Swarm and Evolutionary Computation 27 (2016) 1–30

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Survey Paper

Recent advances in differential evolution – An updated survey

Swagatam Das [a,*], Sankha Subhra Mullick [a], P.N. Suganthan [b,*]

[a] Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata 700108, India
[b] School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 10 October 2015
Received in revised form
13 January 2016
Accepted 15 January 2016
Available online 1 February 2016

Keywords:
Differential evolution
Continuous evolutionary optimization
Numerical optimization
Parameter adaptation
Recombination

ABSTRACT

Differential Evolution (DE) is arguably one of the most powerful and versatile evolutionary optimizers for the continuous parameter spaces in recent times. Almost 5 years have passed since the first comprehensive survey article was published on DE by Das and Suganthan in 2011. Several developments have been reported on various aspects of the algorithm in these 5 years and the research on and with DE have now reached an impressive state. Considering the huge progress of research with DE and its applications in diverse domains of science and technology, we find that it is a high time to provide a critical review of the latest literatures published and also to point out some important future avenues of research. The purpose of this paper is to summarize and organize the information on these current developments on DE. Beginning with a comprehensive foundation of the basic DE family of algorithms, we proceed through the recent proposals on parameter adaptation of DE, DE-based single-objective global optimizers, DE adopted for various optimization scenarios including constrained, large-scale, multi-objective, multi-modal and dynamic optimization, hybridization of DE with other optimizers, and also the multi-faceted literature on applications of DE. The paper also presents a dozen of interesting open problems and future research issues on DE.

1. Introduction

In an attempt to find the global optimum of non-linear, non-convex, multi-modal and non-differentiable functions defined in the continuous parameter space ($\subseteq \mathbb{R}^d$), Storn and Price proposed the Differential Evolution (DE) [173,174] algorithm in 1995. Since then, DE and its variants have emerged as one of the most competitive and versatile family of the evolutionary computing algorithms and have been successfully applied to solve numerous real world problems from diverse domains of science and technology [134,40]. Starting with a uniformly random set of candidate solutions sampled from the feasible search volume, every iteration (commonly known as *generation* in evolutionary computing terminology) of DE operates through the same computational steps as employed by a standard Evolutionary Algorithm (EA). However, DE differs markedly from the well-known EAs like Evolution Strategies (ESs) and Evolutionary Programming (EP) in consideration of the fact that it mutates the base vectors (secondary parents) with scaled difference(s) of the distinct members from the current population. As iterations pass, these differences tend to adapt to the natural scales of the objective landscape. For example, if the population becomes compact in one variable but remains widely dispersed in another, the difference vectors sampled from it will be smaller in the former variable, yet larger in the latter. This automatic adaptation significantly improves the search moves of the algorithm. This property is also known as the self-referential mutation. In other words, while ES, EP, and some other real coded Genetic Algorithms (GAs) require the specification or adaptation of the absolute step size for each variable over iterations, the canonical DE requires only the specification of a single relative scale factor F for all variables. Unlike several other evolutionary computation techniques, basic DE stands out to be a very simple algorithm whose implementation requires only a few lines of code in any standard programming language. In addition, the canonical DE requires very few control parameters (3 to be precise: the scale factor, the crossover rate and the population size) — a feature that makes it easy to use for the practitioners. Nonetheless, DE exhibits remarkable performance while optimizing a wide variety of objective functions in terms of final accuracy, computational speed, and robustness. It is interesting to note that the variants of DE have been securing front ranks in various competitions among EAs organized under the IEEE Congress on Evolutionary Computation (CEC) conference series (for details, please see http://www.ntu.edu.sg/home/epnsugan/index_files/cec-benchmarking.htm). It is evident that no other single search paradigm has been able to secure competitive ranking in nearly all the CEC competitions on single-objective, constrained, dynamic, large-scale, multi-objective, and multi-modal optimization problems.

* Corresponding authors.
E-mail addresses: swagatam.das@isical.ac.in (S. Das), mullicksankhasubhra@gmail.com (S.S. Mullick), epnsugan@ntu.edu.sg (P.N. Suganthan).

http://dx.doi.org/10.1016/j.swevo.2016.01.004
2210-6502/

In order to present the flavor of the huge and multi-faceted literature on DE, in 2010, [134] reviewed a number of DE-variants for the single-objective optimization problems and also made an experimental comparison of these variants on a set of standard benchmark functions. However, the article did not address issues like adapting DE to complex optimization environments involving multiple and constrained objective functions, noise and uncertainty in the fitness landscape, very large number of search variables, and so on. Also, it did not focus on the most recent engineering applications of DE and the developments in the theoretical analysis of DE. In this respect, the first comprehensive survey on almost all aspects of the DE family of algorithms was published in 2011 by [40]. Since then, DE has advanced a lot due to the continuous efforts of EC researchers all over the globe. In a recent survey by [49], the authors reviewed two aspects of the DE family of algorithms: the self-adaptive and adaptive parameter control strategies in DE and the hybridization of DE with other algorithms. In this article, we present a more exhaustive account of the recent advances in DE including its basic concepts, different structures, and variants for solving constrained, multi-objective, dynamic, and large-scale optimization problems as well as applications of DE variants to practical optimization problems. In addition we present several open research issues that call for the attention of the DE researchers.

The rest of this paper is arranged as follows. In Section 2, the basic concepts related to classical DE are explained along with the original formulation of the algorithm in the real number space. Section 3 discusses the recently developed parameter adaptation and control schemes for DE. Section 4 provides an overview of several prominent variants of the DE algorithm for the single-objective global numerical optimization. Section 5 provides an extensive survey on the applications of DE to the constrained, multi-objective, multi-modal, combinatorial, and dynamic optimization problems. Hybrid DE algorithms have been reviewed in Section 6. An overview on the recent theoretical studies of DE has been presented in Section 7. Section 8 provides an account of the recently developed parallel and distributed DE schemes. Section 9 highlights the recent and prominent engineering applications of DE. Section 10 discusses some interesting future research issues related to DE. Finally, the paper is concluded in Section 11.

2. The canonical DE algorithm

The initial iteration of a standard DE algorithm consists of four basic steps – initialization, mutation, recombination or crossover, and selection, of which, only the last three steps are repeated into the subsequent DE iterations. The iterations continue till a termination criterion (such as exhaustion of maximum functional evaluations) is satisfied.

2.1. Initialization of the decision variable vectors

DE searches for a global optimum point in a d-dimensional real decision variable space $\Omega \subseteq \mathbb{R}^d$. It begins with a randomly initiated population of Np d-dimensional real-valued decision vectors. Each vector, also known as *genome/chromosome*, forms a candidate solution to the multi-dimensional optimization problem. We shall denote subsequent iterations in DE by $t = 0, 1, ..., t_{\max}$. Since the parameter vectors are likely to be changed over different iterations, we may adopt the following notation for representing the ith vector of the population at the current iteration:

$$\mathbf{x}_i^{(t)} = \left(x_{i,1}^{(t)}, x_{i,2}^{(t)}, \ldots, x_{i,d}^{(t)}\right). \tag{1}$$

For each decision variable of the problem, there may be a certain range within which the value of the decision variable should be restricted, often because decision variables are related to physical components or measures that have natural bounds (for example if one decision variable is a length or mass, we would want it not to be negative). The initial population (at $t = 0$) should cover this range as much as possible by uniformly randomizing individuals within the search space constrained by the prescribed minimum and maximum bounds: $\mathbf{x}_{\min} = (x_{\min,1}, x_{\min,2}, \ldots, x_{\min,d})$ and $\mathbf{x}_{\max} = (x_{\max,1}, x_{\max,2}, \ldots, x_{\max,d})$. Hence, we may initialize the jth component of the ith decision vector as

$$x_{i,j}^{(0)} = x_{\min,j} + rand_{i,j}[0,1]\left(x_{\max,j} - x_{\min,j}\right), \tag{2}$$

where $rand_{i,j}[0, 1]$ is a uniformly distributed random number lying between 0 and 1 (actually $0 \le rand_{i,j}[0, 1] \le 1$) and is instantiated independently for each component of the i-th vector.

2.2. Mutation with difference vectors

After initialization, DE creates a *donor/mutant* vector $\mathbf{v}_i^{(t)}$ corresponding to each population member or *target* vector $\mathbf{x}_i^{(t)}$ in the current iteration through mutation. Five most frequently referred mutation strategies are listed below:

$$\text{“DE/rand/1”:}\mathbf{v}_i^{(t)} = \mathbf{x}_{R_1{}^i}^{(t)} + F\left(\mathbf{x}_{R_2{}^i}^{(t)} - \mathbf{x}_{R_3{}^i}^{(t)}\right). \tag{3a}$$

$$\text{“DE/best/1”:}\mathbf{v}_i^{(t)} = \mathbf{x}_{best}^{(t)} + F\left(\mathbf{x}_{R_1{}^i}^{(t)} - \mathbf{x}_{R_2{}^i}^{(t)}\right). \tag{3b}$$

$$\text{“DE/current}-\text{to}-\text{best/1”:}\mathbf{v}_i^{(t)} = \mathbf{x}_i^{(t)} + F\left(\mathbf{x}_{best}^{(t)} - \mathbf{x}_i^{(t)}\right) + F\left(\mathbf{x}_{R_1{}^i}^{(t)} - \mathbf{x}_{R_2{}^i}^{(t)}\right). \tag{3c}$$

$$\text{“DE/best/2”:}\mathbf{v}_i^{(t)} = \mathbf{x}_{best}^{(t)} + F\left(\mathbf{x}_{R_1{}^i}^{(t)} - \mathbf{x}_{R_2{}^i}^{(t)}\right) + F\left(\mathbf{x}_{R_3{}^i}^{(t)} - \mathbf{x}_{R_4{}^i}^{(t)}\right). \tag{3d}$$

$$\text{“DE/rand/2”:}\mathbf{v}_i^{(t)} = \mathbf{x}_{R_1{}^i}^{(t)} + F\left(\mathbf{x}_{R_2{}^i}^{(t)} - \mathbf{x}_{R_3{}^i}^{(t)}\right) + F\left(\mathbf{x}_{R_4{}^i}^{(t)} - \mathbf{x}_{R_5{}^i}^{(t)}\right). \tag{3e}$$

The indices ${R_1}^i, {R_2}^i, {R_3}^i, {R_4}^i$ and ${R_5}^i$ are mutually exclusive integers randomly chosen from the range [1, Np], and all are different from the base index i. These indices are randomly generated anew for each donor vector. The scaling factor F is a positive control parameter for scaling the difference vectors. $\mathbf{x}_{best}^{(t)}$ is the best individual vector with the best fitness (*i.e.* with the lowest objective function value for a minimization problem) in the population at iteration t. The general convention used for naming the various mutation strategies is DE/x/y/z, where DE stands for Differential Evolution, **x** represents a string denoting the vector to be perturbed and y is the number of difference vectors considered for perturbation of **x**. z stands for the type of crossover being used (*exp*: exponential; *bin*: binomial). Note that in the DE/current-to-best/1 scheme the vector which is being perturbed with the scaled difference of the two other population members is basically a convex combination of the current target vector and the best population member for $F < 1$. This means here the base vector for mutation denotes a point on the line joining the target vector and the best population member and it is an arithmetic recombination between $\mathbf{x}_i^{(t)}$and $\mathbf{x}_{best}^{(t)}$. Thus, the resulting donor vector can be thought of as a mutated recombinant.

2.3. Crossover

Through crossover, the donor vector mixes its components with the target vector $\mathbf{x}_i^{(t)}$ to form the trial/offspring vector $\mathbf{u}_i^{(t)} = (u_{i,1}^{(t)}, u_{i,2}^{(t)}, \ldots, u_{i,d}^{(t)})$. The DE family of algorithms commonly uses two crossover methods-*exponential* (or two-point modulo) and *binomial* (or uniform) [40]. We here elaborate the binomial and the exponential