Adaptive Robot Coordination using Interference Metrics[1]

Avi Rosenfeld, Gal A Kaminka, and Sarit Kraus
Bar Ilan University
Department of Computer Science
Ramat Gan, Israel
{rosenfa, galk, sarit}@cs.biu.ac.il

Abstract. One key issue facing robotic teams is effective coordination mechanisms. Many robotic groups operate within domains where restrictions such as limiting areas of operation are liable to cause spatial conflicts between robots. Our previous work proposed a measure of coordination, *interference*, that measured the total time robots dealt with resolving such conflicts. We found that a robotic group's productivity was negatively correlated with interference: Effective coordination techniques minimized interference and thus achieved higher productivity. This paper uses this result to create adaptive coordination techniques that are able to dynamically adjust the efforts spent on coordination to match the number of perceived coordination conflicts in a group. Our robots independently calculate a projected level of interference they will encounter. By using this metric as a guide, we are able to create adaptive coordination methods that can quickly and effectively adjust to changing conditions within their environment. We present an adaptation heuristic that is completely distributed and requires no communication between robots. Using thousands of simulated trials, we found that groups using this approach achieved a statistically significant improvement in productivity over non-adaptive coordination methods.

1 Introduction

Groups of robots are likely to accomplish certain tasks more quickly and robustly than single robots [3, 5]. However, the physical environment where such teams operate often pose a challenge for the robots to perform properly. For example, domains such as robotic search and rescue, vacuuming, and waste cleanup are all characterized by limited operating spaces where the robots are likely to collide. Improved coordination methods in such domains result in more productive groups.

Our previous work [9] defined a measure called *interference* to facilitate comparison between various coordination methods. Interference is defined as the total time each robot spends in resolving conflicts with other robots. This not only includes the time robots collide, but also the time robots spend preventing such collisions and the time they engage in resolution behaviors after such an event. It was found that a strong negative correlation exists between interference in a group and its productivity. However, this does not mean that robots should avoid the coordination activities which constitute interference, as such behaviors are often critical for achieving cohesive team behavior. Rather, the coordination method of choice needs to appropriately match the needs of the domain. As such, interference should be kept to a minimum, while still sufficiently high to meet the coordination requirements of the environment.

This paper builds on this idea by presenting a method for dynamically adapting coordination efforts to minimize interference by taking into account fluctuations in spatial conflicts that are common in robotic domains. The idea is to spend more efforts on coordination as the possibility of collisions become frequent, and to reduce such efforts if such events are rare. In order to quickly adapt to a changing environment, we use a weight-based heuristic by which every robot in the group is capable of quickly tweaking its resolution methods to match its estimates of resource conflicts. Our approach is completely distributed, and requires no communications between robots.

To evaluate this approach, we modified two different basic robot coordination mechanisms to use this adaptation heuristic. We ran extensive simulated trials comparing the adaptive methods to a variety of non-adaptive versions. We show that the adaptive methods result in statistically significant higher average productivity than that of non-adaptive methods. While a specific non-adaptive method may work well in small groups, but not in large groups, the adaptive approach we use results in improved performance regardless of the robotic group size.

The remainder of this paper is organized as follows. The next section presents the correlation between interference and a group's productivity. We discuss the problem of matching the best coordination method to a given domain. Section 3 develops our adaptive coordination algorithm, and introduces our hypothesis that such an approach can effective adapt to the dynamic nature of many robotic domains. Such a method will be able to overcome the shortcoming in static methods. In section 4 we present and evaluate our experiments with dynamic groups to confirm this hypothesis. We discuss related work in section 5. Section 6 concludes and describes possible future directions.

2 Interference versus Productivity

A strong inverse relationship seems to exist between a robotic group's productivity and the amount of time these robots engage in coordination behaviors. We previously found [9] a strong negative correlation between the total amount of time robots spend in resolution behaviors, a concept referred to as *interference*, and the productivity of the group. While adding robots may speed up the time to complete certain tasks, and can even be necessary for completing other tasks, these robots can trigger collisions which detract from the group's performance.

Our previous work [9], contrasted various coordination algorithms within the foraging domain. The foraging domain has been exten-

[1] This material is based upon work supported in part by the NSF under grant #0222914 and by the BSF under grant #2002401. Sarit Kraus is also affiliated with UMIACS.

sively studied, and is formally defined as locating target items from a search region *S*, and delivering them to a goal region *G* [4]. Various coordination methods have been developed that could work within this domain [10, 3, 11, 8]. We compared algorithms including the concepts of *Aggression* [11], a *dynamic Bucket Brigade* [8], and the use of a repulsion schema mechanism (*Noise* group) [1]. Among others, we compared three additional groups called *Gothru*, *Repel Fix* and *Timeout*. Gothru represents idealized group behavior without any possibility for interference and can only exist in simulation. These robots were never affected by obstacles, and were allowed to simply pass through teammates. Repel Fix resolved collisions by moving away from a teammate for a fixed period of 50 seconds once a teammate was sensed within one robot width (approximately 45 cm). The Timeout method only reacted once a robot detected it had not sufficiently moved for 5 seconds. After this point, it attempted to become unstuck by entering a random walk for 7.5 seconds.

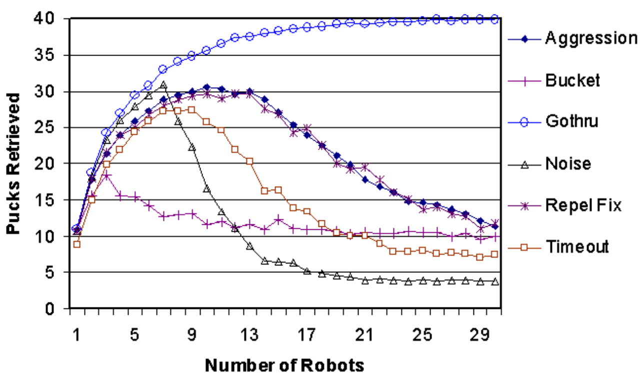

Figure 1. Comparing Foraging Coordination Methods

Figure 1 graphically presents the results that motivate this work. The X-axis represents the number of robots in the group, and the Y-axis corresponds to the number of foraging pucks that the group brought to the goal region. Notice how Gothru is the only group to achieve positive gains in productivity over all group sizes. The levels of interference that existed in all other groups eventually caused the group's productivity to decrease with the addition of robots. We found a very high negative correlation between the total time groups spent reasoning about and reacting to collisions, and the corresponding productivity. However, no one group was successful in minimizing this level of interference across all group sizes. Our conclusion was that static coordination methods are often not equally suited for minimizing interference over all conditions.

Our work is motivated by these results. Our goal is to create adaptable coordination methods that are able to react to the dynamic conditions within robotic domains. We present a method of adaptation that is based on the robot's internal estimate of interference. For example, the Noise group in this domain had simple coordination methods that were effective in small groups. However, these methods were not capable of resolving collisions in larger groups – resulting in high interference levels and low productivity. Other methods, such as the Aggression and Repel Fix groups spent more time in coordination behaviors. These more aggressive coordination methods were capable of reducing interference at higher levels, and thus approximate Gothru's theoretical performance in larger group sizes. However, the high overhead of these coordination methods itself constituted interference and resulted in lower productivity than simpler coordination methods in small groups. We believe that by using a robot's internal measure of interference, we can create one coordination method that adapts to the domain conditions as needed.

3 Adaptive Coordination

The dynamic nature of robotic environments makes the challenge of creating adaptive coordination formidable. While traditional reinforcement learning methods may be useful, the number of iterations such algorithms require make them unproductive without a significant training period [7]. In the previous example, each team's productivity result was averaged over 100 trials for statistical significance. While 100 trials may be sufficient for reinforcement learning, the result would likely be optimal for specific environment settings. Once that environment is slightly modified, as would occur if one robot ceased functioning, the result would no longer be relevant. As productivity of robotic groups is often time critical, a tradeoff between finding an optimal solution and speed is likely to be worthwhile. We therefore instead focus on an adaptive method which uses a weighting heuristic to dynamically modify the coordination algorithms to match perceived environmental changes. We present the advantage of this method compared to static methods.

3.1 The Dynamic Coordination Algorithm

We begin by analyzing the *Repel* and *Timeout* coordination methods previously mentioned. As our next section demonstrates, the best length of time to spend in Repel and Timeout behaviors depends on the nature of the domain. Once again, a strong correlation between interference and productivity emerges — the longer a robot engages in interference resolution behaviors the lower its productivity will be. For example, if a Repel robot repels for too long after a potential collision, it will take longer to complete its task. However, in situations where collisions are likely to occur, too short a repulsion period results in the robot not resolving its projected collision and quickly re-triggering its resolution behavior for the same event. A similar problem exists in the Timeout group. If the timeout threshold is set too low, the robots will consider themselves inactive even while performing necessary tasks such as slowing down to attempt to take a target puck. Too long a timeout threshold results in the robots wasting time before attempting to resolve a legitimate problem.

We resolve this problem by basing the strength of the coordination method to match an approximation of the interference level, V, each robot senses within the domain. Specifically, our algorithm works as follows: We first initialize a base value that represents the supposed interference level the domain will contain V_{init}. For each cycle that passes where no impending collisions are detected the value of V is decreased by a certain amount W_{down}. For each cycle where the robots sense a collision is likely, the value of V is increased by a certain amount W_{up}. Thus, the value V is constantly in flux based on the robot's perception of its environment. We use this value to dictate the repel length and timeout threshold in our Repel and Timeout groups. Thus, our groups adapt their coordination methods to the likelihood of collisions within their environment.

3.2 Shortcomings of Static Methods

In order to demonstrate the shortcomings within static methods, we studied 5 variations of the Repel and Timeout groups. We chose values of 10, 50, 100, 200, and 500 cycles for the length of time the Repel group would repel after nearing a collision. We also used these same values as various threshold values for the Timeout group. After these times the random walk behavior would attempt to resolve the collisions between various groups of Timeout robots. As was the case in the previous work, we used the robotic simulator, Teambots [2], to collect data for these groups. We left other details of our setup identical to the implementation previously used. Teambots [2] simulated the activity of groups of Nomad N150 robots in a foraging area that measured approximately 5 by 5 meters. We used a total of 40 such target pucks, 20 of which where stationary within the search area, and 20 moved randomly. For each group, we measured how many pucks were delivered to the goal region by groups of 1 – 30 robots within 9 minutes. For statistical significance, we averaged the results of 50 trials with the robots being placed at random initial positions for each run. Thus, this experiment simulated a total of 15,000 trials of 9 minute intervals.

In both groups, the best coordination method depended on the size of the group. The larger the group, the more aggressive the coordination method required to fight collisions. Among the Repel groups, Repel50 had the highest productivity in the groups up to 10 robots. Between 10 and 15 robots the Repel100 group did best. The Repel200 group fared better over the next 5 robots, and the Repel500 group had the highest productivity between 20 – 30 robots. Overall, the Repel200 fared the best with an average productivity of 23.00 pucks. However, this group only had the highest productivity over a range of 5 robots. Our algorithm will need to adjust the repel value based on the values of V perceived by each robot.

Figure 2 graphically depicts the results of the equivalent timeout experiment. The X-axis represents the size of the group, and the Y-axis corresponds to the average number of pucks the group collected. In this example, the group with the highest timeout threshold fared the best with small groups. Essentially, such a high value rendered the timeout behavior dormant. As the group size grew, more aggressive treatment of interference issues was needed, and the best resolution method had a lower interference threshold. For groups of up to three robots the 500 cycle threshold worked best. Between 4 – 7 robots the Timeout200 group had the highest productivity. The Timeout100 group did best with groups of 8 and 9 robots. The Timeout50 group had the highest productivity between 10 – 20 robots, and the Timeout10 group did best with 21 – 30 robots. On average, the Timeout50 group had the best productivity with 17.60 robots. However, this group did not fare as well with smaller group sizes. Our adaptive algorithm will need to use the robots' value of V to facilitate this adaptation.

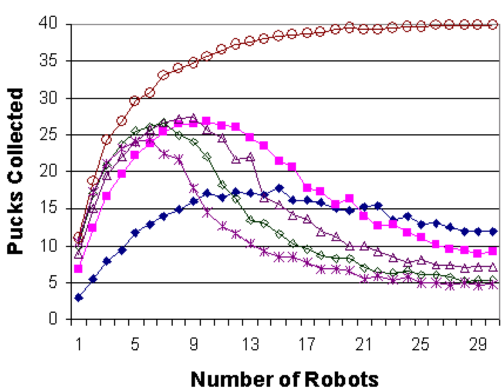

Figure 2. Static Timeout Group Productivity

4 Creating and Evaluating Dynamic Methods

In this section we discuss the process by which we set the weights used in our dynamic approach. We present details of our experiments used to create these groups. We found that this approach did indeed outperform the static methods we studied in a statistically significant fashion.

4.1 Setting the Weight Values

We experimented with various values of V_{init}, W_{up}, and W_{down} within our adaptive Repel and Timeout groups. We found many nearly optimal combinations for the values of V_{init}, W_{up}, and W_{down}. Our adaptive approach was flexible in that there were multiple weight values that resulted in similar group productivity. A value of V_{init} being originally set too high was soon corrected by the weights in W_{down}. Conversely an initial value set too low can be quickly rectified by the weights in W_{up}. Figure 3 depicts the productivity of three dynamic groups whose value of V_{init} ranged from 300 to 600 cycles. All groups used identical values for W_{up} and W_{down}, implying this level of flexibility within the system.

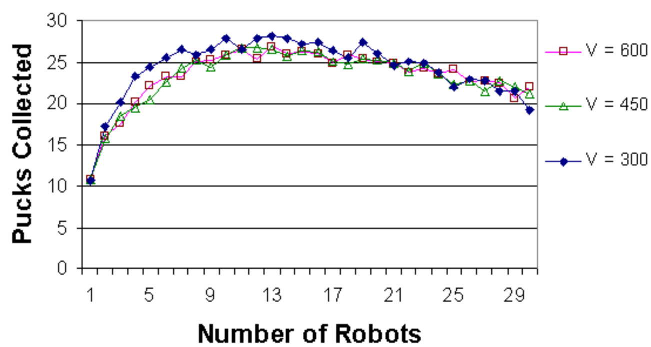

Figure 3. Three Dynamic Repelling Groups

One simple way of improving on any static method is to choose the value of V_{init} to be equal to that of the static value. For example, if trying to improve on the Repel200 method, simply set V_{init} to this amount. However, while improved results are not hard to create with this algorithm, approximating optimal results can be more difficult. We found that using a range of weights for W_{down} and W_{up} was beneficial in achieving better results. As such, W_{down} and W_{up} were modified to reflect a map of weights based on the proximity of other robots. We used different values for W_{up} when robots were actually colliding, nearly touching, and approaching but not yet within a certain distance. Optimal values, if achievable through this method, are possibly domain specific and based on the exact coordination method being used.

Within the dynamic Repel groups we studied, we found that a value of V_{init} = 350 seemed to work best. We used values for W_{down} ranging from 200 to 0 based on how quickly the repel mechanism was triggered. Our values for W_{up} ranged from 550 to 0 based on how soon the robot found itself nearing a collision. This led to a heuristic that took a graduated approach—it would adjust the amount it would repel in the case of a collision fairly quickly up or down based on how frequently collisions occurred within the domain.

Similarly we experimented with the weights the Timeout group used. Initially, V_{init} was set to 500. During every cycle that passed, all robots checked if a teammate or other obstacle was nearby and adjusted its weights for W_{down} and W_{up} accordingly. The values for W_{down} ranged from 15 down to zero based on the location of another robot. Values for W_{up} were between 2 and 0. This set of weights assumed little interference existed in the domain, but quickly lowered its threshold value as necessary.

4.2 Evaluation of Results

As figure 4 demonstrates, our heuristics did adapt their coordination measurements as the likelihood of collisions grew with the addition of robots to the group. In this graph, the X-axis represents the group size and the Y-axis marks the value of V (measured in cycles) averaged between the distinct values of all members in the group. In the Repel group this value constituted the length it moved backwards once it neared a collision. Every 10 cycles of V constituted one second of time the robot would repel once it approached a teammate. Similarly in the Timeout group, every 10 cycles of V translated into one second added to its inactivity threshold. Notice how the Timeout group originally used a very high timeout value and proceeded to lower this threshold. As the group size grew, it became necessary to react to inactivity more aggressively. Similarly, the Repel group needed to increase the length of time it moved away from another robot as its group size grows. The use of more aggressive coordination methods was justified as collisions in the domain became more frequent.

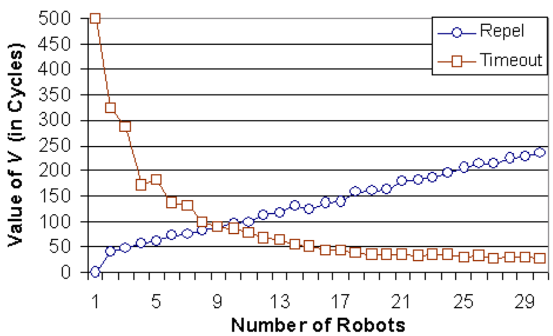

Figure 4. Dynamic Coordination Threshold Values (V)

We found that our dynamic coordination group produced statistically significant higher levels of performance than that of even the best static method we studied. Once again for statistical significance we ran our dynamic Repel and Timeout groups for 50 trials over a range of 1 - 30 robots. The dynamic Repel team on average collected 24.5 pucks, better than the average of 23.0 pucks the Repel200 group produced. Nearly half of the time (13 out of 30 instances) this group even collected more pucks than the best of the 5 static options we tested. Figure 5 graphically depicts the success of this group.

In order to evaluate the statistical significance of these results, we conducted the two tailed paired t-test on our data. We first compared the averaged productivity values of our adaptive Repel group to all of the non-adaptive methods over the range of 30 robots. All scores were far below the needed 0.05 for significance with the highest p-value for the Null hypothesis being only 0.00013 (between our dynamic group and the Repel100 group). This strongly supports our hypothesis that our dynamic method improved results from the dynamic methods in a statistically significant fashion.

Figure 6 demonstrates the success of the dynamic Repel group in minimizing interference. The X-axis in this graph represents the

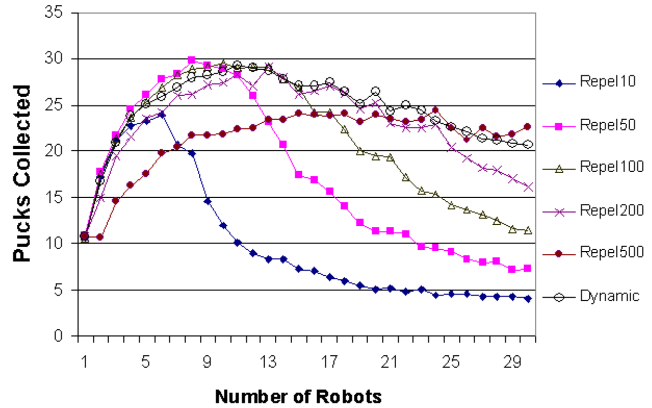

Figure 5. Adapting Productivity in Repel Group

group size, and the Y-axis corresponds to the cycles of interference that groups registered. The dynamic group consistently registered the lowest level of interference from among the static groups which it was based upon. This substantiates the argument that high productivity and low interference are correlated.

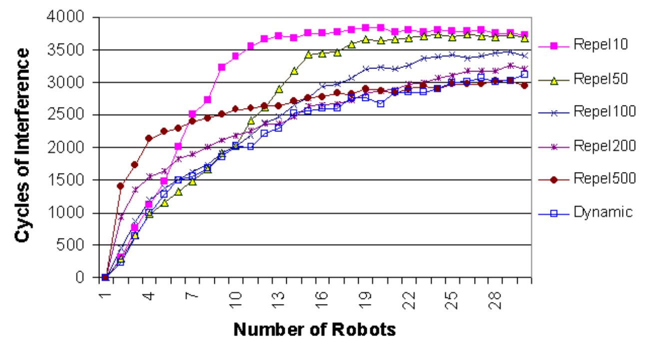

Figure 6. Interference Levels in Dynamic and Static Repel Groups

The dynamic Timeout group also performed better than the static methods. As figure 7 shows, the dynamic coordination method was able to achieve the best performance, or nearly the best, from among the various static amounts. On average, this group collected 19.2 pucks, more than the 17.6 average pucks the best static group (Timeout50) we studied. For over half of the group sizes (18 out of 30) the dynamic group even outperformed the best static method.

The t-test scores comparing our adaptive Timeout group with the static methods also confirm the statistical significance of our findings. All scores were well below the needed 0.05 needed for significance with the highest p-value of 0.0014 found between our adaptive Timeout group and the Timeout50 method (the group that performed the best of the static Timeout methods). A very high p-value of 0.98 also exists between our dynamic group and the maximum productivity value taken from among all the static Timeout methods over each of the 30 group sizes. This statistically confirms that this adaptive method very closely approximates the best performance among all static methods. We conclude that using dynamic methods is effective in achieving higher productivity by adapting their coordination method to the needs of their environment.

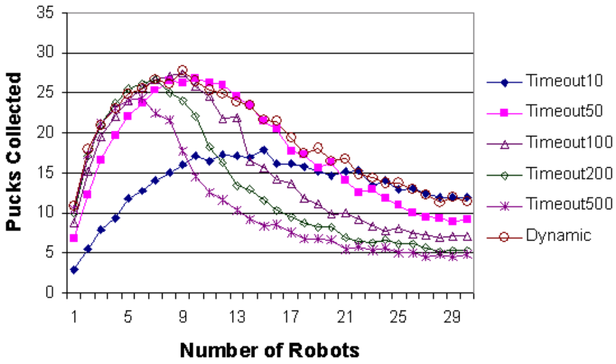

Figure 7. Adapting Productivity in Timeout Group

5 Related Work

The robots used in our algorithms work with no communication and are not preprogrammed to operate only within certain zones. Many coordination methods share this similarity such as those developed by Arkin and Balch [1], Vaughan et al. [11], and Ostergaard et al. [8]. Other algorithms such as those within the work of Fontan and Matarić [10] and the territorial arbitration scheme in Goldberg and Matarić [3] prevent collisions by limiting robots to specific areas within foraging domains. Jäger and Nebel [5] present an algorithm that can dynamically create these areas in a vacuuming domain, but require the robots to communicate locally. Another group of algorithms preassign values so that certain robots inherently have a greater priority to resources than others. Vaughan et al.'s fixed hierarchy system [11] and Goldberg and Matarić's caste arbitration algorithm [3] implement variations of this idea on foraging robots. We leave the study of adaptation within these other classes of coordination, including coordination methods for heterogeneous groups, for future work.

The concept of attempting to have robots learn from their environment has been extensively studied. Previous work by [6] found reinforcement learning based on Q Learning to be quite effective for a box pushing robot. While they concede that behavior based learning is especially slow to converge within robotic domains, using a behavior based approach did speed the process. Matarić [7] studied various reinforcement learning approaches on foraging robots and stressed that the time to learn can be quite long if certain events (such as collisions in our domain) occur sporadically. However, the time to learn certain tasks could be diminished by using behaviors that use implicit knowledge of their domain. Both of these approaches highlight the difficulty in exclusively using traditional learning methods within robotic domains.

Our approach attempts to quickly fit a coordination method to its environment by using heuristics. The advantage of this approach over reinforcement learning lies in its speed and simplicity. Factors such as collisions in a robotic domain are often quite in flux and robots need to be able to react quickly to changing conditions. In tasks such as interference resolution where robots must react quickly and near-optimal results are sufficient, our method is likely to be of an advantage. Indeed our dynamic coordination method did successfully adapt to deal with changes in their environment. One main disadvantage of our approach lies in the manual initial work in setting the weights within our heuristic. Before our adaptive coordination methods could begin, work was needed to set the weights in our methods. We also cannot guarantee any converge on an optimal solution as reinforcement methods do through maximization of reward. We leave for future work how the process of initially setting these weights can be simplified. One possible method would be to pass information based on previous trials and use a combination of classical reinforcement learning in addition to our heuristic based approach. The use of learning would likely also lead towards an eventual convergence of optimal weight values.

6 Conclusion and Future Work

In this paper we presented a method for dynamically adjusting coordination methods based on the conditions robots sense in its operating domain. Our use of interference metrics allowed us to create these powerful adaptive heuristics. We studied two basic coordination methods, Repel and Timeout, and empirically demonstrated how group performance within our foraging robots was significantly improved by using our algorithm. The spatial constrictions which cause interference in the foraging domain are common to many areas such as waste cleanup, area coverage in vacuuming, search and rescue domains, and planning collision-free trajectories in restricted spaces. We believe our approach of dynamic coordination methods will benefit designers of robotic groups in these domains as well.

For future work, several directions are possible. This paper uses interference metrics to achieve better productivity within one type of coordination method. The best productivity level this type of adaptation can reach is to attain productivity levels equivalent to that of the best theoretical static group. Further work is required to use interference metrics to dynamically select between different coordination methods. Such a system may achieve better results than those where adaptation is used only within one coordination method. Furthermore, the addition of communication may speed the adaption process within coordination groups. Additional research is required to evaluate the impact communication has on interference and group productivity.

REFERENCES

[1] R.C. Arkin and T. Balch, 'Cooperative multiagent robotic systems', in *Artificial Intelligence and Mobile Robots*. MIT Press, (1998).

[2] T. Balch. Teambots, www.teambots.org.

[3] D. Goldberg and M. Matarić, 'Interference as a tool for designing and evaluating multi-robot controllers', in *AAAI/IAAI*, pp. 637–642, (1997).

[4] D. Goldberg and M. Matarić, 'Design and evaluation of robust behavior-based controllers for distributed multi-robot collection tasks', in *Robot Teams: From Diversity to Polymorphism*, (2001).

[5] M. Jager and B. Nebel, 'Dynamic decentralized area partitioning for cooperating cleaning robots.', in *ICRA 2002*, pp. 3577–3582.

[6] Sridhar Mahadevan and Jonathan Connell, 'Automatic programming of behavior-based robots using reinforcement learning', in *National Conference on Artificial Intelligence*, pp. 768–773, (1991).

[7] M. Mataric, 'Reinforcement learning in the multi-robot domain', in *Autonomous Robots*, pp. 73–83, (1997).

[8] E. Ostergaard, G. Sukhatme, and M. Matarić, 'Emergent bucket brigading', in *Autonomous Agents*, (2001).

[9] A. Rosenfeld, G. Kaminka, and S. Kraus. A study of marginal performance properties of robotic groups, Poster to appear in AAMAS '04.

[10] M. Schneider-Fontan and M. Matarić, 'A study of territoriality: The role of critical mass in adaptive task division', in *From Animals to Animats IV*. MIT Press, (1996).

[11] R.T. Vaughan, K. Støy, G.S. Sukhatme, and M.J. Matarić, 'Go ahead, make my day: robot conflict resolution by aggressive competition', in *Proceedings of the 6th int. conf. on the Simulation of Adaptive Behavior*, (2000).