Expert Systems With Applications 55 (2016) 508–519

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Regular articles

GACE: A meta-heuristic based in the hybridization of Genetic Algorithms and Cross Entropy methods for continuous optimization

P. Lopez-Garcia[a,*], E. Onieva[a], E. Osaba[a], A.D. Masegosa[a,b], A. Perallos[a]

[a] Deusto Institute of Technology (DeustoTech), University of Deusto, Bilbao 48007, Spain
[b] IKERBASQUE, Basque Foundation for Science, Bilbao 48011, Spain

ARTICLE INFO

Article history:
Received 20 July 2015
Revised 22 January 2016
Accepted 16 February 2016
Available online 27 February 2016

Keywords:
Genetic Algorithm
Cross Entropy
Hybrid algorithm
Continuous optimization
Optimization functions

ABSTRACT

Metaheuristics have proven to get a good performance solving difficult optimization problems in practice. Despite its success, metaheuristics still suffers from several problems that remains open as the variability of their performance depending on the problem or instance being solved. One of the approaches to deal with these problems is the hybridization of techniques. This paper presents a hybrid metaheuristic that combines a Genetic Algorithm (GA) with a Cross Entropy (CE) method to solve continuous optimization functions. The algorithm divides the population into two sub-populations, in order to apply GA in one sub-population and CE in the other. The proposed method is tested on 24 continuous benchmark functions, with four different dimension configurations. First, a study to find the best parameter configuration is done. The best configuration found is compared with several algorithms in the literature in order to demonstrate the competitiveness of the proposal. The results shows that GACE is the best performing method for instances with high dimensionality. Statistical tests have been applied, to support the conclusions obtained in the experimentation.

1. Introduction

In the last decades, metaheuristics have been used extensively to solve complex optimization problems. Many of these algorithms have been inspired by natural phenomena and have great value in solving high dimensional problems. In this category are algorithms like Particle Swarm Optimization (PSO) (Kennedy, 2010), Genetic Algorithm (GA) (Holland, 1975), Ant colony Optimization(ACO) (Dorigo & Gambardella, 1997), Differential Evolution (DE) (Neri & Tirronen, 2010), or Simulated Annealing (SA) (Van Laarhoven & Aarts, 1987). Since their formulation, these algorithms have been applied to optimization problems in Wang et al. (2011), Thakur (2014), Ciornei and Kyriakides (2012) and Cai and Ma (2010). Also, probabilistic techniques such as Cross Entropy (CE) (Rubinstein, 1999) or Covariance Matrix Adaptation (CMA) (Hansen & Ostermeier, 2001) have been applied to this kind of problem (Deb, Anand, & Joshi, 2002; Kroese, Porotsky, & Rubinstein, 2006).

Despite its success in continuous problems and the large number of existing techniques, metaheuristics still suffers from several problems that remains open. One of them is the variability of its performance, depending on the characteristics of the optimization function. Another issue to take into account are the weaknesses strengths that each technique presents. For example, population-based metaheuristics like GA and ACO have problems with the exploitation of the search space (Talbi, 2002). On the other hand, regarding trajectory-based algorithms, as SA or Tabu Search, they easily become stuck in local optima because of their bad exploratory behavior (Wang, Wong, & Rahman, 2004). In the case of DE, the specific way in which new individuals are created or the potential to generate only a limited number of different trial solutions within one generation are identified as problems to this method (Segura, Coello, & Hernández-Díaz, 2015). Another problem to take into account is the called Algorithm Selection Problem. Due to the many available algorithms, it is not an easy task to know which one is able to exploit better the information. An example of this kind of problem for continuous optimization is presented in Muñoz, Kirley, and Halgamuge (2013). One of the approaches to deal with these problems is the association of two or more algorithms in order to obtain a better one or counteract their drawbacks. In fact, choosing a satisfactory combination of algorithms can be an essential part for achieving better performance in many hard optimization problems. This combination is called Hybridization (Topcuoglu, Demiroz, & Kandemir, 2007).

The principal aim to hybridize different algorithms is to benefit from the sinergy between their complementary weaknesses and

* Corresponding author. Tel.: +34 944139003x2050.
E-mail addresses: p.lopez@deusto.es (P. Lopez-Garcia), enrique.onieva@deusto.es (E. Onieva), e.osaba@deusto.es (E. Osaba), ad.masegosa@deusto.es (A.D. Masegosa), perallos@deusto.es (A. Perallos).

http://dx.doi.org/10.1016/j.eswa.2016.02.034
0957-4174/

strengths. Hybrid algorithms have proved to be promising in many fields in general (Fujikawa & Takashi, 2005; Olama et al., 2015; Purwar & Singh, 2015), and in particular in optimization problems such as constrained problems (Hernández, Leguizamón, & Mezura-Montes, 2013), nonlinear problems (Abd-El-Wahed, Mousa, & El-Shorbagy, 2011), or real world problems (Asafuddoula, Ray, & Sarker, 2011; Mandal, Das, Mukherjee, Das, & Suganthan, 2011).

In this article, a novel hybridization technique is presented. The proposal is based on a hybridization between GA and CE for solving continuous optimization problems. There are several reasons which have motivated the development of this study:

- The proposal aims at finding sinergies between the good exploration and exploitation abilities of GA and CE, respectively.
- Both methods have been successfully applied separately and many papers have been published focused on them (Busoniu, Ernst, De Schutter, & Babuska, 2011; Wang, Zhang, & Yang, 2013; Zhao, Wang, Yu, & Chen, 2013). However, as far as we know, the hybridization between these two methods has not been done before. Therefore, it could be an interesting approach to hybridize both method in order to achieve better performance than for its own.

The basic concept of the proposed technique is the following: the algorithm divides the population into two sub-populations of a given size. Then, GA is applied to one of these sub-populations and CE is applied to the other. As result, the new individuals created by the algorithms will form the new population. The algorithm has been tested over 24 benchmark functions extracted from Black-Box Optimization Benchmarking (BBOB),[1] which is part of the GECCO and CEC international conferences. Furthermore, it will be compared with reference algorithms in the literature to demonstrate its performance on this kind of problem. To conclude, the objectives of this paper can be summarized as follows.

- Hybridize two well-known algorithms, such as GA and CE, in order to improve on the performance obtained by those algorithms on their own.
- Apply this hybridization to continuous optimization problems.
- Find a successful combination of parameters to obtain a good performance in all the functions used.
- Compare the performance of the proposal with that of methods in the literature to prove its potential.

The work developed in this article is an extension of the research presented by the authors in Genetic and Evolutionary Computation Conference 2015 as two-page Late-Breaking Abstract (Lopez-Garcia, Onieva, Osaba, Masegosa, & Perallos, 2015). The novelties in this work are listed below:

- The number of functions used have been increased to 24, the double as in previous work.
- A wide study of the parameters used in the algorithm has been done. Population sizes and special parameters of each part of the method have been studied in order to obtain the best configuration possible.
- The number of different dimension values considered have been increased.
- The proposal have been compared with new high-performance methods from literature.
- Statistical tests have been applied in order to prove the significance of the results obtained by the presented method.

The research posed in this paper is of high relevance in artificial intelligence due to the wide number of applications that continuous optimization methods have in this field. One of the most common applications is the tuning of the hyper-parameters of machine learning models. For example, a previous version of our proposal was applied to the optimization of Fuzzy Rule-Based Systems, and then used in short-term congestion forecasting (Lopez-Garcia, Onieva, Osaba, Masegosa, & Perallos, 2016). Other application areas within the area of artificial intelligence are text categorization (Ghareb, Bakar, & Hamdan, 2016), optimization of real-world application problems (Yi, Zhou, Gao, Li, & Mou, 2016), robotics (Hsu & Juang, 2013), artificial vision (Santamaría, Damas, García-Torres, & Cordón, 2012), or speech segmentation (Iliya, Neri, Menzies, Cornelius, & Picinali, 2014).

This article is structured as follows. In Section 2 a brief explanation of the different parts of the proposal and its application in the literature is given. A brief explanation about types of hybridized methods, and state of the art about them are described in Section 3. Section 4 explains how the proposal works and what operators are used. The results of the experimentation and an analysis of the results are presented in Section 5. Finally, some conclusions and avenues for further research are presented in Section 6.

2. Background

In this section, a brief background of the different parts of the proposal is given. In Section 2.1, an explanation of GA is given and some of the related literature is reviewed. On the other hand, Section 2.2 contains the description of CE and some related research.

2.1. Genetic Algorithm

GAs were introduced by Holland (1975).They were designed to mimic some of the processes observed in natural evolution. A GA maintains a population of solutions, called individuals, and iteratively modifies them using different operators in order to achieve improvements. Its adaptability to hard problems has led GAs to appear in the literature both on their own (Osaba, Diaz, & Onieva, 2014; Osaba et al., 2013), as well as combined with different techniques (Onieva et al., 2011; Qiao, Yang, & Gao, 2011), to solve a wide variety of problems. A GA is formed most of time by the four operators: selection, crossover, mutation, and replacement.

The state of the art about GA is wide. Interested readers are referred to Lim (2014), Kumar and Beniwal (2013), and Karakatic and Podgorelec (2015) for extensive reviews of GAs in the literature.

In Algorithm 1, a pseudocode of the basic GA is depicted,

Algorithm 1: Pseudocode of workflow followed by GA.

Data: $Size_{POP}$, p_c, p_m, T_{max}
Result: $Best\ individual\ found$
$t \leftarrow 0$
$POP_0 \leftarrow$ Initialize($Size_{POP}$)
Evaluate POP_0
while $t < T_{max}$ **do**
  $Parents \leftarrow$ Select parents from POP_t
  $Offspring \leftarrow$ Crossover($Parents$, p_c)
  $Offspring \leftarrow$ Mutate($Offspring$,p_m)
  Evaluate $Offspring$
  $POP_{t+1} \leftarrow$ Replacement process with actual Population POP_t and $Offspring$
  $t \leftarrow t + 1$
end

where p_c and p_m denote the crossover and mutation probabilities, respectively.

[1] http://coco.gforge.inria.fr/doku.php?id=bbob-2013.