Wasp Nests for Self-Configurable Factories

Vincent A. Cicirello
The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
cicirello@ri.cmu.edu

Stephen F. Smith
The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
sfs@cs.cmu.edu

ABSTRACT

Agent-based approaches to manufacturing scheduling and control have gained increasing attention in recent years. Such approaches are attractive because they offer increased robustness against the unpredictability of factory operations. But the specification of local coordination policies that give rise to efficient global performance and effectively adapt to changing circumstances remains an interesting challenge. In this paper, we introduce a new approach to this coordination problem, drawing on various aspects of a computational model of how wasp colonies coordinate individual activities and allocate tasks to meet the collective needs of the nest. We focus specifically on the problem of configuring machines in a factory to best satisfy (potentially changing) product demands over time. Our system models the set of jobs queued in front of any given machine as a wasp nest, wherein wasp-like agents interact to form a social hierarchy and prioritize the jobs that they represent. Other wasp-like agents external to the nest act as overall machine proxies, and use a model of wasp task allocation behavior to determine which new jobs should be accepted into the machine's queue. We show for simple factories that our multi-agent system achieves the desired effect. For a given job mix, the system converges to a factory configuration that maximizes overall performance, and as the job mix changes, the system quickly adapts to a new, more appropriate configuration.

1. INTRODUCTION

The factory is a complex dynamic environment and manufacturing organizations are constantly faced with the need to rearrange production. New and evolving market opportunities lead to changing product demands and manufacturing priorities. Changes in resource availability affect production capacity and force reassessment of current production goals. Such changing circumstances are quite frequently at odds with attempts to build advance schedules. Though advance scheduling can provide a basis for configuring factory resources to optimize performance relative to (currently) known requirements and constraints, these prescriptive solutions also tend to be quite brittle and they can quickly become invalidated by unexpected events.

In practice, manufacturing operations are often coordinated in a decentralized manner. The use of local dispatch scheduling policies [13], for example, is commonplace in many manufacturing environments. By making decisions only when needed to keep execution going and by basing them on aspects of the current dynamic state, dispatch-based strategies are quite insensitive to unexpected events and yield very robust behavior. This advantage can also be a disadvantage, however, as decisions are made myopically and this can lead to suboptimal factory performance.

The desire for a more robust basis for coordination has similarly motivated research into agent-based approaches to manufacturing scheduling and control [10, 11, 14, 15, 17], and there have been a few interesting successes (e.g., [12]). However, these approaches are also susceptible to suboptimal and even chaotic global behavior [1]. The ability to orchestrate good global performance via local interaction protocols and strategies remains a significant and ill-understood challenge.

One approach to this class of problem is to view establishment of appropriate coordination policies as an adaptive process. There are many examples of effective, adaptive behavior in natural multi-agent systems [7, 8, 9, 20], 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., [2, 3, 6, 16]). In this paper, we draw on aspects of a model of wasp behavior previously developed in [5, 20, 18, 19] to specify a dynamic, multi-agent approach to routing and scheduling jobs through a factory. We show for simple factories consisting of multiple, multi-purpose machines: (1) that the system converges to factory configurations (i.e., specialized product flows) which maximize overall performance for a given mix of job types, and (2) that the system appropriately adapts to better configurations as the job mix changes over time. We believe that the approach is scalable and can extend naturally to encompass more complex factory environments.

The remainder of the paper is organized as follows. In Section 2 we summarize the wasp behavioral models of task allocation and social interaction that underpin our approach. In Section 3 these models are applied to specify a multi-agent system for coordinating the flow of jobs through a factory. Section 4 presents experimental results. Finally, in Section 5 we conclude and discuss future work.

AGENTS'01, May 28-June 1, 2001, Montréal, Quebec, Canada

2. WASP BEHAVIORAL MODEL

In [20], Theraulaz et al. present a model for the self-organization that takes place within a colony of wasps and suggest its use for the coordination of a group of robots. 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. Theraulaz et al. model these two aspects of wasp behavior as distinct behaviors without making any explicit connection between the two.

The model of [20] 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 [5] a model in which these thresholds remain fixed over time. But in [19], a threshold for a given task decreases during time periods when that task is performed and increases otherwise. Bonabeau et al. [4] demonstrate how this model leads to a distributed system for allocating mail retrieval tasks to a group of mail carriers. In this paper, we similarly adopt this task allocation model to assign (or route) jobs to machines in a distributed factory setting.

The model of [20] 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. Social rank is modeled by what they call the individual's force variable F. The successful wasp is chosen probabilistically based on the value of the Fermi function of the force variables of the two interacting wasps. The value of F is increased for the successful wasp and decreased for the unsuccessful. Through such interactions as these, wasps within the colony self-organize themselves into a dominance hierarchy. In [18], Theraulaz, Bonabeau, and Deneubourg discuss in greater detail the self-organization of dominance hierarchies among wasps. For example, they discuss a number of ways of modeling the probability of interaction during an encounter which range from always interacting to interacting based upon certain tendencies of the individuals. For our purposes, we use the concept of force to model job priority, and exploit this wasp-to-wasp interaction model to prioritize jobs in a given machine queue. High priority jobs correspond to high ranking wasps in the social hierarchy of the nest.

3. MACHINES AS WASP NESTS

In this section, we apply the wasp behavioral model of Theraulaz et al. to develop a multi-agent system for coordinating the flow of jobs through a factory to satisfy given

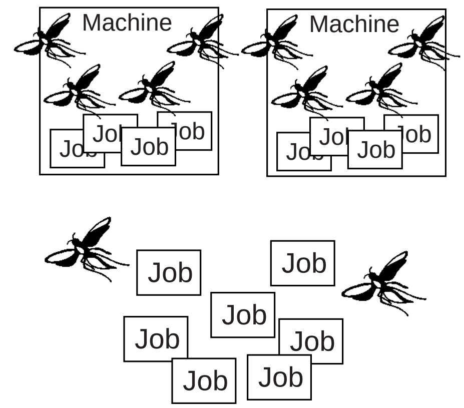

Figure 1: Factory as a collection of wasp nests. The queue of jobs for a given machine is modeled as a wasp nest. This nest contains a colony of wasp-like agents in charge of prioritizing jobs in the queue. Wasps external to the nests are in charge of routing new jobs to its associated nest.

product demands. We 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.

In their biological model, Theraulaz et al. do not draw any connection between the two types of wasp behavior that the model describes. Similarly, we have adopted each type of wasp behavior for distinct aspects of factory floor control. In our 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 jobs in the queue. Figure 1 illustrates our system. It is interesting that these distinct aspects of wasp behavior are seen to complement each other in our system, despite the lack of coupling between them in the underlying biological model.

The routing wasp maintains response thresholds for each of the job types that the machine it is associated with can process and responds to stimuli from jobs of these types. Once routed to a given nest (i.e., machine), a job sits in a queue waiting for the machine to process it. In the simplest variation of our system, machines naively process jobs first-in first-out (FIFO). In our complete system, machines still pick jobs from the front of the queue for processing, but the order of the jobs in the queue can dynamically change due to the behavior of a colony of wasp-like agents in charge of scheduling. Jobs are each assigned to individual scheduling wasps. A scheduling wasp's position in the social hierarchy of the colony is determined by the position of its associated job in the queue. Given that the machine processes jobs according to the order of this queue, the most dominant individual is the one whose job is first in line. Upon completion of a job on the machine, a series of dominance interactions may take place between the wasp whose job is next in line and wasps further back in the queue. The success of these encounters is based upon factors such as whether the jobs associated with these wasps require setup time and the position of the associated job in the queue. A wasp whose job is toward the end of the queue is weaker than one whose job is towards the beginning. But a wasp whose job does not require additional setup time is stronger than a wasp whose job does. The outcomes of these encounters are chosen probabilistically. The routing wasps and scheduling wasps are discussed in more detail in Section 3.1 and Section 3.2 respectively.

3.1 Routing Wasps

Each machine in our system has an associated routing wasp. 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 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 [4, 19] rather than the Fermi function used in [20]. 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.

The threshold values $\theta_{w,j}$ may vary in the range [θ_{min}, θ_{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:

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

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 are analogous to that of the model described in [4, 19]. The third is to encourage a wasp associated with an idle machine to take whatever jobs it can get. Even without this idle machine update rule, the routing wasp will eventually begin taking jobs it is not currently specialized for as they begin emitting greater stimuli, but this rule simply acts to enhance performance.

3.2 Scheduling Wasps

Within the nest, the machine processes jobs according to the order in its queue. However, this order is not rigid and may change at any time. When the machine completes the processing of a job, the wasps associated with jobs further back in the queue have the opportunity to challenge the wasp whose associated job is next in line for the machine. If the job that is first in line is of the same type as that of the job that has just completed, then no wasps will challenge the wasp associated with this job. This is analogous to low ranking individuals in the wasp social hierarchy being less likely to take part in a dominance interaction with high ranking individuals.

However, if the job that is first in line requires setup time, then wasps associated with jobs that are further back in the queue which would not require setup time if they were to be processed next will each in turn challenge the wasp of the first job. Actually the challenges occur asynchronously with the challenged wasp deciding if any were successful and if multiple successes occurred picking one at random. If any of these challengers are successful then the job associated with this successful wasp trades positions in the queue with the job of the wasp which lost the interaction.

In order to define the probability of a challenger winning the interaction, we must first define the force of a wasp w. As previously stated, we equate this force value with job priority. The job that should be processed next corresponds to the most dominant wasp in the colony. We wish to maximize system throughput without a detrimental effect on cycle time. To accomplish this we define force as:

$$F_w = T_w^p + T_w^s + i_w \tag{6}$$

where T_w^p is the processing time required by the job associated with wasp w, T_w^s is the setup time required by the job associated with wasp w if it was to be processed next, and i_w is the position in the queue of the job associated with wasp w. In the actual biological model of [20], force is a variable that varies in some range based on wins and losses of interactions. But in our application of the model we have chosen to impart on the wasps some domain knowledge through the force variables. In future extensions of this work we plan on considering systems with due dates and urgent jobs. Both considerations can be accounted for in a new definition of F_w.

The probability of the challenging wasp winning the dominance interaction can now be defined according to the Fermi function (as used in [20]) of the force variables but with a twist. In [20], the wasp with the higher value for its force variable is the stronger individual. But the way that force is defined here, lower values imply the stronger individual because the lower the value of the force means the less time is required for the job and the closer to the front of the queue is the job. This amounts to changing the order of the force variables within the Fermi function. Noting this, the probability of the challenging wasp c defeating the current first ranking wasp p is:

$$\mathcal{F}(F_c, F_p) = \frac{1}{1 + e^{\eta(F_c - F_p)}} \qquad (7)$$

where η is a constant. If F_c is less than F_p then the challenger has a better than 50% chance of being successful. If F_c is greater than F_p then the challenge does not occur and corresponds to weaker wasps being less likely to even challenge a more dominant wasp.

4. WASPS IN ACTION

To evaluate the performance of our proposed system, we simulate its use in coordinating operations in a simple factory under various mixes of product demands. The factory of interest in these experiments is illustrated in Figure 2. The factory consists of two identical machines, each capable of processing either of two job types. The processing of a job of either type requires 10 time units on both machines. However, if a machine is not already set up for the given job type, then 20 time units are first required to changeover the machine. Given this constraint, it is beneficial to sequence jobs of the same type together when possible. Jobs of both types arrive periodically according to pre-defined job arrival rates.

We consider problems covering three variations in job mix:

- 50/50 mix - jobs of type A and B may arrive at any given time unit with probability 0.1. To maximize throughput in this case, we would expect the factory to configure itself into two dedicated machines.
- 75/25 mix - jobs of type A may arrive at a given time unit with probability 0.1125 and jobs of type B with probability 0.0375. In this case, we would expect one machine to become dedicated and the other to service both job types.
- dynamic mix - for these problems, the mix changes from 75/25 to 25/75 over the course of the simulation period (by simply swapping the probabilities given above for the 75/25 mix part way through the simulation.)

In all cases, job arrival rates were designed in such a way as to approximate a problem for which the optimal solution would require that every job be processed with no machine idle time. Otherwise some jobs would be left unprocessed at the end of the simulation. Hence, the experiments correspond to heavily loaded factory conditions.

We consider two variations of the agent coordination and task allocation scheme outlined in Section 3. In the first, which we will refer to as R-Wasps, only the routing wasps are active. Rather than using the scheduling wasps to coordinate the sequencing of jobs on the machines, the naive policy of processing jobs in a first-in-first-out (FIFO) order

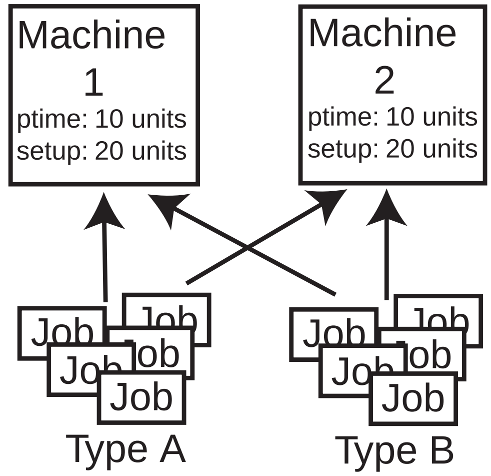

Figure 2: Two identical machines each capable of processing two job types but with setup time to switch between the two.

is used by the machines. The second scheme we consider, which we refer to as R&S-Wasps, additionally incorporates scheduling wasps to dynamically manage machine queues. In all simulations, the response thresholds were initialized to θ_{min} for the job type for which the machine is initially set and to random values in the range $[\theta_{min}, \theta_{max}]$ otherwise. This is analogous to some species of social insects in which individuals have some initial specialization. The values of various model parameters were set through trial and error and for all simulation runs were fixed as follows: $\theta_{min} = 1$, $\theta_{max} = 12$, $\delta_1 = 0.01$, $\delta_2 = 0.01$, $\delta_3 = 0.1$, and $\eta = 0.1$.

In Figure 3 we see plots of the response thresholds for the routing wasps of the R-Wasps scheme for a number of different job mixes. The plots in the first column are for job type A and the plots in the second column are for job type B. Lower values of a response threshold signify that the corresponding wasp (and associated machine) is more interested in jobs of the associated type than is the other wasp. The plots shown are averages of five 5000 time unit simulations in all cases. We can see that for the 50/50 job mix the system produces the desired effect: machine one specializes to job type A and machine two specializes to job type B. For the 75/25 job mix, we find that both machines appear to have equal interest in job type A while machine two has a stronger preference for job type B. Although we did expect both machines to take on jobs of type A, we were not expecting both to have equally strong preferences.

The final row of plots for R-Wasps summarizes results for the dynamically changing mix case. In these experiments, the job mix is 75/25 for the first 2000 time units and then changed to 25/75 for the next 2000 time units. No new jobs arrive during the final 1000 time units. We can see that over the first half of the plot both machines have strong preferences for job type A and only one machine is specialized in job type B. In the second half of the plots, the configuration begins adapting to the new job mix.

Overall, the results obtained through R-Wasps coordination seem promising, but the basic task allocation model does not appear sufficient to fully achieve optimal factory configurations.

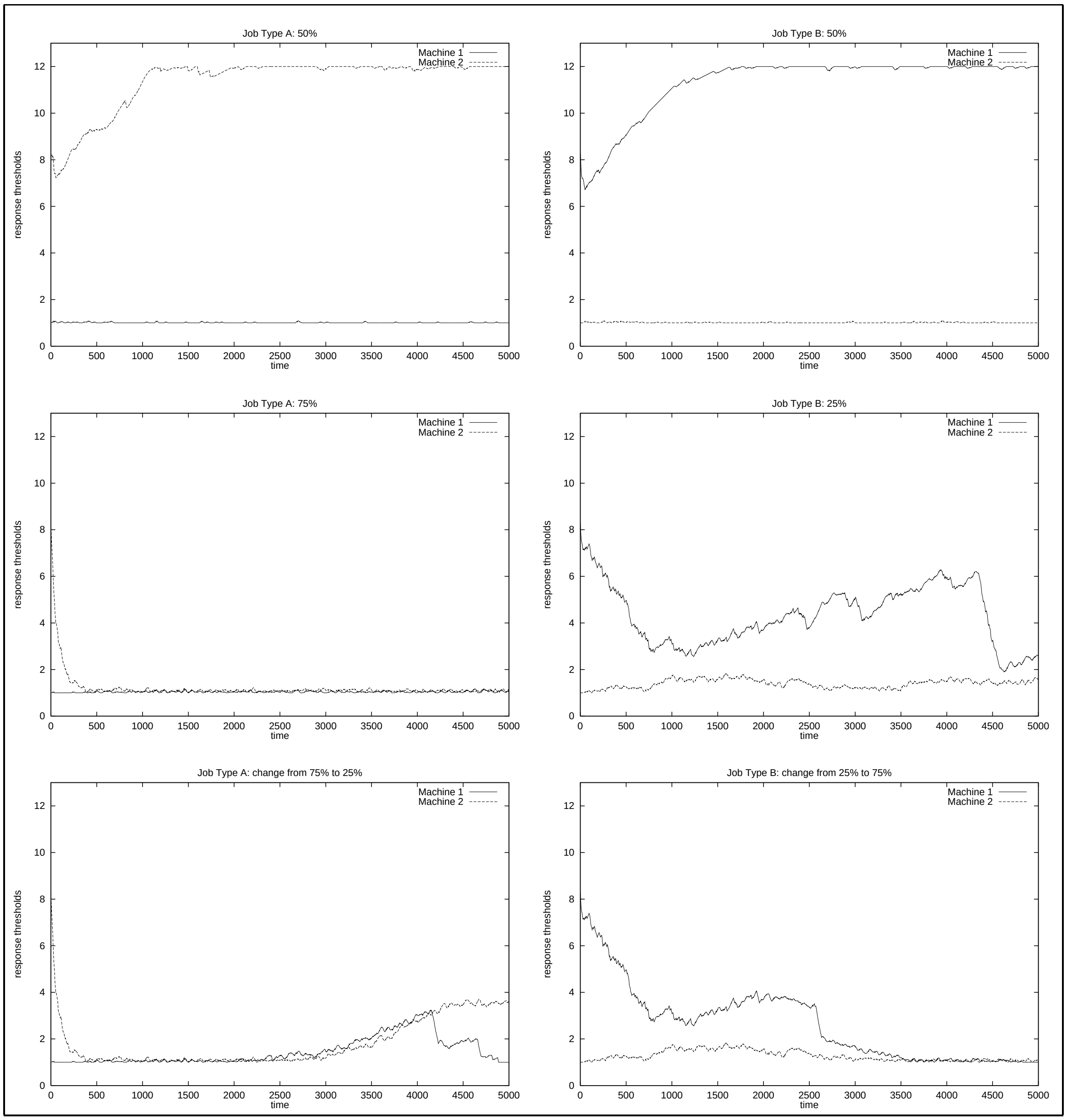

Figure 3: Routing Wasps Only: plots of the average response thresholds over time of the routing wasps associated with both machines for different job mixes.

Table 1: Average throughput for different job mixes. Two variations of the system are compared: 1) routing wasps only (R-Wasps) and 2) routing and scheduling wasps (R&S-Wasps). Total jobs is the average number of jobs released per simulation.

		R-Wasps	R&S-Wasps	Total Jobs
50/50 mix	A	465.0	487.6	509.0
	B	453.6	478.1	502.6
	All	918.6	965.7	1011.6
75/25 mix	A	453.2	567.6	573.6
	B	133.0	193.8	195.0
	All	586.2	761.4	768.6
Changing mix	A	300.2	322.8	322.8
	B	253.4	311.8	311.8
	All	553.6	634.6	634.6

In Figure 4 we display similar plots of the response thresholds for the routing wasps of the augmented R&S-Wasps scheme. These plots show the amplifying effect of scheduling wasps on the behavior of the routing wasps. With a more intelligent sequencing scheme, which results from the interactions of the scheduling wasps, the system as a whole is able to better adapt to job mixes, including job mixes that change over time. For the 50/50 job mix, we do not see any difference worth noting as both models fully converge to the right configuration. However, for the 75/25 job mix we do see some improvement in comparison to the R-Wasps results. In particular, machine one is slightly more specialized to job type A than is machine two and hence is likely to take on more jobs of this type. Machine two still has the stronger interest in job type B.

The most dramatic improvement is seen in the dynamically changing job mix experiments. For the first 2000 time units, we see the exact behavior of the 75/25 job mix simulations as expected. Then, almost immediately after time unit 2000 when the job mix changes, we see a complete reversal of roles. Jobs of type A get routed primarily to machine one and both machines process jobs of type B in the same manner that jobs of type A where handled in the 75/25 experiment. Additionally, we see complete drop-offs of all response thresholds shortly after the 4000 time unit, signifying that the system has processed all jobs and the machines are sitting idle. On average, all jobs are completed by time unit 4090. In the R-Wasps runs, in contrast, the factory was still processing jobs when the simulation came to an end at time unit 5000. The interactions of the scheduling wasps within the nest clearly have a positive impact on the behavior of the routing wasps external to the nest.

The improvement in throughput performance is quantified in Table 1, which shows average throughput (i.e., number of jobs completed) results for all 5000 time unit simulations. All values are averages across the same five simulations for which response thresholds have been previously plotted. For 50/50 job mix simulations R&S-Wasps perform roughly 5.1% better than do R-Wasps. The performance differential increases to 29.9% for the 75/25 job mix simulations. Finally, for the dynamically changing job mix simulations, the R&S-Wasps perform roughly 14.6% better than R-Wasps. However, given that all jobs are completed more than 1000 time units sooner under R&S-Wasps control, the increase in throughput is actually much greater.

Table 2: Average setup time per job broken down by machine for the time intervals 1-2000 and 2001-5000.

		R-Wasps		R&S-Wasps	
		M1	M2	M1	M2
50/50 mix	1-2000	1.2	1.2	0.3	0.6
	2001-5000	0.4	0.7	0.1	0.1
75/25 mix	1-2000	2.9	7.9	2.1	2.5
	2001-5000	4.0	10.3	2.0	2.8
Changing mix	1-2000	2.9	7.9	2.1	2.5
	2001-5000	5.9	7.9	2.4	1.3

In Table 2, we see average setup time per job over the first 2000 time units, and over the remaining portion of the simulation horizon. In the case of the 50/50 and 75/25 job mix experiments, these intervals correspond roughly to "transition" and 'steady state" production phases. In the changing job mix experiments, alternatively, both intervals encompass a "transition" phase. We can first note that in all cases, average setup time is lower for R&S-Wasps as compared to R-Wasps. This confirms that the scheduling wasps are in fact reducing the amount of setup time accrued by the system. In most cases, for R&S-Wasps, there is a decrease in average setup time as the system configures to the job mix. In the 75/25 job mix, however, average setup time on machine two has increased slightly. This is because it has taken some time for this machine to adapt to taking jobs of both types. For the changing job mix, the time required for reconfiguration is also the explanation for the sharp drop in average setup time for machine two and the slight increase for machine one.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a multi-agent approach to factory routing and scheduling, based on various aspects of a model of the adaptive behavior observed in wasp colonies. In our system, wasp-like agents perform routing activities in a manner analogous to task allocation among real wasps. Furthermore, within each machine queue (i.e., nest), other wasp-like agents coordinate and prioritize jobs while self-organizing themselves into a simulated dominance hierarchy. We have demonstrated that our system is robust and is capable of efficiently adapting to dynamically changing and uncertain factory environments.

We are currently extending this work to more complex types of factory coordination problems and environments. Although not discussed in this paper, our system produces comparable results for scalable versions of the problems considered here (e.g., 3 machines, 67/33 mix), and we are likewise investigating applicability to multi-stage production environments. Along a different dimension, we also have interest in adapting our model to operate with respect to other common manufacturing constraints such as due dates and priorities, and other objective criteria (e.g., minimizing weighted tardiness). To accomplish these goals, we are investigating the formulation of alternative force variables for the wasps internal to the nest (e.g., through encoding of appropriate dispatch heuristics).

More generally, we believe that our multi-agent model offers a flexible, decomposable approach to coordinating material flows to meet changing demands, and as such, it should

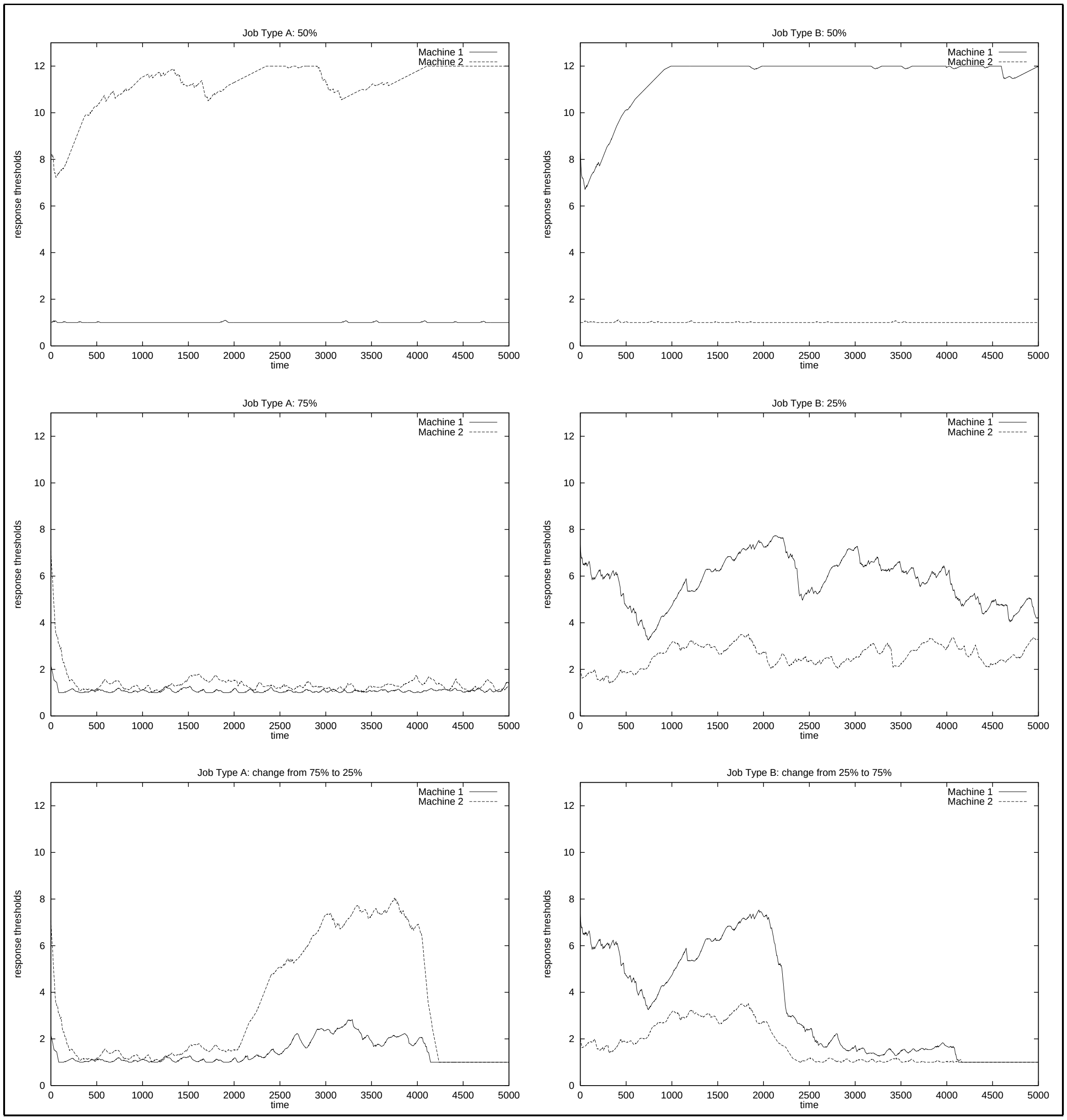

Figure 4: Routing and Scheduling Wasps: plots of the average response thresholds over time of the routing wasps associated with both machines for different job mixes.

also be naturally applicable to more global supply-chain coordination problems. With continuing trends toward specialization on core competencies, manufacturing organizations must rely increasingly on coupling their respective capabilities and partnering to capitalize on new market opportunities, and the ability to rapidly and dynamically reconfigure supply chains becomes increasingly important.

6. ACKNOWLEDGMENTS

Thanks to Debbie Cicirello for the wasp drawing that appeared in Figure 1.

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 F30602-97-2-0227 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.

7. REFERENCES

[1] T. Beaumariage and K. Kempf. Attractors in manufacturing systems with chaotic tendencies, 1995. Presentation at INFORMS-95, New Orleans, http://www.informs.org/Conf/NewOrleans95/TALKS/TB07.3.html.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

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

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

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

[10] G. Y. J. Lin and J. J. Solberg. Integrated shop floor control using autonomous agents. *IIE Transactions*, 24(3):57–71, 1992.

[11] J. S. Liu. *Coordination of Multiple Agents in Distributed Manufacturing Scheduling*. PhD thesis, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 1996.

[12] D. Morley and C. Schelberg. An analysis of a plant-specific dynamic scheduler. In *Final Report, Intelligent Dynamic Scheduling for Manufacturing Systems*, pages 115–122, June 1993.

[13] T. E. Morton and D. W. Pentico. *Heuristic Scheduling Systems: With Applications to Production Systems and Project Management*. John Wiley and Sons, 1993.

[14] P. S. Ow, S. F. Smith, and R. Howie. A cooperative scheduling system. In M. D. Oliff, editor, *Expert Systems and Intelligent Manufacturing*, pages 43–56. Elsevier Science Publishing Co., Inc., 1988.

[15] V. Parunak, A. Baker, and S. Clark. The AARIA agent architecture: From manufacturing requirements to agent-based system design. In *Proceedings of the ICAA'98 Workshop on Agent-Based Manufacturing*, Minneapolis, MN, May 1998.

[16] 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.

[17] K. P. Sycara, S. F. Roth, N. Sadeh, and M. S. Fox. Resource allocation in distributed factory scheduling. *IEEE Expert*, 6(1):29–40, February 1991.

[18] 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.

[19] 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.

[20] 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.