Applied Soft Computing 11 (2011) 5181–5197

Contents lists available at ScienceDirect

Applied Soft Computing

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

A survey on parallel ant colony optimization

Martín Pedemonte, Sergio Nesmachnow*, Héctor Cancela

Facultad de Ingeniería, Universidad de la República, Uruguay

ARTICLE INFO

Article history:
Received 8 November 2010
Received in revised form 26 April 2011
Accepted 23 May 2011
Available online 31 May 2011

Keywords:
Ant colony optimization
Parallel implementations
Taxonomy

ABSTRACT

Ant colony optimization (ACO) is a well-known swarm intelligence method, inspired in the social behavior of ant colonies for solving optimization problems. When facing large and complex problem instances, parallel computing techniques are usually applied to improve the efficiency, allowing ACO algorithms to achieve high quality results in reasonable execution times, even when tackling hard-to-solve optimization problems. This work introduces a new taxonomy for classifying software-based parallel ACO algorithms and also presents a systematic and comprehensive survey of the current state-of-the-art on parallel ACO implementations. Each parallel model reviewed is categorized in the new taxonomy proposed, and an insight on trends and perspectives in the field of parallel ACO implementations is provided.

1. Introduction

In the last twenty years, the research community has been searching for new optimization techniques that are able to improve over the traditional exact ones, whose large computational requirements often make them useless for solving complex real-life optimization problems in acceptable times. In this context, nature-inspired metaheuristic methods have emerged as flexible and robust tools for solving NP-hard optimization problems, exploiting their ability to compute accurate solutions in moderate execution times [13,49]. Ant colony optimization (ACO) is a swarm intelligence population-based metaheuristic inspired in the social behavior of ant colonies, which applies the key concepts of distributed collaboration, self-organization, adaptation, and distribution found in ant communities, in order to efficiently solve real-life optimization problems [41].

Parallel implementations became popular in the last decade in order to improve the efficiency of population-based metaheuristics. By splitting the population into several processing elements, parallel implementations of metaheuristics allow reaching high quality results in a reasonable execution time, even when facing hard-to-solve optimization problems [2]. Parallel algorithms not only take benefit of using several computing elements to speed up the search, they also introduce a new exploration pattern that is often useful to improve over the result quality of the sequential implementations.

Many papers can be found in the related literature stating that parallel implementations are useful to improve the ACO exploration pattern; Fig. 1 shows the number of publications per year in this area. However, researchers often lack a generalized point of view, since they usually tackle a unique implementation to solve a specific problem.

Dorigo [39,40] first suggested the application of parallel computing techniques to enhance both the ACO search and its computational efficiency, while Randall and Lewis [84] proposed the first classification of ACO parallelization strategies. The book chapter by Janson et al. [57] and the article by Ellabib et al. [46] are the only previous works that have collected bibliography of published papers proposing parallel ACO implementations. Janson et al. reviewed parallel ACO proposals published up to 2002, focusing on comparing "parallelized" standard ACO algorithms, specific parallel ACO methods, and hardware parallelization; although they did not include an explicit algorithmic taxonomy. Ellabib et al. briefly commented parallel ACO implementations up to 2004, focusing in describing the applications, and they only distinguished between coarse-grain and fine-grain models for parallel ACO.

The classic proposals of parallel ACOs focused on traditional supercomputers and clusters of workstations. Nowadays, the novel emergent parallel computing architectures such as multicore processors, graphics processing units (GPUs), and grid environments provide new opportunities to apply parallel computing techniques to improve the ACO search results and to lower the required computation times.

In this line of work, the main contributions of this article are: (i) to introduce a new taxonomy to classify software-based parallel ACO algorithms, (ii) to present a systematic and comprehensive survey of the current state-of-the-art on parallel ACO implementations, and (iii) to provide an insight of the current trends and perspectives in the field. The survey focuses mainly on the parallel models, addressing the principal characteristics of each proposal, the experimental analysis – including the optimization problems

* Corresponding author.
E-mail address: sergion@fing.edu.uy (S. Nesmachnow).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.05.042

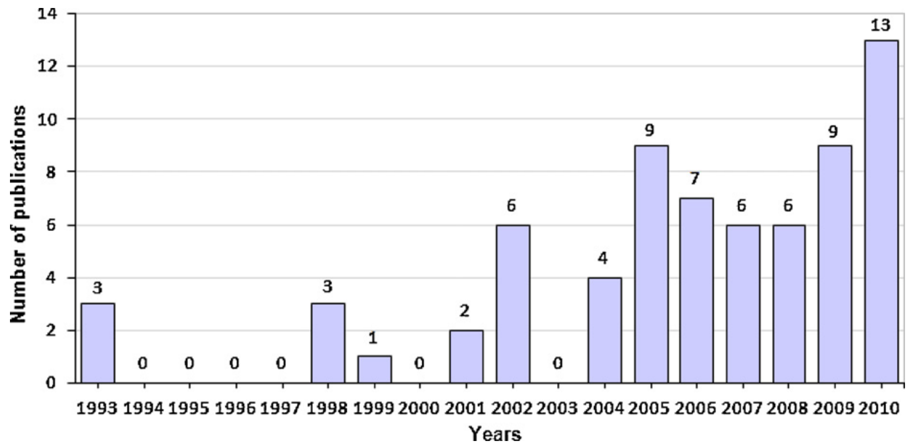

Fig. 1. Number of reviewed publications by year.

faced, the test cases and the parallel platform used in the experiments, and the reported results –, and the main contributions of the reviewed works. Each parallel ACO proposal is categorized in the new taxonomy proposed.

The manuscript is structured as follows. Next section describes the research methodology used in the review. Section 3 describes the main features of the ACO technique and briefly introduces the most popular ACO variants. Section 4 presents the generic concepts of the strategies for ACO parallelization and comments previous classification criteria. It also describes the new taxonomy proposed in this work to categorize parallel ACOs. Section 5 reviews the previous work on parallel ACO implementations and categorizes each proposal using the new taxonomy. A comparative analysis regarding the computational efficiency and quality of results is offered in Section 6. Section 7 presents the trends and perspectives in the field of parallel ACO implementations, before stating the conclusions of the survey in Section 8.

2. Methodology

The research methodology used in the review involved searching and reviewing papers from soft computing/computational intelligence conferences, journals, and books, where the application of parallel processing techniques to ACO have been proposed.

2.1. Sources and search methods

A comprehensive search was performed in conferences, journals and books about metaheuristics and parallelism. The databases searched in this study include ScienceDirect, Scopus, Thomson Reuters (formerly ISI) Web of Knowledge, ACM Digital Library, IEEE Explore, Elsevier, SpringerLink, Citeseer, as well as many others Open Access Publishing databases. The reviewed papers come out from leading conferences and journals about soft computing, such as International Conference on Parallel Problem Solving from Nature, Conference on Genetic and Evolutionary Computation, Conference on Evolutionary Computation, Journal of Heuristics, Information Sciences, Lecture Notes in Computing Science, IEEE Transactions on Evolutionary Computation, Applied Soft Computing, Future Generation Computer Systems, and Journal of Artificial Intelligence Research, among others.

The search of related papers in specific databases was done using a group of keywords that include ant colony optimization, parallel, distributed, parallelism, and soft computing. Additionally, the reference section of each paper found was reviewed to locate additional studies of interest. As a result, the final references consist of **69** papers: 19 published in journals, 44 in referred conferences, 3 in books, and 3 M.Sc./Ph.D thesis. The analysis of related works was mainly focused on the features of the parallel models, describing the distinctive characteristics of each parallel ACO proposal, the optimization problem tackled, the test cases and the parallel platform used in the experiments, and the efficiency and quality results reported. Each parallel ACO proposal is categorized in the new taxonomy proposed in this work. In order to study the recent contributions about parallel ant colony optimization, those papers published in the last five years (2005–2010) were further studied to analyze the main trends and perspectives about parallel ACO implementations.

2.2. Scope

The review focuses in papers that have proposed explicitly parallel implementations of ACO, disregarding those proposals using implicit parallelism or a distributed-agent-based search. A complete description of the implicit-parallel and other ACO categories that have been left apart from the review are summarized in Section 4.3.

The reviewed works face optimization problems from a large spectrum of application domains. Only single-objective, static optimization problems are covered in this survey, since they are the large class of problems frequently solved using parallel ACO. The algorithmic structure of ACO to solve multi-objective and dynamic optimization problems is different to the traditional ACO algorithm, so they have not been included in the scope of this review

3. Ant colony optimization

Ant Colony Optimization [45] is a population-based metaheuristic for solving optimization problems, originally proposed by Dorigo and Di Caro [41]. ACO uses artificial ants to construct solutions by incrementally adding components that are chosen considering heuristic information of the problem and pheromone trails that reflect the acquired search experience.

Algorithm 1 presents the skeleton of an ACO algorithm applied to a combinatorial optimization problem for minimizing one objective function. At first, the ACO sets the initial pheromone trails values (T) and the heuristic value of the solution components (H, known as *visibility*). After that, the algorithm iterates until a given stop condition is reached. Every iteration step is divided in four stages. First, each ant of the colony concurrently, independently, and asynchronously constructs a solution by selecting components using a probabilistic rule that considers both the experience acquired during the search (through the trace of pheromone deposited) and heuristic information of the considered components