# Performance Evaluation of Differential Evolution Algorithm Using CEC 2010 Test Suite Problems

Amit Khaparde
Research Scholar, Computer Science and Engineering, GHRCE.
Faculty, Information Technology, RGCER.Nagpur, India.
*khaparde.amit@gmail.com*

Dr. M. M. Raghuwanshi,
Professor, Computer Technology
YCCE, Nagpur,India
*m_raghuwanshi@rediffmail.com*

Dr. L. G. Malik
Professor, Computer science and Engineering, GHRCE, Nagpur, India.
*Email: latesh.malik@raisoni.net*

***Abstract—*** Differential evolution algorithm (DE) is a novel parallel direct search evolutionary algorithm. Here we measure the performance of differential evolution algorithm on CEC 2010 test suite problems. It has found that the performance of standard differential evolution algorithm depend upon the value of decision parameters I,e parameter setting and DE require more explorative strategy during population evolution for large dimension problem.

***Keywords-*** *heuristic; functions;evolution;*

*****

## I. Introduction

R. Stone and K. Price [1] presented a new heuristic approach for minimizing possibly nonlinear and non-differentiable continuous space functions. This new method converges faster and with more certainty than many other acclaimed global optimization methods. this new method required few control variables, it was robust, easy to use, and lends itself very well to parallel computation.

As indicated by the recent studies on DE [2],[3], [4], [5] despite its simplicity, DE exhibits much better performance in comparison with several others like G3 with PCX, MA-S2, ALEP, CPSO-H, and so on of current interest on a wide variety of problems including unimodal, multimodal, separable, non-separable and so on.

Gamperle et al. [6] evaluated different parameter settings for DE on the Sphere, Rosenbrock's, and Rastrigin's functions. Their experimental results revealed that the global optimum searching capability and the convergence speed are very sensitive to the choice of control parameters NP, F, and Cr. Furthermore, a plausible choice of the population size NP is between 3-D and 8-D, the scaling factor F = 0.6, and the crossover rate Cr is between [0.3, 0.9]. Recently, the authors in [7] state that typically $0.4 < F < 0.95$ with $F = 0.9$ can serve as a good first choice. They also opine that Cr should be in (0, 0.2) when the function is separable, while in (0.9, 1) when the function's parameters are dependent. Lampinen J. Zelinka I. [8], Describe the basic nature and mechanism behind of stagnation. They also gave some reasons for stagnation and advices for reducing the risk of stagnation. As can be perceived from the literature, several claims and counter-claims were reported concerning the rules for choosing the control parameters and these can potentially confuse engineers, who may try to solve practical problems with DE. Further, most of these claims lack sufficient experimental justifications. Some objective functions are very sensitive to the proper choice of the parameter settings in DE [9]. Therefore, researchers naturally started to consider some techniques such as self-adaptation to automatically find an optimal set of control parameters for DE [10], [11], [12], [13].

In [11], a fitness-based adaptation has been proposed for F. A system with two evolving populations has been implemented. The crossover rate Cr has been fixed to 0.5 after an empirical study.

Brest et al. [12] proposed a self-adaptation scheme for the DE control parameters. They encoded control parameters F and Cr into the individual and adjusted them by introducing two new parameters $\tau 1$ and $\tau 2$. In their algorithm (called "jDE").

Qun. at el in [13] comes with self-adaptive schemes like SaDE that adapt the control parameter by using the standard deviation in normal distribution. However, self-adaptive DE performs better than the standard DE because sensitive parameters in DE are replaced by less sensitive parameters in self-adaptive DE.

Zaharie [14] proposed a parameter adaptation strategy for DE (ADE) based on the idea of controlling the population diversity, and implemented a multipopulation approach. Following the same line of thinking, Zaharie and Petcu [15] designed an adaptive Pareto DE algorithm for multiobjective optimization and also analyzed its parallel implementation.

Fan and Lampinen [16] proposed a trigonometric mutation operator for DE to speed up its performance

The concept of opposition-based learning was introduced by Tizhoosh [17] and its applications were introduced in [18],[19]. Rahnamayan et al. [4] have recently proposed an ODE for faster global search and optimization. The algorithm also finds important applications to the noisy optimization problems.

Some of the recent publications [7], [20] indicate that DE faces significant difficulty on functions that are not linearly separable and can be outperformed by CMA-ES. As pointed out by Sutton et al. [21], on such functions, DE must rely primarily on its differential mutation procedure, which, unlike its recombination strategy (with Cr < 1), is rotationally invariant.

## II. DIFFERENTIAL EVOLUTION ALGORITHM

Differential evolution (DE) is a novel parallel direct search method which utilizes NP parameter vectors:

$x_{i,G}$, i = 1,2,3 . . . . . NP-1 as a population for each generation G. NP doesn't change during the minimization process. If nothing is known about the system then initial population is chosen randomly otherwise a uniform probability distribution use for all random decisions. If a preliminary solution is available, then initial population is often generated by adding normally distributed random deviations to the nominal solution $x_{nom,o}$ . DE generates new parameter vectors by adding the weighted difference between two population vectors to a third vector , called mutation.

### A. Mutation

For each target vector $x_{i,G}$, i = 1, 2, 3 . . . NP, a mutant vector is generated according to

$$vi_{;G} = x_{r1,G} + F * (x_{r2,G} - xr_{3,G})$$

Whereas $r_1, r_2, r_3 \in 1 . . . NP$, integer, mutually different and $F > 0 \in [0, 2]$ The randomly chosen integers $r_1$, $r_2$ and $_{r3}$ are also chosen to be different from the running index 'i' . F is a real and constant factor which controls the amplification of the

differential variation $(xr2_{,G} - xr_{3,G})$.

The mutated vector's parameters are then mixed with the parameters of another predetermined vector, the target vector, to yield the so-called trial vector. Parameter mixing is referred to as crossover.

### B. Crossover

It increases the diversity of the perturbed parameter vectors. Here the trial vector

$u_{i,G+1} = (u_{1i,G+1}, u_{2i,G+1} . . . u_{Di,G+1})$ is formed, where

$$u_{ji,G+1} = \begin{cases} v_{ji,G} & \text{if } (randb(j) \le CR) \text{ or } j = rnbr(i) \\ x_{ji,G} & \text{if } (randb(j) > CR) \text{ and } j \ne rnbr(i) ; \end{cases}$$

$$j = 1, 2 . . . D:$$

randb(j) is the jth evaluation of a uniform random number generator $\in [0, 1]$.CR is the crossover constant $\in [0,1]$. rnbr(i) is a randomly chosen index $\in$ 1,2….D which ensures that $u_{i,G+1}$ gets at least one parameter from $vi_{,G}$

If the trial vector yields a lower cost function value than the target vector, the trial vector replaces the target vector in the following generation. This last operation is called selection.

Selection

To decide whether or not new vector should become a member of generation G+1, the trial vector $u_{i,G+1}$ is compared to the target vector $x_{i,G}$ using the greedy criterion. If vector $u_{i,G+1}$ yields a smaller cost function value than $x_{i,G}$, then $x_{i,G+1}$ is set to $u_{i,G+1}$; otherwise, the old value $x_{i,G}$ is retained

Each population vector has to serve once as the target vector so that NP competitions take place in one generation. This process is repeat until the stopping criteria are not meet

### C. Other variants of DE

There are number of variants in DE. In order to represent the variants the following notation is used ; DE=x/y/z is

Where as

- x: The vector to be mutated which currently can be "rand" (a randomly chosen population vector) or "best" (the vector of lowest cost from the current population).
- y: Number of difference vectors used.
- z: The crossover scheme.

TABLE I. STRATEGIES OF DE

| Sr . No | Strategy | S r. No | Strategy |
|---|---|---|---|
| 1 | DE/best/1/exp | 6 | DE/best/1/bin |
| 2 | DE/rand/1/exp | 7 | DE/rand/1/bin |
| 3 | DE/rand-to-best/1/exp | 8 | DE/rand-to-best/1/bin |
| 4 | DE/best/2/exp | 9 | DE/best/2/bin |
| 5 | DE/rand/2/exp | 10 | DE/rand/2/bin |

Using this notation, the basic DE-strategy is DE/rand/1/bin. Nevertheless, one highly beneficial method that deserves special mention is the method DE/best/2/bin: Price (1996), where

$$v_{i,G+1} = x_{best;G} + F * (xr_{1,G} + xr_{2,G} - xr_{3,G} - xr_{4,G})$$

## III. EXPERIMENTAL SETUP

The Performance of Differential evolution is checked on CEC 2010 test suite problems . the parameter setting are as follows :

TABLE II. PARAMETER SEETING

| Cr | F | N | Max-Fes |
|---|---|---|---|
| 0.9 | 0.9 | 20 | 3e+03 |

The classical DE has been tested on the benchmark function which are given in CEC 2010 [21] competition. All the twenty problem can be divided in the following manner each problem have 1000 variables. The problems are as follows:
Separable Functions (3)

*a) F1: Shifted Elliptic Function*

b) *F2: Shifted Rastrigin's Function*

c) *F3: Shifted Ackley's Function*

*Single-group m-non-separable Functions (5)*

d) *F4: Single-group Shifted and m-rotated Elliptic Function*

e) *F5: Single-group Shifted and m-rotated Rastrigin's Function*

f) *(c) F6: Single-group Shifted and m-rotated Ackley's Function*

g) *F7: Single-group Shifted m-dimensional Schwefel's*

h) *F8: Single-group Shifted m-dimensional Rosenbrock's Function*

*3. D/2m-group m-non-separable Functions (5)*

i) *(a) F9: D/2m-group Shifted and m-rotated Elliptic Function*

j) *F10: D/2m-group Shifted and m-rotated Rastrigin's Function*

k) *F11: D/2m-group Shifted and m-rotated Ackley's Function*

l) *F12: D/2m-group Shifted m-dimensional Schwefel's Problem*

m) *F13: D/2m-group Shifted m-dimensional Rosenbrock's Function*

*4. D/m-group m-nonseparable Functions (5)*

n) *F14: D/m-group Shifted and m-rotated Elliptic Function*

o) *F15: D/m-group Shifted and m-rotated Rastrigin's Function*

p) *F16: D/m-group Shifted and m-rotated Ackley's Function*

q) *F17: D/m-group Shifted m-dimensional Schwefel's Problem 1.2*

r) *F18: D/m-group Shifted m-dimensional Rosenbrock's Function*

*5. Nonseparable Functions (2)*

s) *F19: Shifted Schwefel's Problem*

t) *F20: Shifted Rosenbrock's Function*

Solution quality for each function when the FEs counter reaches

- 1.20E+03
- 2.00E+03
- 3.00E+03

The best, median, worst function values mean and standard deviation of the 25 runs is recorded

Form Table 3 and table 4 , it can be see that , except in problem no, 1, there is no change in best value of population because population has lost its diversity . There is unnecessary function evaluation from 2.0E3 to 3.oE3. This can be solved by increasing the number of population. Except problem no. 3 and 11 , the DE is not converge to its global optima , this can be solve by increasing the value of Max_Fes and give maximum chance to each member to evolve .

## IV. CONCLUSION

The results of the Classical DE with the other algorithms which took part in the competition have been tested. It has found that the classical DE performance is not up to the mark. The performance of DE is depending upon the decision parameters. To improve the performance of classical DE, there is need to find out the more explorative strategy with less number of decision parameter for population evolution in classical DE.

## REFERENCES


[1] STORN R. , PRICE K. "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces". Journal of Global Optimization 11: 341–359, 1997.

[2] KARABOGA D. OKDEM S. "A Simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm " Turk J Elec Engin, VOL.12, NO.1 2004

[3] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, *"Differential evolution using a neighborhood based mutation operator,"* IEEE Trans. Evol. Comput., vol. 13, no. 3, pp. 526–553, Jun. 2009

[4] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, *"Opposition- based differential evolution,"* IEEE Trans. Evol. Comput., vol. 12, no. 1, pp. 64–79, Feb. 2008

[5] J. Zhang and A. C. Sanderson, *"JADE: Adaptive differential evolution with optional external archive,"* IEEE Trans. Evol. Comput., vol. 13, no. 5, pp. 945–958, Oct. 2009

[6] GAMPERLE R,. MULLER S, KOUMOUTSAKOS P. *".A Parameter Study for Differential Evolution".* WSEAS NNA-FSFS-EC 2002, Interlaken, Switzerland, Feb. 11-15, 2002.

[7] J. Ronkkonen, S. Kukkonen, and K. V. Price, *"Real parameter optimization with differential evolution,"* in Proc. IEEE CEC, vol. 1. 2005, pp. 506–513.

[8] Lampinen J. Zelinka I. *"On stagnation of the differential evolution algorithm,"* in Proc. 6th Int. Mendel Conf. Soft Compute., Jun. 2000, pp. 76–83.

[9] J. Liu and J. Lampinen, *"On setting the control parameters of the differential evolution method",* in: R. Matoušek and P. Ošmera, (eds.) Proc. of Mendel 2002, 8-th International Conference on Soft Computing, pp. 11–18, 2002.

[10] J. Liu and J. Lampinen, *" Adaptive parameter control of differential evolution "*, R. Matoušek and P. Ošmera, (Eds.) Proc. of Mendel 2002, 8-th International Conference on Soft Computing, pp. 19–26, 2002.

[11] M. M. Ali and A. Törn, " Population set based global optimization Algorithms : Some modifications and numerical studies," Comput. Oper. Res., vol. 31, no. 10, pp. 1703–1725, 2004.

[12] J. Brest, S. Greiner, B. Boˇskovi´c, M. Mernik, and V. ˇZumer, *"Self adapting 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.

[13] 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

[14] D. Zaharie, *"Control of population diversity and adaptation in differential evolution algorithms",* In D. Matousek, P. Ošmera (eds.), Proc. of MENDEL 2003,In 9th International Conference on Soft Computing, Brno, Czech Republic, pp. 41-46, June 2003.

[15] D. Zaharie and D. Petcu, *"Adaptive Pareto differential evolution and its parallelization",* Proc. Of 5th International Conference on Parallel Processing and Applied Mathematics, Czestochowa, Poland, Sept. 2003.

[16] H.-Y. Fan and J. Lampinen, "A trigonometric mutation operation to differential evolution," J. Global Optimization, vol. 27, no. 1, pp. 105–129, 2003

[17] H. R. Tizhoosh, "Opposition-Based Learning: A New Scheme for Machine Intelligence", Int.Conf. on Computational Intelligence for Modeling Control and Automation - CIMCA'2005, Vol.I, pp. 695-701, Vienna, Austria, 2005.

[18] H. R. Tizhoosh, "Reinforcement learning based on actions and opposite actions". ICGST International Conference on Artificial Intelligence and Machine Learning (AIML-05), Cairo, Egypt, 2005.

[19] H.R. Tizhoosh, "Opposition-based reinforcement learning", Journal of Advanced Computational Intelligence and Intelligent Informatics", Vol. 10, No. 3, 2006.

[20] N. Hansen and S. Kern, "Evaluating the CMA evolution strategy on multimodal test functions", In Proceedings of 8th International Conference on Parallel Problem Solving from Nature (PPSN), pages 282–291, Berlin, Germany, 2004.

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

TABLE III. PERFORMANCE OF DE ON 1-10 PROBLEMS

| | **Parameters** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **1.20E+03** | **Best** | 1E+11 | 3E+06 | 21.4 | 5E+13 | 1E+09 | 2E+07 | 4E+10 | 2E+12 | 7E+10 | 3E+06 |
| | **Median** | 2E+11 | 3E+06 | 21.5 | 5E+14 | 1E+11 | 2E+07 | 2E+11 | 3E+14 | 4E+11 | 3E+06 |
| | **Worst** | 3E+11 | 3E+06 | 21.7 | 1E+16 | 2E+11 | 2E+07 | 6E+12 | 3E+17 | 5E+11 | 3E+06 |
| | **Mean** | 2E+11 | 3E+06 | 21.5 | 1E+15 | 8E+10 | 2E+07 | 4E+11 | 2E+16 | 3E+11 | 3E+06 |
| | **Std** | 7E+10 | 54846 | 0.07 | 1E+15 | 6E+10 | 46971 | 4E+11 | 6E+16 | 1E+11 | 50400 |
| **2.00E+03** | **Best** | 1E+11 | 3E+06 | 21.4 | 5E+13 | 1E+09 | 2E+07 | 4E+10 | 2E+12 | 7E+10 | 3E+06 |
| | **Median** | 2E+11 | 3E+06 | 21.5 | 5E+14 | 1E+11 | 2E+07 | 2E+11 | 3E+14 | 4E+11 | 3E+06 |
| | **Worst** | 3E+11 | 3E+06 | 21.7 | 1E+16 | 2E+11 | 2E+07 | 6E+12 | 3E+17 | 5E+11 | 3E+06 |
| | **Mean** | 2E+11 | 3E+06 | 21.5 | 1E+15 | 8E+10 | 2E+07 | 4E+11 | 2E+16 | 3E+11 | 3E+06 |
| | **Std** | 8E+10 | 54307 | 0.07 | 1E+15 | 6E+10 | 46956 | 4E+11 | 5E+16 | 1E+11 | 52323 |
| **3.00E+03** | **Best** | 9E+10 | 3E+06 | 21.4 | 5E+13 | 1E+09 | 2E+07 | 4E+10 | 2E+12 | 7E+10 | 3E+06 |
| | **Median** | 1E+11 | 3E+06 | 21.5 | 5E+14 | 1E+11 | 2E+07 | 2E+11 | 2E+14 | 4E+11 | 3E+06 |
| | **Worst** | 3E+11 | 3E+06 | 21.7 | 1E+16 | 2E+11 | 2E+07 | 6E+12 | 3E+17 | 5E+11 | 3E+06 |
| | **Mean** | 2E+11 | 3E+06 | 21.5 | 1E+15 | 8E+10 | 2E+07 | 4E+11 | 2E+16 | 3E+11 | 3E+06 |
| | **Std** | 8E+10 | 53667 | 0.07 | 1E+15 | 6E+10 | 46928 | 4E+11 | 5E+16 | 1E+11 | 54359 |

TABLE IV. PERFORMANCE OF DE ON 10-20 PROBLEMS

| | Parameters | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1.20E+03 | **Best** | 237.3 | 7E+06 | 3E+12 | 9E+10 | 3E+06 | 432.2 | 2E+07 | 7E+12 | 7E+07 | 8E+12 |
| | **Median** | 237.5 | 2E+07 | 4E+12 | 4E+11 | 3E+06 | 432.39 | 3E+07 | 8E+12 | 2E+08 | 9E+12 |
| | **Worst** | 238.4 | 7E+15 | 4E+12 | 5E+11 | 3E+06 | 433.26 | 1E+08 | 8E+12 | 8E+08 | 9E+12 |
| | **Mean** | 237.5 | 4E+11 | 4E+12 | 4E+11 | 3E+06 | 432.34 | 3E+07 | 8E+12 | 2E+08 | 9E+12 |
| | **Std** | 0.135 | 5E+13 | 2E+11 | 1E+11 | 43809 | 0.1533 | 2E+07 | 2E+11 | 7E+07 | 2E+11 |
| 2.00E+03 | **Best** | 237.3 | 7E+06 | 3E+12 | 9E+10 | 3E+06 | 432.2 | 1E+07 | 7E+12 | 7E+07 | 8E+12 |
| | **Median** | 237.5 | 2E+07 | 4E+12 | 4E+11 | 3E+06 | 432.2 | 3E+07 | 8E+12 | 2E+08 | 9E+12 |
| | **Worst** | 238.4 | 7E+15 | 4E+12 | 5E+11 | 3E+06 | 433.26 | 1E+08 | 8E+12 | 8E+08 | 9E+12 |
| | **Mean** | 237.5 | 4E+11 | 4E+12 | 4E+11 | 3E+06 | 432.29 | 3E+07 | 8E+12 | 2E+08 | 9E+12 |
| | **Std** | 0.135 | 5E+13 | 2E+11 | 1E+11 | 44377 | 0.1371 | 2E+07 | 2E+11 | 7E+07 | 2E+11 |
| 3.00E+03 | **Best** | 237.3 | 7E+06 | 3E+12 | 7E+10 | 3E+06 | 431.83 | 1E+07 | 7E+12 | 7E+07 | 8E+12 |
| | **Median** | 237.5 | 2E+07 | 4E+12 | 4E+11 | 3E+06 | 432.2 | 2E+07 | 8E+12 | 2E+08 | 9E+12 |
| | **Worst** | 238.4 | 7E+15 | 4E+12 | 5E+11 | 3E+06 | 433.26 | 1E+08 | 8E+12 | 8E+08 | 9E+12 |
| | **Mean** | 237.5 | 4E+11 | 4E+12 | 4E+11 | 3E+06 | 432.26 | 3E+07 | 8E+12 | 2E+08 | 9E+12 |
| | **Std** | 0.136 | 5E+13 | 2E+11 | 1E+11 | 45009 | 0.1193 | 2E+07 | 2E+11 | 7E+07 | 2E+11 |