Robotica (2019) volume 37, pp. 851–867. © Cambridge University Press 2019
doi:10.1017/S0263574718001364

Two Adaptive Communication Methods for Multi-Robot Collision Avoidance

Avi Rosenfeld*

Department of Computer Science, Jerusalem College of Technology, Jerusalem, Israel

(Accepted December 4, 2018. First published online: January 14, 2019)

SUMMARY
Designers of robotic groups are faced with the formidable task of creating effective coordination architectures that can deal with collisions due to changing environment conditions and hardware failures. Communication between robots is a mechanism that can at times be helpful in such systems, but can also create a time, energy, or computation overhead that reduces performance. In dealing with this issue, different communication schemes have been proposed ranging from those without any explicit communication, localized algorithms, and centralized or global communicative methods. Finding the optimal communication act is typically an intractable problem in real-world problems. As a result, we argue that at times group designers should use computationally bounded team communication approaches. We propose two such approaches: an algorithm selection approach to communication whereby robots choose between a known group of communication schemes and a parameterized communication framework whereby robots can reason about how large a communication radius is needed for a given problem. Both solutions use a novel coordination cost measure, combined coordination costs, to find the appropriate level of communication within such groups. Robots can then use this measure to create adaptive communication approaches that select between communication approaches as needed during task execution. We validated this approach through conducting extensive experiments in a canonical robotic foraging domain and found that robotic groups using these adaptive methods were able to significantly increase their productivity compared to teams that used only one type of communication scheme.

KEYWORDS: Control of robotic systems; Multi-robot systems; Navigation; Robot dynamics; Collision avoidance.

1. Introduction

Groups of robots are likely to accomplish certain tasks more quickly and robustly than single robots.[1–3] Many robotic domains such as robotic search and rescue, surveillance, and exploration, demining, vacuuming, and waste cleanup are characterized by limited operating spaces where robots are likely to collide.[4] In order to maintain group cohesion under such conditions, some type of information transfer is likely to be helpful in facilitating coherent behavior in robotic groups to help them better achieve their task. 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 continued study.[3–15] In theory, communication should always be advantageous—the more information a robot has, the better. However, assuming communication has a cost, one must also consider the resources consumed in communication, and whether the cost of communication appropriately matches the needs of the task. In some real-world environments, these costs will often likely come from the limitations in robots' hardware or their environment. For example, time constraints may arise from the

* Corresponding author. E-mail: rosenfa@jct.ac.il

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

overhead due to robots computing and processing information—something that may be critical in time-sensitive tasks. Energy constraints may arise from different energy consumption levels of the communication schemes—something that may be highly important if robots' batteries are limited. Environmental factors may also be important such as if only a limited bandwidth is available for robotic communication. Given these and similar costs, we believe that different communication schemes are each best suited for different environmental conditions. Because no one communication method is always most effective, an important challenge is to find a mechanism for switching between different communication schemes in response to the given environmental needs.

This paper presents two such adaptive communication frameworks for preventing collisions among robots. We use a novel combined coordination cost (CCC) measure that we developed to quantify all resources spent on coordination activities. Our model explicitly includes resources such as the time and energy spent communicating. In situations where collisions 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. In our first approach, robots uniformly switch their communication scheme between differing communication approaches. In this framework, robots store full implementations of several communication schemes including implicit (no-communication), localized, and global (centralized) methods, and switch between them as needed. In contrast, our second approach represents one generalized communication scheme that allows each robot to adapt independently to its domain conditions. In this approach each robot creates its own communication range radius, which we refer to as its *neighborhood of communication*, to create a sliding scale of communication between localized and centralized methods. Each robot uses its coordination cost estimate to determine how large its neighborhood should be.

To evaluate these adaptive methods, we performed thousands of trials using an established robotic simulator in the canonical multi-robot foraging task.[2,16–18] We tested groups of varying sizes and communication methods. We found that groups that used the adaptive methods often significantly exceeded the best productivity levels of the non-adaptive algorithms they were based on.

2. Related Work

A major challenge for designers of robotic groups exists in choosing an optimal communication method for preventing collisions within cluttered environments. Many practical frameworks have been presented for use within robotic teams and can generally be assigned to categories of implicit approaches with no-communication, localized, and centralized or global approaches.[1–5,12,19–27]

2.1. Different robotic communication schemes

It is possible to create effective group behavior without any communication.[28] Coordination without communication can potentially facilitate better adaptability, robustness, and scalability qualities over methods using communication.[10,19,25] In addition, the lack of communication also allows such methods to be implemented on simpler robots. However, such algorithms often require powerful and accurate sensing capabilities.[23] Also, our results demonstrate that groups implementing these methods did not always provide the highest levels of productivity, especially within dynamic domains where frequent collision conflicts exist.

A second set of approaches attempt to improve group performance by having robots locally communicate information.[3,9,23,24] For example, work of Jäger and Nebel[3] presents a method whereby robots nearing 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 five robots. In contrast, Matarić[23] reported that a local communication scheme scaled well with group size. One key difference seems to lie within the local 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 global information.[12,15] This centralized body, which could also be implemented as using an "auctioneer,"

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

"controller," or "expert" teammate, would could then be able to easily share its pooled information with other teammates. While this approach allows for complete information sharing and can thus improve performance, several drawbacks are evident. First, the centralized mechanism creates a single point of failure. The cost of communication is also likely to be large, and requires 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 task.

2.2. Finding the best communication scheme

A major challenge to designers of robotic groups exists in choosing an optimal communication method to share group information. Toward addressing this problem, Pynadath and Tambe[29] proposed a theoretical framework for analyzing the type of communication needed for achieving cohesive behavior. However, they also demonstrated that finding such a method in real-world environments is difficult to assess and computationally difficult.[29] 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.[17] Thus, various forms of adaptation and learning are likely to be needed to achieve improved coordination during task execution.

This work uses a CCC to compare a given set of communication methods and to create adaptive methods based on matching the best method to a given task. The concept of switching between groups of coordination methods was already envisioned as part of the TAEMS framework.[30] However, their work concedes the necessity of preplanning or replanning for contingencies, making the system unable to adapt to runtime dynamics. While work by Excelente-Toledo and Jennings[31] 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 transferable 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.

The CCC measure in this work is an extension of our previously developed robotic performance measures.[17,32] We had previously focused on the resources each robot spends in resolving collision conflicts with other robots and found a strong negative correlation between this measure and the group's 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. In addition, 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 schemes require standardized communication between all team members. We address these issues through creating two novel adaptation methods, uniform adaptation or communication neighborhoods, based on the CCC measure presented.

In the uniform adaptation approach in this work, we assume that representative communication methods from the implicit, localized, and centralized categories are predefined, and have been implemented with optimal values for their exact parameters given domain conditions. Several approaches exist for finding these parameters within a given coordination method. For example, work by Yoshida et al.[27] presented a framework to derive an optimal localized communication area within groups of robots to share information in a minimum amount of time. This approach assumes domain conditions such as spatial distributions and the probability of information transmission can be readily calculated. Similarly, work by Yan and Mostofi[13] presents how robots can optimize the bits of information they need to transmit under both energy and time constraints as they move along a predefined trajectory. While both of these approaches are impressive, they represent only one optimized possibility from among the previously described range of communication approaches. Thus, we propose subsuming these and other approaches within the library of communication schemes represented by the general uniform adaptation approach we present in this paper. Furthermore, even if these values are optimized, we claim that group dynamics may require changing these values during task execution. The second framework we present does just this by changing the size of the communication neighborhood.

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

3. Creating Adaptive Robotic Group Communication Methods with Costs

In this section we formalize the concepts of a coordination problem instance and discuss the impact different communication schemes will have on possible coordination instances within a given domain. We then present a generalized robotic density measure that can often significantly reduce the state space to allow us to identify which parameters in the problem instance affect the choice for the best communication method. We then present how the CCC measure is based on this understanding and helps facilitate adaptive communication. In the next section we present two adaptation frameworks and provide implementation details that use this information for improving robotic group communication.

3.1. Problem formalization

Many coordination problems have a variety of parameters that may affect which coordination algorithm is best. Generally, these parameters can include factors such as the number of agents in the group, possible actions such as movement or speech acts, environmental issues including the time to interact. Our hypothesis is that not all of these parameters will impact the group's performance and can be safely ignored in a given coordination instance.

We formalize problem instances as follows. Following previous work, we define a robot's *state* as an individual feature instance of that environment, while a set of *world states*, S, represents all possible combinations of values over all state instances.[29] Let $\{P_1, \ldots, P_N\}$ refer to a set of N possible domain problem parameters each corresponding to a different state in S. Each of these parameters can contain a value V_i with V_i being taken the domain of P_i and $V_i \in \{V_1, \ldots, V_N\}$. We define a problem instance as a tuple with values of all parameters such that one instance $\mathrm{I} = < v_1, v_2, v_3, \ldots, v_N >$ where $v_i \in V_i$. Thus, the total set of instances, S, of the problem would be the Cartesian product: $S = |V_1| \times |V_2| \times |V_3| \times |V_N|$. Let $\mathcal{CA}$ represent the library of deterministic communication algorithms that can be used by agents within the team to coordinate domain-level actions. Again following previous work,[29] we define domain-level actions, A_α, as $\{A_i\}_{i \in \alpha}$ to constitute the set of actions for each robot to perform to change its environment. These actions can include both communication acts and non-communication acts—such as moving to avoid collisions. Any one communication act, ca, is one of the set of possible communication acts, CA. Each communication act yields a deterministic utility value, such that Util: $|\mathcal{CA}| \times |S| \rightarrow \mathbb{R}$ for every problem instance $I \in S$.

The state space size grows exponentially because of the large number of possible interactions within the model. While any one parameter P_i may only have a relatively small number of possible values V_i, the Cartesian product of these possible values grows exponentially. Assuming every agent within the group can independently choose its actions, the problem space grows to: $|\mathcal{CA}| \times |S| \times |A|$, with A being the number of agents in the group. However, as we consider a group environment, each agent must coordinate its individual action with others in the group to form *joint actions*. Following work by Klein et al.,[33] we define joint actions as an extended set of actions that are carried out by a robot who is coordinating with other members of its group.[33] Finally, we assume each agent can make repeated decisions over the course of the task completion over t time units; this space becomes: $|\mathcal{CA}| \times |S| \times |A| \times |t|$.

Unfortunately, this state space is typically intractably large due to the number of joint actions possible,[29] requiring novel approaches to make tractable decisions. The popular Markov-based approach assumes that the actions at each and every time cycle can be made independently, thus eliminating the need to consider previous states. However, this approach is not feasible given the state space size. Instead, we focus on reducing the state space by individually analyzing the different parameters. We further cluster problem instances around which ones are best suited for each communication algorithm. This process significantly reduces the problem state size without any significant loss of information.

This paper presents two forms of *adaptive communication* to facilitate effective group behavior despite the inherent complexity within this task. Similar to work by Shen et. al,[34] we define adaptive communication as the process by which each robot reasons about its current state and adjust its communication strategy accordingly. While their work focused on physically re-configuring robots, our work proposes two approaches for adaptive communications without physically re-configuring robots.

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

One key contribution to the uniform adaptation framework is its approach to using the algorithm selection paradigm to greatly simplify the decision of which communication algorithm to use. As such each robot's possible *joint actions* reduce to one of only a small number of communication algorithms taken from $\mathcal{CA}$. Because our goal is to choose the best algorithm, a relative reward function value is sufficient. Once an algorithm has been identified as the best algorithm to use in a specific coordination problem, it receives the highest *reward function*, and thus should always be chosen. However, discovering these values for every one of the *joint action states* must still be addressed.

A second key contribution in the paper, evident in both adaptive communication frameworks, is that the group's reward function can be locally and autonomously approximated with novel teamwork metrics. Specifically, we posit that the CCC measure we developed facilitates an effective estimate of the reward using a given communication algorithm will have. Thus, robots do not need to reason about all possible joint actions, the costs of these actions, and the communication acts to produce them given all possible joint states. Within the first adaptive framework they only need to consider the estimated cost of the communication acts from the algorithms taken in $\mathcal{CA}$ and match the best algorithm to the given domain conditions. Within the second adaptive framework, they use the CCC measure to control the size of their communication neighborhood. The success of both frameworks is based on the assumption that the reward functions of different communication algorithms are quantifiable by the CCC measure. To help explain why this assumption is correct, we present the importance of understanding robot density, a measure that is locally perceptible. The next two sections further explain the CCC measure and its application to robotic communication.

3.2. The importance of monitoring robot density

We propose that performance 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 $B(r)$ surrounding p. During task completion, robots constantly move in and out of $B(r)$ with a certain heading β. At any given time t, there are k robots within any given area $B(r)$, where $k \leq N$. We denote the density, $\phi(r)$ as the total area of these k robots divided by the total area $B(r)$. The value of $\phi(r)$ will impact the group's performance. For example, $\phi(r) = 1$ indicates $B(r)$ contains no free space, and all robots mutually block, creating a deadlock situation. In these instances all productivity of the group will be lost until the area is cleared, and the density is 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 1 illustrates an example taken from the Teambots simulator with $k = 3$ robots within a radius $r = 1.5$ (m). Note that we study groups of homogeneous robots where each robot has a radius of approximately 0.25 m. We denote the area of each robot as B', where $B' = 0.25^2\pi$ or 0.20. Thus, the density $\phi(1.5)$ as illustrated here would be $(kB')/B(r)$ or $(3 \times 0.20)/7.1$ or 0.08.

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 m after a collision will have a lower density than a method that only requires robots to move away 1 m. Similarly, if only two robots are within the area r, they are not deadlocked and can easily move around each other. However, if one robot, say the central one in Fig. 1, is surrounded by robots, it is likely deadlocked, and its density is greater.

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

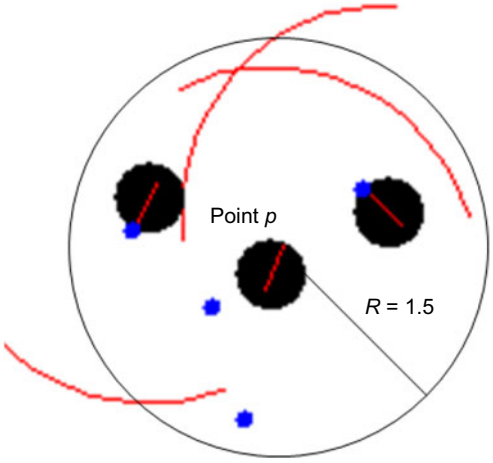


Fig. 1. Three robots within $B(r)$ where $r = 1.5$. Picture is taken from the Teambots simulator and is drawn to scale.

As a result, it will require more aggressive coordination methods that also coordinate the trajectories or many other robots around it.

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 $\phi(r)$. In these cases, implicit methods with little or no-communication as well as having small neighborhoods of communication will be best suited—especially after considering the costs of communication. As N grows, k will naturally grow as well, and naive methods will result in too high values for $\phi(r)$. In these cases, more aggressive communication methods will be needed to coordinate robustly dispersing the robots.

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. This speed change must be exactly calculated given the robot's hardware and task limitations. Second, any model we must include information about the robots' positions and headings throughout task completion, something that can be calculated within the environment we studied, but may not be easy to calculate within other robot sensors. In general, every robot heading toward p will have a velocity vector V_i based on its heading β from its initial position POS_i toward 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 $POS_{i_{x+\epsilon}}$ and $POS_{i_{y+\epsilon}}$ instead of the location POS_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.

Even without an optimal value for $\phi(r)$, we previously demonstrated two important characteristics based on our model:[17] (i) differences in density exist due to differences in coordination methods and (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 communication methods to specific domains based on their derived density as we now present. Specifically, we show how robots can autonomously estimate their density using coordination cost measures, thus facilitating a better selection of their communication protocol.

3.3. Using coordination costs to facilitate communication selection

Previously we developed a CCC measure that was useful in facilitating adaptation between different robotic collision avoidance algorithms. However, in the previous work,[17] none of the considered algorithms involved any communication. In this work, we extend the CCC model to address facilitating identifying which communication method is most suitable for the given environment.

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

We describe the CCC 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, that is, $\sum_{j=1}^{t} \mathbf{P}_j = 1$. 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,[16] we model resources spent in coordination even before a specific conflict, such as a robotic collision, occurs. Furthermore, multiple costs can be considered such as the resources spent on communication as well as the resources spent on avoiding collisions. For example, while methods with no-communication have no $\mathcal{C}_i$ for communication, this method could not always successfully resolve collisions and then spent more resources on collision resolution behaviors, or another $\mathcal{C}_i$. Conversely, centralized methods incurred a communication cost $\mathcal{C}_i$ that often eclipsed the needs of the task and weighed heavily on productivity. Other communication issues, such as bandwidth limitations, can similarly be categorized as additional cost factors as they impact any specific robot. Bandwidth costs could modeled as per other related works.[36] For example, if a robot needed to retransmit a message due to limited shared bandwidth, costs in terms of additional time latency and energy used in retransmission are likely to result. Thus, the total CCC measure is likely to be impacted by both of these factors.

In this paper we consider two uses of the CCC measure. First, we demonstrate how different communication methods are best suited for different environments. We show that as spatial constraints grow, coordination methods that use communication, despite its cost, are more effective in resolving collisions. This, in turn, yields a lower group density and higher performance. Second, we focus on using the CCC measure for online adaptation between communication schemes. We present two types of adaptive methods based on the CCC measure: (i) uniform communication adaptation and (ii) adaptive neighborhoods of communication. Both methods led to significant increases in productivity over static approaches (see Section 5). 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 task. While our implementation did in fact use a few domain-specific features, 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.

4. Implementing Two Adaptive Communication Frameworks

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 test bed as it has been extensively studied and is representative of other cooperative settings characterized by robots operating in spatially constrained areas.[37] This domain is defined as locating target items from a search region S and delivering them to a goal base G.[2] The total number of pucks delivered constitutes the group's *productivity*. Foraging robots often collide as they approach the home base(s) within their area of operation. These types of collision conflicts are common to many other domains such as formation control[20] and trajectory planning.[3]

We used the Teambots[38] 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 10×10 m. We measured how many pucks were delivered to the goal base within 9 min 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 base, G, which was located in the center of the operating area. As previous foraging studies found,[2] 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

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

involved the time spent in communication and trajectory correction behaviors out of each trial's total time of 9 min. We assumed robots pairs stopped for 1/5 of a second to communicate, representing some methods[3] 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 s, used to maintain basic sensors and processing. In the energy-based experiments, we assumed time costs did not exist and thus did not model these costs, but did instead assume robots spent 0.3 units of fuel per communication exchange. In addition to communication costs, we measured coordination costs as being the amount of time or fuel that was used in 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 method 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 $5\mathcal{S}$, meaning that after each projected collision the robot traveled along a repulsion trajectory composed of five times the distance of $\mathcal{S}$.

The localized method used less domain-specific information and is similar to the localized methods previously proposed.[3,23,24] 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 s.

Our final method, *Centralized*, used a centralized server with a global database of the location of all the robots similar to other centralized methods.[12,15] 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.

4.1. Uniform switching between methods

In our first adaptation framework, all robots simultaneously switch between mutually exclusive communication methods as needed. In order to facilitate this form of adaptation, each robot autonomously maintains an estimate of $\mathcal{C}_i$, V, used to decide which communication method to use. When a robot detects no collision 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.

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

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 do later in this paper. Second, an approach to quickly set the weights, W_{up}, and W_{down} used within our algorithms is needed. While traditional learning methods, such as Q-learning[39] may converge on an optimal policy, this approach is difficult to implement because of two major reasons. First, Q-learning is based on a a concept of "state" that is not readily definable 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. Even assuming an optimal policy could be learned, a second, more fundamental problem exists. Robotic domains often contain dynamics such as changes in robots' functioning or environment that render a learned policy obsolete very quickly.[40] 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. We describe the details of this approach in Section 4.3.

Next, it must be noted that uniform adaptation 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, this method assumes that the hardware within the robot allows for receiving simultaneous notifications that change is necessary with negligible cost. Even if a robot is within a "no-communication" method, we assume that this ability to receive such notifications is existent. This assumption is clearly hardware specific, but true within the simulated environment we considered. We relaxed these requirements in the second adaptive method, presented in the next section.

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. We ensure that the difference in the values of V be large enough that the change threshold cannot be quickly reached, thus preventing such oscillations or stability issues such as chattering or deadlock. 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.

4.2. 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 cost. 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,[1,3,12,20,27] we reason that a similar approach is likely to work in these and other domains as well.

The basis of this approach is introducing a parameter to control how large a radius of communication is used by each robot. This method uses a 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 $\Gamma_d(v) = \{u |\ u \text{ robot}, dist(u, v) \leq d\}$.

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

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

is trivially equivalent to the no-communication method. Assuming d is set to some small amount, 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 this method becomes similar to the centralized method. Thus, the degree of centralization exclusively depends on the value of d.

While we consider the neighborhood communication approach to be a parameterized generalization of the three previously described categories, some implementation details differ in this method over the static ones it emulates. Within this method, once any 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 projected trajectories. Robot A then sorts all robots' trajectories by their relative distances from the home base in the domain. This robot then reports back to every robot within $\Gamma_d(A)$ a repel value based on that robot's relative position in the neighborhood. All robots, including the initiating robot (robot A), then accept this value and immediately engage in repel behaviors 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 larger repel value regardless of the sender.

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 repel values are obtained. Robots in previous methods only repelled based on communication received after dropping within the ε distance. In this method, robots may repel 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. Thus, proactively assigning repel values was crucial for containing communication costs as Γ_d grew.

4.3. 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 collision 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[39] and other methods[41,42] guarantee the ability to find an optimal policy, 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.[43] Additionally, 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[29] demonstrated that finding an optimal policy in such cases is of intractable complexity (nondeterministic exponential time).

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.

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

Similar to the work by Kohl and Stone,[43] 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 five or more trials are needed for each data point due to the noise common within any given trial, even evaluating a group range of 1–30 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 either the communication scheme with the middle amount of communication (localized) in the first framework, or a medium sized neighborhood in the second framework. In the initial state, both adaptive methods coordination method could be viewed as containing a π with fixed values of V_{init}, W_{up}, and W_{down}. These weights would then impact the communication used in based frameworks. Within the first framework, assuming W_{down} lowers V below a certain threshold, then the implicit communication method is used. If W_{up} raises V above a second threshold, then the centralized method can be used. Within the neighborhood method, W_{down} and W_{up} are used to directly effect the neighborhood size, thus similarly adapting the group's communication but without the need for the two V thresholds. Hill Climbing and Gradient Learning algorithms were then used to further refine the weight values from this baseline and the thresholds used in the first framework as per Algorithms 1 and 2. In the first framework, these algorithms used a value of 150 for V_{init} and converged on threshold values of V for each of the three communication algorithms at 100 and 200. Thus, when W_{down} lowered V below 100, the implicit communication method was used, and when W_{up} raised V above 200, then the centralized method was used. Within the second framework the learned value for W_{down} and W_{up} controlled the radius of communication as a linear function based on V.

Hill Climbing algorithms have the advantage that they are intuitive for this and similar parameterization problems.[43] As is common within exploration heuristics, random perturbations for the values of V_{init}, W_{up}, and W_{down} are then evaluated.[41] 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 pseudo-code in Algorithm 1 describes this approach.

Algorithm 1: Hill Climbing

```
π ⇐ 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
        π ⇐ π_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

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

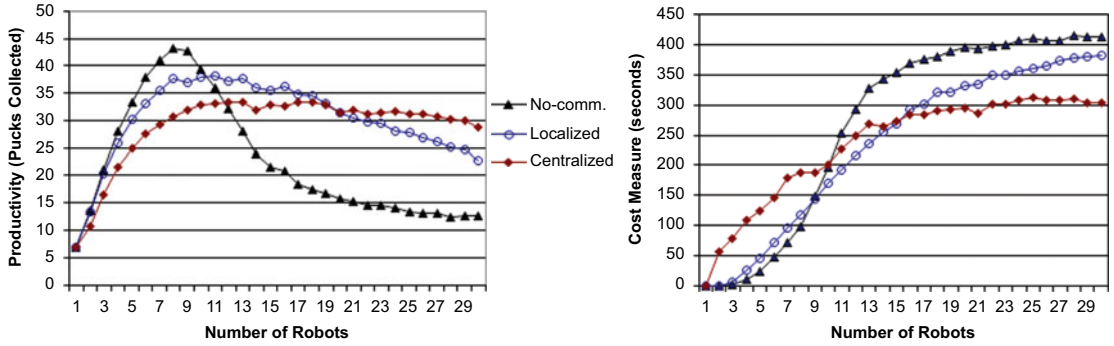


Fig. 2. Comparing levels of *time* spent on communication in different group sizes. Results averaged from 100 trials per datapoint.

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

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

to create a normalized gradient direction. The pseudo-code in Algorithm 2 describes this algorithm:

Algorithm 2: Gradient Learning

1: $\pi \Leftarrow$ Initial Policy (same as in approach no. 1)
2: **while** not done **do**
3: generate small variations for each parameter in the value of $\pm\ \varepsilon$ Specifically:
4: Generate an ϵ change (perturbation) in parameter V_{init}
5: Evaluate new V_{init} policy
6: Generate an ϵ change (perturbation) in parameter V_{down}
7: Evaluate new V_{down} policy
8: Generate an ϵ change (perturbation) in parameter V_{up}
9: Evaluate new V_{up} policy
10: 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
11: **end while**

As the next section details, these learning approaches were effective in creating both adaptive communication methods.

5. Experimental Results

The first set of experiments was designed to test the underlying hypothesis, that the CCC measure is in fact correlated to the productivity of the different groups. Our results from experiments involving time and energy costs do in fact support the claim that the best method of communication does change with domain conditions. Figure 2 contains the results from the time based coordination cost trials involving time costs. In the left side of the graph, the X-axis represents the group size, and the Y-axis the total number of pucks successfully retrieved within each group until the end of the 9 min allotted. 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 right side of Fig. 2.

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 collision resolution without communication. The localized group maintained near linear levels of its coordination cost with

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

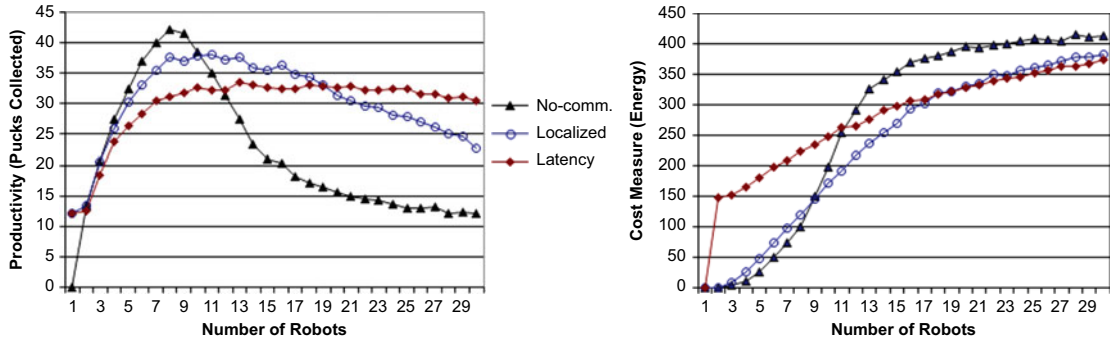


Fig. 3. Comparing levels of *energy* spent on communication in different group sizes. Results averaged from 100 trials per datapoint.

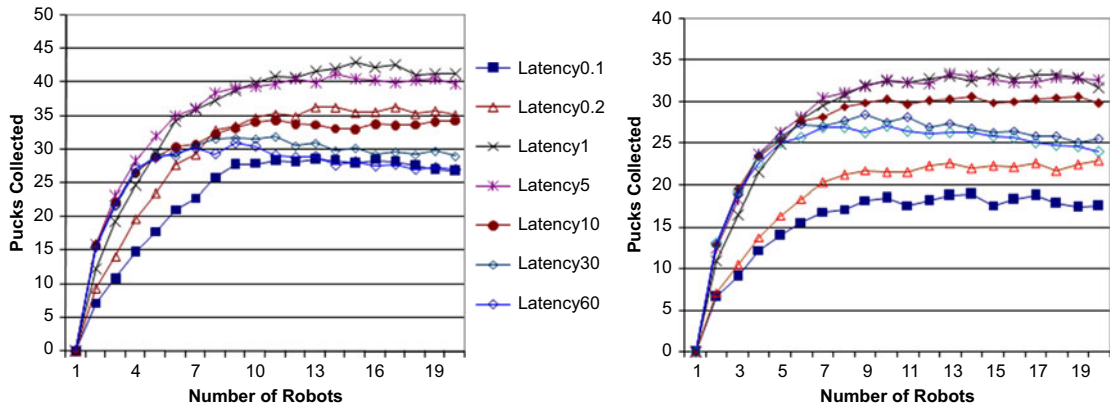


Fig. 4. Comparing latency differences and productivity levels for centralized method in time (left) and energy (right) experiments. Results averaged from 100 trials per datapoint.

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, the results of which are shown in Fig. 3. 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.

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 these methods affected the coordination cost, and thus the productivity outcome. For example, we studied seven latency variations within the centralized method in both experiment sets. These groups enforced maximal latency periods of L set to 0.1, 0.2, 1, 5, 10, 30, and 60 s. In the time-based experiments we found that a latency of 1 s often yielded average productivity level near 45 pucks. In the energy based experiments, a latency of 1 or 5s yielded similar results of an average productivity of less than 35 pucks (see Fig. 4). 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

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

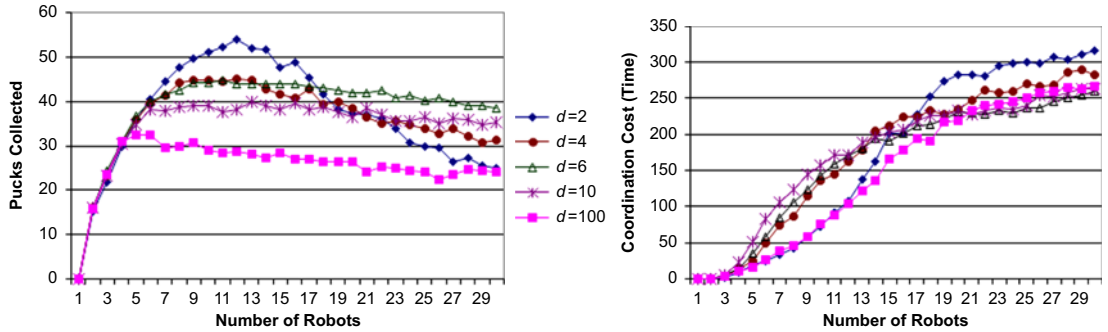


Fig. 5. The impact of varying neighborhood sizes (d) on productivity levels and costs in time experiments. Results averaged from 100 trials per datapoint.

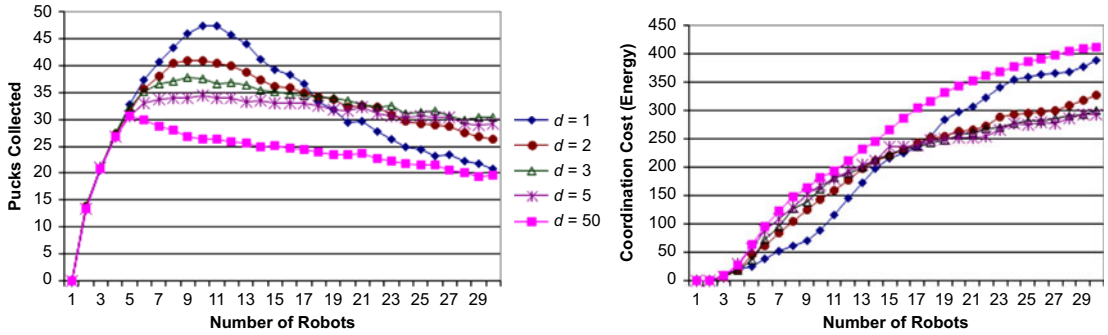


Fig. 6. The impact of varying neighborhood sizes (d) on productivity levels and costs in energy experiments. Results averaged from 100 trials per datapoint.

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, Fig. 4 displays our latency productivity variations in the time (displayed on the left) and energy trial sets (on right). We graphed the productivity levels (Y-axis) of the seven 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, in the time experiments, Latency0.1 (communication every 0.1 s) converged with Latency60 (communication once a minute). Latency0.1's frequent communication had its cost primarily due to communication, Latency60's infrequent communication often made the database of teammates' positions inaccurate. While the similarity between Latency0.1 and Latency60 is likely due to the implementation details of these robots (such as their speed), the overall relationship is domain independent. In general, an attempt to unwisely reduce communication, and this type of cost, led to an increase of repulsion behaviors, or a second-type coordination cost.

Just as no-communication scheme always fared best across all robotic group sizes, we found that no one neighborhood size always fared best. In studying this phenomenon we compared the productivity levels of foraging groups where we present results of d being 1, 2, 3, 5, and 50 robot lengths in the energy experiments (see Fig. 6) and 2, 4, 6, 10, and 100 in the time cost experiments (see Fig. 5). Recall that ε is approximately one robot length (1.5 radii). Thus Γ_1 represents the nearly localized variation with Γ_{50} and Γ_{100} corresponding to the nearly centralized versions of this method.

Figures 5 and 6 represent the relative productivity levels for these static neighborhood groups relative to the time and energy costs 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. 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_6 and Γ_5 resulted in the highest productivity in the

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

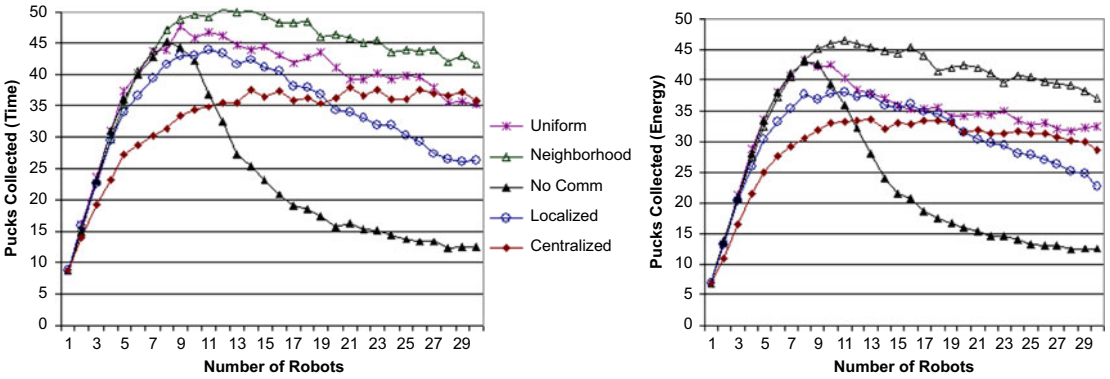


Fig. 7. Comparing adaptive communication methods based on time and energy costs to static methods. Results averaged from 100 trials per datapoint.

time (Fig. 4) and energy (Fig. 5) cost experiments, respectively. However, 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. Note from left side of Fig. 4 that the productivity level of the Γ_{100} method in the time experiments, which created too large a neighborhood, approached those of Γ_2, which did not create a large enough one. Similarly, note from the left side of Fig. 5 that the productivity level of the Γ_{50} method in the energy experiments, which created too large a neighborhood, approached those of Γ_1, which did not create a large enough one.

Based on the confirmed hypothesis that the cost measure is indeed correlated (negatively) with performance, the next set of experiments evaluated the performance of the two adaptive methods compared to the static methods on which they were based. Figure 7 shows the results from these experiments when considering both time costs (left side of the figure) and energy costs (right side of the figure). Notice that both adaptive approaches approximated or significantly exceeded the highest productivity levels of the static methods (no-communication, local, and centralized methods) they were based on, especially in medium to large groups. We attribute the success of both methods to their ability to change communication methods to the needs of the task. We believe that the neighborhood method outperformed the uniform one as it was allowed to create locally different neighborhood sizes, something none of the static neighborhood methods were capable of. This in turn facilitated better adaptation and higher productivity.

To evaluate the statistical significance of these results, we conducted the two tailed t-test and a one-factor ANOVA test comparing our adaptive groups and the three static groups they were based on. In all cases, in both time and energy categories, the null hypothesis p values were below 0.001. This provides support for the hypothesis that we can improve productivity through creating adaptive methods based on communication costs.

6. Conclusion

This work presents two novel adaptive communication frameworks for multi-robot collision avoidance. Both of these frameworks are based on our novel CCC measure to quantify all coordination costs, including those arising from communication. The advantage of using the CCC measure for adaptation is that it tractably quantifies the cost of coordination between robots allowing them to locally evaluate the relative effectiveness of different forms of communication. To the best of our knowledge, this paper is unique in that it demonstrates that robots can and should at times change between different types of communication schemes. One way this can be done is through uniform communication adaptation. In doing so, robots store a library of different communication schemes and utilize their locally observed CCC values to select the best choice from the options based on the task. We demonstrate how this can be done through switching between representative implementations of state-of-the-art algorithms ranging from those with no-communication, localized to global communication networks. A second way this can be done is through creating communications neighborhoods which represent a parameterized version of this selection with neighborhoods ranging from trivially small and thus approximating no-communication schemes to those the size of the domain, thus approximating global communication schemes. To demonstrate the effectiveness

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

of the CCC in quantifying all coordination costs and both adaptive communication frameworks, we performed thousands of simulated experiments with foraging robots in different group sizes. We show that the CCC measure is indeed strongly negatively correlated with productivity, even when considering communication costs related to time and energy. Furthermore, the CCC measure was highly effective in both adaptive frameworks, thus demonstrating the usefulness in implementing adaptive methods that switch between different communication approaches. In the canonical foraging task we studied, we found that the neighborhood adaptive method was more effective than the uniform communication approach.

References

1. G. Dudek, M. Jenkin and E. Milios, "A taxonomy for multi-agent robotics," *Robot Teams: From Diversity to Polymorphism* (T. Balch and L.E. Parker, eds.), vol. 3 (A K Peters, Natick, MA, 2002), pp. 3–22.
2. 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* (A K Peters/CRC Press, 2001) pp. 315–344.
3. M. Jäger and B. Nebel, "Decentralized Collision Avoidance, Deadlock Detection, and Deadlock Resolution for Multiple Mobile Robots," *IROS* (2001) pp. 1213–1219.
4. M. Hoy, A. S. Matveev and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey," *Robotica* **33**, 463–497 (2015).
5. M. Colby, J. Chung and K. Tumer, "Implicit Adaptive Multi-robot Coordination in Dynamic Environments," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2015)* (2015) pp. 5168–5173.
6. M. S. Couceiro, C. M. Figueiredo, D. Portugal, R. P. Rocha and N. M. F. Ferreira, "Initial Deployment of a Robotic Team - A Hierarchical Approach Under Communication Constraints Verified on Low-Cost Platforms," *IROS* (2012) pp. 4614–4619.
7. L. Liu, N. Michael and D. A. Shell, "Communication constrained task allocation with optimized local task swaps," *Auton. Rob.* **39**(3), 429–444 (2015).
8. I. Mezei, M. Lukic, V. Malbasa and I. Stojmenovic, "Auctions and iMesh based task assignment in wireless sensor and actuator networks," *Comput. Commun.* **36**(9), 979–987 (2013).
9. V. Nazarzehi and A. V. Savkin, "Distributed self-deployment of mobile wireless 3D robotic sensor networks for complete sensing coverage and forming specific shapes," *Robotica* **36**(1), 1–18 (2018).
10. S. Sen, M. Sekaran and J. Hale, "Learning to Coordinate without Sharing Information," *Proceedings of the Twelfth National Conference on Artificial Intelligence* (1994) pp. 426–431.
11. P. Stone and M. Veloso, "Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork," *Artif. Intell.* **110**(2), 241–273 (1999).
12. A. Tews, "Adaptive Multi-robot Coordination for Highly Dynamic Environments," *CIMCA* (2001).
13. Y. Yan and Y. Mostofi, "Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments," *IEEE Trans. Wireless Commun.* **12**(4), 1562–1572 (2013).
14. Y. Yan and Y. Mostofi, "To go or not to go on energy-aware and communication-aware robotic operation," *IEEE Trans. Control Netw. Syst.* **1**(3), 218–231 (2014).
15. K. Zhang, E. G. Collins, Jr. and D. Shi, "Centralized and distributed task allocation in multi-robot teams via a stochastic clustering auction," *ACM Trans. Auton. Adapt. Syst.* **7**(2), 21:1–21:22 (2012).
16. D. Goldberg and M. Matarić, "Interference as a Tool for Designing and Evaluating Multi-robot Controllers," *AAAI/IAAI* (1997) pp. 637–642.
17. A. Rosenfeld, G. Kaminka, S. Kraus and O. Shehory, "A study of mechanisms for improving robotic group performance," *Artif. Intell. J. (AIJ)* **172**(6–7), 633–655 (2008).
18. A. Rosenfeld, G. A. Kaminka and S. Kraus, "Adaptive Robot Coordination Using Interference Metrics," *Proceedings of the European Conference on Artificial Intelligence (ECAI)* (2004) pp. 910–916.
19. M. Anderson and N. Papanikolopoulos, "Implicit cooperation strategies for multi-robot search of unknown areas," *J. Intell. Rob. Syst.* **53**(4), 381–397 (2008).
20. G. A. Kaminka and R. Glick, "Towards Robust Multi-robot Formations," *ICRA-06* (2006) pp. 582–588.
21. R. A. Knepper, C. I. Mavrogiannis, J. Proft and C. Liang, "Implicit Communication in a Joint Action," *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction, HRI '17* (2017) pp. 283–292.
22. A. Knezevic, Q. Nguyen, J. A. Tran, P. Ghosh, P. Sakulkar, B. Krishnamachari and M. Annavaram, "CIRCE - A Runtime Scheduler for DAG-Based Dispersed Computing: DEMO," *SEC 2017* (2017) pp. 31:1–31:2.
23. M. J. Matarić, "Using Communication to Reduce Locality in Multi-robot Learning," *AAAI/IAAI* (1997) pp. 643–648.
24. A. Prorok, A. Bahr and A. Martinoli, "Low-Cost Collaborative Localization for Large-Scale Multi-robot Systems," *2012 IEEE International Conference On Robotics And Automation (ICRA)* (2012) pp. 4236–4241.
25. B. L. Wellman, S. Dawson and M. Anderson, "Multirobot Coverage Using Observation-Based Cooperation with Backtracking," *Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, FLAIRS* (2013).

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press

26. Z. Yan, N. Jouandeau and A. A. Cherif, "A survey and analysis of multi-robot coordination," *Int. J. Adv. Rob. Syst.* **10**, 1–18 (2013).
27. E. Yoshida, T. Arai, M. Yamamoto and J. Ota, "Local communication of multiple mobile robots: Design of optimal communication area for cooperative tasks," *J. Rob. Syst.* **15**(7), 407–427 (1998).
28. T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Trans. Rob. Autom.* **14**(6), 926–939 (1998).
29. D. V. Pynadath and M. Tambe, "The communicative multiagent team decision problem: Analyzing teamwork theories and models," *JAIR* **16**, 389–423 (2002).
30. 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," *Auton. Agents Multi-Agent Syst.* **9**(1), 87–143 (2004).
31. C. B. Excelente-Toledo and N. R. Jennings, "The dynamic selection of coordination mechanisms," *Auton. Agents Multi-Agent Syst.* **9**, 55–85 (2004).
32. A. Rosenfeld, A Study of Dynamic Coordination Mechanisms *Ph.D. Thesis* (Bar Ilan University, 2007).
33. G. Klien, D. D Woods, J. M. Bradshaw, R. R. Hoffman and P. J. Feltovich, "Ten challenges for making automation a "team player" in joint human-agent activity," *IEEE Intell. Syst.* **19**(6), 91–95 (2004).
34. W.-M. Shen, B. Salemi and P. Will, "Hormone-inspired adaptive communication and distributed control for CONTRO self-reconfigurable robots," *IEEE Trans. Rob. Autom.* **18**(5), 700–712 (2002).
35. K. Yoon and C. Hwang, *Multiple Attribute Decision Making: An Introduction* (Prentice Hall, Thousand Oaks: Sage, 1995).
36. M. Adler, R. K. Sitaraman and H. Venkataramani, "Algorithms for optimizing the bandwidth cost of content delivery," *Comput. Netw.* **55**(18), 4007–4020 (2011).
37. T. Balch, "Reward and Diversity in Multirobot Foraging," *IJCAI-99 Workshop on Agents Learning About, From and with Other Agents* (1999).
38. T. Balch, www.teambots.org (2000).
39. C. J. C. H. Watkins, Learning from Delayed Rewards *Ph.D. Dissertation* (Kings College, 1989).
40. D. Erusalimchik and G. A. Kaminka, *Towards Adaptive Multi-robot Coordination Based on Resource Expenditure Velocity* (IOS Press, Amsterdam, Netherlands, 2008).
41. R. S. Sutton, "Reinforcement Learning: Past, Present and Future," *SEAL* (1998) pp. 195–197.
42. R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.* **8**, 229–256 (1992).
43. N. Kohl and P. Stone, "Machine Learning for Fast Quadrupedal Locomotion," *AAAI* (2004) pp. 611–616.

https://doi.org/10.1017/S0263574718001364 Published online by Cambridge University Press