Neurocomputing 149 (2015) 1038–1047

Contents lists available at ScienceDirect

Neurocomputing

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

A binary differential evolution algorithm learning from explored solutions

Yu Chen [a,b,*], Weicheng Xie [c], Xiufen Zou [b]

[a] School of Science, Wuhan University of Technology, Wuhan 430070, China
[b] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China
[c] Shenzhen Key Laboratory of Spatial Smart Sensing and Services, School of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518052, China

ARTICLE INFO

Article history:
Received 3 October 2013
Received in revised form
23 May 2014
Accepted 23 July 2014
Communicated by Swagatam Das
Available online 2 August 2014

Keywords:
Binary differential evolution algorithm
Convergence in probability
Renewal metric
Refinement metric

ABSTRACT

Although real-coded differential evolution (DE) algorithms can perform well on continuous optimization problems (CoOPs), designing an efficient binary-coded DE algorithm is still a challenging task. Inspired by the learning mechanism in particle swarm optimization (PSO) algorithms, we propose a binary learning differential evolution (BLDE) algorithm that can efficiently locate the global optimal solutions by learning from the last population. Then, we theoretically prove the global convergence of BLDE, and compare it with some existing binary-coded evolutionary algorithms (EAs) via numerical experiments. Numerical results show that BLDE is competitive with the compared EAs. Further study is performed via the change curves of a renewal metric and a refinement metric to investigate why BLDE cannot outperform some compared EAs for several selected benchmark problems. Finally, we employ BLDE in solving the unit commitment problem (UCP) in power systems to show its applicability to practical problems.

1. Introduction

1.1. Background

Differential evolution (DE) [26], a competitive evolutionary algorithm emerging more than a decade ago, has been widely utilized in the science and engineering fields [24,4]. The simple and straightforward evolving mechanisms of DE endow it with the powerful capability to solve continuous optimization problems (CoOPs), but hamper its applications to discrete optimization problems (DOPs).

To take full advantage of the superiority of mutations in classic DE algorithms, Pampará and Engelbrecht [21] introduced a trigonometric generating function to transform the real-coded individuals of DE into binary strings, and proposed an angle modulated differential evolution (AMDE) algorithm for DOPs. Compared with the binary differential evolution (BDE) algorithms that directly manipulate binary strings, AMDE was much slower, but outperformed BDE algorithms with respect to accuracy of the obtained solutions [7]. Meanwhile, Gong and Tuson proposed a binary DE algorithm by forma analysis [9], but it cannot perform well on binary constraint satisfaction problems due to its weak exploration ability [31]. Attempting to simulate the operation mode of the continuous DE mutation, Kashan et al. [14] designed a dissimilarity based differential evolution (DisDE) algorithm incorporating a measure of dissimilarity in mutation. Numerical results show that DisDE is competitive with some existing binary-coded evolutionary algorithms (EAs).

In addition, the performances of BDE algorithms can also be improved by incorporating recombination operators of other EAs. Hota and Pat [12] proposed an adaptive quantum-inspired differential evolution algorithm (AQDE) applying quantum computing techniques, while He and Han [10] introduced the negative selection in artificial immune systems to obtain an artificial immune system based differential evolution (AIS-DE) algorithm. With respect to the fact that the logical operations introduced in AIS-DE tends to produce "1" bits with increasing probability, Wu and Tseng [29] proposed a modified binary differential evolution strategy to improve the performance of BDE algorithms on topology optimization of structures.

1.2. Motivation and contribution

Existing research efforts tried to incorporate the recombination strategies of various EAs to obtain efficient BDEs for DOPs, however, there are still some points to be improved:

- AMDE [21] has to transform real values into binary strings, which leads to an explosion of the computational cost for

* Corresponding author at: School of Science, Wuhan University of Technology, Wuhan 430070, China. Tel.: +86 27 87108027.
E-mail address: chymath@gmail.com (Y. Chen).

http://dx.doi.org/10.1016/j.neucom.2014.07.030
0925-2312/

function evaluations. Meanwhile, the mathematical properties of the transformation function can also influence its performances on various DOPs;

- BDE algorithms directly manipulating bit-strings, such as binDE [9], AIS-DE [10] and MBDE [29]cannot effectively imitate the mutation mechanism of continuous DE algorithms. Thus, they cannot perform well on high-dimensional DOPs due to their weak exploration abilities;
- DisDE [14], which incorporates a dissimilarity metric in the mutation operator, has to solve a minimization problem during the mutation process. As a consequence, the computation complexity of DisDE is considerably high.

Generally, it is a challenging task to design an efficient BDE algorithm perfectly addressing the aforementioned points. Recently, variants of the particle swarm optimization (PSO) algorithm [15] have been successfully utilized in real applications [6,1,23,2,17]. Although DE algorithms perform better than PSO algorithms in some real world applications [28,25,22], it is still promising to improve DE by incorporating PSO in the evolutionary process [3,18,19]. Considering that the learning mechanism of PSO can accelerate the convergence of populations, we propose a hybrid binary-coded evolutionary algorithm learning from the last population, named as the binary learning differential evolution (BLDE) algorithm. In BLDE, the searching process of population is guided by the renewed information of individuals, the dissimilarity between individuals and the best explored solution in the population. Using these, BLDE can perform well on DOPs.

The remainder of the paper is structured as follows. Section 2 presents a description of BLDE, and its global convergence is theoretically proved in Section 3. Then, in Section 4, BLDE is compared with some existing algorithms using numerical results. To test the performance of BLDE on real-life problems, we employ it to solve the unit commitment problem (UCP) in Section 5. Finally, discussions and conclusions are presented in Section 6.

2. The binary learning differential evolution algorithm

2.1. Framework of the binary learning differential evolution algorithm

Algorithm 1. The binary learning differential evolution (BLDE) algorithm.

Randomly generate two populations $\mathbf{X}^{(1)}$ and $\mathbf{A}^{(1)}$ of μ individuals; Set $t:=1$;
whilethe stop criterion is no satisfied **do**
  Let $\mathbf{x}_{gb}=(x_{gb,1},\dots,x_{gb,n})\triangleq \arg\max_{\mathbf{x}\in\mathbf{X}^{(t)}}\{f(\mathbf{x})\}$;
  for all $\mathbf{w}\in\mathbf{X}^{(t)}$ **do**
    Randomly select $\mathbf{x}=(x_1,\dots,x_n)$ and $\mathbf{y}=(y_1,\dots,y_n)$ from $\mathbf{X}^{(t)}$, as well as $\mathbf{z}=(z_1,\dots,z_n)$ from $\mathbf{A}^{(t)}$;
    $\mathbf{tx}=(tx_1,\dots,tx_n)\triangleq \arg\max\{f(\mathbf{y}),f(\mathbf{z})\}$;
    for $j=1,2,\dots,n$ **do**
      if $y_j=z_j$ **then**
        if $x_{gb,j}\neq x_j$ **then**
          $tx_j=x_{gb,j}$;
        else
          if $rand(0,1)\le p$ **then**
            $$tx_j=\begin{cases}0 & \text{with probability } \frac{1}{2};\\ 1 & \text{otherwise.}\end{cases}$$
          end if
        end if
      end if
    end for
    if $f(\mathbf{tx})\ge f(\mathbf{w})$ **then**
      $\mathbf{w}=\mathbf{tx}$;
    end if
  end for
  $t:=t+1$;
  $\mathbf{A}^{(t)}=\mathbf{X}^{(t-1)}$;
end while

For a binary optimization problem (BOP)[1]

$$\max_{\mathbf{x}\in S}\; f(\mathbf{x})=f(x_1,\dots,x_n),\quad S\subset\{0,1\}^n, \tag{1}$$

the BLDE algorithm illustrated by Algorithm 1 possesses two collections of μ solutions, the population $\mathbf{X}^{(t)}$ and the archive $\mathbf{A}^{(t)}$. At the first generation, the population $\mathbf{X}^{(1)}$ and the archive $\mathbf{A}^{(1)}$ are generated randomly. Then, repeat the following operations until the stopping criterion is satisfied.

For each individual $\mathbf{w}\in\mathbf{X}^{(t)}$ a trial solution is generated by three randomly selected individuals $\mathbf{x}$, $\mathbf{y}\in\mathbf{X}^{(t)}$ and $\mathbf{z}\in\mathbf{A}^{(t)}$. At first, initialize the trial individual $\mathbf{tx}=\{tx_1,\dots,tx_n\}$ as the winner of two individuals $\mathbf{y}\in\mathbf{X}^{(t)}$ and $\mathbf{z}\in\mathbf{A}^{(t)}$. $\forall\, j\in\{1,2,\dots,n\}$, if $\mathbf{y}$ and $\mathbf{z}$ coincide on the jth bit, the jth bit of $\mathbf{tx}$ is changed as follows.

- If the jth bit of $\mathbf{x}$ differs from that of $\mathbf{x}_{gb}$, tx_j is set to be $x_{gb,j}$, the jth bit of $\mathbf{x}_{gb}$;
- otherwise, tx_j is randomly mutated with a preset probability p.

Then, replace $\mathbf{w}$ with $\mathbf{tx}$ if $f(\mathbf{tx})\ge f(\mathbf{w})$. After the update of population $\mathbf{X}^{(t)}$ is completed, set $t=t+1$ and $\mathbf{A}^{(t)}=\mathbf{X}^{(t-1)}$.

2.2. The positive functions of the learning scheme

Generally speaking, the trial solution $\mathbf{tx}$ is generated by three randomly selected individuals. Meanwhile, it also incorporates conditional learning strategies in the mutation process.

- By randomly selecting $\mathbf{y}\in\mathbf{X}^{(t)}$, BLDE can learn from any member in the present population. Because the elitism strategy is employed in the BLDE algorithm, BLDE could learn from any *pbest* solution in the population, unlike PSO, where particles can only learn from their own *pbest* individuals.
- By randomly selecting $\mathbf{z}\in\mathbf{A}^{(t)}$, BLDE can learn from any member in the last population. In the early stages of the iteration process, individuals in the population $\mathbf{X}^{(t)}$ are usually different with those in $\mathbf{A}^{(t)}=\mathbf{X}^{(t-1)}$. Combined with the first strategy, this scheme actually enhances the exploration ability of the population and to some extent, accelerates convergence of the population.
- When bits of $\mathbf{y}$ coincide with the corresponding bits of $\mathbf{z}$, trial solutions learn from the *gbest* on the condition that randomly selected $\mathbf{x}\in\mathbf{X}^{(t)}$ differs from $\mathbf{x}_{gb}$ on these bits. This scheme imitates the learning strategy of PSO. The scheme can also prevent the population from being governed by dominating patterns because the increased probability $\mathbf{P}\{x_{gb,j}=x_j\}$ will lead to a random mutation performed on $\mathbf{tx}$, thus preventing duplicates of the dominating patterns in the population.

In PSO algorithms, each particle learns from the *pbest* (the best solution it has obtained so far) and the *gbest* (the best solution the swarm has obtained so far), and particles in the swarm exchange

[1] When a CoOP is considered, the real-value variables can be coded as bit-strings, and consequently, a binary optimization problem is constructed to be solved by binary-coded evolutionary algorithms.