Applied Soft Computing 11 (2011) 2888–2901

Contents lists available at ScienceDirect

Applied Soft Computing

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

The best-so-far selection in Artificial Bee Colony algorithm

Anan Banharnsakun, Tiranee Achalakul, Booncharoen Sirinaovakul*

Department of Computer Engineering, King Mongkut's University of Technology Thonburi, Bangkok, Thailand

A R T I C L E I N F O

Article history:
Received 15 February 2010
Received in revised form 31 May 2010
Accepted 28 November 2010
Available online 4 December 2010

Keywords:
Artificial Bee Colony
Swarm intelligence
Optimization
Image registration
Mutual information

A B S T R A C T

The Artificial Bee Colony (ABC) algorithm is inspired by the behavior of honey bees. The algorithm is one of the Swarm Intelligence algorithms explored in recent literature. ABC is an optimization technique, which is used in finding the best solution from all feasible solutions. However, ABC can sometimes be slow to converge. In order to improve the algorithm performance, we present a modified method for solution update of the onlooker bees in this paper. In our method, the best feasible solutions found so far are shared globally among the entire population. Thus, the new candidate solutions are more likely to be close to the current best solution. In other words, we bias the solution direction toward the best-so-far position. Moreover, in each iteration, we adjust the radius of the search for new candidates using a larger radius earlier in the search process and then reduce the radius as the process comes closer to converging. Finally, we use a more robust calculation to determine and compare the quality of alternative solutions. We empirically assess the performance of our proposed method on two sets of problems: numerical benchmark functions and image registration applications. The results demonstrate that the proposed method is able to produce higher quality solutions with faster convergence than either the original ABC or the current state-of-the-art ABC-based algorithm.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Swarm Intelligence is a meta-heuristic method in the field of artificial intelligence that is used to solve optimization problems. It is based on the collective behavior of social insects, flocks of birds, or schools of fish. These animals can solve complex tasks without centralized control.

Researchers have analyzed such behaviors and designed algorithms that can be used to solve combinatorial and numerical optimization problems in many science and engineering domains. Previous research [1–4] has shown that algorithms based on Swarm Intelligence have great potential. The algorithms that have emerged in recent years include Ant Colony Optimization (ACO) [5] based on the foraging behavior of ants, and Particle Swarm Optimization (PSO) [6] based on the behaviors of bird flocks and fish schools.

Exploration and exploitation are the important mechanisms in a robust search process. While exploration process is related on the independent search for an optimal solution, exploitation uses existing knowledge to bias the search. In the recent years, there are a few algorithms based on bee foraging behavior developed to improve both exploration and exploitation for solving the numerical optimization problems.

The Artificial Bee Colony (ABC) algorithm introduced by D. Karaboga [7] is one approach that has been used to find an optimal solution in numerical optimization problems. This algorithm is inspired by the behavior of honey bees when seeking a quality food source. The performance of ABC algorithm has been compared with other optimization methods such as Genetic Algorithm (GA), Differential Evolution algorithm (DE), Evolution Strategies (ES), Particle Swarm Optimization, and Particle Swarm Inspired Evolutionary Algorithm (PS-EA) [8–10]. The comparisons were made based on various numerical benchmark functions, which consist of unimodal and multimodal distributions. The comparison results showed that ABC can produce a more optimal solution and thus is more effective than the other methods in several optimization problems [11–13].

Yang [14] introduced an algorithm called the Virtual Bee Algorithm (VBA) for solving engineering optimizations that have multi-peaked functions. In the VBA algorithm, the objectives or optimization functions are encoded as virtual foods. Virtual bees are used to search for virtual foods in the search space. The position of each virtual bee is updated via the virtual pheromone from the neighboring bees. The food with largest number of virtual bees or intensity of visiting bees corresponds to the optimal solution. However, the VBA algorithm was only tested using two-dimension functions.

An optimization algorithm inspired by the honey bee foraging behavior based on the elite bee method was proposed by Sundareswaran [15]. The bee whose solution is the best possible solution in each simulation iteration is considered to be the elite

* Corresponding author.
E-mail addresses: anan_cpe@yahoo.com (A. Banharnsakun), tiranee@cpe.kmutt.ac.th (T. Achalakul), boon@kmutt.ac.th (B. Sirinaovakul).

1568-4946/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2010.11.025

```
Initialization:
  For i = 1 to n(FS)                 //do for all food sources
    For d = 1 to D
      x(FS, id) = x(min, d) + rand[0,1] * (x(max, d) − x(min, d))//initialize the feasible solutions in the search space
    Next d
  Next i
While (iteration ≤ MaxIteration)
  For i = 1 to n(EB)                 //do for each employed bees
    For d = 1 to D
      x(EB, id) = x(FS, id)          //assign the food source position to the employed bee
    Next d
    Select d_s                       //Randomly select the dimension of the solution
    Select x_n(FS, d_s)              //Randomly select the neighboring solution
    x(EB, id_s) = x(FS, id_s) + rand[−1,1] * (x(FS, id_s) − x_n(FS, d_s))          //update the position of the employed bee
    If f(x(EB, i)) < f(x(FS, i))          //compare and select the better solution between the old and the new solution
      x(FS, id_s) = x(EB, id_s)      //Replace the old solution with the new solution
      limit_count(x(FS, i)) = 0
    Else
      limit_count(x(FS, i)) + +
  Next i
  For i = 1 to n(EB)                 //do for all employed bees for selecting the best-so-far solution
    If i = 1
      For d = 1 to D
        x_b(FS, d) = x(FS, id)
      Next d
      f(x_b(FS)) = f(x(FS, i))
    Else
      If f(x(FS, i)) < f(x_b(FS))
        For d = 1 to D
          x_b(FS, d) = x(FS, id)
        Next d
        f(x_b(FS)) = f(x(FS, i))
  Next i
  For i = 1 to n(OB)                 //do for each onlooker bee
    Select d_s                       //Randomly select the dimension of the solution
    Select x_s(FS, d_s)              //Select the food source based on equation 2.3
    For d = 1 to D
      x(OB, id) = x_s(FS, d_s) + rand[−1,1] * fitness(x_b(FS)) * (x_s(FS, d_s) − x_b(FS, d_s))          //update the position of the onlooker bee
    Next d
    If f(x(OB, i)) < f(x_s(FS))      //compare and select the better solution between the old and the new solution
      For d = 1 to D
        x_s(FS, d) = x(OB, id)       //Replace the old solution with the new solution
      Next d
      limit_count(x_s(FS)) = 0
    Else
      limit_count(x_s(FS)) + +
  Next i
  For i = 1 to n(FB)                 //do for all food sources for abandoning the food source that cannot improve the further result
    If limit_count(x(FS, i)) > limit
      For d = 1 to D
        x(SB, d) = x(FS, i) + rand[−1,1] * (ω_max − (iteration / MaxIteration)(ω_max − ω_min)) * x(FS, i)          //update the position of the scout bee
      Next d
      If f(x(SB)) < f(x(FS, i))      //compare and select the better solution between the old and the new solution
        For d = 1 to D
          x(FS, id) = x(SB, d)       //Replace the old solution with the new solution
        Next d
        limit_count(x(FS)) = 0
      Else
        limit_count(x(FS)) + +
  Next i
```

Fig. 1. The pseudo-code of the Best-so-far ABC algorithm.

bee. A probabilistic approach is used to control the movement of the other bees, so majority of bees will follow the elite bee's direction while a few bees may fly to other directions. This approach improves the capability of convergence to a global optimum.

To improve the exploration and exploitation of foraging behavior of honey bees for numerical function optimization, Akbari et al. [16] presented an algorithm called Bee Swarm Optimization (BSO). In this method, the bees of the swarm are sorted according to the fitness values of the most recently visited food source and these sorted bees are divided into three types. The bees that have worst fitness are classified as scout bees, while the rest of bees are divided equally as experienced foragers and onlookers. Different flying patterns were introduced for each type of bee to balance the exploration and exploitation in this algorithm.

دریافت فوری ← متن کامل مقاله

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