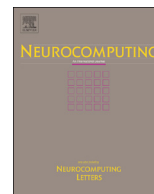




ELSEVIER

Contents lists available at ScienceDirect

## Neurocomputing

journal homepage: [www.elsevier.com/locate/neucom](http://www.elsevier.com/locate/neucom)

# A clustering-based differential evolution with random-based sampling and Gaussian sampling <sup>☆</sup>

Gang Liu <sup>a,\*</sup>, Zhaolu Guo <sup>b</sup><sup>a</sup> School of Computer Science, Hubei University of Technology, Wuhan 430072, China<sup>b</sup> School of Science, Jiangxi University of Science and Technology, Ganzhou 341000, China

## ARTICLE INFO

## Article history:

Received 7 November 2015

Received in revised form

26 January 2016

Accepted 17 April 2016

Communicated by A.M. Alimi

Available online 11 May 2016

## Keywords:

Differential evolution

K-means clustering

Random-based sampling

Gaussian sampling

## ABSTRACT

Differential Evolution (DE) has been widely researched because of its excellent performance and many differential evolution variants have been proposed. However, no variant was able to consistently perform over a wide range of test problems. This paper presents a novel algorithm based on the one-step  $k$ -means clustering, random-based sampling and Gaussian sampling to improve the performance of DE to solve optimization problems efficiently. The proposed enhanced DE utilizes the one-step  $k$ -means clustering to generate  $k$  search spaces. In these spaces, the new mutation operators based on random-based sampling and Gaussian sampling are used to exploit. The resulting algorithms are named as clustering-based differential evolution with random-based sampling and Gaussian sampling (GRCDE). Experimental verifications are conducted on 25 benchmark functions and the CEC'05 competition, including detailed analysis for GRCDE. The results clearly show that GRCDE outperforms other state-of-the-art evolutionary algorithms in terms of the solution accuracy and the convergence rate.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

Global optimization is an attractive research area in the computer science and there are a considerable number of real-world decision processes that require the solving of global optimization problems. In fact, many engineering problems can be transformed into global optimization problems [1–4]. The purpose of global optimization is to determine the solutions for all of the decision variables by optimizing the objective function. Formally, global optimization is stated as follows:

$$\min f(X)$$

$$L_j \leq x_j \leq U_j$$

$$j = 1, 2, \dots, D \quad (1)$$

where  $X$  is a  $D$ -dimensional vector,  $f(X)$  is the objective function and where each  $x_j$  has a lower limit  $L_j$  and an upper limit  $U_j$ . The function  $f(X)$  does not need to be continuous but it must be bounded. In global optimization problems, to find the global optimal solution is particularly challenging when the dimension is

high and there are numerous local optima. Currently, there are many research results for global optimization problems [5–7].

Evolutionary algorithms (EAs) [8] have a long history of successfully solving global optimization problems. EAs contain a wide range of algorithms that have been introduced to solve complex optimization problems, such as the genetic algorithm (GA) [9], particle swarm optimization (PSO) [10], differential evolution (DE) [11–13], etc. In this paper, DE is considered for solving global optimization problems.

Differential evolution is known as an efficient EA. The mutation operator of DE is main search operator and moves the population towards the global optimum. Currently, there are 10 different schemes of DE [12]. But there is no known mutation operator that will consistently perform for all classes of optimization problems. This motivates the researchers to consider new mutation operators in DE for a better coverage of problems. In recent years, a number of improved DE variants have been proposed [14–18]. All of these algorithms are based on the design of mutation operators and these strategies are helpful to improve the performance of DE. To further improve the adaptability of the algorithm, adaptive or self-adaptive mechanisms that control the parameters of DE is used with the design of mutation operators. These DE variants have been proposed [14,19–21].

Generally, DE is good at exploring the search space and locating the region of global minimum but is poor at exploitation of the solutions. In the evolutionary process, the stronger exploitation ability of DE is, the faster convergence speed of reaching the optimal solution should be. Hence, enhancing the exploitation ability is

<sup>☆</sup>The work described in this paper was supported by National Natural Science Foundation of China Foundation No. 61300127. Any conclusions or recommendations stated here are those of the authors and do not necessarily reflect official positions of NSFC.

\* Corresponding author.

E-mail addresses: [lg0061408@126.com](mailto:lg0061408@126.com) (G. Liu), [gzl@whu.edu.cn](mailto:gzl@whu.edu.cn) (Z. Guo).

helpful to improve the overall performance of DE. According to the population distribution, the search space is divided into multiple regions and each region is exploited, respectively. This method can effectively improve the exploitation ability of DE.

Based on these considerations, this paper proposes a novel clustering-based differential evolution with random-based sampling and Gaussian sampling (GRUDE). Random-based sampling is motivated by center-based sampling [22]. Compared with center-based sampling, the search ability of random-based sampling is stronger and more flexible because of the randomness. The basic idea of the proposed algorithm is based on the following consideration. In GRUDE, the population is partitioned into  $k$  subsets by the one-step  $k$ -means clustering, and each subset is considered as a promising area for exploitation. Random-based sampling and Gaussian sampling are used to design the new mutation strategies to search in these promising areas. GRUDE employs two different mutation schemes, the mutation scheme based on random-based sampling and Gaussian mutation scheme. These mutation schemes search in different ways to improve the search efficiency. The best vector created by two mutation schemes is select into the next generation population. The advantages of the proposed technique are as follows: (1) the mutation operators are able to enhance the exploitation ability of DE; (2) our approach is still very simple; (3) our approach does not increase the overall complexity of DE. In order to verify the performance of the proposed approaches, two comprehensive sets of experiments are conducted on 25 well-known numerical benchmark functions and the CEC'05 competition. Compared with other state-of-the-art EAs, our approach performs better, or at least comparably, in terms of the quality of the final solutions and the convergence speed.

The remainder of this paper is organized as follows. Section 2 introduces the traditional DE algorithm and gives a short literature review. Section 3 briefly describes the  $k$ -means clustering used in this work. Section 4 introduces center-based sampling and random-based sampling. The proposed approach is presented in detail in Section 5. Experimental results and discussions are reported in Section 6. Finally, some conclusions and possible paths for future research are provided in Section 7.

## 2. Differential evolution

Differential evolution has become one of the most frequently used evolutionary algorithms for solving the continuous global optimization problems in recent years [23–26]. Just like other EAs, DE is also a population-based stochastic search and contains four main steps, initialization, mutation, crossover and selection. It starts with a population of  $NP$  vectors representing the candidate solutions, where  $NP$  indicates the population size. It assumes that  $X_{i,G} = \{x_{i,1,G}, x_{i,2,G}, \dots, x_{i,D,G}\}$  is the  $i$ th candidate solution vector in generation  $G$ , where  $i = 1, 2, \dots, NP$ ,  $D$  is the dimension of the problem, and  $G$  is the generation index. The differential evolution algorithm is written in Algorithm 1.

**Algorithm 1.** DE algorithm with DE/rand/1/bin: model1.

- 1: Generate the initial population and population scale is  $NP$ , set scaling factor  $F$ , crossover probability  $CR$ ,  $D$  is the number of decision variables, give the maximum number of iterations,  $MaxGens$ , set counter  $G=1$ ;
- 2: Evaluate the fitness for each individual in the population;
- 3: **while**  $G < MaxGens$  **do**
- 4:   **for**  $i=1$  to  $NP$  **do**
- 5:     Chose three vectors  $X_{r1,G}, X_{r2,G}, X_{r3,G}$  randomly from the current population, where  $r1 \neq r2 \neq r3 \neq i$ ;

- 6:     Generate a new trial vector  $V = \{v_{i,1}, v_{i,2}, \dots, v_{i,D}\}$  according to

$$V = X_{r1,G} + F*(X_{r2,G} - X_{r3,G})$$

- 7:     Generate a new vector  $U = \{u_{i,1}, u_{i,2}, \dots, u_{i,D}\}$ , according to the preset vector  $X_{i,G}$  and the vector  $V$  generated in Step 6 as follows:

$$u_{ij} = \begin{cases} v_{ij} & \text{if } (rand(0, 1) \leq CR \text{ or } j = randint[1, n]) \\ x_{ij} & \text{if } (rand(0, 1) > CR \text{ or } j \neq randint[1, n]) \end{cases} \quad j = 1, 2, \dots, D$$

where  $rand(0, 1)$  is a uniform random number in range  $(0, 1)$ , and  $randint[1, n]$  is a randomly chosen index in the set  $\{1, 2, \dots, N_p\}$ , which insures the new vector to get at least one parameter from the new generated vector  $v$ ;

- 8:     Evaluate the offspring  $U$ ;
- 9:     **if**  $U$  is better than  $X_{i,G}$  **then**
- 10:        $X_{i,G+1} = U$
- 11:     **end if**
- 12:   **end for**
- 13:    $G = G + 1$ ;
- 14: **end while**

For the terminal conditions, one can either fix the maximum number of fitness evaluations (Max\_NFEs) or the precision of a desired solution value to reach (VTR). The efficiency of differential evolution is very sensitive to the setting of control parameters. In general conditions, the control parameters depend on the results of preliminary tuning. The mutation scheme achieves to explore the search space. In the DE family, many mutation schemes have been proposed [12]. Some well-known mutation schemes are listed as follows:

$$DE/rand/1 : V_i = X_{r1} + F*(X_{r2} - X_{r3}) \quad (2)$$

$$DE/best/1 : V_i = X_{best} + F*(X_{r1} - X_{r2}) \quad (3)$$

$$DE/rand - to - best/1 : V_i = X_i + \lambda*(X_{best} - X_i) + F*(X_{r1} - X_{r2}) \quad (4)$$

where  $r1, r2, r3 \in \{1, 2, \dots, NP\}$  are randomly chosen integers, which are different from each other and also different from the running index  $i$ .  $X_{best}$  represents the best individual in the current generation.  $F (> 0)$  is a scaling factor which controls the amplification of the differential vector.  $\lambda$  is an additional control parameter.

The aim of crossover is recombination of the solutions from previous generations. Crossover is employed to build trial vectors by recombining the current vector and the mutant one. After the crossover, a greedy selection mechanism is used to select the better one between the parent vector  $X_{i,G}$  and the trial vector  $U$  according to their fitness values. DE's performance highly depends on the selected mutation strategies and corresponding control parameters.

Due to the performance of DE highly depends on appropriately choosing trial vector generation strategies and their associated control parameter values, many researchers have proposed many methods to improve the selection, mutation, crossover strategies of DE and investigating optimum choice of DE control parameters.

The design of new evolution operation or improve original evolution operation is an important research direction. Cai [27] proposed a hybrid DE based on the one-step  $k$ -means clustering (CDE). In CDE, the one-step  $k$ -means clustering is used to generate

Download English Version:

<https://daneshyari.com/en/article/405696>

Download Persian Version:

<https://daneshyari.com/article/405696>

[Daneshyari.com](https://daneshyari.com)