Adaptive Robotic Communication using Coordination Costs for Improved Trajectory Planning*

Avi Rosenfeld[1,2], Gal A. Kaminka[1], and Sarit Kraus[1,2]

[1]Computer Science Department, Bar-Ilan University
Ramat-Gan, Israel 92500

[2]University of Maryland Institute of Advanced Computer Studies
College Park, Maryland, 20742

Email: {rosenfa, galk, sarit}@cs.biu.ac.il

Abstract

Designers of robotic groups are faced with the formidable task of creating effective coordination architectures that can plan and replan trajectories even when faced with changing environment conditions and hardware failures. Communication between robots is one mechanism that can at times be helpful in such systems, but can also create a time and energy overhead that reduces performance. In dealing with this issue, various communication schemes have been proposed ranging from centralized and localized algorithms, to non-communicative methods. In this paper we argue that using a coordination cost measure can be useful for selecting the appropriate level of communication within such groups. We show that this measure can be used to create adaptive communication methods that switch between various communication approaches. Robotic team members that implemented these approaches were able to increase their productivity in a statistically significant fashion over methods that only used one type of communication scheme.

Introduction

Groups of robots are likely to accomplish certain tasks more quickly and robustly than single robots (Dudek, Jenkin, & Milios 2002; Goldberg & Matarić 2001; Jager & Nebel 2001). Many robotic domains such as robotic vacuuming, search and rescue, mine clearing, and waste cleanup are characterized by limited operating spaces where robots are likely to collide. In order to plan and replan robot trajectories in these situations, some type of information transfer is likely to be helpful in improving group productivity. This is especially true as robotic domains are typically fraught with dynamics and uncertainty such as hardware failures, changing environmental conditions, and noisy sensors.

Questions such as what to communicate and to whom have been the subject of recent study (Jager & Nebel 2001; Sen, Sekaran, & Hale 1994; Tews 2001). In theory, communication should always be advantageous–the more information a robot has, the better. However, one must also consider the resources consumed in communication itself, and if the cost of communication appropriately matches the needs of the domain.

*This material is based upon work supported in part by DARPA.

Scalability and productivity issues in robotic groups are likely to be impacted by the type of communication scheme used (Pynadath & Tambe 2002). We believe that each type of communication framework is best suited for planning trajectories in different environment conditions. Thus, one should not attempt to find the one optimal communication method, but a mechanism for optimally switching between different communication protocols so the best protocol is matched to the given environment.

This paper provides such a framework with its use of a coordination cost measure that quantifies all resources spent on coordination activities. Our model explicitly includes resources such as the time and energy spent communicating. In situations where conflicts between group members are common, more robust means of communication, such as centralized models, are most effective. When collisions are rare, coordination methods that do not communicate and thus have the lowest overhead, work best.

We present two novel domain independent adaptive communication methods that use coordination cost estimates to alter their communication approach based on domain conditions. In our first approach, robots uniformly switch their communication scheme between differing communication approaches used for trajectory planning. In this method, robots contain full implementations of several communication methods, and switch between them as needed. In contrast, our second approach represents a generalized communication scheme that allows each robot to adapt independently to its domain conditions. Every robot creates its own radius, which we refer to as its own neighborhood of communication, to create a sliding scale of communication between localized to centralized methods. This approach also uses coordination cost estimates to control how large a communication neighborhood to create, and thus how much communication to use with its teammates.

In order to evaluate these adaptive methods, we performed thousands of trials using an established robotic simulator to empirically confirm the effectiveness of these approaches. We found that groups using these dynamic approaches were more successful in planning and replanning trajectories after projected collisions. As a result, the productivity levels of these groups significantly outperformed those of the non-adaptive algorithms they were based on.

Comparing Communication Algorithms

We begin by contrasting the strengths and weaknesses within No-Communication, Localized and Centralized approaches for trajectory planning. It is possible to create effective group behavior without any communication (Balch & Arkin 1998). At times, coordination without communication has been shown to allow agents better adaptability, robustness and scalability qualities over methods using communication (Sen, Sekaran, & Hale 1994). Additionally, the lack of communication also allows such methods to be implemented on simpler robots. However, such algorithms often require powerful and accurate sensing capabilities (Matarić 1997). Additionally, our results demonstrate that groups implementing these methods did not always provide the highest levels of productivity, especially within dynamic domains where frequent coordination conflicts exist.

A second series of approaches attempt to improve group performance by having robots locally communicate information (Jager & Nebel 2001; Matarić 1997). Within the work of Jäger and Nebel (Jager & Nebel 2001), robots that near a collision stopped to exchange trajectory information. They then successfully detect and resolve deadlock conditions of two or more robots mutually blocking. However, their trajectory planning method was not able to perform well in groups of over 5 robots. In contrast, Mataric (Matarić 1997) reported that a local communication scheme scaled well with group size. One key difference seems to lie within the localized communication implementations. In Jäger's algorithm, one or more robots must stop moving during trajectory replanning. We believe this led to a breakdown in the system once the group size grew. Mataric's locally communicating robots broadcast information while continuing their foraging task. This allowed for better scalability qualities.

A third type of approach involves the use of some type of central repository of information (Tews 2001) to plan trajectories. This centralized body, which could also be implemented as one "expert" teammate, would then be able to easily share its store of pooled information with other teammates. While this approach allows for free information sharing and can thus improve performance, several drawbacks are evident. First, the centralized mechanism creates a single point of failure. The overhead involved with communication is also likely to be large, requiring hardware and bandwidth suitable for simultaneous communication with the centralized body. While these drawbacks are at times significant, they may be justified given the needs of the domain.

Coordination Cost Model

Our coordination cost measure facilitates identifying which communication method is most suitable given the environment. This measure quantifies the total productivity lost due to coordination conflicts. In methods without explicit coordination, such as the ones we previously studied (Rosenfeld, Kaminka, & Kraus 2004), this resulted exclusively from behaviors spent before, during, and after collisions. However, robotic groups that use communication may lose productivity because of their communication mechanisms. At times, these algorithms involve group members stopping to send and receive information and thus time is a limiting production factor (Jager & Nebel 2001). In other settings, resources such as energy and bandwidth are limiting factors (Singh, Woo, & Raghavendra 1998).

We model every robot's coordination cost $\mathcal{C}_i$, as a factor that impacts the entire group's productivity. We analyze two cost categories: (i) costs relating to communication and (ii) proactive and/or reactive trajectory changes to resolve possible collisions. We then combine these factors to create a multi-attribute cost function based on the Simple Additive Weighting (SAW) method (Yoon & Hwang 1995) often used for multi-attribute utility functions.

We found when implementing these methods that the use of communication, or lack thereof, can affect the time or energy used in collision avoidance and resolution behaviors. When possible, less communication should be used as communication itself consumes resources that detract from that group's production ability. However, while methods with no communication have no $\mathcal{C}_i$ associated with this category, this method could not always successfully resolve collisions and then spent more resources on collision resolution behaviors, or another $\mathcal{C}_i$. Conversely, the centralized methods incurred a communication cost $\mathcal{C}_i$ that often eclipsed the needs of the domain and weighed heavily on productivity. However, when collisions were frequent, this cost was justified in preventing and resolving collisions, and this group achieved the highest productivity.

Our hypothesis is that coordination costs, whether measured in terms of time, energy, bandwidth, or other production resources, must appropriately match the needs of the domain. While our implementation did in fact use a few domain specific issues, such as the location of the home base within the domain, we found that many parameters used could be easily changed without effecting the net result. Our next section details our implementation details, and all costs associated with communication.

Implementing Three Communication Types

To the best of our knowledge, our work is unique in that we implemented representative methods of all three communication types within the same domain. Our first goal was to highlight key differences between No-Communication, Localized and Centralized methods within a foraging domain. We chose the foraging domain as our testbed as it has been extensively studied. This domain is defined as locating target items from a search region *S*, and delivering them to a goal region *G* (Goldberg & Matarić 2001). Foraging robots often collide as they approach the home base(s) within their area of operation.

We used the Teambots (Balch 2000) simulator to implement these types of communication within groups of Nomad N150 robots. We used a total of 60 target pucks spread throughout an operating area of approximately 10 by 10 meters. We measured how many pucks were delivered to the goal region within 9 minutes by groups of 2–30 robots using each communication type. We averaged the results of 100 trials for each group size with the robots being placed at random initial positions for each run. Within our foraging implementation there was only one goal region, *G*, which was located in the center of the operating area. As previous foraging studies found (Goldberg & Matarić 2001), spatial conflicts often occurred around this area.

We created experiment sets measuring the time and energy spent in two coordination categories–communication and collision resolution. The coordination costs in our first set of experiments involved the time spent in communication and trajectory correction behaviors out of each trial's total time of 9 minutes. We assumed robots pairs stopped for 1/5 of a second to communicate, representing some methods (Jager & Nebel 2001) where robots stop to exchange information. In our second set of experiments, we allocated each robot 500 units of fuel. We assumed most of the fuel was used by the robots to move, with a smaller amount (1 unit per 100 seconds) used to maintain basic sensors and processing. In the energy based localized experiments, we assumed robots did not stop to communicate, as is the case with other methods (Matarić 1997), but each robot still spent 0.3 units of fuel per communication exchange. Our coordination cost involved the amount of fuel that was used in communication and trajectory correction behaviors.

The three communication schemes we created were similar in that they resolved collisions by changing their trajectories to mutually repel from teammate(s) sensed within a certain safe distance ε, which we set to 1.5 robot radii. Once within this distance, robots acted as they were in danger of colliding and used repulsions schemes to alter their trajectories. The No-Communication was unique in that robots never used time or fuel to communicate, and thus only had costs relating to the repulsion behaviors robots engaged in. This method assumed domain specific information, namely it based itself on the robot's autonomously computed scalar distance, $\mathcal{S}$, from its location to the home base in the domain. Robots used a function of this distance, which we implemented to be $5\mathcal{S}$, as the time to use the repulsion trajectory after a projected collision.

The localized method used less domain specific information and is similar to the localized methods previously proposed (Jager & Nebel 2001), (Matarić 1997). Communication between robots was initiated once it was in danger of colliding–a teammate came within the ε distance. After this event, these group members would exchange information about their trajectories (here their relative distances from their typical target, their home base). The closer robot then moved forward, while the other robot used a repulsion trajectory for a fixed period of 20 seconds.

Our final method, *Centralized*, used a centralized server with a database of the location of all the robots similar to other centralized methods (Tews 2001). Within this method, one of two events triggered communication. First, as with the localized method, robots dropping within the ε distance initiated communication by reporting its position, done here with the centralized server. The server then reported back a repulsion trajectory based on its relative position to all other teammates. However, in order for the server to store a good estimate of the positions of all robots, a second, often more frequent type of communication was needed where each robot reported its position to the server with frequency L. If this communication occurred too frequently, this central database would have the best estimate of positions, but the time or energy spent on communication would spike, and productivity would plummet. If communication was infrequent, the latency of the information stored on the server would create outdated data. This in turn would reduce the effectiveness of this method, and result in more collisions. In order to minimize the lost productivity due to communication, once robots communicated with the server because of a collision, they waited the full latency period, L, before retransmitting their position. As a result, the centralized server often received position information at different times, but still enforced a maximal latency condition.

Analyzing the Cost of Communication

Our results from experiments involving time and energy costs support the claim that the best method of communication does change with domain conditions. Figure 1 contains the results from the time based coordination cost trials. In the top portion of the graph, the X-axis represents the group size, and the Y-axis the number of pucks successfully retrieved within each group. The No-Communication approach worked best in small groups where collisions were less likely. In medium sized groups, the localized approach worked better. As collisions became frequent, the large amount of communication inherent in the centralized method became justified, and this group performed significantly better. The total cost of coordination as a function of time are presented in the lower graph in Figure 1.

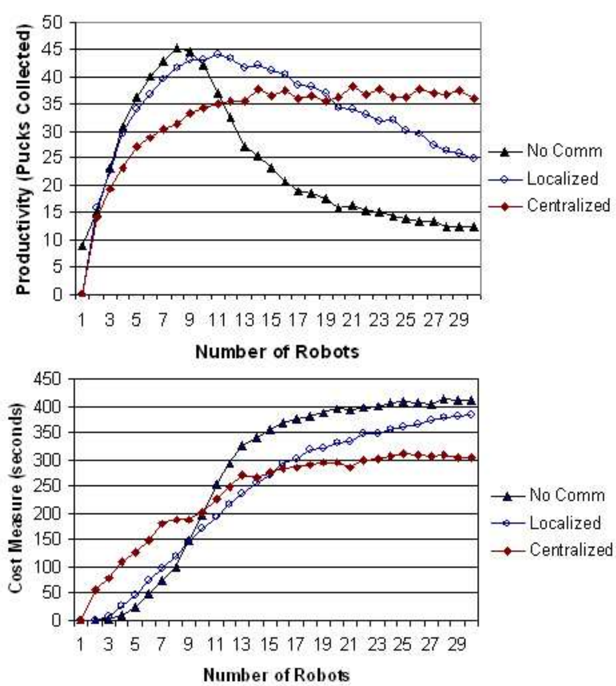

Figure 1: Comparing Levels of Time Spent on Communication in Different Group Sizes

Notice that the No-Communication method was only effective in minimizing this cost (presented as the Y-axis and measured in seconds) for small groups (the X-axis). In larger groups, this method engaged in more repulsion behaviors because it was not successful in trajectory replanning without communication. The localized group maintained near linear levels of its coordination cost with respect to the group size but the communication costs within this group made it less effective in smaller groups. The centralized method had the largest cost overhead, but these costs were not as effected by group size. As a result, this group achieved the highest productivity in large groups.

We also found a very strong negative correlation between the coordination cost based on *energy*, and the groups' corresponding productivity. In these trials, we measured the total energy used by our groups in coordination behaviors, including communication. As was the case in the time based experiments, we again found the best method changed as the group size increased, and thus collisions became more likely. The No-Communication method again fared best in small groups, the localized one in medium groups with the centralized method faring best in larger groups. We omit displaying this graph due to lack of space.

Both sets of experiments had similar results in that the team's productivity was strongly negatively correlated with coordination costs. In the time experiments, we found an average correlation of -0.96 between the productivity found in groups of 2–30 robots and the group's corresponding cost. In the equivalent energy based experiments, we found a value of -0.95.

It is important to stress that we implemented several variations of the parameters used in the No-Communication, Localized and Centralized methods with all variations also demonstrating this same high negative correlation as well. The parameters used within our methods affected the coordination cost, and thus the productivity outcome. For example, we studied 7 latency variations within the Centralized method in both experiment sets. Our groups enforced maximal latency periods of L set to 0.1, 0.3, 1, 5, 10, 30 and 60 seconds. In our time based experiments we found that a latency of 1 second produced the highest productivity in this group. In our energy based experiments (see figure 2 below), we found that a latency of 5 seconds yielded higher productivity. This is because the cost of communication (1/10 of a second) in the first trials was different than this cost (0.3 units of fuel) in the second. However, in both cases the productivity of these variations was highly negatively correlated with their relative coordination costs. In the first case, we found a correlation of -0.95 between these latency variations and the corresponding coordination cost based on time. In the trials based on fuel, this value was -0.97.

Within both experiments we found that latencies set too high typically converged with those groups where it was set too short. For example, figure 2 displays our latency productivity variations in the energy trial sets. We graphed the productivity levels (Y-axis) of the 7 latency variations as a function of the group size (X-axis). Notice how methods that update their information frequently often have the same productivity levels of methods that infrequently communicate. For example, Latency 0.2 (communication every 0.2 seconds) converges with Latency60 (communication one a minute). The coordination cost levels in these pairs are similar as well. While Latency0.2's frequent communication makes its cost primarily due to communication, Latency60's infrequent communication often made the database of teammates' positions inaccurate. An attempt to unwisely reduce communication, and this type of cost, led to an increase of repulsion behaviors, or a second type coordination cost.

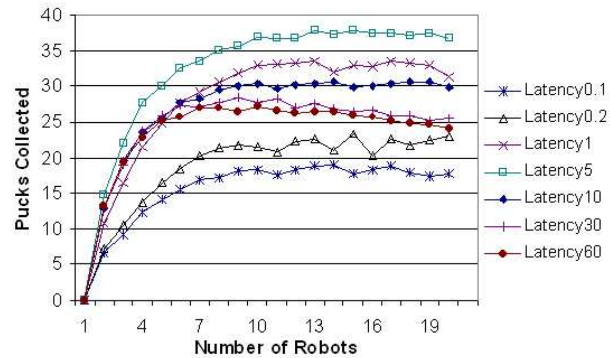

Figure 2: Comparing Latency Differences and Productivity Levels for Centralized Method in Energy Experiments

Communication Adaptation based on Cost

Not only is this measure useful for comparing communication methods, but it can also be used for online adaptation between communication schemes. In this section we present two types of adaptive methods: (i) uniform communication adaptation (ii) adaptive neighborhoods of communication. Both methods led to significant increases in productivity over static approaches.

Uniform Switching Between Methods

In our first method, all robots simultaneously switch between mutually exclusive communication methods as needed. In order to facilitate this form of adaptation, each robot autonomously maintains a cost estimate, V used to decide which communication method to use. As a robot detects no resource conflicts, it decreases an estimate of this cost, V, by an amount W_{down}. When a robot senses a conflict is occurring, the value of V is increased by an amount W_{up}. The values for V are then mapped to a set of communication schemes methods ranging from those with little cost overhead such as those with no communication, to more robust methods with higher overheads such as the localized and centralized methods. As the level of projected conflicts rises (as becomes more likely in larger group sizes) the value of V rises in turn, and the robots use progressively more aggressive communication methods to more effectively resolve projected collisions. While these activities themselves constitute a cost that detracts from the group's productivity, they are necessary as more simple behaviors did not suffice. As different coordination methods often have different costs, $\mathcal{C}_i$ for a given domain, we believed this approach could be used to significantly improve the productivity of the group.

Several key issues needed to be addressed in implementing this method with groups of robots. First, we assumed that all group members are aware of the overheads associated with various coordination methods, and can order them based on their relative complexities. This ordering can be derived from theoretical analysis or through observation (as we did in the previous section). Second, an approach to quickly set the weights, W_{up}, and W_{down} used within our algorithms is needed. It is important to stress that our goal is not to converge on any one optimal communication method as we found that dynamics within the domain require different coordination approaches throughout the task completion. Instead, our goal is create a policy, π, based on the coordination cost estimate V, to optimally change the communication method each agent uses. While traditional learning methods, such as Q-learning (Watkins 1989) may converge on an optimal policy, the large number of trials needed to arrive at this result are not practical to implement on real robots (Kohl & Stone 2004). Thus, our approach is to sacrifice finding a globally optimal policy in exchange for finding a locally optimal policy after a much shorter training period for our weights. Previous work by Kohl and Stone (Kohl & Stone 2004) contrasted Hill Climbing, Amoeba, Generic Algorithms, and Gradient Learning algorithms for finding a faster walking speed for Sony Aibo robots. We used a gradient learning algorithm similar to their approach to arrive at the weight policy π used in our experiments.

Next, it must be noted that this method requires all robots to change communication in sync because of the mutual exclusivity of the methods used. For example, it is impossible for one robot to use a centralized method, with others using one without communication, as the centralized approach is based on information from all team members. As a result, once any one robot in the group autonomously decided it needed to switch communication schemes, a communication change must also occur within all other team members. This could force certain members to use a more expensive communication method than it locally found necessary. Also, the switching process itself likely involves a cost between all robots in the group. For the time based experiments, we again assumed this would take 1/10 of a second, and in the fuel based experiments this change required 0.3 units of fuel. We relaxed these requirements in the second adaptive method, presented in the next section, thus avoiding these issues.

Finally, care must be taken to prevent the robots from quickly oscillating between methods based on their localized conditions. In our implementation, communication adaptation was triggered once one robot's value for V exceeded a certain threshold. After this point, that robot broadcasted which method it was switching to and all group members would change in kind and reinitialize their cost estimates V to this new value. Furthermore, we also used domain specific information, such as prioritizing collisions closer to the home base within our foraging domain. In this fashion, we partially limited the types of triggers to those of importance to the entire group. Once again, our second type of communication adaptation relaxes this requirement and is effective without any such heuristics.

Adaptive Neighborhoods of Communication

The advantage in our first adaptive approach lies in its simplicity. Our uniform adaptive approach switches between existing coordination methods based on estimated coordination costs. Assuming one analyzes a new domain with completely different communication methods, and can order the communication methods based on their communication costs, this approach will be equally valid as it implements existing methods and reaches the highest levels of productivity from among those methods–whatever they may be.

In contrast, our second adaptation method is a parameterized generalization of the three specific categories of communication methods (No-Communication, Localized, and Centralized). As many robotic domains use elements of these same methods (Dudek, Jenkin, & Milios 2002; Kaminka & Glick 2006; Jager & Nebel 2001; Tews 2001), we reason that a similar approach is likely to work in these and other domains as well.

The basis of our hybrid approach is introducing a second parameter to control how large a radius of communication is used. In our previous methods, robots shared information once they detected another teammate within a distance of ε. This hybrid method uses a second distance d inside which robots exchange information, which we term its communication neighborhood. Formally, this radius of communication could be considered a neighborhood Γ of size d, created from robot v and includes all teammates, u, inside this radius. As such, we represent the neighborhood as $\Gamma_d(v) = \{u | \, u \text{ robot}, dist(u, v) \leq d\}$. Once any robot, for example Robot A, detects another robot within the ε distance, it initiates communication with all robots found within the $\Gamma_d(A)$ area. All robots in $\Gamma_d(A)$ must then report back to Robot A with their relative positions. Robot A then sorts all robots' positions by their relative distances from the home base in the domain. This robot then reports back to every robot within $\Gamma_d(A)$ a trajectory heading and the minimum time to use this heading based on that robot's relative position in the neighborhood. All robots, including the initiating robot (robot A), then adopt this new heading for the dictated length of time. It is possible that a robot may be a member of more than one neighborhood. In such cases, robots accept the trajectory value with the larger time regardless of the sender.

While we consider this approach a hybrid of the three previously described categories, some implementation details are different in this method. While the repel amounts of the robot initiating communication (Robot A) are calculated in a similar fashion to the previously described centralized method, here these values are calculated by members of the team, instead of one centralized server. The radius of communication in the centralized approach is the full width of the domain, while the Γ_d radius is typically much smaller. However, the biggest difference in implementing this approach is how these trajectory values are obtained. Robots in previous methods changed trajectories based on communication exchanges which robots within the ε distance shared. In this method, robots may alter course if they enter the Γ_d radius even if they are not in immediate danger of colliding. The reason for this is as follows. As robots within the Γ_d radius are typically close to each other, we found that these robots often would soon initiate their own radii of communication. In other methods this was not a concern, as other teammates were not effected by this phenomenon. However, here this would create multiple neighborhoods involving the same teammates. As a result, proactively assigning new trajectories was crucial for containing communication costs as Γ_d grew.

Despite these differences, adjusting the value of d in Γ_d can be used to approximate the previously studied communication categories. Assuming d is set to zero, no communication will ever be exchanged and this method is trivially equivalent to the No-Communication method. Assuming d is set to ε, this method will become similar to the Localized method and information will be exchanged only with the robot it is about to collide with. If d is set to the radius of the domain, the neighborhood of communication encompasses all teammates making this method becomes similar to the Centralized method, with the notable exception that this method may proactively force teammates to change their trajectories. Thus, the degree as to how centralized this method becomes exclusively depends on the value of d.

The novelty of this method is our ability to set d based on robots' costs estimates. As the neighborhood, Γ_d, becomes larger, robots within larger areas are forced to engage in repulsion behaviors. At times, larger neighborhoods of communication may be useful as they preempt the danger of collisions. As we have previously seen, a tradeoff exists between spending more resources on communication and the potential gain through better trajectory values gained from that information. Properly setting the value for d is critical in determining the success of this method given a group size.

To demonstrate the impact of d on the group's productivity, we first set this value to varying uniform constants within our robotic groups. We compared the productivity levels of foraging groups where d was set to 1, 2, 3, 5 and 50 robot lengths. Recall that ε is approximately 1 robot length (1.5 radii). Thus Γ_1 represents the nearly localized variation with Γ_{50} corresponding to the nearly centralized version of this method.

Figure 3 represents the relative productivity levels for of these static neighborhood groups relative to the energy costs levels measured in these groups. Notice how in small groups, Γ_1 yielded the highest average productivity. As we have seen, when possible, resources spent on coordination, here by creating large communication neighborhoods, should be avoided when possible. As small areas of communication sufficed in small groups, this approach had the highest productivity. As the group size grew, additional communication was necessary to maintain high productivity levels. As a result, larger neighborhoods were necessary and groups with Γ_5 resulted in the highest productivity. However, as we saw in the centralized latency values, forcing too much communication when not necessary created communication costs that reduced productivity to levels found in methods that spend too few resources on communication. In this method, notice that the productivity level of the Γ_{50} method, which created too large a neighborhood, approached those of Γ_1, which didn't create a large enough one. We again found a strong correlation between the various Γ_d variations and the groups' corresponding coordination costs and productivity with an average negative correlation of -0.96.

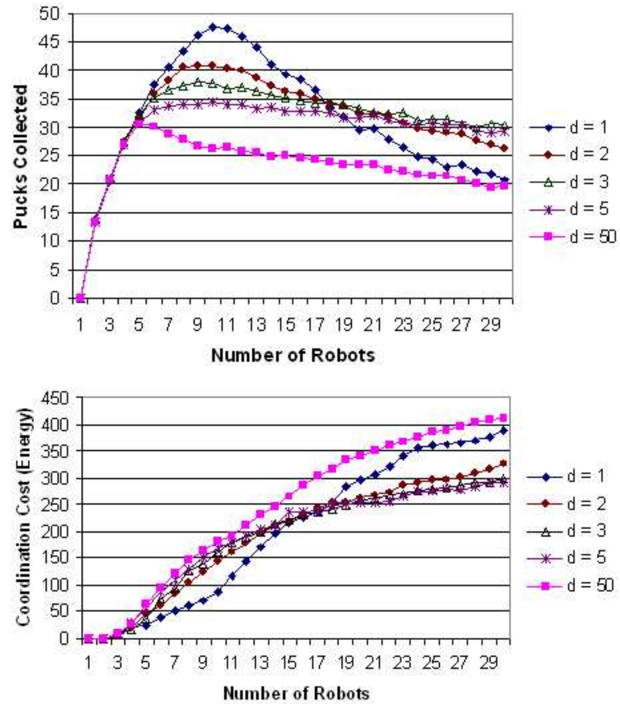

Figure 3: The Impact of Varying Neighborhood Sizes (d) on Productivity Levels and Costs in Energy Experiments

As no one neighborhood size is best suited for all environments, we again used coordination costs to allow robots to autonomously adjust their communication method to the needs of the domain. As in our first adaptive approach, each robot's coordination cost estimate V was autonomously adjusted through its weights W_{up} and W_{down}. We again set these weights through gradient learning to quickly arrive at a near optimal policy π for reacting to changing conditions. This value was then directly applied to set the value d for each robot's communication neighborhood.

The major advantage in this approach is its ability to allow every robot in the group to adjust its Γ_d as it independently calculates. This allows robots, and groups thereof, to easily adjust to localized conditions. For example, one robot could locally use a very large d value, initiating a large centralized communication exchange with teammates in its area while another robot simultaneously uses a much smaller value of d to conduct a much more localized communication exchange in a different area of the domain. In contrast, the uniform adaptive method forces all robots in the domain to switch communication methods in sync, not allowing this level of flexibility. As our next section demonstrates, this difference is quite significant and allowed the adaptive neighborhood communication method to yield even better results than the uniform one.

Adaptive Communication Results

We found that both forms of adaptation on average yielded a statistically significant improvement in productivity over the static groups they were based on. The uniform adaptation method was able to match or exceed the highest productivity levels of the groups it was based on. The neighborhood communication method was even more successful and often outperformed the uniform adaptation method by a significant amount.

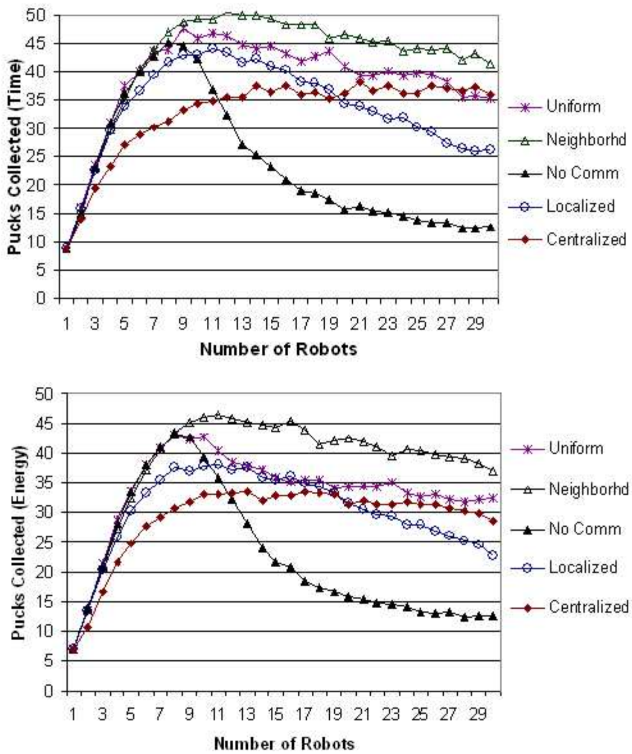

Figure 4: Comparing Adaptive Communication Methods based on Time and Energy Costs to Static Methods

We created uniform and neighborhood adaptive groups based on both time and energy coordination cost measures. Figure 4 shows the productivity results from these methods. For comparison purposes, we also graphed the No Communication, Local, and Centralized methods. Notice that both adaptive approaches approximated or significantly exceeded the highest productivity levels of the static methods they were based on. We attribute the success of both methods to their ability to change communication methods to the needs of the domain, with the neighborhood method having non-uniform adaptation, resulting in further boosting its productivity. For example, compare the energy based adaptive neighborhood method in figure 4 to the static variations in figure 3. Notice that in large groups, the adaptive neighborhood method often exceeded the productivity levels of the highest values from the static neighborhood methods. It is important to stress that the adaptive version allows for locally different neighborhood sizes, something none of the static neighborhood methods were capable of. This in turn facilitated better adaptation and higher productivity.

While both the uniform and neighborhood methods have advantages and disadvantages, each are capable of yielding impressive productivity gains over static approaches. The advantage of the uniform method is its ability to be easily transferred to new domains. Once a range of communication methods and their relative strengths are known, this adaptive method can switch between them to achieve the highest productivity from among static methods. When possible to implement, the neighborhood approach has the advantage of adapting communication to more localized areas in the domain, facilitating even higher productivity.

Related Work

A major challenge to designers of robotic groups exists in choosing an optimal communication method to share group information such as trajectories. Finding such a method in real-world environments is difficult to assess and computationally difficult (Pynadath & Tambe 2002). Towards addressing this problem, Pynadath and Tambe (Pynadath & Tambe 2002) proposed a theoretical framework for analyzing the type of communication needed for achieving cohesive behavior. More generally, formal teamwork approaches such as the Taems framework (Lesser *et al.* 2004) are useful in providing a rule based approach to quantifying coordination relationships. However, Wagner et al. (Wagner, Garvey, & Lesser 1997) demonstrated scheduling optimal behavior within such formal approaches are often subject to exponentially large possibilities, and thus novel approaches for limiting the number of possibilities is required. Planning robotic trajectories is even more difficult in that we found that the best form of coordination changes over the course of time, or as the task is being completed. Thus, various forms of adaptation and learning are likely to be needed to achieve improved coordination during task execution.

This work uses a coordination costs measure to compare a given set of communication methods and to create adaptive methods based on matching the best method to a given domain. The concept of switching between groups of coordination methods was already envisioned as part of the Taems framework (Lesser *et al.* 2004). However, their work concedes the necessity of preplanning or replanning for contingencies, making the system unable to adapt to runtime dynamics. While work by Toledo and Jennings (Excelente-Toledo & Jennings) demonstrated that coordination adaptation was possible during runtime, several key differences exist between their work and ours. First, they were not able to always improve performance through adaptive coordination methods, something that both of our methods are capable of. Also, their formalized reasoning model as to which coordination method to use is not easily transferrable from the theoretical grid world domains they studied to real-world domains or actual coordination algorithms. In contrast, our coordination cost measure is based on the actual resources being consumed in coordination activities, and thus is easily transferable to new domains and coordination methods.

Our coordination cost measure is based on our previously developed interference measure (Rosenfeld, Kaminka, & Kraus 2004). We defined *interference* as the total time each robot spends in resolving conflicts with other robots and found a strong negative correlation between a group's level of interference and its productivity. This work represents a significant extension to this metric as we now focus on all resources spent on coordination such as the time and energy spent in communication. Additionally, this work addresses issues specific to communication. For example, our previous work entertained adaptation between coordination methods where robots were allowed to adapt autonomously between mutually exclusive coordination methods. Such an approach is impossible here as many protocols require standardized communication between all team members. We address this issues through creating two novel communication adaptation methods, uniform or hybrid adaptation, based on our expanded coordination cost model.

Conclusion and Future Work

This work demonstrates how coordination costs can model the relative effectiveness of robotic communication in trajectory planning and replanning. Our measure focuses on the time and energy spent communicating and resolving collisions. We demonstrate the effectiveness of our methods in comparing between very different communication methods falling within categories of no communication, localized and centralized communication trajectory planning approaches. By using this information we are able to match the most effective communication scheme to a given robotic domain. We present two non-domain specific adaptive communication algorithms, uniform and neighborhood methods. We verify our hypothesis through thousands of simulated foraging trials in an accepted robotic simulator. While we find the neighborhood adaptive method to be more effective in the domain we studied, both approaches are likely to be applicable to many other domains (Dudek, Jenkin, & Milios 2002; Kaminka & Glick 2006; Jager & Nebel 2001; Tews 2001). It is possible that the uniform method is easier to implement or will yield better adaptive qualities in other domains.

We believe our cost measure hold promise for addressing several issues within dynamic planning and scheduling problems among autonomous agents. To date distributed scheduling approaches have typically focused on developing novel mechanisms where agents search through plan possibilities. While each agent may have different search criteria, the fundament approach between agents is uniform (Wagner, Garvey, & Lesser 1997). Our uniform adaptive approach provides a point of departure from this idea in that we switch between mutually exclusive methods as needed while the task is being completed. Our hybrid approach further expands on this approach, as we entertain agents using different planning criteria simultaneous within the same domain. For example, we envision that one agent may use local repair algorithms to quickly resolve a scheduling conflict, while another agent simultaneous creates a large neighborhood of communication, analyzing a much larger array of possibilities. We are currently developing methods to apply our coordination cost measure to create adaptive approaches within other planning and scheduling domains.

References

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

Balch, T. 2000. www.teambots.org.

Dudek, G.; Jenkin, M.; and Milios, E. 2002. A taxonomy for multi-agent robotics. *Robot Teams: From Diversity to Polymorphism, Balch, T. and Parker, L.E., eds., Natick, MA: A K Peters* 3:3–22.

Excelente-Toledo, C. B., and Jennings, N. R. The dynamic selection of coordination mechanisms. *Autonomous Agents and Multi-Agent Systems* 9:55–85.

Goldberg, D., and Matarić, M. 2001. Design and evaluation of robust behavior-based controllers for distributed multi-robot collection tasks. In *Robot Teams: From Diversity to Polymorphism*, 315–344.

Jager, M., and Nebel, B. 2001. Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots. In *IROS*, 1213–1219.

Kaminka, G. A., and Glick, R. 2006. Towards robust multi-robot formations. *ICRA-06*.

Kohl, N., and Stone, P. 2004. Machine learning for fast quadrupedal locomotion. *AAAI* 611–616.

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

Matarić, M. J. 1997. Using communication to reduce locality in multi-robot learning. In *AAAI/IAAI*, 643–648.

Pynadath, D. V., and Tambe, M. 2002. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of AI research* 16:389–423.

Rosenfeld, A.; Kaminka, G.; and Kraus, S. 2004. Adaptive robot coordination using interference metrics. In *The Sixteenth European Conference on Artificial Intelligence*, 910–916.

Sen, S.; Sekaran, M.; and Hale, J. 1994. Learning to coordinate without sharing information. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 426–431.

Singh, S.; Woo, M.; and Raghavendra, C. S. 1998. Power-aware routing in mobile ad hoc networks. In *Mobile Computing and Networking*, 181–190.

Tews, A. 2001. Adaptive multi-robot coordination for highly dynamic environments. In *CIMCA*.

Wagner, T. A.; Garvey, A. J.; and Lesser, V. R. 1997. Complex Goal Criteria and its Application in Design-to-Criteria Scheduling. *Proceedings of AAAI-97, also a version is available as UMass Computer Science Technical Report 1997-10* 294–301.

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

Yoon, K., and Hwang, C. 1995. *Multiple attribute decision making: an introduction*. Thousand Oaks: Sage: Prentice Hall.