ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 3 (2011) 83–88

Procedia Computer Science

www.elsevier.com/locate/procedia

WCIT 2010

A comparative study of common and self-adaptive differential evolution strategies on numerical benchmark problems

S. K. Goudos [a], K. B. Baltzis [a] *, K. Antoniadis [a], Z. D. Zaharis[a], C. S. Hilas [b]

[a]RadioCommunications Laboratory, Department of Physics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece
[b]Department of Informatics & Communications, Technological Educational Institute of Serres, Serres 62124, Greece

Abstract

Differential Evolution (DE) is a population-based stochastic global optimization technique that requires the adjustment of a very few parameters in order to produce results. However, the control parameters involved in DE are highly dependent on the optimization problem; in practice, their fine-tuning is not always an easy task. The self-adaptive differential evolution (SADE) variants are those that do not require the pre-specified choice of control parameters. On the contrary, control parameters are self-adapted by using the previous learning experience. In this paper, we discuss and evaluate popular common and self-adaptive differential evolution (DE) algorithms. In particular, we present an empirical comparison between two self-adaptive DE variants and common DE methods. In order to assure a fair comparison, we test the methods by using a number of well-known unimodal and multimodal, separable and non-separable, benchmark optimization problems for different dimensions and population size. The results show that SADE variants outperform, or at least produce similar results, to common differential evolution algorithms in terms of solution accuracy and convergence speed. The advantage of using the self-adaptive methods is that the user does not need to adjust control parameters. Therefore, the total computational effort is significantly reduced.

Selection and/or peer-review under responsibility of the Guest Editor.

Keywords: Differential evolution, benchmark functions, global optimization, adaptive parameter control.

1. Introduction

In the past decades, several evolutionary algorithms (EAs) that mimic biological entities behavior and evolution have emerged. EAs are widely used for the solution of single and multi-objective optimization problems. An evolutionary algorithm that has recently gained popularity is Differential Evolution (DE) [1,2]. DE is population-based stochastic global optimization algorithm. The control parameter setting in EAs and DE has been extensively studied in the literature [3-6]. The effect of the population size was reported in [7]. Several DE variants or strategies exist [8-12]. The classical DE algorithm has been applied to a large number of engineering problems such as microwave structures and antenna design [13-16].

DE produces better results than Particle Swarm Optimization (PSO) on numerical benchmark problems with low or medium dimensionality (30 and 100 dimensions) [17]. One of its advantages is the adjustment of very few parameters. The control parameters and the learning strategies involved in DE are highly dependent on the optimization problem to be solved. Thus, its drawback is the excessive time required for strategy selection and for

* K. B. Baltzis. Tel.: +30-2310-998285; fax: +30-2310-998069.
E-mail address: kmpal@physics.auth.gr.

doi:10.1016/j.procs.2010.12.015

fine-tuning the control parameters. Recently, a novel DE algorithm, the Self-adaptive Differential Evolution (SADE), has been applied to numerical benchmark problems that self-adapts these control parameters [18].

2. The Differential Evolution (DE) and the Self-adaptive Differential Evolution (SADE) algorithms

A population in DE consists of *NP* vectors $\overline{x}_{i,G},\ i=1,2,...NP$ (*G* is the number of generations). The possible solutions are represented with *D*-dimensional vectors $\overline{x}_{i,G}=(x_{i,1,G},x_{i,2,G},...x_{i,D,G})$. The population is randomly initialized from a uniform distribution between the low and the upper bounds defined for each variable; these bounds are user-specified according to the nature of the problem.

The initial population evolves in each generation with the use of mutation, crossover and selection operators. Depending on the form of these operators, several DE variants or strategies exist [2,19]. The choice of the best DE strategy depends on the type of the problem [20]. Two popular strategies are the *DE/best/1/bin* and the *DE/rand/1/bin*. In these, a mutant vector $\overline{v}$ for every target vector $\overline{x}_{i,G}$ is computed, respectively, by

$$\overline{v}_{j,G}=\overline{x}_{best,G}+F(\overline{x}_{r_1,G}-\overline{x}_{r_2,G}),\quad r_1\neq r_2 \tag{1}$$

$$\overline{v}_{i,G}=\overline{x}_{r_1,G}+F(\overline{x}_{r_2,G}-\overline{x}_{r_3,G}),\quad r_1\neq r_2\neq r_3 \tag{2}$$

In these expressions, $\overline{x}_{best,G}$ is the best vector found at generation *G*, *F* is a mutation control parameter and r_1, r_2, and r_3 are randomly chosen indices from the population. After mutation, the crossover operator generates a trial vector $\overline{u}_{i,G}$ with elements

$$u_{i,j,G}=\begin{cases} v_{i,j,G}, & \text{if } rnd_j\leq CR\ or\ j=rn(j) \\ x_{i,j,G}, & \text{otherwise}\end{cases} \tag{3}$$

where j = 1,2,…*D*, rnd_j is a number from a uniform random distribution from the interval [0,1], $rn(j)$ is a randomly chosen index from (1,2,…*D*) and *CR* is the crossover constant from the interval [0,1]. Differential evolution uses a greedy selection operator. According to this, the selection scheme for minimization problems is

$$\overline{x}_{i,G+1}=\begin{cases} \overline{u}_{i,G}, & \text{if } f(\overline{u}_{i,G})<f(\overline{x}_{i,G}) \\ \overline{x}_{i,G}, & \text{otherwise}\end{cases} \tag{4}$$

where $f(\overline{u}_{i,G})$, $f(\overline{x}_{i,G})$ are the fitness values of the trial and the old vector, respectively. Common stopping criteria are the number of generations or the number of the objective-function evaluations.

Storn has suggested [2] that the DE control parameters are adjusted as $F\in[0.5,1]$, $CR\in[0.8,1]$ and $NP=10D$. In [18], a novel approach was proposed for self-adapting of the DE control parameters. The method probabilistically selects one out of several available learning strategies for each individual in the current population and automatically adapts parameters settings during evolution. This strategy was based on the *DE/rand/1/bin* scheme. Each vector was extended with its own *F* and *CR* values and the control parameters were self-adjusted in every generation for each individual according to the scheme:

$$F_{i,G+1} = \begin{cases} F_l + rnd_1 \cdot F_u, & \text{if } rnd_2 < 0.1 \\ F_{i,G}, & \text{otherwise} \end{cases} \quad \text{and} \quad CR_{i,G+1} = \begin{cases} rnd_3 & \text{if } rnd_4 < 0.1 \\ CR_{i,G}, & \text{otherwise} \end{cases} \tag{5}$$

where $rnd_{1\ldots4}$ are uniform random numbers in the interval [0,1], and F_l, F_u are the lower and the upper limits of *F*. The latter are set to 0.1 and 0.9, respectively [18]. Therefore by using the self-adaptive algorithm the user does not have to adjust the *F* and *CR* parameters while the time complexity does not increase (a detailed analysis of the SADE algorithm can be found in [18]). Recently, this algorithm has been successfully applied to microwave absorber and pyramidal horn design problems [21,22]. We point out that in [18] the self-adapting algorithm was based only on *rand/1/bin* strategy. In [20], the authors concluded that the *best/1/bin* strategy is the most competitive approach regardless of the characteristics of the problem at hand. In this paper, we also present a new version of the SADE algorithm based on the *best/1/bin* strategy. We denote the two self-adaptive variants used in our study as *SADE/rand/1/bin* and *SADE/best/1/bin.*

3. Test Functions

Usually, a comparative study of different optimization methods is performed by using a set of test functions from the literature. In this paper, six well-known benchmark functions are used. We have chosen two unimodal and four multimodal functions; the functions can also be grouped into separable or non-separable. These are the Sphere function, the Schwefel's problem 1.2, the generalized Rosenbrock's function, the Ackley's function, the generalized Rastrigin's function and the Salomon's function. The above functions are defined [20] respectively as:

$$f_1(x) = \sum_{j=0}^{D-1} x_j^2, \quad |x_j| \le 5.12 \quad \text{and} \quad f_1(0,0,\ldots0) = 0 \tag{6}$$

$$f_2(x) = \sum_{k=0}^{D\text{-}1}\left(\sum_{j=0}^{k} x_j\right)^2, \quad |x_j| < 100 \quad \text{and} \quad f_2(0,0,\ldots0) = 0 \tag{7}$$

$$f_3(x) = \sum_{j=0}^{D-2}\left(100\left(x_{j+1} - x_j^2\right)^2 + \left(x_j - 1\right)^2\right), \quad |x_j| \le 2 \quad \text{and} \quad f_3(1,1,\ldots1) = 0 \tag{8}$$

$$f_4(x) = -20\exp\left(-\frac{1}{5}\sqrt{\frac{1}{D}\sum_{j=0}^{D-1} x_j^2}\right) - \exp\left(\frac{1}{D}\sum_{j=0}^{D-1}\cos\left(2\pi x_j\right)\right) + 20 + \exp(1), \quad |x_j| \le 10 \quad \text{and} \quad f_4(0,0,\ldots0) = 0 \tag{9}$$

$$f_5(x) = \sum_{j=0}^{D-1}\left(x_j^2 - 10\cos\left(2\pi x_j\right) + 10\right), \quad |x_j| \le 5.12 \quad \text{and} \quad f_5(0,0,\ldots0) = 0 \tag{10}$$

$$f_6(x) = -\cos\left(2\pi\sqrt{\sum_{j=0}^{D-1} x_j^2}\right) + 0.1\sqrt{\sum_{j=0}^{D-1} x_j^2} + 1, \quad |x_j| \le 1.5 \quad \text{and} \quad f_6(0,0,\ldots0) = 0 \tag{11}$$

The sphere function is one of the simplest benchmarks. It is a continuous, unimodal and separable problem. The second one is a unimodal and separable problem. The generalized Rosenbrock's global optimum lies inside a parabolic shaped flat valley. It is easy to find the valley but convergence to the global optimum is difficult. This problem is multimodal and non-separable. The Ackley's function is a multimodal non-separable problem and has many local optima and a narrow global optimum. The generalized Rastrigin function is a complex multimodal separable problem with many local optima. The sixth problem is highly multimodal and non-separable.

4. Experimental Results and Discussion

This section presents a comparative analysis of the common DE strategies *DE/best/1/bin* and *DE/rand/1/bin* and their self-adaptive variations *SADE/best/1/bin* and *SADE/rand/1/bin.* In order to perform a fair comparison, we use the benchmark functions that are given in Section 3. We study two cases with $D = 30$ and $D = 50$. In the first case, we set $NP = 100$; in the second one, the population size is set equal to 150. As it is reported in [7], a large population size affects the ability of the algorithm to find the correct search direction. Therefore, we decided to work with relatively small population sizes as in [20]. In each experiment, the total number of iterations is equal to 3000. The results of all the experiments are averaged over 50 independent runs. All experiments are executed 50 times. The control parameters for the common DE strategies are $F = 0.6$ and $CR = 0.9$.

Figures 1 and 2 show the convergence graphs for the conducted experiments on 30 and 50 dimensions, respectively. Table 1 presents the corresponding mean and standard deviation values. The bold font indicates the best results for every case. We notice that in the unimodal and separable problems (Sphere and Schwefel’s 1.2) the self-adaptive algorithms convergence faster than the corresponding common DE variants; also, the *best/1/bin* is faster than the *rand/1/bin*. These results are consistent with [20]. In the case of the Rosenbrock’s function, the *SADE/best/1/bin* variant obtains the best results and the fastest convergence. For the Ackley’s function, the *rand/1/bin* variants converge faster than the *best/1/bin* ones. The obtained results up to numerical errors are similar for *DE/rand/1/bin, SADE/rand/1/bin,* and *SADE/best/1/bin.* The *best/1/bin* variants showed clearly a better performance over the *rand/1/bin* ones for the case of the generalized Rastrigin function, which is probably one of the hardest problems to solve. The *SADE/rand/1/bin* showed the best performance for the 30-D case. However, for the 50-D case, the best results are obtained by *SADE/best/1/bin* in terms of both solution accuracy and convergence speed. The *rand/1/bin* variants provide better results for the Salomon’s function. The *best/1/bin* variants and the *SADE/rand/1/bin* converge at similar speeds. Finally, we must also point out that the obtained results are quite similar in both 30-D and 50-D problems.

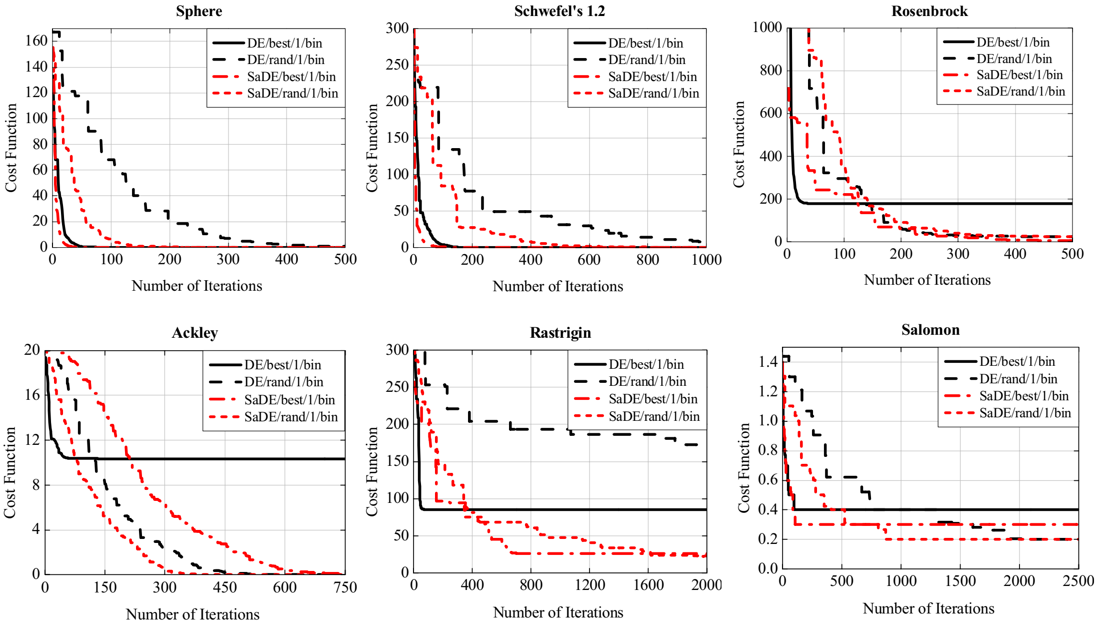

Fig. 1. Convergence graphs of DE and SADE variants on benchmark problems with 30 dimensions.

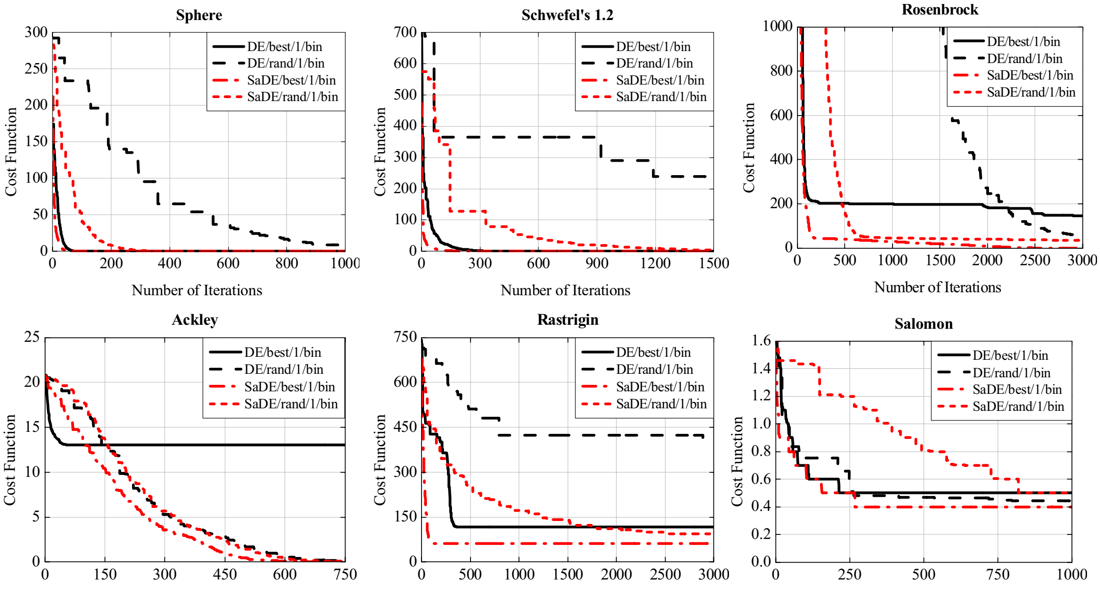

Fig. 2. Convergence graphs of DE and SADE variants on benchmark problems with 50 dimensions.

Table 1. Average value and standard deviation of DE and SADE variants on benchmark problems with 30 and 50 dimensions.

Benchmark problem	*D*	*DE/best/1/bin*		*DE/rand/1/bin*		*SADE/best/1/bin*		*SADE/rand/1/bin*	
		Avg	StD	Avg	StD	Avg	StD	Avg	StD
Sphere function	30	0.002	0.003	0.001	$4\text{x}10^{-4}$	**0**	**0**	**0**	**0**
	50	0.171	0.014	1.002	0.196	**0**	**0**	**0**	**0**
Schwefel's problem 1.2	30	**0**	**0**	11.800	2.844	**0**	**0**	0.251	0.133
	50	$\mathbf{2.48\text{x}10^{-8}}$	$\mathbf{1.55\text{x}10^{-8}}$	291.599	38.229	$3.31\text{x}10^{-7}$	$2.95\text{x}10^{-7}$	4.290	0.969
Rosenbrock's function	30	168.519	64.682	3.749	1.277	$\mathbf{9.251\text{x}10^{-15}}$	$\mathbf{1.771\text{x}10^{-14}}$	12.254	0.169
	50	145.024	61.896	40.597	0.669	**0.284**	**1.026**	24.301	0.483
Ackley's function	30	10.118	0.889	$1.375\text{x}10^{-13}$	$5.127\text{x}10^{-14}$	$\mathbf{1.261\text{x}10^{-14}}$	$\mathbf{3.389\text{x}10^{-15}}$	$1.585\text{x}10^{-9}$	$5.188\text{x}10^{-10}$
	50	13.366	1.119	$1.171\text{x}10^{-8}$	$3.752\text{x}10^{-9}$	$\mathbf{4.273\text{x}10^{-11}}$	$\mathbf{1.615\text{x}10^{-11}}$	$1.339\text{x}10^{-9}$	$4.431\text{x}10^{-10}$
Rastrigin's function	30	97.942	25.440	171.157	7.195	21.970	4.987	**13.602**	**3.859**
	50	102.934	26.082	383.871	11.631	**101.810**	**23.419**	109.861	10.086
Salomon's function	30	0.394	0.067	0.201	0.012	0.304	0.022	**0.199**	**0**
	50	0.489	0.079	0.445	0.065	0.371	0.039	**0.261**	**0.047**

5. Conclusions

We have presented a performance comparison of some common and self-adaptive DE algorithms. The methods were implemented and studied on six well-known benchmark problems on thirty and fifty dimensions. The results showed that the *SADE/best/1/bin* variant outperforms or produces similar results with the other methods in terms of solution accuracy and convergence speed. The obtained results show that the *best/1/bin* variants are the most suitable optimizers for solving unimodal problems. For multimodal problems the *SADE/best/1/bin* strategy was the most competitive in most of the cases. Regarding convergence speed in most cases the *best/1/bin* strategy is faster. Our results are consistent with those found in [20]. The major advantage of the self-adaptive DE is that it does not require the pre-specified choice of control parameters thus reducing significantly the users' effort. In our future work we plan to further compare the *SADE/best/1/bin* variant with other competitive algorithms and extend the search to 100 or more dimensions.

References

1. R. Storn and K. Price, Differential evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces, ICSI Tech. Rep. TR-95-012 (1995).
2. R. Storn and K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, J. Global Optimization 11 (1997) 341-359.
3. A. E. Eiben, R. Hinterding, and Z. Michalewicz, Parameter control in evolutionary algorithms, IEEE Trans. Evol. Comput. 3 (1999) 124-141.
4. T. Krink and R. K. Ursem, Parameter control using the agent based patchwork model, Proc. 2000 Congr. Evol. Comput., La Jolla (2000).
5. J. Liu and J. Lampinen, On setting the control parameter of the differential evolution method, Proc. 8th Int. Conf. Soft Comput., Brno (2002).
6. J. Liu and J. Lampinen, Adaptive parameter control of differential evolution, Proc. 8th Int. Conf. Soft Comput., Brno (2002).
7. V. Feoktistov and S. Janaqi, Generalization of the strategies in differential evolution, Proc 18th Int. Parallel Distrib. Process. Symp., Santa Fe (2004).
8. I. L. Lopez Cruz, L.G. Van Willigenburg, and G. Van Straten, Efficient differential evolution algorithms for multimodal optimal control problems, Appl. Soft Comput. 3 (2003) 97-122.
9. J. Liu and J. Lampinen, A fuzzy adaptive differential evolution algorithm, Soft Comput. 9 (2005) 448-462.
10. J. Sun, Q. Zhang, and E. P. K. Tsang, DE/EDA: A new evolutionary algorithm for global optimization, Inform. Sci. 169 (2005) 249-262.
11. S. Kukkonen and J. Lampinen, GDE3: The third evolution step of generalized differential evolution, Proc. 2005 Congr. Evol. Comput., Edinburgh (2005).
12. J. Zhang, V. Avasarala, A. C. Sanderson, and T. Mullen, Differential evolution for discrete optimization: An experimental study on combinatorial auction problems, Proc. 2008 Congr. Evol. Comput., Hong Kong (2008).
13. D. G. Kurup, M. Himdi, and A. Rydberg, Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm, IEEE Trans. Antennas Propag. 51 (2003) 2210-2217.
14. S. Yang, Y. B. Gan, and A. Qing, Antenna-array pattern nulling using a differential evolution algorithm, Int. J. RF Microw. Comput.-Aid. Eng. 14 (2004) 57-63.
15. S. Caorsi, A. Massa, M. Pastorino, and A. Randazzo, Optimization of the difference patterns for monopulse antennas by a hybrid real/integer-coded differential evolution method, IEEE Trans. Antennas Propag. 53 (2005) 372-376.
16. Y. Chen, S. Yang, and Z. Nie, The application of a modified differential evolution strategy to some array pattern synthesis problems, IEEE Trans. Antennas Propag. 56 (2008) 1919-1927.
17. J. Vesterstrom and R. Thomsen, A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems, Proc. 2004 Congr. Evol. Comput., Portland (2004).
18. J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (2006) 646-657.
19. R. Storn, Differential evolution research - Trends and open questions, Stud. Comput. Intell. 143 (2008) 1-31.
20. E. Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello, A comparative study of differential evolution variants for global optimization, Proc. 8th Genetic Evol. Comput. Conf., Seattle (2006).
21. S. K. Goudos, Design of microwave broadband absorbers using a self-adaptive differential evolution algorithm, Int. J. RF Microw. Comput.-Aid. Eng. 19 (2009) 364-372.
22. D. Pappas, S. K. Goudos, K. B. Baltzis, and K. Siakavara, Design of optimum gain pyramidal horn using self-adaptive differential evolution algorithms, Int. J. RF Microw. Comput.-Aid. Eng. (in print).