Two Enhanced Differential Evolution Variants for Solving Global Optimization Problems

Pravesh Kumar
Indian Institute of Technology, Roorkee, India
praveshtomariitr@gmail.com

Millie Pant
Indian Institute of Technology, Roorkee, India
millidma@gmail.com

Ajith Abraham
Machine Intelligence Research Labs (MIR Labs), WA, USA
VSB-Technical University of Ostrava, Czech Republic
ajith.abraham@ieee.org

***Abstract*—Differential Evolution (DE) algorithms are very robust, effective and highly efficient in solving the global optimization problems. Thus, they are usually able to mitigate the drawback of long computation times commonly associated with Evolutionary algorithms. However, in certain cases the performance of DE is observed not to be completely flawless. In this paper we have proposed the two enhanced variants of DE using a modified mutation operator. The DE versions named as EDE-1 and EDE-2 are tested on six benchmark problems and a real time molecular potential energy problem. The simulation results prove the efficiency as well as the effectiveness of the proposed variants.**

Keywords- Differential Evolution, Donor Mutation, Global optimization, Molecular potential energy.

I. INTRODUCTION

Differential Evolution (DE) algorithm was proposed by Price and Storn in 1995 [1], [2]. It is a population based, direct-search algorithm used for optimisation in continuous domains. Its effectiveness and efficiency have been successfully demonstrated in many application fields such as pattern recognition, communication, and mechanical engineering and so on [1], [3], [4]. DE outperforms many other optimization algorithms in terms of convergence speed and robustness over common benchmark problems and real world applications. However, it has been observed that the convergence rate of DE do not meet the expectations all the time, especially in case of highly multimodal problems. In order to improve the performance of DE, its several variants have been proposed. There are various mutation strategies available in the literature [5]-[7] and [11]-[12]. In this paper we have taken the basic DE strategy *DE/rand/1/bin* [7], [12]. We shall refer to it as simple DE (SDE)

In the present study we have proposed two new variants of DE, using a modified mutation operator, named EDE-1 and EDE-2.

The rest of paper is organized as follows: Section II provides a compact overview of DE. Section III presents the proposed EDE-1 and EDE-2 algorithms. In Section IV benchmark problems and the real life problem of Molecular Potential Energy problem is given. Experimental settings are given in Section V. Results and discussions are reported in Section VI, and finally the conclusions derived from the present study are drawn in Section VII

II. DIFFERENTIAL EVOLUTION (DE)

Basic DE (SDE) algorithm is a kind of evolutionary algorithm, which is used to optimize real valued functions. In this paper the term classic DE refers to the DE/rand/1/bin scheme. DE starts with an initial population vector, which is randomly generated when no preliminary knowledge about the solution space is available. Let $x^i_G = (x^i_{1,G}, x^i_{2,G} \ldots, x^i_{D,G})$ $i=1,2\ldots,NP$ are the solution vector, where i denote the population, D denotes the dimension of the search space and G denote the generation in which the population belongs.

For classical DE (DE/rand/1/bin), the mutation, crossover, and selection operators are defined as follows:

a) Mutation: For each vector x^i_G (target vector), select three distinct vectors x^{r1}_G, x^{r2}_G and x^{r3}_G (different from each other and also from the target vector) randomly from the current population other than vector x^i_G. Now generate a new population vector m^i_{G+1} (called *perturbed* or *mutated* vector) as:

$$m^i_G = x^{r1}_G + F(x^{r2}_G - x^{r3}_G) \tag{1}$$

Where $F \in [0, 2]$ is called scale factor which is used to control the amplification of the differential variation $(x^{r2}_G - x^{r3}_G)$.

b) Crossover: Perform crossover operation to create a trial vector $u^i_{G+1} = (u^i_{1,G+1}, u^i_{2,G+1} \ldots, u^i_{D,G+1})$ as:

$$u^i_{j,G+1} = \begin{cases} m^i_{j,G+1}, if\ Cr < rand(01) \forall k = j \\ x^i_{j,G+1} \quad otherwise \end{cases} \tag{2}$$

Where j = 1, 2, …, D; $rand(0, 1)$ is uniform random number between 0 and 1; Cr is the crossover constant takes values in the range [0, 1] and k 1, 2,..., D; is the randomly chosen index.

c) Selection: Generate new population vector x^i_{G+1} for next generation $G+1$ by using equation given below:

$$x_{G+1}^{i} = \begin{cases} u_{G+1}^{i}, if\ f(u_{G+1}^{i}) < f(x_{G}^{i}) \\ x_{G}^{i} \quad otherwise \end{cases} \tag{3}$$

III. PROPOSED ALGORITHMS

In this Section, we describe the proposed *EDE-1* and *EDE-2*. In our proposed algorithms, we used two new mutation strategies based on Donor mutation [8] and then selected the mutation strategy stochastically either from the basic DE or from the newly proposed strategy. For this purpose first we fix a probability (*Pr*) and then generate a uniform random number (*R*) between 0 & 1. If the value of *R* is less than *Pr* then select a new mutation strategy otherwise select basic mutation strategy (as per eq. 1).

The new strategies, say M1 and M2, are defined as:

$$M1: m_{G+1}^{i} = (\mu_1 x_G^{r1} + \mu_2 x_G^{r2} + \mu_3 x_G^{r3}) + F(x_G^{r2} - x_G^{r3}) \tag{4}$$

Here μ_i *i*=1, 2 are uniform random number between 0 and 1 and $\mu_3 = 1-(\mu_1+\mu_2)$ (satisfies the condition, $\sum_{i=1}^{3}\mu_i = 1$)

The other strategy is defined as:

$$M2: m_{G+1}^{i} = (\lambda_1/\lambda)x_G^{r1} + (\lambda_2/\lambda)x_G^{r2} + (\lambda_3/\lambda)x_G^{r3} + F(x_G^{r2} - x_G^{r3}) \tag{5}$$

Where λ_i, *i*=1, 2, 3 are uniform random number between 0 and 1.and $\lambda = \sum_{i=1}^{3}\lambda_i$

A. Computational Steps for the proposed variants

1. Initialize the population
2. Perform Mutation:
   a) Perform Donor mutation with Equation (4) for *EDE-1* or with Equation (5) for EDE-2 with a probability *Pr*
   b) Perform original mutation as use in DE (with a probability 1-*Pr*).
3. Perform crossover operation.
4. Evaluate the objective function.
5. Selection.
6. Repeat from step 2 to 5.

B. Pseudo Code for Proposed Algorithm

Let *P* be population of size N_P and let x^i_G be any individual of dimension *D* in population *P*.

Begin

1. Create a uniformaly random initial population x^i *i, i=1, 2, ...NP*
   $x^i_G = x^{lower}_{,G} + rand(0, 1)*(x^{upper}_G - x^{lower}_G)$
2. Evaluate $f(x^i)$
3. **While**(*BFV* > *VTR* and *NFE*<Max_{NFE}) // *NFE*= Number of Function Evaluation, *VTR*= value to reach
4. **For** *i*=1 to *NP* **Do**
   Select three random parents x^{r1}_G, x^{r2}_G and x^{r3}_G from the current population where *i*≠*r1*≠*r2* ≠*r3*
   R=rand (0 1) // uniform random number between 0 and 1
5. **If** (*R*<*Pr*)
   Perform mutation m^i_{G+1} from equation-4 (*EDE-1*) or equation-5 (*EDE-2*)
6. **else**, Perform Mutation m^i_{G+1} from equation-1
7. **End If**
8. *k=randint* (*1*, *D*)
9. **For** j=1 to D **Do**
10. If ($rand_j$<*Cr* or *j=jrand*)
11. $u^i_{j,G+1} = m^i_{j,G+1}$
12. Else, $u^i_{j,G+1} = x^i_G$
13. **End If**
14. **End For**
15. Evaluate $f(u^i_{G+1})$
16. If($f(u^i_{G+1}) < f(x^i_G)$)
17. $x^i_{G+1} = u^i_{G+1}$
18. *else* $x^i_{G+1} = x^i_G$
19. **End if**
20. **End For**
21. **End While**
22. **End**

IV. APPLICATION OF PROPOSED ALGORITHMS

A. Benchmark Problems

6 common benchmark functions with boundary constraints are used for experiments. These problems are selected from [8]. This test bed though narrow forms a good launch pad for validating the efficiency of an optimization algorithm.

- Ackley Function:

$$F_1 = -20\exp\left(-0.2\sqrt{1/n\sum_{i=0}^{n}x_i^2}\right) - \exp\left(\sqrt{1/n\sum_{i=0}^{n}\cos(2\pi x_i}\right) + 20 + e$$

With $-32 \le x_i \le 32$ and min $F_1(0, 0..., 0) = 0$.

- Griewenk Function

$$F_2 = \frac{1}{4000}\sum_{i=1}^{n}x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

With $-600 \le x_i \le 600$ and min $F_2(0, 0..., 0) = 0$

- Noise Function

$$F_3 = \sum_{i=1}^{n} i x_i^4 + random(0\ 1)$$

With $-1.28 \le x_i \le 1.28$, min $F_3(0, 0... 0) = 0$.

- Restrigin's Function:

$$F_4 = 10n + \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi x_i)\right)$$

With $-5.12 \le x_i \le 5.12$ and min $F_4(0, 0..., 0) = 0$

- Sphere Function:

$$F_5 = \sum_{i=1}^{n} x_i^2$$

With $-5.12 \le x_i \le 5.12$ and min $F_5(0, 0... 0) = 0$

- Step Function:

$$F_6 = \sum_{i=1}^{n}(\lfloor x_i + 0.5 \rfloor)^2$$

With $-5.12 \le x_i \le 5.12$ and min $F_6(0, 0... 0) = 0$

B. Molecular Potential Energy Problem [9],[10]

To further validate the performance of the proposed variants we tested them on a real time problem of minimizing the potential energy of a molecule. The mathematical model of this problem is multimodal in nature.

A simplified molecular model consists of a linear chain of n beads centred at $x_1, x_2, ...x_n$ in a 3-dimensional space is considered. For every pair of consecutive beads x_i and x_{i+1}, let $r_{i,i+1}$ be the bond length which is the Euclidean distance between them. For every three consecutive beads $x_1, x_2, ...x_n$, let $\theta_{i,i+1}$ be the bond angle corresponding to the relative position of the third bead with respect to the line containing the previous two.

Likewise, for every four consecutive beads, $x_i, x_{i+1}, x_{i+2}, x_{i+3}$ let $\omega_{i, i+3}$ be the angle, called the torsion angle, between the normal's through the planes determined by the beads x_i, x_{i+1}, x_{i+2} and $x_{i+1}, x_{i+2}, x_{i+3}$.

The force field potentials corresponding to bond lengths, bond angles and torsion angles will be defined respectively as

$$\begin{aligned} E_1 &= \sum_{(i,j)\in M_1} c^1_{i,j}(r_{i,j} - r^0_{i,j})^2 \\ E_2 &= \sum_{(i,j)\in M_2} c^2_{i,j}(\theta_{i,j} - \theta^0_{i,j})^2 \\ E_3 &= \sum_{(i,j)\in M3} c^3_{i,j}(1 + \cos(3\omega_{i,j} - \omega^0_{i,j})) \end{aligned} \quad (6)$$

where $c^1_{i,j}$ is bond stretching force constant, $c^2_{i,j}$ is angle bending force constant and $c^3_{i,j}$ is the torsion force constant. The constants $r^0_{i,j}$ and $\theta^1_{i,j}$ represent the "preferred" bond length and bond angle respectively and $\omega^1_{i,j}$ is the phase angle that defines the position of the minima. M_k *k=1, 2, 3* represents the set of pair of atoms separated by *k* covalent bonds. In addition to the above, there is also a potential E_4 which characterizes the 2-body interactions between every pair of beads separated by more than two covalent bonds along the chain. We use the following function to represent E_4:

$$E_4 = \sum_{(i,j)\in M_3}\left(\frac{(-1)^i}{r_{i,j}}\right) \quad (7)$$

where r_{ij} is the Euclidean distance between the beads x_i and x_j. The general problem is to minimize the total molecular potential energy $E=E_1+ E_2+ E_3+ E4$, leading to the optimal spatial position of the beads. Using the parameters defined in [9] potential energy function takes the following form

$$\begin{aligned} E &= \sum(1 + \cos(3\omega_{i,i+3})) \\ &+ \sum_i\left(\frac{(-1)^i}{\sqrt{10.60099896 - 4.141720682\cos}(\omega_{i,i+3})}\right) \end{aligned} \quad (8)$$

where *i=1, 2,...n-3* and *n* is the number of beads in the given system. The problem thus reduces to find $\omega_{i,i+3}$,.*i=1, 2,..,n*

it can easily be observed from equation (8) that *E* is a nonconvex function and involving numerous local minimizers even for small value of *n*.

These local minimizers correspond to a state which is not truly stationary but is almost stationary called *metastable state* of the molecule.

The number of local minimizers of the function defined by equation (8) is 2^N, where *N=n-3* is the total number of beads in a molecule [9].

By restricting $\omega_{i,j}$: $0<\omega_{i,j}<5$ the existence of only one global minimum is guaranteed.

V. EXPERIMENTAL SETTING

Both EDE-1 and EDE-2 are implemented in Dev-C++ and the experiments are conducted on a computer with 2.00 GHz Intel (R) core (TM) 2 duo CPU and 2- GB of RAM.

The parameters used are set as follows:

Pop size (*NP*)	100
Dimension (*D*)	15, 25 (for Benchmark Problem) 15, 20, 25 (For Potential energy Problem)
Scale Factor (*F*) ,	0.5,
Crossover rate (*Cr*)	0.5,
Probability (Pr)	0.1
Value to reach (*VTR*)	10^{-04}
Max NFE	10^6

Over all acceleration rate AR, which is taken for the purpose of comparison is defined as [11]:

$$AR = \left(1 - \frac{\sum_{i=1}^{\mu} NFE\ by\ one\ a\lg orithm}{\sum_{i=1}^{\mu} NFE\ by\ another\ a\lg orithm}\right) * 100 \quad (9)$$

Where, μ is number of functions.

In every case, a run was terminated when the best function value obtained is less than a threshold for the given function or when the maximum number of function evaluation (NFE=10^6) was reached.

VI. SIMULATION RESULT AND DISCUSSION

In this section we analyze the proposed EDE-1 and EDE-2 algorithms by comparing it with basic DE. We compare the convergence speed of DE and MPDE by measuring the NFE.

Smaller NFE indicates higher convergence speed. The termination criterion is to find a value smaller than the value-to-reach (VTR=10^{-04}) before reaching the maximum number of function evaluation (NFE=10^6).

In order to minimize the effect of the stochastic nature of the algorithms, every result is taken as average of 30 different runs.

A. Numerical Result for Benchmark Problems

The proposed algorithms are compared with the SDE and the results are recorded in Table I and Table II.

The performance metric used to analyze the performance of the proposed algorithms include comparison of mean fitness values, total NFE, percentage improvement in terms of NFE and average time taken for the execution of 6 benchmarks function by each algorithm.

From Table I, which gives the mean fitness and standard deviation (SD), we can see that all the three algorithms perform more or less in a similar manner although EDE2 outperforms the others in terms of average fitness value in most of the cases. Thus judging the algorithms on the basis of fitness does not give a concrete conclusion.

The performance of the algorithms can be distinguished more clearly from Table II, which tabulates the results in terms of average NFE and CPU time.

It is clearly evident from this TABLE that in terms of NFE and time taken by EDE-2 is significantly better than both EDE-1 and the basic DE or SDE.

For solving 6 problems the average NFE taken by EDE-2 are 466780 and 1196260 for D=15 and 25 respectively while NFE taken by DE, and EDE-1 are 713800, and 696170 for D=15 and 2759350 and 2225660 for D=25 respectively.

This implies that acceleration rate for EDE-2 in comparison to DE is 34.61% and 56.64 % for D=15 and D=25 respectively while acceleration rate for EDE-1 in comparison to DE is only 2.46% and 19.34% for D=15 and D=25 respectively.

A similar observation about the superior performance of EDE-2 can be made from the average execution time given in Table II, which shows that for all the test problems EDE2 took lesser CPU time to converge.

TABLE I. AVERAGE MEAN FITNESS VALE AND STANDARD DEVIATION IN 30 RUN

Function	**Dim**	**Mean Fitness Value and Standard Deviation**		
		SDE	*EDE-1*	*EDE-2*
Ackley	15	**0.00089** (6.8e-005)	0.00095 (4.3e-005)	0.00092 (3.4e-005)
	25	0.00095 (5.11e-005)	0.00095 (5.6e-005)	**0.00094** (3.5e-005)
Griewank	15	0.00088 (6.6e-005)	0.00083 (0.0001)	**0.00077** (0.0001)
	25	0.00091 (7.5e-005)	0.00091 (6.1e-005)	**0.00083** (0.0001)
Noise	15	0.00082 (0.00019)	**0.00080** (0.00016)	0.00083 (0.00013)
	25	0.00087 (0.00012)	0.00087 (9.7e-005)	**0.00077** (0.00017)
Restring	15	**0.00084** (0.00014)	0.00088 (8.8e-005)	0.00087 (9.4e-005)
	25	0.00092 (5.9e-005)	0.00094 (5.7e-005)	**0.00086** (0.00013)
Sphere	15	0.00088 (7.1e-005)	0.00087 (7.6e-005)	**0.00079** (0.00011)
	25	0.00093 (3.9e-005)	**0.00083** (0.00012)	0.00090 (7.9e-005)
Step	15	0 (0)	0(0)	0 (0)
	25	0 (0)	0 (0)	0 (0)

B. Numerical Result for Molecular Potential Energy Problems

TABLE-III gives the result and comparison for Molecular Potential Energy problem for different dimension.

From this Table we can see that the total NFE taken by EDE-2 are 46900, 106140 and 164060 for dimensions D= 15, 20 and 25 respectively while NFE taken by EDE-1 are 52640, 137420 and 279130 respectively and the NFE taken by SDE are 54900, 144550 and 390890 for D=15, 20 and 25 respectively.

This implies that AR of EDE-2 in comparison to DE is 14.97%, 26.57% 58.02% while AR of EDE-1 in comparison to DE is.4.2%, 4.9% and 28.59% for D=15, 20 and 25 respectively.

Thus, we can see that when the dimension is increase then our proposed EDE-1 and EDE-2 gives better results in comparison to SDE.

TABLE II. AVERAGE NFE AND CPU TIME IN 30 RUN

Function	Dim	*NFE and Time (sec)*		
		SDE	*EDE-1*	*EDE-2*
Ackley	15	30750 (1.3 sec)	28510 (1.6 sec)	**25420 (1 sec)**
	25	54610 (3.7 sec)	46810 (3 sec)	**39770 2.4 sec**
Griewank	15	43980 (2.1 sec)	36650 (1.7 sec)	**30030 (1.4 sec)**
	25	55180 (4.3 sec)	41980 (3.1 sec)	**36350 (2.5 sec)**
Noise	15	307220 (12.2 sec)	*316440* (10.7 sec)	**203410 (7.5 sec)**
	25	1269450 (78.5 sec)	866710 (43.7 sec)	**586170 32.6 sec**
Restring	15	313490 (12 sec)	297790 13.4 sec	**192350 (7 sec)**
	25	1345920 (65 sec)	1239940 (50 sec)	**508740 (39.6 sec)**
Sphere	15	15000 (0.7 sec)	13970 (0.7 sec)	**12680 (0.6 sec)**
	25	28030 (1.7 sec)	24460 (1.5 sec)	**20570 (1.2 sec)**
Step	15	3360 (0.2 sec)	**2810 (0.2 sec)**	2890 (0.2 sec)
	25	6310 (0.6 sec)	5760 (0.4 sec)	**4690 (0.4 sec)**
Total NFE AR(%)	15	713800	696170 2.46%	**466780 34.61%**
	25	2759350	2225660 19.34%	**1196260 56.64%**

TABLE III. AVERAGE NFE, MEAN VALUE AND CPU TIME IN 30 RUN FOR MOLECULRA POTENTIAL ENERGY PROBLEM

Algorithms		**D=15**	**D=20**	**D=25**
SDE	*NFE*	54900	144550	390890
	Mean fitness	-0.4930	-1.0001	-0.9045
	CPU Time	0.8 sec	2.7 sec	10.5 sec
EDE-1	*NFE*	52640	137420	279130
	Mean fitness	-0.4931	-1.0001	-0.9045
	CPU Time	0.9 sec	3.5 sec	7.2 sec
EDE-2	*NFE*	**46900**	**106140**	**164060**
	Mean fitness	-0.4930	-1.00009	-0.9045
	CPU Time	**0.6 sec**	**2.1 sec**	**4.3 sec**

VII. CONCLUSIONS

In the present study, we proposed two enhanced variants of DE algorithm namely EDE-1 and EDE-2. In both the algorithms new mutation schemes (called M1 and M2) are introduced.

The basic structure of both the variants is same except for the mutation strategies that are used in them.

In EDE-1, which uses M1 and the basic mutation scheme of SDE stochastically, three random numbers from population are taken and each random number is multiplied by weighted random number whose summation is equals to 1 whereas in EDE-2, which uses M2 and SDE mutation scheme stochastically, the process is same but the summation is not taken as 1. The proposed EDE-1 and EDE-2 variants are evaluated on a set of 6 benchmark problems and a practical problem of minimizing the Molecular Potential Energy.

Numerical results show that out of the proposed variants, the convergence rate of EDE-2 is around 35% faster than the basic DE for the benchmark problems. Similarly for the real life problem also, it can be seen that EDE2 converges much faster than the SDE.

The proposed work can be extended in several directions. In future we plan to test our algorithm on more complex benchmark and real life problems and to compare EDE-1 and EDE-2 with other variants of DE.

REFERENCES

[1] R. Storn. and K. Price.: "Differential evolution—A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces". Berkeley, CA, Tech. Rep. 1995 TR-95-012

[2] R. Storn and K. Price, "Differential Evolution - A simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, 11, 1997, pp. 341-359.

[3] K. Price, "An introduction to DE". In: D. Corne, M., Dorigo. and F. Glover, (eds.): New Ideas in Optimization. McGraw-Hill, London (UK), 1999, pp. 78–108

[4] R. Storn. and K. Price,"Minimizing the real function of the ICEC'96 contest by DE", IEEE International Conference on Evolutionary Computation, Nagoya, 1996. pp. 842–844

[5] E. M. Montes and J. V. Reyes, " A comperative study of differential evolution variants for global optimization", GECCO, Seattle Washington USA, 2006, pp 485-492.

[6] M. Pant, M. Ali and A. Abraham, "Mixed mutation strategy embedded differential evolution", IEEE Congress on Evolutionary Computation, 2009, pp 1240-1246.

[7] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search", IEEE Transactions on Evolutionary Computation, vol. 12, 2008, pp 107-125.

[8] H. Y. Fan, J. Lampinen and G S.Dulikravich, "Improvements to mutation donor formulation of differential evolution", International Congress on Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems Eurogen, 2003.

[9] C. Lavor and N. Maculan, "A function to test methods applied to global minimization of potential energy of molecules"; Numerical Algorithms (Springer), 35, 2004, pp 287–300.

[10] C. D. Maranas and C. A. Floudas, "A deterministic global optimization approach for molecular structure determination"; Journal of Chemical Physics (A I P), 100(2), 1994, pp 1247-1261.

[11] M.Ali, M. Pant and A. Abraham, "Simplex differential evolution," Acta Polytechnica Hungarica, vol 6, 2009, pp 95-115.

[12] S. Das, A. Abraham, U. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator", IEEE Trans Evol Comput 13(3), 2009, pp-526–553.

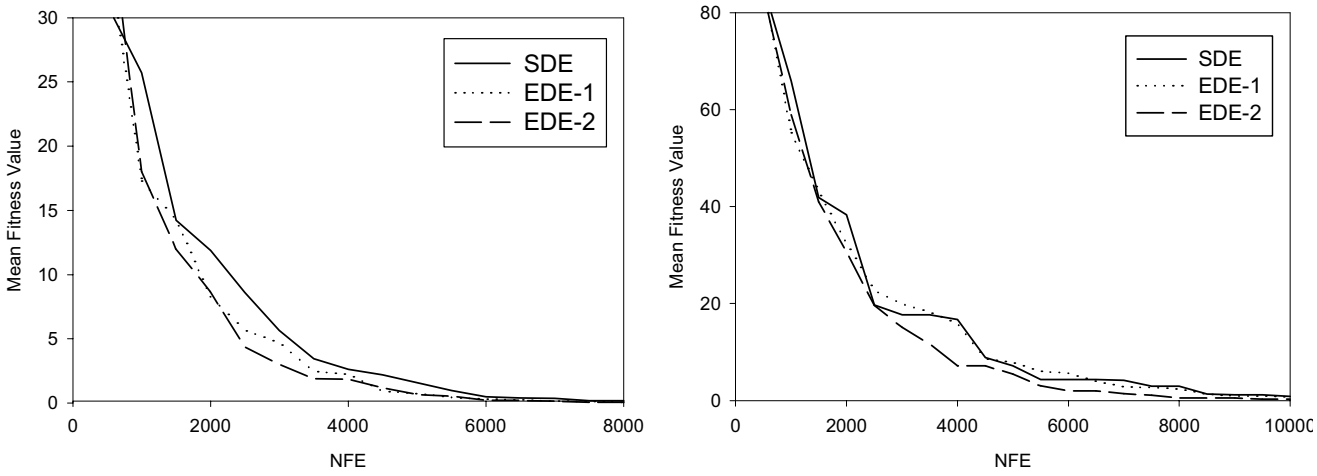

Figure-1 Convergence Graph of Sphere Funtion for D=15 and D=25.

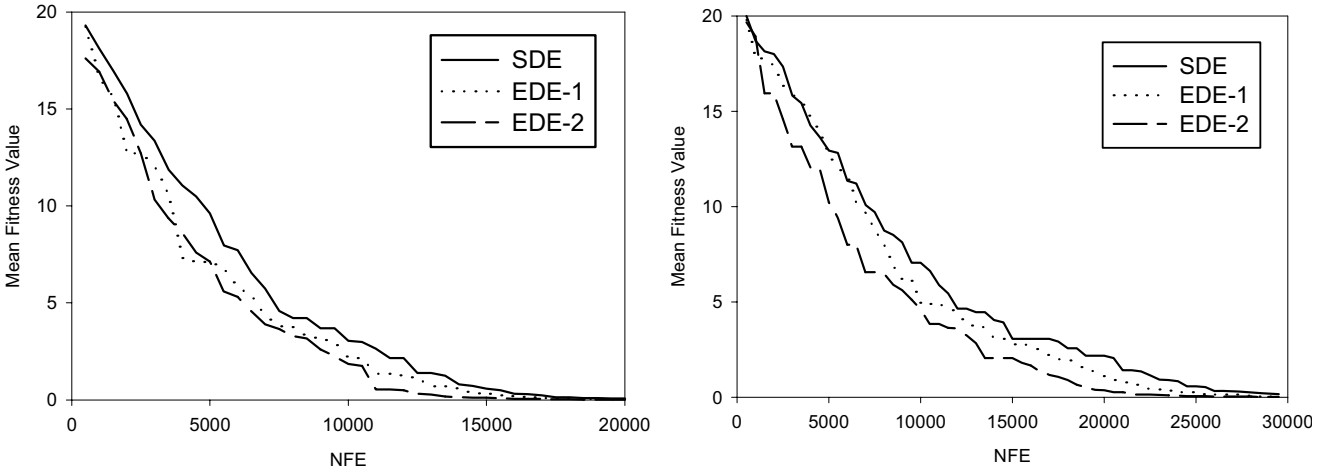

Figure-2 Convergence Graph of Ackley Function for D=15 and D=25.

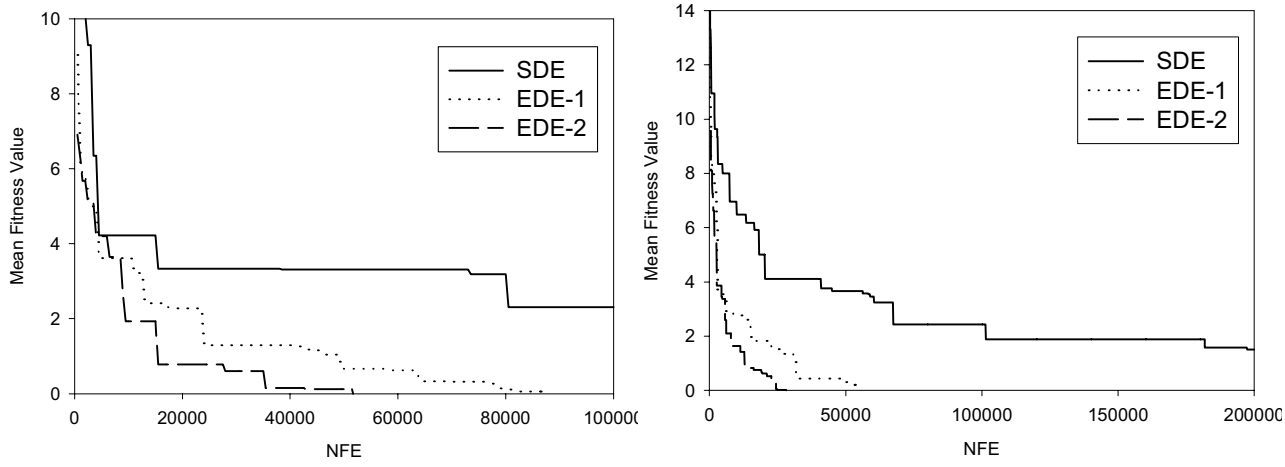

Figure 3 Convergence Graph of Molecular Potential Energy Problem for D=20 and D=25