An introduction to differential evolution, McGraw-Hill, London (UK), 79-108, 1999.
27. J. Zhang, Jade: Adaptive differential evolution with optional external archive, IEEE Transactions on Evolutionary Computation **13** (2009), no. 5, 945-958.
28. A. Iorio and X. Li, "Solving rotated multi-objective optimization problems using differential evolution," Australian Conference on Artificial Intelligence, Cairns, Australia., 2004, pp. 861-872.
29. D. Zaharie, Influence of crossover on the behavior of differential evolution algorithms, Applied Soft Computing **9** (2009), no. 3, 1126-1138.
30. E. Mezura-Montes, J. Velazquez-Reyes and C. A. Coello Coello, Modified differential evolution for constrained optimization, IEEE Congress on Evolutionary Computation, 2006, p.^pp. 25-32.
31. U. K. Chakraborthy, S. Das and A. Konar, Differentail evolution with local neighborhood, Proceedings of Congress on Evolutionary Computation, IEEE press, 2006, p.^pp. 2042-2049.
32. H. A. Abbass, The self-adaptive pareto differential evolution algorithm, Proceedings of the IEEE Congress on Evolutionary Computation **1** (2002), 831-836.
33. J. Liu and J. Lampinen, A fuzzy adaptive differential evolution algorithm, Soft Computing **9** (2005), no. 6, 448-462.
34. D. Zaharie and D. Petcu, Adaptive pareto differential evolution and its parallelization, Proc. of 5th International Conference on Parallel Processing and Applied Mathematics, 2003, p.^pp. 261-268.
35. J. Teo, Exploring dynamic self-adaptive populations in differential evolution, Soft Computing **10** (2006), no. 8, 673-686.
36. Z. Yang, K. Tang and X. Yao, "Self-adaptive differential evolution with neighborhood search," Proceedings of the 2008 IEEE Congress on Evolutionary Computation (CEC2008), Hong Kong, 2008, pp. 1110-1116.
37. S. Z. Zhao, P. N. Suganthan and S. Das, Self-adaptive differential evolution with multi-trajectory search for large scale optimization, Soft Computing, accepted (2010).
38. S. Das, A. Abraham, K. C. Uday and A. Konar, Differential evolution using a neighborhood-based mutation operator, IEEE Trans. on Evolutionary Computation **13** (2009), no. 3, 526-553.
39. D. Zaharie, Influence of crossover on the behavior of differential evolution algorithms, Applied Soft Computing **9** (2009), no. 3, 1126-1138