Expert Systems With Applications 44 (2016) 1–12

Contents lists available at ScienceDirect

Expert Systems With Applications

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

An improved adaptive differential evolution algorithm for continuous optimization

Wenchao Yi, Yinzhi Zhou[1], Liang Gao, Xinyu Li*, Jianhui Mou

State Key Laboratory of Digital Manufacturing Equipment & Technology, Huazhong University of Science and Technology, Wuhan 430074, PR China

ARTICLE INFO

Keywords:
Differential evolution
*p*best retention mechanism
Adaptive parameter control
*p*best roulette wheel selection
Real-world application problems

ABSTRACT

A novel differential evolution algorithm based on adaptive differential evolution algorithm is proposed by implementing *p*best roulette wheel selection and retention mechanism. Motivated by the observation that individuals with better function values can generate better offspring, we propose a fitness function value based *p*best selection mechanism. The generated offspring with better fitness function value indicates that the *p*best vector of current individual is suitable for exploitation, so the *p*best vector should be retained into the next generation. This modification is used to avoid the individual gather around the *p*best vector, thus diversify the population. The performance of the proposed algorithm is extensively evaluated both on the 25 famous benchmark functions and four real-world application problems. Experimental results and statistical analyses show that the proposed algorithm is highly competitive when compared with other state-of-the-art differential evolution algorithms.

1. Introduction

Differential evolution (DE) algorithm, proposed by Storn and Price (1997), has been proven to be simple yet effective evolutionary algorithm (EA) (Bäck, 1996). What's more, DE algorithm presents competitive performance in diverse fields. DE algorithm has been applied successfully in fields including constrained optimization problems (Becerra & Coello, 2006; Huang, Wang, & He, 2007), multi-objective optimization problems (Gong & Cai, 2009; Tan, Jiao, Li, & Wang, 2012) and engineering design optimization problems (Liao, 2010; Yildiz, 2013). However, recent research works showed that the performance of DE is related to mutation operators (Das & Abaham, 2009; Epitropakis, Tasoulis, Pavlidis, Plagianakos, & Vrahatis, 2011; Fan & Lampinen, 2003; Piotrowski, 2013; Zhou, Li, & Gao, 2013). Mutation operator manipulates the balance between exploitation and exploration. We will mainly focus on introducing some representative works on the improvement of the mutation operators and a series of improvements on the efficient DE variant called adaptive differential evolution algorithm with optional external archive (JADE) in the following paragraph.

Some researchers have made some contributions to the innovation of the mutation operator adopted in DE algorithm. Fan and Lampinen (2003) proposed the trigonometric region based mutation operator, in which a trigonometric region was formed to limit the generated mutant vectors. Das and Abraham (2009) introduced the hybrid DE mutation operator that was a linear combination of two mutation operators to balance exploration and exploitation ability. A static ring topology was used to select the neighborhood best individual to enhance local search. Based on this, Piotrowski (2013) introduced an improved version of neighborhood-based mutation operator by splitting the mutation directly into global and local ones. Epitropakis et al. (2011) presented the proximity-based mutation operator. By incorporating the information about neighboring individuals, more specifically, by giving each individual selection probability that is inversely related to its distance from the base vector. An insect mutation operator proposed by Zhou et al. (2013) divided the individuals into better part and worse part. Based on the division, the novel mutation operator lets the better part to exploit, the worse part to explore. Most researchers used "DE/rand/1", "DE/current-to-best/1" and "DE/best/1" or their modified versions. "DE/rand/1" emphasizes on the exploration, while the last two are rarely used mainly due to their premature convergence to multimodal problems. However, "DE/current-to-best/1" and "DE/best/1" can obtain good performances when dealing with the unimodal problems. So, it is a challenge to combine their merits together.

Some researchers achieved excellent improvements in DE algorithm, in which a DE variant called JADE presented by Zhang and Sanderson (2009a) attracted wide attentions in recent years for its promising results. A novel adaptive control parameter and a new mutation operator called "DE/current-to-*p*best" were proposed in JADE. The control parameters of each individual are updated according to

* Corresponding author. Tel.: +86 27 87557742; fax: +86 27 87543074.
E-mail addresses: janety1989@gmail.com (W. Yi), leitezhou@gmail.com (Y. Zhou), gaoliang@mail.hust.edu.cn (L. Gao), lixinyu@mail.hust.edu.cn (X. Li), mjhcr@163.com (J. Mou).

[1] Co-first author.

http://dx.doi.org/10.1016/j.eswa.2015.09.031

the successful information of F and CR for each individual in the last generation. The top 100*p% (p is a predefined number between 0 and 1) individuals in the current generation are stored and then randomly choose any of the top 100*p% individual to replace role that the best individual plays in "DE/current-to best/1". JADE is a highly competitive DE variant and some improvements based on the framework of JADE are studied. Gong, Cai, Ling, and Li (2011) introduced the strategy adaptive mechanism. The mechanism chooses a mutation operator at random if the generated random number is smaller than a predefined parameter or uses the operator in the previous generation, where a family of different mutation operators used in JADE is chosen to formulate the pool. Islam, Das, Ghosh, Roy, and Suganthan (2012) proposed an improved JADE, in which *p*best crossover operator was proposed and modified adaptation schemes were utilized. Based on JADE, a repairing crossover rate technique based JADE was proposed by Gong, Cai, and Wang (2014) The crossover rate is repaired by calculating its corresponding binary string, which can be calculated and updated according to binary string for the each individual in the last generation. This algorithm is one of the best DE algorithms that obtain the best performances over CEC2005 competition benchmark functions. From this series of research works, we found that JADE is an efficient DE variant for global optimization problems. However, as emphasized above, mutation operator has a great impact on the DE algorithm's performance. Although "DE/current-to-*p*best" can help the offspring gather around the better individuals, it will decrease the diversity to some extent. So there still exists some space for further improvements.

Based on the above analysis, we present a modified algorithm based on JADE. In JADE, one top vectors are chosen for mutation operation for each individual randomly. However, this process does not consider the function value of these alternative vectors. As Rank-JADE algorithm shows, the terminal vector of the difference vectors plays an important role in the DE variants' performance. DE algorithm has shown excellent performance in exploring the feasible region, the motivation of our research is to enhance its exploitation ability by making the better top vectors to be chosen to survive into the next generation. The better here has two-fold meanings, one of which means the top vector with better fitness function value, the other indicates the top vector that can produce better offspring. Based on the above motivation, a *p*best roulette wheel selection operation is executed according to the function value of top vectors. If this top vector can achieve better trial vector, it can survive into the next generation, even it may not be included in the top vectors anymore. Then we combined the improved mutation operators with the repairing crossover rate mechanism. We call this algorithm as *p*best roulette wheel selection and retention mechanism based repairing crossover rate in adaptive DE algorithm ($p\text{best}_{rr}$-JADE).

The remainder of the paper is organized as follows. Section 2 introduces the classical DE algorithm and reviews the related works on one of competitive DE variants called JADE. Then the proposed algorithm is presented in Section 3. Experimental results on CEC2005 competition benchmark functions and real-world application problems are reported in Section 4. Finally, Section 5 concludes the paper. To give a vivid description of the whole paper, the graphical abstract is presented in the Appendix.

2. Related works of JADE

In this section, we will present the classical DE as a foundation. Then one of the most competitive DE variants called JADE will be introduced. In the following section, a recently proposed R_{cr}-JADE will be presented.

2.1. Classical DE algorithm

In this section, an introduction of the classical DE algorithm [1] will be presented, which facilitates the explanation of the improved DE algorithm later.

DE that is an effective evolutionary algorithm utilizes NP D-dimensional individuals, i.e. $x_{i,G} = \{x^1_{i,G}, \ldots, x^D_{i,G}\},\ i = 1, \ldots, NP$, where G denotes the number of generations. Each dimension of the individual is constrained by $x_{\min} = \{x^1_{\min}, \ldots, x^D_{\min}\}$ and $x_{\max} = \{x^1_{\max}, \ldots, x^D_{\max}\}$. Usually, the initial population is randomly generated in the feasible region, which can be expressed as follows:

$$x^j_{i,0} = x^j_{\min} + rand(0,1) * (x^j_{\max} - x^j_{\min}) \quad (1)$$

where *rand*(0,1) denotes a random number that is uniformly distributed and generated in the range of 0 and 1.

2.1.1. Mutation

Then the mutation operator is utilized to generate the mutant vectors $v_{i,G}$, DE/rand/1 is the most commonly used operator, where the generated $v_{i,G}$ can be represented as:

$$v_{i,G} = x_{r1,G} + F * (x_{r2,G} - x_{r3,G}),\ r1 \neq r2 \neq r3 \neq i \quad (2)$$

where $x_{r1,G}$, $x_{r2,G}$, $x_{r3,G}$ that are chosen from the current population and individual i are four mutually different individuals. F is the mutation control parameter to scale the difference vector. Similarly, we give another five frequently used mutation operators as follows:

(1) "DE/rand/2"

$$v_{i,G} = x_{r1,G} + F * (x_{r2,G} - x_{r3,G}) + F * (x_{r4,G} - x_{r5,G}),\quad r1 \neq r2 \neq r3 \neq r4 \neq r5 \neq i \quad (3)$$

(2) "DE/best/1"

$$v_{i,G} = x_{best,G} + F * (x_{r1,G} - x_{r2,G}),\quad r1 \neq r2 \neq i \quad (4)$$

(3) "DE/best/2"

$$v_{i,G} = x_{best,G} + F * (x_{r1,G} - x_{r2,G}) + F * (x_{r3,G} - x_{r4,G}),\quad r1 \neq r2 \neq r3 \neq r4 \neq i \quad (5)$$

(4) "DE/rand-to-best/1"

$$v_{i,G} = x_{i,G} + F * (x_{best,G} - x_{i,G}) + F * (x_{r1,G} - x_{r2,G}),\quad r1 \neq r2 \neq i \quad (6)$$

(5) "DE/current-to-best/1"

$$v_{i,G} = x_{i,G} + F * (x_{r1,G} - x_{i,G}) + F * (x_{r2,G} - x_{r3,G}),\quad r1 \neq r2 \neq r3 \neq i \quad (7)$$

where $x_{best,\,G}$ defines the individual that has the best fitness function value at the G-th generation.

2.1.2. Crossover

Then the binomial crossover operator that is often used can be selected to generate the trail vector $u_{i,\,G}$ between $x_{i,\,G}$ and $v_{i,\,G}$, which can expressed by the formula below:

$$u^j_{i,G} = \begin{cases} v^j_{i,G}, & \text{if } rand_j \le CR_i \text{ or } j = n_j \\ x^j_{i,G}, & \text{otherwise} \end{cases} \quad i = 1, 2, \ldots, NP; \quad j = 1, 2, \ldots, D \quad (8)$$

where $rand_j$ is a random number that is uniformly distributed and generated within the range of [0, 1]. $CR_i \in (0, 1)$ is crossover control parameter and n_j is a random integer generated within the range [1, D].

2.1.3. Selection

Then a better individual between trail vector $u_{i,\,G}$ and target vector $x_{i,\,G}$ will be selected. The better one will survive into the next generation based on the comparison of the fitness value. The greedy selection is performed as shown below:

$$x_{i,G+1} = \begin{cases} u_{i,G}, & \text{if } f(u_{i,G}) \le f(x_{i,G}) \\ x_{i,G}, & \text{otherwise} \end{cases} \quad (9)$$

where $f(x_{i,\,G})$ and $f(u_{i,\,G})$ are the fitness function value of target vector $x_{i,\,G}$ and trail vector $u_{i,\,G}$.