Adaptive Behavior
http://adb.sagepub.com

Dynamic Scheduling and Division of Labor in Social Insects

Mike Campos, Eric Bonabeau, Guy Théraulaz and Jean-Louis Deneubourg
Adaptive Behavior 2000; 8; 83
DOI: 10.1177/105971230000800201

The online version of this article can be found at:
http://adb.sagepub.com/cgi/content/abstract/8/2/83

Published by:
SAGE Publications
http://www.sagepublications.com

On behalf of:
ISAB
International Society of Adaptive Behavior

Adaptive Behavior 2000, 8(2), 83-96

Dynamic Scheduling and Division of Labor in Social Insects

MIKE CAMPOS[1], ERIC BONABEAU[2],
GUY THÉRAULAZ[3] and JEAN-LOUIS DENEUBOURG[4],
[1]*Northwestern University and Eurobios*
[2]*Santa Fe Institute and Icosystem Corporation*
[3]*Université Paul Sabatier*
[4]*Université Libre de Bruxelles*

A method for assigning tasks or resources, based on a model of division of labor in social insects, is introduced and applied to a dynamic flow shop scheduling problem. The problem consists of assigning trucks to paint booths in a truck facility to minimize total makespan and the number of paint flushes. Similarities between the *ant-based* approach and a *market-based* approach are highlighted. Both systems are able to adapt well to changing conditions.

1 INTRODUCTION

Market-based algorithms have been introduced several years ago (Kurose & Simha, 1989; Waldspurger *et al.*, 1992; Huberman & Hogg, 1995; Clearwater & Huberman, 1994; Clearwater,1995; Wellman, 1993, 1995; Ykke *et al.*, 1997; Ykke, 1998) as a new paradigm for controlling complex, unpredictable systems. In a market-based algorithm, resources or tasks are allocated efficiently through a market-clearing mechanism. Agents bid for resources and the agent with the highest bid gets the resource. The market-clearing mechanism ensures that all tasks or resources have been allocated. Agents adjust their bids according to prior successes or failures in getting the resource. For instance, Waldspurger *et al.* (1992) have developed Spawn, a computational system where each task, starting with a certain amount of "money" that corresponds to its relative priority, bids for the use of machines on a network. Examples of tasks include searching through a database for an item, printing a paper, etc. Through bidding for the use of machines, each task can allocate its budget to those resources that are most important for it (Waldspurger *et al.*, 1992; Huberman & Hogg, 1995). When prices are low enough, some tasks may use several machines in parallel; the number of machines used by, and the computation time devoted to, each task are adjusted to the demand from other tasks. In another example (Clearwater & Huberman, 1994), the problem of thermal resource distribution in a building was solved using market-based control: computational agents that represent individual temperature controllers bid to buy or sell cool or warm air via a double-blind computerized auction moderated by a central computer auctioneer. Clearwater and Huberman (1994) have shown that this system results in an equitable temperature distribution throughout the building at a reduced cost. Ygge *et al.* (1997), who analyzed the performance of Clearwater and Huberman's (1994) approach and carefully compared it with other approaches, argue that this problem lends itself to a treatment with a market-based approach in that the goal state can be formulated as a market equilibrium; they also indicate that other multi-agent-based approaches might work equally well or even better. Other, more sophisticated, market-based approaches have been developed in the recent years: for example, Walsh *et al.* (1998) have formulated distributed scheduling as a resource allocation problem, applied general equilibrium theory to show the existence of, and characterize, equilibrium solutions, and devised auction protocols for implementing solutions.

Social insects –ants, bees, termites and wasps–provide us with another metaphor for controlling complex systems (Bonabeau *et al.*, 1999). A social insect colony *is* a complex system (Wilson, 1971) often characterized by division of labor (Oster & Wilson, 1978; Robinson, 1992): workers tend to be specialized in certain tasks. But this specialization is flexible.

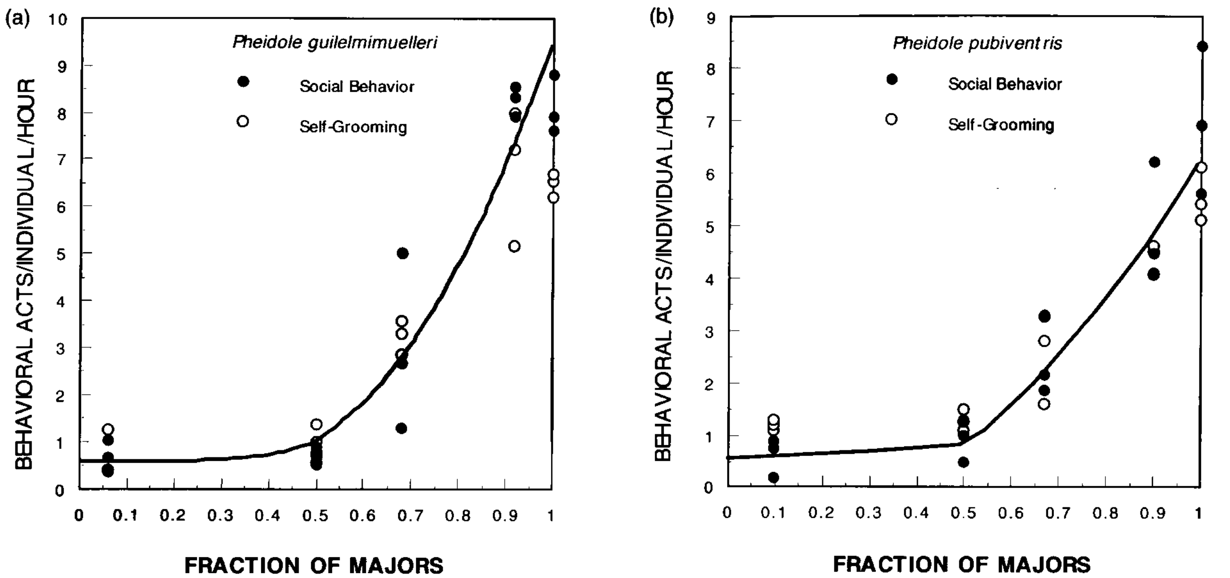


Figure 1: Number of behavioral acts (social behavior and self-grooming) per major per hour as a function of the fraction of majors in the colony: (a) *Pheidole guilelmimuelleri*, (b) *P. pubiventris*. Modified from Wilson (1984).

The flexibility of task allocation exhibited at the colony level is connected to the *elasticity* of individual workers. Such factors as food availability, predation, climatic conditions, phase of colony development, or time of year influence the size and structure of a colony's worker population in natural conditions. The worker force must be allocated to tasks so as to adjust to such changing conditions. Changes in the pattern of task allocation can be induced experimentally by altering colony size, structure, or demography, or by increasing the need for nest maintenance, nest repair, defense, etc. Wilson (1984) experimentally altered the structure of colonies of polymorphic ant species from the *Pheidole* genus. In most species of this genus, the worker population is divided into two morphological subcastes the minor and major subcastes. Minors, which take care of most of the quotidian tasks of the colony, are smaller than the large-headed majors (often called soldiers), which are specialized either for seed milling, abdominal food storage, defense, or some combination of these functions. Wilson (1984) showed that majors in the species he studied exhibit *elasticity*: their normally limited behavioral repertoire can be stretched back and forth in a predictable manner in response to perturbations. He artificially tuned the ratio of majors to minors and observed a change in the rate of activity within one hour of the ratio change. When the fraction of minors in the colony becomes small, majors engage in tasks usually performed by minors and efficiently replace the missing minors. Wilson (1984) studied the number of behavioral acts per major per hour as a function of the fraction of majors in the colony for two tasks that are normally performed by minors, for two species of the genus *Pheidole*. The usual proportion of majors in these species varies between 5% and 30%. Majors become progressively more and more involved in performing minors' tasks as the fraction of majors increases (and as the fraction of minors decreases) (see figure 1). These experiments suggest that the flexibility of a colony as a whole, that is, the degree to which the colony responds to alterations, is determined by the elasticity observed at the individual level.

A simple model based on response thresholds allows a formal connection between the level of individual workers and the colony level (Bonabeau *et al.*, 1996, 1997, 1998; Théraulaz *et al.*, 1998). Workers with low response thresholds respond to lower levels of stimuli than workers with high response thresholds. Task performance reduces the intensity of stimuli. If workers with low thresholds perform their normal tasks the task-associated stimuli never reach the thresholds of the high-threshold workers. But if, for any reason the number of workers with low thresholds decreases or the intensity of task-associated stimuli increases, high-threshold workers engage in task performance. There is a good agreement between the model and experimental observations of colony-level flexibility in ants of the genus *Pheidole* (Wilson, 1984). According to this model, tasks that need to be performed represent a *job market* in which individual insects are (more or less) actively seeking a job. Tofts and Franks (1992) speak of "foraging-for-work". Insects with low response thresholds get the job; in the absence of low-threshold individuals, insects with high response thresholds will perform the task. An extension of this model includes a simple form of learning (Théraulaz *et al.*, 1991, 1998). Within individual workers, performing a given task induces a decrease of the corresponding threshold, and not performing the task induces an increase of the threshold. This double reinforcement process leads to the emergence of specialized workers, that is, workers that are more responsive to stimuli associated with particular task requirements, from a group of initially identical individuals. This reinforcement process also permits the adjustment, in response to changing internal or external conditions, of the numbers of workers engaged in different tasks.

At some level of abstraction, following the job market analogy, a comparison between task allocation in a social insect colony and the allocation of resources in a market is therefore possible. For example, bidding high is similar to having a low threshold. The aims of the present paper are to: (1) show the feasibility of a robust scheduling algorithm inspired by task allocation in social insects (the *ant-based approach*), and (2) highlight the similarities between a market-based approach and an ant-based approach.

Both paradigms are applied to a *dynamic* flow shop scheduling problem. The problem, described in more detail in section 2, consists of assigning trucks to paint booths in a truck facility to minimize total makespan and the number of paint flushes in constantly changing conditions. The corresponding *static* problem, that is one in which one knows ahead of time all colors and booths never go down, is a hard combinatorial optimization problem. In reality customer orders are processed in real time and paint booths may go down unexpectedly, which makes a preplanning unreliable and the dynamic problem hard to solve. The market-based approach (Morley, 1996; Morley & Ekberg, 1998), described in section 3, had been implemented in a real truck factory. Section 4 is dedicated to the ant-based approach. In section 5 we describe the genetic algorithm (GA) used to find good parameters for both the market-based and the ant-based algorithms. Both algorithms have been run on 4 test sets: results are presented and discussed in section 6.

2 DESCRIPTION OF THE PROBLEM

The problem considered here (Morley, 1996; Morley & Ekberg, 1998) is to assign trucks to paint booths in a truck facility: trucks get out of the assembly line at a quick pace (one per minute) to be painted. The color that a truck should have is predetermined directly by a customer order. The paint fit-and-finish operations are a traditional bottleneck that can significantly reduce production throughput. It takes 3 minutes to paint a truck, but it takes more time if the color of a paint booth must be changed: if, for example, a given paint booth is applying black paint and the next truck to be processed by that paint booth requires white paint, the booth will be down for 3 minutes, the time it takes to flush out the old paint and fill the booth with the appropriate color. The cost of paint changeover is also high.
Therefore, the number of such changeovers should be minimized online.

The algorithms described in the next sections have been tested on a simulated (and therefore simplified) version of the problem. In the simulations, we have 20 possible colors, 6 to 15 booths, and a maximum queue for each booth set at 5 trucks. Also, at any given time some booths will likely be inoperable. To simulate the unpredictable operation of the paint booths we implemented a 1 in 20 chance of one, randomly selected booth going down at each time step. A paint booth that goes down is inoperable for a random amount of time, uniformly distributed between 0 and 20 min. Finally, trucks in the queue of a booth that went down cannot be redispatched and therefore have to wait until the paint booth is operating again.

3 THE MARKET BASED APPROACH

The algorithm developed by Morley (1996; Morley & Ekberg, 1998) is a manufacturing application and many of its details are protected. Therefore the market-based approach described in this section is *our* version of Morley's algorithm, not necessarily the most optimal one; nor is it the most sophisticated or most up-to-date market-based approach. *Ad hoc* choices were made based on our understanding of the real implementation. Remember that our goals are to show that it is possible to design a (robust) scheduling algorithm inspired by (flexible) division of labor in social insects, and to highlight the similarities between market mechanisms and task allocation mechanisms in social insects.

If a paint booth which is applying black is the only one that is available or has a small waiting queue, and if painting the white truck is urgent, the white truck should be allocated to that paint booth. Morley (1996; Morley & Ekberg, 1998) developed a system in which each paint booth is an agent that follows four simple rules:

1. Try to take another truck the same color as the current color.

2. Take particularly important jobs.

3. Take any job to stay busy.

4. Do not take another job if paint booth is down or queue is full.

These rules were implemented through a bidding system and a market-clearing mechanism (every truck must be painted). Booths bid based on their ability to do the job efficiently—low cost, minimal delay. A paint booth that is down or the queue of which is full does not participate in the bidding. The optimal parameters of the bidding system were determined with evolutionary computation techniques (see section 5). The bidding system, implemented in a real truck factory, resulted in a 10% reduction in paint usage, half as many paint changeovers, a higher global throughput (booths are busier), and a significantly shorter software code than with the previous method, which was based on a global preoptimization of the paint schedule. Most of these nice results come from the fact that the system is more robust and more flexible: a booth that breaks down has a limited impact on the throughput because the schedule is generated online.

More precisely, a bid is based on three factors (Morley, 1996; Morley & Ekberg, 1998): the length of the line in front of the booth, the priority of the truck, and the necessity of a paint flush. In other words, with varying relative degrees of importance a booth will make a high bid for a truck if the truck is of the same color, if the truck is an important job, and if there is room in the booth's queue. At the beginning of each time step the booths make bids on the truck that just comes out of assembly. The truck is assigned colors (c_i =0-19) and priorities (w_i =1-100) in the simulation with uniformly distributed random numbers from the appropriate ranges. A truck is assigned to the booth with the highest bid. Booths that are down or that have no room in their queue do not make bids.

The bid function B_k of booth k for truck i is given by:

$$B_k = \frac{P \cdot w_i \cdot (1 + C \cdot c_{i,k})}{\Delta T^L} \qquad (1)$$

where w_i is the priority of truck i, $c_{i,k}$ has a value of 1 if the last truck in the queue of booth k matches the color truck i needs to be painted, and a value of 0 if taking on truck i requires a paint flush, ΔT is the time it would take before truck i would be painted, and P, C, and L are parameters to set the relative importance of the three components, w_i, $c_{i,k}$ and ΔT. The variable ΔT is computed by summing the product of the paint time t_p=3 *min* and the number of trucks q_k in booth k's queue, plus the product of the flush time t_f=3 *min* and the number of times n_k^f that the next truck in line requests a different color than the truck in front of it, plus the remaining time t_k^r the booth is painting the truck that is currently in the booth:

$$\Delta T = q_k \cdot t_p + n_k^f \cdot t_f + t_k^r \qquad (2)$$

In addition a method of breaking ties has to be defined. In the case that more than one booth make the same highest bid, the contending booths are checked for agreement between the color of the last truck in their lines with the color of the considered truck, i.e. which booths can paint the truck without a paint flush. In the case that none of the contending booths has a matching color, one of the booths with the highest bid is selected at random. If there are more than one contending booth with the same color, the one with the shortest line is selected. If it is still undecided, then the lowest numbered of the remaining booths is given the truck.

In the event that no booth makes a bid (because all of the paint booths are either down or have full queues) the truck is assigned to temporary storage. In the running of the program it is apparent that if there are at least eight paint booths, then trucks are assigned to storage very rarely.

The bids are compared and the highest bidder has the new truck assigned to the end of its queue. Next the booths check if there are any trucks in storage. If there are the booths make bids on these trucks. Since a truck only goes to storage in the case that all of the operable booth's queues are full, it is possible that the booths are still unable to receive more trucks. In that case the truck simply stays in storage.

If a booth is neither painting, nor flushing, nor down, is checks to see if it has any trucks in its queue. If it does, and the truck's color matches the color of the paint inside the booth it begins working and the rest of the trucks move up in line. If the colors do not agree the booth begins to flush. If there are no trucks in the queue then the booth waits until the next time step. Each time step represents one minute.

This process repeats for a maximum of 1000 time steps, but stops after the last truck is painted, when applicable. The last truck comes out of the assembly line at the 419th time step, and all of the trucks are usually painted within 10 time steps after that. This is analogous to a truck coming out of the assembly line every minute for seven hours.

4 THE ANT-BASED ALGORITHM

The ant-based algorithm deals with the exact same external conditions. The difference comes in how the trucks are assigned to the booths. First, a global demand D_j is established for each color j given by the sum of the priorities of the unassigned trucks in each particular color, i.e. the truck that have just exited the assembly line plus the priority of any truck of the same color in storage. Using the same notations as in the previous section, we have

$$D_j = \sum_i w_i \cdot \delta(c_i - j) \tag{3}$$

where c_i is the color of truck i, $\delta(\cdot)$ is the Dirac function and the sum runs over all trucks, with $w_i = 0$ by convention if truck i has been assigned to a booth. Next the booths consider the demand of the color that the truck that just comes out of assembly needs to be painted. The tendency of a booth to respond to the demand of the color of unassigned truck i is quantified by

$$P_k = \frac{D_{c_i}^2}{D_{c_i}^2 + \alpha \cdot \theta_{k,c_i}^2 + \Delta T^{2 \cdot \beta}} \tag{4}$$

where c_i is the color of truck i, θ_{k,c_i} is the threshold booth k has for color c_i, ΔT is the same as in the market-based algorithm, and α and β are parameters to weigh the relative importance of their respective terms. P_k is used as follows: values of P_k for the different paint booths are compared and the booth with the largest value is assigned the truck. Ties of the largest value are also decided in the same way as in the market-based algorithm.

After truck i is assigned to booth k the values of θ_{m,c_i} are updated for all of the booths (denoted by m). θ_{k,c_i} decreases by an amount ξ:

$$\theta_{k,c_i} \leftarrow \theta_{k,c_i} - \xi \tag{5}$$

and the thresholds θ_{m,c_i} of all other paint booths for color c_i increase by an amount ϕ:

$$\theta_{m,c_i} \leftarrow \theta_{m,c_i} + \phi \tag{6}$$

Variations of θ_{m,c_i} take place within the bounds θ_{min} and θ_{max}. Equations 5 and 6 express the fact that booth k tends to *specialize* on color c_i, because it increases its probability of responding to a truck with color c_i by decreasing its response threshold θ_{k,c_i}, while while all other booths tend to lose their sensitivity to color c_i by increasing their response thresholds θ_{m,c_i}.

5 GENETIC ALGORITHM

To determine the best combination of values for the parameters of both the market-based and the ant-based algorithms, a genetic algorithm (GA) was used (Goldberg, 1989; Riolo, 1992). There are 3 parameters in the market-based algorithm (P, L, C) and 6 parameters in the ant-based algorithm (α, β, ξ, ϕ, θ_{min}, θ_{max} with which to assign relative importance to certain components of the algorithms. The ranges of variation of these parameters, summarized in Table 1, define the limits to the parameter space that the GA searches.

MarketBased
$0.0 \le P < 100.0$
$0.0 \le L < 5.0$
$0.0 \le C < 10000.0$
Ant Based
$0.0 \le \alpha < 1000.0$
$0.0 \le \beta < 5.0$
$0.0 \le \xi < 10.0$
$0.0 \le \phi < 20.0$
$0.0 \le \theta_{min} < 10.0$
$0.0 \le \theta_{max} < 200.0$

Table 1. Ranges of variation of the parameters for the market-based and the ant-based algorithms.

The same GA procedure was applied to both the market-based and the ant-based algorithms. First a random population of 50 individuals is generated. An individual's *Genome* is a 24- or 48-bit string of 0's and 1's that is a binary representation of a set of parameters, and its *Fitness* is a measure of how well that set of parameters performs in the simulation. Each parameter is represented as a binary string of eight bits (or genes), and these strings together comprise the 24 or 48-bit *Genome* (Riolo, 1992). This combination of parameters of an individual is fed into the program for 30 runs so that the range of *Fitness*, which depends somewhat on the randomly assigned values of color and priority to the trucks, can be averaged out. The GA was run on four test sets:

1. Each one of the 20 colors has the same probability of occuring. One truck comes out of the line per time step (one time step represents one minute). The probability per time step that one paint booth goes down is set to 0.05.

2. Same as test set 1 except that an alternate distribution of colors was created in which 70% of the trucks had to be painted in black (color 0), 15% in white (color 1), 7% in red (color 2), and 4% in blue (color 3), with a random uniform distribution of 16 other colors for the remaining 4% of the trucks.

3. Same as test set 2 except that two trucks come out of the line at every time step.

4. Same as test set 3 except that the probability per time step that one paint booth goes down is set to 0.25 (on average one booth goes down every fourth time step).

For the first 50 generations the *Fitness* is defined as 1000 minus the number of time steps over 420 needed until the last truck is painted during the 30 test runs. In the remainder of the generations the *Fitness* is this number minus the number of flushes needed by the booths during the 30 test runs. The idea behind this fitness function is that it is most necessary that the trucks be painted quickly, and that being accomplished, we next want to minimize the amount of wasted paint (although, of course, the two aspects are not unrelated).

After all of the individuals are tested and their *Fitnesses* assigned, a tournament is staged to decide which individuals pass on their genes to the next generation. Two individuals are chosen from the population at random and the fitnesses of each compared. The more fit of the two individuals is copied into the new population with a 75% chance, otherwise the less fit of the two is copied. There are as many tournaments as there are individuals in the population so that the population size remains constant.

Following the tournament the new population is modified. Again two individuals are picked at random. There is a 75% chance that crossover will occur between their genomes. If there will be a crossover event, a uniformly distributed random number ranging from 0 and the length of the genome is chosen as the point of crossover, then the genes (single bits) of the individuals' genomes are exchanged from the beginning of their *Genomes* to that point. A small likelihood of mutation is also introduced. There is a 0.3% chance that any bit in any individual is changed from 1 to 0, or vice versa.

MarketBased
$P = 91.8106$
$L = 4.09477$
$C = 1791.99$
Ant Based
$\alpha = 617.188$
$\beta = 4.66797$
$\xi = 7.85156$
$\phi = 17.7344$
$\theta_{min} = 5.50781$
$\theta_{max} = 39.6875$

Table 2. Best parameters found by the GA for the market-based and the ant-based algorithms.

6 RESULTS

The GA was run for 100 generations for each algorithm (market-based and ant-based). The best and average fitnesses increased significantly during the GA search. The genomes of the most fit individuals map to the parameter values summarized in Table 2.

The performance of the best market-based and ant-based algorithms are compared on several configurations. We consider the use of storage to be the failure of the system, so in all tables presented in this section, we show data starting from the lowest necessary number of booths without storage (that is, storage was never used in any of the simulations performed with this number of booths). Table 3 shows the comparison of the two algorithms on test set 1. The two algorithms exhibit comparable performance with respect to makespan but the ant-based algorithm performs significantly better in terms of the number of flushes (t–test, df=998, $P<0.01$). Figure 2 shows the full histogram of flushes for the ant-based algorithm. As expected, the required number of flushes increases as the number of booths decreases.

Results obtained on test set 2 are presented in Table 4. Here both algorithms exhibit comparable performance with respect to both the makespan and the number of flushes (t–test, df=998, $P>0.05$). Figure 3 shows the color number (0 to 19) for 4 randomly selected booths out of 8. Number 21 corresponds to a booth that is inoperable. A spike is added to the curve each time a truck is processed. As can be seen, certain booths specialize in the most frquent color (0), while others are more polyvalent but still tend to process trucks with low color codes (the most common).

Table 5 shows the results obtained on test set 3. Here again both algorithms exhibit similar performance with respect to the makespan (t–test, df=998, $P>0.05$), but the ant-based algorithm performs better with respect to the number of flushes (t–test, df=998, $P<0.01$). Obviously more booths are needed since there is a physical limit to the number of trucks that can be processed. If each truck required a paint flush it would take 6 minutes to paint a truck, and therefore require 6 booths to keep up with the trucks as they come out of the line. While it really does not require

Test set 1								
	Market-Based				Ant-Based			
	Time		Flushes		Time		Flushes	
Booths	Avg.	Std. Dev.	Avg.	Std. Dev	Avg.	Std. Dev.	Avg.	Std. Dev
8	5.21	3.02	326.82	10.85	5.20	3.44	315.65	16.19
10	3.01	1.13	298.39	11.17	2.88	0.87	260.96	11.89
12	2.72	1.04	263.52	13.15	2.60	1.13	220.06	12.18
15	2.27	1.31	211.49	13.74	2.14	1.34	162.12	12.92

Table 3. Comparison of the two algorithms on test set 1. *Time* is the number of time steps after the last truck comes out of the line required to have painted all of the trucks. The time stops when a booth begins to paint the last truck, which is why some values are less than the paint time, t_p = 3 *min*. *Flushes* is the number of flushes required in one run of the program. The average and standard deviation are computed for a sample size of 1000.

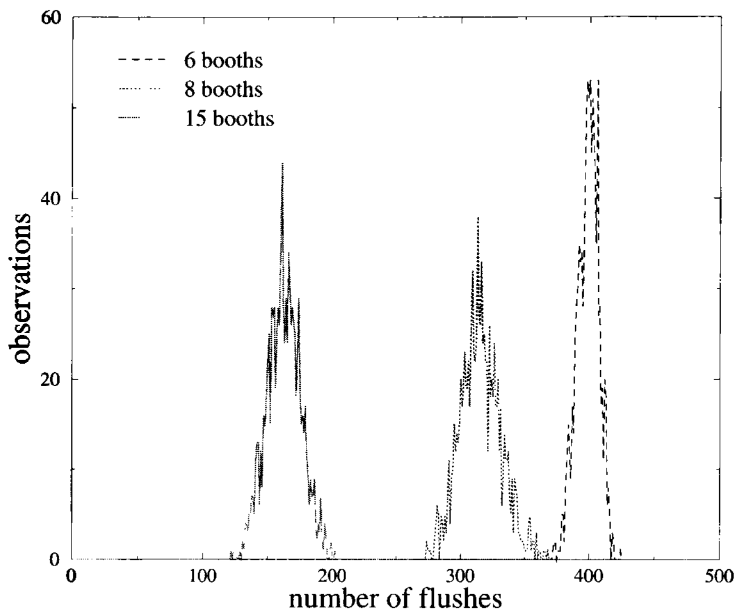


Figure 2. Histogram of flushes over 1000 runs for 3 different numbers of booths: from left to right, 15, 8, and 6.

	Test set 2							
	Market-Based				Ant-Based			
	Time		Flushes		Time		Flushes	
Booths	Avg.	Std. Dev.	Avg.	Std. Dev	Avg.	Std. Dev.	Avg.	Std. Dev
6	5.94	6.80	122.94	16.13	6.82	7.86	114.46	21.95
7	1.82	2.13	95.72	12.87	2.24	1.76	77.52	12.36
8	1.19	1.95	79.15	11.29	1.81	1.58	61.94	9.53

Table 4. Same comparison as in Table 3, but on test set 2.

	Test set 3							
	Market-Based				Ant-Based			
	Time		Flushes		Time		Flushes	
Booths	Avg.	Std. Dev.	Avg.	Std. Dev	Avg.	Std. Dev.	Avg.	Std. Dev
11	4.03	3.19	161.10	20.16	3.43	1.75	131.60	19.67
12	2.80	2.95	136.15	17.30	2.74	1.33	102.61	14.27
13	2.26	2.48	119.09	15.75	2.41	1.34	88.03	12.55

Table 5. Same comparison as in Table 3, but on test set 3.

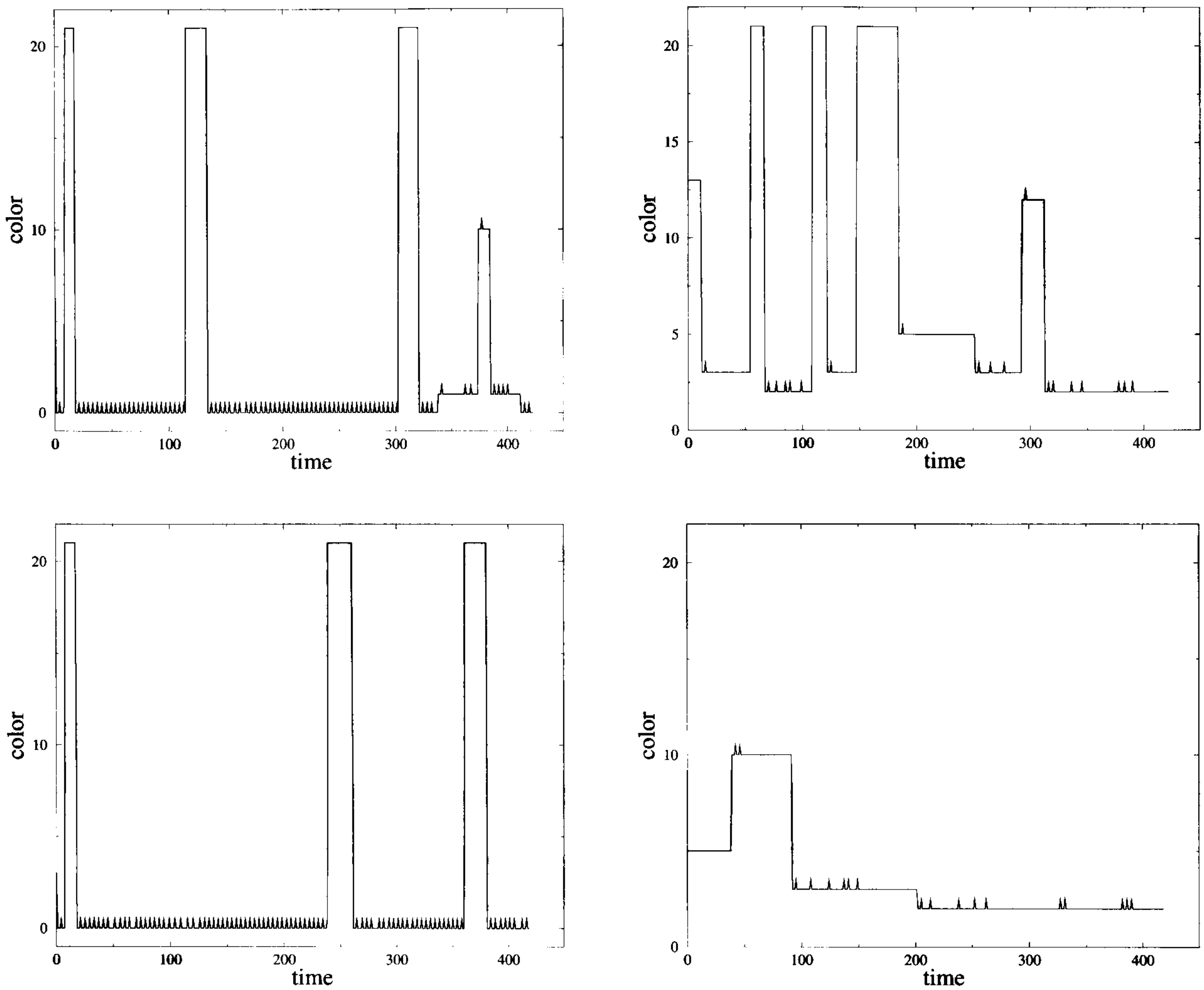


Figure 3. Color number as a function of time for 4 randomly selected booths (test set 2). Number 21 represents a booth that is inoperable and a spike is added to the curve each time a truck is processed.

Test set 4								
	Market-Based				Ant-Based			
	Time		Flushes		Time		Flushes	
Booths	Avg.	Std. Dev.	Avg.	Std. Dev	Avg.	Std. Dev.	Avg.	Std. Dev
13	6.49	6.03	226.32	27.29	6.20	5.96	208.00	30.64
14	3.72	3.02	194.46	19.01	3.50	1.71	170.41	18.10
15	3.06	2.62	176.31	16.90	3.06	1.53	150.02	15.02

Table 6. Same comparison as in Table 3, but on test set 4.

this many booths since some trucks will be painted without a paint change, we must also consider the effects of booths going down.

Table 6 shows the results obtained on test set 4. Both algorithms exhibit similar performance with respect to both the makespan and the number of flushes.

In all of the examples, when the ant-based algorithm performs significantly better, it is mostly with respect to the number of flushes, which results from the specialization of paint booths. But specialization could also be added to the market-based algorithm, although with a different vocabulary. Again, the comparison was aimed at showing how similar the two approaches are rather than at proving that one is better than the other. The results of the comparison itself should not be considered significant in that a lot of *ad hoc* choices were made.

7 CONCLUSION

In conclusion we have shown that a model of task allocation in social insects could serve as a dynamic scheduling algorithm. The parallel between our ant-based algorithm and a market-based algorithm was made explicit. For example, a black paint booth does not bid high for white trucks but may still get a white truck if no other better-positioned paint booth is available; in a polymorphic ant colony, majors do not perform brood care unless minors are not doing it. These results suggest that the social insect metaphor, which is powerful for static combinatorial optimizationproblems and for routing in communications networks (Dorigo *et al.*, 1996; Dorigo and Gambardella, 1997; Shoonderwoerd *et al.*, 1998; Bonabeau *et al.*, 1998), may also be efficient at dynamic optimization, an assumption often made but rarely tested. More testing is required to determine whether the ant-based algorithm is really efficient, especially given our lack of knowledge about the competing approach and the number of *ad hoc* choices made in this paper. One thing is interesting, however: the ant-based algorithm (as well as the market-based algorithm for that matter) does perform better than the fixed statistics-based scheduling approach used previously by the truck factory, for which the number of flushes ended up being almost as large as the number of trucks to be painted because of incidents and glitch propagation.

We do not believe that one particular agent-based approach is intrinsically better than another one, especially when they basically implement the same features with different names. We do believe, however, that thinking in terms of agents is a big advantage in distributed, heterogeneous problems, and that thinking in terms of ant-like agents (with division of labor and learning) may sometimes be more fruitful than other types of agents, even if in the end, the same processes happen to be implemented.

Another perspective on this work is also possible. Task allocation can be seen as a scheduling problem, which is continually solved by ants in a variable environment. One major difference between a market and an insect colony is the role that evolution has played in shaping social insect colony organization. Evolutionary theory suggests that solutions found by ants may be close to global optimality if the scheduling formulation is relevant to the behavior of ants. Auction protocols, on the other hand, have been designed by man to generate optimal resource allocation: why not use evolutionary algorithms to produce optimal auction protocols?

ACKNOWLEDGMENTS

M. C. was supported by the National Science Foundation REU's program during this work. E. B. is supported by the Interval Research Fellowship at the Santa Fe Institute.

REFERENCES

Bonabeau, E., G. Théraulaz, and J.-L. Deneubourg. Quantitative Study of the Fixed Threshold Model for the Regulation of Division of Labour in Insect Societies." *Proc. Roy. Soc. London B* **263** (1996): 1565–1569.

Bonabeau, E., A. Sobkowski, G. Théraulaz, and J.-L. Deneubourg. "Adaptive Task Allocation Inspired by a Model of Division of Labor in Social Insects." In *Bio-Computation and Emergent Computing*, edited by D. Lundh, B. Olsson, and A. Narayanan, 36–45. Singapore: World Scientific, 1997.

Bonabeau, E., G. Théraulaz, and J.-L. Deneubourg. "Fixed Response Thresholds and the Regulation of Division of Labor in Insect Societies." *Bull. Math. Biol.* **60** (1998): 753–807.

Bonabeau, E., M. Dorigo, and G. Théraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford, UK: Oxford University Press, 1999.

Clearwater, S. H., and B. A. Huberman. "Thermal Markets for Controlling Building Environments." *Energy Eng.* **91** (1994): 26–56.

Clearwater, S. H. *Market-Based Control: A Paradigm for Distributed Resource Allocation.* Singapore: World Scientific, 1995.

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

Dorigo, M., and L. Gambardella. "Ant colony system: a cooperative learning approach to the traveling salesman problem." *IEEE Trans. Evol. Comp.* **1** (1997): 53–66.

Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning.* Reading, MA: Addison-Wesley, 1989.

Huberman, B. A., and T. Hogg. "Distributed Computation as an Economic System." *J. Econom. Perspect.* **9** (1995): 141–152.

Kurose, J. F., and R. Simha. "A Microeconomic Approach to Optimal Resource Allocation in Distributed Computer Systems." *IEEE Trans. Computers* **38** (1989): 705–717.

Morley, R. "Painting Trucks at General Motors: The Effectiveness of a Complexity-Based Approach." In *Embracing Complexity: Exploring the Application of Complex Adaptive Systems to Business,* 53–58. Cambridge, MA: The Ernst & Young Center for Business Innovation, 1996.

Morley, R., and G. Ekberg. "Cases in Chaos: Complexity-Based Approaches to Manufacturing." *In Embracing Complexity: A Colloquium on the Application of Complex Adaptive Systems to Business,* 97–702. Cambridge, MA: The Ernst & Young Center for Business Innovation, 1998.

Oster, G., and E. O. Wilson. *Caste and Ecology in the Social Insects.* Princeton, NJ: Princeton University Press, 1978.

Riolo, R. L. "Survival of the Fittest Bits." *Sci. Am.* **267** (1992): 114–116.

Robinson, G. E. "Regulation of Vision of Labor in Insect Societies." *Annu. Rev. Entomol.* **37** (1992): 637–665.

Schoonderwoerd, R., O. Holland, and J. Bruten "Ant-based load balacing in telecommunications networks." *Adaptive Behav.* **5** (1996): 169-207.

Théraulaz G., S. Goss, J. Gervet, and J.-L. Deneubourg. "Task Differentiation in *Polistes* Wasp Colonies: A Model for Self-Organizing Groups of Robots." In *Proceedings First International Conference on Simulation of Adaptive Behavior: From Animals to Animats,* edited by J.-A. Meyer and S. W. Wilson, 346–355. Cambridge, MA: MIT Press, 1991.

Théraulaz, G., E. Bonabeau, and J.-L. Deneubourg. "Threshold Reinforcement and the Regulation of Division of Labour in Insect Societies." *Proc. Roy. Soc. London B* 265 (1998): 327–335.

Tofts, C., and N. R. Franks. "Doing the Right Thing:\enskip Ants, Honey Bees and Naked Mole-Rats." *Trends Ecol. Evol.* **7** (1992): 346–349.

Waldspurger, C. A., T. Hogg, B. A. Huberman, and J. O. Kephart. "Spawn: A Distributed Computational Economy." *IEEE Trans. Softw. Engineer.* **18** (1992): 103–117.

Walsh, W. E., M. P. Wellman, P. R. Wurman, and J. K. MacKie-Mason. "Some economics of market-based distributed scheduling." In *Eighteenth Int. Conf. on Distributed Computing Systems,* 325–332.

Wellman, M. P. "A market-oriented programming environment and its application to distributed multicommodity flow problems." *J. Artif. Intel. Res.* **1** (1993): 1–23.

Wellman, M. P. "The economic approach to artificial intelligence." *ACM Computing Surveys* (1995): 360–362.

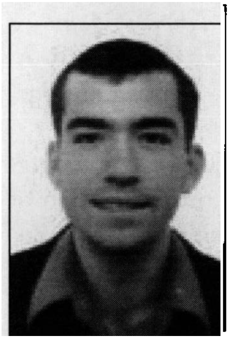
Wilson, E. O. *The Insect Societies.* Cambridge, MA: Harvard University Press, 1971.

Wilson, E. O. "The Relation Between Caste Ratios and Division of Labour in the Ant Genus *Pheidole* (Hymenoptera: Formicidae)." *Behav. Ecol. Sociobiol.* **16** (1984): 89–98.

Ygge, F., and J. M. Akkermans. "Making a case for multi-agent systems." In *Multi-Agent Rationality* (Boman, M., and W. Van de Velde, eds), 156-176. Springer Verlag, Berlin.

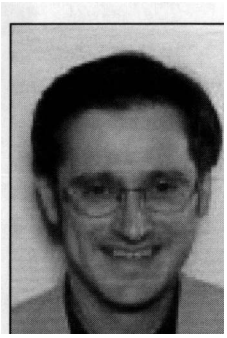
Ygge, F. *Market-Oriented Programming and its Application to Power-Load Management.* PhD Dissertation, Lund University, Sweden, 1998.

ABOUT THE AUTHORS



Mike Campos

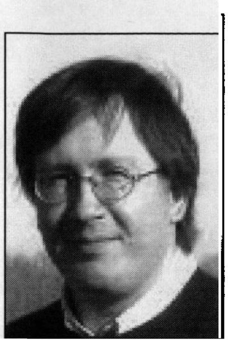
Michael Campos recently graduated with a B.A. in integrated science from Northwestern University and he plans to begin a Ph. D. program in Computation and Neural Systems at the the California Institute of Technology after the conclusion of his current year-long engagement as a scientist at Eurobios in Paris. As an intern at the Santa Fe Institute he worked on adaptive algorithms, then as an undergaduate at Northwestern he worked with a theoretical chemistry group to model DNA in solution, followed by a project systems neuroscience, which led to his thesis concerning reference frames utilized the primate oculomotor system.



Eric Bonabeau

Eric Bonabeau is chief scientist with Icosystem Corporation, a Paris- and Boston-based "idea incubator" that invents novel business models using the tools of complexity science. Prior to his current position, Dr. Bonabeau was an R&D engineer with France Telecom and Cadence Design Systems, and the Interval Research Fellow at the Santa Fe Institute. The work described in this paper was performed during his tenure at the Santa Fe Institute. He has written more than one hundred science articles and is the co-author of three books: Intelligence Collective, (Hermès, 1994), Swarm Intelligence in Natural and Artificial Systems (Oxford University Press, 1999) and Self-Organization in Biological Systems (Princeton University Press, 2001). Dr. Bonabeau is also co-editor-in-chief of Advances in Complex Systems and a member of the editorial and scientific committees of more than thirty international journals and conferences. A telecomunications engineer by training, Eric Bonabeau also holds a PhD in theoretical physics from the University of Paris at Orsay.

ABOUT THE AUTHORS



Guy Théraulaz

Guy Theraulaz received a Ph.D. in ethology in 1991 from the Provence University, (Marseille, France). He is a Research Associate with the French Centre National de la Recherche Scientifique (CNRS), and is currently working at the Ethology and Animal Cognition Laboratory, Paul Sabatier University in Toulouse. His research interests include patterns formation and collective decision-making in animal societies, and distributed adaptive algorithms inspired by social insects. In 1996 he was awarded by the bronze medal of the CNRS for his work on Swarm Intelligence. He is the co-author of Swarm Intelligence (Oxford University Press, 1999) and Self-organization in biological systems (Princeton University Press, 2001).

Jean-Louis Deneubourg

Jean-Louis Deneubourg is a Research Fellow with the Belgian Fund for Scientific Research (FNRS) at the Center for Non-Linear Phenomena and Complex Systems, Universite Libre de Bruxelles, Belgium. He is also a Professor of Behavioral Ecology at this university, from which he earned his PhD in sciences in 1979. He is the author of over 170 papers. His research interests include the dynamics of social organization in animals and the role of individual behavior and self-organization in pattern formation and decision making.