Information Sciences 367–368 (2016) 487–511

Contents lists available at ScienceDirect

Information Sciences

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

A self-adaptive binary differential evolution algorithm for large scale binary optimization problems

Akbar Banitalebi, Mohd Ismail Abd Aziz*, Zainal Abdul Aziz

UTM Center for Industrial and Applied Mathematics, Ibnu Sina Institute for Scientific and Industrial Research, Universiti Teknologi Malaysia, UTM Skudai, 81310, Johor, Malaysia

ARTICLE INFO

Article history:
Received 2 October 2015
Revised 16 May 2016
Accepted 23 May 2016
Available online 7 June 2016

Keywords:
Binary optimization
Large-Scale global optimization
Self-adaptation
Differential evolution algorithm
Knapsack problem

ABSTRACT

This study proposes a new self-adaptive binary variant of a differential evolution algorithm, based on measure of dissimilarity and named SabDE. It uses an adaptive mechanism for selecting how new trial solutions are generated, and a chaotic process for adapting parameter values. SabDE is compared against a number of existing state of the art algorithms, on a set of benchmark problems including high dimensional knapsack problems with up to 10,000 dimensions as well as on the 15 learning based problems of the Congress on Evolutionary Computation (CEC 2015). Experimental results reveal that the proposed algorithm performs competitively and in some cases is superior to the existing algorithms.

1. Introduction

Binary optimization problems arise in many different disciplines ranging from social sciences and management to computer sciences and mathematics. A typical instance of these problems can be illustrated by a mathematical optimization or feasibility problem of detecting an optimal object from a finite set of objects. The solutions of many real life optimization problems can be modeled in the form of binary optimization problems. These include classification and clustering problems, compression related problems, network optimization, cell formation, unit commitment and knapsack problems [2,29].

The traditional approaches to address binary optimization problems include enumeration schemes, Lagrangian techniques, relaxation methods, reduction schemes, branch-and-bound methods and genetic algorithm [44,45]. These methods have either been designed for some specific classes of problems or worked effectively on low scale problems. It seems very tempting to start enumerating all the solutions to find a global optimizer. However, the scales of practical problems are usually very large which prohibits exhaustive search. As the dimension of problems increases, the problems become more and more complicated and being riddled with many local optima. The issue is even more complicated if the problems involve big data. In addition, the binary feature of the decision space leads to non-differentiability and discontinuity of the problems which makes it very difficult to apply deterministic methods.

Alternatively, there have been many studies on stochastic optimization methods aiming to address discrete binary problems. These methods include Particle Swarm Optimization (PSO) [2], Differential Evolution (DE) [7], Artificial Bee Colony (ABC) [36], Gravitational Search Algorithm (GSA) [34], Harmony Search (HS) [28], and Simultaneous Perturbation Stochastic Approximation (SPSA) [46]. As these methods have initially been developed under assumption that the search space is

* Corresponding author. Tel: +6075530231; fax: +6075579385.
E-mail addresses: akbar.banitalebi@utm.my (A. Banitalebi), mismail@utm.my, mohdismail.abdaziz@gmail.com (M.I.A. Aziz), zainalaz@utm.my (Z.A. Aziz).

http://dx.doi.org/10.1016/j.ins.2016.05.037

continuous, the original methods are not effective on binary problems. Therefore, a few modification techniques have been proposed as follow:

- *Transfer function*: Kennedy and Eberhart [24] devised a sigmoid function to approximate real variables into binary. This technique first applied on PSO where a signomial function absorbs the value of velocity into zero or one depending on its original real value. This idea was further improved by introducing new transfer functions [2,35].
- *Angle modulation*: This technique was derived from the telecommunication industry and it is primarily used in the field of signal processing. The underlying idea behind this method is to use a composed sine and cosine function to generate binary arrays. The function has four real valued parameters and each set of parameters represents only one binary array. This facilitates to convert a binary search space into an equivalent four dimensional continuous space[37].
- *Quantum-inspired bits*: As the name suggests this technique was inspired by the field of quantum computing where each agent is defined by a pair of numbers. A rotation matrix is then used to generate trial solutions where the rotation angle is continuously updated [34].
- *Genetic operators*: The binary feature of the search space permits to employ genetic operators such as binary crossover and swap operator in evolutionary algorithms. For instance, Ozturk et al.[36] proposed an efficient binary variant of the ABC algorithm where the genetic operators are employed to design special search strategies.
- *Binary operators*: As the search space consists of binary arrays, it is also possible to make use of binary operators such as *xor*, *and*, *or*, and *not* operators to generate trial solutions [11].
- *Measure of dissimilarity*: The techniques in this approach define a method to calculate the dissimilarity of two binary arrays. This eases the application of modern common search strategies for binary optimization. The Jaccard's coefficient of similarity and Hamming distance are often considered as measure of dissimilarity [23].
- *Heuristic methods*: There are other experimental methods which are specially designed [7].

Based on the complexity and scale of the target problems, any of the aforementioned methods can be selected to design a suitable search strategy. The binary algorithms that follow the first two approaches work with continuous search engines. Although the angle modulation technique reduces any binary problem into a four dimensional continuous problem, if the dimension of the binary problem is high, the solution of the continuous problem is required to be extremely accurate. However, these methods are efficient for low scale problems. On the other hand, the quantum-inspired bits and binary or genetic operators are versatile techniques that have been employed in developing innovative modern binary algorithms [34,36]. Furthermore, the techniques based on measuring the dissimilarity of binary arrays enable equipping binary algorithms with conventional search strategies.

In this paper, following the differential evolution framework and dissimilarity measure approach, a new self-adaptive binary differential evolution (SabDE) algorithm is presented. A novel technique is introduced in the proposed algorithm to improve the quality of the solution, accelerate the convergence of the algorithm and yet preserve the main characteristics of the DE. The SabDE is then implemented and compared with some state of the art binary algorithms.

The rest of this paper is organized as follow, the next section briefly overviews some related works, the third section consists of a detailed description of the proposed algorithm, following that this algorithm is implemented, its performance is analyzed and compared against the recently proposed binary algorithms. Finally, the conclusion is given.

2. Related works

In this section, the DE algorithm and its variants are briefly reviewed, and some prominent binary optimization algorithms are also introduced.

2.1. The DE

The original DE algorithm was introduced by Storn and Price [48]. The DE soon emerged as a powerful search technique, and it was established into a competitive derivative free optimization framework. The algorithm consists in some simple steps and benefits from a small number of parameters. It is configured by setting proper values for the parameters, scale factor (F), crossover rate (CR), and number of population (NP). It is then initialized by generating a uniform random population of trial individuals $\mathbf{X}_{i,g} = (x_{ij,g})$ for $i = 1, 2, \dots, NP$ and $j = 1, 2, \dots, D$ where D is the dimension of the problem. The index g shows the generation number that is initially set at 1.

The next steps iteratively improve the quality of the trial populations. At each iteration a new generation takes place of the previous one. In that, a new tentative population $\mathbf{V}_{i,g+1} = (v_{ij,g+1})$ is generated using a suitable search strategy. The index $g+1$ indicates the next generation. The most applied strategies are listed as follow:

$$v_{ij,g+1} = x_{r_1j,g} + F(x_{r_2j,g} - x_{r_3j,g}) \tag{1}$$

$$v_{ij,g+1} = x_{bj,g} + F(x_{r_2j,g} - x_{r_3j,g}) \tag{2}$$

$$v_{ij,g+1} = x_{r_1j,g} + F(x_{r_2j,g} - x_{r_3j,g}) + F(x_{r_4j,g} - x_{r_5j,g}) \tag{3}$$