Reward and Diversity in Multirobot Foraging

Tucker Balch
Computer Science Department
Carnegie Mellon University

`http://www.cs.cmu.edu/ trrb`

Abstract

This research seeks to quantify the impact of the choice of reward function on behavioral diversity in learning robot teams. The methodology developed for this work has been applied to multirobot foraging, soccer and cooperative movement. This paper focuses specifically on results in multirobot foraging. In these experiments three types of reward are used with Q-learning to train a multirobot team to forage: a *local* performance-based reward, a *global* performance-based reward, and a heuristic strategy referred to as *shaped reinforcement*. Local strategies provide each agent a specific reward according to its own behavior, while global rewards provide all the agents on the team the same reward simultaneously. Shaped reinforcement provides a heuristic reward for an agent's action given its situation. The experiments indicate that local performance-based rewards and shaped reinforcement generate statistically similar results: they both provide the best performance and the least diversity. Finally, learned policies are demonstrated on a team of Nomadic Technologies' Nomad-150 robots.

1 Introduction

Most research in multirobot systems has centered on homogeneous teams, with work in heterogeneous systems focused primarily on mechanical and sensor differences (e.g. Parker's work [9]). In contrast, this research examines teams of mechanically identical robots. These systems are interesting because they may be homogeneous or heterogeneous depending only on the behavior of the agents comprising them. Behavior is an especially flexible dimension of heterogeneity in learning systems because the agents converge to hetero- or homogeneous solutions on their own.

This investigation is focused on quantifying the relationship between the type of reward used to train a robot team and the diversity and performance of the resulting system. This paper reports results in the multirobot foraging domain, but the same methodology has also been applied to robot soccer and cooperative movement tasks. For a complete description of the results in all three domains the reader is referred to [5].

Previously, foraging robot teams were configured as either homogeneous or heterogeneous *a priori*, then their performance comparatively evaluated. In one representative study, Goldberg and Matarić evaluate the relative merits of heterogeneous and homogeneous behavior in foraging tasks [7]. Like the research reported in this paper, their work focuses on mechanically identical, but behaviorally different agents. To reduce robot-robot interference in foraging they suggest *pack* and *caste* arbitration as mechanisms for generating efficient behavior. In the pack scheme, each agent is arbitrarily assigned a place in the "pack hierarchy." Agents higher in the hierarchy are permitted to deliver attractors before the others. In the caste approach, only one agent completes the final delivery; the other robots leave their attractors on the boundary of a designated "home zone." They find that the homogeneous systems performed best.

In another investigation, Balch demonstrates a relationship between diversity and performance in hand-coded foraging teams [3]. He compares the performance of two heterogeneous and one homogeneous strategy. The performance of each system is evaluated in simulation and also ranked according to an information theoretic measure of diversity called *social entropy* [5]. The results indicate strong negative correlation between performance and diversity in multirobot foraging systems — i.e. homogeneity is preferred in this task.

The research reported here is distinguished from other work because diversity is investigated as an *outcome* rather than an initial condition of robot experiments. This approach enables the investigation of diversity from an ecological point of view — as an emergent property of agents interacting with their environment. The robots in this research are initialized with random policies, then allowed to learn (using one of several reward strategies). Performance and diversity are evaluated after the agents have converged to stable policies.

Reinforcement learning plays a growing role in the programming of autonomous multirobot teams. A key issue in this field is how to select appropriate reward functions for the learning robots. In the most closely related multiagent reinforcement learning work Matarić asserts that the delayed reinforcement often utilized in Q-learning hinders an agent's ability to learn quickly [8]. Instead,

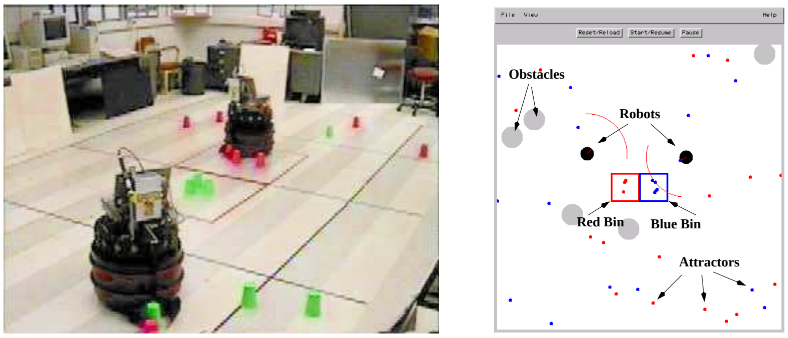

Figure 1: Real and simulated robot foraging. Left: two robots forage for colored attractors in the laboratory; after grasping an object, they deposit it in one of two delivery zones according to color. Right: in simulation, robots are represented as black circles, arcs indicate the robots' visual sensing range, obstacles are drawn as gray circles, the small discs are attractors. The robots deliver the attractors to the color-coded squares representing delivery areas.

she proposes a heuristic strategy called *shaped reinforcement* to speed and improve learning performance. In this paper we compare the performance and diversity of foraging robot teams trained using shaped reinforcement with others using delayed rewards.

The rest of this paper is organized as follows: The next section describes the multiagent foraging task in more detail. Later sections explain the development of behaviors and reward functions used to train robots to accomplish the task. The quantitative performance of the resulting systems is compared Section 5. Diversity is examined in Section 6. Section 7 describes the implementation of the foraging behaviors on mobile robots. We conclude with a review of the results and a discussion of their implication.

2 The multi-foraging task

The forage task for a robot is to wander about the environment looking for items of interest (attractors). Upon encountering an attractor, the robot moves towards it and grasps it. After attachment, the robot returns the object to a specified home base. Foraging has a strong biological basis. Many ant species, for instance, perform the forage task as they gather food. Foraging is also an important subject of research in the mobile robotics community; it relates to many real-world problems [1; 2; 4; 7; 6]. Among other things, foraging robots may find potential use in mining operations, explosive ordnance disposal, and waste or specimen collection in hazardous environments (e.g. a planetary rover).

In most robotic foraging research to date the robots collect attractors of a single type and deliver them to a single destination. This basic task is referred to as *simple foraging*. Simple foraging is an important robotic capability, but many practical industrial and military tasks call for more functionality. Consider, for example, a janitorial robot responsible for collecting and sorting recyclable trash objects into glass, aluminum and paper bins. Similarly, many assembly and construction tasks involve collecting parts or materials and placing them in a specific location. These more complex tasks are referred to as *multi-foraging* tasks. In general, the multi-foraging task calls for several types of objects to be collected and placed in specific locations according to type. Here *multi* refers to the multiple types of object to deliver, not the number of robots engaged in the task. An example of robots executing a multi-foraging task is presented in Figure 1.

Performance in the multi-foraging task is measured as the number of attractors collected and properly delivered by the robots in a 10 minute trial. Several environmental parameters affect the rate at which the agents collect and deliver the attractors including the number of attractors, obstacles in the environment, playing field size and the number of robots.

The following conditions were present in simulation experiments: 40 attractors (20 of each type, red and blue) and five 1 m^2 obstacles (5% coverage) randomly distributed about a 10 by 10 meter field with one to eight simulated robots. In laboratory runs there were 20 attractors and no obstacles (except arena boundaries) on a 5 by 10 meter playing field with one or two robots.

3 Behaviors for multi-foraging

A schema-based reactive control system is used for robot programming. In this approach, an agent is provided several pre-programmed skills (or more formally, behavioral assemblages) that correspond to steps in achieving the task (e.g. *wander*, *acquire*, *deliver*, and so on). Binary perceptual features are used to sequence the robot through steps in achieving the task. Selection of the appropriate behavior, given the situation, may be programmed by hand or discovered by the robot through reinforcement learning. In addition to the learning strategies investigated here, these behaviors were also used to build successful hand-coded foraging strategies, including a winning entry in the AAAI-97 Robot Competition [3].

A range of skills were developed to support a number of foraging strategies and to avoid bias towards any particular approach. The repertoire is suitable for building behaviorally homogeneous foraging teams as well as various heterogeneous strategies. The behaviors are summarized below:

- *wander:* move randomly about the environment in search of attractors. Upon encountering an attractor, most agents learn to transition to an appropriate *acquire* behavior.
- *stay_near_home:* similar to the *wander* assemblage, but with an additional attractive force to keep the agent close to the homebase. This assemblage might be utilized in a territorial foraging strategy.
- *acquire_red:* move towards the closest visible red attractor. When close enough to grasp the attractor, most agents learn to close their gripper and transition to a *deliver* assemblage.
- *acquire_blue:* move towards the closest visible blue attractor.
- *deliver_red:* move towards the red delivery area. When close enough to deposit the attractor in the delivery area, most agents learn to open their gripper and transition to one of the *wander* assemblages.
- *deliver_blue:* move towards the blue delivery area.

All of the above behaviors include a provision for obstacle and robot avoidance.

Depending on its *perceptual state* (an abstract representation of the agent's situation) each robot selects which of the six behaviors to activate at each movement step. In the language of the reinforcement learning community, agent learns to select an *action* (behavior/skill) depending on it's *state* (perceptual state). The association of actions to states specifies the robot's *policy*.

The perceptual state is a combination of nine *perceptual features*. Each feature is a single, abstracted bit of environmental or sensor state germane to the robot's task (e.g. whether or not the robot is holding an attractor in its gripper). The perceptual features used in this work are cataloged in Table 1. In addition to the features advising the robot whether an attractor is visible, there are also features indicating whether attractors are visible outside the delivery (or "home") zone. The visibility cues are used to allow hand-coded territorial agents (reported in separate work [3]) to search for attractors at a distance from the delivery zone (home zone) while ignoring the others (and *vice-versa*).

Instead of being provided a pre-coded sequencing strategy however, the robots in this work must learn an effective policy as they interact with the environment and are provided feedback (in the form of a reward). The learning agents are provided information about the environment in the 9-bit perceptual state vector. Altogether there are 512 potential perceptual states. In practice however, some states never occur. It is impossible, for instance, for a robot to be both in the red delivery area and outside the home zone simultaneously.

4 Learning strategies for foraging

The approach is to provide each agent a reward function that generates feedback at each movement step regarding the agent's progress, then to use that function over many trials to train the robot team. Q-learning is used to associate actions with state. The learning agents are initialized with random Q-tables, thus random, poorly performing policies. Since each agent begins with a different policy, the teams are initially maximally diverse. They improve their policies using the reinforcement functions described below.

The reinforcement function used to train a robot is usually closely coupled to the performance metric for the task. In fact in many reinforcement learning investigations performance, task and reward are viewed as one and the same. Since learning agents strive to maximize the reward signal provided them, performance is maximized when their reward closely parallels performance. It is sometimes the case however, that robots cannot or should not be rewarded strictly according to overall system performance. Some examples include: the robot's sensors do not provide enough information for an accurate evaluation of performance; the delay in receiving a reward is too great — learning a sequential task is too difficult and/or takes too long; performance depends on the actions of other robots over which the agent has limited knowledge and/or control. As a result, the performance metric (task) and reward function are often quite different and must be treated separately. A taxonomy introduced by Balch is adopted to help distinguish between the various reward functions investigated in this work [5].

Three reward functions are investigated here:

- **Local performance-based reinforcement:** each agent is rewarded individually when it delivers an attractor.
- **Global performance-based reinforcement:** all agents are rewarded when any agent delivers an attractor.
- **Local shaped reinforcement:** each agent is rewarded progressively as it accomplishes portions of the task [8].

In both types of performance-based reinforcement the reward is tied directly to the performance metric; in this case, attractor delivery. A performance-based reward is advantageous for the designer because it allows her to succinctly express the task for an agent. There is no need to enumerate how the task should be carried out (as is necessary in hand-coded teams). Instead, the agents learn behavioral sequences autonomously. In contrast, heuristic or shaped reinforcement functions provide rewards to the agent as it achieves parts of the task; for instance, when grasping an attractor, when heading for the delivery area, and when depositing it in the delivery area.

Assuming the task proceeds in discrete steps, the local performance-based reinforcement function for foraging at timestep t is:

$$R_{\text{local}}(t) = \begin{cases} 1 & \text{if the agent delivered} \\ & \text{an attractor at time } t-1. \\ -1 & \text{otherwise.} \end{cases}$$

perceptual feature	meaning
`red_visible`	a red attractor is visible.
`blue_visible`	a blue attractor is visible.
`red_visible_outside_homezone`	a red attractor is visible outside the three meter radius home zone.
`blue_visible_outside_homezone`	a blue attractor is visible outside the home zone.
`red_in_gripper`	a red attractor is in the gripper.
`blue_in_gripper`	a blue attractor is in the gripper.
`close_to_homezone`	the agent is within 3 meters of the homebase.
`close_to_red_bin`	close enough to the red delivery area to drop an attractor in it.
`close_to_blue_bin`	close enough to the blue delivery area to drop an attractor in it.

Table 1: Perceptual features available to the foraging robots. Each feature is one bit of environmental state; the entire perceptual state is a nine-bit value.

The global performance-based function is defined as:

$$R_{\mathrm{global}}(t) = \begin{cases} 1 & \text{if any agent delivered.} \\ & \text{an attractor at time } t-1. \\ -1 & \text{otherwise} \end{cases}$$

The global function will reward all team members when an attractor is delivered. The global function is implemented using an inter-robot communication scheme that allows the agents to communicate their individual rewards. In terms of the reinforcement function taxonomy developed in [5], R_{global} and R_{local} are similar in that they are both `INTERNAL_SOURCE`, `PERFORMANCE`, `DELAYED` and `DISCRETE` reward functions. Of course they differ in locality; one is `LOCAL` while the other is `GLOBAL`

A potential problem with these reward functions is that the reinforcement is *delayed.* The agent must successfully complete a sequence of steps before receiving a reward. This makes credit assignment in the intervening steps more difficult. To address this issue, Matarić has proposed an alternate reward scheme where the agent is provided intermediate rewards as it carries out the task [8]. The agent is not only rewarded for delivering an attractor, but also for picking one up, for moving towards a delivery area when it is holding an attractor, and so on. This heuristic strategy, referred to as *shaped reinforcement*, is defined as a sum of three component functions:

$$R_{\mathrm{shaped}}(t) = R_{\mathrm{event}}(t) + R_{\mathrm{intruder}}(t) + R_{\mathrm{progress}}(t)$$

$R_{\mathrm{event}}(t)$ encapsulates the reward for events like delivering an attractor or dropping it in the wrong place. $R_{\mathrm{intruder}}(t)$ is used to punish the agent for prolonged interference with other agents. Finally, $R_{\mathrm{progress}}(t)$ is activated when the agent is holding an attractor, and rewards the agent for moving towards the delivery point. $R_{\mathrm{event}}(t)$ is defined more formally as:

$$R_{\mathrm{event}}(t) = \begin{cases} 1 & \text{if delivered attractor} \\ & \text{at time } t-1. \\ 1 & \text{if picked up attractor} \\ & \text{at time } t-1. \\ -3 & \text{if dropped attractor} \\ & \text{outside bin at time } t-1. \\ -1 & \text{otherwise.} \end{cases}$$

Matarić sets R_{event} to 0 in the default case, instead of -1 as above. The choice was made to use -1 here because Q-learning converges more quickly with negative rewards before task completion. $R_{\mathrm{progress}}(t)$ is defined as:

$$R_{\mathrm{progress}}(t) = \begin{cases} 0.5 & \text{if holding attractor and moving} \\ & \text{towards bin at time } t-1. \\ -0.5 & \text{if holding attractor and moving} \\ & \text{away from bin at time } t-1. \\ 0 & \text{otherwise.} \end{cases}$$

Because the individual behaviors used in this work already include a provision for agent avoidance, $R_{\mathrm{intruder}}(t)$ is not used. R_{shaped} is an `INTERNAL_SOURCE`, `HEURISTIC`, `IMMEDIATE`, `DISCRETE` and `LOCAL` reward function.

5 Performance results

Statistical results were gathered in thousands of simulation trials. Each type of learning system under investigation was evaluated using one to eight simulated robots in five randomly generated environments. Performance is evaluated as the number of attractors collected in 10 minutes. 300 trials were run in each environment, or 12,000 runs overall.

Agents are able to learn the task using all three types of reinforcement. A plot of the average performance for each learning strategy versus the number of agents on the team is presented in Figure 2. (In separate research, the performance of three different hand-coded systems was also evaluated [3]; performance of the best hand-coded system (a homogeneous strategy) is included in the graph for comparison).

The plot shows that, of the learning strategies, local performance-based and heuristic (shaped) reinforcement systems perform best. Performance in the globally reinforced system is worse than the other learning teams. Note that the performance plots for teams using local and shaped rewards are nearly identical and that one's confidence interval overlaps the other's mean value. Both also overlap the performance of the hand-coded homogeneous policy. In fact, there is no statistically significant difference between the homogeneous hand-coded systems and the best learning systems. **Local and shaped reinforcement systems perform as well as the best hand-coded systems**.

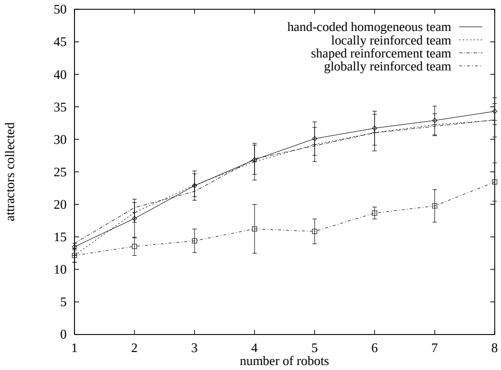

Figure 2: Performance of foraging teams versus the number of robots on a team. The errorbars indicate 95% confidence intervals.

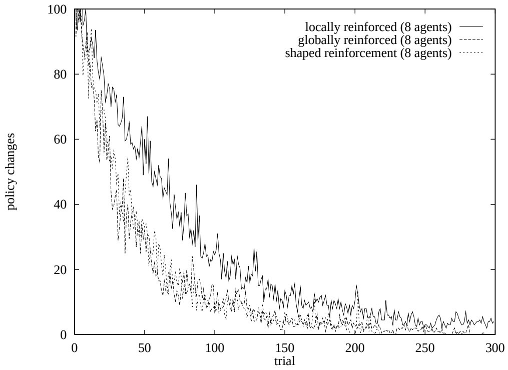

Figure 3: Convergence for learning systems, measured as policy changes per trial; low numbers indicate convergence to a stable policy.

The rate at which agents converge to stable policies is evaluated by tracking the number of times an agent's policy changes during each trial. A policy change is a revision of the agent's Q-table such that it will select a different action in some perceptual state. The average number of policy changes per trial is graphed for each system in Figure 3. The figure shows plots for systems with eight agents. All three reinforcement strategies show good convergence properties, but the systems using shaped reinforcement converge the quickest.

6 Diversity results

Previously, diversity in multirobot teams was evaluated on a bipolar scale with systems classified as either *heterogeneous* or *homogeneous*, depending on whether any of the agents differ [6; 7; 9]. Unfortunately, this labeling doesn't tell us much about the *extent* of diversity in heterogeneous teams.

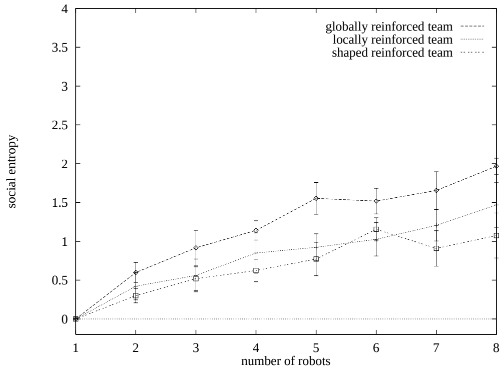

Figure 4: Social entropy (diversity) versus size of the team for learning teams; larger numbers indicate greater diversity, error bars indicate 95% confidence intervals.

Heterogeneity is better viewed on a sliding scale providing for quantitative comparisons. Such a metric enables the investigation of issues like the impact of diversity on performance, and conversely, the impact of other task factors on diversity. *Social entropy*, inspired by Shannon's information entropy [11], is used as a measure of diversity in robot teams. The metric captures important components of the meaning of diversity, including the number and size of groups in a society. Social entropy is briefly reviewed here. For more details please see [5].

To evaluate the diversity of a multirobot system, the agents are first grouped according to behavior [1] (e.g. all red-collecting agents are placed in one group). Next, the overall system diversity is computed based on the number and size of the groups. Social entropy for a multirobot system composed of M groups is defined as:

$$H(X) \quad = \quad -\sum_{i=1}^{M} p_i \, \log_2(p_i) \qquad (1)$$

where p_i represents the proportion of agents in group i. We will use this metric in the evaluation of the experimental foraging strategies.

The average diversity is computed for robot teams trained with each type of reinforcement. Results are plotted versus the size of robot teams in Figure 4. In all cases with two or more agents, the globally reinforced teams are most diverse. In all but one case the teams using shaped reinforcement are the least diverse and locally reinforced teams lie between the two extremes.

Spearman's Rank-order Correlation Test is used to evaluate the relationship between diversity and performance in these systems [10]. The test measures the correlation between rankings in one dimension (e.g. performance) and another (e.g. diversity). Spearman's test indicates the rankings are strongly negatively correlated,

[1] We use numerical hierarchical overlapping clustering techniques to group agents according to policy similarities. Please see [5] for details.

configuration/trial	performance before training	after
1 robot trial 1	1.0	9.0
trial 2	0.0	10.0
trial 3	0.0	8.0
trial 4	0.0	7.0
trial 5	0.0	8.0
average	**0.2**	**8.4**
2 robots trial 1	0.0	15.0
trial 2	1.0	15.0
trial 3	0.0	16.0
trial 4	1.0	14.0
trial 5	0.0	13.0
average	**0.4**	**14.6**

Table 2: Summary of performance in learning foraging robot trials. Policies learned using local performance-based rewards were used in all trials.

with $r = -0.96$. The probability of the null hypothesis being true (that the rankings occur by chance) is 0.000028. **Diversity and performance are negatively correlated in these learning teams.**

7 Implementation on mobile robots

To verify the simulation results, the learning systems were ported to Nomad 150 mobile robots. The Java-based behavioral configuration system used in this work enables the behaviors and features to be utilized on mobile robots and in simulation. **Identical control software was employed in simulation and on the mobile robots.**

Performance was evaluated before and after learning using local performance-based rewards on one and two robots. In each case, the robots were initialized with a random policy (the behavior for each situation is set randomly), then evaluated in a 10 minute trial. The Q-tables were transferred to the simulation system and trained for 300 trials. After training, the policies were transferred back to the robots for another evaluation. The process was repeated five times for each number of robots. Performance of the robots running learned policies is summarized in Table 7. A photograph one of the mobile robot trials is presented in Figure 1.

As in simulation the robots perform much better after the learning phase. However, they do not collect as many attractors as comparable simulated systems. This is due to the reduced number of attractors available for collection.

8 Discussion and summary

The experimental results reported here show that the choice of reinforcement function significantly impacts the diversity and performance of learning teams in a foraging task. Separate studies (using the same methodology) in robot soccer and cooperative movement support this result in other domains as well [5].

Interestingly, the relationship between diversity and performance in soccer (positive correlation), is exactly opposite the relationship reported for foraging in this work (negative correlation). The reasons for this difference aren't known for certain, but we believe they are due to the differences in task. Soccer is unavoidably a *team* activity while foraging can be accomplished by an individual agent. We believe that when multiple agents are required, it is more likely that the team will benefit from diversity.

These experiments in foraging show that agents using local reinforcement strategies converge to more homogeneous societies and perform better than robots using a global reward structure. Greater homogeneity with local reinforcement is due to the fact that individuals are rewarded for their own actions, thus making reinforcement of the same state/action pair more likely in different agents than with global reinforcement. The relationship between diversity and performance is exactly opposite that found in robot soccer experiments (reported separately), but in both soccer and foraging, local rewards lead to greater homogeneity [5].

In addition to the local and global performance-based reward structures, a local heuristic, or *shaped reinforcement* method was evaluated [8]. In these experiments teams trained using shaped reinforcement learn the task more quickly (converge faster) than teams using delayed rewards. However, after approximately 150 trials the performance of systems using shaped reinforcement is nearly identical to that of systems using delayed performance-based rewards. In general we believe "standard" performance-based rewards are preferable to tailored heuristic rewards because they provide greater generality and less programmer bias. But when quick learning is imperative, shaped rewards may be a better choice.

The diversity of each system was evaluated using the social entropy metric introduced in [5]. Globally-rewarded teams were found to be the most diverse, followed by the locally rewarded teams. Teams using shaped reinforcement were the least diverse. This is because agents using shaped reinforcement are provided more uniform "guidance" in finding a policy, and are thus less likely to settle on diverse solutions. In these learning systems, diversity and performance are negatively correlated with $r = -0.96$ and prob $= 0.000028$.

References

[1] R.C. Arkin. Cooperation without communication: Multi-agent schema based robot navigation. *Journal of Robotic Systems*, 9(3):351–364, 1992.

[2] R.C. Arkin, T. Balch, and E. Nitz. Communication of behavioral state in multi-agent retrieval tasks. In *Proceedings 1993 IEEE Conference on Robotics and Automation*, Atlanta, GA, 1993.

[3] T. Balch. The impact of diversity on performance in multirobot foraging. In *Proc. Autonomous Agents 99*, Seattle, WA, 1999.

[4] T. Balch and R.C. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1), 1995.

[5] Tucker Balch. *Behavioral Diversity in Learning Robot Teams.* PhD thesis, College of Computing, Georgia Institute of Technology, 1998.

[6] M. Fontan and M. Mataric. A study of territoriality: The role of critical mass in adaptive task division. In *From Animals to Animats 4: Proceedings of the Fourth International Conference of Simulation of Adaptive Behavior*, pages 553–561. MIT Press, 1997.

[7] D. Goldberg and M. Mataric. Interference as a tool for designing and evaluating multi-robot controllers. In *Proceedings, AAAI-97*, pages 637–642, July 1997.

[8] Maja Matarić. Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83, January 1997.

[9] Lynne E. Parker. *Heterogeneous Multi-Robot Cooperation.* PhD thesis, M.I.T. Department of Electrical Engineering and Computer Science, 1994.

[10] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C.* Cambridge University Press, 1988.

[11] C. E. Shannon. *The Mathematical Theory of Communication.* University of Illinois Press, 1949.