Computers & Operations Research 40 (2013) 1256–1265

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

# An efficient and robust artificial bee colony algorithm for numerical optimization

Wan-li Xiang [a,b,*], Mei-qing An [a]

[a] School of Traffic & Transportation, Lanzhou Jiaotong University, Lanzhou, Gansu 730070, PR China
[b] Institute of Systems Engineering, Tianjin University, Tianjin 300072, PR China

ARTICLE INFO

Available online 20 December 2012

*Keywords:*
Artificial bee colony algorithm
Initialization based on chaos
Reverse selection based on roulette wheel
Solution search equation
Chaotic search

ABSTRACT

Artificial bee colony (ABC) algorithm has already shown more effective than other population-based algorithms. However, ABC is good at exploration but poor at exploitation, which results in an issue on convergence performance in some cases. To improve the convergence performance of ABC, an efficient and robust artificial bee colony (ERABC) algorithm is proposed. In ERABC, a combinatorial solution search equation is introduced to accelerate the search process. And in order to avoid being trapped in local minima, chaotic search technique is employed on scout bee phase. Meanwhile, to reach a kind of sustainable evolutionary ability, reverse selection based on roulette wheel is applied to keep the population diversity. In addition, to enhance the global convergence, chaotic initialization is used to produce initial population. Finally, experimental results tested on 23 benchmark functions show that ERABC has a very good performance when compared with two ABC-based algorithms.


## 1. Introduction

Up to now, there are a variety of biological-inspired optimization algorithms for different kinds of optimization problems, such as genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization (PSO), artificial bee colony (ABC) algorithm, and so on. The ABC algorithm was developed by Karaboga in 2005 through simulating the foraging behavior of honey bee swarm [1]. In some researches, the performance of ABC algorithm has already been justified better than some other evolutionary algorithms, such as GA, PSO, differential evolution (DE), and so on [2–4]. In view of its lesser adjustable parameters and the advantage of easy to code, the ABC has caught considerable attention and has been widely applied to various research fields, such as function optimization [2,5–13], clustering analysis [14,15], image processing [6], vehicle routing problem [16], signal processing [17], engineering design [18], chaotic system [19], and so on.

Unfortunately, like other evolutionary algorithms, ABC also faces up to some insufficiencies. For example, ABC can easily get trapped in local optima when solving complex multimodal function optimization problems [2], and its convergence speed is also an issue in some cases [9]. The reason for these issues is that it is too difficult for ABC to well balance between the exploration and exploitation. Consequently, more and more researchers are paying close attention to the improvement of ABC so as to overcome these shortages. Recently, a few modified or improved algorithms based on the classical ABC algorithm are proposed. For instance, Zhu and Kwong proposed the gbest-guided ABC (GABC) by employing a global best individual to guide the search [5]; Banharsakun et al. proposed a best-so-far ABC with modified solution search equation [6]; Gao and Liu proposed a modified artificial bee colony (MABC) algorithm [8] by using a modified solution search equation together with a novel chaotic initialization; Li et al. proposed two kinds of improved artificial bee colony algorithms, called IABC and PSABC, by introducing a set of combinatorial solution search equations [7]. These modified or improved artificial bee colony algorithms have shown a better performance than the classical ABC algorithm. Especially, the PSABC algorithm has shown very fast convergence speed and very accurate convergence precision in most cases.

However, so far, there is no specific algorithm to achieve the best solution for all optimization problems. Namely, as far as most algorithms are concerned, it is difficult to simultaneously balance the ability of exploration and exploitation for all the optimization problems. For example, both PSABC and IABC proposed in [7] have an issue on the convergence speed together with convergence precision on the benchmark functions $f_3, f_5$, etc. In other words, most algorithms are difficult to have a better performance in the aspects of convergence speed and convergence precision for all optimization problems at the same time.

Therefore, in order to further improve the convergence speed and the convergence precision of ABC, a modified artificial bee

* Corresponding author.
*E-mail addresses:* 583603444@qq.com, xiangwl@tju.edu.cn (W.-l. Xiang).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.12.006

colony algorithm is presented in this paper. The modified ABC algorithm, called an efficient and robust ABC algorithm (ERABC for short), has the characters of high-efficiency and robustness. That is, the proposed ERABC can achieve high-quality solutions not only with extremely fast convergence speed but also for almost all the benchmark functions employed in the paper and also used in Ref. [7]. So, the ERABC can be regarded as an efficient and robust algorithm. In addition, by comparing ERABC with ABC, there are four major differences between them. They are chaotic initialization based on logistic map, reverse selection based on roulette wheel, a new combinatorial solution search equation composed of best-so-far solution search equation together with its modified version, and chaotic search technique used on onlooker bees phase. Finally, experimental results tested on a set of 23 benchmark functions show that ERABC is superior to both ABC and PSABC in most cases. Even though ERABC is a little weaker than either ABC or PSABC in a few cases, both convergence precision and convergence speed of ERABC are extremely close to those of the better one between ABC and PSABC.

The rest of the paper is organized as follows: Section 2 summarizes the classical ABC. Section 3 describes the proposed ERABC in detail. And then the numerical experiments are carried out and the simulation results are analyzed in Section 4, which includes three parts. Finally, Section 5 draws a conclusion of the paper.

## 2. The classical artificial bee colony algorithm

The classical artificial bee colony algorithm developed by Karaboga in 2005 for numerical optimization [1] is a biological-inspired algorithm by mimicking the foraging behavior of honey bee swarm. In the ABC algorithm, the artificial honey bee colony consists of three groups of honey bees: employed bees, onlooker bees, and scout bees. As a matter of fact, half of the colony is employed bees, and the rest is onlooker bees. Namely, the number of employed bees is equal to the number of onlooker ones, and equal to the number of food sources or solutions. Thereinto, the employed bees take charge of searching food around the food sources in their memory, after that, they share the food amount information with the onlooker bees by dancing in the nearby hive. Thereafter, onlooker bees are inclined to select a better food source according to their fitness, that is, the more the food amount of the food source is, the larger the probability that the corresponding food source is chosen. Then the onlooker bees further search a new food source around the chosen food source. In addition, if the food amount of the new food source found by an onlooker bee is more than that of the previous one, the employed bee's food source position is replaced by the new one in their memory. Accordingly, a so-called parameter *trial*, which is used to record the number denoting whether the employed bee's food source position is replaced, is set to zero when the employed bee's food source position is replaced, or else the parameter *trial* is going to be plus one in the employed bee's memory. Although the parameter *trial* is greater than another parameter *limit*, which is often a predetermined constant in ABC, the employed bee's food source is abandoned, and the corresponding employed bee becomes a scout bee. Subsequently, the scout bee randomly searches a new food source to substitute the abandoned food source.

In a word, the ABC algorithm is an iterative optimization process, which can be described as follows.

### 2.1. The initial phase

At first, in order to find a better food source, the actual bees start to explore the environment randomly. That is, randomly distributed $SN$ food source positions are generated for $SN$ employed bees. And each food source position $X_i$ corresponds to a solution, which is a $D$-dimensional vector (i.e., $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})$). The process of initial phase can be represented by the equation below:

$$x_{i,j} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \quad (1)$$

where $i = 1,2,\ldots,SN$; $j = 1,2,\ldots,D$. $x_j^{min}$ is the lower bound of the food source position in dimension $j$ and $x_j^{max}$ is the upper bound of the food source position in dimension $j$.

### 2.2. The employed bee phase

At the stage, for each position of employed bee's food source $X_i$, a new food source position $V_i$ is generated by the equation as follows:

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) \quad (2)$$

where $k \in \{1,2,\ldots,SN\}$ and $j \in \{1,2,\ldots,D\}$. $k$ and $j$ are randomly generated, and $k$ must be different from $i$. $\phi_{i,j}$ is a random number between $[-1,1]$. The above explanation implies that the other components of $V_i$ except for dimension $j$ are the same as the ones of $X_i$. Then, a greedy selection is made between $X_i$ and $V_i$.

### 2.3. The probabilistic selection phase

After completing their update process, employed bees share their nectar amount information related to the food sources with the onlooker bees in the nearby hive. Here and now, an onlooker bee will randomly choose a food source with a probability value $p_i$, which is calculated by the following form:

$$p_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \quad (3)$$

where $fit_i$ denotes the fitness value of solution $X_i$. It is clear that the higher the fitness value of solution $X_i$ is, the higher the probability of selecting the $i$th food source corresponding to solution $X_i$. Besides, the fitness value $fit_i$ is defined as follows:

$$fit_i = \begin{cases} \dfrac{1}{1+f(X_i)} & \text{if } f(X_i) \geq 0, \\ 1+|f(X_i)| & \text{if } f(X_i) < 0, \end{cases} \quad (4)$$

where $f(X_i)$ denotes the objective function values of the decision vector $X_i$.

### 2.4. The onlooker bee phase

According to the probability value $p_i$ calculated by Eq. (3), each onlooker bee randomly chooses a food source corresponding to the solution $X_i$ with a probability value $p_i$. And then, it searches a new food source corresponding to solution $V_i$ around the chosen food source by Eq. (2). On the onlooker bees phase, the greedy selection is also applied between the solution $X_i$ and $V_i$.

### 2.5. The scout bee phase

Although the number *trial*, denoting the times that the food source position corresponding to the solution $X_i$ is not improved continuously in the honeybee's memory, is greater than the predetermined parameter *limit* employed in ABC, the corresponding employed bee abandons the food source and becomes a scout bee, whereafter, the scout bee randomly searches a new food source using equation below:

$$x_{i,j} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \quad (5)$$

where $j = 1,2,\ldots,D$.