Information Sciences 372 (2016) 773–795

Contents lists available at ScienceDirect

Information Sciences

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

A new algorithm for adapting the configuration of subcomponents in large-scale optimization with cooperative coevolution

Giuseppe A. Trunfio [a,*], Paweł Topa [b], Jarosław Wąs [c]

[a] DADU, University of Sassari, P.zza Duomo, 6, 07041 Alghero (SS), Italy
[b] AGH University of Science and Technology, Department of Computer Science, Kraków, Poland
[c] AGH University of Science and Technology, Department of Applied Computer Science, Kraków, Poland

ARTICLE INFO

Article history:
Received 13 April 2016
Revised 21 August 2016
Accepted 26 August 2016
Available online 28 August 2016

Keywords:
Large scale optimization
Cooperative coevolution
Adaptation
Differential evolution

ABSTRACT

The cooperative coevolutionary (CC) approach can be very effective in solving problems of large-scale continuous optimization (LSGO) through their decomposition into lower-dimensional subcomponents. However, it is well known that the CC performance can be significantly influenced by the adopted decomposition. Moreover, since the method may require evolving a number of populations, also the size of the latter can largely affect the optimization process. In this article, focusing on equally sized decompositions, we present the results of an in-depth investigation concerning the effects of both the size of populations and the dimensionality of subcomponents on the performance of a CC optimizer. According to our study, in several cases only a small set of suitable configurations corresponds to a high optimization performance. Furthermore, we propose a new CC algorithm in which part of the available computational budget is spent for adapting both the dimensionality of subcomponents and the number of evolved individuals during the optimization process. Using a rich set of benchmark problems, we show that the proposed approach can outperform a state-of-the art algorithm based on adaptive equally sized decompositions.

1. Introduction

In recent years, the study of effective and efficient search strategies to address high-dimensional optimization problems in the continuous domain has been recognized as a relevant field of research, which is often referred to as Large-Scale Global Optimization (LSGO) [18,22]. At present, a variety of approaches have been proposed to effectively address LSGO problems by improving the scalability of evolutionary optimizers [22,44]. Among the most successful algorithms, there are those based on Cooperative Coevolution (CC), which is a divide-and-conquer strategy originally proposed by Potter and De Jong in [31]. An algorithm based on CC decomposes the original high-dimensional problem into a set of lower-dimensional *subcomponents*, which should be easier to deal with. Inside each subcomponent, population-based evolutionary optimizers operate independently, except for fitness evaluations, which require a cooperation in terms of information exchange.

According to the literature, the CC approach has been successfully tested with several optimizers, such as Genetic Algorithms [31], Particle Swarm Optimization (PSO) [6,19,30,45], Ant Colony Optimization [5], Simulated Annealing [36], Differ-

* Corresponding address. Fax: +39 079 9720420.
E-mail address: trunfio@uniss.it (G.A. Trunfio).

http://dx.doi.org/10.1016/j.ins.2016.08.080

ential Evolution (DE) [38,47], Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [20,24], Firefly Algorithm [42] and many others.

Clearly, the underlying optimizer, together with its parameters setting, population size and initialization, provide different optimization powers with different problems and, thus, strongly influence the search performance of a CC algorithm.

Moreover, high-dimensional continuous problems admit a number of possible decompositions, which typically correspond to a different CC optimization efficiency. A well-known issue related to the decomposition of search space may originate from the presence of interacting variables [31,32]: placing them into different subcomponents may cause a significant decay of the optimization performance. For this reason, automatic decomposition methods have been developed for dealing with problems in which subsets of interacting variables can be recognized and grouped together [1,11,24,27,28,39,43]. Nevertheless, the resulting groups of variables may still give rise to high-dimensional problems, for which a further decomposition can be useful. In addition, many LSGO problems are either fully separable (i.e., they do not contain interacting variables) or fully non-separable (i.e., all the variables interact with each other). For such cases, a typical decomposition approach consists of placing the same number of variables in each subcomponent. However, also the resulting common size of subcomponents may significantly affect the search efficiency of a given optimizer used within a CC algorithm [29].

For the above reasons, researchers devised CC algorithms able to adapt the size of subcomponents during the search process. According to the literature, the problem has been first addressed by Yang et al. in [48] and, more recently, by Omidvar et al. in [29] through an adaptive approach based on a reinforcement learning (RL) technique [40]. Unfortunately, as noted in [29], learning a suitable size of subcomponents trough RL is not easy due to the non-stationary nature of the evolutionary process. In addition, previous research work neglected to consider the combined effect of both subcomponents size and number of individuals composing the corresponding populations, which can have instead a significant influence in the case of a CC algorithm. Indeed, differently from an ordinary evolutionary optimizer based on a single population, CC may involve a number of populations. Thus, when there is an overall constraint on the available computational budget, choosing a suitable number of individuals to evolve in each population can be much more critical to the optimization process.

Focusing on the issue of equally sized subcomponents, in this article we make two contributions. Firstly, using a suite of LSGO benchmark problems, we present and discuss a comprehensive computational study on the effect that the chosen configuration of subcomponents (i.e., their size and number of evolved individuals) may have on the search performance of a standard CC optimizer. In particular, we show that choosing the correct configuration may be crucial in determining the success of the optimization. As a second contribution, we present a new approach to address the difficult problem of adapting the configuration of subcomponents in a CC algorithm. In the proposed method, a pool of alternative configurations operate in parallel during short *comparison phases*. Such an approach, compared with the use of one decomposition size at a time, as proposed in [29,48], enables a more reliable evaluation of the candidate configurations. Using a numerical investigation, we show that the proposed algorithm can outperform the most effective method previously presented in the literature.

The article is organized as follows. Section 2 outlines a typical CC optimizer. Then, Section 3 provides a background on the most relevant methods dealing with the adaptation of subcomponents size. In Section 4, we describe in detail the proposed approach. Section 5 contains a comprehensive computatonal study in which: (*i*) we highlight the importance of selecting a suitable configuration for the CC algorithm; (*ii*) we investigate the proposed algorithm using a well-known suite of benchmark LSGO problems, also comparing the proposed approach with a state-of-the-art algorithm for adaptive equally sized decomposition. Section 6 concludes the article and outlines possible future work.

2. Cooperative coevolution

We consider the following minimization problem:

$$\min f = F(\mathbf{x}), \quad \mathbf{x} \in \mathbf{S} \tag{1}$$

where $\mathbf{S} \subseteq \mathbb{R}^d$ denotes the search space with d variables and $F : \mathbf{S} \to \mathbb{R}$ is a real-valued objective function. The CC idea [31] consists of partitioning the d-dimensional set of search directions $G = \{1, 2, \dots, d\}$ into k sets $G_1 \dots G_k$. Each group G_i of variables corresponds to a new search space $\mathbf{S}^{(i)}$ for the problem defined by Eq. (1), in which the remaining variables x_j, with $j \notin G_i$, are kept constant. Thus, the whole search procedure is decomposed into k *subcomponents* associated to lower-dimensional problems, which are typically addressed through population-based evolutionary algorithms.

The latter can be executed independently within each subcomponent, except when evaluating the fitness function F. In fact, a candidate solution in $\mathbf{S}^{(i)}$ contains only some elements of the d-dimensional vector $\mathbf{x} \in \mathbf{S}$. Thus, in the CC strategy a common d-dimensional *context vector* $\mathbf{b}$ is built using a representative individual (e.g., the current best) provided by each subcomponent. Then, the candidate solutions are evaluated by complementing them through the appropriate elements of the context vector. In practice, the cooperation consists of using a common vector in which the subcomponents make available their contribution for the fitness evaluation of all individuals.

After the first applications presented in [31], researchers found that the CC approach can significantly improve the scalability of an evolutionary optimizer as the problem dimensionality increases [21].

In the original implementation, the d-dimensional problem was decomposed into d sub-populations (i.e., $G_i = \{i\}$). Subsequently, the idea was generalized by introducing a decomposition of the original d-dimensional search space into k subcomponents with the same dimension $d_k = d/k$ [45]. Such an approach can be formalized by defining the groups of dimensions