



Composite artificial bee colony algorithms: From component-based analysis to high-performing algorithms



Doğan Aydın^{a,b,*}

^a Dumlupınar University, Computer Engineering Dept., 43000 Kütahya, Turkey

^b IRIDIA, Université Libre de Bruxelles (ULB), Brussels, Belgium

ARTICLE INFO

Article history:

Received 21 January 2015

Received in revised form 19 March 2015

Accepted 23 March 2015

Available online 7 April 2015

Keywords:

Artificial bee colony

Continuous optimization

Component-based analysis

Integration of algorithmic components

ABSTRACT

The artificial bee colony (ABC) algorithm is a swarm intelligence algorithm inspired by the intelligent foraging behavior of a honeybee swarm. In recent years, several ABC variants that modify some components of the original ABC algorithm have been proposed. Although there are some comparison studies in the literature, the individual contribution of each proposed modification is often unknown. In this paper, the proposed modifications are tested with a systematic experimental study that by a component-wise analysis tries to identify their impact on algorithm performance. This study is done on two benchmark sets in continuous optimization. In addition to this analysis, two new variants of ABC algorithms for each of the two benchmark sets are proposed. To do so, the best components are selected for each step of the Composite ABC algorithms. The performance of the proposed algorithms were compared against that of ten recent ABC algorithms, as well as against several recent state-of-the-art algorithms. The comparison results showed that our proposed algorithms outperform other ABC algorithms. Moreover, the composite ABC algorithms are superior to several state-of-the-art algorithms proposed in the literature.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

The artificial bee colony (ABC) algorithm is a recent swarm intelligence algorithm inspired by the foraging behavior of honeybee swarms. It was introduced by Karaboğa [1] who illustrated the algorithm applying it to continuous optimization problems. Following these early works, there has been a large number of follow-up researches and a large number of papers have proposed variants of the ABC algorithm. These variants often change only one single or at most two components of the original ABC algorithm. Examples of such modifications are changes in an equation that is defining the modifications applied to solutions or different ways of how to initialize the colony.

In a previous study [2], a comprehensive, experimental analysis has been conducted to investigate the behavior of nine different ABC variants on the same benchmark function set. In particular, we have examined the impact the (automatic) tuning of ABC algorithm parameters and the introduction of a local search phase have on performance and the contribution specific ABC variants make. Therefore, discussion about the general advantages and

disadvantages of ABC variants can be found in our previous study [2]. However, the impact individual proposals of modifications for ABC algorithm components have were not analyzed in that previous work. In particular, a systematic analysis of the individual contribution of each new ABC algorithm component on performance was not done.

This article has two main contributions. The first is a systematic study of the impact that specific modifications of the ABC algorithm have on its performance. It is done by decomposing the ABC algorithm in a number of algorithm components (for example the initialization, the search equation used by onlooker / employed bees, the usage or not of a local search etc.). Then, each of these components is studied and it is tried to identify their impact on algorithm performance. This study is done on two well accepted benchmark sets in continuous optimization. One of these sets is the CEC 2005 benchmark set that comprises the benchmark functions that have been used in the IEEE CEC'05 competition on real-parameter optimization [3]; the other set is the SOCO benchmark set that was used for a special issue of the Soft Computing journal on large-scale continuous optimization problems [4]. These sets are used as they are rather large sets of functions with different characteristics and by using them, the results become comparable to those of other algorithms from the literature.

Once the impact specific ABC algorithm components have on performance have been analyzed, a next natural step is to show that

* Correspondence to: Dumlupınar University, Computer Engineering Dept., 43000 Kütahya, Turkey. Tel.: +90 5306075741.
E-mail address: daydin@ulb.ac.be

the obtained knowledge is useful not only for a better understanding of ABC algorithms, but also for the design of new algorithms. In particular, as a next step we try to exploit the knowledge gained for designing new, high performing algorithms. This step is the second, main contribution of the article. In particular, we aim at identifying for each of the two benchmark sets that have been used in the component-wise analysis to define new ABC algorithm variants. Two new variants of ABC algorithms are proposed. They are called “composite artificial bee colony algorithms” (*CompABC_{cec}* and *CompABC_{soco}* in short for the composite ABC algorithms that are proposed for the CEC and the SOCO benchmark set, respectively). Obviously, putting together the algorithm components that had the best individual effect does not guarantee to obtain the best overall final algorithm. This is mainly due to interactions that may exist between these components. However, such an approach may be the one that is (and often has) been chosen by algorithm designers. Still, setting of the numerical algorithm parameters is helped by the use of automatic algorithm configuration techniques, which somehow alleviates the human algorithm designer from choosing specific algorithm parameters based on intuition and limited experiments.

The proposed algorithms were compared to several ABC variants and other state-of-the-art algorithms to assess their performance. The encouraging results of the proposed algorithm indicates the importance of component-based analysis for further new variants of ABC algorithms. It also indicates that the approach of designing algorithms based on combining promising algorithm components, as it has been done here, is a feasible procedure.

This paper is structured as follows. Section 2 briefly reviews the original ABC algorithm. The component-based view of ABC algorithms is presented in Section 3. Section 4 elaborates the experimental procedures and function suites, ABC variants and their components used in the experiments. Section 5 analyzes the impact the specific modifications have on ABC performance. In Section 6, the composite ABC Algorithms for two benchmark suites (*CompABC_{cec}* and *CompABC_{soco}*) are introduced. Comparisons of the Composite ABC algorithms to other ABC variants and state-of-the-art algorithms are presented in Section 7. Section 8 concludes the article.

2. Artificial bee colony algorithm

Real bee colonies consist of three different types of bees: employed bees, onlooker bees and scout bees [1]. Each bee has a specialized task in the colony and their aim is to maximize the amount of nectar stored in the hive using division of labor and self-organization [5]. This organizational nature and the behavior of the bees has inspired the artificial bee colony (ABC) algorithm. It was proposed in 2005 [1] for bound-constrained continuous optimization problems, which can be summarized as follows:

$$\begin{aligned} \min \quad & f(X) \\ \text{st} : \quad & x_j \in [x_j^{\min}, x_j^{\max}], \quad j = 1, 2, \dots, D \end{aligned} \quad (1)$$

where $f(X)$ is the objective function, $X = \{x_1, x_2, \dots, x_D\}$ is the set of decision variables, D is the number of decision variables, which is also called dimension (of the search space), and x_j^{\min} and x_j^{\max} are the lower and upper bound values for each decision variable x_j , respectively.

In ABC, each food source is placed in the D -dimensional search space and presents a potential solution to the problem. The amount of nectar at the food source is assumed to be the fitness value of a food source. Generally, the population size of the bee colony is considered to be twice the number of food sources. Half of the population is considered to be the employed bees and the other half to be onlooker bees.

Algorithm 1. The pseudo-code of the original artificial bee colony algorithm

```

Step 1. Initialization
while termination condition is not met do
  Step 2. Employed Bees Step
  Step 3. Onlooker Bees Step
  Step 4. Scout Bees Step
end while
    
```

The ABC algorithm consists of four steps: initialization, employed bees step, onlooker bees step and scout bees step. In the initialization step of the algorithm, the initial population is generated and other parameters are set. Then, the other three main steps of the algorithms are executed repeatedly in a loop until the termination condition is met. The outline of the algorithm is given in Algorithm 1 and the detailed description of the main steps is the following.

Step 1. Initialization In this step, SN food sources (candidate solutions), where SN denotes the size of population, are placed randomly on the D -dimensional search space as follows:

$$x_{i,j} = x_j^{\min} + \varphi_{i,j}(x_j^{\max} - x_j^{\min}), \quad (2)$$

where $\varphi_{i,j}$ is a random number distributed uniformly in $[0, 1]$ for dimension j and food source i . Besides, a visit *limit* of each food value is initialized for each food source.

Step 2. Employed bees There is a one-to-one correspondence between employed bees and food sources in this step. Each employed bee visits the food source to which it is associated and searches in the vicinity of the selected food source for better solutions. To find a new candidate food source location, v_i , the following search equation is used by taking advantage of a randomly chosen second food source, x_{r1} :

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{r1,j}), \quad (3)$$

where j is a randomly selected dimension ($j \in \{1, 2, \dots, D\}$), $\phi_{i,j}$ is a uniform random number in $[-1, 1]$, $x_{i,j}$ and $x_{r1,j}$ are the position of the reference food source x_i and a randomly selected food source x_{r1} in dimension j , respectively. If the candidate food source, v_i , is placed in a better location than the reference food source, x_i , it replaces x_i and becomes the new food source. The important point to note here is that the search equation changes only one dimension of the reference food source. On one hand, this may lead to improve the diversification behavior; on the other hand, it may lead to slow convergence.

Step 3. Onlooker bees Onlooker bees select randomly food sources to visit and search for the new food source locations using Eq. 3 as the employed bees do. Unlike the employed bees, onlooker bees take the amount of nectar on food sources into account when selecting a food source to visit. The better the fitness value, the higher the probability of being selected by an onlooker. The selection probability, p_i , of a food source x_i is calculated as follows:

$$p_i = \frac{fitness_i}{\sum_{n=1}^{SN} fitness_n}. \quad (4)$$

In Eq. 4, $fitness_i$ is the *fitness* value of food source x_i . It is defined as

$$fitness_i = \begin{cases} \frac{1}{1+f_i}, & f_i \geq 0, \\ 1 + abs(f_i), & f_i < 0, \end{cases} \quad (5)$$

where f_i is the objective function value of food source x_i . Using this probabilistic selection rule, onlooker bees usually visit better food sources and try to find new candidate locations around good

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات