Information Sciences 253 (2013) 34–55

Contents lists available at ScienceDirect

Information Sciences

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

Performance analysis of the coarse-grained parallel model of the artificial bee colony algorithm

Alper Basturk [a,*], Rustu Akay [b]

[a] Dept. of Computer Engineering, Erciyes University, Kayseri 38039, Turkey
[b] Graduate School of Natural and Applied Sciences, Erciyes University, Kayseri 38039, Turkey

ARTICLE INFO

Article history:
Received 9 November 2012
Received in revised form 14 June 2013
Accepted 16 August 2013
Available online 26 August 2013

Keywords:
Artificial bee colony optimization algorithm
Global optimization
Parallel computing
Message passing interface

ABSTRACT

Despite the efficiency of evolutionary algorithms is prominent for large scale problems, their running times in terms of CPU time are quite large. Multi processing units served by recent hardware developments can be employed to overcome this drawback reducing the running time and sharing the total workload. However, evolutionary algorithms cannot be directly distributed to processing units due to their cooperative working models. These models need to be modified to be able to run them on distributed environments without causing deterioration in performance. In this study, a detailed performance analysis of a parallel model for the artificial bee colony algorithm, which is one of the recently developed swarm based evolutionary algorithms and a promising numerical optimization tool, is proposed. For this purpose large-scale benchmark problems are solved by the proposed model and also its original sequential counterpart model. The model is also applied to a real-world problem: training of neural networks for classification purposes. Comparative results show that the artificial bee colony algorithm is very suitable to use in parallel architectures since it has the ability to produce high quality solutions with small populations due to its perturbation operator. The proposed model decreases the running time in addition to improving the performance and convergence rate of the algorithm. It can be said that the speedup gained over its sequential counterpart is almost linear.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

Evolutionary algorithms (EAs) model natural evolution and are used to solve a wide range of problems [11]. Many various versions of EAs based on different models have been proposed in the literature [4,36].

The artificial bee colony (ABC) algorithm [15] introduced by Karaboga is a recent swarm-based evolutionary computation technique and models the foraging behavior of honey bees. The algorithm has been further studied by Karaboga and Basturk [9,20,2,21,17,1], and has been found to be a good optimization tool especially for high dimensional and multi-modal problems. The ABC algorithm combines exploration and exploitation in a balanced manner and has just a few parameters including the population size, the number of cycles and the control parameter limit which determines some solutions to be abandoned. Several modifications to the original ABC algorithm have been presented to improve its performance for certain specific problems [30,14,28,24,37]. Karaboga and Akay [17] compared the ABC algorithm with genetic algorithm, particle swarm optimization algorithm, differential evolution algorithm and evolution strategies. They pointed out the differences and similarities between the ABC algorithm and the others. Their study showed that the performance of the ABC algorithm

[*] Corresponding author.
E-mail addresses: ab@erciyes.edu.tr (A. Basturk), akay@erciyes.edu.tr (R. Akay).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.08.035

is better than or similar to those of other population-based algorithms with the advantage of employing fewer control parameters [17].

Like other EAs, the running time of the ABC algorithm can be reduced while solving large-scale problems by using parallel implementations. However, most of the population-based algorithms including the ABC algorithm work sequentially in nature. To make them work in a parallel manner and communicate the units working in parallel with each other, some modifications need to be introduced into the original algorithm without loss of efficiency.

In the literature, some recent parallel ABC algorithm models based on different parallelization strategies are available. The first parallel implementation of the ABC algorithm was used through a shared memory architecture [27]. In that study, a whole colony was divided into local populations, each of which has a local memory, and a global solution is stored in a global shared memory. The approach provided speedup while it did not change the performance of the original algorithm. Luo et al. proposed a ripple communication strategy and investigated the effect of high population sizes and also different migration sizes [23]. Subotic et al. assigned different threads to separate the subpopulations and proposed different types of communications among these subpopulations [32,33]. Benitez and Lopes presented two parallel approaches for the ABC algorithm: a master–slave and a hybrid-hierarchical approach [10]. The hybrid-hierarchical approach improved the quality of solutions while the master–slave approach did not produce satisfactory results. Three parallel models, which are the master–slave, multi-hive with migrations, and hybrid hierarchical, were examined in [29,6]. The multi-population approaches generated better quality with less computational effort. These studies introduced low speedups since they employed a relatively small number of CPUs. Another issue related to these studies is that they used low dimensional problems and did not analyze benchmark problems requiring high computational effort. Very recently, Basturk and Akay implemented a parallel synchronous ABC algorithm based on a master–slave model and the results showed that the proposed model is as efficient as the asynchronous version while it requires much less time to solve large problems [8].

To the best of our knowledge, any study on the coarse-grained model-based parallel ABC algorithm which investigates the effects of the number of subpopulations, migration interval and different migration topologies is not available in the literature yet. One of the aims of this study is to conduct such an analysis. The other purpose of the study is to employ the proposed parallel algorithm to a real-world problem in addition to large-scale benchmark problems. The real-word problem considered in the study is training of an artificial neural network (ANN). The ANN is a computing system that can learn on its own and has been widely used in many real-world applications. The design of the ANNs is a difficult and also an important optimization problem since the success of the ANNs depends greatly on the training process and the training algorithm used [35]. Many optimization algorithms have been applied to train the ANNs. The ABC algorithm has also been applied successfully for training the ANNs [16,18,19].

In this study, a coarse-grained model based parallel ABC algorithm (PABC) is implemented through the message passing interface (MPI) [12] among multi processing units. For this purpose, the whole population is divided into small size subpopulations and each subpopulation is evolved independently on a different core. The evolved subpopulations interact with each other and the information regarding a subpopulation migrates to its neighbors in some periods. Throughout the proposed study, some modifications such as the number of food sources in subpopulations, the migration interval, the migration size and migration topology which controls the amount of information exchanged among the subpopulations, are introduced to control the search behavior of the algorithm. The effect of these control parameters are analyzed to determine the parameter set that yields the best solutions in terms of solution quality and CPU time.

The performance of the PABC algorithm is compared to its sequential counterpart's to check the level of improvement. Results show that the modification in the original model provides a decrease in CPU time in addition to an improvement in solution quality. The reason for the decrease in CPU time is parallel implementation of the algorithm and distribution of processing load to multi processing units. Also, the reason for improved solution quality is that the ABC algorithm can work better with low size populations due to its perturbation operator which changes one parameter at a time.

The rest of the paper is organized as follows. Section 2 provides a description of the original sequential ABC algorithm. The proposed parallel implementation is explained in Section 3. Experimental results are reported in Section 4 in detail. Finally, conclusions and future research lines are provided in Section 5.

2. Artificial bee colony algorithm

The ABC algorithm is inspired by the intelligent foraging behavior of a honey bee colony in which bees try to maximize the nectar amount loaded into the hive by bees assigned a specific task. In the foraging behavior, there are three types of bees, each of which has a different search characteristic: these are employed bees, onlooker bees and scout bees. The employed bees are responsible for exploiting the sources in their memory whereas the onlooker bees go to exploit potentially rich sources depending on the information taken by communicating with the employed bees. Scout bees are responsible for exploring undiscovered sources by a random motivation or external clue. The ABC algorithm mimics this foraging behavior of honey bees in the context of a global optimization algorithm. From the perspective of a meta-heuristic, a population of solutions corresponds to the food sources to be discovered and the optimization task is to find the most profitable source by using the different search characteristics of the employed bees, onlooker bees and scout bees. Each type of bee is a phase of the algorithm as given in Algorithm 1. The block diagram of the algorithm is given in Fig. 1.

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