Information Sciences 181 (2011) 5364–5386

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Adaptive strategy selection in differential evolution for numerical optimization: An empirical study ☆

Wenyin Gong [a,b], Álvaro Fialho [c], Zhihua Cai [a,*], Hui Li [a]

[a] School of Computer Science, China University of Geosciences, Wuhan 430074, PR China
[b] State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, PR China
[c] Nokia Institute of Technology, 69048-660 Manaus, AM, Brazil

ARTICLE INFO

Article history:
Received 25 February 2011
Received in revised form 26 July 2011
Accepted 30 July 2011
Available online 11 August 2011

Keywords:
Differential evolution
Adaptation
Strategy selection
Credit assignment
Numerical optimization

ABSTRACT

Differential evolution (DE) is a versatile and efficient evolutionary algorithm for global numerical optimization, which has been widely used in different application fields. However, different strategies have been proposed for the generation of new solutions, and the selection of which of them should be applied is critical for the DE performance, besides being problem-dependent. In this paper, we present two DE variants with adaptive strategy selection: two different techniques, namely *Probability Matching* and *Adaptive Pursuit*, are employed in DE to autonomously select the most suitable strategy while solving the problem, according to their recent impact on the optimization process. For the measurement of this impact, four credit assignment methods are assessed, which update the known performance of each strategy in different ways, based on the relative fitness improvement achieved by its recent applications. The performance of the analyzed approaches is evaluated on 22 benchmark functions. Experimental results confirm that they are able to adaptively choose the most suitable strategy for a specific problem in an efficient way. Compared with other state-of-the-art DE variants, better results are obtained on most of the functions in terms of quality of the final solutions and convergence speed.

1. Introduction

Differential evolution (DE), proposed by Storn and Price [35], is an efficient and versatile population-based direct search algorithm that implements the evolutionary generation-and-test paradigm for global optimization, using the distance and direction informations from the current population to guide the search. Among its advantages are its simple structure, ease of use, speed, and robustness, which enables its application on many real-world applications, such as data mining, IIR design, neural network training [29], power systems [43], financial market dynamics modeling [16], data mining [4], and so on. A good survey of DE can be found in [5], where its basic concepts and major variants, as well as some theoretical studies and application examples to complex environments, are reviewed in detail.

In the seminal DE algorithm [35], a single mutation strategy was used for the generation of new solutions; later on, Price and Storn suggested nine other different strategies [29,36]. In addition, other mutation strategies are also proposed in the DE

☆ This work was partly supported by the Fundamental Research Funds for the Central Universities at China University of Geosciences (Wuhan) under Grant No. CUG100316, the Foundation of State Key Lab of Software Engineering under Grant No. SKLSE2010-08-13, the National Natural Science Foundation of China under Grant No. 61075063, and the Research Fund for the Doctoral Program of Higher Education under Grant No. 20090145110007.

* Corresponding author. Tel.: +86 27 67883716.
E-mail addresses: cug11100304@yahoo.com.cn (W. Gong), fialho@lix.polytechnique.fr (Á. Fialho), zhcai@cug.edu.cn (Z. Cai), huili@vip.sina.com (H. Li).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2011.07.049

literature [50,3,6,8]. Although augmenting the robustness of the underlying algorithm, these many available strategies led the user to the need of defining which of them would be most suitable for the problem at hand – a difficult and crucial task for the performance of DE [31,30,23].

Off-line tuning techniques, such as the F-Race [1], could be used to choose the mutation strategy to be used. However, besides being computationally expensive, such techniques usually output a static setting; while, in practice, the performance of each mutation strategy does not depend on the problem itself, but rather on the characteristics of the region of the search landscape being explored by the population at each generation. Based on this, thus, in order to be more efficient, the autonomous selection of the strategy to be used should be done in a continuous way, while solving the problem, *i.e.*, dynamically adapting itself as the search goes on.

In order to contribute on remedying this drawback, in this paper, we extend our recent work [15] on the use of adaptive strategy selection within DE for global numerical optimization. To do adaptive strategy selection, *i.e.*, to be able to automatically select which is the best mutation strategy for the generation of each offspring while solving the problem, two elements need to be defined [48,18]: (i) how to select between the available strategies based on their recent performance (strategy selection); and (ii) how to measure the performance of the strategies after their application, and consequently update the empirical quality estimates kept for each of them (credit assignment). In this work, two strategy selection techniques, namely *Probability Matching* [12] and *Adaptive Pursuit* [41], are independently analyzed in combination with each of four credit assignment techniques based on the relative fitness improvement. In addition, a parameter sensitivity analysis is conducted to investigate the impact of the hyper-parameters on the performance of the resulting adaptive strategy selection technique. Experiments have been conducted on 22 widely used benchmark problems, including nine test functions presented in CEC-05 [37]. The results indicate that the analyzed approach is able to select the most suitable strategy, while solving a problem at hand. Compared with other state-of-the-art DE variants, better results are obtained on most of the functions in terms of quality of final solutions and convergence speed.

Compared with our previous work in [15], the main contributions of this paper are twofold: (i) in order to pursuit the most suitable strategy at different search stages for a specific problem more rapidly, the *Adaptive Pursuit* technique is used and its performance is compared with the *Probability Matching*-based DE variant; and (ii) the comprehensive experiments are conducted to verify our approach and its performance is analyzed in detail.

The remainder of the paper is organized as follows. Section 2 briefly introduces the background and related work of this paper. In Section 3, we describe the adaptive strategy selection approaches in detail, followed by the experimental results and discussions in Section 4. Finally, Section 5 is devoted to conclusions and future work.

2. Background and related work

2.1. Problem formulation

Without loss of generality, in this work, we consider the following numerical optimization problem:

$$\text{Minimize} \quad f(\mathbf{x}), \quad \mathbf{x} \in S, \tag{1}$$

where $S \subseteq \mathbb{R}^D$ is a compact set, $\mathbf{x} = [x_1, x_2, \ldots, x_D]^T$, and D is the dimension, *i.e.*, the number of decision variables. Generally, for each variable x_j, it satisfies a boundary constraint, such that:

$$L_j \leqslant x_j \leqslant U_j, \quad j = 1, 2, \ldots, D. \tag{2}$$

2.2. Differential evolution

DE [35] is a simple evolutionary algorithm (EA) for global numerical optimization. It creates new candidate solutions by combining the parent individual and several other individuals of the same population. A candidate replaces the parent only if it has an equal or better fitness value. The pseudo-code of the original DE algorithm is shown in Algorithm 1, where D refers to the number of decision variables (or problem dimension); NP is the population size; F is the mutation scaling factor; CR is the crossover rate; $x_{i,j}$ is the jth variable of the solution x_i; $\mathbf{u}_i$ is the offspring. The function rndint$(1,D)$ returns a uniformly distributed random integer number between 1 and D, while rndreal$_j[0,1)$ gives a uniformly distributed random real number in $[0,1)$, generated anew for each value of j. With respect to the population initialization, the widely used method is uniformly random initialization within the search space. Other initialization methods are also available, for example, orthogonal initialization [13], opposition-based initialization [32], chaotical initialization [27], etc.