A SCALABLE, DISTRIBUTED ALGORITHM FOR ALLOCATING WORKERS IN EMBEDDED SYSTEMS

WILLIAM AGASSOUNON, ALCHERIO MARTINOLI, RODNEY GOODMAN

Microsystems Research Laboratory, Collective Robotics Group
California Institute of Technology, Pasadena, CA 91125 USA

Abstract

This paper presents a scalable threshold-based algorithm for allocating workers to a given task whose demand evolves dynamically over time. The algorithm is fully distributed and solely based on the local perceptions of the individuals. Each agent decides autonomously and deterministically to work only when it "feels" that some work needs to be done based on its sensory inputs. In this paper, we applied the worker allocation algorithm to a collective manipulation case study concerned with the gathering and clustering of initially scattered small objects. The aggregation experiment has been studied at three different experimental levels by using macroscopic and microscopic probabilistic models, and embodied simulations. Results show that teams using a number of active workers dynamically controlled by the allocation algorithm achieve similar or better performances in aggregation than those characterized by a constant team size, while using a considerably reduced number of agents over the whole aggregation process. Since this algorithm does not imply any form of explicit communication among agents, it represents a cost-effective solution for controlling the number of active workers in embedded systems consisting of a few to thousands of units.

Keywords: swarm intelligence, labor division, dynamic worker allocation, distributed algorithms, probabilistic modeling, embedded systems.

1 Introduction

Swarm Intelligence[1] (SI) is a new computational and behavioral metaphor for solving distributed problems; it is based on the principles underlying the behavior of natural systems consisting of many agents, such as ant colonies and bird flocks. The abilities of such systems appear to transcend the abilities of the constituent individual agents; in all the biological cases studied so far, the emergence of high level control has been found to be mediated by nothing more than a small set of simple low level interactions among individuals, and between individuals and the environment[3].

The SI approach can be applied to control embedded systems that consist of many autonomous decision making entities endowed with minimal communication and local perception capabilities. In particular, we are interested in understanding task allocation and labor division mechanisms exploited in social insect societies that can be transported and adapted to artificial embedded systems such as mobile robotic platforms.

Recently, several macroscopic models, some of them based on threshold-based response[2], others focusing only on task-switching probabilities[10], have been proposed to explain these mechanisms in natural colonies. However, none of these theoretical approaches has focused on how workers gather the information necessary to decide whether or not to switch task or to engage in a task performance. More specifically, none of them has taken into consideration the partial perception in time and space of the demand and the embodiment of the agents. For instance, partial perceptions of the demand combined with real world uncertainties could strongly influence the optimal distribution of thresholds among teammates.

In the collective robotic literature, we find instead threshold-based approaches that take into account the embodiment of the agents but are not scalable because of communication requirements on a finite bandwidth or the necessity of an external supervisor. For instance, in the pioneering approach proposed by Parker[11] each robot at every instant of time and in every position is aware of the progress in task accomplishment of its teammates based on a global radio networking and an absolute positioning system. In Krieger and Billeter's experiment[5], the demand related to the nest energy is assessed by an external supervisor and globally transmitted to all the robots. Using this method, the team of robots has to be heterogeneous and each agent has to be characterized by a different threshold in order to regulate the activity of the team. This in turn results in a different exploitation of the teammates, the one endowed with the lowest threshold being more active than that with the highest one. Finally, none of these experimental works provides a theoretical framework that allows for comparisons among different robotic platforms, environments, and tasks or the performance prediction of teams consisting of hundreds or thousands of units.

In this paper we present a distributed, scalable algorithm that allows a homogeneous team of autonomous, embodied agents to dynamically allocate an appropriate number of workers to a given task as a function of their individual estimation of the progress in the task accomplishment. Since the agents do not perceive the demand globally but rather estimate it locally, they do not work or rest all at the same time, a behavior that would arise if we broadcasted the demand from an external supervisor. Finally, one of the main strengths of this work is the attempt to create a theoretical framework for real embedded systems provided with threshold allocation mechanisms. These systems are therefore analyzed at several implementation levels, from macroscopic models to embedded experiments (e.g. real robots) through numerical microscopic models and embodied simulations. Models allow for a better understanding of the allocation dynamics and for a generalization to other tasks, environmental constraints, and embedded platforms. Optimal parameters of the allocation algorithms can be investigated much more quickly at more abstract levels and the effectiveness of the devised solution can then be verified using embodied simulations and/or real embedded systems. In this paper we present only a validation of the model predictions using embodied simulations. Real robots experiments will be conducted in the near future.

1.1 The aggregation experiment

The first case study we tackled for assessing the efficiency of the worker allocation algorithm is concerned with the gathering and clustering of small objects scattered in an enclosed arena. In the previous research[6,7], the size of the working team was kept constant during the whole aggregation process. These experiments define our baseline for an efficiency comparison with and without the worker allocation algorithm. In this paper, we are using two team performance measurements previously adopted, both of them based on the aggregation process: the average cluster size and the number of clusters.

1.2 The embodied simulations

We implemented the aggregation experiment in Webots 2.01, a 3D sensor-based, kinematics simulator[8] of Khepera robots[9]. The simulator computes trajectories and sensory inputs of the embodied agents in an arena corresponding to the physical set-up (see Fig. 1a).

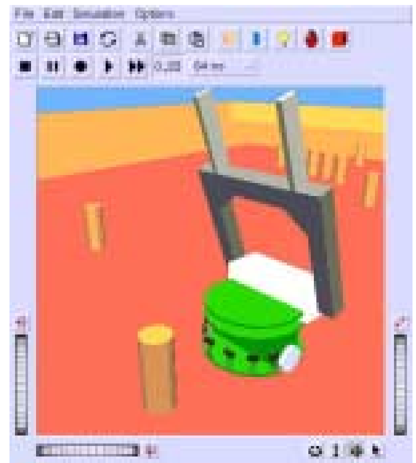

Fig.1a: Close up of a simulated robot (5.5 cm in diameter) in Webots equipped with a gripper turret in front of a seed.

The mean acceleration ratio for this experiment with 10 robots between Webots and real time is about 7 on a PC Pentium III 800 Mhz workstation. The environment is characterized by an 80x80 cm arena (or working zone) where twenty small seeds are randomly scattered at the beginning of the experiment (see Fig. 1b and c). One to ten agents work together in the shared environment. A parking space (resting zone) is appended to the working field. That is the place where non-active agents go to rest (or stay in an idle state to save energy). Agents are endowed with sensor capabilities for distinguishing the border between resting and working zone (e.g. physically speaking, IR sensors beneath the robots' bodies).

Without considering the mode-switching behavior (explained in subsection 2.1), we can summarize each agent's behavior with the following simple rules. In its default behavior, the agent moves straight forwards within the working zone looking for seeds. When at least one of its six frontal proximity sensors is activated, the agent starts a discriminating procedure. Two cases can occur: if the agent is in front of a large object (a wall, another agent, or the body side of a cluster of seeds), the object is considered as an obstacle and the agent avoids it. In the second case, the small object is considered as a seed. If the agent is not carrying a seed, it grasps the seed with the gripper, otherwise, it drops the seed it is carrying close to that it has found; then, in both cases,

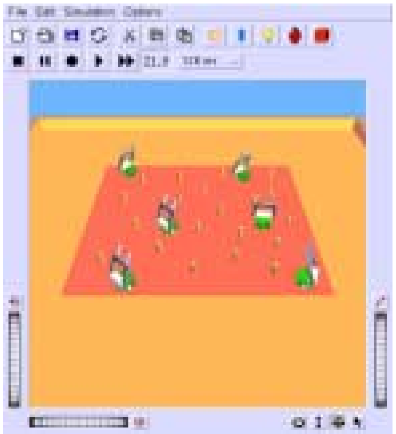

Fig.1b: Experimental setup: the darker area corresponds to the working zone, the lighter area to the resting zone. Typical situation at the beginning of the aggregation with 6 agents.

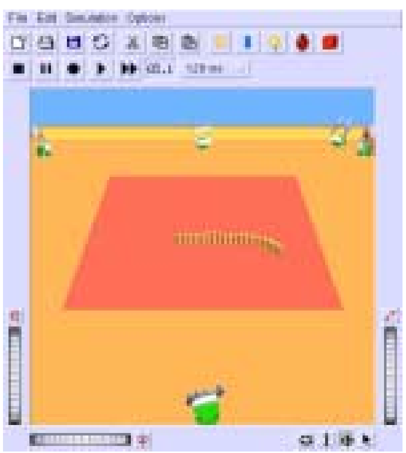

Fig.1c: Typical situation at the end of aggregation experiment (e.g. 4-hour simulated time).

the agent resumes looking for seeds. With this simple individual behavior, the team is able to gather objects in clusters of increasing size. A cluster is defined as a group of seeds whose neighboring elements are separated by at most one seed diameter. Note that, because only the two extreme seeds of a cluster can be identified as seeds (as opposed to obstacles) by the agents, clusters are built in lines. As shown by Martinoli et al.[7], if the probability of creating new clusters by dropping a seed in the middle of the arena without having detected another seed or of splitting clusters in two by picking up an internal seed is zero, the number of clusters is monotonically decreasing and eventually a single cluster will always arise.

1.3 The microscopic model

The central idea of the microscopic probabilistic model is to describe the experiment as a series of stochastic events with probabilities based on simple geometrical considerations. The probability for any agent to encounter any other object present in the arena (e.g. seeds, teammates, the border between the working field and the resting zone, etc.) is given by the ratio of the extended area occupied by that object to the total arena area in which the agent is moving. The extended area occupied by each object is computed by considering the detection range of that object by an active agent (robot) taken from the center of that agent. In this specific collective manipulation case study, seed picking up and dropping probabilities have also to be taken into account once a cluster is found and they depend on the angle of approach of the agent to the cluster (clusters can be modified only at their tips). The states of the agents in the numerical probabilistic model are defined by a finite state machine, but, instead of computing the detailed sensor information and trajectories of the agents, the change of states is determined by the throwing of dice. The overall behavior is then computed by averaging the results of several runs of the same experiment. Fig. 2 illustrates the transformation of a 2D arena space to a 1D probability space used in this model. A more detailed description of this microscopic modeling methodology can be found in previous work[6,7].

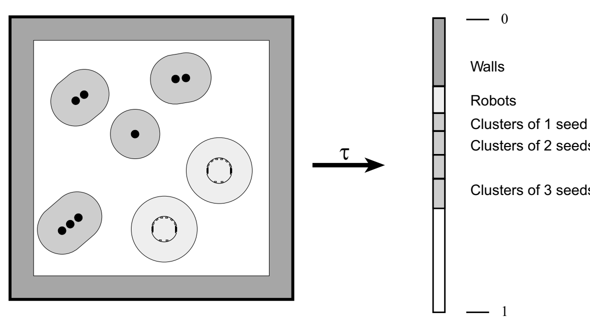

Fig. 2: Example of transformation of the 2D arena space to a 1D probability space.

Working with models also brings an additional time saving in comparison to embodied simulations. The mean acceleration ratio for this experiment with 10 agents between the microscopic probabilistic model and Webots is about 700 on a PC Pentium III 800 MHz workstation.

1.4 The macroscopic model

For this particular experiment, we have extended the previously existing microscopic model to a more compact and abstract macroscopic model. At this modeling level, we capture with a set of difference equations the state changes of the environment (aggregation evolution) as well as of the agents' load (carrying or not carrying a seed).

The advantages of exploring two different modeling levels are three-fold. First, going through the intermediate level of abstraction defined by the microscopic model, the parameters used in the macroscopic model are still so realistic that the resulting predictions are not only qualitatively but also quantitatively accurate. Second, the microscopic model can also be used not only for predicting average team performances but also their variation (standard error, standard deviation). Third, at the macroscopic level, optimization of system parameters is easier and faster.

The macroscopic model is based on the following probabilities (the same used in the microscopic model, all are computed from simple geometrical considerations):

- The time-dependent probability of any agent to encounter a wall: $P_w(t)$. Without worker allocation, this probability is constant throughout the experiment. With the worker allocation, for working or resting agents, the enclosure is located around the working or the resting zones respectively.
- The time-dependent probability of any agent to encounter a teammate: $P_{rb}(t)$. With worker allocation, this probability is not constant since the number of agents in both zones is not constant but defined by the worker allocation algorithm.
- The time-dependent probability of any agent to encounter a cluster of a size n from the favorable seed-picking angle: $P^n_{dec}(t)$.
- The time-dependent probability of any agent to encounter a cluster of a size n from the favorable seed-dropping angle: $P^n_{inc}(t)$.

The last two probabilities change as a function of the aggregation process in both cases, with and without worker allocation.

In general, the number of clusters of size k is recursively given by Eq.1, where: $N_f(t)$ is the number of robots not carrying a seed at time t, $N_c(t)$ is the number of robots carrying a seed at time t, and T_p, the average amount of time a robot needs to pick up or drop a seed.

$$\begin{aligned} N_k(t+1) &= N_k(t) - N_f(t-T_p)P^k_{dec}(t-T_p) \\ &+ N_c(t-T_p)P^{k-1}_{inc}(t-T_p) + N_f(t-T_p)P^{k+1}_{dec}(t-T_p) \\ &- N_c(t-T_p)P^k_{inc}(t-T_p). \qquad \text{Eq.1} \end{aligned}$$

In the right side of Eq. 1, the second and fifth terms represent the average number of clusters of size k having their size incremented or decremented by one seed within T_p time periods, the third and fourth terms represent the average number of clusters of size *k-1* and *k+1* respectively, transformed into clusters of size k within T_p time periods. Similarly, the average number of 'free' agents (i.e. agents not carrying a seed but looking for seeds to pick up) at time $t+1$ is given by Eq. 2, where: T_a is the average amount of time an agent needs to avoid an obstacle, $P_{obs}(t)$ is the probability that any free agent gets involved in an obstacle avoidance process at time t, and M, the total number of seeds.

$$\begin{aligned} N_f(t+1) &= N_f(t) - N_f(t-T_p)P_{obs}(t) \\ &- N_f(t)\sum_{k=1}^{M} P^k_{dec}(t) + N_f(t-T_a)P_{obs}(t-T_a) \\ &+ N_c(t-T_p)\sum_{k=1}^{M} P^k_{inc}(t-T_p) \qquad \text{Eq.2} \end{aligned}$$

In the right side of Eq. 2, the second, third, and fourth terms represent the average number of free agents involved in obstacle avoidance and cluster construction at time t or finishing an obstacle avoidance started at time period $t - T_a$ respectively. The last term represents the average number of agents that dropped a seed within time period $t - T_a$. A similar difference equation gives the dynamic expression of $N_c(t+1)$.

2 The distributed worker allocation

The main objective of this case study is to show that the introduction of worker allocation mechanisms allows the team of agents to increase its efficiency as a whole by allocating the right number of workers as a function of the decreasing demand intrinsically defined by the aggregation process. Intuitively, we can imagine that at the beginning of the aggregation there are several possible manipulation sites (i.e. several scattered seeds) that allow for a parallel work of several agents. As the aggregation process goes on, these sites are more and more reduced and having more agents competing for the same manipulation site decreases their efficiency in aggregation.

In threshold-based systems, the 'propensity' for any given agent to act is given by a *response threshold*. If the demand is above an agent's threshold then that agent continues to perform the task, conversely if the demand is below its threshold then the agent stops performing that particular task.

2.1 The worker allocation algorithm

Our current worker allocation algorithm is as follows. When an agent has not been able to work (i.e. to pick up and drop a seed) for a reasonable amount of time, its propensity to accomplish that particular task is decreased. If the stimulus goes below a fixed threshold (i.e. if the amount of time spent in the search for work to accomplish is above a given T_{search} time-out), a first deterministic switching mechanism prompts the agent to leave the working zone for resting in the adjacent parking space. An agent carrying a seed that decides to become inactive cannot do so until it finds an appropriate spot (i.e. one tip of a cluster) to drop the seed. A second deterministic switching mechanism allows the agent to resume the working activity as soon as the resting time has exceeded a T_{rest} time-out. Thus, with this simple algorithm characterized by two thresholds common to all the teammates, the agents are able to locally evaluate the aggregation demand and to decide whether to work or rest. Since the agents do not perceive the demand globally but rather estimate it locally, they do not stop working all at the same time. Thus, by exploiting the intrinsic noise of the system as well as local perceptions and interactions, we can obtain a self-organized worker allocation based on the local assessment of the current status of the shared resource, i.e. the environment.

2.2 The modified macroscopic model

We modified Eq. 2 to take into account the influence of worker allocation on the different parameters of the system. The key difference in the modeling of the number of free (respectively loaded) agents is that their actions now depend not only on the current environmental state but also on their recent experience. In particular for any given agent, the decision to leave or stay in the working zone at time t is dependent on whether that agent has been able to work (i.e. exactly pick up and drop at least one seed) over the last T_{search} time periods. A posteriori probabilities related to the states of the environment during that past are used to compute the probability for any given agent to stop working at time t. That probability is exactly equal to the probability that the same agent has not been able to find some work to accomplish during that period.

3 Results and discussion

In this section we present and compare results collected at the three different experimental levels, macroscopic modeling, microscopic modeling, and embodied simulations. Each aggregation run lasted 10 h. For each team size, 100 and 30 runs were carried out using the microscopic model and the embodied simulator respectively. All error bars represent the standard deviations among runs. All results reported below were obtained without using any free parameters. All the parameters introduced in the models (e.g. mean obstacle avoidance duration, mean time to pick up/drop a seed, mean time to leave the working zone) were measured from a single embodied agent.

In the following experiments, we arbitrarily hand-coded two values for the two allocation parameters: T_{search} = 25 min, T_{rest} > 10 h. For instance, T_{rest} was chosen based on the following considerations: since in this particular aggregation experiment the demand is monotonically decreasing, there is no need to have inactive agents resuming the working activity. In the case of a demand changing differently, for instance if more seeds were suddenly introduced during the aggregation process, having a T_{rest} shorter than the 10 h observation time would be an interesting solution to investigate. A systematic search for optimal parameters is an interesting topic for future research.

3.1 Aggregation without worker allocation

Fig. 3 presents the model predictions and the embodied simulation results of the aggregation experiment without the use of any task allocation algorithm for groups consisting of 10 agents. In Fig.3, the upper set of (three) curves represents the (increasing) average size of the clusters over time while the other set shows the (decreasing) average number of clusters over time.

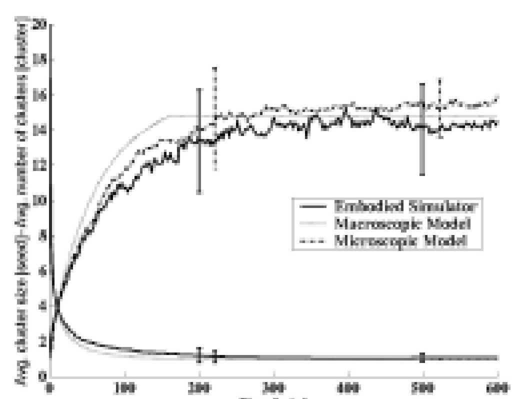

Fig. 3: Aggregation w/o worker allocation with 10 active workers

The good agreement between the results from the three different levels shows how reliable both modeling methods are for an accurate forecast of the evolution of the aggregation process.

3.2 Aggregation with worker allocation

Fig. 4 presents the results of an aggregation experiment using worker allocation and 10 agents at the three different experimental levels.

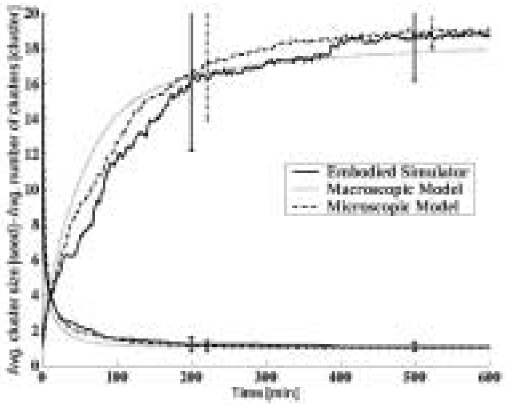

Fig.4a: Aggregation evolution with worker allocation (set of 10 agents).

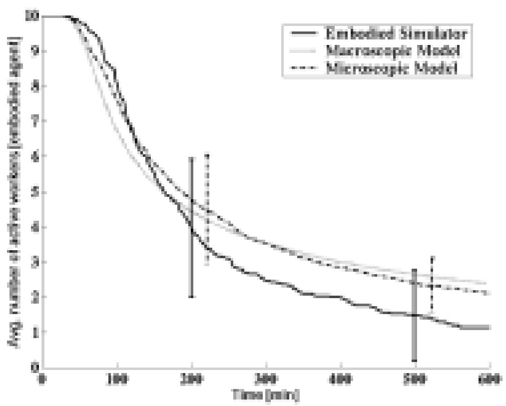

Fig.4b: Average number of active workers over time with the allocation algorithm.

In order to assess quantitatively the advantages of the worker allocation algorithm, we introduced a new metric, the *allocation efficiency*. In this aggregation case study, we defined the allocation efficiency as the ratio of the average cluster size at any given time to one plus the average number of active workers from the beginning till the same time, i.e. *cluster size/(1+number of workers).* Note that the offset at the denominator guaranties an upper bound on the possible values of the metric. In other words, the allocation efficiency corresponds to the amount of work done per number of workers allocated to the task.

Fig. 5 shows the allocation efficiency for this aggregation experiment with a group of 10 agents. The upper two curves represent the allocation efficiency (microscopic model and embodied simulations) of the team with the worker allocation algorithm while the lower two (overlapping) correspond to the allocation efficiency without any worker allocation algorithm. Note that this metric is a hyperbolic function of the number of active workers; therefore, it amplifies the differences illustrated in Fig.4b for small numbers of agents. This explains the net increase in mean values and standard deviations of the allocation efficiency of the embodied simulator related to those of the microscopic model.

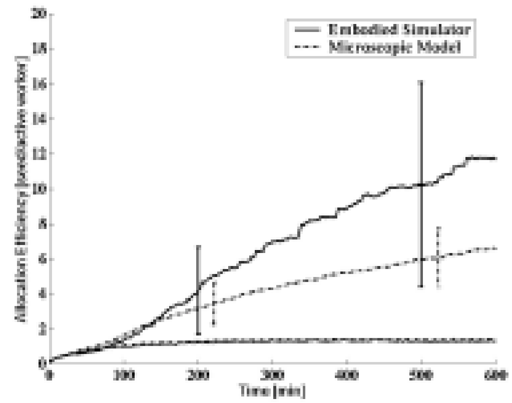

Fig.5: Allocation efficiencies for the aggregation experiment with 10 agents, w/ and w/o worker allocation

Fig. 5 illustrates that the effect of the worker allocation algorithm becomes more significant after about 100 minutes when the first agents start to leave the working zone. It is worth noting that at this point, on average, a single cluster has not necessarily arisen yet. Therefore, by using the worker allocation in the last phase of the aggregation we achieve a similar mean clustering performance as with a full team while using fewer agents. However, the main advantage of worker allocation can be seen after 200 minutes, when a single cluster has already arisen during most of the runs. The size of this single cluster continuously grows in the experiment with worker allocation while it converges to and remains stable around 15 seeds (see Fig. 3) in the experiment without worker allocation. Intuitively, this can be explained by the fact that with only two manipulation sites left in the arena, on average one half of the active agents are always carrying a seed and the other half are not. If the number of active agents is reduced, the size of the single cluster is consequently increased and this explains the high allocation efficiency using the worker allocation algorithm shown in Fig. 5 after 200 minutes.

4 Conclusion

In this paper, we have presented a scalable threshold-based algorithm that allows a homogeneous team of autonomous, embodied agents to dynamically allocate an appropriate number of workers to a given task as a function of their individual estimation of the progress in the task accomplishment. The algorithm is fully distributed and, since it is solely based on the local perceptions of the individuals and does not imply any form of explicit communication among agents, it represents a cost-effective solution for controlling the number of active workers in embedded systems consisting of a few to thousands of units. Results show that teams using a number of active workers dynamically controlled by the allocation algorithm achieve similar or better performances in aggregation than those characterized by a constant team size, while using a considerably reduced number of agents per time unit over the whole aggregation, thus, optimizing both the team performance and the use of the resources of the system. Our results reveal that intrinsic labor division mechanisms can achieve interesting performances without any use of external supervisors or global wireless networking. However, much work remains to be done in optimizing the efficiency of such distributed worker allocation algorithms while maintaining their scalability. Local peer-to-peer explicit communication and specialization through adaptive thresholds[12] are just two possible paths to explore. Furthermore, the good agreement between models and embodied simulations confirms the usefulness of probabilistic modeling tools for achieving fast predictions and understanding aggregation and worker allocation dynamics. Finally, macroscopic models expressed in the form of difference equations seem to be a promising approach to other case studies and to further generalization of the allocation algorithm presented in this paper.

Acknowledgements

We would like to thank Amit Kenjale, Douglas Lanman, and Oran Switzer for the long hours spent writing the codes for the Webots simulations and Adam Hayes for comments on this paper. This work is supported in part by the TRW Foundation and the Caltech Center for Neuromorphic Systems Engineering as part of the NSF Engineering Research Center Program under grant EEC-9402726. Further support was received from the DARPA grant DAAK60-97-K-9503, ONR grant N00014-98-1-0821, and ARO-MURI grant DAAG55-98-1-0266.

References

[1] Beni G. and Wang J., Swarm Intelligence. *Proc. of the Seventh Annual Meeting of the Robotics Society of Japan*, Tokyo, Japan, pp.425-428, 1989.

[2] Bonabeau E., Theraulaz G., and Deneubourg J.-L., Fixed Response Thresholds and the Regulation of Division of Labour in Insect Societies. *Bulletin of Mathematical Biology*, Vol.60, pp.753-807, 1998.

[3] Bonabeau E., Dorigo M., and Theraulaz G., Swarm Intelligence: From Natural to Artificial Systems. *SFI Studies in the Science of Complexity*, Oxford University Press, New York, NY, 1999.

[4] Hayes A. T., Martinoli A., and Goodman R. M. Comparing Distributed Exploration Strategies with Simulated and Real Autonomous Robots. *Proc. of the Fifth Int. Symp. on Distributed Autonomous Robotic Systems DARS00*, Knoxville, TN, pp.261-270, 2000.

[5] Krieger M. B. and Billeter J.-B. The Call of Duty: Self-Organized Task Allocation in a Population of Up to Twelve Mobile Robots. *Robotics and Autonomous Systems*, Vol.30, No.1-2, pp.65-84, 2000

[6] Martinoli A., Ijspeert A. J., and Mondada F., Understanding Collective Aggregation Mechanisms: From Probabilistic Modeling to Experiments with Real Robots. *Robotics and Autonomous Systems*, Vol.29, pp.51-63, 1999.

[7] Martinoli A., Ijspeert A. J., and Gambardella L. G., A Probabilistic Model for Understanding and Comparing Collective Aggregation Mechanisms. *Proc. of the Fifth Int. European Conf. on Artificial Life ECAL-99*, Lausanne, Switzerland. Lectures Notes in Computer Science, pp.575-584, 1999.

[8] Michel O., Webots: Symbiosis Between Virtual and Real Mobile Robots. *Proc. of the First Int. Conf. on Virtual Worlds*, Paris, France, Springer Verlag, pp.254-263, 1998.

[9] Mondada F., Franzi E., and Ienne P. Mobile Robot Miniaturization: A Tool for Investigation in Control Algorithms. *Proc. of the Third Int. Symp. on Experimental Robotics*, Kyoto, Japan, 1993.

[10] Pacala S. W., Gordon D. M., and Godfray H. C. J., Effects of Social Group Size on Information Transfer and Task Allocation. *Evolutionary Ecology*, Vol.10, pp.127-165, 1996.

[11] Parker E. L., ALLIANCE: An Architecture for Fault Tolerant Multirobot Cooperation. *IEEE Transactions on Robotics and Automation*, Vol.14, No.2, pp. 220-240, 1998.

[12] Theraulaz G., Bonabeau E., and Deneubourg J.-L., Response Threshold Reinforcement and Division of Labour in Insect Societies. *Proc. of the Royal Society of London Series B*, Vol.265, pp.327-332, 1998.