DISTRIBUTION OF AGENT BASED SIMULATION WITH COLORED ANT ALGORITHM

Cyrille Bertelle, Antoine Dutot, Frédéric Guinand, Damien Olivier

DISTRIBUTION OF AGENT BASED SIMULATION WITH COLORED ANT ALGORITHM

Cyrille Bertelle, Antoine Dutot
Frédéric Guinand and Damien Olivier
LIH - Le Havre University
25 Rue Philippe Lebon - BP 540
76058 Le Havre cedex, France
E-mail: {Cyrille.Bertelle, Antoine.Dutot,
Frederic.Guinand, Damien.Olivier}@univ-lehavre.fr

KEYWORDS

Distributed Computing, Ant Algorithm, Simulation, Migration, Colored Pheromones.

ABSTRACT

In distributed simulations, many entities (mobile agents, distributed objects, processes) communicate and may evolve continuously. Volumes of communications, groups of communicating entities, as well as computational needs may vary a lot during the execution. Due to these dynamic characteristics, the need of a migration for one or several entities is generally difficult to evaluate. A dynamic communication graph constitutes the model for the multiagent system. The proposed method evaluates the traditional trade-off between communication overhead and load-balancing by identifying clusters of highly communicating entities. A variant of classical ant algorithms was designed for that purpose. In our implementation pheromones are colored. The vertices of the communication graph are colored according to the effective communication between each other. A change of color for a vertex informs that a migration of the corresponding entity on the corresponding processor should be beneficial.

1. INTRODUCTION

Nature presents many examples of complex systems with a sophisticated global evolution while entities composing it have simple but evolving behaviors. One way of tackling the problem of simulating such systems consists in identifying each kind of entity and in modeling its main characteristics and their evolution: capabilities, behavior, interactions with the environment and interactions with other entities. This may require many entities, behaving independently and evolving at their own speed in an asynchronous way. So distributed agent based systems is the more suitable architectures for implementing such applications. Even if performance is not the main reason for choosing distributed systems, some problems have to be addressed in order to avoid major perturbations that might affect the execution of the application. Among these problems, we are interested in the management of the communication traffic. Indeed, if the communications are not regulated, the execution of highly communicating groups of entities, composing interesting emerging structures, may be perturbated.

The paper is organized as follows : Section 2 presents the graph model. In section 3, we present some classical ant algorithms for graph problems. In section 4, we define the general method called colored ant system. In section 5, we describe the experimentations and present some results. Finally, the last section concludes the work and some perspectives are discussed.

2. DYNAMIC COMMUNICATION GRAPH MODEL

We study the previously described problem from the agent migration point of view. We call *effective communications*, communications between agents located on distinct processors.

The communications between agents are model with a graph $G = (V, E)$ where V is the set of vertices corresponding to the entities and E is the set of edges. Each edge $e = (v_1, v_2)$ represents the communications (whatever the direction) between agents associated to vertices v_1 and v_2. The volume of data exchanged between both agents is the label of edge $e = (v_1, v_2)$.

During execution, communications between agents may vary, may start or may stop. Such events change the numerical parameters of the graph, but even more important, they may change the structure of the graph.

In this context, the only way of limiting the volume of effective communications consists in gathering highly communicating agents on the same processor, which can

be achieved by a migration operation. However to avoid the migration of all the agents on a single processor, this criterion has to be coupled with a load balancing criterion.

The originality of this work consist in performing both graph partitioning (clusters of highly communicating entities) and load balancing for a dynamic application. The main goal of our work consists in identifying groups of highly communicating entities for giving some indication about the relevance of a migration operation. Moreover, this information may also be profitable for some simulation application in which multi-scales phenomena appear (Tranouez et al., 2001).

Two approaches can be considered. The first consists in applying classical mapping strategies for static deterministic graphs by considering, periodically, a snapshot of the application graph. But the fact that the graph may change just after the snapshot, making the future mapping out-of-date before its computation, and the fact that the solution may be not reusable for the next snapshot are the two main drawbacks of this approach. The second approach considers the application graph as a changing environment. We want an anytime solution for the agents allocation. For that purpose, the allocation is continuously computed, and a change in the structure or in the numerical parameters is reported in the graph and is taken into account for the computation. One way of performing such computation consists in deploying computing elements. They identify relations between vertices by analyzing the traffic. Ant algorithms are well suited for that task as it has been shown in (Dorigo et al., 1996) or in (Caro and Dorigo, 1997; Schoonderwoerd et al., 1997; Heusse et al., 1998; Bonabeau et al., 1998) for the adaptive routing problem in communications networks.

3. ANT ALGORITHMS FOR GRAPH

Ant algorithms are based on natural ants behaviour. Experimental studies show (Gordon, 1995) that natural ants continously forage their territories to search food. Myrmecologists have also observed that very complex tasks can be achieved by cooperation between ants: bridge construction, nest edification... This kind of collective behavior emerges from the interaction between ants as shown by Langton (Langton, 1987).

The principle of ant algorithms consists in exploiting the ability of ants to find near optimal solutions, for difficult problems. The intelligent collective behavior arises from their interactions based on indirect communications known as stigmergy. Sematectonic stigmergy produces changes in the physical environment, construction of the nest for instance. Stigmergy based on signal uses the environment as support. The resolution process is based on this volatile chemical signal, a.k.a. pheromone. This approach provide robust solutions for a problem with parameters changes. It is intrinsically distributed and scalable and uses only local information for computing high-quality global solutions. These characteristics constitute the main quality of this approach.

Ant-based algorithms have been succesfully applied to various combinatorial optimization problems like the Traveling Salesman Problem (Dorigo and Gambardella, 1997) or routing in networks (Caro and Dorigo, 1997; Schoonderwoerd et al., 1997; Bonabeau et al., 1998; Heusse et al., 1998).

4. COLORED ANT SYSTEM (CAS)

Our work concerns large-scale simulations on distributed systems. These application can be modeled as a dynamic communication graph where each vertex is associated to one agent and where each edge represents the volume of communications between a couple of agents. The idea is to use an ant-based approach for the detection of clusters of communicating agents. Morever, the allocation needs a load balancing criterion. For that purpose, we have introduce the notion of *colored pheromones*.

Definition 1 (Dynamic communication colored graph) *A* dynamic communication colored graph *is a weighted undirected graph $G = (V, E, C)$ such that:*

- *C is a set of p colors where p is the number of processors of the distributed system.*
- *V is the set of vertices. Each vertex has a color belonging to C.*
- *E is the set of edges. Each edge is labelled with a weight. A weight $w(u, v) \in \mathbb{R}^+$ associated to an edge (u, v) corresponds to a volume of communications between the couple of agents corresponding to vertices u and v.*

On the example figure 1, as further explained in section 5, the application is made of 12 agents and is distributed on a four-processors system. Each vertex is colored (from light gray to dark). Vertices of the same color corresponds to agents located on the same processor.

The method we conceived proposes to change the color of some vertices if these changes lead to an improvement of the allocation. During the execution, a vertex may change its color several times, depending mainly on the variations of the data exchanges between agents.

We describe now the *Colored Ant System (CAS) Algorithm for the Dynamic Distribution* which is inspired by Ant System (Dorigo et al., 1996). We consider a dynamic communication colored graph $G = (V, E, C)$ (see definition 1). For each color $c \in C$ a set of n ants is created ($n \gg p$).Thus $n \times p$ ants are used by our method. $\mathcal{F}$ denotes the set of all ants.

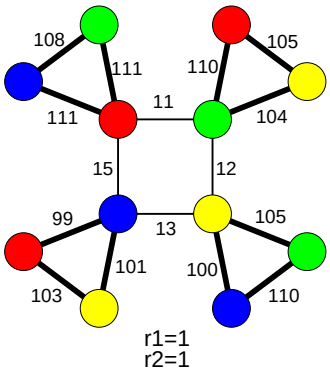


Figure 1: An exemple of dynamic communication colored graph.

1. Initially, ants are uniformly distributed among the vertices. The color of each ant is fixed by its initial vertex.

2. The algorithm is based on an iterative process. Between steps $t-1$ and t, each ant cross one edge and reach a new vertex. During its move, it drops pheromone of its color, on the edge crossed. Moreover, each ant has the ability to memorize the last vertex crossed.

   We define the following positive numbers:

   - The quantity of pheromone of color c dropped by the ant x on the edge (u,v), between the steps $t-1$ and t, is noted $\Delta_x(u,v,c)$.
   - The quantity of pheromone of color c dropped by ants on the edge (u,v), between the steps $t-1$ and t is noted
   $$\Delta(u,v,c) = \sum_{x \in \mathcal{F}} \Delta_x(u,v,c) \quad (1)$$
   - The total quantity of pheromone of all colors dropped by ants on the edge (u,v), between the steps $t-1$ and t is noted
   $$\Delta(u,v) = \sum_{c \in C} \Delta(u,v,c) \quad (2)$$
   - if $\Delta(u,v) \neq 0$, the rate of pheromone of color c on the edge (u,v) between the steps $t-1$ and t is noted
   $$K_c(u,v) = \frac{\Delta(u,v,c)}{\Delta(u,v)} \quad (3)$$
   This rate verifies $K_c(u,v) \in [0,1]$.

3. The current quantity of pheromone of color c present on the edge (u,v) at step t is denoted by $\tau^{(t)}(u,v,c)$. Its initial value (when $t=0$) is 0 and then it is computed following the recurrent equations:

   - if $\Delta(u,v) \neq 0$,
   $$\begin{aligned}\tau^{(t)}(u,v,c) &= \rho K_c^{\gamma}(u,v)\tau^{(t-1)}(u,v,c) \\ &\quad + K_c(u,v)\Delta_x(u,v,c) \quad (4)\end{aligned}$$
   - if $\Delta(u,v) = 0$,
   $$\tau^{(t)}(u,v,c) = \rho\tau^{(t-1)}(u,v,c) \quad (5)$$

   $\rho \in [0,1]$ represents the pheromone persistence due to its evaporation. $K_c(u,v)$ and γ represent repulsion factors. These values decrease the quantity of pheromone when many ants of other colors have crossed the arc. Thus $\rho K_c^{\gamma}(u,v)$ depends on the quantity of the other colors and contributes to the evaporation of the c pheromone.

4. Let us define $p(u,v_k,c)$ the transition probability of an edge (u,v_k) incident to vertex u for an ant of color c and whose communication volume is noted $w(u,v_k)$.

   - At the initial step ($t=0$),
   $$p(u,v_k,c) = \frac{w(u,v_k)}{\sum_{v \in \mathcal{V}_u} w(u,v)} \quad (6)$$
   - After this initial step ($t \neq 0$),
   $$p(u,v_k,c) = \frac{(\tau^{(t)}(u,v_k,c))^{\alpha}(w(u,v_k))^{\beta}}{\sum_{v_q \in \mathcal{V}_u}(\tau^{(t)}(u,v_q,c))^{\alpha}(w(u,v_q))^{\beta}} \quad (7)$$

   where $\mathcal{V}_u$ is the set of vertices adjacent to u.

   The relative values of α and β give the ponderation between pheromone factor and communication volume. We show later in the experimental section that this ponderation is one of the major factors in the algorithm convergence.

   The choice of the next edge crossed by an ant depends on previous probabilities. However, to avoid the ants moves to oscillate between two vertices, we introduce in the probability formula, a penalisation factor $\eta \in [0,1]$. Given $\overline{v}_x$ the last visited vertex by the ant x, the new probability formula is:

   $$p_x(u,v_k,c) = \frac{(\tau^{(t)}(u,v_k,c))^{\alpha}(w(u,v_k))^{\beta}\eta_{x,k}}{\sum_{v_q \in \mathcal{V}_u}(\tau^{(t)}(u,v_q,c))^{\alpha}(w(u,v_q))^{\beta}\eta_{x,q}} \quad (8)$$

   where
   $$\eta_{x,q} = \begin{cases} 1 & \text{if} \quad v_q \neq \overline{v}_x \\ \eta & \text{if} \quad v_q = \overline{v}_x \end{cases} \quad (9)$$

5. The color of a vertex u, noted $\xi(u)$ is obtained from the main color of its incident arcs:

$$\xi(u) = \arg\max_{c \in C} \sum_{v \in \mathcal{V}_u} \tau^{(t)}(u, v, c) \qquad (10)$$

5. EXPERIMENTATION

The model has been implemented and used to determine valid numerical values for its parameters. In the following, we give the parameters used to tune the behaviour of the CAS:

α The relative importance of pheromone in probability formula (8).

β The relative importance of weight in probability formula (8).

ρ The pheromone persistence on an edge.

η The penalisation factor that retains an ant from returning on the vertex it just left.

In these experiments, the parameter γ is held to 1.

The initial parameters, number of ants, ants per color, etc. are set this way:

- An equal number of ants is allocated to each color. As colors represent computing resources (processors), this repartition gives an equal importance to each resource. Changing it would give more power to one or more computing resources compared to others.
- Then among these ants, an equal number of them is allocated to each vertex of a given color.
- Initially, for each vertex, the number of ants, allocated to it, must be equal or greater than its degree (i.e. its number of connected edges).

The validation tests are computed on several graphs defined to form clusters of highly communicating agents, while these clusters are linked to the others by low communication edges (see figure 1).

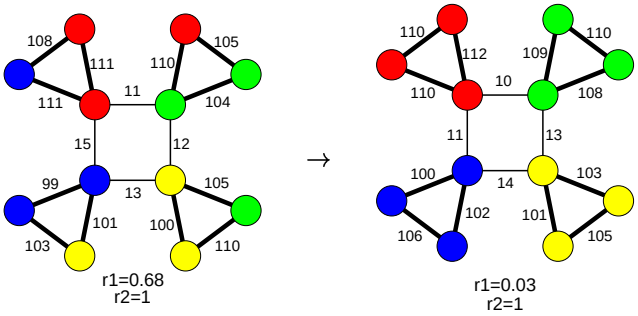
We defined two methods to express the quality of solutions:

- We use communication graphs where clusters are already identified. These graphs are randomly perturbed in term of color allocation. Then they are submitted as initial configuration for the algorithm which tries to find the initial allocation on the graph, as a solution.
- We compute global communication costs, noted e, by summing effective communications on the graph (between agents located on distinct processors i.e. allocated to different colors, as defined in the beginning of the section 2). Then we compute a ratio r_1 among the total communication, noted s, on the graph ($r_1 = e/s$).

We have a second criterion, noted r_2, measuring the load balancing: for each color c, we have v_c the number of vertices having color c and p_c the power of processor affected to c. Then we have

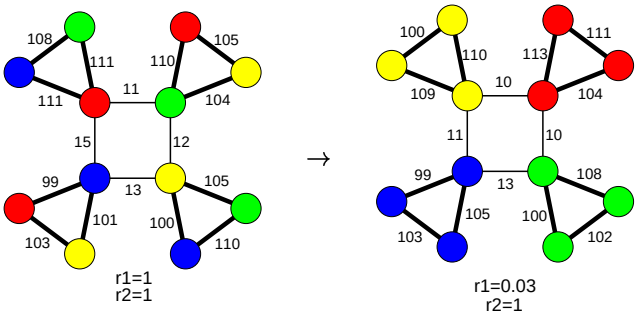
$$r_2 = \frac{min\mathcal{E}}{max\mathcal{E}} \quad \text{where} \quad \mathcal{E} = \left\{ \frac{v_c}{p_c}; c \in C \right\}$$

We first investigate a dynamic graph with four colors where four clusters should appear. Initially, each of them has a predominant color, and one vertex of another color, as shown under:



The CAS algorithm finds clusters after five cycles, using 100 ants, and the following parameters: $\{\alpha = 1, \beta = 5, \rho = 0.8, \eta = 0.01\}$.

Another test uses a similar dynamic graph and the same number of ants, but with a different initial configuration where each vertex in the four cluster has a distinct color, as shown under:

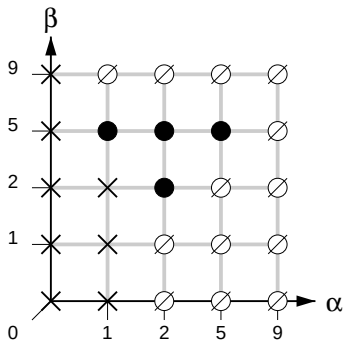


Here the optimal parameters where: $\{\alpha = 1, \beta = 5, \rho = 0.3, \eta = 0.0001\}$ (notice ρ and η), and the solution was found after seven to height steps.

The algorithm becomes very sensitive to parameter settings when graphs are initially very perturbed like the second example shown above. However, a graph already correctly colored (that is where clusters of highly communicating entities are already of the same color) is almost never changed by the CAS algorithm.

The following diagram may help understand the relative importance of the pheromone trail and edge weight. The diagram characterizes the algorithm convergence in function of

the value of α parameter (which grows with relative importance of pheromone trail) and in function of the β parameter (which grows with relative importance of edge weight). In this diagram, a $\times$ means the algorithm did not found a good enough solution, and did not stabilized. A $\emptyset$ means the algorithm did not found a good enough solution, but stagnated. Finally a $\bullet$ means the algorithm found a good solution, and stabilized on it.



6. CONCLUSION AND PERSPECTIVES

This paper presents a variant of Ant System, using multi-colored pheromones and called Colored Ant System. The aim of this method is to solve the allocation problem in dynamic agent-based simulation, respecting some load balancing aspects. First validations are presented and some numerics values have been computed to estimate the solution quality. This approach offers to give advices for agent migration. We study actually a model for the coupling and the feed-back on multi-agent simulations, respecting some of their specific constraints and taking advantages of these advices.

REFERENCES

Bonabeau, E., Henaux, F., Guérin, S., Snyers, D., Kuntz, P., and Théraulaz, G. (1998). Routing in telecommunications networks with "smart" ant-like agents. In *Proceedings of Intelligent Agents for Telecommunications Applications'98*.

Caro, G. D. and Dorigo, M. (1997). Antnet: A mobile agents approach to adaptive routing. Technical report, IRIDIA, Université libre de Bruxelles, Belgium.

Dorigo, M. and Gambardella, L. (1997). Ant colony system: A cooperative learning approach to the traveling saleman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66.

Dorigo, M., Maniezzo, V., and Colorni, A. (1996). The ant system: optimization by a colony of cooperating agents. *IEEE Trans. Systems Man Cybernet.*, 26:29–41.

Gordon, D. (1995). The expandable network of ant exploration. *Animal Behaviour*, 50:995–1007.

Heusse, M., Snyers, D., Guérin, S., and Kuntz, P. (1998). Adaptive agent-driven routing and load balancing in communication networks. In *Proceedings of the 1st International Workshop on Ant Colony Optimization, Oct. 15-16, Brussels, Belgium*.

Langton, C., editor (1987). *Artificial Life*. Addison Wesley.

Schoonderwoerd, R., Holland, O. E., and Bruten, J. L. (1997). Ant-like agents for load balancing in telecommunications networks. In *Proceedings of the 1st ACM International Conference on Autonomous Agents, Feb. 5-8, Marina del Rey, CA, US*, pages 209–216.

Tranouez, P., Bertelle, C., and Olivier, D. (2001). Changing the level of description of a fluid flow in a agent-based simulation. In *ESS 2001 Conference*, Marseilles (France).