Applied Soft Computing xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

A novel artificial bee colony algorithm with Powell's method

Q1 Wei-feng Gao [a,*], San-yang Liu [a], Ling-ling Huang [b]

[a] School of Science, Xidian University, Xi'an, Shannxi 710071, China
[b] School of Science, China University of Petroleum, Qingdao 266580, China

ARTICLE INFO

Article history:
Received 25 November 2012
Received in revised form 21 March 2013
Accepted 17 May 2013
Available online xxx

Keywords:
Evolutionary algorithms
Artificial bee colony algorithm
Search equation
Powell's method

ABSTRACT

Artificial bee colony (ABC) algorithm is a relatively new optimization technique which has been shown to be competitive to other population-based algorithms. However, there is still an insufficiency in ABC regarding its solution search equation, which is good at exploration but poor at exploitation. To address this concerning issue, we first propose a modified search equation which is applied to generate a candidate solution in the onlookers phase to improve the search ability of ABC. Further, we use the Powell's method as a local search tool to enhance the exploitation of the algorithm. The new algorithm is tested on 22 unconstrained benchmark functions and 13 constrained benchmark functions, and are compared with some other ABCs and several state-of-the-art algorithms. The comparisons show that the proposed algorithm offers the highest solution quality, fastest global convergence, and strongest robustness among all the contenders on almost all test functions.

1. Introduction

The algorithms for global optimization problems are of increasing importance in many fields of science and engineering. They can be divided into two categories: derivative-based methods and derivative-free methods. For a variety of reasons there have always been many real-world problems where derivatives are unavailable or unreliable. Thus, as an important branch of derivative-free methods, evolutionary algorithms (EAs) have shown considerable evolutionary algorithms (EAs) have shown considerable success in solving optimization problems characterized by nonconvex, discontinuous, non-differentiable and so on, and attracted more and more attention in recent years. The most prominent EAs proposed in the literatures are genetic algorithm (GA) [1], particle swarm optimization (PSO) [2], differential evolution (DE) [3], ant colony optimization (ACO) [4], biogeography-based optimization (BBO) [5] and artificial bee colony (ABC) algorithm [6] and so on.

In this paper, we concentrate on ABC, developed by Karaboga [6] in 2005 based on simulating the foraging behavior of honey bee swarm. Numerical comparisons demonstrated that the performance of ABC is competitive to that of other population-based algorithms with an advantage of employing fewer control parameters [7–10]. Due to its simplicity and ease of implementation, ABC has captured much attention and has been applied to solve many practical optimization problems [11–13] since its invention.

However, the solution search equation of ABC is good at exploration but poor at exploitation [14], which results in the poor convergence. To improve the performance of ABC, a number of variant ABC algorithms have, hence, been proposed to achieve these goals. One active research trend is to study search equation. Until now, various search equations have been suggested, such as [14–20]. The most representative of them is that inspired by PSO, Zhu and Kwong [14] proposed an gbest-guided ABC (GABC) by incorporating the information of global best (gbest) solution into the solution search equation to improve the exploitation. Hybridization of ABC with other operators have also been studied widely, such as [21–28]. For example, Kang et al. [23] used the Rosenbrock's rotational direction method to implement the exploitation phase and proposed the Rosenbrock ABC algorithm. Alatas [24] introduced the chaotic maps into the initialization and chaotic search into the scout bee phase and proposed the chaotic ABC. It is necessary to emphasize that our work falls in both of the categories.

We note that Powell's method [29] has a strong ability of local search. In order to make use of the good exploitation of Powell's method, we propose an improved ABC algorithm-Powell ABC (PABC) for global optimization in this paper. First, a new search equation is employed to generate a candidate food position in the onlookers phase to improve the search ability. Next, Powell's method is incorporated to ABC as a local search tool to enhance the exploitation of the algorithm. Therefore, ABC and Powell's method have complementary advantages, and the proposed algorithm can result in a faster and more robust method. The efficiency of the new algorithm was tested by a suite of unimodal/multimodal benchmark functions.

* Corresponding author.
E-mail address: gaoweifeng2004@126.com (W.-f. Gao).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.05.012

The rest of this paper is organized as follows. Section 2 summarizes artificial bee colony algorithm. Powell's method is reviewed in Section 3. The proposed PABC approach is presented in Section 4. Section 5 presents and discusses the experimental results. Finally, the conclusion is drawn in Section 6.

2. Artificial bee colony algorithm

ABC is a recently introduced optimization algorithm proposed by Karaboga [6] in 2005 which is inspired by the intelligent foraging behavior of honeybee swarm. In ABC, the colony of artificial bees is divided into three groups: employed bees, onlookers and scouts. Half of the colony consists of the employed bees, and another half consists of the onlookers. Employed bees are responsible for searching available food sources and gathering required information. They also pass their food information to onlooker bees. The onlookers select good food sources from those found by the employed bees to further search the foods. When the quality of the food source is not improved through a predetermined number of cycles, the food source is abandoned by its employed bee. And then the employed bee becomes a scout and starts to search for a new food source in the vicinity of the hive.

In ABC, the position of a food source corresponds to a possible solution to the optimization problem, and the nectar amount of each food source represents their quality (fitness) of the associated solution. The number of the employed bees equals to the number of food sources. At the initialization step, ABC generates a randomly distributed initial population P of SN solutions (food source positions), where SN denotes the size of employed bees or onlooker bees. Each initial solution $X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,D}\}$ is produced randomly within the range of the boundaries of the parameters as follows:

$$x_{i,j} = x_{min,j} + rand(0,1)(x_{max,j} - x_{min,j}), \quad (2.1)$$

where $i = 1, 2, \ldots, SN$, $j = 1, 2, \ldots, D$ and D is the number of optimization parameters; $x_{min,j}$ and $x_{max,j}$ are the lower and upper bounds for the dimension j, respectively.

After the initialization, the population of the food sources (solutions) is subjected to repeated cycles of the search processes of the employed bees, onlookers and scouts. Each employed bee always remembers its previous best position and produces a new position within its neighborhood in its memory. If the new food source has equal or better quality than the old source, the old source is replaced by the new source. Otherwise, the old source is retained. After all employed bees finish their search process, they will share the information about nectar amounts and positions of food sources with onlookers. Each onlooker chooses a food source depending on the probability value associated with the food source and searches the area within its neighborhood to generate a new candidate solution. And then, as in the case of the employed bee, the greedy selection is applied again. When the nectar of a food source is abandoned, the corresponding employed bee becomes a scout. The scout will randomly generate a new food source to replace the abandoned position.

An onlooker bee chooses a food source depending on the probability value p_i associated with that food source, where

$$p_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j}, \quad (2.2)$$

and fit_i is the fitness value of solution i. In this way, the employed bees exchange their information with the onlookers.

In order to produce a candidate food position V_i from the old one X_i in memory, ABC uses the following expression:

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}), \quad (2.3)$$

where $k \in \{1, 2, \ldots, SN\}$ and $j \in \{1, 2, \ldots, D\}$ are randomly chosen indexes; k has to be different from i, and ϕ_{ij} is a random number in the range $[-1, 1]$.

If a position cannot be improved further through a predetermined number of cycles, then that food source is assumed to be abandoned. The value of predetermined number of cycles is an important control parameter of ABC, which is called *limit* for abandonment. Assume that the abandoned source is X_i. Then the scout produces a food source randomly as in Eq. 2.1 to replace with X_i.

3. Powell's method

Powell's method, strictly Powell's conjugate gradient descent method, is an algorithm proposed by M.J.D. Powell [29] for finding a local minimum of a function. The function need not be differentiable, and no derivatives are taken.

The method minimizes the function by a bi-directional search along each search vector, in turn. The new position can then be expressed as a linear combination of the search vectors. The new displacement vector becomes a new search vector, and is added to the end of the search vector list. Meanwhile the search vector which contributed most to the new direction, i.e. the one which was most successful, is deleted from the search vector list. The algorithm iterates an arbitrary number of times until no significant improvement is made. The detailed pseudo-code of Powell's method procedure is presented as follow.

Algorithm 1. (Powell's method)

Initialize the starting point X_1, independent vectors $d_i = e_i$ ($i = 1, 2, \ldots, D$), the tolerance for stop criteria ε, set $f(1) = f(X_1)$, $X_c(1) = X_1$, k=1.
While (stopping criterion is not met, namely $|\triangle f| > \varepsilon$) **do**
  For $i = 1$ to D **do**
    If ($k \geq 2$) **then**
      $d_i = d_{i+1}$
    End If
    $X_{i+1} = X_i + \lambda_i d_i$, λ_i is determined by Minimizing $f(X_{i+1})$
  End For
  $d_{i+1} = \sum_1^D \lambda_i^* d_i = X_{D+1} - X_D$, $X_c(k+1) = X_{D+1} + \lambda_k d_{i+1}$, $f(k+1) = f(X_c(k+1))$
  $k = k+1$, $X_1 = X_c(k)$, $\triangle f = f(k) - f(k-1)$
End While

4. Powell artificial bee colony algorithm

4.1. A new search equation

To improve the performance of ABC, one active research trend is to study its search equation. Until now, various search equations have been suggested, such as [14–20]. The most representative of them is that inspired by PSO, which, in order to improve the exploitation and take advantage of the information of the global best (gbest) solution to guide the search of candidate solutions, Zhu and Kwong [14] presented the following solution search equation in GABC:

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) + \psi_{i,j}(y_j - x_{i,j}), \quad (4.1)$$

where the third term in the right-hand side of Eq. 4.1 is a new added term called gbest term, y_j is the jth element of the global best solution, $\psi_{i,j}$ is a uniform random number in [0,1.5]. However, since the guidance of the last two terms may be in opposite directions, it may cause an "oscillation" phenomenon. This phenomenon causes inefficiency to the search ability of the algorithm and delays convergence.

However, we note that a well-designed search equation is usually beneficial to enhance the performance of the algorithm. Hence, we present a new search equation as follows:

$$v_{i,j} = x_{k,j} + rand(0,1)(x_{best,j} - x_{k,j}), \quad (4.2)$$

Please cite this article in press as: W.-f. Gao, et al., A novel artificial bee colony algorithm with Powell's method, Appl. Soft Comput. J. (2013), http://dx.doi.org/10.1016/j.asoc.2013.05.012