2009

ANALYZING MYOPIC APPROACHES FOR MULTI-AGENT COMMUNICATION

RAPHEN BECKER
University of Massachusetts - Amherst

Computational Intelligence, Volume 25, Number 1, 2009

ANALYZING MYOPIC APPROACHES FOR MULTI-AGENT COMMUNICATION

RAPHEN BECKER, ALAN CARLIN, VICTOR LESSER, AND SHLOMO ZILBERSTEIN
Department of Computer Science, University of Massachusetts, Amherst

Choosing when to communicate is a fundamental problem in multi-agent systems. This problem becomes particularly challenging when communication is constrained and each agent has different partial information about the overall situation. We take a decision-theoretic approach to this problem that balances the benefits of communication against the costs. Although computing the exact value of communication is intractable, it can be estimated using a standard myopic assumption—that communication is only possible at the present time. We examine specific situations in which this assumption leads to poor performance and demonstrate an alternative approach that relaxes the assumption and improves performance. The results provide an effective method for value-driven communication policies in multi-agent systems.

Key words: multi-agent systems, decentralized MDPs, communication, decision-theoretic planning.

1. INTRODUCTION

Deciding when to communicate is a fundamental challenge in multi-agent systems. Communication enables agents to base their decisions on more complete knowledge of the overall situation. However, when communication incurs a cost, finding the optimal communication policy is usually intractable. Depending on the specific characteristics of the domain, the computational complexity of this problem ranges from NP-complete to NEXP-complete (Pynadath and Tambe 2002; Goldman and Zilberstein 2004). The main objective of this work is thus to develop cost-effective methods for deciding when to communicate in decentralized settings.

Because communication provides information to other agents, it is natural to measure the benefits of communication based on the value of the information. We take a decision-theoretic approach to this problem and define the value of communication as the net gain from communicating, which is the difference between the expected improvement in the agents' performance and the costs associated with communication. The optimal communication policy—which we define formally in the next section—involves the agents choosing the communicative act at each step that maximizes the expected future utility, much like choosing an optimal action in an MDP.

The approach we take builds on the formal notion of the value of information proposed by Howard (1966). Intuitively, the value of information is the expected increase in the value of the best plan as a result of obtaining the information. This framework has already been extended to evaluate processes that produce information such as alternative computations (Horvitz 1988; Russell and Wefald 1991). However, even in situations involving a single decision maker where the value of information theory has been extensively used, finding the exact value is very difficult. The typical approach to dealing with this complexity is to approximate the value of information using two common *myopic* assumptions (sometimes referred to as *myopic-greedy* assumptions): (1) each source of information is evaluated in isolation, and (2) a 1-step horizon is used in sequential decision making (Pearl 1988; Russell and Wefald 1991). In the context of *centralized* decision making, some useful nonmyopic approximation methods have been developed (Heckerman, Horvitz, and Midleton 1993). However, they have not yet been generalized to multi-agent settings.

Address correspondence to Shlomo Zilberstein, Department of Computer Science, 140 Governors Drive, University of Massachusetts, Amherst, MA 01003-9264; e-mail: shlomo@cs.umass.edu

A myopic approach has already been developed to derive communication policies in multi-agent systems (Tambe 1997; Gmytrasiewicz and Durfee 2001; Goldman and Zilberstein 2003). Frequently, however, the exact assumptions being made and their implications are not clearly stated. Additionally, a careful analysis of the impact of these assumptions on the quality of the resulting communication policies has not been made. While the myopic assumptions may be an appropriate way to approximate the value of information in the single-agent case, it is not obvious that they remain as effective in multi-agent settings.

This work is aimed at improving the understanding of communication in multi-agent systems by examining the implications of the myopic assumptions and proposing ways to overcome their deficiencies. First, we clearly state the basic myopic assumptions and formally show how to compute optimal communication policies given these assumptions. We then identify and describe two facets of the assumptions that introduce error, and provide an improved way to compute communication policies that compensates for this bias.

The analysis of communication is performed using the Decentralized MDP model with Communication (Dec-MDP-Com), which is simply a Dec-POMDP-Com (Goldman and Zilberstein 2003) with *joint* full observability. The Dec-POMDP-Com model, in turn, is equivalent to the COM-MTDP model (Pynadath and Tambe 2002) under the perfect recall assumption. These relationships among the models have been established by Seuken and Zilberstein (2008). The analysis in this work will furthermore assume transition independence (Becker et al. 2004) as well as observation independence. Observations of each agent will therefore correspond to that agent's local state.

We decided to study the control of communication in the context of decentralized MDPs for several reasons. First, decision-theoretic models provide a natural formal way to describe the problem in terms of maximizing expected utility. Decentralized MDPs in particular have been used widely in recent years within the multi-agent community (Xuan, Lesser, and Zilberstein 2001; Bernstein et al. 2002; Pynadath and Tambe 2002; Emery-Montemerlo et al. 2004; Goldman and Zilberstein 2004; Rabinovich, Goldman, and Rosenschein 2003; Roth, Simmons, and Veloso 2006; Seuken and Zilberstein 2008). Second, in this framework each agent has a separate local view of the world. The agents choose actions based on their own local views, without necessarily knowing the actions taken by the other agents (even if all the action selection policies are fixed and known). Centralized models, such as the MMDP (Boutilier 1999), do not distinguish between the private information available to each agent and the overall belief state. Finally, we have previously developed algorithms for finding optimal joint policies assuming no communication (Becker et al. 2004; Petrik 2007). These algorithms provide the foundations for this work and a baseline for the analysis of the benefits of communication.

The analysis we perform is separable into two components. The first component is an exact algorithm such as (Becker et al. 2004), which takes a Dec-MDP (with no communication) as input and returns an optimal joint policy for the Dec-MDP, along with the value of the joint policy. Then, this algorithm is used as a subroutine of the second component, which decides when to communicate.

An important aspect of the model we use is that it isolates the effect of communication on the expected value of a plan by distinguishing between domain-level actions and communicative acts. Other decision-theoretic multi-agent models allow domain-level actions to include implicit forms of communication (Bernstein et al. 2002; Pynadath and Tambe 2002; Goldman and Zilberstein 2003), which complicates the analysis of communication policies. Implicit communication occurs when one agent gains information about another agent's state through a noncommunicative act. This communication is often a byproduct of the agent's observations or the transition function, and is thus difficult to quantify. For example, when a robot attempts to move forward and fails, the failure could be caused by the wheels spinning

in place or by another robot sitting in front of it. Therefore, its failure to move forward changes its belief about the location of the other robot.

Several different aspects of communication in multi-agent systems have been studied in recent years (Stone and Veloso 1999; Shen, Lesser, and Carver 2003). Some researchers have managed to avoid the myopic assumptions, for example using reinforcement learning (RL) (Ghavamzadeh and Mahadevan 2004; Szer and Charpillet 2004). One advantage of using RL is that a complete model of the domain is not required. But using on-line learning in multi-agent settings could lead to poor performance, particularly in the early stages of learning. Convergence on local maxima presents another problem. Other researchers have addressed different questions, such as *what* the agents should communicate (Shen et al. 2003) instead of *when* to communicate. Xuan and Lesser (2002) have studied the use of communication as a way to reduce uncertainty. This work complements and builds on their approach by using the value of information to measure the benefits of reducing uncertainty.

2. PROBLEM DESCRIPTION

We examine the myopic approach for communications using transition-independent Dec-MDPs (Becker et al. 2004) enhanced with explicit communication. The model is composed of n cooperative agents. Each agent i works on its own local subproblem that is described by an MDP, $\langle S_i, A_i, P_i, R_i \rangle$. The local subproblem for agent i is completely independent of the local subproblems for the other agents, and is completely observable only by agent i. This means that at each step agent i takes action $a_i \in A_i$, transitions from state $s_i \in S_i$ to $s'_i \in S_i$ with probability $P_i(s'_i \mid s_i, a_i)$, and receives reward $R_i(s'_i)$. The *global* state of the domain is composed of the local states of all the agents.

At each time step, each agent first performs a domain-level action (one that affects its local MDP) and then a communication action. The communication actions are simply *communicate* or *not communicate*. If at least one agent chooses to communicate, then **every** agent broadcasts its local state to every other agent. This corresponds to the sync model of communication in Xuan et al. (2001), as it synchronizes the world view of the agents, providing each agent complete information about the current world state. The cost of communication is C if at least one agent initiates it, and it is treated as a negative reward. An optimal joint policy for this problem is composed of a local policy for each agent. Each local policy is a mapping from the current local state $s_i \in S_i$, the last synchronized world state $\langle s_1 \ldots s_n \rangle \in \langle S_1 \ldots S_n \rangle$, and the time T since the last synchronization to a domain-level action and a communication action, $\pi_i : S_i \times \langle S_1 \ldots S_n \rangle \times T \rightarrow A_i \times \{yes, no\}$. We will occasionally refer to domain-level policies and communication policies as separate entities, which are the mappings to A_i and $\{yes, no\}$, respectively.

In addition to the individual agents accruing rewards from their local subproblems, the system also receives reward based on the joint states of the agents. This is captured in the global reward function $R : S_1 \times \cdots S_n \rightarrow \Re$. To the extent that the global reward function depends on past history, the relevant information must be included in the local states of the agents just as with the local rewards. The goal is to find a joint policy $\langle \pi_1 \ldots \pi_n \rangle$ that maximizes the global value function V, which is the sum of the expected rewards from the local subproblems and the expected reward the system receives from the global reward function.

Definition 1. The global value function is

$$V(s_1 \ldots s_n) = \sum_{s'_1 \ldots s'_n} P(s'_1 .. s'_n \mid s_1 \ldots s_n, a_1 .. a_n) \left[\sum_{i=1}^{n} R_i(s'_i) + R(s'_1 \ldots s'_n) + V(s'_1 \ldots s'_n) \right]. \quad (1)$$

To summarize, the class of problems we study can be defined by n MDPs, a global reward function R, and synchronizing communication acts with a fixed cost C. Transitions on the MDPs are independent of each other; we will therefore assume $P(s'_1, \ldots s'_n \mid s_1 \ldots s_n, a_1 \ldots a_n) = \prod_{i=1}^{n} P_i(s'_i \mid s_i, a_i)$.

The complexity of of finding optimal policies for this class of problems has been shown to be NP-complete (Goldman and Zilberstein 2004), which is lower than the doubly exponential complexity (NEXP-hard) of general decentralized decision making. A key structure in the model that keeps the complexity in NP is the synchronizing communication protocol. When any information is transferred between the agents it is complete information so only the last communication must be memorized. Without this, the agents might have to remember the entire history of communication to make optimal decisions, which results in an exponential increase in the size of the policies and a double-exponential increase in solution time.

2.1. Sample Application

We illustrate this class of problems with the following multi-agent data collection example. This example can be viewed as an abstraction of many different types of data collection problems, though we consider autonomous rover coordination. Consider n rovers exploring a landscape and collecting data. Each rover has its own partially ordered list of sites it can visit, see Figure 1 (left-hand side). Each site is numbered as shown in the figure, and is of a particular class, A ,B, or C. The class is not known a priori. Instead, the rover has a distribution over the classes for each site. Figure 1 (right-hand side) represents the possible class of Site 1. For example, the site could be an interesting rock formation. With 70% probability it could be (A) a sedimentary rock, 25% (B) an igneous rock, and 5% (C) a fossil. The value of discovering and collecting data from a fossil may be significantly higher than collecting data from yet another sedimentary rock.

When a rover arrives at a site it has two choices. First, it can gather the information through a Detailed Analysis (DA) without knowing what class of information it is collecting. In this case, the rover proceeds directly to state A, B, or C (whichever is the true identity of the site), as shown in the figure, and also receives the appropriate reward. Alternatively, the rover can perform a Quick Analysis (QA) to determine the class of information available at

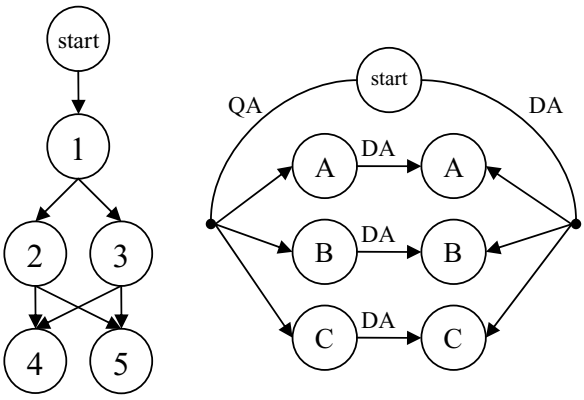

FIGURE 1. Graphical depiction of a sample decision problem. (Left-hand side): A partially ordered list of five sites. (Right-hand side): A decision problem for one site with three potential classes.

the site before choosing whether to collect the information. It will only receive the reward if it performs a DA next. The rover may not be able to collect information at all the sites due to limited resources, such as time and battery power.

The value of a DA comes from the information collected. The value of a QA is that it consumes fewer resources than a DA and allows the rover to make a more informed decision. The system receives reward based on the total information collected by all of the rovers. Each class of information has a base value. If the information in a particular class is redundant then the total value for collecting that class more than once may be only slightly higher than the base value. Alternatively, a class could be complementary, in which case the value for two pieces of information may be significantly greater than twice the base value. The values of the information are provided by the global reward function.

3. BASIC MYOPIC APPROACH

Using a myopic algorithm is a common way of dealing with the complexity inherent in finding an optimal solution. We start with a simple algorithm for determining when the agents should communicate. This algorithm is optimal assuming that communication must be initiated by the current agent (agent i in the following description) and that the current step is the only time step in which communication is possible. For clarity, the equations are presented for two agents i and j, but the approach easily extends to n agents. The complexity results still include all n agents.

While the problems we solve are distributed in nature (each agent chooses an action based on its own local view) the planning algorithm itself computes offline the policies for each agent in a centralized manner using a fully specified model of the problem. Then, the individual policies are given to the agents during execution time. This does not trivialize the problem, nor does it reduce it to a single MDP because the solution found is still executed in a decentralized manner. We chose this approach for two reasons. First, individual agents often lack the computational resources necessary to generate high-quality solutions. Second, individual agents often lack a global view of the problem. While the resulting communication policies are only conditioned on the local information available at run-time, there is no need to impose the same restriction on the off-line planning process.

The algorithm works as follows. As long as no communication is initiated, each agent follows the optimal policy assuming no future communication, which was obtained at planning time using a subroutine such as the Coverage Set Algorithm (CSA) (Becker et al. 2004) or a bilinear program (Petrik 2007). The subroutine takes a Dec-MDP with no communication as input, and provides joint-policies as well as their values as its output. At each state during execution time, agents choose whether to communicate or not by computing the net *value of communication* (VoC). If the VoC > 0, then the agent initiates communication causing all of the agents to broadcast their local states. This synchronizes the local views of all of the agents to the world state. The agents then compute a new optimal policy assuming no future communication, using their synchronized world state as the starting state. The domain-level actions the agents take always come from this zero-communication policy.

In the case where there are two agents i and j, the VoC from agent i's perspective depends on i's current local state s_i, the previous synchronized world state (or original starting state) $\langle s_i^0, s_j^0 \rangle$, and the time since the last synchronization t. It also implicitly depends on the optimal joint policy assuming zero communication that the agents have been following since the previous synchronization, $\langle \pi_i^0, \pi_j^0 \rangle$.

Definition 2. The Value of Communication (VoC) is the difference between the expected value when communicating and the expected value for remaining silent.

$$\text{VoC}\left(s_i, \langle s_i^0, s_j^0\rangle, t\right) = \sum_{s_j} P\left(s_j \mid s_j^0, t, \pi_j^0\right)\left[V^*(s_i, s_j) - \mathcal{C} - V(s_i, s_j)\right], \tag{2}$$

where $P(s_j \mid s_j^0, t, \pi_j^0)$ is agent i's belief about agent j's current local state, $V(s_i, s_j)$ is the expected value for following the current local policy, and $V^*(s_i, s_j) - \mathcal{C}$ is the expected value if the agents communicate now and follow a new zero communication policy after synchronizing.

The complexity of computing the VoC depends on the size of the local state space in this two agent case.

Theorem 1. Computing the Value of Communication can be done in time polynomial in the number of local states and exponential in the number of agents.

Proof. See Appendix A.

A final point about the complexity is the number of times VoC must be calculated to generate the joint communication policy. While the worst case appears to be quite large, $O(n\,|S|^{n+2})$, in practice it is not nearly that bad. The reason is that many of the combinations of variables are not reachable. For example, if communication is frequent, then the time since the last communication, t, will remain low. If communication is infrequent then the number of reachable synchronized world states $\langle s_i^0, s_j^0\rangle$ remains low because the world state is only synchronized through communicating. Additionally, there will be substantial overlap in computation between calls to VoC and caching can greatly reduce the running time in practice.

4. IMPLICATIONS OF THE MYOPIC ASSUMPTION

The myopic assumption allows a simple, straightforward computation of the value of communication. While this may be a reasonable assumption for the single agent case, there are additional implications that may not be readily apparent in a multi-agent setting. We examine these implications by identifying and analyzing two sources of error in the basic myopic approach, illustrating each with a simple example.

4.1. Modeling the Other Agents

Consider the situation of Alice and Bob, who are at home and want to cook a meal but lack ingredients. They form a plan for Alice to go to the store and purchase the ingredients, while Bob starts cooking the rest of the meal. In the improbable event that the store is out of ingredients, they will need to cook a different meal. Both are equipped with cell phones with inexpensive (but not free) calling plans. From Bob's *Basic* perspective, he should continuously call Alice after she leaves the house, to find out if her car has broken down, if the store is out of ingredients, if there is traffic on the road. But in real life, this is not necessary, because *Bob knows that if Alice had encountered a problem, she would have called*. It is this type of reasoning, neglected by the *Basic* approach, that we explore in this section.

The *Basic* myopic approach (Definition 2) assumes the simplest of models for the other agents—they never initiate communication. However, because every agent is following a

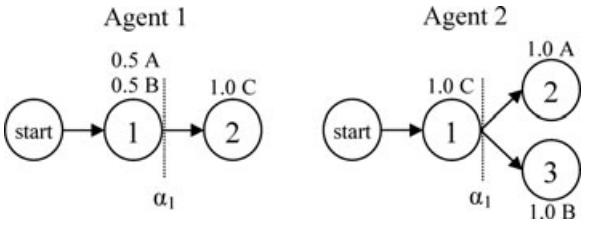

FIGURE 2. A simple example that illustrates how a simple model for the other agent introduces error.

communication policy based on computing the value of communication, this is an inaccurate model. One implication of a more accurate model of the other agents is that not communicating is in itself a form of communication. The distribution of states agent j can be in after t steps, $P(s_j \mid s_j^0, t, \pi_j^0)$, changes because j is known to not have passed through states in which it would have communicated.

Another implication is that at the current step, agent i may not need to initiate communication to acquire valuable information from agent j if j can be relied on to initiate communication when it has the information. Figure 2 illustrates this with a simple example where agent 1 collects information valuable to agent 2. At site 1, the figure shows that agent 1 has an equal chance of collecting an A or a B. If both agents collect A's or B's, suppose that the system receives a reward of 10 (not shown in the figure). Suppose that the system also receives a reward of 1 every time class C is collected. α_1 is the communication point of interest.

The initial zero-communication policy is for agent 2 to collect data from site 2. The only reason to communicate is if agent 1 collects a B at site 1, agent 2 needs to change its policy to go to site 3 rather than site 2. Based on the initial policy, 50% of the time the agents will receive the maximum reward of 12 and 50% of the time the minimum reward of 2. When agent 1 collects a B, it knows that the system will receive a reward of 12 with a probability of 1.0 if it communicates, and 2 with a probability of 1.0 if it does not. Therefore its VoC $= -\mathcal{C} + 1.0[12 - 2] = -\mathcal{C} + 10$. As long as the cost $\mathcal{C} < 10$, agent 1 will initiate communication. Agent 2 does not know what agent 1 has collected, so its VoC $= -\mathcal{C} + 0.5[12 - 12] + 0.5[12 - 2] = -\mathcal{C} + 5$. When the cost of communication $\mathcal{C} < 5$ agent 2 will communicate because its VoC > 0. Half of the time this communication is unnecessary because agent 1 had collected an A. When $\mathcal{C} \geq 5$ it is no longer valuable for agent 2 to initiate the communication and their communication policies are optimal.

The *Basic* line in Figure 3 shows the performance of the basic myopic strategy. As the cost of communication increases from 4.5 to 5, it exhibits a jump in value. This undesirable behavior is caused by error introduced into the VoC by not accounting for the other agent's communication policy. This error can be removed from the approximation by computing the best **joint** communication policy for each step (still assuming no future communication) instead of a **local** communication policy.

To compute the optimal joint communication policy for the current step, the agents must maximize the expected value over all possible world states they could be in. They do this by creating a table M with rows representing the possible states of agent 1 and columns representing states of agent 2 for the current step (see Figure 4).[1] The elements in the table are the value of communicating in that world state weighted by the probability that it is the current world state,

[1]This table does not correspond to the problem in Figures 2 and 3.

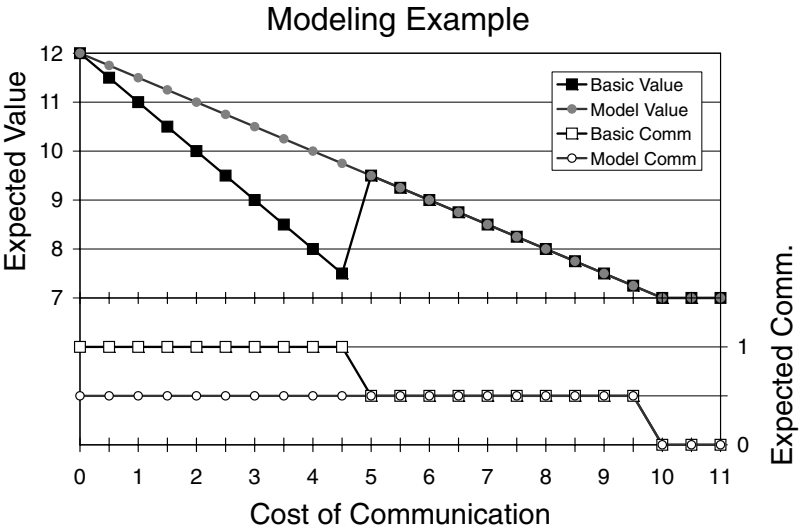

FIGURE 3. Performance comparison of the *Basic* and *Model* approaches.

	s_2^1	s_2^2	s_2^3	π_{1c}	VoC
s_1^1	**-1**	0	-1	*no*	-2
s_1^2	**4**	**-1**	**-1**	***yes***	**2**
s_1^3	**-2**	-1	1	*no*	-2
π_{2c}	***yes***	*no*	*no*		
VoC	**1**	-2	-1		

FIGURE 4. A Table M showing the expected gain in value for communicating for each world state.

$$M_{xy} = P\left(s_1^x \mid s_1^0, t, \pi_1^0\right) P\left(s_2^y \mid s_2^0, t, \pi_2^0\right)\left[V^*\left(s_1^x, s_2^y\right) - \mathcal{C} - V\left(s_1^x, s_2^y\right)\right]. \tag{3}$$

That is, each entry in the table corresponds to the VoC for a single joint state. As discussed in Section 3, this is the value of continuing along the current policy if the agents are in that joint state, subtracted from the value of the policy which the agent would follow after communication, and adjusted by the cost of communication. It should be clear that if agent 1 follows the *Basic* approach at a state s, it communicates if the sum of row s is positive, and likewise agent 2 communicates if its column is positive. We seek to evaluate whole communications policies across all states, not just the policy of one row or column. In Figure 4, Agent 1 has decided that it should communicate from s_1^2, because its VoC of 2 (the sum of its row) is positive, and Agent 2 has decided that it should communicate from state s_2^1, because its VoC of 1 is positive. VoC decisions are shown in the figure as π_{1c} and π_{2c}. In the figure, all joint-states that result in communication are bolded. This strategy double counts certain elements in the table and can result in choosing a communication policy worse than not communicating at all! The expected value of a joint communication policy for one step

is the sum of all entries in the table where communication happens (an entry is only counted once, even if both agents initiate communication). This corresponds to all joint-states where communication happens, weighted by their probability. In the example, the *Basic* policy given has a value of -1, computed by summing the bold entries. The reason for this negative value is because $M_{2,1}$ was counted twice for determining the policies (once for each policy), but only once for determining the value of the table. If agent 2 did not communicate in s_1 then the value would be 2. Never communicating ($\pi_{ic} = \{no, no\}$) will always have a value of 0.

The best joint communication policy is the joint policy that maximizes the bolded value of this table. This leads to a policy where agent 1 communicates in s_1^2 and agent 2 does not communicate. Finding the best joint policy is exponential in the size of the table, but a simple hill-climbing algorithm can find a Nash equilibrium in polynomial time. The line labeled *Model* in Figure 3 optimizes this table to eliminate the error, resulting in the best policy for this the example in Figure 2. Creating the table costs no more than the original approach because each entry represents a reachable world state.

The *Model* approach described in this section is not to be confused with Q-POMDP (Roth et al. 2005), which is a technique designed to account for uncertainty of belief state in a multi-agent POMDP. In Q-POMDP, each agent's environment is partially observable, and an agent will communicate when it deduces that communicating its state will change the action of the other agent, much like the *Basic* approach. It is enhanced to consider the true joint belief state in partially observable problems, but not the communication policy of the other agent. In the *Model* approach described above, each agent accounts for the state of the other agent as well as its communication policy.

4.2. Myopic View of the Future

The second facet of the myopic assumption is that no agent will communicate in the future. This approximates the true value of communication by introducing error in two ways. The first is due to the greedy nature of the algorithm. When communicating immediately has a positive value, VoC > 0, the agent communicates without considering whether the expected value would be even higher if it waited to communicate until a future step. To compensate, the agents can compute the value of (possibly) communicating after a 1-step delay:

$$\mathrm{VoC}_{delay}\big(s_i, \langle s_i^0, s_j^0\rangle, t\big) = \sum_{s_i'} P\big(s_i' \mid s_i, \pi_i^0\big) \times \max\big(0, \mathrm{VoC}\big(s_i', \langle s_i^0, s_j^0\rangle, t+1\big)\big).$$

The agent will initiate communication when its VoC $>$ VoC$_{delay}$ and VoC > 0. This does not imply that the agent really will initiate communication in the next step because the same condition will be reevaluated at that time with respect to later steps. As long as the expected value for delaying one step is greater than the value of communicating immediately, the agent will delay communication.

Figure 5 illustrates this with a simple example. If agent 1 collects A at site 1 then agent 2 should go to site 3, otherwise agent 2 should go to site 4. Similarly, with agent 2 collecting B at site 2. As with the previous example, two A's or two B's have a reward of 10, and each C adds a reward of 1. α_1 and α_2 are the two communication points. The *Basic* approach will always communicate at both α_1 and α_2 when the communication cost is low (see Figure 6). When the cost increases to 0.5, the agents will only communicate when they have valuable information. Agent 1 will initiate communication 50% of the time at α_1 and agent 2 will initiate 50% of the time at α_2, for a total expected communication of $0.5 + 0.5 = 1.0$. The *Delay* policy, however, recognizes that waiting a step is beneficial and will only communicate

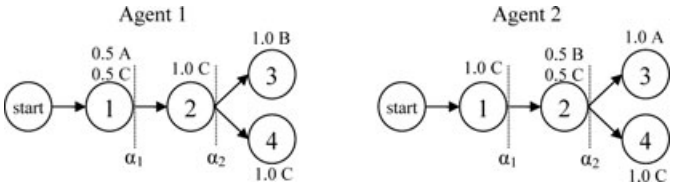

FIGURE 5. A simple example that illustrates how delaying communication can improve the expected value.

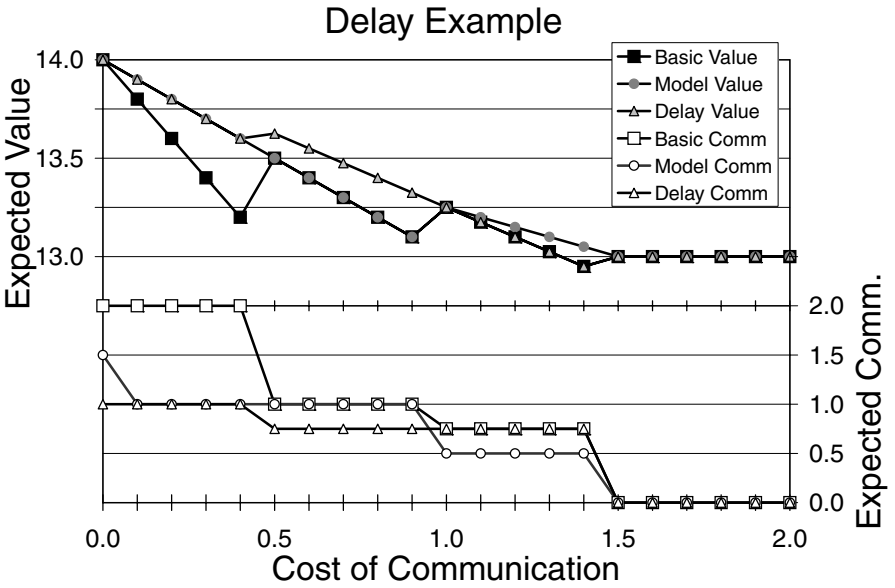

FIGURE 6. The expected value and expected amount of communication as a function of cost.

at α_2, which reduces the communication cost to .5 without decreasing the expected reward, yielding a higher expected value.

When the cost goes above 1, the *Model* approach realizes that it is more efficient to have only one agent initiate communication when it has valuable information. Specifically, the agents will initially plan to try for the joint reward, a change in plan will notify the other agent to collect *C*, which is worth 1.0. If communication cost is between 1.0 and 1.5, it can not be worth it for either agent to communicate at site 1, and at site 2, one agent will communicate 50% of the time while the other will always stay silent, resulting in communication 50% of the time. By contrast, under *Basic* or *Delay*, each agent will communicate if it has collected a *C*, resulting in communication 75% of the time.

This illustrates that the *Model* and *Delay* approaches address different sources of error and neither dominates the other.

A second source of error in the assumption of no future communication is built into the policies generated by the CSA. These policies may avoid situations which are valuable only when close coordination is possible. The optimal solution can exploit the possibility of future communication, while the domain-level policies generated here always assume no future communication. This source of error can also be partially compensated for by extending the 1-step delay to consider *h*-steps into the future.

5. MODEL-LOOKAHEAD APPROACH

This section demonstrates how the *Model* approach of 4.1 and the *Delay* approach of 4.2 can be merged together and extended to consider further steps into the future. The basic idea is an algorithm that makes optimal communication decisions within a lookahead horizon h given fixed domain-level policies based on zero communication. We call the merged approach *Model-Lookahead*.

To start, we introduce two new value functions. $V^h(s_i, s_j)$ is the expected value of not communicating in the current step, following an optimal communication policy for the next h steps, and then not communicating again after h steps. $V^{*h}(s_i, s_j) - \mathcal{C}$ is similar but starts with an immediate communication. When the lookahead horizon is 0 these value functions are equivalent to the single-step value functions from Definition 2, $V^0(\cdot) = V(\cdot)$, $V^{*0}(\cdot) = V^*(\cdot)$.

$$V^h(s_i, s_j) = \sum_{s_i', s_j' \in \text{Comm}} P\left(s_i' \mid s_i, \pi_i^0\right) P\left(s_j' \mid s_j, \pi_j^0\right) \left[\mathcal{R}(s_i', s_j') + V^{*h-1}(s_i', s_j') - \mathcal{C}\right] + \sum_{s_i', s_j' \in \neg\text{Comm}} P\left(s_i' \mid s_i, \pi_i^0\right) P\left(s_j' \mid s_j, \pi_j^0\right) \left[\mathcal{R}(s_i', s_j') + V^{h-1}(s_i', s_j')\right], \quad (4)$$

where $\mathcal{R}$ is the sum of the reward functions, $\mathcal{R}(s_i', s_j') = R_i(s_i') + R_j(s_j') + R(s_i', s_j')$. Comm is the set of states in which communication will take place. How it is computed becomes clear when we transform the equation as follows. The details of the derivation of Equations 4 and 5 can be found in Appendix A.

$$V^h(s_i, s_j) = V(s_i, s_j) + \sum_{s_i', s_j' \in \text{Comm}} P\left(s_i' \mid s_i, \pi_i^0\right) P\left(s_j' \mid s_j, \pi_j^0\right) \left[V^{*h-1}(s_i', s_j') - \mathcal{C} - V^{h-1}(s_i', s_j')\right] + \sum_{s_i', s_j'} P\left(s_i' \mid s_i, \pi_i^0\right) P\left(s_j' \mid s_j, \pi_j^0\right) \left[V^{h-1}(s_i', s_j') - V(s_i', s_j')\right]. \quad (5)$$

The agents must find the set of communication states for the next step that maximizes $V^h(s_i, s_j)$. The next step communication policy only affects the second line of Equation (5), which bears a remarkable similarity to Equation (3), except that this is a recursive function. Thus the same table algorithm can be applied to generate optimal communication policies over the lookahead horizon.

6. EXPERIMENTS

Figure 7 illustrates the performance of this approach on a larger problem with six time steps. The agents represent Mars rovers traversing sites and collecting data. State reflects the current site of the agent and data at that site, and battery life (from 0 to 8) remaining to the agent. The first agent's state and transition matrices correspond to Figure 1. Actions available are Move Left, Move Right, Wait, Quick Analysis, Detailed Analysis. The effects of Move Left and Move Right for the first agent can be seen on the left-hand side of the figure. The second agent simply moves from site 0 to sites 1,2, and 3 in a straight line. The right-hand side of the figure corresponds to the classes of data available at a specific site. There are five classes of possible data in all, A–E. Each site has a probability distribution over the classes

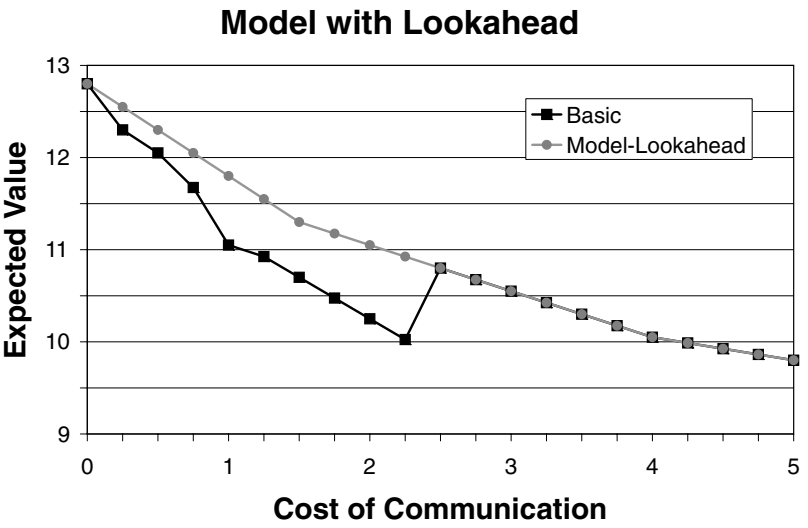

FIGURE 7. Performance of the *Model-Lookahead* Approach with lookahead horizon 2.

available. A Quick Analysis determines the class at a site, and a Detailed Analysis actually obtains the reward. Each agent starts with eight energy units on its battery. Movement costs one energy unit, as does a Quick Analysis. A Detailed Analysis costs two energy units. A reward of 10 is obtained for jointly obtaining classes A–D, while a reward of 1 is received for obtaining class E.

The *Model-Lookahead* approach performs significantly better than the original *Basic* approach and demonstrates a smooth and monotonic reduction of the expected value as the cost for communication increases.

Figure 8 shows the running time of *Model-Lookahead* compared to *Basic*. The *Basic* approach took about 11 seconds to generate the entire policy while *Model-Lookahead* took 50% longer with a lookahead horizon of 0 due to the added cost of finding the optimal communication policies of the tables. The worst case complexity of *Model-Lookahead* is exponential in the size of the lookahead horizon, but due to caching and the structure of the problem, in practice this is not always the case. In this example, the running time started out with an exponential curve but that changed as the lookahead horizon approached the number of steps in the problem.

To further test the generality of the approach, we ran experiments on a second domain, using a different methodology for constructing policies. Our goal was to prove robustness by demonstrating the use of VoC in conjunction with a second algorithm, besides CSA. We also changed some of the characteristics of the domain, allowing actions to vary in duration as well as in their effects.

The selection of the domain was motivated by mapping scenarios from NASA and the U.S. Geological Survey (Morris et al. 2008), whereby data from different imagers can be assimilated. Suppose our agents are sensors on separate satellites, which scan geographical locations on different bands. Data is most worthwhile if it gets scanned by both satellites at the same time. Actions available to the satellites are to Scan the current location or to Wait. Rewards can be both local and joint, for performing a scan. A joint reward is only received if the scan is initiated at the same time by both satellites. After a satellite is done scanning one location, it moves on to the next location. The time taken to perform a scan is a distribution.

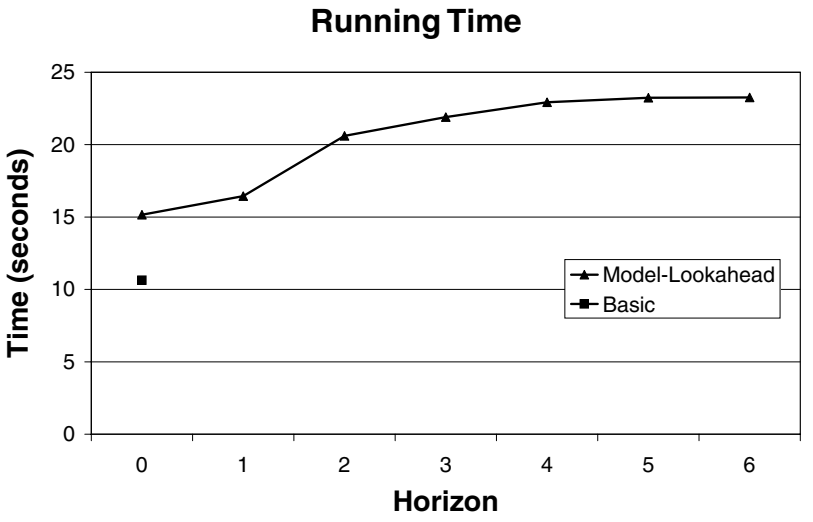

FIGURE 8. Comparison of the time to compute the policy for the *Basic* approach versus the *Model-Lookahead* approach of various depths.

In contrast to the previous examples, the uncertainty in this domain is with respect to time, rather than with respect to the type of data collected. Furthermore, instead of using CSA to solve the Dec-MDPs, we used a more recent and faster algorithm, which converts the Dec-MDP into a bilinear program and solves it (Petrik 2007). Through the use of bilinear programing, we could solve larger problems.

We first converted the state space in this example into a state space appropriate to the Value of Communication methodology. There are two types of states, the first type is when the satellite is at a location and can choose to scan or not to scan. This defines lh states where l is the number of locations, and h is the total time horizon of the problem. To make the domain appropriate for Value of Communication analysis, it is necessary that each agent have a defined state for each time step. To assure this, one can simply include additional states for the case when a satellite has initiated an scan and is waiting for it to finish. This is a tuple (s, f, l), where s is the current time, f is the time at which the action will be finished, and l is the location of the agent when the scan is finished. Combinations of these tuples introduce sfl new states. Thus, the total number of states is $lh + h^2l$.

In particular, we chose an example with $h = 8$ and $l = 4$. This defined 289 states for each agent, and 578 state/action pairs. We chose local rewards for the four sites to be .5, 5, 5, and 10, respectively. There was a shared joint reward of 20 if and only if the second site was explored by both rovers at the fifth time step. The duration of the scan of the first site would always be one step for the first agent, and a uniform distribution centered at four steps for the second agent. Successive scans by both agents would take a mean duration of 3 with a standard deviation of 1.6.

Results are shown in Figure 9. The figure shows that—as we observed in the Rovers domain—following the *Basic* communication strategy results in overcommunication. The key to this problem is that there is a large reward for completing all the scans, and an even larger reward for performing the valuable joint scan of the second site at step 5. The first satellite needs to choose between completing all the scans it can, versus waiting and attempting the joint scan.

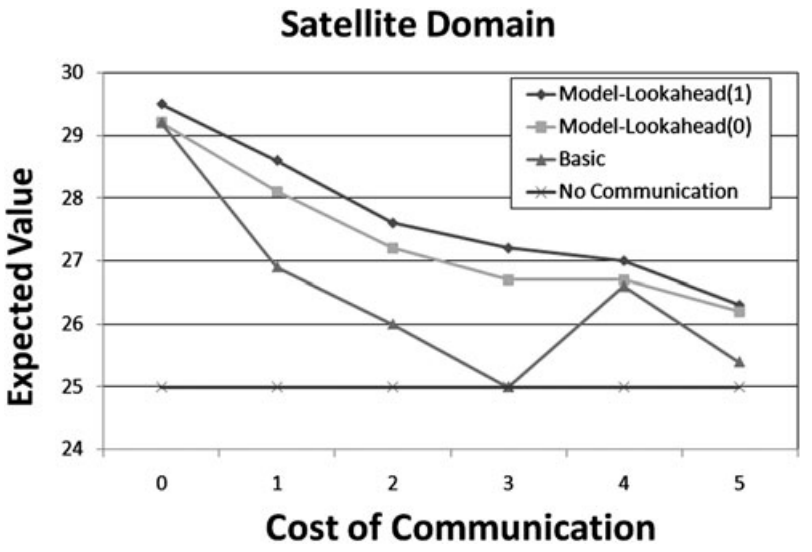

FIGURE 9. Results on satellite domain, showing the fixed value of no communication, compared with the values of the Basic strategy, Model with horizon 0, and Model with horizon 1.

Under the *Basic* strategy, the first satellite will overcommunicate after completing its first scan. The decision on whether communication is beneficial is mostly dependent on the second agent. If the second agent's scan terminates quickly, there will be time to synchronize for a second joint scan, and the agents should communicate to perform it. If it does not, then there is no need to communicate and synchronize. When the *Basic* policy is followed, the first satellite merely computes expected value of communication from its own perspective, without considering the policy of the other agent, and as a result overcommunication occurs, as described in the previous sections. The problem is corrected when the *Model Lookahead* policy with a delay is followed, which accounts for both the communication policy of the other agent as well as the ability to defer communication to future time steps. *Model Lookahead* consistently outperforms the *Basic* communication strategy, except when communication is either ubiquitous (at $Cost = 0$), or never useful.

To summarize, the *Model Lookahead* approach offers a simple but effective way to overcome the limitations of a naive myopic approach to communication. In two different domains it produced smooth and monotonous degradation of value as communication cost increases. This approach, however, does have its limitations. Even when the lookahead horizon is equal to the number of steps in the decision problem, the policy generated is not guaranteed to be an optimal joint policy. This is because the domain-level actions taken by the agents are generated assuming no future communication. Future work will focus on extending this algorithm to allow a larger domain-level action lookahead horizon.

7. CONCLUSION

We analyze the problem of choosing when to communicate in a multi-agent system. The conditions for communication are formulated based on the value of information. We show how a standard myopic approach leads to an efficient way to generate communication

policies, based on the assumption that communication is only possible at the present time. This extends previous work on myopic approximation of the value of information and the value of computation in single-agent settings.

We then examine the implications of the myopic assumption and show that it can lead to poor agent behavior. We identify two sources of error and provide modifications to the original algorithm to address these problems. Together, these modifications result in an improved algorithm for generating a decentralized joint policy. Moreover, the computational overhead of our modifications is small for a small horizon. Controlling the horizon presents a useful trade-off between solution quality and computation time.

While the sources of error that we identify and the general approach to addressing them are common to many multi-agent systems, the equations and specific algorithms we present do rely on certain structure being present in the problem. The key structure in the model is the synchronizing communication protocol. Without this, the agents might need to memorize the entire history of communication to operate optimally, which results in an exponential increase in the size of the policies and a double-exponential increase in solution time.

There are two components that together allow the use of synchronizing communication as an exact model. First is the fixed cost of communication. If the agents can send partial state information at a reduced cost then the optimal solution may include communication that does not synchronize the agents' view of the world. Second is the transition and observation independence between the domain-level actions. If the agents are able to take domain-level actions that affect the observations or transitions of another agent, then the agents have a form of implicit communication, which must be taken into account when considering state probabilities as well as belief states of the other agent. This generalization is beyond the scope of this paper.

Despite these assumptions, the overall paradigm of introducing and controlling communication based on the value of information is quite general. Identifying the sources of error common to many myopic approaches and showing how relatively simple modification could improve the performance of myopic approximations, will help design better communication algorithms for multi-agent systems.

ACKNOWLEDGMENTS

We thank Hala Mostafa and Marek Petrik for helpful discussions of this work. Marek also provided the code of the bilinear program for solving Dec-MDPs. This work was supported in part by the National Science Foundation under grants number IIS-0535061 and IIS-0812149, and by the Air Force Office of Scientific Research under grants number FA9550-05-1-0254 and FA9550-08-1-0181. Any opinions, findings, conclusions or recommendations expressed in this manuscript are those of the authors and do not reflect the views of the U.S. government.

REFERENCES

BECKER, R., S. ZILBERSTEIN, V. LESSER, and C. V. GOLDMAN. 2004. Solving transition independent decentralized Markov decision processes. Journal of Artificial Intelligence Research, **22**:423–455.

BERNSTEIN, D. S., R. GIVAN, N. IMMERMAN, and S. ZILBERSTEIN. 2002. The complexity of decentralized control of Markov decision processes. Mathematics of Operations Research, **27**(4):819–840.

BOUTILIER, C. 1999. Sequential optimality and coordination in multiagent systems. *In* Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, pp. 478–485.

EMERY-MONTEMERLO, R., G. GORDON, J. SCHNEIDER, and S. THRUN. 2004. Approximate solutions for partially observable stochastic games with common payoffs. *In* Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems. ACM Press, New York, pp. 136–143.

GHAVAMZADEH, M., and S. MAHADEVAN. 2004. Learning to communicate and act in cooperative multiagent systems using hierarchical reinforcement learning. *In* Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems. ACM Press, New York, pp. 1114–1121.

GMYTRASIEWICZ, P. J., and E. H. DURFEE. 2001. Rational communication in multi-agent environments. Journal of Autonomous Agents and Multi-Agent Systems, **4**(3):233–272.

GOLDMAN, C. V., and S. ZILBERSTEIN. 2003. Optimizing information exchange in cooperative multi-agent systems. *In* Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems, Melbourne, Australia. ACM Press, New York, pp. 137–144.

GOLDMAN, C. V., and S. ZILBERSTEIN. 2004. Decentralized control of cooperative systems: Categorization and complexity analysis. Journal of Artificial Intelligence Research, **22**:143–174.

HECKERMAN, D. E., E. J. HORVITZ, and B. MIDDLETON. 1993. An approximate nonmyopic computation for value of information. IEEE Transactions on Pattern Analysis and Machine Intelligence, **15**(3):292–298.

HORVITZ, E. J. 1988. Reasoning under varying and uncertain resource constraints. *In* Proceedings of the Seventh National Conference on Artificial Intelligence, Minneapolis, MN, Morgan Kaufmann, San Francisco, pp. 111–116.

HOWARD, R. A. 1966. Information value theory. IEEE Transactions on Systems Science and Cybernetics, **SSC–2**(1):22–26.

MORRIS, R. A., J. GASCH, L. KHATIB, and S. COVINGTON. 2008. Local search for optimal global map generation using mid-decadal landsat images. *In* Proceedings of the Twenty-First Conference on Artificial Intelligence, Chicago, Illinois, pp. 1706–1711.

PEARL, J. 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (2nd ed.). Morgan Kaufmann, San Francisco.

PETRIK, M., and S. ZILBERSTEIN. 2007. Anytime coordination using separable bilinear programs. *In* Proceedings of the Twenty-Second Conference on Artificial Intelligence, Vancouver, British Columbia, pp. 750–755.

PYNADATH, D. V., and M. TAMBE. 2002. The communicative multiagent team decision problem: Analyzing teamwork theories and models. Journal of Artificial Intelligence Research, **16**:389–423.

RABINOVICH, Z., C. V. GOLDMAN, and J. S. ROSENSCHEIN. 2003. The complexity of multiagent systems: The price of silence. *In* Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems, Melbourne, Australia, ACM Press, New York, pp. 1102–1103.

ROTH, M., R. SIMMONS, and M. VELOSO. 2005. Reasoning about joint beliefs for execution-time communication decisions. *In* Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems, Utrecht, the Netherlands, pp. 786–793.

ROTH, M., R. SIMMONS, and M. VELOSO. 2006. What to communicate? Execution-time decision in multi-agent POMDPs. *In* Proceedings of the Eighth International Symposium on Distributed Autonomous Robotic Systems (DARS), Minneapolis, MN.

RUSSELL, S., and E. WEFALD. 1991. Principles of metareasoning. Artificial Intelligence, **49**(1–3):361–395.

SEUKEN, S., and S. ZILBERSTEIN. 2008. Formal models and algorithms for decentralized decision making under uncertainty. Journal of Autonomous Agents and Multi-Agent Systems, **17**(2):190–250.

SHEN, J., V. LESSER, and N. CARVER. 2003. Minimizing communication cost in a distributed Bayesian network using a decentralized MDP. *In* Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems, Melbourne, Australia, ACM Press, New York, pp. 678–685.

STONE, P., and M. VELOSO. 1999. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. Artificial Intelligence, **110**(2):241–273.

SZER, D., and F. CHARPILLET. 2004. Improving coordination with communication in multi-agent reinforcement learning. *In* Proceedings of the Sixteenth IEEE International Conference on Tools with Artificial Intelligence, Boca Raton, Florida, ACM Press, New York, pp. 436–440.

TAMBE, M. 1997. Towards flexible teamwork. Journal of Artificial Intelligence Research, **7**:83–124.

XUAN, P., V. LESSER, and S. ZILBERSTEIN. 2001. Communication decisions in multi-agent cooperation: Model and experiments. *In* Proceedings of the Fifth International Conference on Autonomous Agents, Montreal, Canada, ACM Press, New York, pp. 616–623.

XUAN, P., and V. LESSER. 2002. Multi-agent policies: From centralized ones to decentralized ones. *In* Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy, ACM Press, New York, pp. 1098–1105.

APPENDIX A

This appendix provides the proof of Theorem 1.

Theorem 1. Computing the Value of Communication can be done in time polynomial in the number of local states and exponential in the number of agents.

Proof. There are four components to computing the VoC that add to the complexity:

- $P(s_j \mid s_j^0, t, \pi_j^0)$ is the t-step transition function for agent j. Given the assumption that j will never initiate communication,

$$P\left(s_j \mid s_j^0, t, \pi_j^0\right) = \sum_{s_j'} P\left(s_j' \mid s_j^0, t-1, \pi_j^0\right) P\left(s_j \mid s_j', \pi_j^0\right). \tag{A.1}$$

  This takes $O(|S_j|)$ if the values from $t-1$ were cached from a previous call to VoC and $O(|S_j|^2)$ to compute from scratch.
- $V(s_i, s_j)$ and $V^*(s_i, s_j)$ are both expected values (see Definition 1). The only difference is that they assume different domain-level policies. With dynamic programming they can be solved in time polynomial in the number of world states, which is exponential in the number of agents, $O(|S_i|^n)$.
- The difficult part of computing the VoC is finding the new optimal joint policy with no communication for the different possible world states. Note that the CSA does not need to be run in its entirety each time. Instead, most of the computation can be cached and only the final step of the algorithm must be rerun for each world state. That step involves searching through a small set of policies for each agent for the optimal joint policy. This step takes time exponential in the number of agents.
- When there are $n > 2$ agents, the summation in the VoC is over all possible local states of the other agents. The loop, therefore, must be repeated $O(|S_j|^{n-1})$ times. However, it is useful to note that $V^*(s_i, s_j) - V(s_i, s_j) \geq 0$ and therefore the summation can terminate as soon as it becomes greater than $\mathcal{C}$ instead of looping through all possible next states.

The net result is a complexity polynomial in the number of local states for the agents and exponential in the number of agents. ■

APPENDIX B

This appendix shows how Equation (5) was derived. This is shown for two agents, i and j. The current local states for the agents are s_i and s_j. We always use s_i' and s_j' for successor

states. The previous synchronized world state is $\langle s_i^0, s_j^0 \rangle$, which happened t steps earlier. When the agents communicate in s_i, s_j, the new synchronized world state becomes $\langle s_i, s_j \rangle$, and $t = 0$. The agents take domain-level actions based on an optimal policy assuming no future communication, $\langle \pi_i^{\langle s_i^0, s_j^0 \rangle}, \pi_j^{\langle s_i^0, s_j^0 \rangle} \rangle$. C is the cost for communicating. Comm′ is the set of world states in which the agents will communicate. We explain how this is computed at the end.

The global value function assuming no communication is:

$$V\left(s_i, s_j, \langle s_i^0, s_j^0 \rangle, t\right) = \sum_{s_i', s_j'} P\left(s_i' \mid s_i, \pi_i^{\langle s_i^0, s_j^0 \rangle}\right) P\left(s_j' \mid s_j, \pi_j^{\langle s_i^0, s_j^0 \rangle}\right) \times \left[R_i(s_i') + R_j(s_j') + R(s_i', s_j') + V\left(s_i', s_j', \langle s_i^0, s_j^0 \rangle, t+1\right)\right].$$

Note: The derivation is the same if you include a discount factor.

A superscript next to V represents the horizon in which communication is considered:

- $V^0(s_i, s_j, \langle s_i^0, s_j^0 \rangle, t)$ is the expected value of never communicating.
- $V^0(s_i, s_j, \langle s_i, s_j \rangle, 0) - C$ is the expected value of communicating immediately and never again.
- $V^1(s_i, s_j, \langle s_i^0, s_j^0 \rangle, t)$ is the expected value of not communicating in the current step, allowing communication if appropriate in the next step, and never communicating after the next step.
- $V^1(s_i, s_j, \langle s_i, s_j \rangle, 0) - C$ is the expected value of communicating immediately, allowing communication if appropriate in the next step, and never communicating after the next step.
- $V^h(s_i, s_j, \langle s_i^0, s_j^0 \rangle, t)$ is the expected value of not communicating in the current step, allowing communication where appropriate for the next h steps, and never communicating after that.
- $V^h(s_i, s_j, \langle s_i, s_j \rangle, 0) - C$ is the expected value of communicating immediately, allowing communication where appropriate for the next h steps, and never communicating after that.

Now, we give an inductive definition of the value allowing communication over a horizon. First is the base case, $h = 0$. $V^0(s_i, s_j, \langle s_i^0, s_j^0 \rangle, t) = V(s_i, s_j, \langle s_i^0, s_j^0 \rangle, t)$. This is just the global value function defined above.

Assume that $V^{h-1}(s_i, s_j, \langle s_i^0, s_j^0 \rangle, t)$ is the expected value of not communicating in the current step, allowing communication for the next $h-1$ steps, and never communicating after that. Also assume that $V^{h-1}(s_i, s_j, \langle s_i, s_j \rangle, 0) - C$ is the expected value of communicating immediately, allowing communication for the next $h-1$ steps, and never communicating after that.

To compute $V^h(\cdot)$ we divide the next possible world states into two categories, those in which the agents would choose to communicate, Comm′, and those in which they would not, ¬Comm′. For both cases, it is the sum of the probability that the agents transition to that world state times the immediate rewards plus the expected value allowing future communication up to the original horizon.

$$
\begin{aligned}
V^h\big(s_i, s_j, \langle s_i^0, s_j^0\rangle, t\big) = &\sum_{s_i', s_j' \in \text{Comm}'} P\Big(s_i' \,\Big|\, s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\Big) P\Big(s_j' \,\Big|\, s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\Big) \\
&\times \big[R_i(s_i') + R_j(s_j') + R(s_i', s_j') + V^{h-1}(s_i', s_j', \langle s_i', s_j'\rangle, 0) - C\big] \\
&+ \sum_{s_i', s_j' \in \neg\text{Comm}'} P\Big(s_i' \,\Big|\, s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\Big) P\Big(s_j' \,\Big|\, s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\Big) \\
&\times \big[R_i(s_i') + R_j(s_j') + R(s_i', s_j') + V^{h-1}\big(s_i', s_j', \langle s_i^0, s_j^0\rangle, t+1\big)\big].
\end{aligned}
$$

We now transform the equation as follows. First, we separate the rewards from the expected values:

$$
\begin{aligned}
= &\sum_{s_i', s_j' \in \text{Comm}'} P\Big(s_i' \,\Big|\, s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\Big) P\Big(s_j' \,\Big|\, s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\Big) [R_i(s_i') + R_j(s_j') + R(s_i', s_j')] \\
&+ \sum_{s_i', s_j' \in \text{Comm}'} P\Big(s_i' \,\Big|\, s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\Big) P\Big(s_j' \,\Big|\, s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\Big) \big[V^{h-1}(s_i', s_j', \langle s_i', s_j'\rangle, 0) - C\big] \\
&+ \sum_{s_i', s_j' \in \neg\text{Comm}'} P\Big(s_i' \,\Big|\, s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\Big) P\Big(s_j' \,\Big|\, s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\Big) [R_i(s_i') + R_j(s_j') + R(s_i', s_j')] \\
&+ \sum_{s_i', s_j' \in \neg\text{Comm}'} P\Big(s_i' \,\Big|\, s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\Big) P\Big(s_j' \,\Big|\, s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\Big) \big[V^{h-1}\big(s_i', s_j', \langle s_i^0, s_j^0\rangle, t+1\big)\big].
\end{aligned}
$$

Then, we combine the rewards and add/subtract two new components, (B.4)/(B.5) and (B.6)/(B.7):

$$
\begin{aligned}
= &\sum_{s_i', s_j'} P\Big(s_i' \,\Big|\, s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\Big) P\Big(s_j' \,\Big|\, s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\Big) [R_i(s_i') + R_j(s_j') + R(s_i', s_j')] && \text{(B.1)} \\
&+ \sum_{s_i', s_j' \in \text{Comm}'} P\Big(s_i' \,\Big|\, s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\Big) P\Big(s_j' \,\Big|\, s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\Big) \big[V^{h-1}(s_i', s_j', \langle s_i', s_j'\rangle, 0) - C\big] && \text{(B.2)} \\
&+ \sum_{s_i', s_j' \in \neg\text{Comm}'} P\Big(s_i' \,\Big|\, s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\Big) P\Big(s_j' \,\Big|\, s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\Big) \big[V^{h-1}\big(s_i', s_j', \langle s_i^0, s_j^0\rangle, t+1\big)\big] && \text{(B.3)} \\
&+ \sum_{s_i', s_j'} P\Big(s_i' \,\Big|\, s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\Big) P\Big(s_j' \,\Big|\, s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\Big) \big[V\big(s_i', s_j', \langle s_i^0, s_j^0\rangle, t+1\big)\big] && \text{(B.4)} \\
&- \sum_{s_i', s_j'} P\Big(s_i' \,\Big|\, s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\Big) P\Big(s_j' \,\Big|\, s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\Big) \big[V\big(s_i', s_j', \langle s_i^0, s_j^0\rangle, t+1\big)\big] && \text{(B.5)} \\
&+ \sum_{s_i', s_j' \in \text{Comm}'} P\Big(s_i' \,\Big|\, s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\Big) P\Big(s_j' \,\Big|\, s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\Big) \big[V^{h-1}\big(s_i', s_j', \langle s_i^0, s_j^0\rangle, t+1\big)\big] && \text{(B.6)} \\
&- \sum_{s_i', s_j' \in \text{Comm}'} P\Big(s_i' \,\Big|\, s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\Big) P\Big(s_j' \,\Big|\, s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\Big) \big[V^{h-1}\big(s_i', s_j', \langle s_i^0, s_j^0\rangle, t+1\big)\big]. && \text{(B.7)}
\end{aligned}
$$

Next, we combine (B.1) with (B.4), (B.2) with (B.7), and (B.3) with (B.6) and then with (B.5):

$$
\begin{aligned}
&= \sum_{s_i', s_j'} P\left(s_i' \mid s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\right) P\left(s_j' \mid s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\right) \\
&\quad \times \left[R_i(s_i') + R_j(s_j') + R(s_i', s_j') + V\left(s_i', s_j', \left\langle s_i^0, s_j^0\right\rangle, t+1\right)\right] \\
&\quad + \sum_{s_i', s_j' \in \text{Comm}'} P\left(s_i' \mid s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\right) P\left(s_j' \mid s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\right) \\
&\quad \times \left[V^{h-1}(s_i', s_j', \langle s_i', s_j'\rangle, 0) - C - V^{h-1}\left(s_i', s_j', \left\langle s_i^0, s_j^0\right\rangle, t+1\right)\right] \\
&\quad + \sum_{s_i', s_j'} P\left(s_i' \mid s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\right) P\left(s_j' \mid s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\right) \\
&\quad \times \left[V^{h-1}\left(s_i', s_j', \left\langle s_i^0, s_j^0\right\rangle, t+1\right) - V\left(s_i', s_j', \left\langle s_i^0, s_j^0\right\rangle, t+1\right)\right] \qquad \text{(B.8)}
\end{aligned}
$$

Line (B.8) is simply the expected value with zero communication.

$$
\begin{aligned}
&= V\left(s_i, s_j, \left\langle s_i^0, s_j^0\right\rangle, t\right) \qquad \text{(B.9)} \\
&\quad + \sum_{s_i', s_j' \in \text{Comm}'} P\left(s_i' \mid s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\right) P\left(s_j' \mid s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\right) \\
&\quad \times \left[V^{h-1}(s_i', s_j', \langle s_i', s_j'\rangle, 0) - C - V^{h-1}\left(s_i', s_j', \left\langle s_i^0, s_j^0\right\rangle, t+1\right)\right] \qquad \text{(B.10)} \\
&\quad + \sum_{s_i', s_j'} P\left(s_i' \mid s_i, \pi_i^{\langle s_i^0, s_j^0\rangle}\right) P\left(s_j' \mid s_j, \pi_j^{\langle s_i^0, s_j^0\rangle}\right) \\
&\quad \times \left[V^{h-1}\left(s_i', s_j', \left\langle s_i^0, s_j^0\right\rangle, t+1\right) - V\left(s_i', s_j', \left\langle s_i^0, s_j^0\right\rangle, t+1\right)\right] \qquad \text{(B.11)}
\end{aligned}
$$

This is Equation (5). We represent the equation in this way because it is much easier to use. This transformed equation also demonstrates how to compute the communication policy for the next step. We want to find a joint communication policy for the next step that maximizes this value function. Lines (B.9) and (B.11) do not depend on the communication policy for the next step, so we just need to maximize line (B.10). This is what the *Model* approach does.