A Study of Mechanisms for Improving Robotic Group Performance*

Avi Rosenfeld[1,2], Gal A Kaminka[2], Sarit Kraus[2] and Onn Shehory[2]
[1]Department of Industrial Engineering
Jerusalem College of Technology, Jerusalem, Israel 91160
[2]Department of Computer Science
Bar-Ilan University, Ramat-Gan, Israel 92500
Email: {rosenfa, galk, sarit}@cs.biu.ac.il, onn@il.ibm.com

September 25, 2007

Abstract

Many collaborative multi-robot application domains have limited areas of operation that cause spatial conflicts between robotic teammates. These spatial conflicts can cause the team's productivity to drop with the addition of robots. This phenomenon is impacted by the coordination methods used by the team-members, as different coordination methods yield radically different productivity results. However, selecting the best coordination method to be used by teammates is a formidable task. This paper presents techniques for creating adaptive coordination methods to address this challenge. We first present a combined coordination cost measure, CCC, to quantify the cost of group interactions. Our measure is useful for facilitating comparison between coordination methods, even when multiple cost factors are considered. We consistently find that as CCC values grow, group productivity falls. Using the CCC, we 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. We present two adaptation heuristics that are completely distributed and require no communication between robots. Using these heuristics, robots independently estimate their combined coordination cost (CCC), adjust their coordination methods to minimize it, and increase group productivity. We use simulated robots to perform thousands of experiment trials to demonstrate the efficacy of our approach. We show that using adaptive coordination methods create a statistically significant improvement in productivity over static methods, regardless of the group size.

Keywords: Multiagent systems, Adaptive Coordination, Localized Decisions

*This material is based upon work supported in part by the NSF under grant IIS0705587 and by the BSF under grant #2002401. Sarit Kraus is also affiliated with UMIACS.

1 Introduction

Groups of robots are used to enhance performance in many tasks [8, 11, 16, 28]. However, the physical environment where such groups operate often pose a challenge for the robots to properly coordinate their activities. 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 [2, 11, 16, 28]. Thus while adding robots can potentially improve group performance, collisions are likely to become more frequent. To address these issues, a variety of collision avoidance and resolution techniques have been previously presented [2, 10, 11, 26, 29, 32]. However, no one method is best in all domain and group size settings.

Matching the best coordination method for a given robotic team and its operating domain is a formidable task. To date, several coordination frameworks have been suggested for reasoning about teamwork and coordination [14, 21, 31]. One possible approach is to use decision theoretic models such as Markov Decision Processes (MDP) [27] within any of these formalized frameworks. This could potentially allow robots to choose the optimal coordination method as needed during task completion.

However, while each of these frameworks has been shown to be effective under certain conditions, in many real-world applications the problem of making the optimal coordination decision is computationally intractable [27]. The inherent complexity in using these approaches demonstrates the necessity of creating novel algorithms and heuristics to effectively deal with real-world issues in a tractable fashion.

Our approach is to investigate a combined coordination cost measure, CCC, that quantifies the production resources spent on coordination conflicts. We present this multi-attribute cost measure to quantify resources such as time and fuel each group member spends in coordination behaviors during task execution. The CCC measure facilitates comparison between different group methods. We found a high negative correlation between this measure and group productivity, allowing us to understand why certain groups were more effective than others.

This negative correlation between performance and CCC facilitates development of adaptive coordination methods. The key idea is that if robots dynamically reduce their CCC, group productivity will be improved. To demonstrate this, we create robotic groups which dynamically adapt their coordination techniques based on each robot's CCC estimate. Robotic agents calculate CCC estimates autonomously by noting the frequency of events in which collisions are possible (and may or may not take place). This is done in a distributed fashion and without any feedback from group members—no communication is necessary.

We present two adaptive coordination methods suitable for homogeneous robots based on the CCC estimates. The first method of adaptation works by tweaking the parameters of a given coordination method, adapting them to the frequency of possible collisions. The second approach proceeds to dynamically self-select between a range of mutually exclusive coordination methods. In order to quickly adapt to a changing environment, we use weight-based heuristics by which every robot in the group is capable of quickly modifying its coordination method to match its estimated CCC.

We use a well-tested multi-robot simulator, Teambots [3, 4] to simulate groups of up to 30 robots engaged in both search and foraging tasks. We perform thousands of experiment trials, to demonstrate the efficacy of this approach, with various team sizes and compositions.

We find that these adaptive coordination approaches result in a statistically significant increase in group productivity in the domains we study, even when faced with dynamically changing conditions. During task execution, different robots in the group engage in varied coordination resolution behaviors. In fact, we find that the best form of coordination changes over the course of time, or as the task is being completed. Thus, various forms of coordination are likely to be needed at different times during task execution.

While we cannot guarantee the optimality of these heuristic approaches, the experiments demonstrate that this approach is effective in achieving a statistically significant improvement in productivity without a prolonged training period. We believe that this is likely to be needed in many robotic domains, as environment dynamics and noise make traditional learning approaches difficult to implement.

2 Productivity Increases in Robotic Groups

This paper focuses on understanding the interplay between group coordination and productivity in robot groups. A closely related topic, of the scalability of labor, has been extensively studied in the field of economics. According to the *Law of Marginal—or Diminishing— Returns*, as additional production resources are added, the additional productivity yielded as a result decreases [7]. The highest returns on production resources are from the first beginning of the production cycle. They then diminish with additional production expenditure, until a point where it typically becomes economically impractical to add more production resources; the cost of additional production resources outweighs the productivity they add.

To date, there have been limited—and often conflicting—studies into how robotic team productivity scales with the addition of robots. Rybski et al. [28] demonstrated that groups of identical robots can exhibit marginal returns, with productivity curves resembling logarithmic functions. The first several robots in the groups they studied added the most productivity per robot, and each robot added successively less. However, they did not study group sizes larger than five robots. In contrast, work by Fontan and Matarić [29] found robotic groups operating within a robotic foraging domain contained a certain group size, a point they called "critical mass", after which the net productivity of the group dropped. Similarly, Vaughan et al. [32] also reported that adding robots decreased performance after a certain group size. The motivation for this work lies in understanding when coordination methods would be successful in consistently realizing marginal gains, and when one could expect to encounter a "critical mass" in group size.

2.1 Group Differences in Performance

Our study begins with a simulated foraging domain, in which we investigate how robot productivity is affected as group size is scaled up. Foraging is formally defined as locating target items from a search region *S*, and delivering them to a goal region *G* [12]. The foraging domain is characterized by a limited area of operation where spatial conflicts between group members are likely to arise [10, 11, 12, 26, 28, 29, 32]. Many robotic tasks such as waste cleanup, search and rescue, planetary exploration, and area coverage share this trait.

We used a well tested robotic simulator, Teambots [3, 4], to collect data. We preferred using a simulator over performing experiments with real robots as it allowed us the ability to perform thousands of trials of various team sizes and compositions. The sheer volume of this data allowed us to make statistical conclusions that would be hard to duplicate with manually setting up trials of physical robots. However, it is important to note that code created in the Teambots simulator has been shown to directly port to Nomad N150 robots; all behaviors and features found within the simulator can be equally applicable to these physical robots [3].

Using Teambots [4], we simulated a foraging environment measuring approximately 10 by 10 meters. There were a total of 40 target pucks within the field, 20 of which were stationary within the search area, and 20 which moved randomly. Each trial measured how many pucks were delivered by groups of 1–30 robots using each of the coordination methods we studied, within 9 simulated minutes of activity. To overcome any dependencies on initial positions, we averaged the results of 100 trials with the robots being placed at random initial positions for each run. Thus, this experiment simulated a total of 21,000 trials (7 groups $\times$ 30 group sizes $\times$ 100 trials per size) of 9 minute intervals.

We implemented a total of 7 coordination methods based on previously developed collision resolution and avoidance algorithms, and variations thereof. All algorithms operated without prior knowledge of the domain, nor with communication. We chose to contrast coordination methods from this category to focus exclusively on issues relating to coordination resolution behaviors without needing to consider other factors.

The implementation of the *Noise* method was included in the Teambots [4] package. Balch and Arkin [2] described this method as a system of using repulsion schema any time a robot projected it was in danger of colliding. These robots then also added a noise element into their direction vectors to prevent them from becoming stuck at a local minima.

Vaughan et al. [32] described an algorithm that uses *Aggression* to resolve possible collisions by pushing its teammate(s) out of the way. They posited that possible collisions can best be resolved by having the robots compete and having only one robot gain access to the resource in question. In our implementation of this method, for every cycle a robot found itself within 2 radii of a teammate, it selected either an aggressive or timid behavior, with probability of 0.5. If the robot selected to become timid, it backed away for 100 cycles (10 simulated seconds). Otherwise it proceeded forward, executing the aggressive behavior. As robots chose to continue being “aggressive” or to become “timid” every cycle, the probability that two robots would collide in this implementation was near zero.

Similar to the Aggression group, the *Repel* group backtracked for 500 cycles (50 seconds) but mutually repelled using a direction of 180 degrees away from the closest robot. The *TimeRand* group contained no repulsion vector to prevent collisions. However, when robots sensed that they did not significantly move for 100 cycles (10 seconds), they proceeded to move with a random walk for 150 cycles (15 seconds). The *TimeRepel* also only reacted after the fact to collisions. Once these robots did not move for 150 cycles (15 seconds), they then moved backwards for 50 cycles (5 seconds).

Finally, we created two groups that lacked any coordination mechanism. The *Gothru* group was allowed to ignore all obstacles, and as such spent no time engaged in coordination behaviors. This “robot” could only exist in simulation as it simply passes through obstacles and other robots. This group represented a theoretical group performance without any productivity lost to collisions. At the other extreme, the *Stuck* group also contained no coordination behaviors but simulated a real robot. As such, these robots were likely to become stuck and lose all productivity when another robot blocked its path.

Figure 1 graphically represents the foraging results from these coordination methods. The X-axis depicts the various group sizes ranging from 1 to 30 robots. The Y-axis depicts the corresponding average number of pucks the group collected averaged over 100 trials.

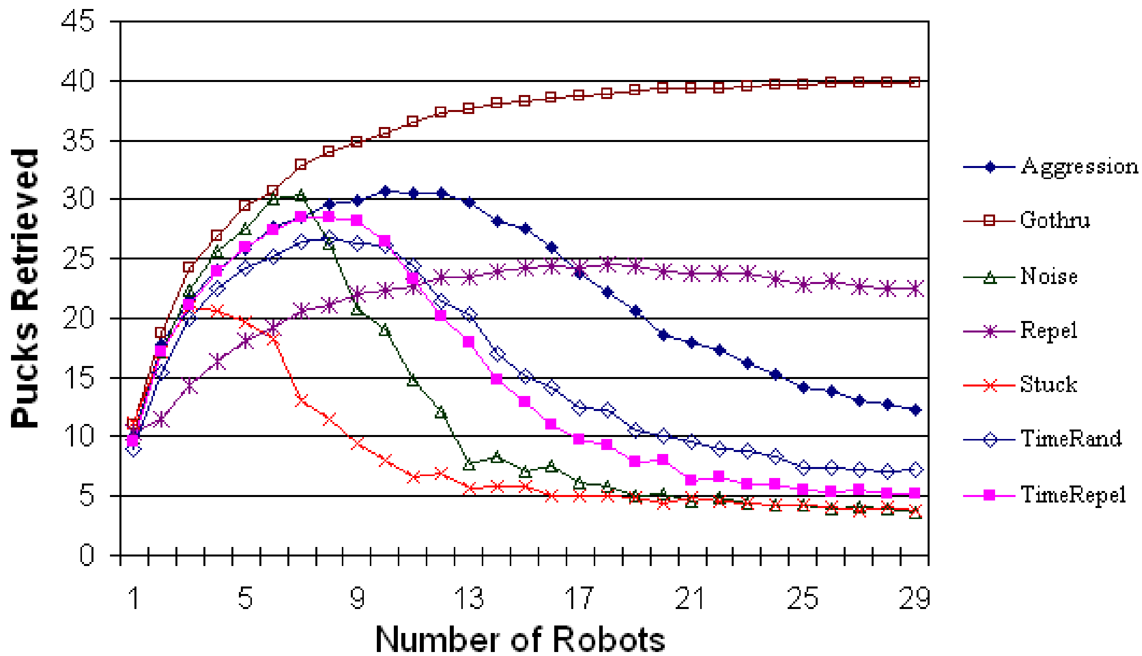

Figure 1: Motivating results comparing seven foraging groups. Each data-point represents the average pucks returned to the domain's home-base using that coordination method (Y-Axis) given that group size (X-axis).

According to economic theory, diminishing marginal returns are achieved when one or more production resources are held in fixed supply, while the quantity of homogeneous labor increases. In the foraging domain, the fixed number of pucks and limiting domain area acted as limiting factors of production. Consequently, one would expect to find production graphs consistent with economic marginal returns. However, only the theoretical Gothru group consistently demonstrated this quality over the full range of group sizes. All other groups contained a critical point where maximal productivity was reached. After the group size exceeded this point, productivity often dropped precipitously. For example, the Aggression group reached a maximum of 30.84 pucks collected in groups of 13 robots. Additionally, the coordination behaviors had a profound impact on each productivity level. For example, when examining foraging groups of 10 robots, the Aggression method averaged over 30 pucks collected, the Noise group averaged approximately 20 pucks, and the Stuck group on average collected fewer than 8 pucks.

2.2 The Impact of Coordination on Robot Density

We propose that differences between coordination methods in spatially constrained domains can be explained based on robot density. As one adds robots into a domain, the density of robots, on average, should rise. Within spatially constrained domains, this can lead to certain area(s) having a bottleneck condition where robots cannot effectively complete their task, resulting in loss of productivity. However, having too low a density results in agents not reaching goal areas within the domain and thus not properly completing their task. As different coordination methods impact the group's density, it is critical that we properly match the coordination method to the domain conditions to achieve the best productivity for the group.

We can model robot density as follows: Let us pick a point p within a spatially constrained domain where a group of N robots must pass to complete their task. Given a radius r around this point, we focus on an area $A(r)$ surrounding p. During task completion, robots constantly move in and out of $A(r)$ with a certain heading α. At any given time t, there are k robots within any given area $A(r)$, where $k \leq N$. We denote the density, $\phi(r)$ as the total area of these k robots divided by the total area $A(r)$. The value of $\phi(r)$ will impact the group's performance. For example, $\phi(r) = 1$ indicates $A(r)$ contains no free space, and all robots mutually block. In these instances all productivity of the group will be lost until the area is cleared, and the density lowered. Conversely, assuming $\phi(r) = 0$, no robots are within the area. Assuming this value remains zero, no robots will complete their task, and the group's productivity will be zero until robots are allowed into the constrained area and $\phi(r)$ rises.

Figure 2 illustrates an example taken from the Teambots simulator with $k = 3$ robots within a radius $r = 1.5$ (meters). Note that we study groups of homogeneous robots where each robot has a radius of approximately 0.25 meters. We denote the area of each robot as A', where $A' = 0.25^2\pi$ or 0.20. Thus, the density $\phi(1.5)$ as illustrated here would be $(kA')/A(r)$ or $(3 \times 0.20)/7.1$ or 0.08.

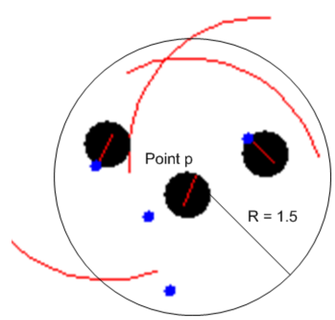

Figure 2: Three robots within $A(r)$ where $r = 1.5$. Picture is taken from the Teambots simulator and is drawn to scale.

Every coordination method impacts the way in which robots prevent and resolve collisions, thus impacting $\phi(r)$. In general, coordination mechanisms that involve collision prevention behaviors well before robots collide will result in lower densities than methods that only trigger these behaviors once robots are closer. Similarly, methods that more aggressively space robots after collisions will result in lower densities than less aggressive methods. For example, a group whose coordination method requires robots to move away for a distance of 5 meters after a collision will have a lower density than a method that only requires robots to move away 1 meter.

We claim that as robots are added or taken away from a domain, the best coordination method will change. When the group size (N) is small, the number of robots (k) coming within the constrained area is also likely to be small. In these cases, coordination methods should allow robots to complete their task uninhibited, and not further reduce $\phi(r)$. As N grows, k will naturally grow as well, and naive methods will result in too high values for $\phi(r)$. In these cases, methods that more robustly disperse the robots will be needed.

Determining the exact optimal value for $\phi(r)$ for a given domain and set of robots is a complex challenge, as many factors must be accounted for. First, we must model the speed of robots with regard to various domain conditions and behaviors. For example, the robots we studied slowed down to pick up objects, deviating from their maximal speed. Such phenomena must be exactly accounted for. Second, we must model the robots' exact positions and headings throughout task completion. In general, every robot heading towards p will have a velocity vector V_i based on its heading α from its initial position P_i towards its final destination point p. For an exact model, every coordination method's response to different positions and headings must be precisely calculated. Finally, a simplified model assumes robots mutually block only in head on collisions. In fact, even indirect collisions also block robots, and thus the "collision area" of a robot needs to be modeled as $P_{i_{x+\epsilon}}$ and $P_{i_{y+\epsilon}}$ instead of the location P_i the robot is currently situated in. Given the complexity of modeling these different factors, we leave calculation of an optimal $\phi(r)$ for future work.

Nevertheless, we can generally demonstrate two important characteristics based on our model: (i) Differences in density exist between coordination methods; (ii) Given a certain radius r, some density value $\phi(r)$ results in the best group performance, regardless of the group size (N) operating within the domain, or the specifics of the coordination method used. The latter is a very important observation, as it may provide guidelines for matching coordination methods to specific domains based on their derived density. To demonstrate these claims, we logged the value of ϕ as a function of various distances r from the home-base (point p) within the foraging domain, and various group sizes. Specifically, we studied how values of ϕ corresponded between coordination methods taken at distances of $r =$ 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, and 5.0. As was the case in Figure 1, we averaged every value from 100 simulated runs.

First, we compared the Aggression, Noise, Repel and Stuck coordination methods defined in the previous section. Recall from Figure 1 that the Aggression method performed best in groups of 10 robots, and Repel performed best in groups of 30. In Figure 3 we plot the density functions for $N = 10$ (the graph on the left) and 30 robots (the graph on the right). Note, that differences in coordination methods' densities were most pronounced when studying smaller distances for r around point p. As one would expect, as $A(r)$ encompasses progressively larger portions of the entire domain area, the number of robots within this area (k) eventually equals N and no differences should be expected between coordination methods. Consequently, we only focus on density differences within small values for r. The Aggression method, which performed well in medium sized groups, did not successfully resolve conflicts in larger groups. This is reflected by an increase in density when moving from 10 robots to 30 robots. Conversely, Repel, which was effective in larger groups, exhibits a (too) low density in small and medium sized groups, reflecting a relatively lower productivity.

Second, when carefully inspecting the density levels for which the coordination methods have arrived at maximal productivity, it appears that some optimal density level exists. Specifically, one can observe that the density graphs for Aggression in groups of 10 and for Repel in groups of 30, are nearly identical (recall that these graphs correspond to methods performing optimally for a given group size). Inspecting the density values arrived at by these methods shows that they are almost identical $\phi(0.5) = 0.18$, $\phi(1.0) = 0.15$, etc., from which we can conclude that optimal performance corresponds to a common density pattern. As we show below, other observations support this conclusion.

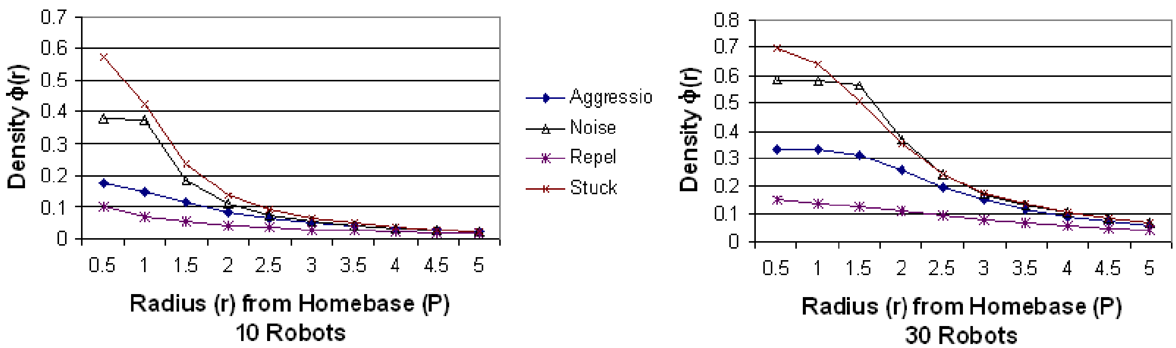

Figure 3: Robotic density for four coordination methods for groups of 10 robots (on left) and 30 robots (on right)

Similarly, one may question if the parameters within the coordination methods provide optimal densities. The Repel method we defined in the previous section backtracks for 50 seconds after a detected collision. We posit that different backtracking amounts would create different densities, each most appropriate for different domain conditions. To support this claim, we created variations of the Repel behavior where repel values of 5, 10, 20, and 50 seconds (Repel50, Repel10, Repel20, and Repel500 respectively) were used. Figure 4 displays these density functions for group sizes N =10, 20 and 30. Note that the density graphs of Repel50 in groups of 10, Repel200 in groups of 20, and Repel500 in groups of 30 are quite similar, and again reflect values similar to those seen in Figure 3. In fact, as we will see within the experiments sections (see Figure 10) that these Repel values yielded the highest productivity in these group sizes.

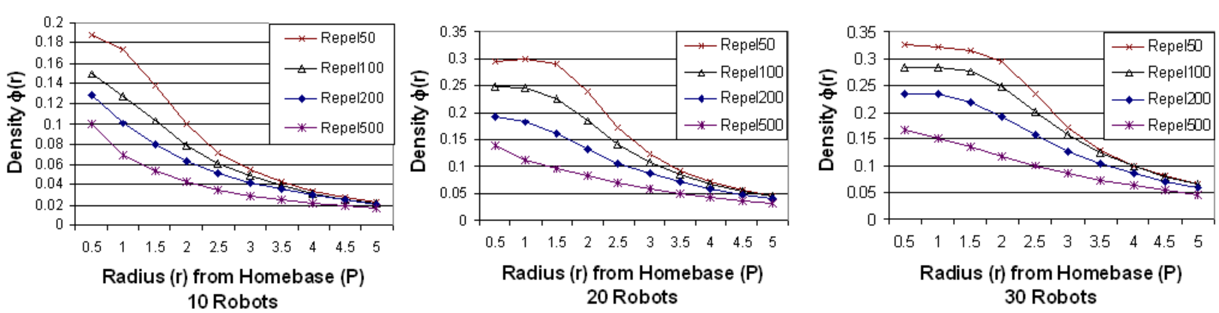

Figure 4: Comparing robotic density for coordination methods Repel50, Repel100, Repel200, and Repel500 in Groups of 10 (left), 20 (middle) and 30 (right)

We believe that the model could theoretically be used to calculate an optimal density for a given domain. A group designer could then compare the coordination methods at her disposal, and select the one closest to this optimal density. Furthermore, this model may also give us insight into predicting what the productivity of a group should be, and the amount a specific coordination mechanism deviates from that theoretical optimal performance level. This information could then be used to create improved coordination methods. For example, if one would know the density needed to achieve optimal performance, one could adjust the repel values within this coordination method to ensure that this condition is met.

However, this paper's assumption is that the number of variables involved with creating this precise model, and their associated states, makes determination of the optimal density impractical, for this and most real-world settings. Instead, we focus on developing a CCC measure that is significantly easier to calculate and can be autonomously measured by each robot. This measure requires no prior knowledge of the specifics of the coordination methods being used, or a-priori knowledge of domain parameters. Nonetheless, as the next section demonstrates, this measure is still effective in modeling differences in resources spent on resolving coordination conflicts. Furthermore, as Sections 4 and 5 demonstrate, this measure can also be used to create adaptive methods that quickly and effectively adapt the coordination of the team to the task.

3 Quantifying the Cost of Coordination: the CCC Measure

A mechanism is needed to measure why certain coordination mechanisms are more effective than others. In this section we present such a measure of coordination, the Combined Coordination Cost measure (CCC). We find that this measure and productivity are strongly correlated, and use this measure to explain differences in productivity between all teams. As one might expect, the more effort the group spends in coordination behaviors, its ability to complete the task at hand is diminished. We posit that in the absence of coordination conflicts such as those caused by spatial conflicts, all teams should consistently demonstrate marginal gains during scale up. We confirm this idea by easing the spatial conflicts inherent in the domains and note that all groups consistently demonstrate increasing marginal productivity returns.

3.1 Measuring Combined Coordination Costs

The CCC is defined as the sum of resources a group member expends because of its interactions with other members, in particular resolving conflicts between agents (preventing conflicts and managing their consequences). Examples of these resources may include the time, fuel, and money spent in coordination activities or in any combination of factors. Each agent expends a coordination cost $\mathcal{C}_i$, that impacts the entire group's productivity. This cost can consist of multiple factors, $\mathcal{C}_i^j$, with each one containing a relative weight of $\mathbf{P}_j$. We create a multi-attribute cost function based on the Simple Additive Weighting (SAW) method [35] often used for multi-attribute utility functions.

We describe the combined coordination cost of a specific agent as follows. Let $G = \{a_1, \ldots, a_N\}$ be a group of N agents engaged in some cooperative behavior. Let $\mathcal{C}_i = \{\mathcal{C}_i^j\}, 1 \leq j \leq t$ be the set of t coordination costs in the system derived from the actions of agent a_i. Let

$\mathbf{P}_j$ be the ratio of each factor of $\mathcal{C}_i$ in the total cost calculation, i.e., $\sum_{j=1}^{t} \mathbf{P}_j = 1$. As the total coordination cost of each agent is the simple weighed sum [35] of all of these costs, the final cost equation is:

$$\mathcal{C}_i = \sum_{j=1}^{t} \mathcal{C}_i^j \cdot \mathbf{P}_j \tag{1}$$

In contrast to Goldberg and Matarić *interference* measure [11], we model resources spent in coordination even before a specific conflict, such as a robotic collision, occurs. For example, the Aggression group's timid and aggressive behaviors to avoid collisions all constitute coordination costs by our definition. The TimeRand and TimeRepel groups have costs only after a collision is detected. The Gothru group's CCC measure was always zero because it never engages in collision resolution behaviors and thus represents idealized group performance.

According to the hypothesis, we expected to see a negative correlation between CCC measures and productivity, in two major respects. First, the degree to which a group deviates from idealized marginal gains is proportional to the average CCC level within the group. This in turn impacts the group size where the group reaches its maximal performance. Second, even before groups hit their maximum productivity point, we hypothesized that the more productive groups have lower CCC levels than their peers. This accounts for the varying productivity levels in equally sized groups.

3.2 Measuring CCC from Various Resources

In order to confirm this hypothesis, we reran the seven foraging groups and logged their average CCC levels. Figure 5 represents the result from this trial. The X-axis once again represents the group size over the 1–30 robot range, and the Y-axis represents the average time that each robot within the group spent in coordination behaviors (out of 540 seconds) over the 100 trials.

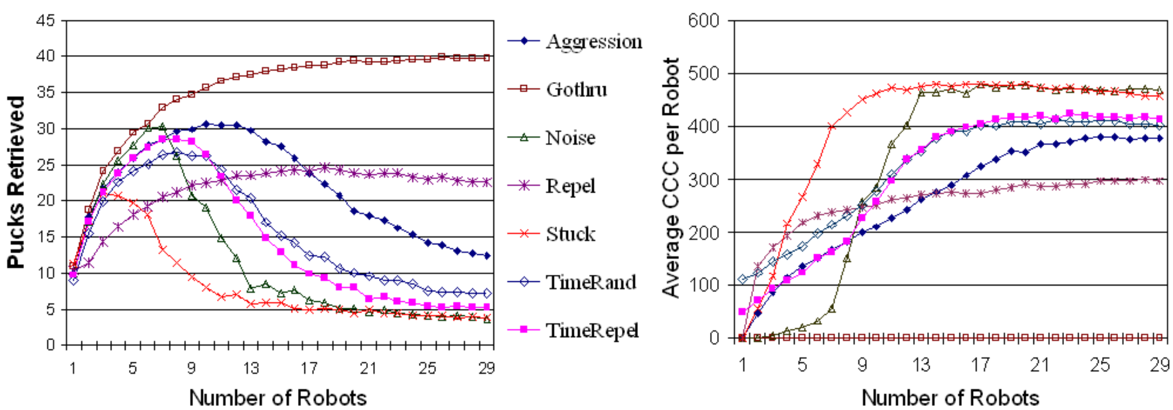

Figure 5: Comparing foraging groups' coordination costs

Overall, we found a strong negative correlation (average -0.94) between groups' performance and their CCC levels, in all groups sized 1 to 30 robots. The lower the average robots' coordination cost, the higher that groups' average productivity. The intuitive explanation is that since the task was bounded only by time, the more time spent on coordination behaviors, the less time was available for properly completing the task. Thus, groups that minimized this cost were more effective.

However, the CCC measure is also capable of taking other costs into consideration. We also implemented these same coordination methods, but used fuel instead of time as the one limiting production resource, i.e. $\mathbf{P}_1 = 1$ again. In this experiment we allocated each robot 300 units of fuel. We assumed the fuel used was proportional to the distance traveled, with a much lower amount of fuel (1 unit per 100 seconds) consumed for basic robot sensing and computation. Fuel was not transferable. Once a robot ran out of fuel, it stopped functioning and became an obstacle. Once again, we reasoned that certain methods would be more successful than others in minimizing this measurement under varying domain conditions.

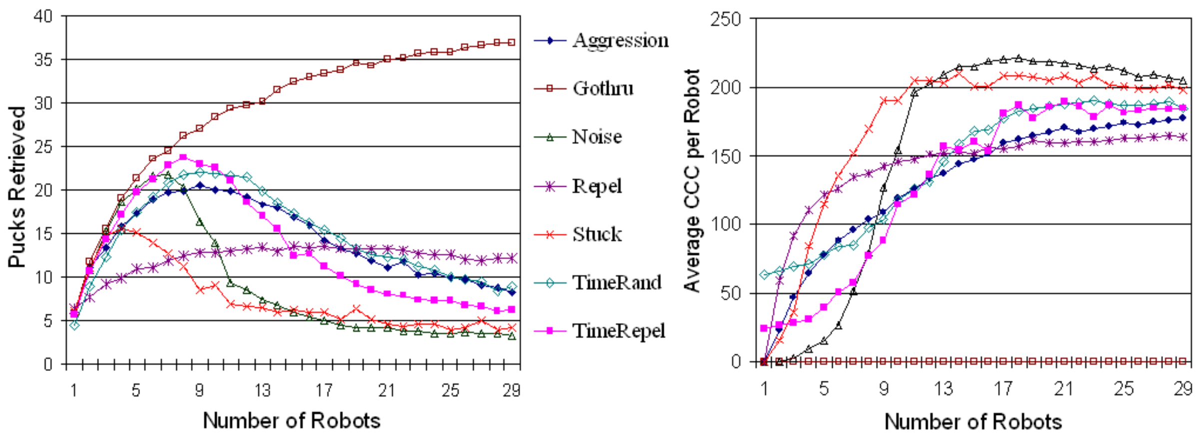

Figure 6: Comparing group productivity and coordination fuel cost measures in foraging groups

Figure 6 graphically presents the foraging productivity results over the group range of 1–30 robots when only accounting for coordination cost based on fuel. We again found a strong negative correlation (average -0.95) between the coordination cost at the agent level, and the group's productivity. Notice that the cost functions of these methods are effected by the new domain requirements (productivity bounded by fuel instead of time) and the ordering of the best coordination methods changes as a result. In these trials the Timeout based groups (TimeRand and TimeRepel) fared best in medium sized groups, while these groups never had the highest productivity in the first set of experiments.

Realistically, some combination of production resources are likely to bound an agent's productivity. As a result, we also studied cases of multi-attribute cost functions, and present the results for $\mathbf{P}_{Time} = 0.7$ and $\mathbf{P}_{Fuel} = 0.3$. While time and fuel are different resources, we created a combined cost function by viewing the cost $\mathcal{C}_i^{Time}$ as a constant amount of fuel that was detracted every second of the robot's operation, independent of its movement. This allowed us to normalize the time cost to approximately 70 percent of the total cost function and create a cost function composed of these two factors. Figure 7 presents the results for this multi-cost attribute function, with the lower Y-axis here measuring the combined cost of both factors, out of 300 total units. The multi-attribute measurement was still strongly negatively correlated (-0.94 on average) to each group size and its corresponding average productivity level.

The CCC measure is equally applicable to other domains as well. To demonstrate this claim,

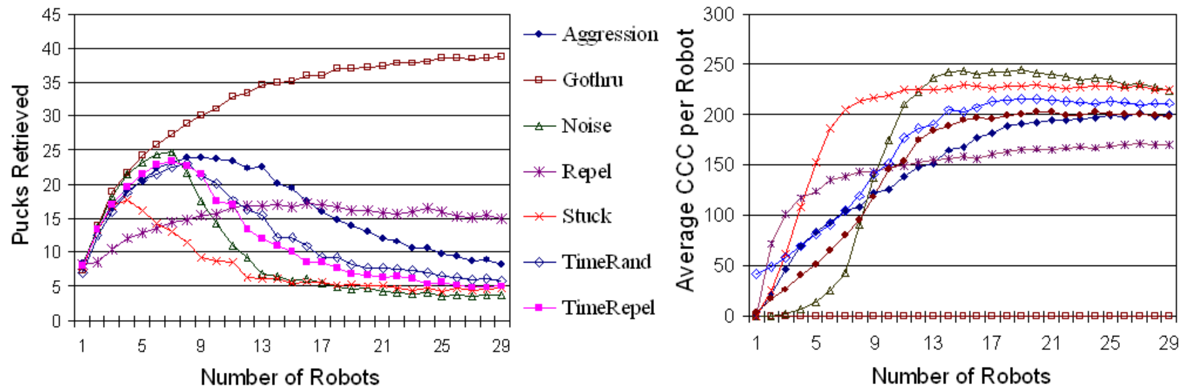

Figure 7: Comparing group productivity and multi-attribute coordination cost measures

we studied a spatially limited search domain constructed as follows. Using the Teambots [4] simulator, we created a room of approximately 3 by 3 meters with one exit 0.6 meters wide and placed groups of robots inside (for comparison purposes each robot is approximately 0.5 meters wide). We measured the time until the first robot found a target item outside the room. We ran trials of groups of six out of seven coordination methods (the Gothru method is not applicable to search tasks) in sizes from 1–23 robots (the room holds a maximum of 23 robots) and averaged the results from 50 trials. We measured the coordination cost in terms of the time and/or fuel used per robot in coordination behaviors while accomplishing this task.

We again found a high correlation between the cost measurement based on the robot's time spent in resolving conflicts, and the total time it took for the group to complete its task. We first considered the case of only the time cost being important ($\mathbf{P}_{Time} = 1$ and $\mathbf{P}_{Fuel} = 0$). We capped each experiment at 15 minutes of activity, after which we assumed the task could not be completed by that group. The results from these experiments are presented in Figure 8. In the left portion of the graph, we display the time length (in seconds) until the task was completed as the Y-axis with the X-axis showing the different group sizes. We found that most groups were able to complete their task more quickly with small groups of robots. After some group size, we again found that adding additional robots detracted from the group's overall productivity. The right graph displays the average CCC measurement based on time alone. The Y-axis depicts the number of seconds (out of 900 seconds) the robots were engaged with, on average, dealing with spatial conflicts. As the robots spent more time resolving group conflicts, more time was needed to complete the task.

We found a very high correlation (average 0.97) between the average measurement of each robot's time cost measurement, and the time to complete the task. Note that in this domain, lower search times are better, thus higher productivity is represented by lower values. Therefore, the high correlation in the search domain is positive, while it was strongly negative in the foraging domain. Still, in both cases, as the CCC measure increased, the group's productivity decreased.

The relationship between coordination costs in energy based cost measures and multi-attribute costs also applied to this new domain. In the experiments where $\mathbf{P}_{Time} = 0$ and $\mathbf{P}_{Fuel} = 1$ we allotted each search robot with 300 units of fuel. As was the case in the foraging domain, the robots used this fuel to move, but also used a smaller amount to maintain basic sensors

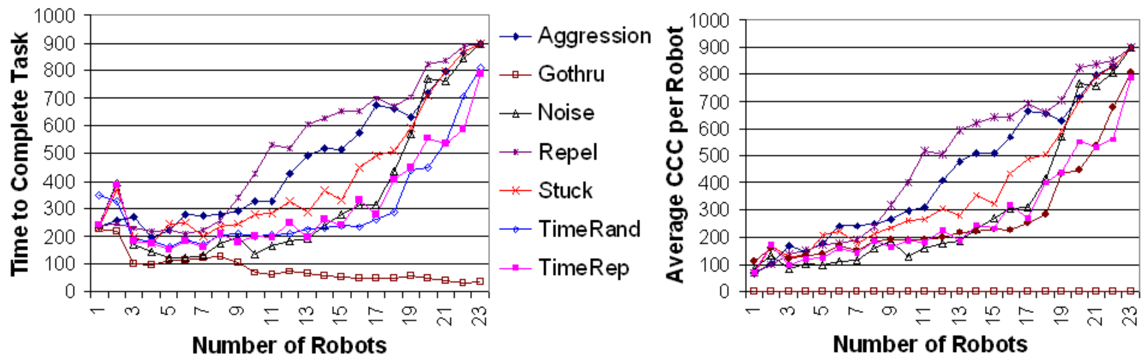

Figure 8: Comparing group productivity and coordination time cost measures in search groups

and processing capabilities. We also created experiments for $\mathbf{P}_{Time} = 0.7$, and $\mathbf{P}_{Fuel} = 0.3$ with the same standardization between time and fuel as found in the foraging domain. The fuel-only experiments had a correlation of 0.99 between the fuel used in resolving conflicts, and the average fuel used until the first robot completed the task, while the equivalent weighted experiments had a correlation of 0.98. As opposed to the foraging domain, the ordering of the most effective coordination methods was not effected by the cost functions of $\mathbf{P}_{Time} = 1$ and $\mathbf{P}_{Fuel} = 0$, or $\mathbf{P}_{Time} = 0$ and $\mathbf{P}_{Fuel} = 1$, or $\mathbf{P}_{Time} = 0.7$, and $\mathbf{P}_{Fuel} = 0.3$. In all cases, the Noise group had the best time to task completion and the lowest fuel usage to task completion in small groups. The TimeRand group had the best time to complete the task and the lowest fuel usage in larger groups. This result is intuitive, as many domains exist when fuel usage and time to task completion are correlated.

Thus, in both domains the CCC measure was successful in predicting the relative effectiveness of coordination methods. In the foraging domain the correlation between the group productivity and this measure ranged from -0.94 to -0.96. In the search domain it was even slightly higher, and ranged between 0.97 and 0.99.

3.3 Coordination Conflicts: The Trigger for Large CCC Values

According to the density model, different coordination methods effect robots' interactions within spatially constrained domains and the goal must be to properly match the best coordination method to the needs of the domain. Care must be taken not to spend too much on coordination, and thus unnecessarily lower the group's density, or too little, and thus resulting in too high a density. Robots with too low a density have spent too many resources preventing collisions. If robots have too high a density, they have not spent enough on coordination and will constantly retrigger collision resolution behaviors too quickly.

The CCC measures this expenditure of the resources spent before and after coordination conflicts. It is for this reason that the CCC can effectively measure (after the fact) which method in total spent the least on coordination, and thus achieved the best density and highest productivity.

However, the goal is also to develop mechanisms to improve group performance. In order to do so, the robots must be aware of the conflicts that trigger coordination resolution behaviors. In this section we demonstrate that the spatial conflicts inherent in the domains we studied triggered the CCC costs. Once we removed the reason for conflicts, groups consistently achieved marginal gains, and differences between coordination methods became less pronounced.

Within the foraging domain, spatial conflicts revolved around the one home-base within the operating area. We modified the foraging group requirement of returning the pucks to one centralized home base location. Instead, robots were allowed to deposit their pucks as soon as they picked them up, without returning them to any one location. We left all other environmental factors such as the number of trials, the size and shape of the field and the targets to be delivered identical. Teambots [4] was again used to simulate 21,000 trials (7 groups $\times$ 30 group sizes $\times$ 100 trials per size) of 9 minute intervals in this experiment.

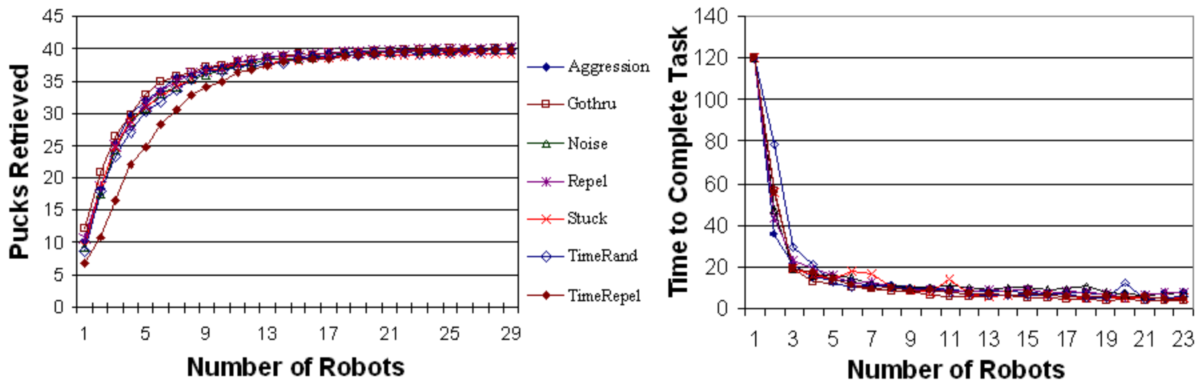

Figure 9: Modified foraging and search domains

As the left side of Figure 9 shows, all groups did indeed always achieve marginal returns in the modified foraging domain. While Gothru still performed the best, the differences between it and other groups' coordination methods were not as pronounced. Most groups had very similar coordination costs, and also productivity levels. The exception was the RepelRand group which had relatively high costs in small groups, and also lower performance. However, even this group consistently demonstrated marginal gains in productivity as the group size grew.

Within the search domain, we hypothesized that limitations in the room size and width of the exits created coordination costs during scale up. In order to ease this restriction, we doubled the size of the room to become approximately 6 by 6 meters, and widened the exit to allow free passage out of the room by more than one robot. Once again, we measured the time elapsed (in seconds) until the first robot left the room and averaged 100 trials for each point. This experiment also constituted nearly 14,000 trials (6 groups $\times$ 23 group sizes $\times$ 100 trials) of varying lengths. The right side of Figure 9 graphically shows that the modified domain consistently realized marginal increases in faster search times with respect to group size. Once again, cost levels were also negligible in the new domain. Thus, we concluded that achieving marginal productivity gains was always possible once competition over spatial resources was removed.

4 Improving Productivity through Coordination Metrics

In this section, we demonstrate how the CCC measure is useful for helping robots self-evaluate the effectiveness of their coordination methods online during task execution. By monitoring the triggers of coordination conflicts, robots are able to adapt their coordination methods to the needs of their environment. Robots that use such an approach demonstrate a statistically significant improvement in productivity over non-adaptive methods.

The dynamic nature of robotic environments makes the challenge of creating adaptive coordination formidable. While traditional reinforcement learning methods have been used within some robotic environments [23, 25], the number of iterations such algorithms require makes them unproductive without a significant training period. Even after robots could learn the theoretically optimal coordination method for their specific environment, events such as changes in the environment or hardware failures would likely render these policies obsolete. Furthermore, finding the optimal coordination method for a group is even a harder problem, with typically intractable complexity [27]. This is because the state-space of all possible actions, taken together with all possible interactions, is of exponential size. As such, even without considering system dynamics, finding the optimal coordination action is not always feasible.

We therefore focus on using CCC heuristically, to allow robots to dynamically select coordination algorithms during task execution. The approach requires no prior knowledge of the domain's physical dimensions, boundaries, number of obstacles, or number of other teammates. The possible state-space is limited to mapping values of CCC to the coordination methods at the group designer's disposal—a tractable problem that can be quickly addressed.

We present two adaptive coordination methods and their advantages above static methods. In the first technique we have the robots self adjust parameters within one coordination method to match the perceived environmental conditions. The second technique involves adaptation between a number of distinct and mutually exclusive, coordination methods. We found that both approaches did indeed significantly outperform the static methods we studied in both the foraging and search domains.

4.1 Adaptive Coordination Algorithms

The adaptive approaches are based on having each robot maintain an estimate of local coordination conflicts. This estimate is adjusted as collisions occur and/or are resolved and is thus sensitive to the triggers of the CCC costs. Specifically, the algorithm works as follows: Every robot autonomously measures its own estimate, V to represent the likelihood coordination conflicts are about to be encountered. We first initialize V to a base value, V_{init}. For each cycle that passes where that robot detects no impending collisions, it decreases its value of V by a certain amount, W_{down}. For each cycle where a robot senses a collision is likely, it increases its value V by a certain amount, W_{up}. This process continues autonomously for all robots within a group. Furthermore, this process does not require any communication between group members. Thus, it is conceivable, and even likely, that robots will have different values for V based on the localized conditions it is currently encountering.

The value V is pivotal for determining the coordination method to be used. When V is low, the robot has resolved all coordination conflicts and should use methods with low coordination overhead (low CCC cost) that do not further lower the group's density. This allows the robot to finish its task as quickly as possible. When conflicts are more common and V is high, more costly methods are needed to reduce the group's density. This removes a potential bottleneck condition, allowing some of the robots to complete their task within the spatially constrained area.

In the first group of adaptation methods, we translate values for V directly as a parameter of the coordination method. For example, we use this value to determine the number of cycles the Repel method uses to repel once it detects a collision is imminent or the time period chosen by the TimeRand method before engaging in collision resolution behaviors. This way, each robot can autonomously control the strength of its resolution behaviors.

In the second adaptation method the values for V are used to switch between a set of coordination techniques that have been pre-ordered based on their coordination overheads as ranging from simple to complex ones. Ranges of values for V are then mapped to these mutually exclusive methods. V_{init} corresponds to the starting point represented by the coordination method with the lowest overhead, and the values of V_{up} and V_{down} are then used to change the robot's fundamental coordination mechanism. Once the value V rises or falls below a certain threshold, that robot will change its fundamental coordination method as needed.

4.2 Quickly Setting the Weight Values

We now discuss how the weights, V_{init}, W_{up}, and W_{down} can be quickly set. It is important to stress that these weights form an approach to resolving coordination conflicts online. Our goal is not to find any one optimal coordination method as we found that dynamics within the domain require different coordination methods throughout the task completion. For example, assume one robot ceases functioning in the middle of the task, it may be required to switch coordination methods because of this event. Thus, the goal is to find a theoretical policy, π, based on the robot's estimate V that can be used to change the coordination method each agent uses in an optimal fashion.

While traditional learning methods, such as Q-learning [33] and other methods [30, 34] guarantee the ability to find an optimal policy, there are several major challenges in implementing this approach here. The first is procedural. Q-learning is based on a Markov based decision process that requires a concept of "state" that is difficult to define during task execution. As opposed to clearly defined discrete domains, there is no reward for any given cycle of activity in the robotic domains we studied. Thus, the ability to evaluate the effectiveness of any given action can only be done after a relatively long trial. This in turn leads to a second problem—namely, the amount of exploration data typically needed in Q-learning and other traditional learning methods to converge on an optimal solution. The thousands of trials that might be needed are impractical for physical robot trials [19]. For example, in the foraging domain previously mentioned, we studied 7 groups of coordination methods over group sizes of 1–30 robots. Each productivity data point was averaged from 100 trials for statistical significance, or a total of 21,000 trials. Third, even if a theoretical optimal policy might be found, dynamics within robotic domains may render these policies obsolete very quickly and a new learned policy π would need to be created. Finally, even if some form of learning could produce optimal weights for one robot's value of V_{init}, W_{up}, and W_{down}, there is no guarantee that these weights form the optimal coordination policy for the group. This is because the robots' sensors yield only a partial observable picture of their environment, and make no use of communication to attempt to complete that picture. Work by Pynadath and Tambe [27] demonstrated that finding an optimal policy in such cases is NEXP-complete.

As a result, the goal is *improved* productivity through an adaptive policy over the static methods upon which it is based, which may or may not form *the* actual optimal policy. Our approach is to facilitate autonomous adaptation based on the CCC measure. This measure can be locally estimated without communication, and can be used for quickly achieving significant productivity gains without a prolonged learning period.

Similar to work by Kohl and Stone [19], we used two different learning approaches for setting the weights: Hill Climbing and Gradient Learning. For each learning method, we used two different types of evaluation functions. In one possibility, the average productivity from the entire range of robot group sizes was considered. As the coordination adaptation methods are intended to work for any group size, when evaluating the effectiveness of π, the average productivity from the entire group range should be calculated. The downside of this approach is the number of trials required for policy evaluation. Assuming 5 or more trials are needed for each data point due to the noise common within any given trial, even evaluating a range of 30 robots requires 150 trials–a number that would be difficult to perform once, let alone multiple times to converge on an optimal value. As a result, we also used an evaluation function that analyzed a selective group sampling of each policy. According to this approach, representative group sizes are used to evaluate the new policy. In the experiments, we analyzed representative groups of small, medium and large group sizes. We selected the end points (group sizes of 2 and 30) as well as the middle group size (15 robots). We believed this would provide a reasonable estimate over the entire range with much fewer trials needed to evaluate any given policy. Variations of this idea are possible, such as randomly selecting the representative group size for evaluation from within a set group range, learning the best group sizes to evaluate, and various heuristics. We leave the development of these ideas for future work.

In both of the algorithms, we set the initial π to approximate the parameters of the static coordination that served as a basis for adaptation. Any static coordination method could be viewed as containing a π with fixed values of V_{init}, W_{up} and W_{down}. One naive way of improving on any static method is to choose random values for W_{up} and W_{down}, which should improve performance beyond this point. For example, assume one is trying to create an adaptive Repel method based on a static method that repels for 200 cycles after a projected collision. Once one sets V_{init} to 200, any naive values of W_{up} and W_{down} should represent a policy improvement from this point. In the second type of adaptation, V_{init} could similarly be set to represent the method with the highest average productivity. Again, any resulting policy changes resulting from W_{up} and W_{down} should only help. Hill Climbing and Gradient Learning algorithms were then used to further refine the weight values from this baseline.

Hill Climbing algorithms have the advantage that they are intuitive for this and similar parameterization problems [19]. In this method, random perturbations for the values of V_{init}, W_{up}, and W_{down} are evaluated. If these values represent an improvement in the group's overall productivity, judged through either of the two methods evaluation functions previously described (either average sampling over the entire range, or selective sampling), these new values are accepted for π. Otherwise, the changes are discarded. The following pseudo-code describes the approach:

Algorithm 1 Hill Climbing

$\pi \Leftarrow$ Initial Policy (as described in paper)
while not done **do**
  Create variation of π policy, π_{new}, with random perturbations in V_{init}, V_{up}, and V_{down}
  if Productivity(π_{new}) $>$ Productivity (π) **then**
    $\pi \Leftarrow \pi_{new}$
  end if
end while

The Gradient Learning implementation is built upon the Hill Climbing approach. In both cases, perturbations in values for V_{init}, W_{up}, and W_{down} are created and evaluated. However, in this approach, each change is evaluated individually. Instead of simply accepting a change as is, a function of the improvement caused by this factor is accepted. In the experiments, we used a normalized value in the change multiplied by a small constant, or

$$\Delta(|V_{New-weight} - V_{Old-weight}|)/V_{Old-weight} \times Constant \tag{2}$$

to create a normalized gradient direction. The following pseudo-code describes this algorithm:

Algorithm 2 Gradient Learning

$\pi \Leftarrow$ Initial Policy (same as in approach $\#1$)
while not done **do**
  generate small variations for each parameter in the value of +/- ε Specifically:
   Generate an ϵ change (perturbation) in parameter V_{init}
    Evaluate new V_{init} policy
   Generate an ϵ change (perturbation) in parameter V_{down}
    Evaluate new V_{down} policy
   Generate an ϵ change (perturbation) in parameter V_{up}
    Evaluate new V_{up} policy
  Create a new π policy based on gradient learning based on the combined evaluation of all three sub-policies. Specifically:
    $\pi \Leftarrow$ modified old policy with normalized gradient changes in all three parameters
end while

As the next section details, both learning approaches were effective in significantly improving productivity over non-adaptive methods.

5 Adaptation Experimental Results

In this section we present the results in applying both adaptive approaches within the foraging and search domains we studied. The first type of adaptation, parameter tweaking within one method, was effective in raising productivity levels to the highest levels of the static levels they were based on. Adaptation between methods was even more successful and often significantly exceeded the productivity levels of the static methods they were based on, especially in the foraging domain.

Section 5.1 presents the results of both of these adaptive methods in the foraging domain, and Section 5.2 discusses the respective results in the search domain. We also found that there was some flexibility in setting the weights, and near “out of the box” productivity improvements were found. As we demonstrate in Section 5.3, even suboptimal weight values were still successful in significantly improving a group’s performance. Finally, in Section 5.4, we present support for why the approach is so successful. We attribute the success to the robots’ ability to quickly and effectively change coordination approaches based on their localized conditions in the dynamic environments in which they operate.

5.1 Adaptation in Multi-Robot Foraging

The first type of adaptation uses each robot’s CCC estimate to adjust the strength within one given coordination method. In order to demonstrate the efficacy of this approach, we began by analyzing the strength of coordination behaviors within the Repel and TimeRand coordination methods previously mentioned. In the previous experiments, we chose a length of 500 cycles (50 seconds) with the Repel group to move away from a robot nearing a collision. Our TimeRand group waited 10 seconds before a robot considered itself stopped by another robot. As we described in Section 2.2, these parameter values are likely to be optimal only for certain group sizes. Once again, the optimal density, and thus the amount of resources each robot spends in these behaviors, must be properly matched to the group size and needs of the domain. 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 too high a density, and robots will become stuck within the spatially constrained domain. A similar problem exists in the TimeRand 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 inappropriately high densities, and robots will become stuck for long periods before attempting to resolve conflicts.

To demonstrate this phenomenon, we studied 5 variations of the Repel groups, choosing values of 10, 50, 100, 200, and 500 cycles as the length of time robots repelled after projected collisions. We found that the best variation of the Repel coordination method depended on the size of the group. As the group size grew, robots collided more frequently, and increasingly more aggressive coordination methods were needed to lower the group’s density. 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 pucks. However, this group only fared the best over a range of 5 robots. The left side of Figure 10 represents the productivity of these static methods.

We proceeded to create an adaptive Repel group where each robot used its CCC estimates to autonomously choose which repel value to use. The left side of Figure 10 also displays the productivity results from the Hill Climbing Repel adaptive algorithm and coordination costs $\mathbf{P}_{Time} = 1$ and $\mathbf{P}_{Fuel} = 0$. These results were taken after 5 learning iterations using the first evaluation function (taking the average productivity from 5 trials over the entire possible robot population). Similar results were obtained from learning trials of the other learning variations. Notice that the adaptive method often matches the highest productivity levels from the static

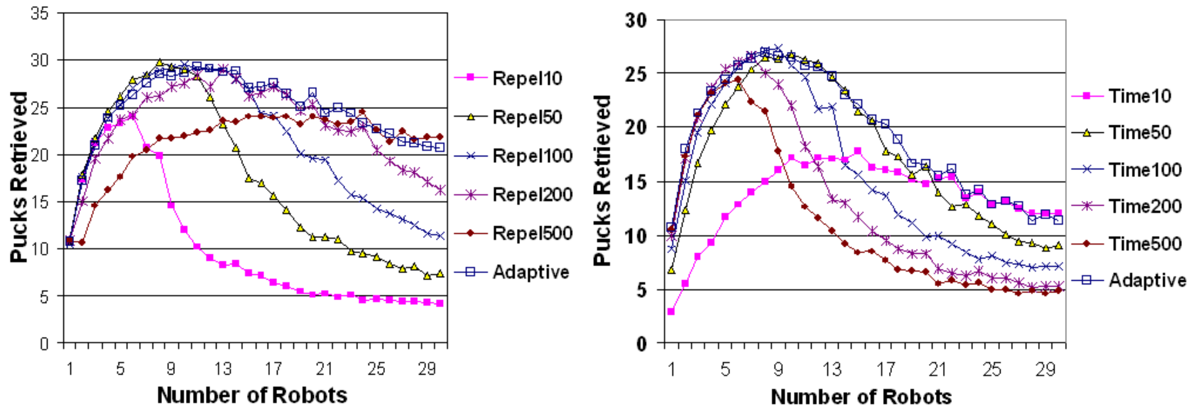

Figure 10: Productivity graphs in Repel (left) and TimeRand (right) Groups. Each data-point represents average productivity levels taken from 50 trials.

groups. For statistical significance we ran all Repel groups for 50 trials over a range of 1–30 robots.

In order to evaluate the significance of these results, we conducted a two-tailed paired t-test on the data. We first compared the averaged productivity values of the adaptive Repel group to all of the non-adaptive methods over the range of 30 robots. All scores were far below the 0.05 significance level with the highest p-value for the Null hypothesis being only 0.00013 (between the adaptive group and the Repel 100 group), strongly supporting the hypothesis that this adaptive method statistically improved results over static methods.

We also studied 5 variations of the TimeRand group, again choosing values of 10, 50, 100, 200, and 500 cycles as the length of time robots waited before engaging in resolution behaviors. The dynamic TimeRand group also performed better than the static methods. The right side of figure 10 displays the results from the adaptive Hill Climbing TimeRand algorithm for $\mathbf{P}_{Time} = 1$ and $\mathbf{P}_{Fuel} = 0$. Again, this dynamic coordination method was able to achieve the best performance, or nearly the best, from among the various static amounts. To confirm the statistical significance of these findings, we again performed the two tailed t-test. When comparing the dynamic timeout group to all static ones, we found p-scores of 0.0014 or less (p=0.0014 was found between the adaptive group and the Time50 method, which had performed the best of the static TimeRand methods). A very high statistical correlation coefficient of 0.98 also existed between the dynamic group and the maximum productivity value taken from among all the static timeout methods over each of the 30 group sizes. Thus, we concluded that this form of adaptation was effective in raising productivity in robotic groups.

The second adaptation method used the value of V to switch between 3 distinct coordination methods. In the case of $\mathbf{P}_{Time}$ = 1 and $\mathbf{P}_{Fuel}$ = 0, this involved adaptation between the Noise, Aggression, and Repel methods. The Noise group has the least costly coordination method, and was most effective in small groups up until 7 robots. At the other extreme, the Repel method fared poorly in small groups, but had the best productivity in groups larger than 17 robots. For the case $\mathbf{P}_{Time}$ = 0 and $\mathbf{P}_{Fuel}$ = 1 this type of adaptation would involve switching between the Noise, TimeRepel, and Repel methods.

In the implementation of all adaptive methods from this category, we set the values of both W_{down} and W_{up} to be one. Thus, we limited the learning problem to find the threshold values of V to switch between the basic coordination methods. We again implemented versions of gradient learning and hill climbing algorithms to converge on values for these weights. Our learning algorithms converged on threshold values of V for each of the three states at 100, 200 and 300 accordingly. Thus, if V increased by a total of 100, the robot would assume a more robust coordination method was required and would transition to use the next most robust coordination method, say from Noise to Aggression. If this method was still insufficient to resolve this instance of a projected collision, W_{up} would increase the value of V until the next threshold was reached and once again the robot would move to the next coordination method. Conversely, if that method was sufficient to resolve that incident of a projected collision, the value of W_{down} would begin to decrease the value of V and the robot could eventually move down to the next lower method of coordination.

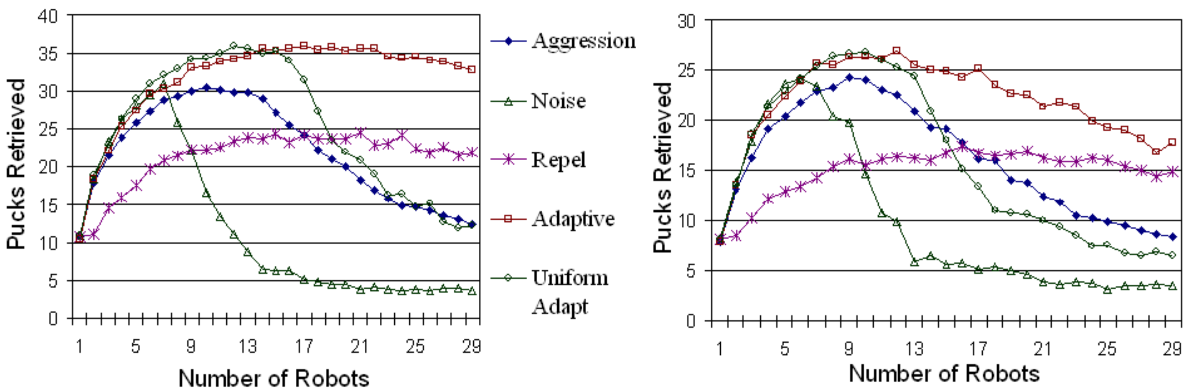

Figure 11: Adaptation between static groups for $\mathbf{P}_{Time} = 1$ and $\mathbf{P}_{Fuel} = 0$ (on left) and adaptation between static groups for $\mathbf{P}_{Time} = 0.7$ and $\mathbf{P}_{Fuel} = 0.3$ (on right)

This second adaptive coordination heuristic was even more effective than the first approach—adaptation only within one method. Figure 11 contains the results from the cases where $\mathbf{P}_{Time}$ = 1.0 and $\mathbf{P}_{Fuel}$ = 0.0 on the right side and $\mathbf{P}_{Time}$ = 0.7 and $\mathbf{P}_{Fuel}$ = 0.3 on the left. In both of these cases, we graphed the productivity levels of the 3 static methods with the highest productivity as well as that of the adaptive method (learned here through Gradient Learning). The adaptive method here yielded strong productivity gains, often in excess of more than 20 percent compared to the static methods it was based on. We again performed the two-tailed paired t-test on the data and found a p-value below 0.0001 between all basic methods with the adaptive ones, demonstrating this strong improvement.

The basic assumption of the adaptive methods we present is that all coordination acts can be done independently. Therefore, in the domains we studied, robots are able to independently choose a coordination method without impacting other team members. For example, it is possible to have one robot use the "Noise" coordination collision resolution mechanism while other robots use the "Aggression" mechanism.

However, many communication protocols exist where standardized coordination is required. To represent these situations, we also implemented an adaptive group, *Uniform Adapt* (also found in Figure 11). In this method, once one robot deemed it necessary to switch methods, it broadcasted the selected method to all other robots (a global communication network was simulated) and all robots switched in turn. In order to prevent robots from quickly switching back, all robots also set their cost estimate V to the base value of this method. Potentially, this method could force certain members to use a coordination method not appropriate for its localized conditions. We hypothesized that allowing robots to autonomously adapt to their localized conditions facilitates even further productivity gains. We further develop this idea in Section 5.3.

5.2 Adaptation in Multi-Robot Search

We believe the approach can be generalized to domains other than foraging. To support this claim, we implemented both adaptive methods within the search domain (previously studied in Section 3.2).

Our first type of adaptation involves having agents adjust the strength of their coordination methods based on the needs of the domain. Again in the search domain, we demonstrate the shortcomings within static methods, and implemented the same five TimeRand variations of 10, 50, 100, 200, and 500 cycles. We then implemented an adaptive TimeRand search method using the same weight learning algorithms to set values for V_{init}, W_{up} and W_{down} as described in the previous sections. The result was a policy π which translated V to the number of cycles used when resolving any given collision event. The results of this trial for $\mathbf{P}_{Time}$ = 1 and $\mathbf{P}_{Fuel}$ = 0 are also found in Figure 12. On average, we found a statistical improvement in performance in the adaptive group, with average search scores down nearly 10 percent in the adaptive group over the best levels among the static ones (TimeRand50).

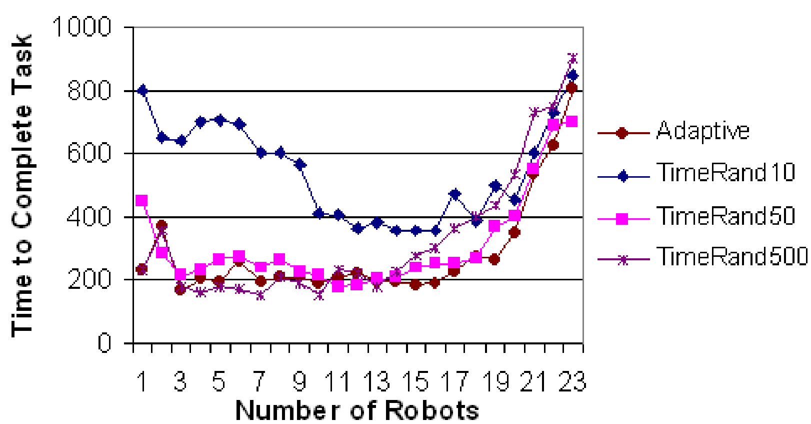

Figure 12: Search adaptation within TimeRand method using multi-attribute coordination costs

We were also successful in creating adaptive coordination methods that switched between the most effective coordination methods in this domain. Note that in this domain the Noise and TimeRand were always the best two methods, regardless if the cost comprised of $\mathbf{P}_{Time} = 1$

and $\mathbf{P}_{Fuel} = 0$, $\mathbf{P}_{Time} = 0$ and $\mathbf{P}_{Fuel} = 1$, or $\mathbf{P}_{Time} = 0.7$ and $\mathbf{P}_{Fuel} = 0.3$. We used the same methodology to create an adaptive search method with each robot using the CCC cost estimate V to effectively switch between these methods.

Figure 13 shows the Noise, TimeRand and Adaptive groups in the instance of $\mathbf{P}_{Time} = 0.7$ and $\mathbf{P}_{Fuel} = 0.3$. On the left side, we denote the productivity graphs where the X-axis represents the size of the group, and the Y-axis displays the search time, measured in seconds, until that group completed its task. On the right side, we display the CCC measures for these groups, with the Y-axis displaying the normalized CCC measure weighted between time and fuel (normalized out of 250 units). In order to establish the statistical significance of the results we performed the two-tailed paired t-test between the adaptive methods and the static ones they were based on. All results were below the 0.05 confidence level (between 0.01 and 0.04 in all three groups).

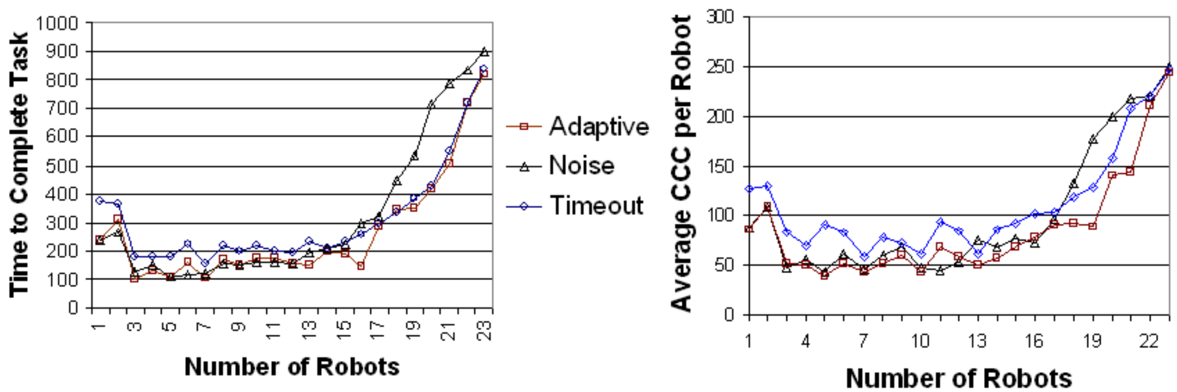

Figure 13: Search adaptation using multi-attribute coordination costs

5.3 Quickly and Significantly Improving Performance

We found that some flexibility exists in setting the weights: V_{init}, W_{up}, and W_{down}. Our results demonstrate that even results that were far from optimal were still a significant improvement from the static methods they were based on. This is because a value of V_{init} being initially 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 14 depicts the productivity of three adaptive repel foraging groups with values for V_{init} of 300, 450 and 600 and identical values for W_{up} and W_{down}. Note that while differences exist, these differences were not statically significant for most group sizes.

Figure 15 demonstrates the success of the weighted heuristic approach with only minimal learning. This graph represents three iterations in the gradient learning implementation for the adaptive foraging repel method. Our initial policy was based on Repel200, which on average had the highest average productivity over the 1–30 robot interval. In the first adaptive iteration (Gradient1) we used a value of 200 for V_{init} and naive values of 10 for both W_{up} and W_{down}. In subsequent trials (Gradient2, Gradient3), gradient learning was used to tweak these naive values. Two issues are noteworthy in this graph: First, recall that in the first evaluation method, the policy π is evaluated from averaging five trials over the entire group range. Notice the

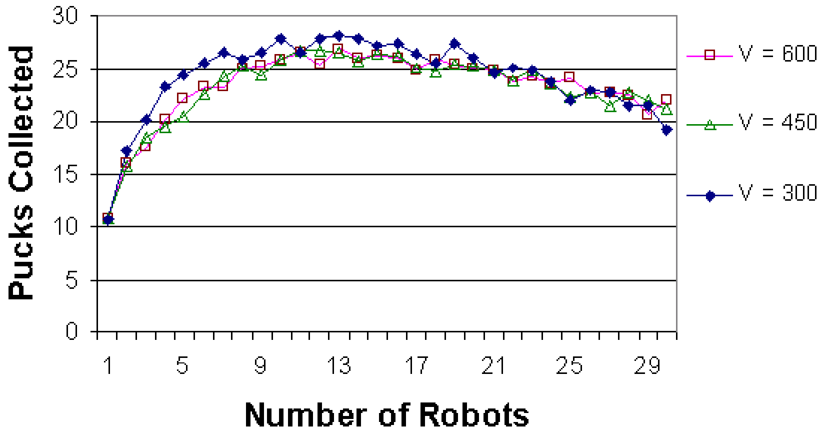

Figure 14: Three adaptive repel groups with different values for V_{init}

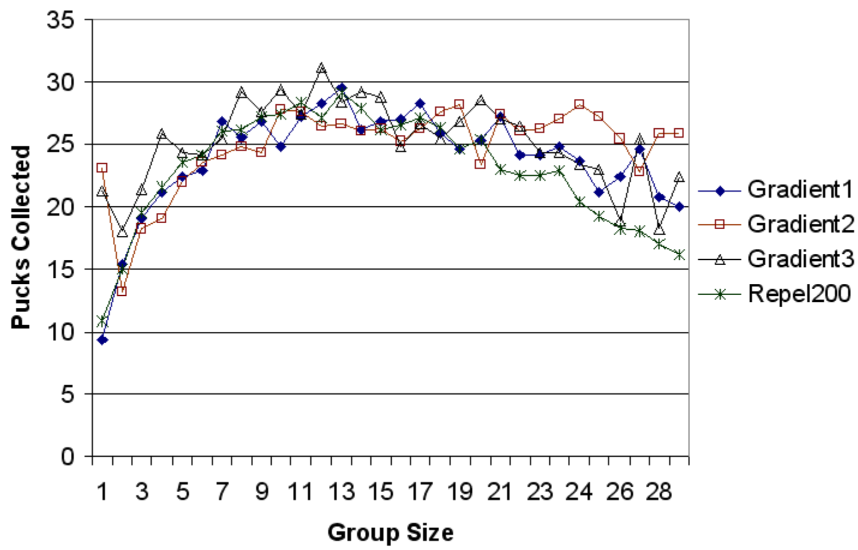

Figure 15: Three iterations of the adaptive repelling groups using gradient learning

large variance between trials. This illustrates the difficulty in learning an optimal weight value without extensive trials. Second, note that despite this difficulty, gradient learning quickly improved the weights used in the algorithms. Even within the first iteration (Gradient2) the adaptive group averaged approximately 5% improved performance, while by only the third iteration, a near local optimum was achieved with an average performance increase of 10%.

5.4 Large Productivity Gains

Not only does coordination adaptation based on CCC estimates yield productivity gains after short learning periods, but these productivity gains are often quite large—beyond any of the static methods they are based on. For example, we previously presented two types of foraging adaptive groups, *Adaptive* and *Uniform Adapt* that often significantly exceeded the productivity levels of the methods they were based on. At first glance, this result is surprising. One would assume adaptation is only capable of achieving results in line with the best levels of productivity for the methods it was based on, not significantly higher.

We claim that the root of these productivity gains is the ability of these methods to switch between coordination methods as dictated by fluctuating domain conditions. Thus, during the course of one trial, one robot may switch between its Noise, Aggression, and Repel coordination methods many times. Our goal is not to converge on any one coordination method, as that method can often change as the possibility of collisions grows or dissipates. To demonstrate this point, we studied the average CCC estimate, V, within robots in the various group sizes. Recall that this value ranged from 0–300, with values of 0–100 mapped to the Noise method, values of between 100 and 200 mapped to the Aggression method, and larger values to the Repel method. Assuming the goal was to converge on the one static method with the highest productivity, one would assume these robots would have average values of V of over 200 in groups larger than 17 (where the static Repel group fared best). However, as Figure 16 demonstrates, this was not the case, and average values for V ranged between 0 and 200 regardless of the Adaptive group's size. This result implies that even in large groups, robots did not use the most expensive method (Repel) for large portions of the trials. For example, in one foraging trial of 25 robots using the Adaptive method, the entire team spent on average 56 percent of its time in the Noise behavior, 11 percent in Aggression behavior, and 33 percent in the Repel behavior. Thus, the average value of V never rose above 200 because the group never spent a majority of its time using the most costly coordination methods.

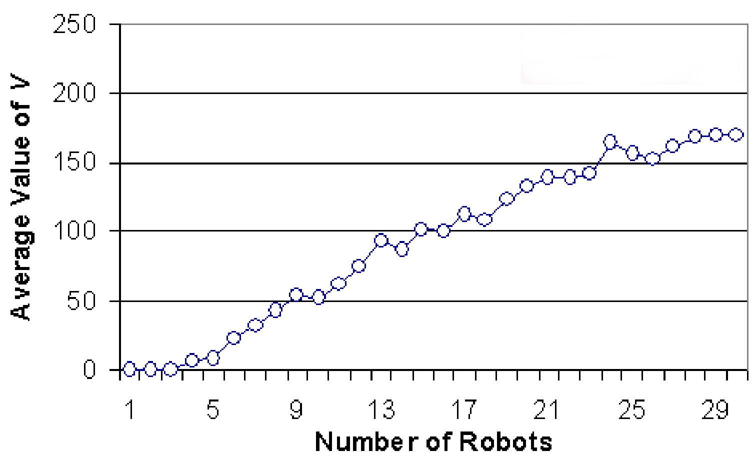

Figure 16: Average threshold values, V, between robots using adaptive coordination method when $\mathbf{P}_{Time}$ = 1.0 and $\mathbf{P}_{Fuel}$ = 0.0

Our working hypothesis is that fluctuations in the level of collisions even within one trial allow for this adaptive method to outperform the static ones it is based on. The Adaptive method adapted to these fluctuations, yielding the marked improvement in this group's productivity over other groups. As empirical evidence of these fluctuation within trials, Figure 17 represents the percentage of robots that are colliding throughout the course of three trials (540000 cycles) in groups of 25 robots. The X-axis in this graph represents the number of cycles elapsed in the trial (measured in hundreds of cycles), while the Y-axis measures the percentage of robots colliding at that time. We found that these values do in fact fluctuate, at times sharply, throughout almost all foraging trials. This further illustrates the danger in attempting to converge on one ideal coordination method, even within one trial.

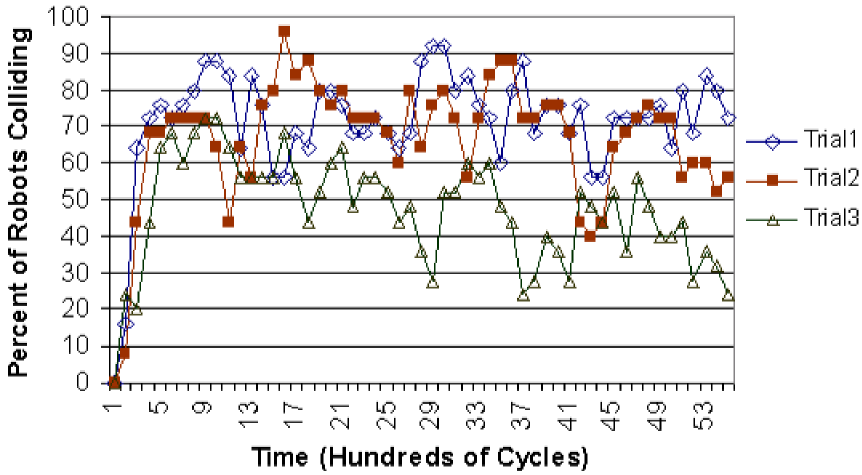

Figure 17: Fluctuations in collisions over time

We believe this is also the reason why the Adaptive method significantly outperformed the Uniform Adapt group in larger group sizes. At times, the Uniform Adapt approach may be advantageous as some robots could cue others as to the best coordination method to use. Notice how this group did have slightly higher productivity in small to medium groups (refer back to Figure 11). However, we believe the Uniform Adapt method has two major drawbacks. First, it requires communication between robots, a factor that would likely add another coordination cost, $\mathcal{C}_i^j$ to every agent in a group. However, even beyond this point, we believe the first approach is more effective in allowing robots to adapt to their local domain conditions. In domains with dynamics, such as the ones we studied, at least one robot is typically not colliding, and thus would naturally choose the least costly Noise coordination method. In the Uniform method, this one robot could force the entire group to switch back to this method, accounting for the lower productivity in this group when more costly methods were justified. In the future, we hope to further study how adaptation can yield improvements in productivity, even when standardized adaptation is required.

As further empirical evidence of the effectiveness of this adaptive approach, we present portions of simulated foraging runs captured from the Teambots simulator [4]. These results can be found at: http://www.jct.ac.il/ rosenfa/movies/aij2007.htm. Note that the simple Noise method is not able to resolve coordination conflicts in groups of 20 robots, while it is quite successful in smaller groups of 5 robots. Both the Repel and Aggression methods are moderately successful in resolving conflicts in groups of 20 robots. However, as the Adaptive movie demonstrates, this method is able to significantly outperform these three methods by effectively switching between the static methods.

Finally, observe that the gains from the Adaptive approach in the foraging domain that switched between coordination methods (see Figure 11) were much greater than the adaptive methods that tweaked the parameter strength within one method (Figure 10). We believe this difference is primarily due to the large differences in the density distributions and cost functions (refer to Figures 3 and 5) created by these methods in this domain. As a result, when the Adaptive approach switched between these sharply different coordination methods, it benefited from larger productivity gains.

In contrast, the first type of adaptation, i.e. adaptation within one coordination method, did not have as large differences in the variations within one coordination method (see Figure 4). As a result, adaptation did not facilitate radically different approaches to coordination, and productivity gains from this category of adaptation did not significantly outperform the methods it was based on. Similarly, the search domain only had two methods to switch between, with only modest differences in their cost functions (Figure 13). We believe that this prevented the adaptive methods in this domain from realizing even larger productivity improvements.

6 Related Work

This work uses a novel CCC group measure to create dynamic coordination that improve a group's productivity. Our approach is related to several existing research topics including: Algorithm selection [1, 13, 20, 22], coordination and teamwork [14, 21, 24, 31], group behavior measures [5, 15, 17] and dynamic coordination [9]. We discuss these below.

6.1 Algorithm Selection

We draw inspiration from previous work in automatic algorithm selection, where the challenge is to find a mapping between a portfolio of algorithms and problem instances. Allen and Minton [1] suggest running all algorithms in the portfolio for a short period of time, and then selecting the best algorithm based on secondary performance characteristics compiled from this preliminary trial. Gomes and Selman [13] suggest running several algorithms (or randomized instances of the same algorithm) in parallel.

A different approach uses machine learning to learn the mapping of algorithms to problems. For example, Brown et al. [22], use a machine learning boosting approach to create a classifier to select the best algorithm. They predict which algorithm will be best based on this classifier, and then execute the algorithm based on this prediction. Lagoudakis and Littman [20] concentrate on recursive algorithms such as sorting order statistic selection problems, and use a Markov Decision Process model to select the best algorithm.

All of these previous approaches involve a single agent or system, and thus the selection process is centralized. Moreover, in most cases (with the exception of [20]), algorithm selection occurs at a global level. In contrast, coordination problems are inherently distributed, and our approach involves local—and distributed—adaptation.

6.2 Coordination and Teamwork Models

Coordination can be defined as "managing dependencies between activities" [24]. Previous work by Malone and Crowston study how coordination is defined across multiple disciplines including organization theory, psychology, economics, and artificial intelligence. While their work presents a number of theoretical definitions for coordination, none of these are directly applicable for describing how one may “manage dependencies” in an optimal or near optimal fashion [24].

More generally, many different coordination frameworks have been previously proposed within the distributed artificial intelligence community [14, 18, 21, 31]. While these approaches generally address teamwork issues, they do not address specific coordination measures or the relative effectiveness between approaches. The SharedPlans approach [14] consists of creating teamwork recipes based on models of beliefs and intentions. Tambe’s STEAM [31] provides a domain-independent teamwork engine. The TAEMS framework [21] consists of a rule based approach to quantifying coordination relationships. BITE [18] allows the designer of a robot team to mix-and-match different coordination methods to different points in the execution of a task, but the choice is made before run-time. These previous investigations did not explore on-line adaptation of the coordination methods.

One set of approaches [6] suggest using a game-theoretic decision framework to negotiate a decision between agents about which behavior to choose. However, these approaches are useful for characterizing self-interested agents, while our work focuses on a cooperative environment. Other approaches model these problems based on a Markov Decision Process (MDP) model [27] which can be used even within cooperative environments. However, these approaches demonstrate the inherent complexity in selecting the optimal action within these models, and certainly cannot trivially solve which action to choose. Pynadath and Tambe demonstrate that finding optimal teamwork behaviors, even in small groups, is a computationally intractable problem for most real-world problem instances. Thus, the question of the optimality of any one of these approaches is difficult to ascertain. Our adaptive coordination methods based on the CCC measure may be of significance in helping determine which type, or family of coordination methods to use, or even to switch between coordination models during task execution.

6.3 Group Behavior Measures

To date, very few studies have been conducted contrasting a group’s composition and its task performance.

The CCC, as a coordination measure, is most closely related to Goldberg and Matarić’s interference measure [11]. Both our work and theirs focus on the efforts spent on coordination in forming a coordination measure. However, there is a difference in the definition of the measure. The CCC measure focuses on resources spent on resolving group conflicts regardless if they are before, during, or after events such as collisions. In contrast, Goldberg and Matarić’s interference measure studies the time robots actually collide. This difference in definition may account for differences in findings: They report increased productivity as their interference measure grew, while we found that productivity decreased as the CCC measure grew. Additionally, the work by Goldberg and Matarić equates coordination methods with homogeneous and heterogeneous capabilities. In contrast, our work studies groups of homogeneous robots. We believe that in order to incorporate heterogeneous capabilities into the CCC coordination measure, some normalizing must occur to equate group members. We are currently researching what extensions are necessary to create this unified measure.

Balch [5] presents a metric of *social entropy* which can measure the level of diversity or how heterogeneous a group is. He shows that certain tasks are intrinsically better suited for homogeneous groups, with others for heterogeneous ones. He finds his measure positively correlated with the group's productivity in some domains, and negatively correlated in others. Our CCC does not measure heterogeneity, but individual resource expenditure. We believe it is always negatively correlated with performance. Furthermore, we show that adaptive methods can be created based on the CCC. We believe it may be possible to expand the coordination measure to account for heterogeneous robots, and hope to study these types of groups in the future.

Kaminka and Tambe [17] use an *average time to agreement* (ATA) measure to study a team's behavior in the RoboCup simulated soccer domain. This measure evaluates the relative effectiveness of social monitoring of team behaviors. Similar to our work, this measure aims to provide feedback about team effectiveness through their measure. However, the question of correlation between productivity and the ATA measure was left open.

Hogg and Jennings [15] introduce a *willingness to cooperate factor* which defines the degree to which social agents engage in individual versus group considerations. Similar to our structure, they use their measure to alter agents' activity to resource constraints that are sensed during run-time. However, their formalized structure is less flexible to change than ours and requires a Q-learning model to allow for adaptation. As a result, it is unclear how their model could be applied for quickly reacting to domain dynamics. Furthermore, it is unclear how their framework could be modified or applied to specifically address coordination issues.

6.4 Dynamic and Adaptive Coordination

Our main focus lies in the ability to use the CCC measure to create adaptive coordination methods that improve a group's performance. We achieved this goal because the CCC measure can be easily estimated during task execution and can thus be used to match the best method to given domain conditions. Previous works envisioned parts of this idea.

The concept of switching between groups of coordination methods was previously described as part of the TAEMS theoretical framework [21]. However, their work concedes the necessity of preplanning or replanning for contingencies, making the system unable to adapt to runtime dynamics.

While the work by Toledo and Jennings [9] demonstrates that coordination adaptation is possible even during runtime, several key differences exist with our work. Their formalized reasoning model as to which coordination method to use may not be easily transferable from the theoretical grid-world domains they studied to real-world domains or actual groups of coordination algorithms. Furthermore, their system does not always improve the group's performance. However, their work is quite significant as it can be viewed as a mature departure point for our work. By using the CCC coordination cost measure based on the actual resources being consumed in coordination activities, the CCC-based methods are easily transferable to new domains and coordination methods. Furthermore, both of our methods do improve performance, at times by significant amounts beyond the static methods they are based on.

7 Conclusion and Future Work

In this paper we argue that the coordination cost a single robot generates is a primary factor in determining the productivity of the entire group. In theory, robots should consistently demonstrate increasing marginal productivity gains. However, limiting production resources, such as the spatial limitations inherent in many robotic groups, prevents productivity gains by this theoretical amount. At times, adding robots then hurts performance, as was previously noted [29, 32]. We present a model for evaluating multi-attribute coordination cost functions that a single robot contains. Our CCC (combined coordination cost) measure quantifies a weighted sum of all production resource conflicts between members of a group. While other team measurements are possible, we find that focusing on this cost alone facilitates effective comparison between different coordination methods. This approach requires no centralized mechanism, with accurate coordination measures being taken autonomously by members of the group. We present two adaptive coordination methods based on the CCC measure, which both improve the group's performance and scalability properties in a statistically significant fashion in the foraging and search robotic domains we studied.

For future work, several directions are possible. We believe it may be possible to use the coordination measurements to predict when adding an agent to the group will be helpful. Team sizes could thus be modified to maximize the use of production resources. We also hope to study if similar measurements could model gains each robot adds to its group. Such a measurement would be useful for purposes of task allocation as it could identify which team member is best suited to perform given tasks. We are hopeful that the use of the CCC measure will replace domain and task specific cost functions. We believe this approach could facilitate additional advances in agent and robotic team research.

References

[1] John A. Allen and Steven Minton. Selecting the right heuristic algorithm: Runtime performance predictors. In *Canadian Conference on AI*, pages 41–53, 1996.

[2] T. Balch and R.C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.

[3] Tucker Balch. The impact of diversity on performance in multi-robot foraging. In *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pages 92–99, 1999.

[4] Tucker Balch. www.teambots.org, 2000.

[5] Tucker R. Balch and Ronald C. Arkin. *Behavioral diversity in learning robot teams*. PhD thesis, 1998.

[6] Guido Boella and Leendert W. N. van der Torre. A game theoretic approach to contracts in multiagent systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 36(1):68–79, 2006.

[7] Stanley L. Brue. Retrospectives: The law of diminishing returns. *The Journal of Economic Perspectives*, 7(3):185–192, 1993.

[8] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3:375–397, 1996.

[9] Cora Beatriz Excelente-Toledo and Nicholas R. Jennings. The dynamic selection of coordination mechanisms. *Autonomous Agents and Multi-Agent Systems*, 9:55–85, 2004.

[10] M. Fontan and M. Matarić. Territorial multi-robot task division. *IEEE Transactions of Robotics and Automation, 14(5), 1998.*, pages 815–822, 1998.

[11] D. Goldberg and M. Matarić. Interference as a tool for designing and evaluating multi-robot controllers. In *AAAI/IAAI*, pages 637–642, 1997.

[12] 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*, pages 315–344, 2001.

[13] Carla P. Gomes and Bart Selman. Algorithm portfolios. *Artificial Intelligence (AIJ)*, 126(1-2):43–62, 2001.

[14] Barbara J. Grosz and Sarit Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[15] L. Hogg and N. Jennings. Socially intelligent reasoning for autonomous agents. *IEEE Transactions on Systems, Man and Cybernetics - Part A*, 31(5):381–399, 2001.

[16] M. Jager and B. Nebel. Dynamic decentralized area partitioning for cooperating cleaning robots. In *International Conference on Robotics and Automation*, pages 3577–3582, 2002.

[17] G. A. Kaminka and M. Tambe. Robust multi-agent teams via socially-attentive monitoring. *Journal of Artificial Intelligence Research*, 12:105–147, 2000.

[18] Gal A. Kaminka and Inna Frenkel. Flexible teamwork in behavior-based robots. In *AAAI-05*, 2005.

[19] N. Kohl and P. Stone. Machine learning for fast quadrupedal locomotion. In *The Nineteenth National Conference on Artificial Intelligence*, pages 611–616, 2004.

[20] Michail G. Lagoudakis and Michael L. Littman. Algorithm selection using reinforcement learning. In *17th International Conference on Machine Learning*, pages 511–518, 2000.

[21] V. Lesser, K. Decker, T. Wagner, N. Carver, A. Garvey, B. Horling, D. Neiman, R. Podorozhny, M. NagendraPrasad, A. Raja, R. Vincent, P. Xuan, and X.Q. Zhang. Evolution of the GPGP/TAEMS Domain-Independent Coordination Framework. *Autonomous Agents and Multi-Agent Systems*, 9(1):87–143, 2004.

[22] Kevin Leyton-Brown, Eugene Nudelman, Galen Andrew, Jim McFadden, and Yoav Shoham. Boosting as a metaphor for algorithm design. In *CP2003*, pages 899–903, 2003.

[23] Sridhar Mahadevan and Jonathan Connell. Automatic programming of behavior-based robots using reinforcement learning. In *National Conference on Artificial Intelligence*, pages 768–773, 1991.

[24] Thomas W. Malone and Kevin Crowston. The interdisciplinary study of coordination. *ACM Comput. Surv.*, 26(1):87–119, 1994.

[25] M. Matarić. Reinforcement learning in the multi-robot domain. In *Autonomous Robots*, pages 73–83, 1997.

[26] E. Ostergaard, G. Sukhatme, and M. Matarić. Emergent bucket brigading - a simple mechanism for improving performance in multi-robot constrainedspace foraging tasks. In *Proceedings of the 5th International Conference on Autonomous Agents*, pages 29–30, 2001.

[27] David V. Pynadath and Milind Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *JAIR*, 16:389–423, 2002.

[28] P. Rybski, A. Larson, M. Lindahl, and M. Gini. Performance evaluation of multiple robots in a search and retrieval task. In *Workshop on Artificial Intelligence and Manufacturing*, pages 153–160, 1998.

[29] 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*, pages 553–561, 1996.

[30] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[31] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.

[32] 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.

[33] C. J. C. H. Watkins. Learning from delayed rewards. *Ph.D. Dissertation, Kings College*, 1989.

[34] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[35] Ka Yoon and Ca Hwang. *Multiple Attribute Decision Making: an Introduction*. Prentice Hall, Thousand Oaks: Sage, 1995.