Computers & Operations Research 75 (2016) 132–149

Contents lists available at ScienceDirect

Computers & Operations Research

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

A novel differential evolution algorithm using local abstract convex underestimate strategy for global optimization

Xiao-gen Zhou, Gui-jun Zhang*, Xiao-hu Hao, Li Yu

College of Information Engineering, Zhejiang University of Technology, Hangzhou, Zhejiang 310023, PR China

ARTICLE INFO

Article history:
Received 22 April 2015
Received in revised form
11 February 2016
Accepted 24 May 2016
Available online 27 May 2016

Keywords:
Differential evolution
Global optimization
Supporting hyperplane
Underestimate
Abstract convexity

ABSTRACT

Two main challenges in differential evolution (DE) are reducing the number of function evaluations required to obtain optimal solutions and balancing the exploration and exploitation. In this paper, a local abstract convex underestimate strategy based on abstract convexity theory is proposed to address these two problems. First, the supporting hyperplanes are constructed for the neighboring individuals of the trial individual. Consequently, the underestimate value of the trial individual can be obtained by the supporting hyperplanes of its neighboring individuals. Through the guidance of the underestimate value in the select operation, the number of function evaluations can be reduced obviously. Second, some invalid regions of the domain where the global optimum cannot be found are safely excluded according to the underestimate information to improve reliability and exploration efficiency. Finally, the descent directions of supporting hyperplanes are employed for local enhancement to enhance exploitation capability. Accordingly, a novel DE algorithm using local abstract convex underestimate strategy (DELU) is proposed. Numerical experiments on 23 bound-constrained benchmark functions show that the proposed DELU is significantly better than, or at least comparable to several state-of-the art DE variants, non-DE algorithms, and surrogate-assisted evolutionary algorithms.

1. Introduction

Global optimization has become one of the most widely used techniques for modeling and analyzing practical problems [1]. Much progress in the fields of science, economy, and engineering relies heavily on numerical techniques to obtain global optimal solutions for optimization problems. However, an algorithm may get trapped into the local optimum for large-scale real-world optimization problems. In view of the complexity of the problem, traditional algorithms, such as the gradient-based algorithms, cannot be used to find the global optimum.

Evolutionary algorithms (EAs) are a broad class of stochastic optimization algorithms inspired by the natural evolution of species. EAs have been successfully applied to solve numerous optimization problems in various fields. The EAs family includes differential evolution (DE) [2], genetic algorithm (GA) [3], particle swarm optimization (PSO) [4], evolution strategies (ES) [5], and evolutionary programming (EP) [6]. The properties such as derivative-free and strong robustness make these algorithms attractive to applications of various real-world optimization problems.

DE, which was proposed by Storn and Price [2], has been proven to be a simple yet effective stochastic global optimization algorithm in EAs. It is well-known for its simple structure, ease of use, robustness and speed. Owing to these advantages, DE has been successfully applied in diverse fields [7,8], such as power systems [9], communication [10], chemical engineering [11], optics [12], and bioinformatics [13]. However, despite having several striking advantages and successful applications in diverse fields, DE has been shown to have certain weaknesses. A large number of function evaluations are needed to find optimal solutions, which leads to an increase in computational time, particularly for expensive-to-evaluate optimization problems in real applications. For example, for the gasifier problem in [14], function evaluations account for more than 99% of the entire searching time. Greedy selection strategy accelerates the convergence of the algorithm but makes the algorithm easy to get trapped into local optimum. In addition, DE is fast at exploring but slow at exploiting the solution [15].

Surrogate model algorithms [16–18] have been recently developed to reduce the necessary number of function evaluations while searching the global optimum of the problem. In surrogate model algorithms, a surrogate model is used to replace computationally expensive real function evaluations. A surrogate model usually be represented as $f(x) = s(x) + \epsilon$, where $f(x)$ is the real function value of the point x, $s(x)$ is the value obtained by the surrogate model, and ϵ is the error between them. The main

* Corresponding author.
E-mail address: zgj@zjut.edu.cn (G.-j. Zhang).

http://dx.doi.org/10.1016/j.cor.2016.05.015
0305-0548/

advantage of this approach is that the surrogate model has a lower computational complexity than the original model of the problem, thereby significantly reducing the computational cost. Because of this advantage, surrogate models are widely utilized in many fields, such as the optimization of helicopter rotor blades [19], the optimization design of corrugated beam guardrails [20], and the inverse calculation of in situ stress in rock mass [21]. However, selecting the appropriate surrogate model is a challenge because a certain surrogate model cannot be suitable for all kinds of problems [22].

Abstract convexity theory [23,24] generalizes the property of convex analysis that every convex function is the upper envelop of its affine minorants [25]. Therefore, many nonconvex functions (so-called abstract convex functions) can be represented as lower envelopes of some basic simple functions. Based on this property, Beliakov [26] proposed a cutting angle method (CAM). In CAM, the simple functions are replaced by support functions. By using a sequence of support functions, a lower approximation of the original problem can be obtained from below. Since the minimization of the lower approximation is much simpler than the original problem, it is called a relaxed model. Consequently, the global minimum of the objective problem can be obtained by enumerating all local minima of the relaxed model efficiently. Such underestimate technique is very useful in various applications [27–29]. However, to obtain a promising solution through a more accurate lower approximation, the method needs to use numerous support functions, which lead to the extremely high complexity. Therefore, CAM is efficient only for the problem with less than 10 variables [30–32].

In this paper, an effective DE algorithm based on the underestimate technique presented in CAM is proposed. The proposed algorithm, called DELU, uses local abstract convex underestimate strategy to reduce the number of function evaluations and to balance the exploration and exploitation of DE. Unlike the underestimate technique used in CAM, the local underestimate strategy only constructs the supporting hyperplanes for the neighboring individuals of the trial individual, and the neighboring individuals are selected by the Euclidean distance from the trial individual to the other individuals. Then, in the selection operation, the underestimate value of the trial individual is calculated by the supporting hyperplanes of the individuals near the trial. According to the underestimate value, we can judge whether the trial individual is worth evaluating, thus reducing the number of function evaluations. Specifically, the underestimate information also can be used to safely exclude some invalid regions of the domain where the global optimum cannot be found. This procedure prevents the algorithm from getting trapped in the local minimum in some case and improves exploration efficiency. In addition, exploitation capability is enhanced by employing the descent directions of supporting hyperplanes for local enhancement. Competitive experimental results are observed with respect to commonly used benchmark functions. Compared with four state-of-the-art DE variants, three non-DE algorithms and three surrogate-assisted evolutionary algorithms, our approach performs better, or at least comparably, in terms of the quality of the final solutions and convergence speed. In addition, the proposed local abstract convex underestimate strategy is also integrated into some advanced DE variants to verify the effect on them. Experimental results show that our proposed local abstract convex underestimate strategy is able to enhance the advanced DE algorithms.

The reminder of this paper is organized as follows. In Section 2, some related works of DE are presented. Section 3 briefly describes the DE algorithm and the cutting angle method. Section 4 describes the proposed algorithm at a finer level of detail. Experimental results demonstrating the performance of the proposed algorithm in comparison with five state-of-the art DE variants, four non-DE algorithms, and three surrogate-assisted EAs over a suite of bound-constrained numerical optimization functions are presented in Section 5. Section 6 concludes this paper.

2. Related works

DE has drawn the attention of many researchers all over the world. They have proposed many variants to reduce the computational cost, balance the exploration and exploitation, and improve the optimization capability of DE. In this section, a brief overview of these enhanced approaches is presented.

Some work mainly focuses on the function approximation technique to reduce the computational cost. Queipo [33] proposed a meta-model (or surrogate-model) as the approximation of the original function to replace calls to the expensive function evaluations. Liu [16] proposed a Gaussian process surrogate model assisted evolutionary algorithm, which map the training data to a lower dimensional space by employing a dimension reduction technique, and a new surrogate model-aware search mechanism is used to make the search focus on the promising subregion. Zhou [17] proposed a memetic algorithm using multi-surrogates. It combined the regression and exact interpolating surrogate model for local search to solve the expensive-to-evaluate problems. Liu [34] proposed a fast differential evolution using k-nearest neighbor predictor as function approximation. Jin [35] combined a multi-layer perceptron (MLP), a kind of neural network (NN), with covariance matrix adaptation evolution strategy (CMA-ES) to build an efficient evolutionary optimization with function approximation. Zhang [36] developed a predictive distribution model combining Gaussian stochastic model with fuzzy clustering based model to measure the expected improvement of each individual. Park [37] also proposed an efficient differential evolution using k-nearest neighbor as function estimator to alleviate a burden of a large number of function evaluations.

Many attempts have also been made to balance the exploration and exploitation. Wang [38] proposed a novel hybrid discrete differential evolution algorithm (HDDE), in which a local search algorithm based on insert neighborhood structure is embedded to balance the exploration and exploitation by enhancing the local searching ability. Bhattacharya [9] proposed a hybrid differential evolution with biogeography-based optimization (DE/BBO), which combines the exploration of DE with the exploitation of BBO effectively. Cai [39] integrated the one-step k-means clustering into DE (CDE), which makes the original DE more effective and efficient. Piotrowski [40] proposed a differential evolution with separated groups (DE-SG), which distributes population into small groups, defines rules of exchange of information and individuals between the groups and uses two different strategies to keep balance between exploration and exploitation capabilities. Li [41] proposed a modified differential evolution with self-adaptive parameters method (MDE), in which a probability rule is used to combine two different mutation rules to enhance the diversity of the population and the convergence rate of the algorithm.

Apart from the above methods, some researches consider to improve the optimization capability by the adaptive control parameters and strategies. Qin [42] proposed a self-adaptive differential evolution algorithm (SaDE), in which both trial vector generation strategies and their associated control parameter values are gradually self-adapted by learning from their previous experiences in generating promising solutions. Zhang [43] proposed an adaptive differential evolution with optional external archive (JADE) which improved optimization performance by implementing a new mutation strategy with optional external archive and automatically updated the parameters by evolving the