A Comparative Study of Differential Evolution Algorithm for Global Numerical Optimization

D. K. Mishra[1], Vikas Shinde[2],
[1,2]Department of Applied Mathematics
Madhav institute of technology & Science, Gwalior (M.P.)-474005
Email:mishradilip3826@yahoo.com, v_p_shinde@rediffmail.com

H.M. Dubey[3]
[3]Department of Electrical engineering
Madhav institute of technology & Science, Gwalior (M.P.)-474005
Email:harimohandubeymits@gmail.com

Abstract Differential Evolution (DE) algorithm is employed to evaluate the performance of various functions. We carried out the performance on uni-modal as well as multi-modal functions. The better results have been obtained and compared with ABC, GA and PSO. Comparison of result in terms of statistical parameter provided in recent literature, it is observed that result obtained by DE is found to be superior.

Keywords: DE, Mutation, Cross-over, Selection

1. INTRODUCTION

Differential Evolution is an effective and strong population based stochastic search technique for solving global optimization problems over continuous space, which has been widely applied in many scientific and engineering problems. However the success of DE to handle a specific problem specially depends on the proper choice of a few parameters including the size of population, crossover, mutation and selection. Differential Evolution has been used in a wide spectrum of science and engineering problems. Many researchers employed such algorithm for achieving various numerical and combinatorial optimization solutions in diverse fields. [Storm and Price,(1997)] invented differential evolution (DE) algorithm for solving the optimization problems without using its gradient. [Fan and Lampinen, (2003)] proposed new mutation rules to enhance the local search ability of DE or to overcome the problems of stagnation or premature convergence. [Ali and Torn, (2004)] established the mutation scaling factor; F should be diversified at early stages and intensified at later stages. [Liang et al, (2005)] determined an appropriate population size strategy and its associated parameter values at different stages of evolution / search process. [Wang and Zhang, (2007)] considered stochastic properties of chaotic systems are used to spread the individuals in the search spaces as much as possible. [Das et al, (2009)] presented a detail review of the basic concepts of DE and survey of its major variants its application to multi - objective constrained large scale and uncertain optimization problems. [Karaboga and Akay, (2009)] established a comparative study of Artificial Bee Colony Algorithm. [Mallipedi and Suganthan, (2009)] proposed a DE algorithm with an ensemble of parallel populations in which the number of function evaluation allocated to each population is self adapted by learning from their previous experiences in generating superior solutions. [Qin et al, (2009)] proposed a self adaptive DE (SaDE) algorithm, in which both trial vector generation strategies and their associated control parameter values are gradually self adapted by learning from their previous experiences in generating promising solutions. [Mohamed et al, (2012)] proposed an alternative differential evolution (ADE) algorithm for solving unconstrained global optimization problems.

The remaining paper is arranged as follows. In section 2, we provide a brief description of Differential Evolution (DE). Benchmark problem are given in section 3. Section 4 deals comparison of algorithm. Conclusion is given in section 5.

2. DIFFERENTRIAL EVOLUTION ALGORITHM

Differential Evolution (DE) is a vector population based stochastic optimization method which has been introduced in 1955 by Storm and Price. There are four control parameters in DE algorithm i.e population, mutation, crossover and selection. DE is able to optimize objective function that can modeling of the problems with incorporating constraints. It has three advantages finding the true global minimum regardless of the initial parameter values, fast convergence and using a few control parameters. This main operation is based on the differences of randomly sampled pairs of solutions in the population. DE uses Mutation operation as a search mechanism and Selection operation to direct the search toward the prospective regions in the search space. This also uses a non- uniform crossover that can take child vector parameter from one parent more often than it does from others. In DE population of NP solution vector is randomly created at the beginning. This population is successfully improved by applying Mutation, Crossover and Selection operations.

2.1 ***Population***

Differential Evolution is a population based optimization method. The population includes Np individuals Y_{ig} is the i^{th} individuals of g^{th} generation of the population. It is selected randomly in differential evolution.

2.2 ***Mutation***

There are different techniques for Mutation of individuals in differential evolution. In general, it can be described as follows:

$$\bar{M}_{i,g} = \bar{Z}_{i,g} + \frac{P}{N}\sum_{n=0}^{n-1}(\bar{Y}_{s(2n+1),g} - \bar{Y}_{s(2n+2),g}) \quad (1)$$

where $\bar{Z}_{i,g}$ is the base vector and P is a constant parameter called Mutation scale factor and subscripts show that the individual is chosen randomly in the population . There are four mutation techniques. These are:

DE/rand/1/bin $\bar{M}_{i,g} = \bar{Y}_{s,g} + P(\bar{Y}_{s_{,1}g} - \bar{Y}_{s_2,g})$ (2)

DE/Best/1/bin $\bar{M}_{i,g} = \bar{Y}_{best,g} + P(\bar{Y}_{s_1,g} - \bar{Y}_{s_2,g})$ (3)

DE/Current to best/1/bin $\bar{M}_{i,g} = \bar{Y}_{i,g} + P(\bar{Y}_{best,g} - \bar{Y}_{i,g}) + P(\bar{Y}_{s_{,1}g} - \bar{Y}_{s_2,g})$ (4)

DE/best/2/bin $\bar{M}_{i,g} = \bar{Y}_{best,g} + P(\bar{Y}_{s_1,g} - \bar{Y}_{s_2,g} + \bar{Y}_{s_3,g} - \bar{Y}_{s_4,g})$ (5)

where $\overline{Y}_{best,\,g}$ is the individual. It has the best fitness in the population.

2.3 ***Crossover***

The parent vector is involved with the mutated vector to generate a trial vector. It can be described as

$$U_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} & if \quad s_j \le Cs \ or \ j = jrand \quad j = 1,2,3...n \\ q_{j,i,g} & if \quad s_j > Cs \end{cases} \quad (6)$$

Where s_j is a uniformly distributed random number, $s_j \in [0\ 1]$, Cs is crossover constant $\in [0\ 1]$

2.4 ***Selection***

Finally we choose the better individual for the optimization of the objective function $F(\overline{X})$. This method can be described as:

$$\overline{Y}_{i,\,g+1} = \begin{cases} U_{i,g} & if \quad f(\overline{U}_{i,g}) \le f(\overline{Y}_{i,g}) \\ Y_{i,g} & if \quad f(\overline{U}_{i,g}) > f(\overline{Y}_{i,g}) \end{cases} \quad (7)$$

In the selection method after the mutation and crossover, we get child then the child's vector performance and its parent is compared, it is selected better one. If the parent is quite better, it is held in the population.

3. BENCHMARK PROBLEM

The efficiency of the algorithm depends on it's ability how to solve the diverse set of problems. In this paper we tested the performance of random DE on a set of six standard benchmark problems with box constraints. We have included two uni-modal (having single optimum) functions and four multimodal (having several local and global optimal) function. The measures of the common parameters applied in each algorithm such as population size and number of iteration were selected to be same. Population size was 50. The other specific parameters of algorithms are depicted in table 1.

Table 1: Benchmark problems used in this article

Name	Test Function	Nature of function	Search Space	F_{min}
Ackley($f_1(x)$)	$f_1(x) = -20e^{-0.02\sqrt{D^{-1}\sum_{i=1}^{D} x_i^2}} - e^{D-1\sum_{i=1}^{D}\cos(2\pi x_i)} + 20 + e$	Continuous, Differentiable, Non separable, scalable, multi-model	[-32 32]	*0*
Beale ($f_2(x)$)	$f_2(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1{x_2}^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	Continuous, Differentiable, Non-separable, Non-scalable, multi-model	[-4.5 4.5]	*0*
Goldstein-price ($f_3(x)$)	$f_3(x)$ =[1+($x_{1+}x_2 + 1)^2$(19-14x_{1+}3x_1^2-14x_2+6x_1x_2+3x_2^2)]*[30+$(2x_1 - 3x_2)^2$(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]	Continuous, Differentiable, Non-separable, Non-scalable, multi-model	[-2 2]	*0*
Booth($f_4(x)$)	$f_4(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	Continuous, Differentiable, Non-separable, Non-scalable, uni-model	[-10 10]	*0*
Easom ($f_5(x)$)	$f_{5(x)=}$ –cos(x_1)cos(x_2)exp[-($x_1 - \pi$)2-(x_2-π)2]	Continuous, Differentiable, separable, Non-scalable, multi-model	[-100 100]	*-1*
Matyas ($f_6(x)$)	$f_6(x)$ =0.26(x_1^2+x_2^2)-0.48x_1x_2	Continuous, Differentiable, Non-separable, Non-scalable, Uni-model	[-10 10]	*0*
Schwefel ($f_7(x)$)	f_7(x)=-$\sum_{i=1}^{n}\|x_i\|$	Continuous, Non-Differentiable, Separable, Scalable ,Uni modal	[-10 10]	*0*
Sphere ($f_8(x)$)	f_7(x)=$\sum_{i=1}^{n} x_i^2$	Continuous, Differentiable, Separable, Scalable, Multimodal	[-100 100]	*0*

4. NUMERICAL RESULT AND COMPARISON OF ALGORITHM

The proposed algorithms are compared with the DE in terms of average fitness function value and standard deviation as well as best optimal result. We considered these common criteria to test the efficiency and robustness of an algorithm.

Table 2 gives the average fitness function value and standard deviation. Where fitness function values interpret the optimum solution carried out by the algorithm, standard deviation interprets about the stability of the algorithm. In table 3 and table 4, we vary various parameters such as population size, no. of generation, Scaling Factor, etc,.

We have also observed that all the algorithms gave more or less similar results which are in the vicinity of the true optimal solution of the problem. We were particularly point out the performance of DE algorithm and compare with other algorithm.

Table 2: Comparison Values of Various Fitness Function

Function	**Dimension**		**Fitness**			
			DE	ABC [3]	GA[3]	PSO[3]
$f_1(x)$	30	Best	0.0000	0.0000	0.0000	0.0000
		Mean	0.0000	0.0000	14.67178	0.164622
		SD	0.0000	0.0000	0.178141	0.493867
$f_2(x)$	2	Best	0.0000	0.0000	0.0000	0.0000
		Mean	0.0000	0.0000	0.0000	0.0000
		SD	0.0000	0.0000	0.0000	0.0000
$f_3(x)$		Best	0.0000	3.0000	5.2506	3.0000
		Mean	0.0000	3.0000	5.2506	3.0000
	2	SD	0.0000	0.0000	5.8700	0.0000
$f_4(x)$	2	Best	0.0000	0.0000	0.0000	0.0000
		Mean	0.0000	0.0000	0.0000	0.0000
		SD	0.0000	0.0000	0.0000	0.0000
$f_5(x)$	2	Best	0.0000	-1.0000	-1.0000	-1.0000
		Mean	-1.0000	-1.0000	-1.0000	-1.0000
		SD	0.0000	0.0000	0.0000	0.0000
$f_6(x)$	2	Best	0.0000	0.0000	0.0000	0.0000
		Mean	0.0000	0.0000	0.0000	0.0000
		SD	0.0000	0.0000	0.0000	0.0000
$f_7(x)$	30	Best	0.0000	0.0000	0.0000	0.0000
		Mean	0.0000	0.0000	11.0214	0.0000
		SD	0.0000	0.0000	1.3868	0.0000
$f_8(x)$	30	Best	0.0000	0.0000	0.0000	0.0000
		Mean	0.0000	0.0000	1.11E+03	0.0000
		SD	0.0000	0.0000	74.2144	0.0000

Table 3: Values of Various Fitness Function
Pop size=300, no. of Generation=150, run=10, Scaling Factor F1 (0.5), Scaling Factor F2 (0.5), CR=0

Function	**Standard Deviation**	**Mean**	**Best**	**Min**	**Max**	**Elapsed time**
Ackley	0.0014	8.6482E^-004	0	0	0.0069	76.008615sec
Beale	0.0028	0.0026	0	0.014162	0.0142	71.97287sec
Booth	3.2837E^-005	2.7421E^-005	0	8E-08	1.2104E^-004	74.164174sec
Goldtein price	2.2204E^-016	1.8	1.8	1.8	1.8	73.139817sec
Easom	1.0402	0.5872	0	1.037E-05	5.788	73.026135sec
Matyas	8.7680E^-007	5.5800E^-007	0	0	4.6100E^-006	74.084036
Schwefel	1.8219E^-005	1.1473E^-005	0	0	9.074E^-005	74.07601sec
Sphere	4.638E^-006	2.82E^-006	0	0	2.6010E^-005	67.862180sec

Table 4: Values of Various Fitness Function

Pop size=400, no. of Generation=200, run=10 Scaling Factor F1 (0.4), Scaling Factor F2 (0.3), CR=0.9

Function	**Standard Deviation**	**Mean**	**Best**	**Min**	**Max**	**Elapsed time**
Ackley	0	0	0	0	0	140.423884sec
Beale	0	0	0	0	0	140.027379sec
Booth	0	0	0	0	0	139..325052sec
Goldtein price	0	3	3	3	3	137.407998sec
Easom	2.2204E^-016	-1	-1	-1	-1	136.293305sec
Matyas	0	0	0	0	0	134.4091sec
Schwefel	8.8818e^-016	-7.8906	-7.8906	-7.8906	-7.8906	138.40437sec
Sphere	0	0	0	0	0	130.623212sec

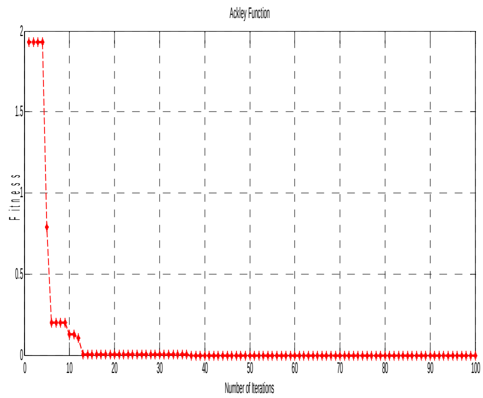

Figure 1

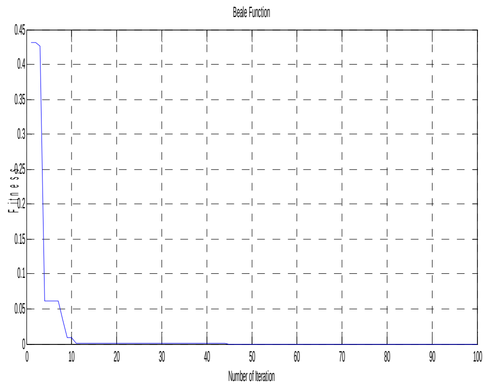

Figure 2

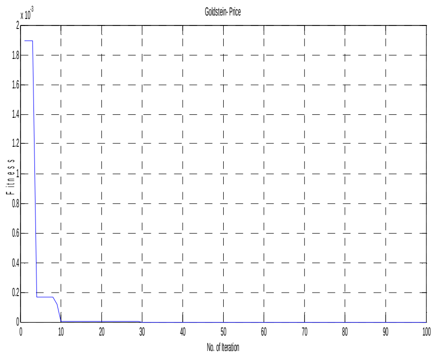

Figure 3

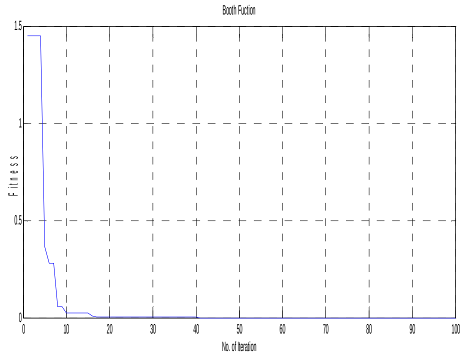

Figure 4

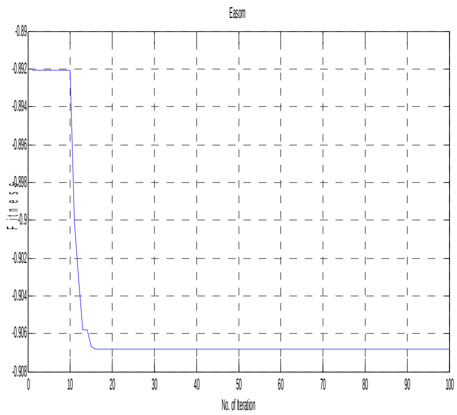

Figure 5

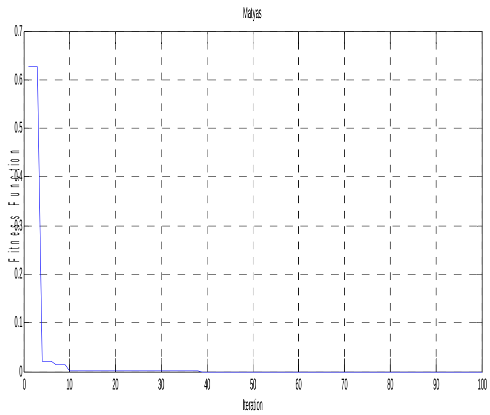

Figure 6

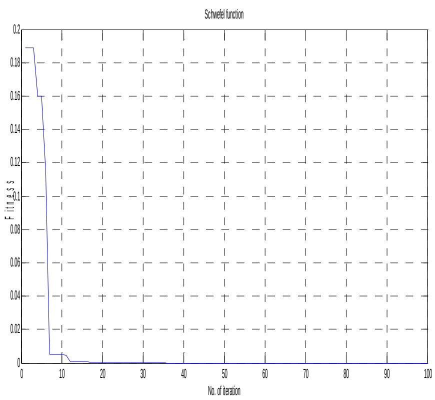

Figure 7

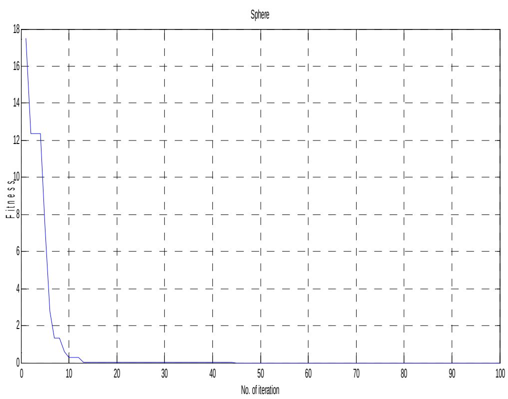

Figure 8

5. CONCLUSION

In this investigation, we have compared the Performance of DE algorithm with others. We have observed that DE algorithm is more effective to get global minimum of initial parameter values, significant convergence and using a few control parameters. It has been also noticed that the convergence rate of DE is better than ABC, GA, PSO. Finally, DE is more promising algorithm to obtain/ determine the solution of numerical optimization problem.

References

[1] Ali, M. M and Torn, A., Torn (2004): Population set based global optimization algorithms : some modifications and numerical studies, Computer Operation Research, 31(10), pp.1703-1725.

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

[3] Fan, H.Y.and Lampinen J., (2003): A trigonometric mutation operation to differential evolution, Global Optimization, 27(1), pp. 105-129.

[4] Gen M. andCheng, A., (2000): Genetic Algorithms and Engineering optimization, John Wiley and Sons, Inc.

[5] ,Karaboga D. and Akay B., (2009): A comparative study of Artificial Bee Colony Algorithm, Applied Mathematics and Computation, 214, pp. 108-132.

[6] Liang, J. J., Suganthan, P.N. and Deb, K., (2005): Novel composition test functions for numerical global optimization, IEEE Swarm Intelligence Symposium, Pasadena, California. pp .68-75.

[7] Mallipedi, R. and Suganthan, P. N., (2009): Differential evolution algorithm with ensemble of populations for global numerical optimization, OPSEARCH, 46(2), pp. 184-213.

[8] Mohm.ed, A. W., Sabery, H..Z. and .Khorshid, M., (2012): An alternative differential evolution algorithm for global optimization, Advanced Research, 3, pp. 149-165.

[9] Price, K..V., Storn, R. and Lampinen, J. A. (2005): Differential evolution a practical approach to global optimization, Springer- verlag Berlin Heidelberg.

[10] Qin, A. K,, Herang V. L., and Suganthan, P. N. (2009): Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Computer, 3(2), pp. 398-417.

[11] Rao S. S., (2009): Engineering optimization theory and practice, 4th edition, John Wiley and sons, Inc., Hoboken, New Jersey.

[12] . Storn, R and Price,K. V., (1997): Differential evolution- A simple and efficient heuristic for global optimization over continuous spaces , Global Optimization, 11, pp. 341-359.

[13] . Wang, Y. J and. Zhang, J. S. (2007): Global optimization by an improved differential evolutionary algorithm, Applied Mathematics Computer,188 (1), pp.669-680.