Centroid Opposition-Based Differential Evolution

Shahryar Rahnamayan, Electrical, Computer, and Software Engineering Department, University of Ontario Institute of Technology, Oshawa, Ontario, Canada

Jude Jesuthasan, Electrical and Computer Engineering Department, University of Waterloo, Waterloo, Ontario, Canada

Farid Bourennani, Electrical, Computer, and Software Engineering Department, University of Ontario Institute of Technology, Oshawa, Ontario, Canada

Greg F. Naterer, Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, Canada

Hojjat Salehinejad, Electrical, Computer, and Software Engineering Department, University of Ontario Institute of Technology, Oshawa, Ontario, Canada

ABSTRACT

The capabilities of evolutionary algorithms (EAs) in solving nonlinear and non-convex optimization problems are significant. Differential evolution (DE) is an effective population-based EA, which has emerged as very competitive. Since its inception in 1995, multiple variants of DE have been proposed with higher performance. Among these DE variants, opposition-based differential evolution (ODE) established a novel concept in which individuals must compete with theirs opposites in order to make an entry in the next generation. The generation of opposite points is based on the current extreme points (i.e., maximum and minimum) in the search space. This paper develops a new scheme that utilizes the centroid point of a population to calculate opposite individuals. The classical scheme of an opposite point is modified. Incorporating this new scheme into DE leads to an enhanced ODE that is identified as centroid opposition-based differential evolution (CODE). The accuracy of the CODE algorithm is comprehensively evaluated on well-known complex benchmark functions and compared with the performance of conventional DE, ODE, and some other state-of-the-art algorithms. The results for CODE are found to be promising.

Keywords: *Centroid Opposition-Based Computation (CODE), Differential Evolution (DE), Evolutionary Computation, Opposition-Based Differential Evolution (ODE), Opposition-Based Learning (OBL), Optimization*

1. INTRODUCTION

The population-based algorithms are useful tools to deal with optimization problems such as in communication systems design (Salehinejad & Talebi, 2012; Koroupi & Salehinejad, 2012), vehicle navigation (Salehinejad & Talebi, 2010), and smart grids (Pouladi & Salehinejad, 2014).

DOI: 10.4018/ijamc.2014100101

The differential evolution (DE) algorithm presents a higher performance over many other EAs because of its simplicity, effectiveness, and lower number of control parameters (Storn & Price, 1995). Over the past decade, because of its high performance, DE has been widely used to solve global optimization problems in various engineering and science fields. Similarly to other EAs, DE is a population-based stochastic algorithm which suffers from slow convergence speed. Many researchers investigated how to accelerate DE. Researches have undertaken to improve the performance of DE by control parameter fine-tuning and trial vector generation strategies for specific problems (Price & Storn, 2005), as well as obtaining those values and strategies self-adaptively (Qin & Huang, 2009). The idea of opposition-based learning can be divided into Type-I and Type-II categories (Salehinejad & Rahnamayan, 2014). The Type-I opposition learning has been employed to accelerate DE convergence speed and obtain more accurate solutions, known as opposition-based differential evolution (ODE) (Rahnamayan & Tizhoosh, 2008). By using opposition-based computation (OBC) concepts (Tizhoosh, 2005), the ODE benefits from a stronger exploration capability resulting in faster convergence. In ODE, OBC served to generate the opposite of current candidate solutions. OBC defines an opposite-point based on a predefined relationship between the extreme points (i.e., max and min) of a current population and the trial point. Thus, applying OBC to a candidate-solution generated by DE means that the opposite solution is calculated by using coordinates of the candidate solution, the maximum and the minimum of the population along each dimension as boundary points. In this paper, a new OBC scheme called centroid opposition-based computation (COBC) is proposed to meet the above features.

This is the first time where OBC is based on the *entire* population, by using centroids, rather than the min and max points. The centroid is the point where the centre of mass lies in a uniform body. For this matter, it is assumed that the entire population of DE is a discrete body; so the unit mass is distributed for each candidate-solution in the search space. The use of a centroid does not strengthen the learning process of an algorithm.

A new variant of ODE that uses the COBC scheme is proposed and named centroid-based opposition different evolution (CODE) (Rahnamayan & Jesuthasan, 2014). The comprehensive experimental results in this paper confirm that the CODE algorithm is faster than both ODE and DE and it also provides higher accuracy than its counterparts.

The remaining of the paper is organized as follows. The conventional DE is briefly explained in Section II. A brief review of conventional OBC is provided in Section III. The proposed CODE algorithm described in Section IV. Then, the performance of CODE is compared with DE, ODE, and other state-of-the-art algorithms in Section V. Finally, the paper is concluded in Section VI.

2. DIFFERENTIAL EVOLUTION (DE)

The population-based algorithms can be divided into standard and micro population algorithms. In the micro-based algorithms, a small size of population is utilized (Salehinejad & Rahnamayan, 2014; Salehinejad & Zadeh, 2014). The DE algorithm is an effective population-based optimization algorithm which can be utilized to solve an optimization problem formalized as follows.

$$\text{Minimize}\, f\left(x\right), x = \left[x_1, x_2, \ldots\ldots, x_D\right] \epsilon\ R \qquad (1)$$

$$\text{Subject to}\, g_i\left(x\right) \leq 0, i = 1,2,\ldots, p$$

where

$$x_i^{(L)} \leq x_i \leq x_i^{(U)}, i = 1, 2, \ldots, D \qquad (2)$$

Related Content