BioSystems 132–133 (2015) 43–53

Contents lists available at ScienceDirect

BioSystems

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

A self adaptive hybrid enhanced artificial bee colony algorithm for continuous optimization problems

Hai Shan, Toshiyuki Yasuda, Kazuhiro Ohkura *

Graduate School of Engineering, Hiroshima University, Higashi-Hiroshima 739-852, Japan

ARTICLE INFO

Article history:
Received 10 November 2014
Received in revised form 12 April 2015
Accepted 7 May 2015
Available online 14 May 2015

Keywords:
Artificial bee colony algorithm
Swarm intelligence
Continuous optimization problems
Self adaptive mechanism
CEC 2014 test suite

ABSTRACT

The artificial bee colony (ABC) algorithm is one of popular swarm intelligence algorithms that inspired by the foraging behavior of honeybee colonies. To improve the convergence ability, search speed of finding the best solution and control the balance between exploration and exploitation using this approach, we propose a self adaptive hybrid enhanced ABC algorithm in this paper. To evaluate the performance of standard ABC, best-so-far ABC (BsfABC), incremental ABC (IABC), and the proposed ABC algorithms, we implemented numerical optimization problems based on the IEEE Congress on Evolutionary Computation (CEC) 2014 test suite. Our experimental results show the comparative performance of standard ABC, BsfABC, IABC, and the proposed ABC algorithms. According to the results, we conclude that the proposed ABC algorithm is competitive to those state-of-the-art modified ABC algorithms such as BsfABC and IABC algorithms based on the benchmark problems defined by CEC 2014 test suite with dimension sizes of 10, 30, and 50, respectively.

1. Introduction

Optimization is an applied science that determines the best values of given parameters for a given problem. The aim of optimization is to obtain the relevant parameter values that enable an objective function to generate the minimum or maximum value (Civicioglu and Besdok, 2013). In the recent years, various kinds of novel optimization algorithms have been proposed to solve real parameter optimization problems, including the IEEE Congress on Evolutionary Computation (CEC) 2005 and CEC 2013 test suite (Liang et al., 2013a). Based on the definition and some comments for modifications of CEC 2013 test suite, CEC 2014 benchmark problems were developed (Liang et al., 2013b). In this CEC 2014 test suite, hybrid and composition functions were additionally defined as complex optimization problems. CEC 2014 test suite is an invaluable resource which includes 30 benchmark functions without making use of surrogates or meta-models. To solve these more complex optimization problems, an effective and efficient swarm intelligence (SI) based or evolutionary optimization algorithm is required.

In the past decade year, swarm intelligence (SI), which is a discipline of artificial intelligence, has attracted the interest of many research scientists in related fields. Bonabeau et al. (1999) defined SI as “any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies”. SI based algorithms include particle swarm optimization (PSO) (Kennedy and Eberhart, 1995), ant colony optimization (Dorigo et al., 2006), artificial bee colony (ABC) (Karaboga, 2005) and cuckoo search (CS) (Yang and Deb, 2009). The most widely used evolutionary algorithms are genetic algorithm (GA) (Holland, 1975) and evolution strategy (ES) (Beyer, 2001) and differential evolution (DE) (Storn and Price, 1997). A recent study showed that the ABC algorithm performs significantly better or at least comparably to other SI algorithms (Civicioglu and Besdok, 2013; Karaboga and Akay, 2009a).

The ABC algorithm was introduced by Karaboga (2005) as technical report. Its performance was initially measured using benchmark optimization functions (Karaboga and Basturk, 2007, 2008). The ABC algorithm has been applied to several fields in various ways (Karaboga and Akay, 2009b; Karaboga et al., 2014), such as training neural networks (Karaboga and Akay, 2007), protein structure prediction (Benitez and Lopes, 2010), sensor deployment (Udgata et al., 2009), Wireless Sensor Network (Okdem et al., 2011), the redundancy allocation problem (Yeh and Hsieh, 2011), engineering design optimization (Akay and Karaboga, 2012a), data mining (Celik et al., 2011) and job shop scheduling (Yin et al., 2011).

* Corresponding author. Tel.: +81 82 424 7550.
E-mail addresses: haishan1010@gmail.com (H. Shan), yasu@hiroshima-u.ac.jp (T. Yasuda), kohkura@hiroshima-u.ac.jp (K. Ohkura).

http://dx.doi.org/10.1016/j.biosystems.2015.05.002
0303-2647/

The ABC algorithm is superior to other SI algorithms in terms of its simplicity, flexibility, and robustness. Karaboga and Akay (2009a) implemented comparison experiments and the results showed that the performance of ABC algorithm was better or similar to GA, DE, and PSO algorithms. In addition to the advantageous properties, the ABC algorithm requires fewer training parameters, so combining it with other algorithms is easier. Given its flexibility, the ABC algorithm has been revised in many recent studies. For example, Alatas (2010) proposed a chaotic ABC algorithm, in which many chaotic maps for parameters adapted from the standard ABC algorithm were introduced to improve its convergence performance. Zhu and Kwong (2010) proposed a gbest-guided ABC algorithm by incorporating the information of the global best solution into the solution search equation to improve the exploitation. In addition, Gao and Liu (2012) proposed a modified ABC (MABC) algorithm that used a modified solution search equation with chaotic initialization; further MABC excluded the onlooker bees and scout bees phases. Akay and Karaboga (2012b) proposed the modified ABC algorithm to overcome the slow convergence speed of the standard ABC algorithm. Banharnsakun et al. (2011) proposed BsfABC algorithm and exploited the best solution found so far. The best solution was used to modify the onlooker bee step, thus leaving employed bee step unchanged. Aydin et al. (2011) proposed incremental ABC (IABC) algorithm that integrates the population growth and local search with standard ABC algorithm.

However, along with the advantages of the improved versions of ABC, a few disadvantages still exist. For example, slower convergence speed for some unimodal problems and easily get trapped in local optima for some complex multimodal problems (Karaboga and Akay, 2009a), and low exploitation abilities. Richards and Ventura (2004) found that uniformity of the initial population plays a more important role in higher dimensional problems (up to 50 dimensions), in contrast, claim that uniform initialization methods lose their effectiveness in problems with dimensionality larger than 12, therefore, the initial population for more higher dimension size of ABC algorithm is not so effective. To overcome these disadvantages, we propose an self adaptive hybrid enhanced ABC algorithm inspired by levy flight (Brown et al., 2007; Pavlyukevich, 2007), a self adaptive mechanism for employed bees and onlooker bees steps, and combined with DE and PSO algorithms, at last, introduced chaotic opposition-based learning (OBL) in scout bee step (Tizhoosh, 2005).

We implemented comparative experiments and set up parameters for our proposed ABC algorithm to demonstrate the efficacy of the algorithm; more specifically, we used the CEC 2014 test suite benchmark problems to show the performance of proposed ABC algorithm. Finally, we implemented comparative experiments using our proposed ABC and the standard ABC, and state-of-the-art ABC algorithms such as BsfABC and IABC algorithms.

In addition to this introductory section, the remainder of this paper is organized as follows. The ABC algorithm is introduced in Section 2. In Section 3, we describe our proposed ABC algorithm. The experimental setup and results are discussed in Section 4, and we conclude our paper in Section 5.

2. The artificial bee colony (ABC) algorithm

The ABC algorithm is a swarm based meta-heuristic algorithm introduced by Karaboga (2005) that has successfully applied to numerical optimization problems (Karaboga and Basturk, 2007, 2008; Karaboga and Akay, 2007; Akay and Karaboga, 2012b). In the ABC algorithm, the artificial bee colony comprises three kinds of bees: employed bees, onlooker bees, and scout bees. Employed bees search for food source sites by modifying the site in their memory, evaluate the nectar amount of each new source, and memorize the more productive site through a selection process; these bees share information related to the quality of the food sources they exploit in the "dance area". Onlooker bees wait in the hive and decide a food source to exploit based on the information coming from employed bees. As such, more beneficial sources have higher probability to be selected by onlookers. Further, onlooker bees choose food sources depending on the given information through probabilistic selection and modify these sources. In order to decide if a source is to be abandoned, the counters which have been updated during search are used. If the value of the counter is greater than the control number of the ABC algorithm, known as the limit, the source associated with the counter is assumed to be exhausted and is abandoned. When the food source is abandoned, a new food source is randomly selected by a scout bee to replace the abandoned source. The main steps of the algorithm are given below:

1) Initialize the population of solutions x_{ij} with

$$x_{ij} = x_{\min,j} + \mathrm{rand}[0,1](x_{\max,j} - x_{\min,j}) \tag{1}$$

where $i \in 1, 2, \ldots, SN$ and $j \in 1, 2, \ldots, D$ are randomly selected indexes, SN is the number of food source, and D is the dimension size.

2) Evaluate the population.

3) Initialize cycle to 1.

4) Produce new solutions v_i for the employed bees by using x_{ij} mentioned in Eq. (1)

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{2}$$

where ϕ_{ij} is a uniformly distributed random number in the range $[-1,1]$; $i, k \in 1, 2, \ldots, SN$ are randomly selected indexes with k different from i and $j \in 1, 2, \ldots, D$ is a randomly selected index.

$$\mathrm{fit}_i = \begin{cases} \dfrac{1}{(1+f_i)}, & \text{if} \quad f_i \geq 0 \\ 1 + |f_i|, & \text{if} \quad f_i < 0 \end{cases} \tag{3}$$

then evaluate the solutions according to fitness value fit_i in minimization problem, where f_i is the cost value of solution v_i

5) Apply the greedy selection process for the employed bees.

6) If the solution does not improve, add 1 to the trail, otherwise, set the trail to 0.

7) Calculate probability values p_i for the solutions using Eq. (4) as

$$p_i = \frac{\mathrm{fit}_i}{\sum_{n=1}^{SN} \mathrm{fit}_n} \tag{4}$$

where fit_i is the fitness value of solution i.

8) Produce new solutions for the onlooker bees from solutions x_i, which is selected depending on p_i, then evaluate them.

9) Apply the greedy selection process for the onlooker bees.

10) If the solution does not improve, add 1 to the trail, otherwise, set the trail to 0.

11) Determine the abandoned solution through the number of limit for the scout, if it exists, and replace it with a new random solution using Eq. (1).

12) Memorize the best solution achieved so far.

13) Add 1 to cycle.

14) Repeat above cycles (4–13) until cycle reaches a predefined maximum cycle number (MCN).

To enhance the exploitation and exploration processes, best-so-far ABC (BsfABC) algorithm was proposed by Banharnsakun et al. (2011). In this BsfABC algorithm, three major changes were introduced. All onlooker bees use the information from all employed bees to make a decision on a new candidate food