An Improved Differential Evolution Algorithm for Real Parameter Optimization Problems

Musrrat Ali, Millie Pant, and V. P. Singh,
Department of Paper Technology,
Indian Institute of Technology Roorkee, Saharanpur Campus, Saharanpur – 247001, India.
musrrat.iitr@gmail.com, millifpt@iitr.ernet.in, singhvp2@yahoo.co.in.

Abstract: **Differential Evolution (DE) is a powerful yet simple evolutionary algorithm for optimization of real valued, multi modal functions. DE is generally considered as a reliable, accurate and robust optimization technique. However, the algorithm suffers from premature convergence, slow convergence rate and large computational time for optimizing the computationally expensive objective functions. Therefore, an attempt to speed up DE is considered necessary. This paper introduces an improved differential evolution (IDE), a modification to DE that enhances the convergence rate without compromising with the solution quality. In improved differential evolution (IDE) algorithm, initial population of individual is partitioned into several sub-populations, and then DE algorithm which utilize only one set of population instead of two as in original DE, is applied to each sub-population independently. At periodic stages in evolution, the entire population is shuffled, and then points are reassigned to sub-populations. The performance of IDE on a test bed of functions is compared with original DE. It is found that IDE requires less computational effort to locate global optimal solution.**

Keywords: **Differential evolution, shuffled complex evolution, optimization.**

I. Introduction

DE was proposed by Storn and Price [1] in 1995. It soon became a popular tool for solving global optimization problems because of several attractive features like having fewer control parameters, ease in programming, efficiency etc. DE is similar to GAs in the sense that it uses same evolutionary operators like mutation, crossover and selection for guiding the population towards the optimum solution. Nevertheless, it's the application of these operators, explained in Section II that makes DE different from GA. DE has been successfully applied to solve a wide range of real life application problems [2 - 3] and has reportedly outperformed other optimization techniques [4 –6]. Despite several positive features, it has been observed that DE sometimes does not perform as good as the expectations. Empirical analysis of DE has shown that it may stop proceeding towards a global optimum even though the population has not converged to a local optimum [7]. The situation when the algorithm does not show any improvement though it accepts new individuals in the population is known as stagnation. Besides this, DE also suffers from the problem of premature convergence due to loss of diversity in the population. The entire population converges to a point which may not even be a local optimal solution. It generally takes place when the objective function is multimodal having several local and global optima. Like other EA, the performance of DE deteriorates with the increase in dimensionality of the objective function. Several modifications have been made in the structure of DE to improve its performance [8 – 10]. In the present study we propose two modifications in the basic scheme of DE. The first modification is the concept of dividing the population into different subpopulations and the second modification is to use a single array in DE in contrast with the two arrays used in basic DE proposed by Storn and Price. The remaining of the paper is organised as follows; in Section 2, we give a brief description of DE. In Section 3, we describe the proposed DE versions. In Section 4, experimental settings and numerical results are given. The paper concludes with section 5.

II. Basic DE

As mentioned earlier in the previous section, there are several variants of the DE-algorithm [11]. Throughout the present study we shall follow the version *DE/rand/1/bin*, which is apparently the most commonly used version and shall refer to it as basic version. This particular scheme may be described as: DE starts with a population of NP candidate solutions which may be represented as $X_{i,G}$, $i = 1, \ldots, NP$, where index i denotes the population and G denotes the generation to which the population belongs. The working of DE depends on the manipulation and efficiency of three main operators; mutation, reproduction and selection.

Mutation:. The mutation operation of DE applies the vector differentials between the existing population members for determining both the degree and direction of perturbation applied to the individual subject of the mutation operation. The mutation process at each generation begins by randomly selecting three individuals in the population. The i^{th} perturbed individual, $V_{i,G+1}$, is then generated based on the three chosen individuals as follows:

$$V_{i,G+1} = X_{r3,G} + F * (X_{r1,G} \text{ “ } X_{r2,G}) \quad (1)$$

Where, $i = 1, \ldots, NP$, $r_1, r_2, r_3 \in \{1, \ldots, NP\}$ are randomly selected such that $r_1 \neq r_2 \neq r_3 \neq i$, F [0, 1+], F is the control parameter.

Crossover: The perturbed individual, $V_{i,G+1} = (v_{1,i,G+1}, \ldots, v_{n,i,G+1})$, and the current population member, $X_{i,G} = (x_{1,i,G}, \ldots, x_{n,i,G})$, are then subject to the crossover operation, that generates the population of candidates, or "trial" vectors, $U_{i,G+1} = (u_{1,i,G+1}, \ldots, u_{n,i,G+1})$, as follows:

$$u_{j,i.G+1} = \begin{cases} v_{j,i.G+1} \; if \; rand_j \leq C_r \vee j = k \\ x_{j,i.G} \qquad\qquad otherwise \end{cases} \quad (2)$$

DOI: 01.IJPIE.02.01.352

j = 1. . . n, k ∈ {1, . . . , n} is a random parameter's index, chosen once for each *i*, and the crossover rate, Cr ∈ [0, 1], is set by the user.

Selection: The population for the next generation is selected from the individual in current population and its corresponding trial vector according to the following rule

$$X_{i.G+1} = \begin{cases} U_{i.G+1} \; if \;\; f(U_{i.G+1}) \le f(X_{i.G}) \\ X_{i.G} \qquad\qquad otherwise \end{cases} \qquad (3)$$

Each individual of the temporary (trial) population is compared with its counterpart in the current population. The one with the lower objective function value will survive from the tournament selection to the population of the next generation. As a result, all the individuals of the next generation are as good as or better than their counterparts in the current generation. In DE trial vector is not compared against all the individuals in the current generation, but only against one individual, its counterpart, in the current generation.

III. Improved differential evolution

In the proposed work we have done two modifications (i) concept of shuffled complex evolution (SCE) [12] and (ii) DE using one set of individuals instead of two as in original DE, throughout of this paper we will call it DE1. The characteristics of SCE and principle of DE1 [13] are briefly explained. The concept of the SCE approach is to treat the global search as a process of natural evolution. The sampled points (NP in number) constitute a population which is partitioned into several complexes, each of which is permitted to evolve independently (i.e., search the space in different directions). After a certain number of generations, the complexes are forced to mix, to form new complexes through a process of shuffling. This procedure enhances survivability by sharing of information (about the search space) gained independently by each complex. Each member of a complex is a potential parent which can participate in the process of reproduction. To ensure that the evolution process is competitive, we require that the probability that better parents contribute to the generation of offspring is higher than that of worse parents. Finally, each new offspring replaces the worst point of the current complex rather than the worst point of the entire population. This ensures that every parent gets at least one chance to contribute to the reproduction process before being replaced or discarded. Thus, none of the information contained in the sample is ignored. The major difference between DE and DE1 is that DE1 maintains only one set of individuals. The set of individuals is updated as and when a better solution is found. Also, these newly found better solutions can take part in mutation and crossover operation in the current generation itself as opposed to DE (where another array is maintained and these better solutions take part in mutation and crossover operations in next generation). Updating the single set continuously enhances the convergence speed leading to less function evaluations as compared to DE which maintains two sets consuming extra memory and CPU-time (more function evaluations). This modification enables the algorithm to get a better trade off between the convergence rate and robustness. In IDE, the two concepts described above are merged together to form a new variant of DE. A population of individuals sampled randomly from the feasible space using uniform probability distribution. Then the population is sorted in increasing order of function values and partitioned in to several sub populations. Each sub population independently executes DE1. At periodic stage in the evolution, the sub population are forced to mix and points are reassigned to ensure information sharing. The working procedure of the algorithm is outlined below:

Step 1: Initialize the population set S uniformly.

Step 2: Sort the population set S in ascending order.

Step 3: Partition S into *p* sub populations $S^1, S^2, ...,S^p$, each containing m points, such that:

$$S^k = \{X_j^k, f_j^k : X_j^k = X_{k+p(j-1)}, f_j^k = f_{k+p(j-1)}, j = 1,...,m\} \; k = 1,...,p.$$

Step4: Apply DE1 to each sub population S^k to maximum number of generation G_{max}.

Step5: Replace sub populations $S^1, S^2, ...,S^p$ into S and check whether the termination criterion met if yes then stop otherwise go to step 2.

IV. Experimental settings and results

Based on the mechanism described in previous section, we propose four schemes based of the number of sub populations and the number of times DE is invoked in each sub population. Since the basic version of DE requires at least 4 points, the population size of the proposed algorithm should be a multiple of 4 so that each sub population has minimum number of required points. The proposed schemes are compared to the original DE in terms of average fitness function value, standard deviation, number of function evaluations and time taken. Experimental settings are population size NP=10*n, F=0.5, C_r=0.5, Maximum NFE=10^6, number of times DE1 called (N_{DE}) =NP/2 &NP/4 and accuracy 10^{-4}.

Numerical results:

The algorithms are compared in terms of average best fitness function value, standard deviation, average number of function evaluations and average time taken (in sec) obtained after 30 runs. The numerical results are given in Table 1. When the algorithms are compared according to the average fitness function value, we can see that all the proposed schemes performed at par with the original DE and in some cases even gave better results. The finer performance of the proposed versions is more evident when we compare the average number of function evaluations and the time taken by the algorithms to meet the stopping criteria, where in almost all the test cases, the proposed versions converged faster than the original DE. However, none of the algorithms converged for the Schwefel function. All problems are tested for dimension 50 and taken from the literature [14].

DOI: 01.IJPIE.02.01. 352

TABLE 1: COMPARISON OF AVERAGE FITNESS FUNCTION VALUE, STANDARD DEVIATION NUMBER OF FUNCTIONS EVALUATIONS AND TIME TAKEN IN SECONDS FOR ALL ALGORITHMS

Fun		Fitness (FF), Standard deviation (STD), NFE, Time in sec				
		DE	IDE1 P = 2, N_{DE}=2n	IDE2 P = 2, N_{DE}=n	IDE3 P = 4, N_{DE}=2n	IDE4 P = 4, N_{DE}=n
f_{ACK}	FF	0.000259178	0.000122811	0.000175384	0.000121801	0.000152979
	STD	0.000210415	0.000111969	0.00016112	0.000128962	0.000148479 850500
	NFE	916500	900500	875500	850500	98.00
	Time	112.0	103.0	100.0	99.00	
f_{EXP}	FF	0.999928	0.999968	0.99992	0.999985	0.999957
	STD	2.40074e-005	1.00833e-005	(7.77316e-006)	(6.46698e-006)	(6.80628e-006) 375500
	NFE	390500	400500	375500	400500	(40.00)
	Time	45.00	43.00	(40.00)	(42.00)	
f_{GW}	FF	8.48061e-005	3.47053e-005	7.68606e-005	5.70359e-005	5.93635e-005
	STD	(7.90575e-006)	(2.98511e-005)	(1.79218e-005)	(5.045e-005)	(1.44842e-005) 700500
	NFE	762800	750500	725500	700500	(84.00)
	Time	(96.60)	(89.00)	(85.00)	(82.00)	
f_{LM2}	FF	9.78409e-005	2.65663e-005	4.53916e-005	4.63788e-005	5.68771e-005
	STD	(8.11364e-005)	(2.00426e-005)	(6.15258e-006)	(4.34424e-005)	(5.43526e-005) 450500
	NFE	483500	500500	475500	450500	(96.00)
	Time	(108.0)	(108.0)	(101.6)	(96.00)	
f_{SWF}		--	--	--	--	--

V. CONCLUSIONS

In the proposed study we suggested four versions of DE based on population partitioning and using a single array DE in place of 2 array original DE. The proposed versions are tested on a set of 5 benchmark problems taken from literature. From the numerical results we can see that the proposed schemes help in improving the solution quality as well as the convergence rate in most of the test cases. The concept of partitioning the population and shuffling the candidates help in maintaining the diversity of the population and the concept of using a single array DE help in faster convergence. In future we plan to apply the proposed schemes on a more comprehensive set of test problems. Also we plan to compare the proposed versions with other models of DE.

REFERENCES

[1] R.Storn, and K.Price, DE-a simple and efficient adaptive scheme for global optimization over continuous space, Technical Report TR-95-012, ICSI, March 1995. Available via the Internet: ftp.icsi.berkeley.edu/pub/techreports/ 1995/tr-95-012.ps.Z, 1995.

[2] M.Omran, A.Engelbrecht and A.Salman, Differential evolution methods for unsupervised image classification." In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 2, 2005a, pp. 966–973.

[3] R.Storn,"Differential evolution design for an IIR-filter with requirements for magnitude and group delay". Technical Report TR-95-026, International Computer Science Institute, Berkeley, CA 1995.

[4] J. Vesterstroem and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," *Proc. Congr. Evol. Comput.*, vol. 2, pp. 1980–1987, 2004.

[5] J. Andre, P. Siarry, and T. Dognon, "An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization," *Advance in Engineering Software 32*, pp. 49–60, 2001.

[6] O. Hrstka and A. Ku¢cerová, "Improvement of real coded genetic algorithm based on differential operators preventing premature convergence," *Advance in Engineering Software 35*, pp. 237–246, 2004.

[7] J. Lampinen and I. Zelinka, "On stagnation of the differential evolution algorithm," in: Pavel Ošmera, (ed.) *Proc. of MENDEL 2000, 6th International Mendel Conference on Soft Computing*, pp. 76 – 83, June 7–9. 2000, Brno, Czech Republic.

[8] M. Omran, A. Salman, and A. P. Engelbrecht, "Self-adaptive differential evolution, computational intelligence and security," *PT 1, Proceedings Lecture Notes In Artificial Intelligence* 3801: 192-199, 2005.

[9] S. Rahnamayan, H.R. Tizhoosh, and M. M. A. Salama, "Opposition-Based Differential Evolution," *IEEE Transactions on Evolutionary Computation*, Vol. 12, Issue 1, pp. 64 – 79, 2008.

[10] U. K. Chakraborty (Ed.) *Advances in Differential Evolution*, Springer-Verlag, Heidelberg, 2008.

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

[12] Q.Y.Duan, V.K.Gupta and S.Sorooshian, "shuffled complex evolution approach for effective and efficient global minimization", journal of optimization theory and applications, 73(3), pp 501-521, 1993.

[13] B.V.Babu and R. Angira, "Modified differential evolution for optimization of non linear chemical processes", computers and chemical engineering 30, pp 989-1002, 2006.

[14] M.M.Ali, C.Khompatraporn and Z.B.Zabinsky,"A numerical evaluation of several stochastic algorithm on selected continuous global optimization test problems", Journal Global optimization, 31, pp 635-672, 2005.

DOI: 01.IJPIE.02.01.352