Knowledge-Based Systems 105 (2016) 96–106

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

An efficient ant colony optimization strategy for the resolution of multi-class queries

Kamil Krynicki[a,*], Michael E. Houle[b], Javier Jaen[a]

[a] ISSI Research Group, Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, Camino de Vera, s/n, Valencia 46022, Spain
[b] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

ARTICLE INFO

Article history:
Received 20 July 2015
Revised 27 April 2016
Accepted 7 May 2016
Available online 9 May 2016

Keywords:
Ant colony optimization
Multi-class queries
Resource queries
Multipheromone

ABSTRACT

Ant Colony Optimization is a bio-inspired computational technique for establishing optimal paths in graphs. It has been successfully adapted to solve many classical computational problems, with considerable results. Nevertheless, the attempts to apply ACO to the question of multidimensional problems and multi-class resource querying have been somewhat limited. They suffer from either severely decreased efficiency or low scalability, and are usually static, custom-made solutions with only one particular use. In this paper we employ Angry Ant Framework, a multipheromone variant of Ant Colony System that surpasses its predecessor in terms of convergence quality, to the question of multi-class resource queries. To the best of the authors knowledge it is the only algorithm capable of dynamically creating and pruning pheromone levels, which we refer to as *dynamic pheromone stratification*. In a series of experiments we verify that, due to this pheromone level flexibility, Angry Ant Framework, as well as our improvement of it called Entropic Angry Ant Framework, have significantly more potential for handling multi-class resource queries than their single pheromone counterpart. Most notably, the tight coupling between pheromone and resource classes enables convergence that is both better in quality and more stable, while maintaining a sublinear cost.

1. Introduction

The efficient resolution of resource queries in networks is an abstract mathematical problem with a multitude of real-life interpretations and uses: from task dispatching in computer grids to document search in peer-to-peer networks and routing of a parcel drop-off vans through an urban neighborhood. In mathematical terms a resource query q in a graph is defined as an act of establishing a path linking a network node r_0, lacking resources, with a node or a chain of i nodes $(r_1, r_2, \dots, r_i)$ that provide them. Resources are transported back to the query emitting node over this newly established path. The path can be reutilized in the future, when a query of a similar nature is launched in the vicinity of r_0. The collecting of resources can be either destructive or not, meaning that it can consume the stock or just simply clone it.

If the resources are indistinguishable between each other (in other words, they belong to one class) we speak of single resource class queries, or *single class queries*. Such queries consist of simply gathering a number of resources at the highest possible goodness. A generalization of a query in a single resource class environment is a query in a multiple resource class environment; *multi-class query* for short. If we assume a number of $|C|$ different resource classes, distinguishable between each other, a multi-class query is defined as a resource query limited to a selected resource class c, written $q(c)$. In this case, the objective is to collect, as efficiently as possible, resources belonging to c only.

Given the underlying graph structure of this problem and given that the problem-solving strategy requires a search process over the problem space, there have been attempts in the past to solve it by using Ant Colony Optimization (ACO) [1]. ACO is a computational technique for a broad class of algorithms that perform path searches in graphs, in both, practical [2–5] and theoretical problems [6–8]. This technique is based on observations in the field of entomology revealing that a group of relatively simple organisms is often capable of completing complex tasks in a surprisingly efficient manner. A broad, generic term for this phenomenon is *emergent* or *swarm intelligence* [9] and the most prominent and well-known examples of it include the social behavior of ants and bees.

Despite the unquestionable success of ACO, with its numerous and diverse applications [10–12], it is well-known that resource querying involving multi-class resources [13], dynamic resource re-

* Corresponding author. Fax: +34 963 87 7359.
E-mail addresses: kkrynicki@dsic.upv.es (K. Krynicki), meh@nii.ac.jp (M.E. Houle), fjaen@upv.es (J. Jaen).

http://dx.doi.org/10.1016/j.knosys.2016.05.009

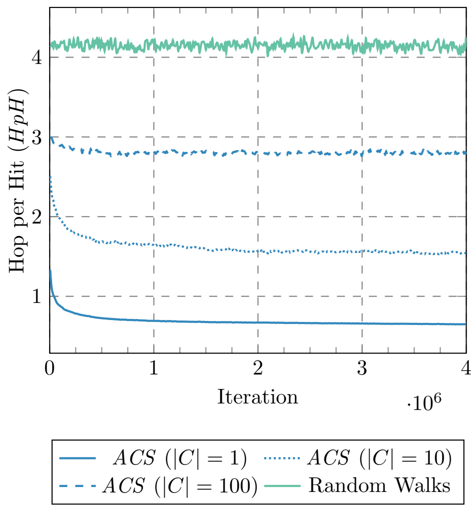

Fig. 1. *ACS* quality loss in function of |*C*|.

distribution [3] and evolving graphs [14] have a less natural representation within its metaheurisitc, which manifests itself in reduced efficiency. The main reason behind the difficulties of modeling with ACO is that it entails a strict limitation, which reflects the constrained nature of real-life ants' communication capabilities [15]. Ants leave trails of pheromone, connecting the colony to locations of interest. The pheromone level can be seen only as a general *goodness* of what is at the end of the trail, without specifying directly what it is, nor the type of resource it might be. Therefore, ACO search agents possess only a rough indication of a good direction to follow. A problem arises if the model in question includes various types of objectives. Consider the example of two ants in search for two different types of food. Each one aware of its task, but limited in its search to following high pheromone trails. The trail is essentially unable to guarantee that it leads towards precisely what the ant searches for. Ants searching for the food type *A* might end up following trails that lead to good type *B*-nodes (and vice versa); yielding a long search without results. A single pheromone value is not sufficient to guarantee efficient routing in models with ontology- or taxonomy-based resource classifications.

In a multi-class query environment the basic *ACO* single pheromone model proves insufficient. If there exist $|C| > 1$ resource classes, the probability that a generic pheromone trail leads to a high quality node for all the possible classes decreases rapidly. In such cases the pheromone values are likely to finish the convergence in a suboptimal state (*suboptimal convergence*). The higher the value of $|C|$ the more probable it is that the full pheromone trail convergence is never achieved. The system enters a state called *pheromone thrashing*, a situation in which agents with different queries continually change the pheromone trails back and forth. No coherent information can be concluded from the pheromone state and the model falls back to a near-random walks.

To illustrate this problem, in Fig. 1 we show the efficiency decline of multi-class queries with the use of *Ant Colony System* (ACS) [16], a well-known and highly acclaimed ACO implementation. If only one class of resources exists ($|C| = 1$) the graph is smooth and the convergence is quick and stable. Already at $|C| = 10$ the algorithm performs substantially worse. Even though convergence is achieved, it is suboptimal: about 145% worse than with $|C| = 1$. With $|C| = 100$ the result is only marginally better than the one obtained by the Random Walks algorithm, which we include for comparison. At $|C| = 100$ the Hop-per-Hit (*HpH*) values are 280% worse than the $|C| = 1$ baseline. In addition, the $|C| = 100$ is much less smooth than the two other due to strong pheromone thrashing. The topic of quality decline of the convergence of ACO algorithms in various scenarios is explored in depth in [13,14].

Our motivation to propose a new solution to this problem is based on the fact that the currently available approaches, as it will be discussed later, often rely on ineffective static multipheromone techniques. These existing works present several drawbacks, such as, requiring prior knowledge of the taxonomy of resources, inability to cope with dynamic problems in which resources are added or removed during the algorithm's execution, performance degradation as a result of having a fixed number of pheromone levels, as well as very low scalability with size and complexity of the problem-space.

In order to efficiently solve the distributed multi-class resource querying problem, we propose the application of a dynamic multipheromone approach *Angry Ant Framework* (AAF). AAF algorithm has already proven its efficiency in the context of single-resource class queries [17], surpassing ACS, one of the most popular ACO implementations. AAF improves the path quality of ACS by a significant margin, while at the same time decreasing the chance that the algorithm converges to a local minimum. This previous work is improved here by proposing a new extension to AAF, called EntropicAAF. The main innovations of this new approach are manifold. First, we show that it is possible to construct an effective ACO strategy to solve multi-class resource searching problems. Second, our approach avoids the previously mentioned problems of other existing static multi-pheromone approaches. Third, our algorithm is able to find, based on entropic information, the best pheromone level for a given ant at each moment of the computation process. Fourth, in terms of search efficiency, our results demonstrate an average 68% improvement over existing ACS and AAF in randomly generated problem domains consisting of up to 100 classes of resources. Fifth, another contribution of this work is to show that EntropicAAF is able to cope both with problem spaces in which well differentiated clusters of resources exist and with problem spaces in which there are overlapping regions where different categories of resources are present. Finally, our performance analysis reveals that the new proposed algorithm is effective in terms of memory and CPU consumption.

In summary, the strongest suite of AAF is that it allows dynamic pheromone stratification, that is, creating ad-hoc multipheromone models, which adapt themselves to the resource ontology, ant traffic and query distribution. The results obtained open a new area of research in which a computational model based on entropy levels of the underlying pheromone state can be applied to the efficient resolution of multi-class resource queries.

The remainder of this paper is organized as follows. In Section 2 we expand on the concepts of single class and multi-class queries in the context of ACO-based queries in graphs and summarize the related works. Section 3 contains the outline of the mathematical model of AAF. The experimental results are presented in Section 4. We conclude with a discussion on our findings in Section 5.

2. Related works

The problem of multi-class queries has only received a moderate attention. In [18] Valdez et al. resolve a problem of a similar nature, which they refer to as Semantic Query Routing problem (SQRP). As in multi-class queries, in SQRP the class of the query impacts the routing of the search agent, which has the task of determining the shortest path from the query issuing node to the node with appropriate resources. Their single-pheromone algorithmic solution to SQRP, called Neighboring-Ant Search (NAS), is reported as an order-of-magnitude improvement over Random Walks, however no comparison with more elaborate techniques is presented. Moreover, NAS uses lookahead techniques that are out-