Université Libre de Bruxelles
Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle

An Ant-Based Algorithm for the Heterogeneous Dynamic Task Allocation Problem

Roberto Ghizzioli, Shervin Nouyan, Mauro Birattari, and Marco Dorigo

IRIDIA – Technical Report Series

Technical Report No.
TR/IRIDIA/2005-005

March 2005

IRIDIA – Technical Report Series
ISSN 1781-3794

Published by:

IRIDIA, *Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle*
Université Libre de Bruxelles
Av F. D. Roosevelt 50, CP 194/6
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2005-005

An Ant-Based Algorithm for the Heterogeneous Dynamic Task Allocation Problem

Roberto Ghizzioli, Shervin Nouyan, Mauro Birattari, and Marco Dorigo

IRIDIA, Université Libre de Bruxelles
Brussels, Belgium

rghizzioli@iridia.ulb.ac.be, {snouyan,mbiro,mdorigo}@ulb.ac.be

Abstract

An ant-based algorithm inspired by the division of labor in ant colonies is proposed and applied to the the solution of an online scheduling problem. A painting facility is considered for illustrating the problem: Trucks leave an assembly line to get painted in painting booths. The goal is to minimize the makespan, that is, the time needed for painting all given trucks. In this paper we propose an extension of the dynamic threshold model of Theraulaz *et al.* [8] which is designed for handling the case in which all booths are identical. In our extension we address the *heterogeneous* case in which different booths might require different processing times for completing a same task. The paper contains an empirical comparison of different insect-based algorithms for the problem discussed.

1 Introduction

Swarm intelligence techniques [1] have been successfully adopted in computer science, engineering, and operations research. For example, ant colony optimization [4] performs as well as, or even better than state-of-the-art combinatorial optimization techniques.

In this paper we propose an ant-based algorithm for tackling the Dynamic Task Allocation (*DTA*) problem, a factory scheduling problem in which tasks are to be allocated to processing units. In previous works, multi-agent algorithms have been developed for the *homogeneous* case, that is, the case in which all agents (processing units) are identical. In this paper we investigate the *heterogeneous* case in which agents can differ in their processing speed. Think for example of a factory with old and new machines or with different sets of machines each optimized for a different class of tasks.

Most of the previously proposed algorithms use paradigms based on the *specialization* concept: Agents tend to specialize for one type of task in order to avoid unnecessary reconfiguration. This typically increases the efficiency of the whole system. Morley [6] has solved a painting problem similar to the homogeneous version of the *DTA* problem. His market-based algorithm was adopted in a General Motors facility with a performance improvement of 10% over the previously adopted centralized scheduler. Furthermore, different insect-based algorithms have been successfully applied to the homogeneous case of the *DTA* problem [2, 3, 7]. These algorithms are inspired by the division of labor in social ants and adopt the *dynamic threshold model* proposed by Theraulaz [8]. Unfortunately, the performance of the aforementioned algorithms on the heterogeneous case is not satisfactory. The aim of this paper is to present an improved version of the threshold model which takes into account the heterogeneous processing speeds of the agents.

Section 2 presents the problem using the example of a painting facility. Section 3 discusses the original threshold model and proposes an improved version inspired by the division of labor of ant castes. Section 4 proposes an experimental analysis that highlights the performance improvement obtained by our algorithm over the ones previously presented in the literature. Section 5 concludes the paper.

2 The problem

The problem considered here is a particular non-deterministic scheduling problem that we refer to as the Dynamic Task Allocation (*DTA*) problem. In the following we describe the problem using the example of a painting facility. A formal definition is given in [5].

Trucks leave an assembly line and are assigned to a painting booth. The number of available colors is fixed and the color of each truck is predetermined by a customer order. A painting booth is an agent able to paint a truck in any available color. Booths may have different processing times for the same type of tasks. Moreover, each booth has a fixed queue length which can be filled by trucks. If the color of a painting booth must be changed, a setup time is necessary. For example, if a booth is applying red and the following truck to be processed by that painting booth requires white, a fixed flush time is required before the booth can start processing the task. If no setup is necessary, the booth starts immediately to paint the next truck in its queue. A setup may also be related to a monetary cost that could, for example, represent the danger of failure when changing a color, or the amount of paint lost during a swap. The problem consists in assigning trucks to painting booths with the goal of minimizing the makespan, that is, the completion time of the last truck in the system.

In the formalization of the *DTA* problem, the task generation process is not specified. For example, the release dates and the colors may be distributed exponentially or normally. The distributions may vary dynamically so that at a random time the probability mix changes and agents need to adapt to the new environment.

3 The threshold model

The threshold model by Theraulaz *et al.* [8] is used by the ant-based algorithms for the *DTA* problem. It describes the division of labor in ant colonies.

Biological studies show that an ant colony performs different activities *simultaneously* exploiting *specialized individuals*. Another key feature of division of labor is the *plasticity* of the colony to the environmental changes. In the following, we detail the original and our improved version of the threshold model.

3.1 Original version

Assume that m tasks need to be performed. Each task j is associated with a stimulus s_j, the level of which increases if it is not satisfied. Let us consider N agents and let $\theta_{i,j}$, with $(i = 1, ..., N)$, be the *response threshold* of agent i concerning task j. In the *DTA* problem, agents are in charge of booths and autonomously bid to paint trucks. Here a stimulus is associated to the color of a truck. Each agent i has a threshold value $\theta_{i,j}$ for each available color j.

In the dynamic threshold model, agent i engages in task j with probability

$$P(s_j, \theta_{i,j}) = \frac{s_j^2}{s_j^2 + \theta_{i,j}^2}. \tag{1}$$

This equation shows that for $s_j \ll \theta_{i,j}$, the probability that an individual i engages in task j, is close to 0 and for $s_j \gg \theta_{i,j}$, the probability is close to 1. Therefore, an agent i with a lower threshold $\theta_{i,j}$ is more likely to respond at a lower level of stimulus s_j.

Let ξ be the coefficient that describes *learning* (specialization) and φ the coefficient that describes *forgetting*. If we consider a time-incremental model, agent i specializes when performing task j during a time period of Δt by changing its threshold as follows:

$$\theta_{i,j} \to \theta_{i,j} - \xi \Delta t. \tag{2}$$

On the other hand, agent i forgets if it does not perform task j for a time period of Δt by changing its threshold as follows:

$$\theta_{i,j} \to \theta_{i,j} + \varphi \Delta t. \tag{3}$$

3.2 Improved version

The definition of the *DTA* problem considers agents that can have different processing time for different task types. A number of algorithms [2, 3, 7] based on the threshold model can be applied to the *DTA* problem. Nevertheless, they do not take into account the different processing speed of the agents, and therefore their performance is rather poor. The problem is that agents bid for a task considering only their specialization level without taking into account their own characteristics, that is, their capacity to process a task quickly or slowly. In order to have better results, we extend the threshold model so that the faster an agent can perform a task, the higher is the probability that the agent bids for it. To this aim we introduced the processing speed of agent i on task j into the probability function $P(s_j, \theta_{i,j})$. The most promising probability function is obtained modifying Equation 1 as follow:

$$P(s_j, \theta_{i,j}) = \frac{s_j^2}{s_j^2 + \theta_{i,j}^2 * (t_{proc,i,j} - tmin_{proc,j} + 1)}, \quad (4)$$

where $t_{proc,i,j}$ is the time that agent i need to process task j and $tmin_{proc,j}$ is the minimum time needed by agents in the system for processing task j. The added term serves as a weighting factor of the specialization level in the bidding process. This new probability function is inspired by the division of labor between *insect casts* observed by Wilson [9] in ant species *Pheidole*.

4 Empirical analysis

In this section different multi-agent algorithms, some of which based on the dynamic threshold model, are compared on a class of instances of the DTA problem. In the following we present the experimental setup, all the analyzed algorithms and the results.

4.1 Experimental setup

The instances in our experimental analysis model a typical working day of a painting facility. Trucks exit from the assembly line for a time of 420 minutes. We consider 12 painting booths. Each painting booth might be broken or anyway unavailable with a probability of 0.02. Each agent has a queue size equal to 5. In this case there are two subsets of agents: the first needs 3 minutes to process the first half of the available types of tasks and 9 minutes for the other half. The other subset has the opposite characteristics. For both subsets the setup time is 10 minutes.

The number of trucks exiting the assembly line is always equal to 840, which is the maximum number of trucks that 12 booths can paint considering an average process time of 6 minutes, no setups and no idle time. For this class we extract the number of colors from the probability distribution presented in Figure 1. Color types are assigned to the trucks according to the two mixes:

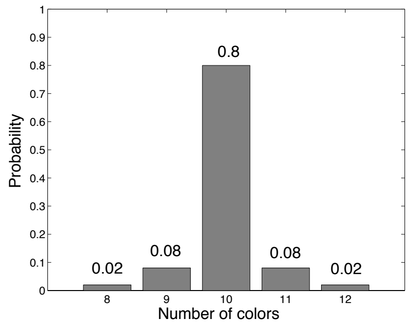

Figure 1: Probability distribution of the number of colors for the DTA experiment.

1. One part of the n colors has a higher probability than the others. Let $P(i)$ be the probability of color i. The probability $P(i)$ is defined as follows:

$$\begin{cases} \sum_{i=1}^{\lfloor n/2 \rfloor} P(i) = 3 \sum_{j=\lfloor n/2 \rfloor+1}^{n} P(j) \\ P(1) = \cdots = P(\lfloor n/2 \rfloor) \\ P(\lfloor n/2 \rfloor + 1) = \cdots = P(n) \end{cases}$$

2. like the previous one, but after 210 minutes the colors that appear more frequently appear more rarely and vice versa.

Each instance features either one mix or the other, with the same probability.

4.2 Summary of the algorithms

In this analysis we compare 4 algorithms developed for the homogeneous DTA, and already described in the literature. The first one is a market-based algorithm proposed by Morley (MBA) [6]; this algorithm is the one adopted by General Motors in one of their painting facilities. The other three algorithms use the original version of the dynamic threshold model and are respectively: the algorithms of Campos *et al.* (ABA) [2], Cicirello *et al.* (R-$WASP$) [3], and Nouyan (ATA) [7]. To evaluate the performance of the improved model, we have applied it to all the insect-based algorithms, which are then named $ABAc$, R-$WASPc$, $ATAc$. Additionally, a trivial non-adaptive *greedy* algorithm named $LOCUST$ is introduced in order to have a performance reference.

4.3 Results

We have run all the algorithms on 1000 instances of the presented class after a rigorous parameter tuning using an evolutionary algorithm. Details are given in [5]. Figure 2 summarizes the

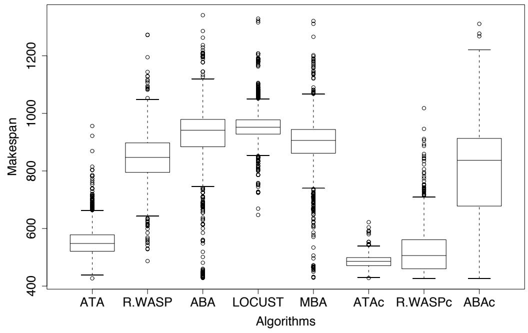

Figure 2: Box plot of the makespan values all the eight algorithms. The figure clearly shows that the improved model gives a high improvement of the performance especially to the *R-WASP* algorithm.

makespan values achieved by each algorithm. *ATAc* obtains the best results concerning the makespan and is very close to the makespan lower bound (420 minutes). On the basis of a paired Wilcoxon test ($\alpha = .05$) we can state that: i) the adaptive algorithms perform *significantly* better than the non-adaptive algorithm *LOCUST*. ii) The algorithms that use the improved model perform *significantly* better than the respective algorithms that use the original one. *R-WASPc* is the algorithm that benefits the most from the improvement.

5 Conclusion

In this paper we have presented an ant-based algorithm for the solution of a scheduling problem in which tasks have to be assigned to processing units. The algorithm is an extension of previously presented methods based on the *dynamic threshold model* and it is specially designed for the case in which the processing units are not homogeneous. Further informations about the algorithm, the procedure used for tuning the parameters, and the experimental results can be found in [5] or at `http://iridia.ulb.ac.be/~rghizzioli/dta/`.

Acknowledgments. This research has been supported by ANTS, an *Action de Recherche Concertée* funded by the French Community of Belgium. Marco Dorigo acknowledges support from the Belgian FNRS, of which he is a Research Director.

References

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, New York, NY, 1999.

[2] M. Campos, E. Bonabeau, G. Theraulaz, and J.-L. Deneubourg. Dynamic Scheduling and Division of Labor in Social Insects. *Adaptive Behavior*, 8:83–96, 2000.

[3] V. A. Cicirello and S. F. Smith. Wasp-like Agents for Distributed Factory Coordination. Technical Report CMU-RI-TR-01-39, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Pittsburgh, PA, December 2001.

[4] M. Dorigo and T. Stützle. *Ant Colony Optimization.* MIT Press, Cambridge, MA, 2004.

[5] R. Ghizzioli, S. Nouyan, M. Birattari, and M. Dorigo. An Ant-Based Algorithm for the Dynamic Task Allocation Problem. Technical Report TR/IRIDIA/2004-16, IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, 2004.

[6] R. Morley. Painting Trucks at General Motors: The Effectiveness of a Complexity-Based Approach. *Embracing Complexity: Exploring the Application of Complex Adaptive System to Business*, pages 53–58, 1996.

[7] S. Nouyan. Agent-Based Approach to Dynamic Task Allocation. In M. Dorigo, G. Di Caro, and M. Sampels, editors, *Proceedings of ANTS 2002 – Third International Workshop on Ant Algorithms*, volume 2463 of *Lecture Notes in Computer Science*, pages 28–39. Springer Verlag, Berlin, Germany, 2002.

[8] G. Theraulaz, E. Bonabeau, and J.-L. Deneubourg. Response Threshold Reinforcement and Division of Labour in Insect Societies. In *Proceedings of the Royal Society of London B*, volume 265, pages 327–335, 1998.

[9] E. O. Wilson. The Relation Between Caste Ratios and Division of Labour in the Ant Genus Phedoile. *Behav. Ecol. Sociobiol.*, 16:89–98, 1984.