Improved Routing Wasps for Distributed Factory Control

Vincent A. Cicirello and **Stephen F. Smith**
The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{cicirello, sfs}@cs.cmu.edu

Abstract

Agent-based approaches to manufacturing scheduling and control are attractive because they offer increased robustness against the unpredictability of factory operations. Previously, we introduced a new approach to coordinating factory routing and scheduling based on a computational model of wasp behavior. The natural multi-agent system of the wasp colony is highly effective in self-organizing the allocation of tasks necessary to fulfill the needs of the nest and has proven useful as an effective model upon which to base our distributed approach to factory control. In this paper, we improve upon our original formulation of the routing wasp with the addition of a tournament of dominance contests among routing wasps competing for the same job. We experimentally evaluate this improved performance.

1 Introduction

There are many examples of effective, adaptive behavior in natural multi-agent systems [Fitzgerald and Peterson, 1988; Gordon, 1996; Kirchner and Towne, 1994; Theraulaz *et al.*, 1991], and computational analogies of these systems have served as inspiration for multi-agent optimization and control algorithms in a variety of domains and contexts (e.g., [Beckers *et al.*, 1994; Bonabeau *et al.*, 2000; Cicirello and Smith, 2001a; Dorigo and Di Caro, 1999; Schoonderwoerd *et al.*, 1997]). In [Cicirello and Smith, 2001c], we drew on aspects of a model of wasp behavior (see [Bonabeau *et al.*, 1998; Theraulaz *et al.*, 1991; 1995; 1998]) to specify a dynamic, multi-agent approach to routing and scheduling jobs through a factory. In this system, wasp-to-environment interactions govern basic task allocation (or routing) decisions; each machine in the factory has an associated *routing wasp* that decides which jobs will be accepted for processing. Alternatively, wasp-to-wasp interactions govern the order in which accepted jobs get processed by a machine; the queue in front of each machine is modeled as a colony of *scheduling wasps* which align themselves with specific jobs and interact to prioritize the queue.

In this paper, we improve upon our previous routing wasp formulation and fix a limitation that our original system possessed. We do not consider in this paper the scheduling wasps of [Cicirello and Smith, 2001c] and instead concentrate on improving the performance of the routing wasps. Previously, if two or more routing wasps simultaneously tried to route the same job to their respective machines, the winner was chosen at random. This unbiased random decision lead to deficient results. In this paper, we instead incorporate a tournament of dominance interactions to decide the winner from among competing routing wasps. Our experiments consider factories with multi-purpose machines that can be reconfigured to perform different tasks (corresponding to the processing of different job or product types). The time required to reconfigure (or changeover) a machine from the setup required to perform one type of task to the setup required for another is significant. Hence, the overall level of throughput attained depends on the ability of the factory to configure itself into specialized product lines, and minimize setup time to the extent that current demand for different job types allows.

The remainder of this paper is organized as follows. In Section 2 we summarize the wasp behavioral model that underpins our approach. In Section 3 we present our original definition of the routing wasps from [Cicirello and Smith, 2001c]. In Section 4 we introduce our improvement to this routing wasp formulation. In Section 5 and Section 6 we present experimental results. In Section 7 we analyze the behavior of the routing wasps qualitatively. Finally, in Section 8 we conclude.

2 Wasp Behavioral Model

In [Theraulaz *et al.*, 1991], Theraulaz et al. present a model for the self-organization that takes place within a colony of wasps. Interactions between members of the colony and the local environment result in dynamic distribution of tasks such as foraging and brood care. In addition, a hierarchical social order among the wasps of the colony is formed through interactions among individual wasps of the colony. This emergent social order is a succession of wasps from the most dominant to the least dominant.

The model of [Theraulaz *et al.*, 1991] describes the nature of interactions between an individual wasp and its local environment with respect to task allocation. They model the colony's self-organized allocation of tasks using what they refer to as response thresholds. An individual wasp has a response threshold for each zone of the nest. Based on a wasp's threshold for a given zone and the amount of stimulus from brood located in that zone, a wasp may or may not become engaged in the task of foraging for that zone. A lower threshold for a given zone amounts to a higher likelihood of engaging in activity given a stimulus. Bonabeau, Theraulaz, and Deneubourg discuss in [Bonabeau *et al.*, 1998] a model in which these thresholds remain fixed over time. But in [Theraulaz *et al.*, 1998], a threshold for a given task decreases during time periods when that task is performed and increases otherwise. Bonabeau et al. [Bonabeau *et al.*, 1997] demonstrate how this model leads to a distributed system for allocating mail retrieval tasks to a group of mail carriers. In this paper (as in our previous work of [Cicirello and Smith, 2001c]), we similarly adopt this task allocation model for our routing wasps to assign (or route) jobs to machines in a distributed factory setting.

The model of [Theraulaz *et al.*, 1991] also describes the nature of wasp-to-wasp interactions that take place within the nest. When two individuals of the colony encounter each other, they may with some probability interact with each other. If this interaction takes place, then the wasp with the higher social rank will have a higher probability of dominating in the interaction. Through such interactions as these, wasps within the colony self-organize themselves into a dominance hierarchy. In our scheduling wasp definition of [Cicirello and Smith, 2001c], we used this concept to model job priority and to prioritize jobs in a given machine queue. In this paper we do not consider these scheduling wasps, but instead use the concept of social dominance to determine which from among a group of routing wasps competing for the same job wins.

3 Routing Wasps

Each machine in our system has an associated routing wasp (see Figure 1 for illustration). Each routing wasp is in charge of leaving its nest and returning with jobs for the machine within its nest to process. Each wasp has a set of response thresholds:

$$\Theta_w = \{\theta_{w,0}, \ldots, \theta_{w,J}\} \tag{1}$$

where $\theta_{w,j}$ is the response threshold of wasp w to jobs of type j. Each wasp only has response thresholds for job types that its associated machine can process.

Jobs in the system that are not currently queued on a machine broadcast to all of the routing wasps a stimulus S_j which is equal to the length of time the job has been waiting to be routed and where j is the type of job. So the longer the job remains unrouted, the stronger the stimulus it emits. Provided that its associated machine is able to process job type j, a routing wasp w will pick up a job emitting a stimulus S_j with probability:

$$P(\theta_{w,j}, S_j) = \frac{S_j^2}{S_j^2 + \theta_{w,j}^2} \tag{2}$$

This is essentially the rule used for task allocation in the model as described in [Bonabeau *et al.*, 1997; Theraulaz *et al.*, 1998]. In this way, wasps will tend to pick up jobs of the type for which its response threshold is lower. But will pick up jobs of other types if a high enough stimulus is emitted.

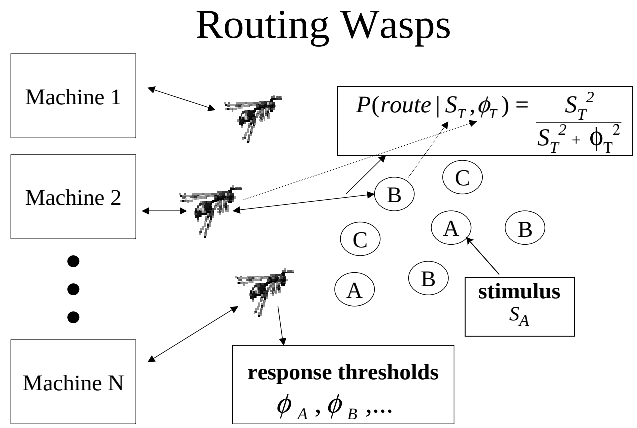

Figure 1: Routing wasps

The threshold values $\theta_{w,j}$ may vary in the range $[\theta_{min}, \theta_{max}]$. Each routing wasp maintains a communications channel to the nest. At all times, it knows what job type the machine is currently processing. This knowledge is used to adjust the response thresholds for the various job types. This updating of the response thresholds occurs periodically. If the machine back at the nest is currently processing job type j or is in the process of setting up to process job type j, then $\theta_{w,j}$ is updated according to:

$$\theta_{w,j} = \theta_{w,j} - \delta_1 \tag{3}$$

If the machine back at the nest is either processing or setting up to process a job type other than j, then $\theta_{w,j}$ is updated according to:

$$\theta_{w,j} = \theta_{w,j} + \delta_2 \tag{4}$$

And if the machine back at the nest is currently idle and has an empty queue, then for all job types j that the machine can process the wasp adjusts the response thresholds $\theta_{w,j}$ according to (t is the length of time the machine has been idle):

$$\theta_{w,j} = \theta_{w,j} - \delta_3^t \tag{5}$$

In this way, the response thresholds for the job type currently being processed are reinforced as to encourage the routing wasp to pick up jobs of the same type. This specialization of wasp nests helps to minimize setup time. The first two ways in which the response thresholds are updated (equations 3 and 4) are analogous to that of the model described in [Bonabeau *et al.*, 1997; Theraulaz *et al.*, 1998]. The third (equation 5) is to encourage a wasp associated with an idle machine to take whatever jobs it can get.

4 Dominance Struggle

The routing wasp formulation of Section 3 did not state what happens if two or more routing wasps respond positively to the same job stimulus. Previously, in [Cicirello and Smith, 2001c], this decision was made at random. But there is a problem with this. Consider the case where one machine has been sitting idle and has an empty job queue. Perhaps this machine has specialized to a job type whose demand has diminished. Now consider a second machine with a long queue of jobs. Suppose a new job arrives at the factory of the type for which this second machine has developed a preference. The idle machine by this point is willing to take any job. Both machines respond to the stimulus from this new job. Previously, both machines would have an equal chance of taking on this new job. But perhaps the idle machine with the empty queue should have a higher probability of getting the job even though it is not currently configured for this job type.

To this end we now define a new method for deciding which routing wasp from a group of competing wasps gets the job. This method is based on the self-organized social hierarchies of real wasps. First define the force F_w of a routing wasp w as:

$$F_w = 1.0 + T_p + T_s \tag{6}$$

where T_p and T_s are the sum of the processing times and setup times of all jobs currently in the queue of the associated machine, respectively. Now consider a dominance struggle between two competing routing wasps. This contest determines which routing wasp gets the job. Let F_1 and F_2 be the force variables of routing wasps 1 and 2, respectively. Routing wasp 1 will get the job with probability:

$$P(F_1, F_2) = \frac{F_2^2}{F_1^2 + F_2^2} \tag{7}$$

In this way, routing wasps associated with machines of equivalent queue lengths will have equal probabilities of getting the job. If the queue lengths differ, then the routing wasp with the smaller queue has a better chance of taking on the new job. In the event that more than two routing wasps compete for a given job, a single elimination tournament of dominance contests is used to decide the winner.

5 Experimental Design

All of the experiments that are presented here were performed in a simulated factory environment implemented in Java and executed on a Pentium III running Linux 5.2. All experiments consider factories which produce two products (henceforth, Job Type A and Job Type B) and multi-purpose machines that can process either of the two product types (only single stage jobs are considered here). Experiments with two and four machines are studied. In all cases, setup time to reconfigure a machine for the alternate job type is 30 time units. Each machine can only process jobs located in its personal queue in a first-in-first-out order and the task of routing to these queues is performed by the routing wasps. When a new job is generated, its process time is 15 plus a Gaussian noise factor.

Jobs are released to the factory floor dynamically according to four different product mixes (3 static and 1 changing). In each, arrival rates are defined by the probability a new job of each type is released during a given time unit. The arrival rates for the two machine problems are as follows (to get the rates for the four machine problems simply multiply these rates by 2):

- 50/50 mix: P(Job Type A) = 0.05, P(Job Type B) = 0.05
- 85/15 mix: P(Job Type A) = 0.0857, P(Job Type B) = 0.0143
- 100/0 mix: P(Job Type A) = 0.133, P(Job Type B) = 0.0

Table 1: Average throughput for different job mixes. 95% confidence intervals and two-tailed p-values from paired T tests are shown.

Two Machine Problem			
	R-Wasps	R-Wasps-2	p-value
50/50 mix	492.82±4.33	492.82±4.33	–
85/15 mix	470.47±2.20	474.00±2.24	<0.0001
100/0 mix	605.37±2.92	624.92±1.65	<0.0001
Changing	444.72±2.96	448.53±2.32	0.0058
Four Machine Problem			
	R-Wasps	R-Wasps-2	p-value
50/50 mix	988.84±6.58	990.46±6.64	<0.0001
85/15 mix	981.89±5.98	994.03±6.03	<0.0001
100/0 mix	1234.39±3.27	1255.39±1.95	<0.0001
Changing	879.25±5.47	899.62±4.02	<0.0001

- Changing mix: For the first half of the simulation P(Job Type A) = 0.0857, P(Job Type B) = 0.0143, then for the second half of the simulation P(Job Type A) = 0.0143, P(Job Type B) = 0.0857

These arrival rates correspond approximately to medium-to-heavily loaded factories.

The values of the various parameters of the system are the following (for all experiments):

- $\theta_{min} = 1$
- $\theta_{max} = 100$
- $\delta_1 = 0.2$
- $\delta_2 = 0.1$
- $\delta_3 = 1.001$

No special tuning process is behind these settings. A few different values were tried and those that seemed to give adequate performance were chosen. A more systematic study of the effects of these parameters is needed.

For each job mix, 100 simulations with different arrival sequences were used to compare the routing wasps of this paper (R-Wasps-2) to the original routing wasp formulation (R-Wasps) of [Cicirello and Smith, 2001c]. Each simulation was 5000 time units in length. The time units correspond to nothing in particular so consider them minutes, hours, or whatever other time unit you fancy. Average results are now presented in Section 6.

6 Experimental Results

In Table 1 we see average throughput (number of jobs processed) results for the various job mixes for both the two machine and four machine problems. For the 50/50 job mix in the two machine case, we see no difference between the behavior of R-Wasps and R-Wasps-2. Each produces the exact same results on all 100 simulations. In this case there are on average equal numbers of both job types and the shop is fairly heavily loaded so each of the two machines is content to specialize to one job type. Due to this, the routing wasps never have to compete for a desired job and thus the system behavior is the same in both cases.

Table 2: Average throughput for different job mixes. 95% confidence intervals and two-tailed p-values from paired T tests are shown.

Two Machine Problem			
	AC^2	R-Wasps-2	p-value
50/50 mix	492.52±4.32	492.82±4.33	0.4142
85/15 mix	403.02±1.89	474.00±2.24	<0.0001
100/0 mix	606.15±2.84	624.92±1.65	<0.0001
Changing	399.51±2.96	448.53±2.32	<0.0001

However, for all other job mixes, R-Wasps-2 outperforms R-Wasps in terms of throughput. Paired T tests were performed and in all cases R-Wasps-2 was seen, with statistical significance, to process more jobs on average than R-Wasps. The dominance contests of R-Wasps-2 result in improved factory routing performance.

In Table 2 we compare the R-Wasps-2 of this paper to the Ant Colony Control (AC^2) system of [Cicirello and Smith, 2001a]. AC^2 is an adaptive multi-agent shop floor routing and control system that is based on an analogy to ant trail following behavior. It is robust but has a tendency to converge upon deficient equilibria. We feel this is at least partially due to all control decisions being made by agents acting for jobs with a highly limited view of the environment. The local environment of these ant-like agents include the job they are immediately in charge of and the machines which can process that job. In contrast, R-Wasps-2 moves all control decisions to the level of the machine. From the machine perspective, each wasp-like agent of this system is aware of the contents of the queue of its associated machine as well as all jobs not currently in any queue. So in some respect, R-Wasps-2 is capable of making more informed control decisions. In all cases except for the 50/50 job mix, R-Wasps-2 outperforms AC^2 with statistical significance. In the 50/50 job mix, no significant difference in performance has been seen. In the 85/15 job mix and the changing job mix, R-Wasps-2 far outperforms AC^2. In game-theoretic terms, AC^2 is converging upon deficient equilibria in these cases; whereas R-Wasps-2 tends to converge upon "better" equilibria.

7 Routing Wasp Behavioral Analysis

In [Balch, 2000] Balch introduced the concept of hierarchic social entropy (HSE) as a measure of diversity among teams of robotic agents. HSE combines the concepts of information entropy and the taxonomic tool known as a dendrogram to give a numerical measure of the diversity of a group of agents that cannot simply be classified as either alike or not alike.

In Table 3, we use HSE to examine the behavior of the routing wasps of our system. We use the Euclidean distance between the vectors of response thresholds normalized to lie between 0 and 1 as the measure of distance between two routing wasps on the similarity scale. The Table shows the average HSE at the end of the 5000 time unit simulations for both R-Wasps and R-Wasps-2. It also lists the information entropy of the job mix itself. In the single job type case (i.e., the 100/0 job mix) we see that the HSE of the routing wasps is very near 0.0 which corresponds to all machines specialized to the one job type currently arriving at the factory. In the 50/50 job mix case, the HSE is near 1.0 which corresponds to half of the machines specialized to one job type and the other half specialized to the other type. Both of these cases correspond to the entropy of the job mix itself and seem to correspond to what intuitively is the optimal thing to do. In the 85/15 job mix and changing mix, it is less clear whether or not the hierarchic social entropy should closely correspond to the information entropy of the job mix. The answer really depends on how loaded the shop we are dealing with is. In the case of a fairly heavily loaded shop (as in the experiments of this paper) it seems to make sense that they should to some degree correspond and that is exactly what we see in the Table. In any case, it is clear that the more diverse the job mix, the more diverse the colony of routing wasps should be; and that the less diverse the job mix, the less diverse the colony of routing wasps should be. So to this extent, the routing wasps seem to be doing the right thing.

Table 3: Average hierarchic social entropy of the routing wasps. 95% confidence intervals are shown. Also listed is the entropy of the job mix itself.

Two Machine Problem			
	R-Wasps	R-Wasps-2	entropy
50/50 mix	0.984±0.004	0.984±0.004	1.000
85/15 mix	0.681±0.006	0.666±0.009	0.610
100/0 mix	0.003±0.002	0.001±0.001	0.000
Changing	0.666±0.009	0.643±0.013	0.610
Four Machine Problem			
	R-Wasps	R-Wasps-2	entropy
50/50 mix	1.016±0.003	1.003±0.002	1.000
85/15 mix	0.597±0.007	0.588±0.006	0.610
100/0 mix	0.001±0.001	0.000±0.000	0.000
Changing	0.724±0.018	0.604±0.015	0.610

In Figure 2, we see plots of the response thresholds of the routing wasps for various job mixes and two machines (the results are for R-Wasps-2). These results are averages of 100 runs. The first column is the response thresholds for job type A and the second column for job type B. The rows, respectively, are for the 50/50, 85/15, 100/0, and changing job mixes. In the 50/50 job mix, we see that each machine specializes to a different job type. In the 100/0 job mix, we see that both machines quickly adapt their configurations to that associated with the single job type in the system. For the 85/15 job mix, one machine specializes to the job type for which there are more jobs, and the other machine is willing to take either job type. The first half of the changing job mix simulations corresponds to that of the 85/15 job mix; while in the second half we see the machines changing roles to handle the new 15/85 mix. If we examine similarly the four machine problems in Figure 3 we find the same sort of behavior. That is, in the 100/0 job mix all machines specialize in the single job type, in the 50/50 job mix half of the machines specialize to each job type, and in the 85/15 job mix all machines are willing to take the job type of higher demand while only one

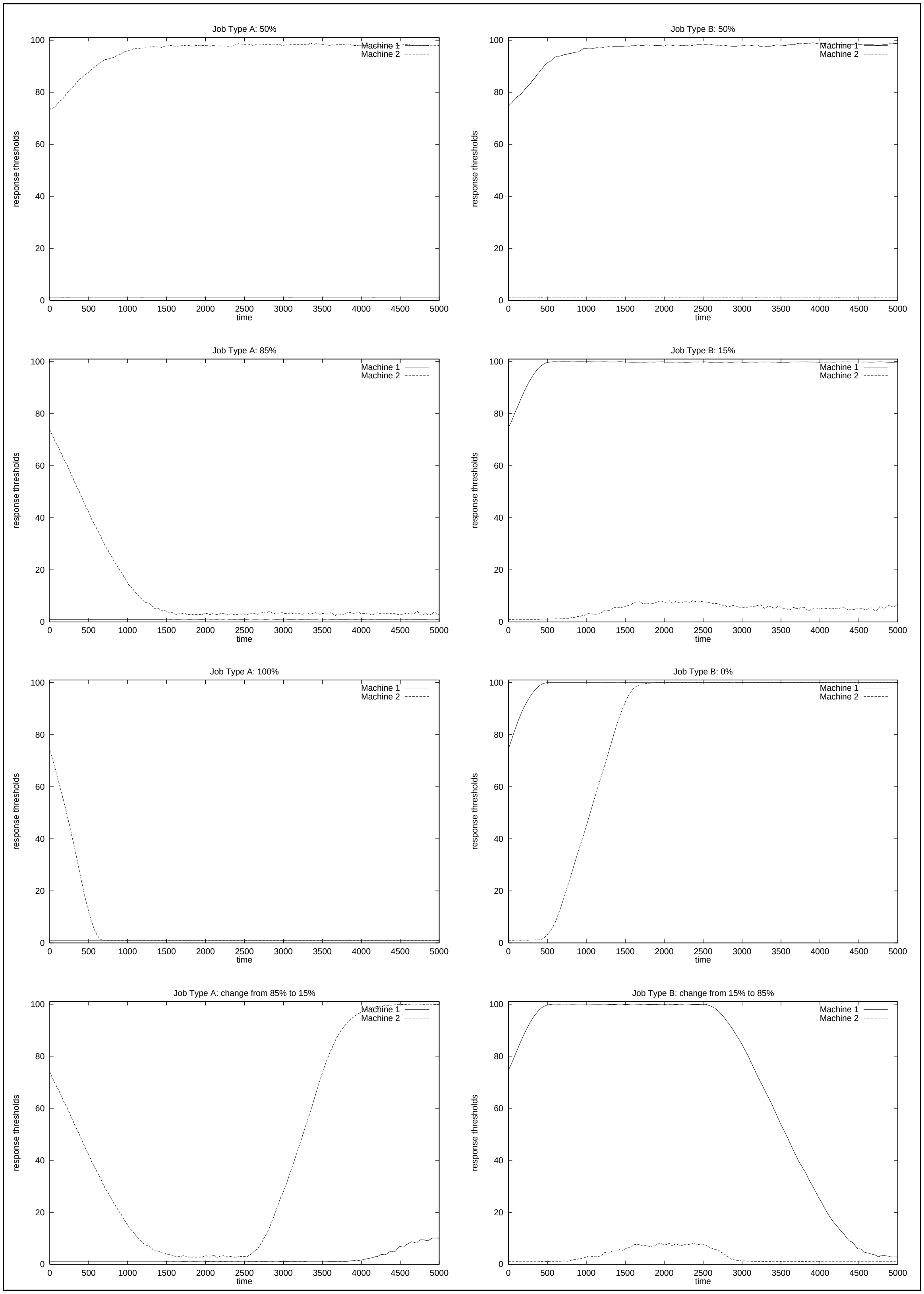

Figure 2: Plots of the average response thresholds over time of the routing wasps for different job mixes and two machines.

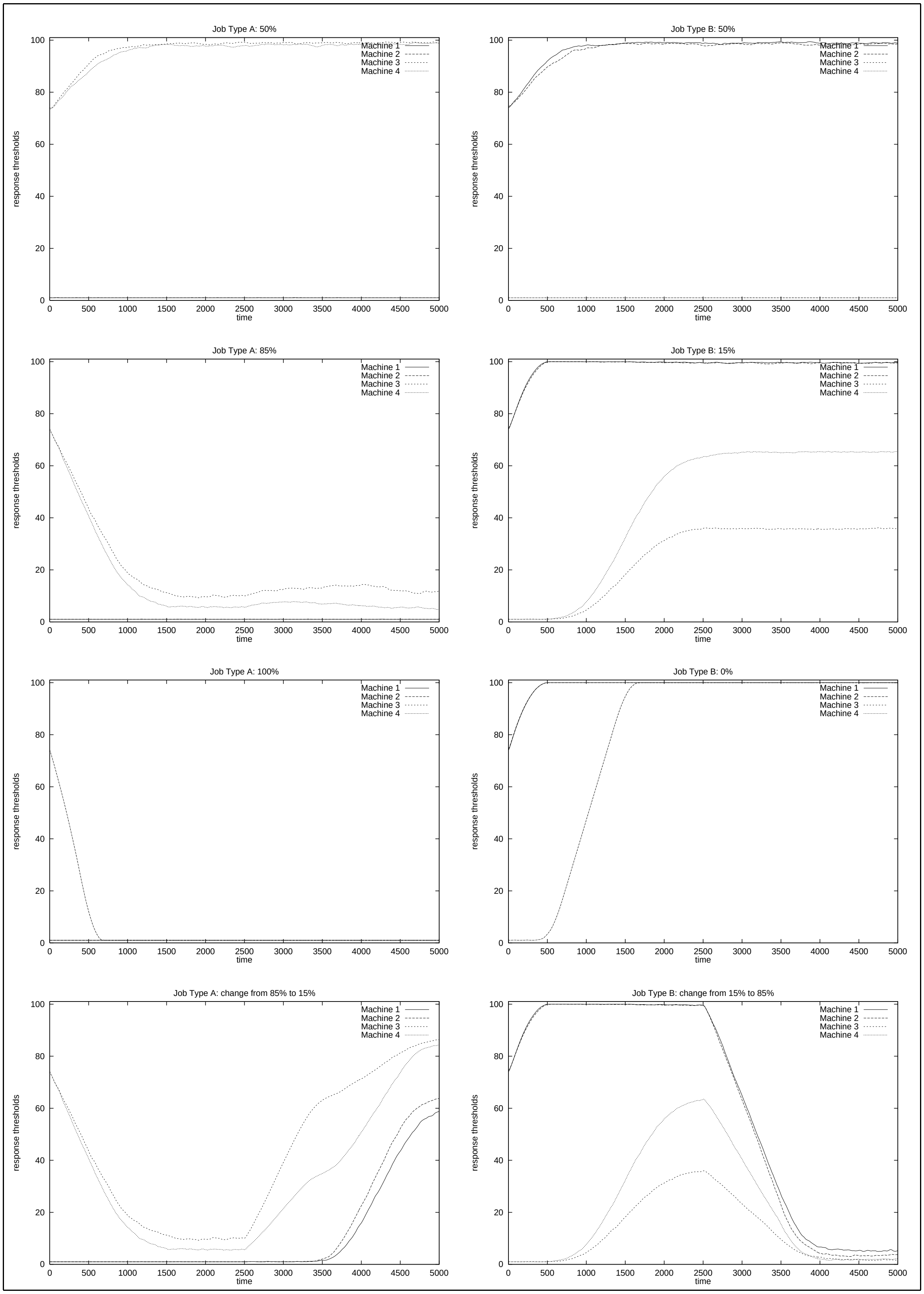

Figure 3: Plots of the average response thresholds over time of the routing wasps for different job mixes and four machines.

has a strong interest in the other job type. This corresponds, intuitively, to the behavior the system should exhibit for "optimal" performance.

8 Conclusion

In this paper we have discussed our distributed approach to dynamic factory routing based on various aspects of a model of the adaptive behavior observed in wasp colonies. In our system, routing activities are performed by wasp-like agents in a manner analogous to task allocation among real wasps. We have introduced an improvement to our previous routing wasp formulation that chooses from among a group of routing wasps competing for a single job via a tournament of dominance contests modeled after the self-organized social hierarchies observed in real wasp colonies.

We plan in the future to examine the performance of our routing wasps in more complex factory environments (e.g. multi-stage jobs, more machines, more job types, etc.). We also are in the process of further extending our scheduling wasps (see [Cicirello and Smith, 2001c]) to handle such constraints as duedates and weights and the weighted tardiness objective [Cicirello and Smith, 2001b]. Upon the completion of a proper study of the scheduling wasps under such constraints, we plan to examine the performance of the complete system including both routing and scheduling wasps in relation to dispatch scheduling heuristics and other locally distributable approaches to factory control.

Acknowledgments

This work has been funded in part by the Department of Defense Advanced Research Projects Agency and the U.S. Air Force Rome Research Laboratory under contracts F30602-97-2-0066 and F30602-00-2-0503 and by the CMU Robotics Institute. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force or U.S. Government.

References

[Balch, 2000] T. Balch. Hierarchic social entropy: An information theoretic measure of robot group diversity. *Autonomous Robots*, 8(3), July 2000.

[Beckers *et al.*, 1994] R. Beckers, O. E. Holland, and J. L. Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. In R. A. Brooks and P. Maes, editors, *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pages 181–189, 1994.

[Bonabeau *et al.*, 1997] 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 and B. Olsson, editors, *Bio Computation and Emergent Computing*, pages 36–45. World Scientific, 1997.

[Bonabeau *et al.*, 1998] 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.

[Bonabeau *et al.*, 2000] E. Bonabeau, M. Dorigo, and G. Theraulaz. Inspiration for optimization from social insect behaviour. *Nature*, 406:39–42, July 2000.

[Cicirello and Smith, 2001a] V. A. Cicirello and S. F. Smith. Ant colony control for autonomous decentralized shop floor routing. In *ISADS-2001, Fifth International Symposium on Autonomous Decentralized Systems*. IEEE Computer Society Press, March 2001.

[Cicirello and Smith, 2001b] V. A. Cicirello and S. F. Smith. Randomizing dispatch scheduling heuristics, 2001. Submitted to the AAAI Fall Symposium: Using Uncertainty within Computation.

[Cicirello and Smith, 2001c] V. A. Cicirello and S. F. Smith. Wasp nests for self-configurable factories. In *Agents 2001, Proceedings of the Fifth International Conference on Autonomous Agents*. ACM Press, May-June 2001.

[Dorigo and Di Caro, 1999] M. Dorigo and G. Di Caro. The ant colony optimization meta-heuristic. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*. McGraw-Hill, 1999.

[Fitzgerald and Peterson, 1988] T. D. Fitzgerald and S. C. Peterson. Cooperative foraging and communication in caterpillars. *BioScience*, 38(1):20–25, January 1988.

[Gordon, 1996] D. M. Gordon. The organization of work in social insect colonies. *Nature*, 380:121–124, March 1996.

[Kirchner and Towne, 1994] W. H. Kirchner and W. F. Towne. The sensory basis of the honeybee's dance language. *Scientific American*, 270(6):74–80, June 1994.

[Schoonderwoerd *et al.*, 1997] R. Schoonderwoerd, O. Holland, and J. Bruten. Ant-like agents for load balancing in telecommunications networks. In *Agents '97, Proceedings of the First International Conference on Autonomous Agents*, pages 209–216. ACM Press, 1997.

[Theraulaz *et al.*, 1991] G. Theraulaz, S. Goss, J. Gervet, and J. L. Deneubourg. Task differentiation in polistes wasp colonies: A model for self-organizing groups of robots. In *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*, pages 346–355. MIT Press, 1991.

[Theraulaz *et al.*, 1995] G. Theraulaz, E. Bonabeau, and J. L. Deneubourg. Self-organization of hierarchies in animal societies: The case of the primitively eusocial wasp polistes dominulus christ. *Journal of Theoretical Biology*, 174:313–323, 1995.

[Theraulaz *et al.*, 1998] G. Theraulaz, E. Bonabeau, and J. L. Deneubourg. Response threshold reinforcement and division of labour in insect societies. *Proceedings of the Royal Society of London B*, 265(1393):327–335, February 1998.