Communication Strategies for Cooperating Behavior-Based Robots

Maja J Matarić
Volen Center for Complex Systems
Computer Science Department
Brandeis University
Waltham, MA 02254
tel: (617) 736-2708 fax: (617) 736-2741
email: maja@cs.brandeis.edu

Abstract

This paper gives a brief overview of our experimental work using groups of robots using simple forms of communication to perform cooperative tasks. We describe three sets of robot experiments using simple communication strategies to augment limited sensing and effectors. In the first set, a group of robots used communication to learn social rules. In the second set, communication was used to compensate for limited sensing on individual robots in order to cooperate on pushing a large box. Finally, in the third set of experiments, the same two robots learned a simple communication protocol that enables them to push the box. In all three cases the communication strategies are implemented as simple behaviors easily incorporated into the robots' behavior–based controllers.

Introduction

Our work is motivated by the problem of learning through social interaction with one or more agents. The goal of the work described here was to study the role of communication in parallel learning of cooperative behaviors. This paper gives a brief overview of our experimental work using groups of robots using simple forms of communication to perform cooperative tasks.

We describe three sets of robot experiments using simple communication strategies to augment limited sensing and effectors. In the first set, a group of robots used communication to learn social rules in order to minimize interference within the group. In the second set, communication was used to compensate for limited sensing on individual robots in order to cooperate on pushing a large box. In the third set of experiments, the same two robots learned a simple communication protocol that enables them to coordinate the joint box–pushing process.

In all of the experimental domains the communication strategies were implemented as simple message–broadcasting and receiving behaviors that were easily incorporated into the robots' behavior–based controllers. The knowledge built into the behaviors, augmented with the information communicated between the robots, allowed for demonstrating robust coordinated group behaviors without the use of centralized or hierarchical control. The next three section describe each set of experiments in turn.

Learning Social Rules

Interference, any influence that opposes or blocks the agents' goal–driven behavior, is one of the main challenges in group and social environments. In societies consisting of agents with similar goals, interference manifests itself as competition for shared resources. In diverse societies, where agents' goals differ, more complex conflicts can persist between agents, including undoing of each other's work, deadlocks, and oscillations (Matarić 1994*a*).

Social rules that minimize interference among agents attempt to direct behavior away from individual greediness and toward global efficiency. Greedy individualist strategies perform poorly in group situations in which resource competition is inevitable and not correctly managed, since resource competition grows with the size of the group (Matarić 1994*b*). Our work has focused on tasks and solutions in which the agents cannot specialize, and instead must find means of optimizing their activity within the same task by developing social rules.

We addressed the problem learning of social rules by each of the individuals during their lifetime and within the context of a task, in this case foraging. In particular, we focused on learning *yielding* and *broadcasting*. Yielding eliminates spatial conflicts by imposing a temporal ordering on the agents' actions. Broadcasting spreads information from one to all other nearby agents. Together they serve to decrease interference and increase efficiency of the group.

The on–line learning process depended on the agents receiving sufficient information from the environment in order to distinguish productive and efficient behaviors, which would be positively reinforced, from unpro-

Figure 1: A typical environment state during the course of a learning experiment. Each robot is a wheeled base equipped with a fork, contact and infrared sensors, as well as radios for communication and positioning. The pucks are the food whose location the robots communicate to each other.

ductive, interfering ones, which would be negatively reinforced. Over time, the agents would learn the most efficient behavior for any social situation. We tested three types of reinforcement:

1. *individual* internally–motivated reinforcement based purely on the feedback the agent received directly,
2. *observational* externally–motivated reinforcement based on what the agents observed from other agents,
3. *communicated* message–motivated reinforcement based on what information was communicated by other agents involved in similar behaviors.

We postulated that all three forms of reinforcement were needed in order to learn social rules. To test this hypothesis, we designed a collection of experiments on mobile robots engaged in a foraging task within a confined space (see Figure 1). The experiments tested the feasibility and effectiveness of learning: 1) to yield when encountering another robot carrying a puck, and 2) to broadcast the location of any found pucks ("food").

The two learning tasks were attempted with different combinations of the three types of reinforcement described above. The results of the experiments, described in detail in Matarić (1994*b*) showed that all three types of reinforcement, including communication, were needed in order for the group to learn the two social behaviors in our foraging domain.

As predicted, performing certain social behaviors under particular conditions was difficult to learn. However, we found that broadcast communication was relatively easily learned compared to yielding and its dual, proceeding. Learning to broadcast was achieved by having the robot attempt, and compare the consequences of two different communication options. In each situation, the robot would either communicate its state (for example "found a puck at location (x,y)"), or not communicate at all. We did not test the option of false communications, i.e. of agents lying to each other. This does not mean, however, that all communicated information is necessarily "true." When a puck location is communicated, it may be incorrect due to errors in sensing on the part of the senders, as well as possible transmission errors. Furthermore, by the time a receiver reaches the advertised puck area, the puck may have already been taken by another robot. Finally, robots engaged in avoidance may move a puck arbitrarily far from its original announced location. All of these conditions occur with sufficient frequency that we cannot model inter–robot communication as completely accurate and reliable.

However, our environment was set up so that the optimal foraging strategy depended on all of the robots cooperating through communication, in order to receive maximum reward. Consequently, after sufficient trials, the robots learned to communicate the location of the found pucks. They also learned the complementary behavior of "listening" and paying attention to the broadcast information. Whenever any robot pursued an advertised puck location and found a puck there, it would receive internal reinforcement and quickly learn the association.

As noted, our society of robots was homogeneous and did not consist of agents that cheated or defected. However, the extension to such societies is simple as long as the robots are assigned identities and given the ability to recognize each other. The later can be achieved through communication as well and need not involve complex physical sensing such as vision. In such heterogeneous scenarios, the robots would learn to associate the identities of particular robots with their past behavior. The resulting social rules they learned would be more specific and thus more similar to some game–theoretic results in which the agents' strategies are determined by long–range interactions(Gasser & Huhns 1989, Rosenschein & Genesereth 1985, Axelrod 1984).

Cooperative Box–Pushing

Our second experimental environment involved a tightly–coupled cooperative task in which two six–legged robots would push a large box to the goal, indicated with a light. The robots we used were equipped with limited sensing, consisting of two whiskers and five pyroelectric sensors for determining the direction of the light, and simple effectors consisting of the 6 legs, the front two used both for walking and pushing the box.

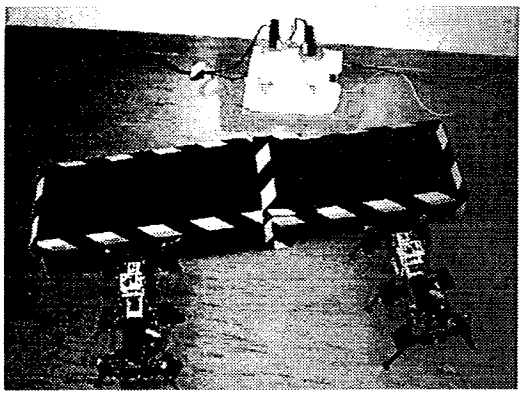

Figure 2: The box–pushing experimental environment. The lights indicate the location of the goal region. The box to be pushed is long enough to require force on both sides. It is also not so high as to obscure the input into the light sensors on the robots.

The robots could sense contact with the box but could not determine the exact global orientation of the box relative to the goal. Thus they could not independently decide how to move the box without using a slow strategy of frequently walking around the entire box. Furthermore, the robots were small enough relative to the box that they could not push it alone without requiring constant correction of course. Thus, the sensory and effector resources were distributed over the two robots, and careful coordination was required in order to effectively achieve the task.

We employed a simple communication and coordination strategy to achieve cooperation at three levels: sensing, action, and control (Matarić, Nilsson & Simsarian 1995). In order to guarantee coordination of both communication and action, we imposed a turn–taking protocol: only one robot moved and communicated at a time. The communication consisted of the robot sending its pyroelectric sensory data, i.e. its own local view of the goal. At each time–step, a robot would either merely communicate or listen and move.

The algorithm, shown below, resulted in robust box pushing which outperformed the single–robot alternative as well as strategies involving two non–communicating robots. The exact results are given in Matarić et al. (1995). Typical performance is illustrated in Figure 3. Communicating the sensory information effectively pooled the sensory resources of the two robots, so each time a robot took an action it had all of the available updated sensory data at its disposal. The turn–taking assured that the sensory data was updated when received; if both robots moved simultaneously, the received data could be incorrect if one of the robots moved after broadcasting it.

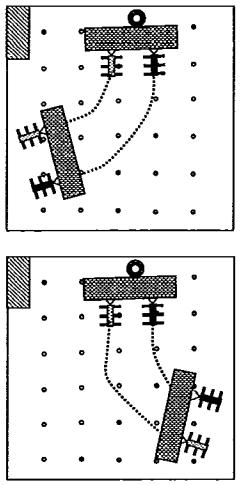

Figure 3: In this case the robots were synchronized and communicating, and were successful, in all cases, in pushing the box toward the goal by cooperatively performing the turning and walking behaviors. The average number of steps to reach the goal was 6.

```
Whenever "my-turn" message is received:
  get own sensory data s
  get other-robot's sensory data o
  combine s and o into current joint state j
  use j to select the best next action a from A
  use a to select the best message m from A
  send m
  perform a
  perform correcting
  send "your-turn" message

Whenever an action-message m is received:
  perform m
  perform correcting
```

The box–pushing experiments served as a demonstration of the effectiveness of simple communication in resource sharing. Communicating sensory information, in conjunction with turn–taking, allowed the robots to effectively push the box to the goal without interfering with each–others actions. This scenario also served as an effective substrate, for adding a learning algorithm, described in the next section.

Learning to Communicate in Cooperative Box–Pushing

The next set of experiments was an extension of the box–pushing scenario. In those, the robots were given

a set of basic behaviors, including pushing, turning, correcting, and stopping. Their goal was to learn how to move and what to communicate to the other robot, in order to push the box effectively to the goal.

The learning scenario differed from the hard–wired box–pushing scenario described in the previous section in that both robots could move in parallel. In both cases, however, one of the robots had control of the turn. In the learning scenario the robots used communication for two purposes: 1) to share sensory information, and 2) to tell each other what action to take. The goal of the learning system was to have both robots learn, in parallel, what to tell each other in each situation.

The job of the learning algorithm is to find the best strategy for concurrent actions for the two robots in order to move the box toward the goal. From the robot's view–point, the goal of the learning is to find the best action for itself and its partner. For example, turning the box to the left can be achieved by having the left robot stop and the right robot turn left. It can also be achieved more slowly by having the left robot turn left and the right robot keep walking.

The robots take turns in acting. At each turn, they select *my–action*, an action to perform themselves, and *its–action*, an action to communicate to the other robot to perform. The two robots perform concurrently for a fixed time period (in our case 11 seconds, an empirically–derived duration based on the robot's velocity). At the completion of the action, a reactive behavior is executed that realigns each of the robots with the box so that both of the whiskers are touching the box, i.e. so that they are perpendicular to it. This realignment is critical in order for the robots to perceive the orientation of the box relative to the light, and use the information obtained from the pyro sensors to compute the new state. The turn–taking protocol, constituting the basis of the control system for each of the robots, is shown below.

At each step, the new state is compared to the old, pre–move state, and the difference is used to compute reinforcement. The reinforcement depends on the side of the box a robot is on, since the ultimate goal is the keep the maximal light intensity on the right of the left robot's pyro array, and on the left, for the right robot's pyro array, thus centering the box relative to the light source.

```
Whenever "my-turn" message is received:
  get own sensory data S1
  get other-robot's sensory data O
  combine S and O into current joint state G
  use G to select the optimal box action B
  use B to select my optimal action A
  combine B and A to select msg-action M to send
  send M
  perform A
  align perpendicular to box
  get own sensory data S2
  use S1 and S2 to calculate reinforcement R1
  get other-robot's calculated reinforcement R2
  use R1 and R2 to update the robot-action matrix
  send "your-turn" message

Whenever an action-message M is received:
  get own sensory data S1
  perform M
  align perpendicular to box
  get own sensory data S2
  use S1 and S2 to calculate reinforcement R2
  get other-robot's calculated reinforcement R1
  use R1 and R2 to update the robot-action matrix
  send R2
```

We compared two different communication strategies. The first strategy used the sensory information of both robots to compute and update the reinforcement of only the robot that initiated the turn, as follows:

```
Compute Rs based on new pyro state
Update and normalize the matrix
```

The second strategy was "more cooperative" in that the robots learned in parallel, computing and exchanging reinforcement at each step, as follows:

```
Compute Rs based on the new pyro state
Get Ro from the other robot
Sum Rs and Ro received from the other robot
Update the matrix with the summed reinforcement
Transmit Rs to the other robot
```

The second strategy should learn faster than the first, twice as fast in the ideal case. However, the intricacies of real–world dynamics as well as the inevitable differences in sensor and effector characteristics between the robots can slow down parallel learning. The goal of our experiments was to test the prac-

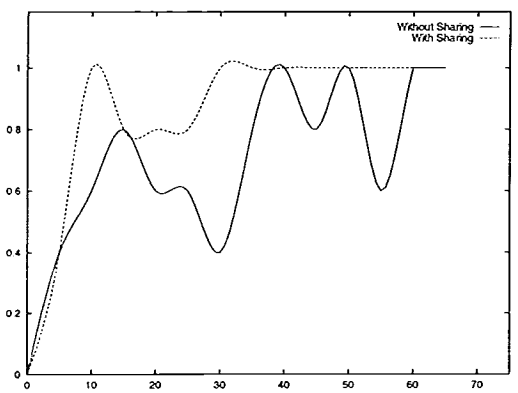

Figure 4: This graph compares the learning performance of two different reinforcement strategies. The solid line shows the non-shared strategy, while the dashed line indicates reinforcement sharing. Both are evaluated using the relaxed evaluation policy matrix. The x-axis plots contiguous trials over a 22-minute run. The y-axis plots the percentage of the learned policy.

tical its viability. Our experiments showed the desired and predicted performance. The strategy that shared reinforcement values and updated the learning matrix at each turn converged faster than the non-sharing case, in both cases using simple communication to perform the information-sharing. Figure 4 compares the performance of the two reinforcement strategies. The details of the described experiments can be found in (Simsarian & Matarić 1995).

Conclusions

This paper has briefly overviewed our experimental work with cooperating, communicating robots. We have demonstrated that simple communication strategies can be used to minimize inter-agent interference in social environments, to compensate for agents' limited sensing, and for establishing careful coordination between robots cooperating on a task.

In all cases we have described, the communication strategies consisted of broadcasting simple messages. As such, they incorporated directly into behavior-based controllers, and allowed for generating robust cooperative robot behaviors. The communication could be effectively wrapped into "sending" and "receiving" or "talking" and "listening" behaviors, each of which was getting information from a sensor (whether it be pyroelectric or radio) and using it to perform actions (whether it be pushing a box or updating a correlation matrix in a learning scheme).

The simplicity of the communication protocols was, in a sense, both sufficient and necessary in the described domains. The simple messages and message-passing strategies were all that was required for achieving even the careful coordination in the box-pushing and learning tasks. Furthermore, any more complex and time-extended protocols would have likely suffered from the noise and errors inherent in physical embodied systems.

We are currently exploring heterogeneous robot societies and studying how the described simple communication strategies scale with the increased complexity of the agent societies and their goals.

Acknowledgements

The box-pushing projects were performed at the Swedish Institute of Computer Science, in collaboration with Martin Nilsson and Kristian Simsarian.

References

Axelrod, R. (1984), *The Evolution of Cooperation*, Basic Books, NY.

Gasser, L. & Huhns, M. N. (1989), *Distributed Artificial Intelligence*, Pitman, London.

Matarić, M. J. (1994*a*), Interaction and Intelligent Behavior, Technical Report AI-TR-1495, MIT Artificial Intelligence Lab.

Matarić, M. J. (1994*b*), Learning to Behave Socially, *in* D. Cliff, P. Husbands, J.-A. Meyer & S. Wilson, eds, 'From Animals to Animats: International Conference on Simulation of Adaptive Behavior', pp. 453–462.

Matarić, M. J., Nilsson, M. & Simsarian, K. T. (1995), Cooperative Multi-Robot Box-Pushing, *in* 'Proceedings, IROS-95', Pittsburgh, PA.

Rosenschein, G. & Genesereth, M. R. (1985), Deals Among Rational Agents, *in* 'IJCAI-85', pp. 91–99.

Simsarian, K. T. & Matarić, M. J. (1995), Learning to Cooperate Using Two Six-Legged Mobile Robots, *in* 'Proceedings, Third European Workshop of Learning Robots', Heraklion, Crete, Greece.