Applied Mathematics and Computation 231 (2014) 601–618

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

An adaptive hybrid differential evolution algorithm for single objective optimization

Md. Asafuddoula *, Tapabrata Ray, Ruhul Sarker

School of Engineering and Information Technology, University of New South Wales, Canberra, Australia

ARTICLE INFO

Keywords:
Differential evolution
Local search
Self adaptive
Constrained single objective optimization

ABSTRACT

A number of efficient differential evolution (DE) algorithms have been proposed in recent years to deal with constrained single objective optimization problems. Past studies have indicated that the performance of such algorithms is largely affected by the choice of parameters e.g., mutation factor, crossover rate, mutation strategy and the type of crossover. A combination of these parameters may work out to be the best for a problem while resulting in poor performance for others. This paper introduces an adaptive hybrid DE algorithm (AH-DEa). The algorithm employs a binomial crossover in early stages of evolution for exploration, while an exponential crossover is employed for exploitation in later stages. In addition, the crossover rate (CR) is adaptively controlled based on the success of offspring/trial solutions generated. A local search is initiated from the best found solution to explore possibilities of further improvement. Results of the proposed algorithm are compared with existing state of the art algorithms on a set of 40 widely studied mathematical benchmarks and two shape matching problems. The benefits of adaptive CR selection are highlighted. The results indicate that the proposed algorithm is able to identify better or comparable results across the wide range of single objective optimization problems.

1. Introduction

There has been a plethora of efficient optimization algorithms proposed in recent years to deal with single objective optimization problems. Many of these algorithms belong to the class of DE. They appear to be very competitive against other nature inspired optimization schemes as evident from their performance on recent mathematical benchmarks of *CEC-2006* [1] and *CEC-2010* [2] competitions. The evolving complexity of these test problems have led to significant changes in the native DE. Improved constraint handling schemes, mutation strategies, adaptive parameter tuning and hybridization has all become an integral part of today's efficient forms of DE. Parallel and similar developments can also be noticed in other forms of nature inspired schemes such as evolutionary algorithms, particle swarm optimization etc.

The behavior of a native DE is controlled by its mutation factor (F), crossover rate (CR), mutation strategy and the crossover type. A crossover rate of 0.9 and a mutation factor of 0.5 have been originally suggested in [3]. Further experiments however indicated that the performance of the algorithm is largely affected by the choice of these parameter values. The following discussion further elaborates on the range of variations and choices adopted in various forms of DE.

The mutation factor is perhaps the first parameter which has been identified to affect the performance of DE. Earlier forms of DEs used a fixed value of $F = 0.5$, while there are reports of F ranging between 0.1 and 1.0 [4], crowding distance based

* Corresponding author.
E-mail addresses: md.asaf@student.adfa.edu.au (M. Asafuddoula), t.ray@adfa.edu.au (T. Ray), r.sarker@adfa.edu.au (R. Sarker).

0096-3003/$ - see front matter
http://dx.doi.org/10.1016/j.amc.2014.01.041

selection [5], use of Gaussian with mean of 0.5 and a standard deviation of 0.3 [6] or the use of a Cauchy distribution with a mean of 0.5 and a standard deviation of 0.1 [7].

The crossover rate (CR) is yet another parameter which has also been identified to affect the performance of DE. Numerical experiments have indicated that a linearly separable problem can be efficiently solved using low CR values, while higher CR values are preferred for non-separable and multi-modal problems [8,9]. Various CR values have been used in past studies ranging from fixed values of 0.9 [3,10–12] for solving linear, non-linear and multi-modal functions, 0.7 [13] for solving cubic, polynomial and quadratic functions and 0.3 [14,15] for noisy optimization problems.

There are also a number of mutation strategies that have been proposed over the years. Commonly adopted mutation strategies include *DE/best/1*, *DE/rand/1*, *DE/current-to-best/1* and *DE/rand/2* [16,17]. Variants have also emerged in recent years, such as *DE/rand/1/-Either-Or-Algorithm* etc. [18]. Observations have indicated that *DE/rand/1* performs well for linearly separable, unimodal or non-separable and noisy functions [19,14]. Experiments also indicate that *DE/current-to-best/1* and *DE/rand/2* are effective for solving multi-modal and non-separable functions [16].

In terms of crossover types, two most promising ones include the binomial crossover and the exponential crossover. Studies in [19] indicate that the binomial crossover undergoing a binomial gene-wise crossover [20] is less greedy and has the ability to solve linearly separable and multi-modal problems [16]. The exponential crossover undergoing sequential participation of multiple genes [21] exploit more and tend to be useful for solving non-linear functions [10]. Two other forms i.e., *trigonometric mutation* and *arithmetic recombination* have also been proposed in recent years.

While in a native DE, three random parents are chosen for mutation [3,22], a number of recent algorithms have modified this basic parent selection scheme. In the works of [10], two parents were randomly selected from an active population, while the third was selected randomly from an archive. Instead of random selection of parents, a selection probability inversely proportional to its distance from the mutated individual was used for selecting multiple parents in [23] for the solution of unconstrained optimization problems. In an effort to further enhance the performance of DE variants, local search strategies have also been incorporated such as in the works of hybrid DE [24–26]. In such approaches, local searches are conducted sparingly and periodically from promising solutions [27]. In addition to all the parameters discussed above, constraint handling methods play an important role in the solution of constrained optimization problems. Notable recent approaches in the context of DEs include the use of ensembles of constraint handling schemes [13], use of epsilon constraint mechanism [21,10], mean violation approach [4], modified epsilon approach [28] and the use of conventional penalty functions as in [29].

With such a varying range of parameter values and strategies adopted within various forms of DEs, it is not surprising to come across studies that tend to adaptively select one or more of such parameters during the course of evolution. Fundamentally, such strategies can be classified into two different forms i.e., one in which the crossover rate (CR) and the mutation factor (F) is encoded within a solution and updated during the course of evolution [4] and others where the control parameters are adaptively identified and modified over generations as in SaDE [30], JADE [31] based on the success of trial solutions.

The purpose of this paper is to introduce an adaptive hybrid DE which inherits benefits of various proven strategies. The first feature is its use of adaptive crossover rates from a given set of discrete values spanning a range from 0.1 to 1.0. The mutation factor (F) is adapted following the scheme proposed by [10] and for the parent selection two parents are selected from the active population and the third from the archive. The basic mutation strategy is selected e.g., DE/rand/1 with a combination of binomial and exponential crossover depending upon the search strategy in different stages of evolution. The algorithm also incorporates a local search at the end of the DE process in an attempt to further improve the best solution obtained through DE evolution [25]. Furthermore, to avoid a premature convergence, a restart mechanism is introduced during the local search phase, wherein the population is reinitialized if no improvement to the best solution is achieved during the course of local search. In order to deal with constraints, an epsilon constraint handling scheme has been used.

The rest of this paper is structured as follows. The proposed DE algorithm along with its component strategies are presented in Section 2. The performance of the algorithm on the mathematical benchmarks and two shape matching problems are presented in Section 3 while Section 4 concludes the paper with final remarks.

2. Proposed algorithm

The proposed algorithm follows the generic structure of a differential evolution scheme. The algorithm maintains an active population and an archive of solutions. The algorithm creates a set of M individuals which is used to initialize the archive. Top N individuals of the archive are copied to the active population. In an attempt to maintain diversity, two parents are selected from the active population randomly, while the third parent is selected from the archive. Offspring solutions are created using binomial crossover during early phases of evolution, while exponential crossover is used in later generations. The fitness of i^{th} solution is determined as follows:

$$fitness_i(\xi) = \begin{cases} f_i(\mathbf{x}), & \mathbf{x} \in \Re^D \\ c_i, & \end{cases} \tag{1}$$

where c_i is the constraint violation measure based on the equality and inequality constraints and D represents the number of variables. Offspring solution is compared with the i^{th} individual in the active population for a possible replacement. If the