An Empirical Comparison of Differential Evolution Variants for High Dimensional Function Optimization

G.Jeyakumar[1]
C.Shanmugavelayutham[2]
[1, 2] Assistant Professor
Department of Computer Science and Engineering
Amrita School of Engineering
Amrita Vishwa VidyaPeetham
Coimbatore, Tamil Nadu, India
[1]g_jeyakumar@ettimadai.amrita.edu
[2]cs_velayutham@ettimadai.amrita.edu

Abstract—**In this paper, we present an empirical comparison of some Differential Evolution (DE) variants to solve high dimensional optimization problems. The aim is to identify the behavior and scalability of DE variants. Most studies on DE are obtained using low-dimensional problems (smaller than 100) , which are relatively small for many real-world problems. We have chosen four problems grouped by feature: unimodal and separable, unimodal and nonseparable, multimodal and separable, and multimodal and nonseparable. Fourteen variants were implemented and tested on four benchmark problems for dimensions of 30, 100, 500 and 1000. The value for the parameter CR is decided based on a bootstrap test conducted for 30 dimensions, and the same CR value is adopted for the dimensions 100, 500 and 1000 also. The analysis is done based on the results obtained for 100 runs, for each Variant-Function-Dimension combination.**

Keywords - Differential Evolution, High Dimensional Function Optimization

I. Introduction

Evolutionary algorithms (EA) have been widely used to solve optimization problems. Differential Evolution [1] is an EA proposed to solve optimization problems, mainly to continuous search spaces. The DE algorithm, a stochastic population-based search method, has been successfully applied to many global optimization problems [2]. As traditional EAs, several optimization problems have been successfully solved by using DE [3]. It shows superior performance in both widely used benchmark functions and real-world application [4]. DE shares similarities with traditional EAs. As in other EAs, two main processes that derive the evolution are the perturbation process (crossover and mutation) which ensures the exploration of the search space and the selection process which ensures the exploitation properties of the algorithm. Both perturbation and the selection process are simpler than those used in other evolutionary algorithms. In the case of DE, the perturbation of a population element is done by probabilistically replacing it with an offspring obtained by adding to a randomly selected element a perturbation proportional with the difference between other two randomly selected elements. The selection is done by one to one competition between the parent and its offspring.

There are three strategy parameters in DE, the population size NP, the crossover rate CR and the scaling factor F. Many works have been done to study the suitable setting of these control parameters [5, 6]. The CR parameter controls the influence of the parent in the generation of the offspring. The F parameter scales the influence of the set of pairs of solutions selected to calculate the mutation value

DE performs the perturbation based on the distribution of the solutions in the current population. In this way, search directions and possible step sizes depend on the location of the individuals selected to calculate the mutation values.

Based on different strategies followed for perturbation, there are various DE variants are exists, they differ in the way how the solution is generated. Besides the suitable setting of control parameters, another important factor when using DE is the selection of the variant. The most popular variant of DE is *DE/rand/1/bin*. There is a nomenclature scheme developed to reference different DE variants. In DE/rand/1/bin, “DE” means Differential Evolution, the word “rand” indicates that the individuals selected to compute the mutation values are chosen at random, “1” is the number of pairs of individuals chosen and finally “bin” means that a binomial crossover is used.

The algorithm for *DE/rand/1/bin* is presented in the “Fig. 1”

```
1. Begin
2.  G=0
3.  Create random initial population $X_{i,G}$ for every i,  i =1.....NP
4.  Evaluate  $f(X_{i,G})$  for every i, i=1.....NP
5.  For G = 1 to MAXGEN Do
6.      For i = 1 to NP Do
7.♦             select randomly $r1 \neq r2 \neq r3$
8. ♦            $j_{rand}$=randint (1,D)
9. ♦            For j=1 to D Do
10. ♦               If($rand_j[0,1)<CR$ or $j=j_{rand}$) Then
11. ♦                   $U_{i,j,G+1}=X_{r3,j,G}+F(X_{r1,j,g}-X_{r2,j,G})$
12. ♦               Else
13. ♦                   $U_{i,j,G+1}=X_{i,j,G}$
14. ♦               End If
15. ♦           End For
16.             If  $f(U_{i,G+1}) \leq f(X_{i,G})$ Then
17.                 $X_{i,G+1} = U_{i,G+1}$
18.             Else
19.                 $X_{i,G+1} = X_{i,G}$
20.             End If
21.     End For
22.     G=G+1
23.  End For
24. End
```

Figure 1: "DE/rand/1/bin" algorithm, steps pointed out with ♦ will change from variant to variant.

II. Related Work

However most of the experimental results on DE are obtained using low-dimensional problems, the reported studies on the scalability of DE derivative algorithms are scarce. In contrast. Other evolutionary algorithms such as evolutionary programming (EP), have been tested on high-dimensional problems, up to 1000 dimension [7].

Cooperative co-evolution architecture was firstly proposed by Potter for genetic algorithm, called CCGA [8], and had been successfully applied to other evolutionary algorithms [7, 9, and 10]. In the context of DE, the cooperative co-evolution has also been introduced, and it was proposed as CCDE [11]. However, CCDE only extended the problem domain up to 100 dimensions, which are relatively small for many real-world problems.

Zhenyu Yang, Ke Tang and Xin Yao, proposed two new DE variants named DECC-I and DECC-II for high dimensional optimization, up to 1000 dimension [12]. These two algorithms are based on a cooperative coevolution framework.

III. Design of Experiment and Results

We selected four variants *DE/rand/1/bin*, *DE/rand/1/exp*, *DE/best/1/bin* and *DE/best/1/exp* with one pair of individual for mutation. The "*rand*" variants select all the individuals to compute mutation at random and the "*best*" variants use the best solution in the population besides the random ones

We selected the following four variants with two pair of individuals *DE/rand/2/bin, DE/rand/2/exp, DE/best/2/bin* and *DE/best/2/exp.* Also, we selected the following six variants *DE/current-to-rand/1/bin, DE/current-to-best/1/exp, DE/current-to-best/1/bin, DE/current-to-best/1/exp, DE/rand-to-best/1/bin* and *DE/rand-to-best/1/exp.*

We have chosen four test functions [13] grouped by the feature *f01*-Schwefel's problem 2.21(unimodal and separable), *f02*-Schwefel's Problem 1.2(unimodal and nonseparable), *f03*-Generalized Rastrigin's Function(multimodal and separable) and *f04*-Ackely's Function(multimodal and nonseparable). The details of the functions are presented in the Appendix. All the functions have its optimum value at zero.

The experimental design consists on testing the, above mentioned, fourteen variants on the four problems, mentioned earlier, with different dimensions (D = 30, 100, 500 and 1000).

We fixed the population size (NP) as 100, a large population affects the ability of the approach to find the correct search direction , so we fixed a moderate population size in all the experiments. We fixed the maximum number of function evaluations as proportional to the dimension, which is equal to D * 5000. We also fixed the stopping criteria as an error values of 1×10^{-12} . The algorithms either will stop at maximum function evaluation or if the tolerance error is reached.

Based on [14], we decided a range for the parameter F as [0.3, 0.9], this value is generated anew at each generation. The same F value is assigned to K, which is used for mutation.

The CR parameter is tuned for each Variant-Function combination. 11 different values for the "CR" parameter were tested {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0} for each Variant-Function combination with dimension D=30. We conducted 50 independent runs for each combination of Variant-Function-CR value. Based on the obtained result a bootstrap test was conducted for each combination in order to determine the confidence interval for the mean objective function value. The "CR" value corresponding to the best confidence interval was chosen to be used in our experiment and the CR value chosen in dimension 30 is adopted for D=100, 500 and 1000 also. The variants along with the CR values for each function are presented in Table I.

We initialized the population with uniform random initialization with in the search space and 100 independent runs were performed for each Variant-Function-Dimension combination. We recorded, mean objective function value (MOV) for 100 runs.

For easier analysis the results were grouped by dimension and by function. In Table II we present the mean objective function value obtained for the function *f01, f02, f03* and *f04*. The values with bold and underline are the least objective function value obtained for each Function-Variant-Dimension combination. In Table III the overall performance analysis of the variants is presented. It presents, for each Function-Dimension combination the best mean objective function values and the corresponding variants. The least mean objective function value for each Function-Variant-Dimension combination is marked with bold and underlines style.

TABLE I

"CR" VALUE FOR EACH PAIR OF VARIANT-FUNCTION

No	**Variant**	***f01***	***f02***	***f03***	***f04***
V1	DE/rand/1/bin	0.9	0.9	0.1	0.9
V2	DE/rand/1/exp	0.1	0.1	0.1	0.1
V3	DE/best/1/bin	0.8	0.3	0.1	0.1
V4	DE/best/1/exp	0	0.1	0.1	0.2
V5	DE/rand/2/bin	0.9	0.9	0.1	0.3
V6	DE/rand/2/exp	0	0.1	0.1	0.1
V7	DE/best/2/bin	0.1	0.8	0.1	0.1
V8	DE/best/2/exp	0.1	0.1	0.1	0.1
V9	DE/current-to-rand/1/bin	0.3	0.9	0.1	0.1
V10	DE/current-to-rand/1/exp	0	0	0.1	0.1
V11	DE/current-to-best/1/bin	0.2	0.9	0.1	0.1
V12	DE/current-to-best/1/exp	0	0	0.1	0.1
V13	DE/rand-to-best/1/bin	0.9	0.9	0.9	0.3
V14	DE/rand-to-best/1/exp	0.1	0.1	0.1	0.1

IV. DISCUSSION

A. *f01 : Unimodal Separable(Schwefel's problem 2.23)*

In dimension 30 for function *f01*, comparatively the best results were provided by the variants *rand-to-best/1/bin* and *rand/1/bin*, they both reached the near optimum value with the average objective function value of 0.78 and 0.91 respectively. The poorest performance was provided by the variants *current-to-rand/1/exp* and *current-to-best/1/exp* with the average objective function value of 55.74 and 55.75 respectively.

In dimension 100,500 and 1000, the results show that it was not solved by any variant. In dimension 100, comparatively the best results were provided by the variants *best/2/bin* and *rand/1/bin*, with the average objective function value of 30.11 and 30.12 respectively. The poorest performances were provided by the variants *current-to-rand/1/exp* and *rand/2/exp* with the average objective function value of 94.5 and 94.49 respectively.

In dimension 500, comparatively the best results were provided by the variants *best/1/exp* and *best/1/bin* with the average objective function value of 41.84 and 64.15 respectively. The poorest performance was provided by *rand/2/bin* with the average objective function values of 98.63.

In dimension 1000, comparatively the best results were provided by the variant *best/1/exp* with the average objective function value of 42.79. The poorest performance was provided by *rand-to-best/1/bin* with the average objective function value of 98.85. This result suggests that variants with "best" selection scheme giving good performance than other variants.

The obtained results showed that variants with binomial recombination type and "best" selection scheme are most competitive in reaching the global optimum.

B. *f02: Unimodal NonSeparable(Schwefel's Problem 1.2)*

The result for the function *f02* in dimension 30 shows that the best results were provided by the variants *best/2/bin* and *best/2/exp*, they both reached the global optimum of 0. The poorest performances were provided by the variant *current-to-best/1/bin* and *current-to-rand/1/bin* with the average objective function value of 13749.3124 and 13937.6965 respectively

In dimension 100, 500 and 1000 the result shows that it was not solved by any variant. In dimension 100, comparatively the best results were provided by the variant *best/2/bin* with the average objective function value of 2248.4262. The poorest performance was provided by *current-to-best/1/bin* and *current-to-rand/1/bin* with the

TABLE II

THE MEAN OBJECTIVE FUNCTION VALUES OBTAINED FOR THE FUNCTION f01, f02, f03 AND f04 IN 100 RUNS

Function – *f01*

Variant	D = 30	D = 100	D = 500	D = 1000
1	0.91	30.12	98.53	98.78
2	17.82	90.95	98.33	98.72
3	28.72	50.09	64.15	64.62
4	25.96	34.71	**41.84**	**42.79**
5	3.34	94.47	98.63	98.8
6	7.13	94.49	98.52	98.78
7	2.6	**30.11**	88.26	89.65
8	9.09	89.61	98.3	98.67
9	18.83	81.3	98.55	98.77
10	55.74	94.5	98.5	98.82
11	17.05	72.85	98.45	98.82
12	55.75	30.17	98.57	98.84
13	**0.78**	91.35	98.46	98.85
14	17.98	91.35	98.34	98.62

Function – *f02*

Variant	D = 30	D = 100	D = 500	D = 1000
1	0.55	7061.26	**378461.24**	1302355.44
2	5.71	239074.44	15216875.17	14342315.34
3	0.78	10464.86	854930.98	3191306.42
4	364.74	144233.48	14870737.35	79492389.71
5	1073.51	218155.96	6632782.433	19785758.6
6	1049.47	311891.24	15328136.09	68431518
7	**0**	**2248.43**	391863.94	1558474.6
8	**0**	209909.29	15183598.61	71901345.6
9	13937.69	667057.83	16347074.17	63430502.4
10	1262.38	440334.78	8048956.17	28044382.29
11	13749.31	657134.55	15664122.22	62995994.8
12	1259.26	442178.83	7985850.685	28555998.2
13	0.54	6853.78	388102.09	**1167663.23**
14	5.54	246681.38	15443684.83	71024066

Function – *f03*

Variant	D = 30	D = 100	D = 500	D = 1000
1	**0**	1579.21	3782.37	17700.42
2	97.63	599.03	7836.49	9146.19
3	3.40	1575.65	**607.87**	17733.15
4	47.40	834.01	7721.86	10808.09
5	16.39	1580.49	4448.48	17729.09
6	161.05	769.45	7947.06	17741.79
7	0.62	1587.85	3920.91	17702.71
8	120.97	677.78	7943.69	9785.23
9	63.66	1586.25	7096.76	17679.43
10	249.82	1504.33	8343.67	17707.06
11	64.89	1586.12	7065.75	17714.14
12	245.38	1503.50	8338.84	17734.21
13	**0**	1583.53	3785.98	17681.53
14	100.97	**595.62**	7843.83	**9107.64**

Function – *f04*

Variant	D = 30	D = 100	D = 500	D = 1000
1	-0.09	1.93	9.05	12.73
2	0.0028	19.56	20.90	21.03
3	3.5209	3.47	2.06	2.65
4	7.3916	20.19	20.98	21.06
5	-0.09	**0.02**	16.88	20.89
6	4.0393	20.02	20.94	21.04
7	-0.09	-0.03	**1.76**	**1.52**
8	0.427	19.33	20.93	21.04
9	1.5784	4.20	20.72	21.00
10	15.6541	20.41	21.01	21.08
11	1.6877	4.25	20.71	20.99
12	15.4411	20.41	21.01	21.08
13	-0.09	0.48	4.54	10.23
14	**0.0008**	19.54	20.89	21.03

average objective function value of 657134.5514 and 667057.8295 respectively.

In dimension 500, comparatively the best results were provided by *rand/1/bin* with the average objective function values of 378461.23, the poorest performance were provided by *current-to-best/1/bin* and *current-to-rand/1/bin* with the average objective function value of 15664122.22 and 16347074.17 respectively.

In dimension 1000, comparatively the best results were provided by the variant *rand-to-best/1/bin* with the average objective function values 1167663.237. The poorest performance was provided by the variant *best/1/exp* with the average objective function value of 79492389.71

The obtained result showed that the variants with binomial recombination type and randomness in selection scheme provide competitive results.

TABLE III

BEST MEAN OBJECTIVE FUNCTION VALUES AND THE CORRESPONDING VARIANTS FOR EACH FUNCTION-DIMENSION COMBINATION

Function	D=30		D=100		D=500		D=1000	
	Best MOV	*Given By*	*Best MOV*	*Given By*	*Best MOV*	*Given By*	*Best MOV*	*Given By*
f01	0.78	V13	30.11	V7	41.84	V4	42.79	V4
f02	0	V7,V8	2248.4262	V7	378461.238	V1	1167663.24	V13
f03	0	V1,V13	595.6227	V14	607.8667	V3	9107.6366	V14
f04	0	V14,V2	0.0219	V5	1.7549	V7	1.522	V7

C. f03*: Multimodal Separable (Generalized Rastrigin's Function)*

The result for the function *f03* in dimension 30 shows that the best results were provided by the variant *rand-to-best/1/bin* and *rand/1/bin*, they both reached the global optimum of 0. The poorest performance was provided by the variant *current-to-rand/1/exp* with the average objective function value of 249.8184.

In dimension 100, 500 and 1000 the result shows that it was not solved by any variants. In dimension 100, comparatively the best results were provided by the variant *rand-to-best/1/exp* with the average objective function value of 595.6227, and the poorest performance was given by the variant *best/2/bin* with the average objective function value of 1587.8524.

In dimension 500, comparatively the best results were provided by the variant *best/1/bin* with the average objective function value of 607.8667. The poorest performances were provided by the variant current-to-best/1/exp and *current-to-rand/1/exp* with the average objective function values of 8338.8363 and 8343.671 respectively.

In dimension 1000, comparatively the best results were provided by the variant *rand-to-best/1/exp* with the average objective function values of 9107.6366. The poorest performance was given by the variant *rand/2/exp* with the average objective function value of 17741.7968.

The obtained result showed that variants with binomial recombination provide better performance than other variants.

D. *f04 : Multimodal Nonseparable(Ackely's Function)*

The result for the function *f04* in dimension 30 shows that the best results were provided by the variants *rand-to-best/1/exp* and *rand/1/exp* they both reached the global optimum of 0. The poorest performance was provided by the variant *current-to-rand/1/exp* with the average objective function values of 15.6541 respectively.

In dimension 100, 500 and 1000, the result shows that it was not solve by any variants In dimension 100, comparatively the best results were provided by the variants *best/2/bin* and *rand/2/bin*, these variants reached near the global optimum with the average objective function values of -0.03 and 0.0219 respectively. The poorest performance was provided by the variants *current-to-rand/1/exp* and *current-to-best/1/exp* with the average objective function values of 20.4166 and 20.4179 respectively.

In dimension 500, comparatively the best results were provided by the variant *best/2/bin* with the average objective function values of 1.7549. The poorest performances were provided by the variants *current-to-best/1/exp* and *current-to-rand/1/exp* with the average objective function values of 21.0113 and 21.0083 respectively.

In dimension 1000, comparatively the best results were provided by the variant best/2/bin with the average objective function values of 1.522. The poorest performances were provided by the variants *current-to-rand/1/exp* and *current-to-best/1/exp* with the average objective function values of 21.0844 and 21.0848 respectively.

The obtained result showed that the variants with binomial recombination type and randomness in selection are most competitive.

V. REMARKS

Based on the overall results, which combines the observation made on Table II and Table III, we highlight the following points.

The variant *best/2/bin* provides competitive results in all dimensions. The variants *rand-to-best/1/bin* and *rand-to-best/1/exp* are also quite well in all the dimensions. The poorest performance in all these function was provided by the variant *current-to-rand/1/bin.*

It is observed from the result that the variants with binomial recombination type are, comparatively, giving good performance than variants with exponential recombination type. And the variants with "best" selection scheme also more suitable..

VI. Conclusion

In this paper, we presented an empirical comparison of some DE variants to solve global optimization problems for various dimensions. Fourteen different variants were implemented and tested on 4 benchmark problems. Regardless of the characteristics and dimension the relatively better results seem to have been provided by the variants with binomial crossover and "best" selection scheme. Despite the fact that almost all the 14 variants did not scale up well to high dimension , it is worth noticing that the parameter CR was tuned for dimension 30 only and the same was adopted for high dimensional cases. Moreover, we feel that the scalability of DE variants can be analyzed further by testing the variants with more number of benchmark functions, which will give better insight.

References

[1] K.V.Price. "An Introduction to differential evolution," In D.Corne, M.Dorigo , and F.Glover, editors, New Ideas in Optimization, pages 79-108. Mc Graw-Hill, UK, 1999

[2] R.Storn and K.Price , "Differential Evolution – A Simple and Efficient Heuristic Strategy for Global Optimization and Continuous Spaces," Journal of Global Optimization. Vol . 11. pp 341-359, 1997

[3] K.Price, R.M.Storn , and J.A.Lampinen . Differential Evolution : A practical Approach to Global Optimzation. Springer-Verlag, 2005. ISBN : 3-540-20950-6.

[4] J. Vesterstrom and R.Thomsen. A Comparative Study of Differential Evolution Particle Swarm Optimization and Evolutionary Algorithm on Numerical Benchmark Problems. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC'2004), Volume 3 , Pages 1980-1987, June 2004

[5] D.Zaharie , "Critical values for the control parameters of Differential Evolution algorithms," Proc. of the 8th International Conference of Soft Computing , pp 62-67, 2002.

[6] R.Gamperle, S.D.Miller and P.Koumoutasakos , " A Parameter Study for Differential Evolution", Advances in Intelligent Systems , Fuzzy Systems , Evolutionary Computation , pages 293-298, 2002.

[7] Y.Liu, X. Yao, Q.Zhao and T.Higuchi, "Scaling Up Fast Evolutionary Programming with Cooperative Coevolution" Proc. of the Congress on Evolutionary Computation, pp. 1101-1108, 2001.

[8] A.M.Potter and K.A.De Jong . " A cooperative co-evolutionalry approach to function optimization " . Proc. of the Third International Conference on Parallel Problem Solving from Nature. Pp. 249-257. Springer-Verlag, 1994.

[9] D. Sofge, K. A. De jong, and A.Schultz, " A blended population approach to cooperative coevolution for decomposition of complex problems," Proc. of the Congress on Evolutionary Computation, pp. 413-418, 2002.

[10] Bergh, F.V.D and A. P. Engelbrecht, "A Cooperative Approach to Particle Swarm Optimization," IEEE Transactions On Evolutionary Computation, vol. 3 pp. 225-239. 2004.

[11] Y.Shi H.Teng and Z.Li . "Cooperative Co-evolutionary Differential Evolution for Function Optimization". Proc. of the First International Conference on Natural Computation. pp 1080-1088, 2005.

[12] Zhenyu Zang, Ke Tang and Xin Yao , Differential Evolution for High-Dimensional Function Optimization", Proc. of IEEE Congress on Evolutionary Computation (CEC 2007), pp 3523 -3530

[13] X.Yao, Y.Liu, K.H Liang and G.Lin . Fast Evolutionary Algorithms. In G.Rozenberg, T.Back, and A.Eiben , editors, Advances in Evolutionary Computing : Theory and Applications, pages 45-94, Springer-Verlag, New York, NY, USA, 2003.

[14] Efren Mezura-Montes , Jesus Velazquez-Reyes and Carios A.Coello Coello, A Comparative Study on Differential Evolution Variants for Global Optimization., GECCO'06, July 8-12, 2006.

Appendix- A

A .Test Functions

The details of the 4 test functions used in this paper are the following [13].

- *f01* - Schwefel's Problem 2.21

$$f_{sch} = max_i\{|x_i|, 1 \le i \le p\}$$

$$-100 \le x_i \le 100$$

$$x^* = (0,0..0); f_{sch}(x^*) = 0$$

- *f02* – Schwefel's Problem 1.2

$$f_{schDS} = \sum_{i=1}^{p}\left(\sum_{j=1}^{i} x_j\right)^2$$

$$-100 \le x_i \le 100$$

$$x^* = (0,0..0); f_{schDS}(x^*) = 0$$

- *f03* – Generalized Restraint's Function

$$f_{Ras}(x) = 10p + \sum_{i=1}^{p}\left(x_i^2 - 10\cos(2\pi x_i)\right)$$

$$-5.12 \le x_i \le 5.12$$

$$x^* = (0,0..0); f_{Ras}(x^*) = 0$$

- f04 – Ackley's Function

$$f_{Ack}(x) = 20 + e - 20exp\left(-0.2\sqrt{\frac{1}{p}\sum_{i=2}^{p} x_i^2}\right) - exp\left(\frac{1}{p}\sum_{i=1}^{p}\cos(2\pi x_i)\right)$$

$$-30 \le x_i \le 30$$

$$x^* = (0,0..0); f_{Ack}(x^*) = 0$$