Information Sciences 316 (2015) 419–436

Contents lists available at ScienceDirect

Information Sciences

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

Designing benchmark problems for large-scale continuous optimization

Mohammad Nabi Omidvar [a,*], Xiaodong Li [a], Ke Tang [b]

[a] Evolutionary Computing and Machine Learning (ECML) Group, School of Computer Science and IT, RMIT University, Melbourne, VIC 3001, Australia
[b] Nature Inspired Computation and Applications Laboratory (NICAL), Department of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China

ARTICLE INFO

Article history:
Received 21 November 2013
Received in revised form 28 November 2014
Accepted 4 December 2014
Available online 9 January 2015

Keywords:
Benchmark
Large-scale optimization
Evolutionary computation

ABSTRACT

Three major sources of complexity in many real-world problems are size, variable interaction, and interdependence of the subcomponents of a problem. With the rapid growth in the size of businesses, the demand for solving large-scale complex problems will continue to grow. In this paper, we propose several major design features that need to be incorporated into large-scale optimization benchmark suites in order to better resemble the features of real-world problems. Non-uniform subcomponent sizes, imbalance between the contribution of various subcomponents of a problem, and the interaction between subcomponents by means of overlapping subcomponents are among these features. The proposed features are designed with the aim of closing the gap between the theory and practice of evolutionary techniques for solving large-scale continuous optimization problems. The general guidelines proposed in this paper can be used to design and construct various benchmark suites to meet different needs. The IEEE CEC'2013 large-scale global optimization benchmark suite [29] is one such implementation. The paper also contains a brief discussion on how the CEC'2013 benchmarks can be extended or modified for various purposes. Finally, a preliminary comparative study is conducted to showcase the performance of several state-of-the-art algorithms on the CEC'2013 large-scale benchmark problems.

1. Introduction

Numerous metaheuristic algorithms have been successfully applied to many optimization problems [3,4,9,14,15,25–27,34,48,56]. However, their performance deteriorates rapidly as the dimensionality of the problem increases [5,31]. There are many real-world problems that exhibit such large-scale property [13,32,54,68,59], and the number of such large-scale optimization problems will continue to grow as we advance in science and technology.

Several factors make large-scale problems exceedingly difficult [62]. Firstly, the search space of a problem grows exponentially as the number of decision variables increases. Secondly, the properties of the search space may change as the number of dimensions increases. For example, the Rosenbrock function is a unimodal function in two dimensions, but it turns into a multi-modal function when the number of dimensions increases [51]. Rosenbrock's function is a well-known test function in numerical optimization and is characterized by its parabolic narrow valley where its global optimum resides.

* Corresponding author.
E-mail addresses: mohammad.omidvar@rmit.edu.au (M.N. Omidvar), xiaodong.li@rmit.edu.au (X. Li), ketang@ustc.edu.cn (K. Tang).

http://dx.doi.org/10.1016/j.ins.2014.12.062

Thirdly, the evaluation of large-scale problems is usually expensive. This is often the case in many real-world problems such as gas turbine stator blades [22], multidisciplinary design optimization [55], and target shape design optimization [37].

Another factor that contributes to the difficulty of large-scale problems is the interaction between variables. Two variables interact if they cannot be optimized independently to find the global optimum of an objective function. Variable interaction is commonly referred to as *non-separability* in continuous optimization literature. In genetic algorithms literature, this phenomenon is commonly known as *epistasis* or *gene interaction* [11,8]. In an extreme case where there is no interaction between any pair of the decision variables, a large-scale problem can be solved by optimizing each of the decision variables independently. The other extreme is when all of the decision variables interact with each other and all of them should be optimized together. However, most of the real-world problems fall in between these two extreme cases [60]. In such problems, usually a subset of the decision variables interact with each other, forming several clusters of interacting variables.

Other general factors that contribute to the difficulty of a problem are noise, modelling issues, constraints, and interaction between subcomponents [34]. However, constrained optimization, optimization of noisy functions, and modelling are beyond the scope of this study. For constrained optimization, it suffices to mention that the method of Augmented Lagrangian [6] is a widely-used technique to convert a constrained optimization into an unconstrained one. Unlike the method of Lagrange multipliers [6], which requires the calculation of the first order derivative of the Lagrangian, Augmented Lagrangian does not require calculation of derivatives.

The IEEE CEC'2010 benchmark suite [57] consists of large-scale modular problems with no interdependence between various subcomponents of a given problem. It is argued that subcomponent interaction is commonplace in many real-world problems [33]. For example, each component of a supply-chain problem is called a *silo*, and most supply-chain problems are *multi-silo* problems with interaction between silos [35]. In an attempt to better represent this class of real-world problems, in this paper we propose a method for creating interaction between subcomponents.

The modular nature of many real-world problems makes a *divide-and-conquer* approach appealing for solving large-scale optimization problems. In the field of evolutionary optimization, these divide-and-conquer approaches are commonly known as decomposition methods [10,17,16]. Some algorithms, such as estimation of distribution algorithms (EDAs) [36,43–46], perform an implicit decomposition by approximating a set of joint probability distributions to represent each interaction group. Other methods such as cooperative co-evolution (CC) [47] explicitly subdivide a large-scale problem into a set of smaller subproblems [61]. Cooperative co-evolutionary methods in particular have gained popularity in recent years in the context of large-scale optimization [7,30,31,39,40,65,64]. When the problem is very large, it is prohibitive to optimize all the decision variables at once. Therefore, it is desirable to divide a large-scale complex problem into a set of smaller problems. However, in the presence of interaction between subcomponents, there is no unique optimal decomposition. It has been suggested recently that cooperative co-evolution is a promising framework for solving complex multi-silo problems [33]. A CC framework is advantageous for two major reasons. Firstly, it can potentially exploit the modular nature of a problem by optimizing various subcomponents separately. Secondly, if some of the subcomponents interact, CC's collaborative scheme for evaluating the potential solutions allows information sharing between subcomponents.

The IEEE CEC'2010 benchmark suite [57] was designed with the aim of providing a suitable evaluation platform for testing and comparing large-scale optimization algorithms. To that end, the CEC'2010 benchmark suite is successful in representing the modular nature of some real-world problems, and building a scalable set of benchmark functions in order to promote research in the field of large-scale global optimization. However, recent advances in this area signal the need to revise and extend the existing benchmark suite. For example, the differential grouping algorithm [38] can now detect the grouping structure of the CEC'2010 benchmark problems with 100% accuracy for most of the functions in the test suite. The aim of this paper is to build upon the ideas originally proposed in the CEC'2010 benchmark suite and propose a set of guidelines for designing large-scale benchmark problems in order to better represent the features of a wider range of real-world problems, as well as posing some new challenges to the existing algorithms, especially to decomposition-based algorithms. As a result, the CEC'2013 benchmark suite for large-scale optimization has recently been proposed [29]. This paper will focus on presenting the following three key features that were included in the new CEC'2013 benchmark suite:

- Non-uniform subcomponent sizes (Section 5.1).
- Imbalance in the contribution of subcomponents (Section 5.2).
- Functions with overlapping subcomponents (Section 5.3).

In this paper, we explain how each of these features poses a challenge to a class of optimization algorithms. Where appropriate, an empirical approach is employed to demonstrate the effect of these features on several selected state-of-the-art metaheuristic optimization algorithms (see Section 3.2).

The organization of the remainder of this paper is as follows: Section 2 gives a brief outline of the existing large-scale benchmark suites and outlines the need for new benchmarks. Section 3 contains the required definitions and explanations of the algorithms used in this paper. Section 4 outlines the mathematical definitions of various categories of functions. In Section 5, the newly proposed features such as non-uniform subcomponent size, imbalance, and overlapping functions are explained, and relevant experimental results are given where appropriate. Section 6 shows how the proposed CEC'2013 benchmark suite can be extended and used for various types of research. Section 7 contains a brief comparative study on the performance of several state-of-the-art algorithms on the CEC'2013 benchmark suite. Finally, Section 8 concludes the paper and gives a list of open research questions.