Chemometrics and Intelligent Laboratory Systems 151 (2016) 164–171

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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

Self-adaptive differential evolution algorithm with crossover strategies adaptation and its application in parameter estimation

Qinqin Fan [a,*], Yilian Zhang [b]

[a] Logistics Research Center, Shanghai Maritime University, Shanghai 201306, China
[b] Key Laboratory of Marine Technology and Control Engineering Ministry of Communications, Shanghai Maritime University, Shanghai 201306, China

ARTICLE INFO

Article history:
Received 12 May 2015
Received in revised form 19 December 2015
Accepted 29 December 2015
Available online 6 January 2016

Keywords:
Differential evolution
Crossover strategy adaption
Evolutionary computation
Self-adaptive
Parameter estimation

ABSTRACT

The performance of differential evolution (DE) is significantly influenced by the choice of crossover strategies; therefore, a self-adaptive differential evolution algorithm with crossover strategies adaptation (CSA-SADE) is proposed in this paper to enhance the performance of DE. In CSA-SADE, the suitable control parameters, mutation strategies, and crossover strategies can be achieved in different evolution stages. To demonstrate the effectiveness of CSA-SADE, the proposed algorithm is compared with eight state-of-the-art evolutionary algorithms. The simulation results indicate that CSA-SADE outperforms five improved DE algorithms and three non-DE approaches on a set of 25 CEC2005 benchmark functions. Additionally, the proposed algorithm is employed to estimate the kinetic parameters of mercury oxidation; the results show that CSA-SADE performs better than the compared algorithms in this simulation example.

1. Introduction

Differential evolution (DE) algorithm is a population-based and competitive stochastic search technique which was proposed by Storn and Price [1,2]. Over the past decades, DE has been widely used to tackle a wide range of benchmark test functions and real-world application problems [3,4]. However, the search performance of DE still needs to be improved since modern optimization problems are becoming increasingly complicated. DE performance is mainly dependent on three control parameters (i.e., mutation control parameter *F*, crossover control parameter *CR*, and population size *PS*), mutation strategies, and crossover strategies. It is difficult but significant to choose appropriate control parameters and strategies in the design of DE algorithm. To enhance the performance of DE, several improved DE algorithms have been proposed, such as a self-adaptive jDE [5], a self-adaptive DE (SaDE) algorithm [6], a composite DE (CoDE) algorithm [7], a differential evolution with dynamic parameters selection (DE-DPS) [8], an ensemble of mutation strategies and control parameters with DE (EPSDE) algorithm [9], a modified differential evolution with *p*-best crossover (MDE_*p*BX) [10], a differential evolution algorithm with self-adaptive strategy and control parameters (SSCPDE) [11]. However, to the best of our knowledge, only a few DE variants reported in the literatures adapt crossover strategies.

Generally, the exploration ability of the binomial crossover operation is better than that of the exponential crossover operation [12]. However, the exponential crossover operation is useful for solving non-linear functions [13]. Therefore, to further improve the performance of DE, a self-adaptive differential evolution with crossover strategies adaptation (CSA-SADE) is proposed in the current study. In the CSA-SADE algorithm, control parameters and mutation strategies adaptation methods are based on the work in Fan and Yan [11], and suitable crossover strategy can be automatically adjusted through a self-adaptive approach during the entire search process. To judge the performance of the proposed algorithm, 25 30- and 50-dimensional benchmark test functions, which are adopted from IEEE CEC2005 test sets [14], are employed. The experimental results show that the overall performance of the proposed algorithm is better than that of the compared algorithms.

The remainder of this paper is organized as follows. Section 2 introduces the DE algorithm. The related works of DE algorithm is reviewed in Section 3. Section 4 introduces the proposed CSA-SADE algorithm. The experimental results and crossover strategy adaptation analysis are given in Section 5. The application of parameter estimation for homogeneous mercury oxidation is presented in Section 6. Finally, Section 7 concludes the work of this paper.

2. Differential evolution algorithm

DE is a simple but yet a competitive evolutionary algorithm. It contains three main operations, i.e., mutation, crossover, and selection.

* Corresponding author at: NO. 1550 Haigang Avenue, Shanghai 201306, China. Tel.: +86 21 38284016.
E-mail address: forever123fan@163.com (Q. Fan).

http://dx.doi.org/10.1016/j.chemolab.2015.12.020
0169-7439/

The vector which contains D optimized variables $x_1, x_2, \cdots, x_D$ is denoted by $\boldsymbol{x}$. $\boldsymbol{x}_i^G = [x_{i,1}^G, x_{i,2}^G, \cdots x_{i,D}^G]$ denotes the ith solution (or individual) in the Gth generation. The population of the Gth generation is denoted by $\boldsymbol{X}^G = [\boldsymbol{x}_1^G, \boldsymbol{x}_2^G, \cdots, \boldsymbol{x}_{PS}^G]$, which contains PS individuals. The minimization problem is expressed as follows:

$$f(\boldsymbol{x}^*) = \min_{\boldsymbol{x}_i \in \Omega} f(\boldsymbol{x}_i), \tag{1}$$

where f denotes the optimization function; $\boldsymbol{x}_i$ is a D-dimensional vector; $\boldsymbol{x}^*$ is the global optimum solution; $\Omega \subseteq R^D$. For the bound-constrained optimization problem, it has the boundary constraints, i.e., $\boldsymbol{x} \in \Omega, x_j \in (x_j^{\text{low}}, x_j^{\text{high}})$, $j = 1, 2, \cdots, D$, where x_j^{low} and x_j^{high} are the lower and upper bounds of the jth variable of the individual, respectively. A D-dimensional space $\boldsymbol{P}_0$ is defined within the region $\{(x_j^{\text{low}}, x_j^{\text{high}}) j = 1, 2, \cdots, D\}$.

The implementation steps of DE can be described as follows:

1) Initialization operation.

Determine the mutation control parameter F, crossover control parameter CR, population size PS, and maximum number of generations $G_{\max}$. Set the current generation $G = 0$. The initial individuals $\boldsymbol{x}_i^0$, $i = 1, 2, \cdots, PS$ is generated randomly in $\boldsymbol{P}_0$.

2) Mutation operation.

For each $\boldsymbol{x}_i^G$ in the parent population, the mutant individual $\boldsymbol{v}_i^{G+1}$ is generated by mutation strategy. Some useful and famous mutation strategies are listed as follows:

$$\text{“}DE/rand/1\text{”} : \boldsymbol{v}_i^{G+1} = \boldsymbol{x}_{r_1}^G + F \cdot \left(\boldsymbol{x}_{r_2}^G - \boldsymbol{x}_{r_3}^G\right), \tag{2}$$

$$\text{“}DE/rand/2\text{”} : \boldsymbol{v}_i^{G+1} = \boldsymbol{x}_{r_1}^G + F \cdot \left(\boldsymbol{x}_{r_2}^G - \boldsymbol{x}_{r_3}^G\right) + F \cdot \left(\boldsymbol{x}_{r_4}^G - \boldsymbol{x}_{r_5}^G\right), \tag{3}$$

$$\text{“}DE/current-to-best/1\text{”} : \boldsymbol{v}_i^{G+1} = \boldsymbol{x}_i^G + F \cdot \left(\boldsymbol{x}_{\text{best}}^G - \boldsymbol{x}_i^G\right) + F \cdot \left(\boldsymbol{x}_{r_1}^G - \boldsymbol{x}_{r_2}^G\right), \tag{4}$$

$$\text{“}DE/current-to-best/2\text{”} : \boldsymbol{v}_i^{G+1} = \boldsymbol{x}_i^G + F \cdot \left(\boldsymbol{x}_{\text{best}}^G - \boldsymbol{x}_i^G\right) + F \cdot \left(\boldsymbol{x}_{r_1}^G - \boldsymbol{x}_{r_2}^G + \boldsymbol{x}_{r_3}^G - \boldsymbol{x}_{r_4}^G\right), \tag{5}$$

$$\text{“}DE/rand-to-best/1\text{”} : \boldsymbol{v}_i^{G+1} = \boldsymbol{x}_{r_1}^G + F \cdot \left(\boldsymbol{x}_{\text{best}}^G - \boldsymbol{x}_i^G\right) + F \cdot \left(\boldsymbol{x}_{r_2}^G - \boldsymbol{x}_{r_3}^G\right), \tag{6}$$

where r_1, r_2, r_3, r_4, and r_5 are randomly chosen within the range $[1, NP]$ and are also different from the index i (i.e. $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$); $\boldsymbol{x}_{\text{best}}^G$ is the individual vector with the best fitness value in the population at generation G.

3) Crossover operation.

In DE, the binomial crossover and exponential crossover are two common and useful strategies. For each individual $\boldsymbol{x}_i^G$, a trial vector $\boldsymbol{u}_i^{G+1}$ produced by the binomial crossover operation can be described as follows:

$$u_{ij}^{G+1} = \begin{cases} v_{ij}^{G+1}, & R_j \le CR \text{ or } j = j_{rand} \\ x_{ij}^G, & \text{otherwise} \end{cases} j = 1, 2, \cdots, D. \tag{7}$$

where R_j is a uniform random number in the range [0, 1], and j_{rand} is a randomly chosen integer within the range $[1, D]$.

The exponential crossover operation can be defined as follows:

$$u_{ij}^{G+1} = \begin{cases} v_{ij}^{G+1}, & \text{if } j = \langle n \rangle_D, \langle n+1 \rangle_D, \cdots, \langle n+L-1 \rangle_D \\ x_{ij}^G, & \text{otherwise} \end{cases} \tag{8}$$

where $\langle\rangle_D$ denotes a modulo function with modulus D. n is a random number and L is an integer drawn from $[1, D]$.

4) Selection operation.

The offspring $\boldsymbol{u}_i^{G+1}$ competes one-to-one with its parent $\boldsymbol{x}_i^G$. The evaluation operation is expressed as follows:

$$\boldsymbol{x}_i^{G+1} = \begin{cases} \boldsymbol{u}_i^{G+1}, & f(\boldsymbol{u}_i^{G+1}) \le f(\boldsymbol{x}_i^G) \\ \boldsymbol{x}_i^G, & \text{otherwise} \end{cases} \tag{9}$$

5) $G = G + 1$.

6) Repeat steps 2–5 until the number of generations is equal to $G_{\max}$.

3. Related works

The performance of DE is directly affected by the choices of the control parameters (i.e., F, CR, and PS) and strategies (i.e., mutation and crossover strategies). When the properties of the optimization problems are unknown, it is difficult to choose appropriate control parameters and strategies for DE algorithm. In the previous studies, DE researchers [2,15–17] introduced several empirical guidelines for selecting the control parameters and strategies to enhance the performance of DE. However, these guidelines are usually lack of sufficient justifications because they are based on specific experiments [6]. Meanwhile, constant parameter settings and strategy cannot adapt to different evolution stages and optimization problems.

To avoid the need for problem-dependent parameter tuning and strategy selection, researchers have proposed various self-adaptive DE algorithms. For example, Liu and Lampinen [18] introduced a fuzzy adaptive differential evolution (FADE) algorithm, in which appropriate control parameters were produced by using fuzzy logic controllers. Zaharie [19] used the population diversity to adjust control parameters. Salman et al. [20] proposed a self-adaptive differential evolution (SDE) algorithm that uses the ring neighborhood topology and employs a normal distribution function to generate the control parameters (i.e., F and CR). Brest et al. [5] proposed a self-adaptive jDE wherein a self-adaptive control mechanism is used to tune the control parameters (i.e., F and CR) that are encoded into the individual. The experimental results indicate that jDE performs better than FADE and other compared algorithms on numerical benchmark problems. Qin et al. [6] introduced a self-adaptive DE (SaDE) that uses successful experience to adaptively control the evolution of mutation strategies and their associated crossover control parameter CR. Moreover, F is produced by a normal distribution function. Zhang and Sanderson [21] proposed a new DE (JADE) which uses a novel mutation strategy and adjust the control parameters (i.e., F and CR) in an adaptive mechanism. Mallipeddi et al. [9] proposed an ensemble of mutation strategies and control parameters with DE (EPSDE). In EPSDE, choices of trial generation strategies and control parameters are based on their success experience in the past generations. Wang et al. [7] introduced a composite DE (CoDE), in which three mutation strategies are randomly combined with three fix control parameter combinations. The results indicate that CoDE can balance between exploration and exploitation capabilities. Fan and Yan [11] proposed a differential evolution algorithm with self-adaptive strategy and control parameter (SSCPDE) wherein the control parameters and mutation strategies are automatically adjusted based on learning experience at the last generation. The more relevant studies can be seen Ref.