Université Libre de Bruxelles

*Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle*

# An Insect-Based Algorithm for the Dynamic Task Allocation Problem

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

**IRIDIA – Technical Report Series**

Technical Report No.
TR/IRIDIA/2005-031

December 2005

S. Nouyan, R. Ghizzioli, M. Birattari, and M. Dorigo (2005) **An insect-based algorithm for the dynamic task allocation problem,** *Künstliche Intelligenz*, 4/2005:25–31.

**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/2005-031

# An Insect-Based Algorithm for the Dynamic Task Allocation Problem

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

**An insect-based algorithm inspired by the division of labor in insect colonies is proposed and applied to 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 address two issues. First, we propose and analyze four modifications of an insect-based algorithm previously introduced by Cicirello and Smith. Second, we propose an extension of the dynamic threshold model of Theraulaz et al. that was originally used 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, for various combinatorial optimization problems ant colony optimization [7] obtains state-of-the-art results.

In this paper we propose an insect-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. Here we investigate also 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 reconfigurations. This typically increases the efficiency of the whole system. Morley [13] 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 and reached 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 [4, 5]. These algorithms are inspired by the division of labor in social insects and adopt the *dynamic threshold model* proposed by Theraulaz [15].

In this paper we address two issues. First, we propose four modifications of an algorithm previously introduced by Cicirello and Smith [5]. A detailed analysis of the impact of each modification is given. Second, we propose a modification of the dynamic threshold model of Theraulaz *et al.* that was originally used for handling the case in which all booths are identical. We present a modified version of the threshold model which takes into account the heterogeneous processing speeds of the agents and we show that the modified version obtains better results than the original version.

Section 2 presents the problem using the example of a painting facility. Section 3 introduces related works, in particular detailing the market-based and two insect-based algorithms. Then, Section 4 explains the algorithm proposed here, showing the modifications we applied to improve the performance. Section 5 proposes an experimental analysis that highlights the performance improvement obtained by the proposed algorithm over the ones previously presented in the literature. Section 6 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 [9, 8].

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 with 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 needed before the booth can start processing the task. If no setup is necessary, the booth starts immediately to paint the following 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 Related Works

In Section 2 we used a painting facility as a real world example of the $DTA$ problem. This example originates from Morley [13], who used a market-based approach for optimizing a scheduler for a GM truck painting facility.

Independently from each other, Campos *et al.* [4] and Cicirello and Smith [5] used similar insect-based approaches to solve the same problem. They were inspired by a threshold model proposed by Bonabeau *et al.* [1, 2] and Theraulaz *et al.* [15].

In Section 3.1 we detail Morley's approach to the homogeneous case of the $DTA$ problem in which booths are identical. In Section 3.2 we introduce the threshold model and then explain the two insect-based approaches. Note that the presented algorithms all refer to the homogeneous case of the $DTA$ problem.

## 3.1 Market Based Approach

Market-based approaches are often used for coordinating asynchronous scheduling operations in the face of imperfect knowledge [6, 12, 10]. The decision process is based on a decentralized bidding mechanism where autonomous agents bid for a task or a resource, which is then assigned to the highest bidder. The agents dynamically adjust their bids according to their capability to resolve a task or according to the availability of a resource. In the following we detail the market-based approach to the homogeneous case of the $DTA$ problem. As the original algorithm developed by Morley is a manufacturing application many details are protected. Therefore, we present here the account of Morley's algorithm that was given by Campos *et al.* [4].

**Morley [13] (MBA):** Each painting booth autonomously bids for painting a truck. If the queue of a booth is full, the latter does not participate to the bidding process. If we consider a task $j$ of color $c_j$ that is in the storage, a booth $k$ that does not have a full queue participates in the bidding process with a value given by:

$$B_k(j) = \frac{P(1 + Ce(k,j))}{\Delta T_k(j)^L} \tag{1}$$

where $e(k,j)$ is a function that equals $1$ if a setup will be required for painting the truck with color $c_j$, and $0$ otherwise. $P$, $C$, and $L$ are parameters that weight each component. $\Delta T_k(j)$ is the time until task $j$ starts to be painted in boot $k$, and is determined by the following equation:

$$\Delta T_k(j) = qt^{proc} + nt^{setup} + t^{working} \tag{2}$$

where $q$ is the number of trucks in the queue of booth $k$, $t^{proc}$ is the time required to paint one truck, $n$ is the number of setups required for the trucks in the booth's queue, $t^{setup}$ is the time required for a setup, and $t^{working}$ is the time necessary to finish the currently painted truck.

All bids are compared and the respective truck is appended to the queue of the highest bidder. If more than one booth submit the same highest bid, the truck is assigned to the booth that requires no setup to paint it. Otherwise, if all or no booth requires a setup the winner is chosen randomly.

## 3.2 Insect Based Approach

As shown by Wilson [16] on ant species from the *Pheidole* genus, the concept of division of labor allows the colony to adapt to changing demands. In most species of the investigated genus workers are physically divided into two fractions: The small *minors*, who fulfill most of the quotidian tasks, and the larger *majors*, who are responsible for seed milling, abdominal food storage, defense, or a combination of these. Wilson experimentally changed the proportion of *majors* to *minors*. By reducing the fraction of *minors*, he observed that *majors* get engaged in tasks that are usually performed by *minors*.

Theraulaz *et al.* have developed a model of response thresholds in order to explain the behaviour observed by Wilson [16]. In this model a set of thresholds is given to each individual performing a task, one threshold for each type of task. A threshold's value represents the level of specialization in that task. Depending on the model, thresholds may remain fixed over time [3], or they may be dynamically updated in respect to the task currently performed [15]. For instance, while an ant is foraging for food the corresponding threshold would decrease, in this way increasing the level of specialization for that task, whereas the thresholds for all other tasks would increase.

A task emits a stimulus to attract the individuals' attention. Based on this stimulus and on the corresponding threshold, an individual will or will not accept the task. The lower a threshold, the higher the probability to accept a task. Thus a lower threshold represents a higher grade of specialization.

This model is the core of the different insect-based algorithms that will be detailed in the following.

**Campos et al. [4] (ABA):** A painting booth is represented by an agent which autonomously competes to paint a truck. Each agent $k$ has a threshold value $\theta_{k,c}$ for each color $c$. A stimulus $s_j$ is associated to a truck $j$. The stimulus $s_{c_j}$ is established for each color available in the system and is given by the sum of the stimuli of the unassigned tasks in each particular color.

The probability of booth $k$ to get engaged in task $j$ is given by:

$$P(s_{c_j}, \theta_{k,c_j}) = \frac{s_{c_j}^2}{s_{c_j}^2 + \alpha\theta_{k,c_j}^2 + \Delta T_k^{2\beta}(j)} \tag{3}$$

where $c_j$ is the color of truck $j$ and $\theta_{k,c_j}$ is the threshold of agent $k$ for color $c_j$. $\alpha$ and $\beta$ are parameters and $\Delta T_k(j)$ is the same time as computed in Equation 2.

Values of $P(s_{c_j}, \theta_{k,c_j})$ are compared and the task is assigned to the booth with the highest value. If the truck $j$ is assigned to booth $k$, the threshold values are updated for all of the booths. $\theta_{k,c_j}$ decreases by the amount $\xi$:

$$\theta_{k,c_j} \rightarrow \theta_{k,c_j} - \xi, \tag{4}$$

and the thresholds $\theta_{m,c_j}$ of all other paint booths for color $c_j$ increase by the amount $\phi$

$$\theta_{m,c_j} \rightarrow \theta_{m,c_j} + \phi, \ \forall m \neq k. \tag{5}$$

The parameters $\xi$ and $\phi$ are the *learning* and *forgetting* factor, respectively. The thresholds $\theta_{*,c_j}$ are constrained to the interval $[1, 500]$. Equation 4 tries to express the fact that booth $k$ tends to specialize on color $c_j$ because it increases its probability to respond to a truck with color $c_j$ by decreasing its response threshold $\theta_{k,c_j}$.

**Cicirello and Smith [5] (R-WASP):** Similar to *ABA*, the algorithm of Cicirello and Smith is also based on the threshold model by Theraulaz [15]. However, there are several differences between the two algorithms.

First of all, the stimulus $s_j$ associated to a given task $j$ is equal to the amount of time the task is already waiting to be assigned to an agent. The probability to respond to a stimulus is given by:

$$P(s_j, \theta_{k,c_j}) = \frac{s_j^2}{s_j^2 + \theta_{k,c_j}^2} \tag{6}$$

where $c_j$ represents the color or the type of task $j$.

Furthermore, in *ABA*, *thresholds* are updated only when a truck is assigned to a paint booth and that rule involves only thresholds $\theta_{*,c_j}$. In *R-WASP*, each agent $k$, at each time step, updates its own thresholds $\theta_{k,*}$ according to the task it is currently processing. At each time step, if agent $k$ is processing or setting up for a task $j$ of color $c_j$ the agent's threshold for this color are decreased according to Equation 4, while all other thresholds are increased according to Equation 5. In addition to these rules which are also employed by *ABA* there is a third threshold update rule for agents that are idle, i.e., currently not processing any task. In that case, the threshold for each color $c$ is decreased. Formally:

$$\theta_{k,c} \rightarrow \theta_{k,c} - \delta^t, \ \forall c \tag{7}$$

where $t$ represents the number of time steps for which the agent has already been idle.

When two or more agents compete for the same task, they interact with each other in a *dominance context* and the winner is assigned the task. Each participant $k$ has a force value $F_k$:

$$F_k = 1 + T^{proc} + T^{setup} \tag{8}$$

where $T^{proc}$ is the sum of the processing times $t^{proc}$ for all the tasks in the queue of booth $k$ and $T^{setup}$ is the sum of their setup times $t^{setup}$. The probability for agent $k$ to win is:

$$P(F_1, ..., F_n) = \frac{\sum_{i \neq k}^{n} F_i^2}{(n-1)\sum_{i=1}^{n} F_i^2} \tag{9}$$

# 4 Modifications to the Insect-Based Approach

In this section we introduce an original insect-based algorithm for to the $DTA$ problem. In Section 4.1 we propose four general modifications to *R-WASP*, with the goal to improve the performance [14]. We name the proposed algorithm Ant-Task-Allocation (*ATA*). In Section 4.2 a modification is introduced that takes into account the different process times for the heterogeneous case of the $DTA$ problem. This modification can be applied to all the insect-based algorithms.

## 4.1 General Modifications: ATA

**Threshold Update Rules (TUR)**: The update rules proposed by Cicirello and Smith are based on the type of task that is currently processed. The threshold of the currently processed task is reduced, all other thresholds are increased. This is motivated by the behaviour in social insects, who specialize in the task they are performing and tend to continue the task rather than switching it, n this way avoiding the cost in time that would be related to such a switch. However, for the $DTA$ problem the situation is not the same. As each painting booth has a queue that can be filled with tasks, the fact whether a given task causes a setup depends on the type of the last task in the agent's queue, not on the one currently processed. Therefore, in *ATA* it is the last job in a machine's queue that determines how the threshold values are updated.

**Calculation of the Force Variable (CFV)**: The force variable used for the dominance contest does not take into account whether a setup would be required for the respective task. Therefore, in *ATA* the force value is modified according to:

$$F_k(j) = 1 + T^{proc} + T^{setup} + t_j^{setup} \tag{10}$$

where $t_j^{setup}$ is 0 in case no setup is necessary, and the required setup time otherwise.

**Dominance Contest (DC)**: Another problem of the dominance contest is that the more machines compete with each other in a dominance contest, the smaller are the differences between the probabilities to win. In general the probability for one competitor to win a dominance contest with $n$ competitors is never higher than $\frac{1}{n-1}$. In *ATA* this problem is overcome by using the following rule instead of the one specified in Equation 9:

$$P_k(F_1, ..., F_n) = \frac{\frac{1}{F_k^2}}{\sum_{i \neq k}^{n} \frac{1}{F_i^2}}. \tag{11}$$

**Idle Machine does not Compete (IMC)**: Equation 7 is applied for idle agents in order to encourage the agent to compete for tasks of any type. In *ATA* an additional rule is introduced that is only applied when an idle agent refuses to compete for a given task. In this case the threshold for the respective color is diminished:

$$\theta_{k,c_j} \rightarrow \theta_{k,c_j} - \gamma. \tag{12}$$

## 4.2 Modification for the Heterogeneous Case

The definition of the $DTA$ problem considers agents that can have different processing times for different task types. The presented algorithms can in principle be applied to the $DTA$ problem. Nevertheless, they do not take into account the different processing speeds of the agents, and therefore their performance is rather poor, as it is shown in Section 5.2. The problem is that agents compete for a task considering only their specialization level without taking into account their own characteristics, that is, their speed in processing a task. In order to overcome this, we extend the threshold model so that the faster an agent can perform a task, the higher is the probability that the agent competes for it. To this aim we introduced the processing speed of agent $k$ on task $j$ into the probability function $P(s_j, \theta_{k,c_j})$ by substituting the threshold $\theta_{k,c_j}$ in Equations 3 and 6 with $\Theta_{k,c_j}$:

$$\Theta^2_{k,c_j} = \theta^2_{k,c_j}(t^{proc}_{k,j} - t^{min.proc}_j + 1), \tag{13}$$

where $t^{proc}_{k,j}$ is the time that agent $k$ requires to process task $j$ and $t^{min.proc}_j$ 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. We have tested several different substitutions of the threshold value in the probability functions, with the one given by Equation 13 yielding the best results. The substitution described in Equation 13 can be used for the three insect-based algorithms described above. In the following we call *ABAc*, *R-WASPc*, and *ATAc*, the modifications of *ABA*, *R-WASP*, and *ATA*, respectively.

# 5 Empirical Analysis

We have conducted two series of experiments in order to analyze (i) the impact of the modifications we have applied to *R-WASP*, (ii) the impact of Equation 13 when applied to the heterogeneous $DTA$ problem. All parameters were tuned separately for each approach using an evolutionary algorithm. All algorithms were devoted the same tuning effort. Table 1 summarizes the tuned parameters. For more details we refer to [9, 8].

| Algorithm | Tuned Parameters |
|---|---|
| *MBA* | $P, C, L$ |
| *ABA*, *ABAc* | $\alpha, \beta, \xi, \phi$ |
| *R-WASP*, *R-WASPc* | $\delta, \xi, \phi$ |
| *ATA*, *ATAc* | $\delta, \xi, \phi, \gamma$ |

Table 1: Summary of the tuned parameters.

## 5.1 General Modifications: ATA

In this section we compare all possible modifications proposed with the *ATA* algorithm on a class of instances of the homogeneous case of the $DTA$ problem. In the following we present the experimental setup, the analyzed algorithms and the results.

**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 24 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. As we focus on the analysis of the four modifications we apply the algorithms to the homogeneous case of the $DTA$ problem. All booths require 5 minutes to paint a truck and an additional 10 minutes in case a setup is necessary.

The number of trucks exiting the assembly line is always equal to 2016, which is the maximum number of trucks that 24 booths can paint considering a process time of 5 minutes when no setups are required[1].

[1] 24 booths * 420 time steps / 5 time steps per truck = 2016.

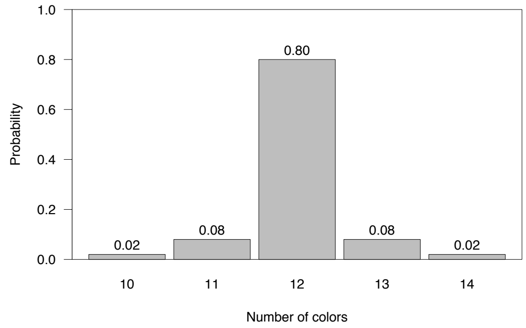



Figure 1: Probability distribution of the number of colors for the homogeneous $DTA$ experiment.

For each instance, the number of different colors is chosen with respect to the distribution shown in Figure 1. Color types are assigned to the trucks according to the two mixes:

1. Two subsets of the $n$ colors are considered. Colors in the first subset have a higher probability of being selected than those in the second subset. The probability $P(c)$ of selecting color $c$ is defined as follows:

$$P(c) = \begin{cases} \frac{1}{4\lfloor n/2 \rfloor} & \text{if } c \leq \lfloor n/2 \rfloor \\ \frac{3}{4\lceil n/2 \rceil} & \text{if } c > \lfloor n/2 \rfloor \end{cases}$$

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.

**Algorithms:** As there are four proposed modifications there are 16 possible combinations of the proposed modifications which are all analyzed. We use a binary notation with four bits to identify these combinations. The first bit represents whether or not TUR is active, the second bit represents CFV, the third DC, and the last IMC. For instance, 0000 indicates that no rules are active, 1000 means that only TUR is active, and so on.

**Results:** We have run all the algorithms on 1000 instances of the presented problem class. Figure 2 summarizes the results, which show that, on average, configuration 1011 performs best, and that *ATA* (configuration 1111) is very close to it. Moreover, all possible configurations have a lower makespan than *R-WASP* (configuration 0000), indicating that each modification has a positive impact on the performance. Furthermore, the rule with the highest impact on the performance is TUR: indeed the configurations on the right-hand side of Figure 2, in which TUR is active, are apparently better than those on the left-hand side.

## 5.2 Modification for the Heterogeneous Case

In this section we compare the presented algorithms on a class of instances of the heterogeneous $DTA$ problem. In the following we present the experimental setup, all the analyzed algorithms and the results.

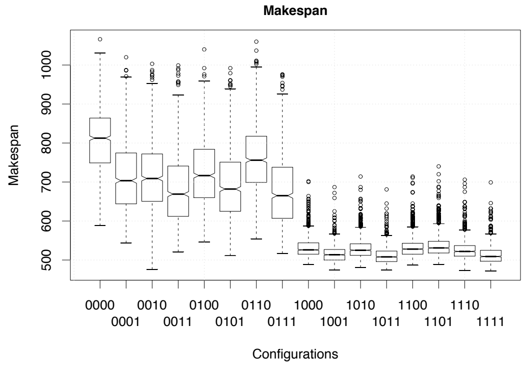



Figure 2: Box plot of the makespan values for all 16 possible combinations of the four modifications to *R-WASP* when applied to the homogeneous $DTA$ problem. Binary numbers indicate usage of the modifications TUR, CFV, DC, and IMC.

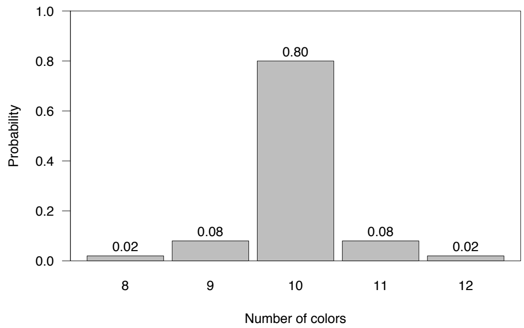



Figure 3: Probability distribution of the number of colors for the $DTA$ experiment.

**Experimental Setup:** The setup used is very similar to the previous one. The differences are that instead of 24 we consider 12 painting booths, and that we extract the number of colors from the probability distribution presented in Figure 3. Furthermore, given that we analyze the heterogeneous case of the $DTA$, 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.

**Algorithms:** In this analysis we compare the four previously described algorithms *MBA*, *ABA*, *R-WASP* and *ATA*. To evaluate the performance of the modified model, we have applied it to all insect-based algorithms, which are named *ABAc*, *R-WASPc*, *ATAc*. Additionally, a trivial non-adaptive *greedy* algorithm named *LOCUST* is introduced in order to have a performance reference. Applying *LOCUST*, trucks are allocated to booths following the dominance contest as used by *R-WASP*.

**Results:** We have run all the algorithms on 1000 instances of the presented class.

Figure 4 summarizes the results for the $DTA$ experiment. The values and ranks are shown in Figure 4(a) and (b) for the makespan, and in Figure 4(c) and (d) for the number of setups. Among the eight compared algorithms, *ATAc* obtains the best results. It achieves makespan values which are very close to the lower bound of 420 minutes. Furthermore, it reaches on average the lowest number of required setups. 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.

Interestingly, *ATA* achieves a low makespan and requires a low number of setups. It is the only algorithm that achieves this without the modification proposed in Section 4.2. This suggests that the four proposed modifications with respect to *R-WASP* are robust enough to cope with the heterogeneous case of the $DTA$ problem. While *R-WASPc* is the algorithm that benefits the most from the modification, the improvement is smaller for *ABA*.

# 6 Conclusion

In this paper we have presented an insect-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 a previously presented algorithm by Cicirello and Smith [5] based on the *dynamic threshold model*, and we showed that each modification proposal leads to a better performance. In particular, the modification of the threshold update rules, TUR (see Section 4.1 for details), is the modification with the highest impact.

Furthermore, we addressed the heterogeneous case of the $DTA$ problem in which different groups of booths require different processing times for completing a same task. We introduced a simple modification to the insect-based algorithms that significantly improves their performances. The modification, given by Equation 13, consists in a substitution of the threshold value in the probability functions, in this way taking into account the different processing times of the painting booths.

A number of issues deserve further analysis. For instance, the substitution of the threshold value we have chosen is just one out of many that would be possible. Even though we have achieved promising results, a problem of the current implementation is that rescaling processing times would lead to a different outcome in the respective probability function. A different parameter set would therefore be required. One possible solution to this problem could be to divide rather than to subtract the two terms $t_{i,j}^{proc}$ and $t_j^{min.proc}$ in Equation 13.

Furthermore, an analysis of the parameter space for each analyzed algorithm might be worthwhile. Kittithreerapronchai and Anderson [11] have already addressed this topic for *ABA* and *MBA*. They found that the parameter surfaces are surprisingly smooth, and were able to identify key parameters.

**Acknowledgements.** 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. The authors thank the reviewers for their helpful suggestions and comments.

(a)

(b)

(c)

(d)

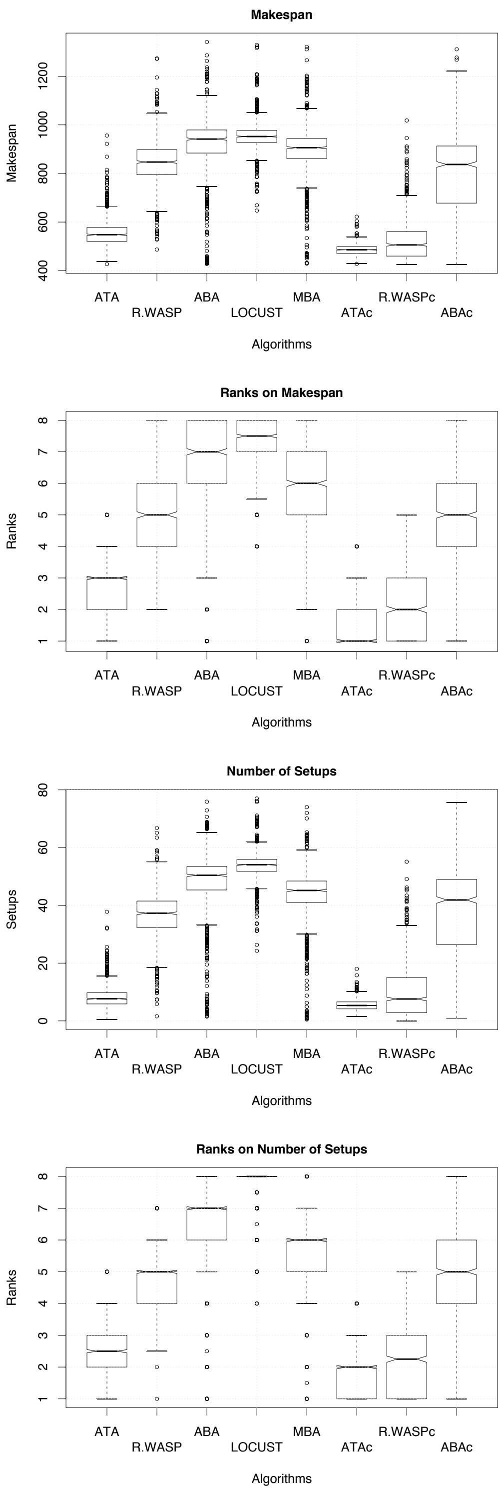



Figure 4: Results for the $DTA$ experiment, showing box plots of (a) the makespan values, (b) the makespan ranks, (c) the values for the number of setups, and (d) the ranks for the number of setups for the eight algorithms. The figure shows that the improved model leads to an improvement of the performance especially to the *R-WASP* algorithm.

# References


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

[2] E. Bonabeau, A. Sobkowski, G. Theraulaz, and J.-L. Deneubourg. Adaptive task allocation inspired by a model of division of labor in social insects. In D. Lundh, B. Olsson, and A. Narayanan, editors, *Biocomputing and Emergent Computation*, pages 36–45, Singapore, 1997. World Scientific.

[3] E. Bonabeau, G. Theraulaz, and J.-L. Deneubourg. Fixed response thresholds and the regulation of division of labor in insect societies. *Bulletin of Mathematical Biology*, 60:753–807, 1998.

[4] 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.

[5] V. A. Cicirello and S. F. Smith. Wasp-like agents for distributed factory coordination. *Journal of Autonomous Agents and Multi-Agent Systems*, 8:237–266, 2004.

[6] S.-H. Clearwater. *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific, Singapore, 1996.

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

[8] R. Ghizzioli. Insect-based approach to the dyamic task allocation (DTA) problem. `http://iridia.ulb.ac.be/~rghizzioli/dta/`, 2005.

[9] 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.

[10] B. A. Huberman and T. Hogg. Distributed computation as an economic system. *Journal of Economic Perspectives*, pages 141–152, 1995.

[11] O. Kittithreerapronchai and C. Anderson. Do ants paint trucks better than chickens? Markets versus response thresholds for distributed dynamic scheduling. In *Proceedings of the 2003 IEEE Congress on Evolutionary Computation*, pages 1431–1439, Canberra, December 2003. IEEE Press.

[12] J. F. Kurose and R. Simha. A microeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Transactions on Computers*, 38:705–717, 1989.

[13] 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.

[14] 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.

[15] 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.

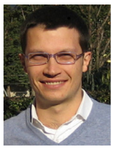
[16] E. O. Wilson. The relation between caste ratios and division of labour in the ant genus phedoile. *Behav. Ecol. Sociobiol.*, 16:89–98, 1984.

## Contacts

Shervin Nouyan, Dr. Mauro Birattari, Prof. Marco Dorigo
IRIDIA, Université Libre de Bruxelles
50, Av. F. Roosevelt, CP 194/6
B-1050 Bruxelles, Belgique
Tel.: +32 (0)2 650-2729
Fax: +32 (0)2 650-2715
Email: {snouyan,mbiro,mdorigo}@ulb.ac.be

Roberto Ghizzioli
Whitestein Technologies AG
Pestalozzistrasse 24
CH-8032 Zürich, Switzerland
Tel.: +41 44-256-5000
Fax: +41 44-256-5001
Email: rgh@whitestein.com

Shervin Nouyan received his master degree in electrical engineering in 2002 from the *Technische Universität München*, and the Diplôme d'Études Approfondies in 2004 from the Université Libre de Bruxelles, Brussels, Belgium. He is a Ph.D. student at IRIDIA, Université Libre de Bruxelles. His research interests are in swarm intelligence and swarm robotics.



Roberto Ghizzioli received his master degree in computer science from the *University of Milano*, Italy, in 2004. He made his diploma thesis at IRIDIA, Université Libre de Bruxelles, Brussels, Belgium. Currently he is a member of the Research and Consulting group of Whitestein Technologies AG in Zürich, Switzerland, working on optimization problems.

Mauro Birattari received a master degree in electronic engineering from *Politecnico di Milano*, Italy in 1997, and a Ph.D. in applied sciences from *Université Libre de Bruxelles*, Belgium in 2004. Dr. Birattari is active in the fields of machine learning and computational intelligence. His research interests include local learning techniques, metaheuristics for combinatorial optimization, and ant colony optimization.

Marco Dorigo is Research Director of the FNRS, the Belgian National Fund for Scientific Research, and a Research Director of IRIDIA, the artificial intelligence laboratory of the Université Libre de Bruxelles. He is the inventor of the ant colony optimization metaheuristic. His current main research interests are in swarm intelligence and swarm robotics. In 1996 he was awarded the Italian Prize for Artificial Intelligence and in 2003 the European Marie Curie Excellence Award.