Adaptive Division of Labor in Large-Scale Minimalist Multi-Robot Systems

Chris Jones and Maja J Matarić
Computer Science Department
University of Southern California
941 West 37th Place, Mailcode 0781
Los Angeles, CA 90089-0781 USA
{cvjones|maja}@robotics.usc.edu

***Abstract*— A Large-Scale Minimalist Multi-Robot System (LMMS) is one composed of a group of robots each with limited capabilities in terms of sensing, computation, and communication. Such systems have received increased attention due to their empirically demonstrated performance and beneficial characteristics, such as their robustness to environmental perturbations and individual robot failure and their scalability to large numbers of robots. However, little work has been done in investigating ways to endow such a LMMS with the capability to achieve a desired division of labor over a set of dynamically evolving concurrent tasks, important in many task-achieving LMMS. Such a capability can help to increase the efficiency and robustness of overall task performance as well as open new domains in which LMMS can be seen as a viable alternative to more complex control solutions. In this paper we present a method for achieving a desired division of labor in a LMMS, experimentally validate it in a realistic simulation, and demonstrate its potential to scale to large numbers of robots and its ability to adapt to environmental perturbations.**

I. Introduction and Motivation

A Large-Scale Minimalist Multi-Robot System (LMMS) is a multi-robot system composed of a large number of robots, each having limited capabilities in terms of sensing, computational power, and communication range and bandwidth. We define a *minimalist robot* as one which maintains little or no state information, extracts limited, local, and noisy information from its available sensors, and lacks the capability for active communication with other robots. Due to these limited capabilities, the world in which a minimalist robot is situated is formally partially-observable and highly non-stationary, and it is therefore not practical to assume that such a robot is capable of reliably knowing a significant portion of the current global state of the environment or of overall task progress.

These limitations in sensing, communication, and computation preclude a minimalist robot from performing tasks requiring significant computation or communication capabilities. Nonetheless, minimalist robots have been shown to be highly effective at a number of collective tasks, such as multi-robot formation control [4], collection [7], and robot soccer [18]. A system composed of a large number of such minimalist robots has the potential of conferring advantages including increased robustness to individual robot failure as no single robot is critical to task performance, the prospect of scaling to increasingly larger numbers of robots as there are few bottlenecks in terms of complex communication, planning, or coordination requirements, and increased adaptability to changes in the environment since individuals act based on local information and are not tied to globally coordinated plans.

The aim of this work is to investigate a method by which to endow a LMMS with the capability to achieve a desired division of labor over a set of dynamically evolving concurrent tasks, a critical requirement of any task achieving large-scale multi-robot system. We define *division of labor* as the phenomenon in which individuals in a multi-robot system concurrently execute a set of tasks. Such division of labor may need to be continuously adjusted in response to changes in the task environment or group performance. The broader scope of this work is in understanding the ways in which to achieve robust, scalable, and efficient coordination in a LMMS.

This paper is organized as follows. In Section II we provide the relevant related work. In Section III we give a detailed description of the concurrent foraging task domain we use as validation of our division of labor mechanism. In Section IV we present the robot controller we use to produce a division of labor in a LMMS. In Section V we describe and analyze experimental results, and in Section VI we draw conclusions from this work.

II. Related Work

Here we summarize briefly the related work in physical LMMS using robots with similar capabilities to those on which our system is based. Matarić [13] provides early work on group coordination in LMMS using a collection of simple basis behaviors. Agassounon and Martinoli [1] present minimalist methodologies for coordination in robot groups. Beckers et al. [2] demonstrate the capabilities of minimalist multi-robot systems in object clustering and sorting. Kube and Zhang [10] present an approach to box-pushing using a group of robots with simple sensors and reactive control. Werger and Matarić [17] present a minimalist solution in the multi-robot foraging domain. Martinoli et al. [12] present work on the probabilistic modeling of robot behavior in the task regulation domain, demonstrating its performance as compared to experiments on physical and simulated robots. Werger [18] presents coordinated behavior in a robot soccer team using a minimalist behavior-based control system. Krieger and Billeter [9] present a decentralized task allocation mechanism for large mobile robot groups based on individual task-associated response thresholds in a collection domain. Holland and Melhuish [8] use probabilistic behavior selection in minimalist robotic clustering and sorting. Goldberg and Matarić [7] precisely define the foraging task for LMMS and provide a collection of general distributed behavior-based algorithms and their empirical evaluation. Fredslund and Matarić [4] present work on the problem of achieving coordinated behavior in the context of formations using a distributed group of physical robots using only local sensing and minimal communication. Lerman and Galstyan [11] describe a method of macroscopic analysis in a multi-robot division of labor domain very similar to the one we experimentally investigate.

In the multi-robot literature, there is work on more communication and computationally complex forms of task regulation in multi-robot systems through the use of publish/subscribe and market-based methods (e.g., [6]) and systems in which significant global state is made known to all robots (e.g., [14]).

Research that studies and simulates insect colonies and their behaviors is also relevant. Théraulaz et al. [15] describe how the adaptability of complex social insect societies is increased by allowing members of the society to dynamically change tasks (behaviors) when necessary. Giving that ability to robots allows a LMMS to operate in domains requiring the simultaneous regulation of many tasks. Bonabeau et al. [3] describe a model of a task regulation mechanism in insect societies through the use of response thresholds for task-related stimuli. Théraulaz et al. [15] extend that model by introducing an adaptive threshold that changes over time based on individual task performance.

The division of labor mechanism we present can be considered an instance of a response threshold model as presented in Bonabeau et al. [3], Krieger and Billeter [9], Théraulaz et al. [15], and Agassounon and Martinoli [1]. However, our task domain and division of labor mechanism differ in that the task-related stimuli are perceived locally by the individual robots and are not altered as a result of task performance. Furthermore, the individual robots are initially homogeneous, as opposed to Krieger and Billeter [9] in which robot are initially assigned different response thresholds, and the robots do not learn or become specialized through adaptive response thresholds as is the case in Théraulaz et al. [15] and Agassounon and Martinoli [1].

III. Concurrent Foraging Task Domain

In order to experimentally validate a mechanism for providing a LMMS with division of labor capabilities, we investigated the division of labor in the concurrent foraging task domain. *Concurrent foraging*, a variation on traditional foraging, consists of an arena populated by multiple types of objects to be collected. Each robot is equally capable of foraging all object types, but can only be allocated to foraging for one type at any given time. Additionally, all robots are engaged in foraging at all times; a robot cannot be idle. A robot may switch the object type according to its control policy, when it determines it is appropriate to do so. It is desirable for a robot to avoid thrashing (i.e., wasting time and energy) by needlessly switching the object type for which it is foraging.

A. Task Description

Our experimental domain of concurrent foraging requires multiple object (puck) types to be foraged from a circular arena. Initially, the arena is randomly populated by two types of pucks: $Puck_{Red}$ and $Puck_{Green}$, which are distinguishable by their color.

In this task, the robots move in an enclosed arena and pick up encountered pucks. When a robot picks up a puck, the puck is consumed (i.e., it is immediately removed from the environment, not transported to another region) and the robot carries on foraging for other pucks. Immediately after a puck is consumed, another puck of the same type is placed in the arena at a random location. This is done so as to maintain a constant puck density in the arena throughout the course of an experiment. In some situations, the density of pucks can have an affect on the division of labor performance. This is an important consideration in mechanisms for division of labor in LMMS for many domains; however, in this work we want to limit the number of experimental variables impacting system performance. Therefore, we reserve the investigation on the impact of varying puck densities on division of labor in LMMS for future work.

The division of labor portion of the task requires the robots to split their numbers by having some forage for $Puck_{Red}$ pucks and others for $Puck_{Green}$ pucks. For the purpose of our experiments, we desire a division of labor such that the proportion of robots foraging for $Puck_{Red}$ pucks is equal to the proportion of $Puck_{Red}$ pucks present in the foraging arena (e.g., if $Puck_{Red}$ pucks make up 30% of the pucks present in the foraging arena, then 30% of the robots should be foraging for $Puck_{Red}$ pucks). In general, the desired division of labor could take other forms. For example, it could be related to the relative reward or cost of foraging each puck type without change to our approach.

As was stated earlier, due to their minimalist capabilities, individual robots do not have direct access to global information such as the size and shape of the foraging arena, the initial or current number of pucks to be foraged (total or by type), or the initial or current number of foraging robots (total or by foraging type). Also, it cannot be assumed that any robot or subset of robots will always be operational or the proportion of pucks will remain constant over time.

IV. The Robots

The robots used in the experimental simulations are realistic models of the ActivMedia Pioneer 2DX mobile robot. Each robot, approximately 30 cm in diameter, is equipped with a differential drive, an odometry system using wheel rotation encoders, 8 evenly spaced sonars covering the front 180 degrees used for obstacle avoidance, and a forward-looking Sony color camera with a 60-degree field-of-view and a color blob detection system (used for puck and robot detection and classification through color). Each robot is also equipped with a 2-DOF gripper on the front, capable of picking up a single 8 cm diameter puck at a time. There is no capability available for explicit, direct communication between robots nor can pucks and other robots be uniquely identified.

A. Behavior-Based Controller

All robots have identical behavior-based controllers consisting of the following mutually exclusive behaviors: Avoiding, Wandering, Visual Servoing, Grasping, and Observing. Descriptions of the behaviors used in the division of labor implementation are given below.

- The **Avoiding** behavior causes the robot to turn to avoid obstacles in its path.
- The **Wandering** behavior causes the robot to move forward and, after a random length of elapsed time, to turn left or right through a random arc for a random period of time.
- The **Visual Servoing** behavior causes the robot to move toward a detected puck of desired type. If the robot's current foraging state is Robot_{Red}, the desired puck type is Puck_{Red}, and if the robots current foraging state is Robot_{Green}, the desired puck type is Puck_{Green}.
- The **Grasping** behavior causes the robot to use its gripper to pick up and consume a puck within the gripper's grasp.
- The **Observing** behavior causes the robot to take an image from its camera and record the detected pucks and robots to their respective histories. The robot then updates its foraging state based on those histories. A description of the histories is given in Section IV-B and a description of the foraging state update procedure is given in Section IV-C.

Each behavior listed above has a set of activation conditions based on relevant sensor inputs and state values. When met, the conditions cause the behavior to be become active. A description of when each activation condition is active is given below. The activation conditions of all behaviors are shown in Table I.

- The **Obstacle Detected** activation condition is true when an obstacle is detected by the sonar within a distance of 1 meter. Pucks are not detectable by the sonar, so are therefore not considered obstacles.
- The **Puck_{Det} Detected** activation condition is true if the robot's current foraging state is Robot_{Det} and a puck of type Puck_{Det} (where Det is Red or Green) is detected by the color camera within a distance of approximately 5 meters and within $\pm$ 30 degrees of the robot's direction of travel.
- The **Gripper Break-Beam On** activation condition is true if the break-beam sensor between the gripper jaws detects an object.
- The **Observation Signal** activation condition is true if the distance traveled by the robot according to odometry since the last time the **Observing** behavior was activated is greater than 2 meters.

B. State Information

All robots maintain three types of state information: foraging state, observed puck history, and observed robot history. The foraging state identifies the type of puck the robot is currently involved in foraging. A robot with a foraging state of Robot_{Red} refers to a robot engaged in foraging Puck_{Red} pucks and a foraging state of Robot_{Green} refers to a robot engaged in foraging Puck_{Green} pucks.

Each robot is outfitted with a colored beacon observable by nearby robots which indicates the robot's current foraging state. The color of the beacon changes to reflect the current state – a red beacon for a foraging state of Robot_{Red} and a green beacon for Robot_{Green}. Thus, the colored beacon acts as a form of local, passive communication conveying the robot's current foraging state. All robots maintain a limited, constant-sized history storing the most recently observed puck types and another constant-sized history storing the foraging state of the most recently observed robots. Neither of these histories contains a unique identity or location of detected pucks or robots, nor does it store a time stamp of when any given observation was made. The history of observed pucks is limited to the last `MAX-PUCK-HISTORY` pucks observed and the history of the foraging state of observed robots is limited to the last `MAX-ROBOT-HISTORY` robots observed.

Obstacle Detected	$Puck_{Det}$ Detected	Gripper Break-Beam On	Observation Signal	Active Behavior
X	X	X	1	Observing
1	X	X	X	Avoiding
0	1	0	0	Visual Servoing
0	X	1	0	Grasping
0	X	X	X	Wandering

TABLE I

BEHAVIOR ACTIVATION CONDITIONS. BEHAVIORS ARE LISTED IN ORDER OF DECREASING RANK. HIGHER RANKING BEHAVIORS PREEMPT LOWER RANKING BEHAVIORS IN THE EVENT MULTIPLE ARE ACTIVE. X DENOTES THE ACTIVATION CONDITION IS IRRELEVANT FOR THE BEHAVIOR.

While moving about the arena, each robot keeps track of the approximate distance it has traveled by using odometry measurements. At every interval of 2 meters traveled, the robot makes an observation. An *observation* consists of the robot taking the current image from its color camera and, using simple color blob detection, classifying all currently visible pucks and robots through their respective colors and adding them to their respective histories. This procedure is nearly instantaneous; therefore, the robot's behavior is not outwardly affected. The area in which pucks and other robots are visible is within 5 meters and $\pm$ 30 degrees in the robot's direction of travel. Observations are only made after traveling 2 meters because updating too frequently leads to over-convergence of the estimated puck and robot type proportions due to repeated observations of the same pucks and/or robots. On average, during our experiments, a robot detected 2 pucks and robots per observation.

C. Foraging State Transition Function

After it makes an observation, the robot re-evaluates its current foraging state given the newly updated puck and robot histories and probabilistically changes foraging state. The probability that a robot with a current foraging state of $Robot_{Green}$ will change its foraging state to $Robot_{Red}$ is given by the probability `P(Green-Red)` shown in Equation 1. Similarly, the probability that a robot with a current foraging state of $Robot_{Red}$ will change its foraging state to $Robot_{Green}$ is given by the probability `P(Red-Green)` shown in Equation 2. In Equations 1 and 2, `RR` is the proportion of $Robot_{Red}$ entries in the Robot History and `RP` is the proportion of $Puck_{Red}$ entries in the Puck History. For an analytical explanation of this and other transition functions see Lerman and Galstyan [11].

$$P(Green-Red) = \begin{cases} (GR-GP)*(1-GP), & \text{if GR}\geq\text{GP}, \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$P(Red-Green) = \begin{cases} (RR-RP)*(1-RP), & \text{if RR}\geq\text{RP}, \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

V. EXPERIMENTAL RESULTS

We experimentally validated our LMMS division of labor mechanism using the probabilistic transition function in the realistic Player/Stage simulation environment. Player [5] is a server that connects robots, sensors, and control programs over the network. Stage [16] simulates a set of Player devices. Together, the two represent a high-fidelity simulation tool for individual robots and robot teams which has been validated on a collection of real-world robot experiments using Player control programs transferred directly to physical Pioneer 2DX mobile robots.

The experimental arena used in all experiments is circular and has an area of approximately 315 square meters. All experiments used 20 robots and 50 pucks and all presented results have been averaged over 30 experimental runs.

To test the adaptability of the division of labor mechanism to external perturbations in puck type proportions, they were dynamically changed at various times during the experimental trials. The experiments began with 30% $Puck_{Red}$ and 70% $Puck_{Green}$ pucks. At time 10000 seconds, the relative proportion of pucks were changed to 80% $Puck_{Red}$ and 20% $Puck_{Green}$ pucks, and at time 20000 the relative proportions were changed to 50% $Puck_{Red}$ and 50% $Puck_{Green}$ pucks. The total number of pucks remained constant throughout the experiment.

The plots in Figure 1 show a comparison between the performance of the probabilistic transition functions using `MAX-PUCK-HISTORY` and `MAX-ROBOT-HISTORY` values of 10, 30, 50, and 100. The transition function achieves a stable division of labor in all cases; however, the rate of convergence slows at larger history lengths. This is intuitive when one considers a robot with a larger history value takes longer to purge its history of an outdated puck proportion estimate after a change in puck type proportions (e.g., at time 10000 and 20000). The fast initial convergence for all history lengths is due to the fact that all robots begin with their histories empty and therefore have no outdated estimates to overcome.

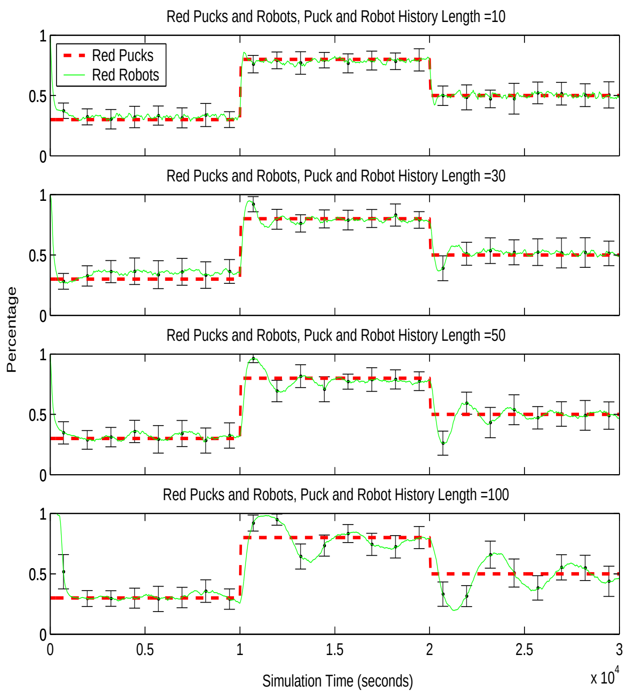

Fig. 1. Proportion of Puck$_{Red}$ pucks and robots foraging for Puck$_{Red}$ pucks over time when using the probabilistic transition function and different puck and robot history lengths, shown with 1 standard deviation error bars.

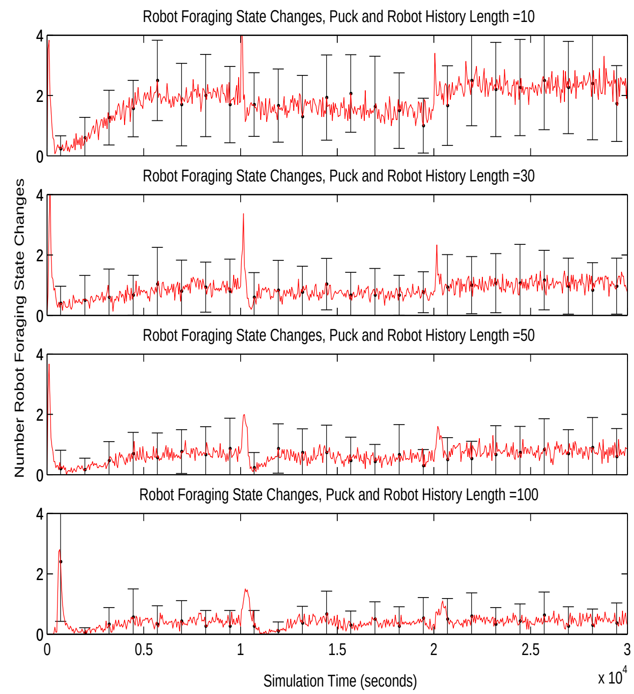

Fig. 2. The number of foraging state changes when using the probabilistic transition function and different puck and robot history lengths, shown with 1 standard deviation error bars.

Another factor in evaluating the performance of this method is through the frequency by which individual robots switch tasks. In some task domains, switching between tasks can be very expensive and should therefore be avoided. Figure 2 shows the cumulative number of times the robots change state during the course of the experiments. The data points are obtained by summing the total number of forage state changes over the course of the previous 50 seconds of the experiment (it is possible that a single robot could change foraging state more than once during this interval). As the plots show, the shorter the puck and robot history lengths, the more foraging state changes occur.

In general, shorter puck and robot history lengths result in faster convergence to the desired division of labor but lead to higher frequency oscillations due to more frequent changes in individual robot foraging state. For a given environment, the appropriate transition function and puck and robot history lengths depend on factors such as the expense of task changes for a robot, the frequency of environment changes, and the speed of such changes.

VI. Conclusions

We have presented a Large-Scale Minimalist Multi-Robot System (LMMS), composed of 20 simulated mobile robots, in which the individual robots maintain a minimal amount of state information, extract a limited amount of information from available sensors, and cannot actively or directly communicate with other robots in the system. Using this LMMS, we have demonstrated a method by which to achieve a desired division of labor in a concurrent foraging task domain, experimentally validated it in a realistic simulation, and demonstrated its robustness and adaptability to environmental perturbations.

Our division of labor mechanism is achieved in a completely distributed manner by having each individual robot maintain a limited history of observed activities of other robots and tasks which need to be performed. Each robot independently estimates the current division of labor of the group over the set of observed tasks to be performed using this history of local observations and potentially modifies its own behavior in an attempt to bring the global division of labor over the observed set of tasks closer to the desired level.

VII. Acknowledgments

This work is supported in part by Defense Advanced Research Projects Agency (DARPA) Grant F30602-00-2-0573 and in part by National Science Foundation Grant EIA-0121141. The authors wish to thank Kristina Lerman and Aram Galstyan for their informative comments and suggestions that greatly improved this work.

References

[1] W. Agassounon and A. Martinoli. Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. In *First Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, pages 1090–1097, Bologna, Italy, 2002. ACM Press.

[2] R. Beckers, O. Holland, and J. Deneubourg. From local actions to global tasks: Stigmergy and collective robotics. In R. Brooks and P. Maes, editors, *Artifical Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pages 181–189, Cambridge, MA, 1994. MIT Press.

[3] E. Bonabeau, G. Théraulaz, and J. Deneubourg. Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies. *Proceedings Royal Society of London B*, 263:1565–1569, 1996.

[4] J. Fredslund and M. J. Matarić. A general, local algorithm for robot formations. *IEEE Transactions on Robotics and Automation, Special Issue on Multi-Robot Systems*, 18(5):837–846, 2002.

[5] B. Gerkey, R. Vaughan, K. Støy, A. Howard, G. Sukhatme, and M. Matarić. Most valuable player: A robot device server for distributed control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS-01)*, pages 1226–1231, Maui, Hawaii, 2001.

[6] B. P. Gerkey and M. J. Matarić. Sold!: Auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation, Special Issue on Multi-Robot Systems*, 18(5):758–768, 2002.

[7] Dani Goldberg and Maja J. Matarić. Design and evaluation of robust behavior-based controllers for distributed multi-robot collection tasks. In Tucker Balch and Lynne E. Parker, editors, *Robot Teams: From Diversity to Polymorphism*, pages 315–344. AK Peters, 2002.

[8] O. Holland and C. Melhuish. Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*, 5(2):173–202, 2000.

[9] M. B. Krieger and J.-B. Billeter. The call for duty: Self-organized task allocation in a population of up to twelve mobile robots. *Robotics and Autonomous Systems*, 30(1-2):65–84, 2000.

[10] C. Kube and H. Zhang. The use of perceptual cues in multi-robot box-pushing. In *IEEE International Conference on Robotics and Automation*, pages 2085–2090, Minneapolis, Minnesota, 1996.

[11] K. Lerman and A. Galstyan. Macroscopic analysis of adaptive task allocation in robots. In *IEEE International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada, Oct 2003.

[12] A. Martinoli, A. J. Ijspeert, and F. Mondada. Understanding collective aggregation mechanisms: From probabilitic modeling to experiments with real robots. *Robotics and Autonomous Systems*, 29:51–63, 1999.

[13] M. Matarić. Designing and understanding adaptive group behavior. *Adaptive Behavior*, 4:1:51–80, December 1995.

[14] L. E. Parker. Alliance: An architecture for fault tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

[15] G. Théraulaz, Bonabeau E., and J. Deneubourg. Threshold reinforcement and the regulation of division of labour in insect societies. *Proceedings Royal Society of London B*, 265:327–335, 1998.

[16] R. Vaughan. Stage: A multiple robot simulator. *Institute for Robotics and Intelligent Systems Technical Report IRIS-00-394, Univ. of Southern California*, 2000.

[17] B. Werger and M. Matarić. Robotic ”food” chains: Externalization of state and program for minimal-agent foraging. In P. Maes, M. Matarić, J.A. Meyer, J. Pollack, and S. Wilson, editors, *From Animals to Animats 4, Fourth International Conference on the Simulation of Adaptive Behavior (SAB-96)*, pages 625–634, Cape Cod, Massachusetts, 1996.

[18] B. B. Werger. Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams. *Artificial Intelligence*, 110:293–320, 1999.