Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 18 (2013) 611 – 620

Procedia Computer Science

International Conference on Computational Science, ICCS 2013

Multi-agent distributed framework for swarm intelligence

Sorin Ilie[a], Costin Bădică[a]

[a] University of Craiova, Software Engineering Department, Bvd. Decebal 107, Craiova 200440, Romania

Abstract

This paper presents a multi-agent distributed framework for Swarm Intelligence (SI) based on our previous work ACODA (Ant Colony Optimization on a Distributed Architecture). Our framework can be used to distribute SI algorithms for solving graph search problems on a computer network. Examples and experimental results are given for SI algorithms of: Ant Colony System (ACS) and Bee Colony Optimization (BCO). In order to use the framework, the SI algorithms must be conceptualized to take advantage of the inherent parallelism determined by their analogy with natural phenomena (biological, chemical, physical, etc.): (i) the physical environment of the swarm entities is represented as a distributed multi-agent system and (ii) entities' movement in the physical environment is represented as messages exchanged asynchronously between the agents of the problem environment. We present initial experimental results that show that our framework is scalable. We then compare the results of the distributed implementations of BCO and ACS algorithms using our framework. The conclusion was that our approach scales better when implementing the ACS algorithm but is faster when implementing BCO.

Keywords: Distributed Applications; Swarm Intelligence; Artificial Intelligence; Ant Colony Optimization; Bee Colony Optimization

1. Introduction

This paper introduces a new distributed framework for a class of Swarm Intelligence algorithms that better exploits the inherently distributed nature of these algorithms.

Swarm Intelligence (SI hereafter) "is the emergence of coherent functional global patterns from the collective behaviors of entities interacting locally with their environment" [1].

SI approaches are usually applied to solve computationally complex problems, such as NP-hard problems. There is a certain level of abstraction at which the solving of such problems can be modeled by distributed computational systems composed of interacting artificial entities. Thus we expect distributed computing, including distributed multi-agent middleware, to have great potential for the application of SI computational approaches.

The framework presented here is an extension of our ACODA (Ant Colony Optimization on a Distributed Architecture) which was initially designed to support only Ant Colony Optimization (ACO). Initial experimental results on the scalability of ACODA were already presented in [2].

In fact our framework can be applied to a class of SI algorithms with the condition of abstracting the process of searching the solution space as the movement of computational swarm entities into their distributed computational

*Corresponding author. Tel. and fax: +40-251-438.198.
E-mail address: cbadica@software.ucv.ro.

Selection and peer review under responsibility of the organizers of the 2013 International Conference on Computational Science
doi:10.1016/j.procs.2013.05.225

environment, similarly with the movement of natural entities (like birds, ants, or bees) in their physical environment. During their movement process, these entities save partial solutions as well as other relevant information onto the environment, sense the environment and exchange information via the environment, thus achieving the continuous improvement of partial solutions into gradually better solutions. Typical examples of SI approaches suitable for our framework are ACO and Bee Colony (BC). In particular, we considered here the algorithms of Ant Colony System (ACS) presented [3] in chapter 3 and Bee Colony Optimization (BCO) presented in [4].

In this paper we present our framework and show how it can be used to implement distributed ACO and BC. Moreover, we updated the initial design of the framework presented in our previous work [2] in order to accommodate the implementation of more types of SI algorithms and problems.

Summarizing, the main contributions of this paper can be outlined as follows:

(i) We generalized our ACODA distributed framework, initially designed only for ACO algorithms, to other types of SI algorithms. The novelty is the new modeling of the SI environment as a multi-agent system and conceptualizing the mobility of swarm entity as messages exchanged by agents.

(ii) We showed that the new framework supports two different types of SI: ACS [3] and BCO [4].

(iii) We performed an experimental comparison of the implementations of BCO and ACS for solving the Traveling Salesman Problem (TSP hereafter) on our framework. In our experiments with BCO we obtained the best value of efficiency (equation (1)) that we found in literature.

The paper is structured as follows. In section 2 we introduce *graph search problems* that are targeted by our framework and we give an overview of ACS and BCO. In section 3 we present our literature review on distributed SI. Section 4 presents our framework. While in section 5 we show how ACS and BCO can be implemented on it. In section VI we present our experiments and in section 7 we present our conclusions and future work.

2. Background

Before presenting our architecture design, we state the target problem and introduce the SI approaches that we used as case studies. Due to lack of space we will not detail the mathematical engines of the implemented SI algorithms, as they are known and can be found in in [3] (ACS) and in [4] (BCO).

SI algorithms are well suited for graph search problems, so our framework targets this type of problems. A *graph search problem* can be formalized as optimizing a real-valued function defined over a subset of graph paths – i.e. paths restricted by specific problem-dependent constraints. For the TSP this subset contains all the Hamiltonian cycles [5] of the graph – i.e. paths containing all vertices such that the source and the destination are the same, while any other two vertices of the path are distinct. Other notable examples of graph search problems include: graph flow problems [5], generalized TSP [6], pickup and delivery problem [7], etc.

2.1. Ant Colony Optimization

ACO [3] refers to a family of SI optimization algorithms that get their inspiration from the metaphor of real ants searching for food. During their searching process, ants secrete pheromone to mark their way back to the anthill. Other colony members sense the pheromone and become attracted by marked paths; the more pheromone is deposited on a path, the more attractive that path becomes.

The pheromone is volatile so it disappears over time. Evaporation erases pheromone on longer paths as well as on paths that are not of interest anymore. However, shorter paths are more quickly refreshed, thus having the chance of being more frequently explored. Intuitively, ants will converge towards the most efficient path, as that path gets the strongest concentration of pheromone.

In the ACO approach to SI, artificial ants are programmed to mimic the behavior of real ants while searching for food. The ants' environment is modeled as a graph while the path to the food becomes the solution to a given graph search problem. Artificial ants originate from the anthills that are vertices of the graph and travel between vertices to find optimal paths, following ACO rules. When a solution path is found, ants mark its edges with pheromone by retracing the path.

We mapped the ACS algorithm [3] to our framework by configuring it with the parameters recommended by the authors. In ACS an ant chooses the next edge to follow towards an unvisited vertex with a probability that increases with the quantity of pheromone deposited on that edge.