A POMDP-based Model for Optimizing Communication in Multiagent Systems

Francisco S. Melo[1] and Matthijs T.J. Spaan[2]

[1] INESC-ID/Instituto Superior Técnico
Porto Salvo, Portugal
fmelo@inesc-id.pt
[2] Delft University of Technology
Delft, The Netherlands
m.t.j.spaan@tudelft.nl

Abstract. In this paper we address the problem of planning in multiagent systems in which the interaction between the different agents is sparse and mediated by communication. We include the process of communication explicitly as part of the decision process and illustrate how this single-agent model can be used to plan for communication. We also use the single-agent model to plan in the multiagent scenario, exploiting the sparse interaction between the agents. Our results show that our approach can be used for efficient and effective planning: without incurring the computational complexity of more elaborate multiagent models (such as Dec-MDPs or Comm-MTDPs), we are able to attain good performance in several test domains while planning for communication.

1 Introduction

Decentralized partially observable Markov decision processes (Dec-POMDPs) provide powerful modeling tools for multiagent decision-making in face of uncertainty. However, the prohibitive computational cost required to compute an optimal decision-rule for this class of models renders them impractical except for the smallest of problems.[3] This has motivated simplifications of general Dec-POMDP models that aim at capturing some of the fundamental features of this class of problems (such as partial observability) while alleviating its associated computational cost.

In this paper we are interested in a recent line of work that exploits simplified models of interaction among the agents in a Dec-POMDP. In many real-world multiagent scenarios, one observes that the tasks of the different agents are not coupled at every decision-step but only in relatively infrequent situations. In the Dec-POMDP literature, early approaches introduced the idea of transition and reward independence [3] as forms of simplified interactions. Independence as well as communication have been shown to greatly reduce the computational complexity of solving decentralized decision models such as Dec-MDPs and Dec-POMDPs [4,1].

[3] Dec-MDPs are known to be NEXP-complete even in 2-agent scenarios.

However, while many multiagent scenarios are not coupled at every decision step, it is also seldom the case that transitions (or even rewards) are completely independent. Several models have been proposed that, while leveraging transition and reward independence between agents, still allow for some "local" dependence in both transitions and rewards. Examples include interaction-driven Markov games [13], distributed POMDPs with coordination locales [14], influence-based policy abstractions in weakly-coupled Dec-POMDPs [15], and models relying on event-driven interactions [2].

As in the aforementioned works, in this paper we are also interested in scenarios where the interactions among agents are sparse and localized. However, our focus in this paper is not so much on how to exploit sparse interactions, but rather on how sparse interactions may impact the communication needs in multiagent planning.

Multi-robot systems constitute the primary motivation for our work and provide a natural example of the class of problems considered herein. In multi-robot systems, interaction among robots is naturally limited by the robot's physical boundaries (workspace, communication range, etc.) and limited perception capabilities. It is therefore natural to subdivide the overall task into smaller tasks that each robot can execute either autonomously or as part of a small group. Moreover, besides being embedded in a physical environment, robots typically have a way of communicating among themselves. Communication capabilities can mitigate issues of partial observability, as they allow agents to share useful information such as sensor readings.

Explicit communication in multiagent planning was already addressed in [9], where the proposed Com-MTDP model allows to explicitly reason about communication in Dec-POMDP-like scenarios. However, being a generalization of Dec-POMDPs, it shares the discouraging computational complexity of the latter model. The actual process of communication has been investigated in [5]. Roth et al. [10] propose to exploit a factored Dec-MDP model and policy representation, in which agents query other agents' local states when this knowledge is required for choosing their local actions. Another closely related work is that of Wu et al. [16] where communication is used as a means to decrease the planning complexity in Dec-POMDP models.

In this work, we consider a Dec-POMDP model in which agents need to plan about when to query other agents' local observations. Our approach is distinct from those surveyed above in several ways. First of all, unlike [10,16], we explicitly plan for communication, considering the associated cost-benefit tradeoff. Furthermore, unlike [10], we query observations and not states. Moreover, we do not assume to have available an *a-priori* centralized policy, which is very hard to obtain in the multiagent POMDP case. Instead, we optimize each agent's local policy in a round-robin fashion, which is much more scalable.

Our representation of interactions is closest to interaction-driven Markov games (IDMGs) [13]. This model leverages the independence between the different agents in a Dec-POMDP to decouple the decision process in significant portions of the joint state space. In those situations in which the agents interact,

IDMGs rely on communication to bring down the the computational complexity of the joint decision process. The use of communication to overcome partial observability differentiates this approach from other approaches that also exploit local interactions among the agents. However, Spaan & Melo [13] assume communication to *always* take place and to be error-free. In our case, we add explicit query actions to each agent's action repertoire, enabling it to query another agent's observation, subject to certain constraints. For instance, two robots may only be able to share information when they are physically close. Also, we assume that communication is subject to errors and comes at a cost that must be considered.

2 Background

We start by reviewing *decentralized partially observable Markov decision processes* (Dec-POMDPs) and related decision theoretic models. An N-agent Dec-POMDP $\mathcal{M}$ is specified as a tuple $\mathcal{M} = (N, \mathcal{X}, (\mathcal{A}_k), (\mathcal{Z}_k), \mathsf{P}, (\mathsf{O}_k), r, \gamma)$, where $\mathcal{X}$ is the joint state-space; $\mathcal{A} = \times_{i=1}^{N} \mathcal{A}_i$ is the set of joint actions, with each $\mathcal{A}_i$ the individual action set for agent $i, i = 1, \ldots, N$; each $\mathcal{Z}_i$ represents the set of possible local observation for agent $i, i = 1, \ldots, N$; $\mathsf{P}(y \mid x, a)$ represents the transition probabilities from joint state x to joint state y when the joint action a is taken; each $\mathsf{O}_i(z_i \mid x, a)$ represents the probability of agent i making the local observation z_i when the joint state is x and the last joint action taken was a, and $r(x, a)$ represents the expected reward received by all agents for taking the joint action a in joint state x. The scalar γ is a discount factor.

An *N-agent Decentralized Markov decision process* (Dec-MDP) is a particular class of Dec-POMDP in which the state is *jointly fully observable*. Formally this can be translated into the following condition: for every joint observation $z \in \mathcal{Z}$, with $\mathcal{Z} = \times_{i=1}^{N} \mathcal{Z}_i$, there is a state $x \in \mathcal{X}$ such that $\mathbb{P}\left[X(t) = x \mid Z(t) = z\right] = 1$, where $X(t)$ is the joint state of the process at time t and $Z(t)$ the corresponding joint observation. Similarly, a *partially observable Markov decision process* (POMDP) is a 1-agent Dec-POMDP and a *Markov decision process* (MDP) is a 1-agent Dec-MDP. Finally, an *N-agent multiagent MDP* (MMDP) is an N-agent Dec-MDP that is *fully observable*, *i.e.*, for every individual observation $z_i \in \mathcal{Z}_i$ there is a state $x \in \mathcal{X}$ such that $\mathbb{P}\left[X(t) = x \mid Z_i(t) = z_i\right] = 1$.

In this partially observable multiagent setting, an individual (non-Markov) policy for agent i is a mapping $\pi_i : \mathcal{H}_i \longrightarrow \Delta(\mathcal{A}_i)$, where $\Delta(\mathcal{A}_i)$ is the space of probability distributions over $\mathcal{A}_i$, and $\mathcal{H}_i$ is the set of all possible finite histories for agent i. The purpose of all agents is to determine a joint policy π that maximizes the total sum of discounted rewards. In other words, considering a distinguished initial state $x^0 \in \mathcal{X}$ that is assumed common knowledge among all agents, the goal of the agents is to maximize

$$V^{\pi} = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r\big(X(t), A(t)\big) \mid X(0) = x^0\right]. \tag{1}$$

For a more detailed introduction to Dec-POMDPs and related models see, e.g., [11].

3 A Model for Observation Querying

We depart from an N-agent Dec-POMDP model, and address the problem of when communication can be beneficial to improve the performance in such a model. Unlike other communication-based approaches to Dec-POMDPs (e.g., [10]), we adopt a relatively general communication model, in which the messages exchanged between the agents are taken as part of the (noisy) observations available and depend on explicit information- querying actions by them. For the purposes of our study, we will ignore the decision process of all except one agent, which we refer as agent k.

We represent the (finite) state-space of the Dec-POMDP as a set $\mathcal{X}$ and assume that it can be factorized as $\mathcal{X} = \mathcal{X}_k \times \mathcal{X}_{-k}$, where the elements $x_k \in \mathcal{X}_k$ correspond to agent k's local state. The state at time t, $X(t)$, is thus a pair $\langle X_k(t), X_{-k}(t)\rangle$.

We further assume that all remaining agents follow a Markov policy π_{-k} which depends only on the state of the system at time t, $X(t)$, *i.e.*,

$$\begin{aligned}&\mathbb{P}\left[A_{-k}(t) = a_{-k} \mid H(t)\right]\\ &= \mathbb{P}\left[A_{-k}(t) = a_{-k} \mid X(t) = x\right] = \pi_{-k}(x, a_{-k}),\end{aligned} \tag{2}$$

where $A_{-k}(t)$ denotes the action taken by all agents other than k at time t, $H(t)$ denotes the whole history of the process up to time t and $a_{-k} \in \mathcal{A}_{-k}$.

We also assume that the observations of each agent do not depend on the actions of the remaining agents, *i.e.*,

$$\mathbb{P}\left[Z_i(t) = z_i \mid X(t), A(t)\right] = \mathbb{P}\left[Z_i(t) = z_i \mid X(t), A_i(t)\right],$$

for all $i = 1, \ldots, N$. Therefore, we can simply write the observation probabilities as $\mathsf{O}_i(z_i \mid x, a_i), i = 1, \ldots, N$.

3.1 Query Actions and Resulting Observations

We assume that each agent has the ability to *query* the other agents for their local state information. In order to make this explicit, we differentiate between *communication actions* and the remaining actions—henceforth referred as *primitive actions*, and write the set of individual actions for agent k as the cartesian product of the set of communication actions, $\mathcal{A}_k^C$, and the set of primitive actions, $\mathcal{A}_k^P$, *i.e.*, $\mathcal{A}_k = \mathcal{A}_k^C \times \mathcal{A}_k^P$. We also assume that transition probabilities are independent of the communication actions, *i.e.*,

$$\mathsf{P}(y \mid x, \langle a_{-k}, (a_k^C, a_k^P)\rangle) = \mathsf{P}(y \mid x, \langle a_{-k}, (b_k^C, a_k^P)\rangle)$$

for any $x, y \in \mathcal{X}$, $a_{-k} \in \mathcal{A}_{-k}$, $a_k^P \in \mathcal{A}_k^P$ and $a_k^C, b_k^C \in \mathcal{A}_k^C$.

We also differentiate between *communication observations*—*i.e.*, observations that result from communication actions—and *primitive observations*, that do not depend on the communication actions. Formally, we write the set of individual observations for agent k as the cartesian product of the set of communication observations, $\mathcal{Z}_k^C$, and primitive observations, $\mathcal{Z}_k^P$, *i.e.*, $\mathcal{Z}_k = \mathcal{Z}_k^C \times \mathcal{Z}_k^P$. We

further assume that the communication observations do not depend on primitive actions, and that primitive observations do not depend on communication actions. This means that we can decouple the observation probabilities as

$$\mathsf{O}_k\big((z_k^C, z_k^P) \mid x, (a_k^C, a_k^P)\big) = \mathsf{O}_k^C(z_k^C \mid x, a_k^C)\mathsf{O}_k^P(z_k^P \mid x, a_k^P),$$

where

$$\mathsf{O}_k^C(z_k^C \mid x, a_k^C) = \mathbb{P}\left[Z_k^C(t) = z_k^C \mid X(t) = x, A_k^C(t) = a_k^C\right]$$
$$\mathsf{O}_k^P(z_k^P \mid x, a_k^P) = \mathbb{P}\left[Z_k^P(t) = z_k^P \mid X(t) = x, A_k^P(t) = a_k^P\right].$$

Finally, we assume that the reward function can be decomposed as the sum of two components, one which is independent on the primitive actions of agent k and on the actions of the other agents, and one other that does not depend on the communication actions of agent k. Formally, if $a = \langle a_{-k}, a_k\rangle$ and $a_k = (a_k^C, a_k^P)$, this means that the reward r can be written as

$$r(x, a) = r^P(x, \langle a_{-k}, a_k^P\rangle) + r^C(x, a_k^C). \tag{3}$$

From these assumptions, it follows that agent k can be modeled as a (single-agent) POMDP that we describe in the continuation.

3.2 POMDP Model for a Single Agent

Let $\mathcal{M} = (N, \mathcal{X}, (\mathcal{A}_k), (\mathcal{Z}_k), \mathsf{P}, (\mathsf{O}_k), r, \gamma)$ be a Dec-POMDP verifying the assumptions above. Let π_{-k} denote the (state-dependent) reduced joint policy for all agents other than k. The single-agent POMDP model for agent k is a tuple $\mathcal{M}_k = (\mathcal{X}, \mathcal{A}_k, \mathcal{Z}_k, \mathsf{P}_k, \mathsf{O}_k, r_k, \gamma)$, where:
- $\mathcal{X}$ corresponds to the original Dec-POMDP state-space.
- $\mathcal{A}_k$ is the individual action-space for agent k.
- $\mathcal{Z}_k$ is the individual observation-space for agent k.
- P_k are the transition probabilities obtained from the original transition probabilities. In particular, given an action $a_k = (a_k^C, a_k^P)$, we have

$$\mathsf{P}_k(y \mid x, a_k) = \sum_{a_{-k} \in \mathcal{A}_{-k}} \pi_{-k}(x, a_{-k})\mathsf{P}(y \mid x, \langle a_{-k}, a_k^P\rangle)$$

- O_k are the observation probabilities for agent k, that match the original Dec-POMDP observation probabilities. In particular, given an action $a_k = (a_k^C, a_k^P)$, we have

$$\mathsf{O}_k(z_k \mid x, a_k) = \mathsf{O}_k^C(z_k^C \mid x, a_k^C)\mathsf{O}_k^P(z_k^P \mid x, a_k^P), \tag{4}$$

where $z_k = (z_k^C, z_k^P)$.
- r_k is the reward function obtained from the original Dec-POMDP reward function after averaging over the other agents' policy, π_{-k}. In particular, given an

action $a_k = (a_k^C, a_k^P)$, we have

$$r_k(x, a_k) = \sum_{a_{-k} \in \mathcal{A}_{-k}} \pi_{-k}(x, a_{-k}) r(x, \langle a_{-k}, a_k \rangle)$$
$$= \sum_{a_{-k} \in \mathcal{A}_{-k}} \pi_{-k}(x, a_{-k}) r^P(x, \langle a_{-k}, a_k^P \rangle) + r^C(x, a_k^C).$$

Given this POMDP model, we can use standard POMDP solution techniques to explore the trade-off between the costs and benefits of communication for agent k.

This POMDP, however, exhibits several appealing features, one of which is the fact that the state $X(t)$ can be decomposed in two components, $X_k(t)$ and $X_{-k}(t)$, the second of which does not depend on the actions of Agent k. This fact can be leveraged to derive more efficient planning methods, much like in the so-called *mixed-observability Markov decision process* described in [7].

3.3 An Illustrative Example

We now illustrate the application of our proposed model in a simple navigation scenario, corresponding to the environment depicted in Fig. 1. In this variation of the scenario two robots must navigate to their corresponding goal states (marked with a boxed 1 and 2). At the same time, they must avoid colliding in the narrow doorway (the central state), since it leads to a large penalty. Agent 2 starts with equal probability in any of the shaded states on the right, and Agent 1 in the lightly shaded states on the left.

The application of our model allows us to better understand under which circumstances the benefits of using communication compensate for its costs. For this purpose, we assume each agent can observe its location and fix the policy of Agent 2 as shown in Fig. 1. As explained above, given such a policy we can construct a POMDP from the point of view of Agent 1, in which it can query Agent 2's observations at any time step, at a particular communication cost. Note that initially Agent 1 does not know exactly where Agent 2 is located, but does so after querying its observation (which in this case reveals Agent 2's location). However, in the case of noisy transitions, without querying every time step Agent 1's belief regarding Agent 2's location will flatten. We test several experimental conditions, that include the presence or absence of transition noise and different costs for the communication actions.

We note that, as the cost of the communication goes up, performance in terms of value goes down (Fig. 2a). Also, when the cost is 0, agent 1 queries very often, but with increasing communication cost the agent reduces its communication (Fig. 2b). Interesting, however, is to compare the difference between the deterministic and noisy transition cases. In the former case, the agent stops communicating when the communication cost reaches 0.3, while in the latter case the agent communicates up to and until the communication cost is 0.75. Given the increased stochasticity in the domain, the value of querying the other agent is higher, so even with a higher penalty the agent will communicate.

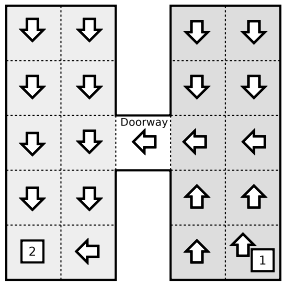


Fig. 1: H-environment: Arrows indicate the fixed policy for Agent 2.

Besides how often Agent 1 queries Agent 2's observation, it is also interesting to examine in which states it does so. When communication is free (Figs. 2c and d), Agent 1 queries in all the states it passes through.[4] With a communication cost of 0.3 (Figs. 2e and f), however, it only queries when near to and left of the doorway. In these states it is crucial to know Agent 2's location to avoid potential collisions, an intuition that is exploited automatically by our model.

Note that, in the deterministic environment, agent 1 can always "out-wait" the other agent, since the policy of the other is known and there is only so much time that it can take to cross the doorway. Therefore, communication is only worth it as long as the cost for communication is smaller than the cost paid for waiting.

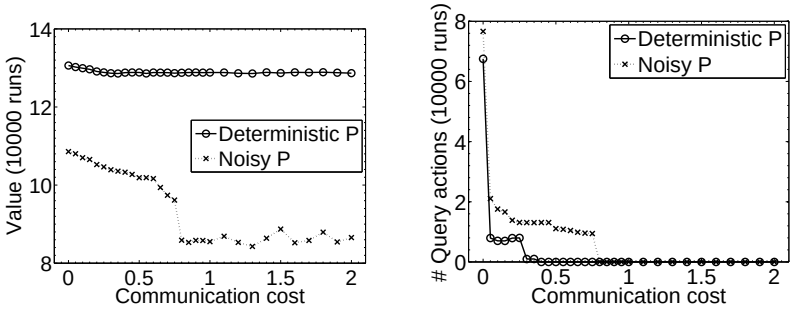
4 Computing Policies for Multiple Agents

In the previous section we proposed using a POMDP model to compute the policy for one agent k, assuming that the policy for the other agents is fixed, known and verifies (2). Given this POMDP model for agent k we can compute the corresponding optimal policy using any preferred POMDP solution technique. We used this approach to better understand the communication needs of one agent in a simple multiagent navigation scenario, and to determine in which situations the cost of communication outweighs its value.

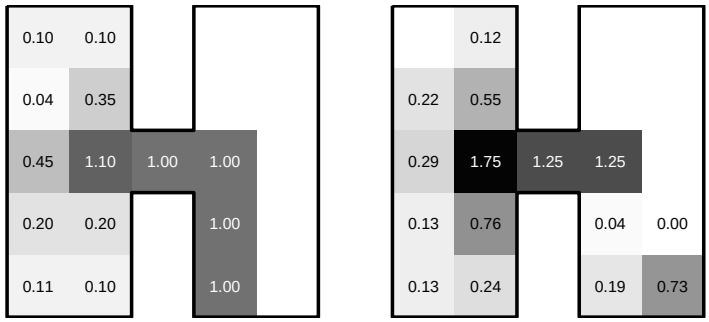
We now want to extend these ideas and actually compute the policy for *all* agents in the Dec-POMDP. One possibility for doing this is to fix the policies of all agents except one, derive the POMDP model from the previous section and compute the policy for that agent, and then move to the next agent, in a round-robin fashion. The main difficulty with this approach is related with the requirement in (2): since most POMDP solution techniques provide history-dependent policies, it is seldom the case that (2) will hold.

The requirement (2) is due to the need for the POMDP model for agent k to predict the dynamic behavior of the multiagent system without accessing the individual observations made by the other agents. If the POMDP model is not able to provide such accurate predictions, then using it to optimize the policy

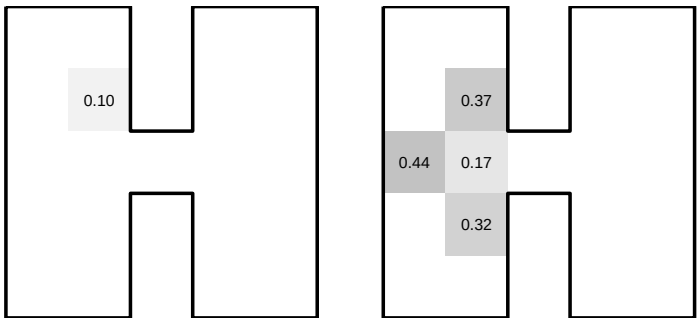
[4] We note that, due to the transition noise, an agent can remain in the same state more than one consecutive time-step, and hence the values > 1.



(a) Value. (b) # of queries.



(c) Det.1 P, $r^C = 0.0$. (d) Noisy P, $r^C = 0.0$.



(e) Det. P, $r^C = -0.3$. (f) Noisy P, $r^C = -0.3$.

Fig. 2: Results for the H-environment. (a) Average total discounted reward for agent 1; (b) Average number of query actions for agent 1; (c)-(f) States in which agent 1 chooses to query the other's state, varying in deterministic or noisy transitions and communication cost.

for agent k may lead to poor performance. Clearly, if the policy of any of the other agents depends on its individual observation history, it is not possible to build a POMDP model for these agents without an exponential blow up in the dimension of the POMDP, rendering the solution inviable. We note, however, that in scenarios with sparse interaction, we expect the policy of one agent to be mostly independent of the other agents. This means that, in a significant part of the state-space, the information about the other agents will have little impact on the action choice of our agent. Whenever this is not the case, the agent should use communication to overcome its perceptual limitations.

In our results, we adopt a simplifying approach and compute a *memoryless policy* for each individual POMDP. This ensures that (2) is trivially true and the POMDP models that we use for each agent are actually accurate—at the cost of restricting our agent behaviors to memoryless policies. We note that, while memoryless policies may lead to poor performance in general POMDPs [12], in environments with sparse interactions we expect that the consideration of memoryless policies may have a manageable impact in the performance of our agent. In particular, in problems where local observations provide rich information about the local state (such as Dec-MDPs) we expect our model to actually lead to very good results, since the sparse interactions allow the decisions of one agent to be mostly independent of the other agents.

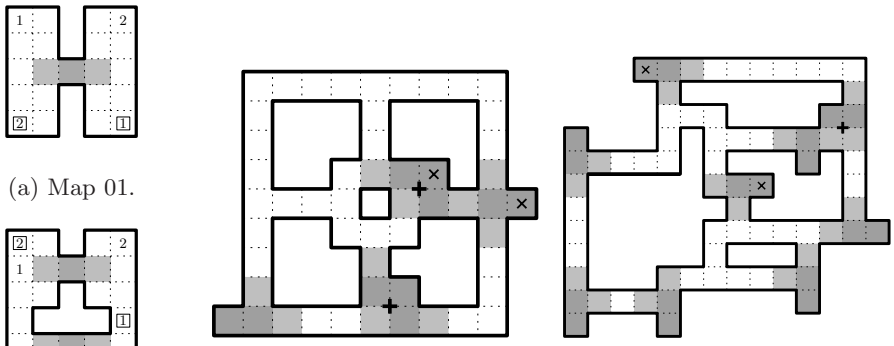
We conclude by noting that other possibilities exist to extend our ideas to the other agents. For example, it is possible to actually compute (or approximate) the optimal POMDP policies at the cost of inaccurate individual POMDP models. Other possibilities include using POMDP solutions based on fixed-length memory [6] or finite-state controllers [8], at the cost of augmenting the individual POMDP models, but we do not explore them in this paper.

5 Experiments

In this section we illustrate the application of the method described in the previous section to some simple navigation scenarios extracted from the POMDP and Dec-POMDP literature. We use robot navigation scenarios to test our algorithms (see Fig. 3), since our model is particularly suited for modeling multi-robot problems. Furthermore, results can be easily visualized and interpreted in this class of problems.

In each of the test scenarios, two robots must each reach one specific state. In the smaller environments (Maps 1 and 2), the goal state is marked with a boxed number, corresponding to the number of the robot. The cells with a simple number correspond to the initial states for that robot. In the larger environments, the goal for each robot is marked with a cross, $\times$, and the robots each depart from the other's goal state, in an attempt to increase the possibility of interaction.

Each robot has 4 actions that move the robot in one of the four possible direction with probability 0.8 and fail with probability 0.2. It also has available a fifth "NoOp" action. The shaded regions correspond to areas inside of which the agents are able to communicate. The darker cells correspond to states where



(a) Map 01.

(b) Map 02.

(c) PENTAGON.

(d) CIT.

Fig. 3: Environments used in the experiments. The dark gray areas correspond to states where coordination is required and the light gray areas to the states where the agents can communicate. We refer to the main text for details.

the agents receive a penalty of -20 if they stand there simultaneously. Also, in these interaction states, the rate of action failure is increased to 0.36.

All agents have full local state observability and when an agent queries another agent, it incurs a cost of -0.05 and successfully observes the local state of that agent with a probability of 0.8. With a probability of 0.2 it receives no observation about the state of the other. When an agent reaches its goal position, it receives a reward of 10 and moves to a rewardless absorbing state. Throughout the experiments, we used $\gamma = 0.95$.

For each of the different scenarios in Fig. 3 we ran our proposed algorithm and then tested the computed policy for $1,000$ independent trials of 250 steps each. The obtained performance in terms of total discounted reward can be found in Table 1. For comparison purposes, we also provide the obtained total discounted reward for:

1. A set of agents following *individual MDP* policies, disregarding the existence of other agents in the environment (INDIVIDUAL);
2. A set of agents following *fully observable MMDP* policies without incurring any communication cost (JOINT);
3. A set of agents similar to the one from our previous example, where one of the agents follows the actual MMDP policy and the other a corresponding single-agent POMDP policy (dubbed SINGLE-POMDP);
4. A set of agents communicating at every time-step and acting according to either the MDP or the MMDP policies, depending on the observation (dubbed ALWAYS COMM);
5. Three sets of agents similar to the previous one, but that communicates only every k time-steps, with $k = 2, 3, 4$ (dubbed COMM $k = 2, 3$ or 4).

Table 1: Total discounted reward for each set of agents in each of the test-scenarios. The results are averaged over 1,000 independent Monte-Carlo runs. Entries in *italic* in the same column are not statistically different.

Environment	**Map 1**	**Map 2**	**cit**	**pentagon**
INDIVIDUAL	−1.362	1.709	*5.306*	5.641
JOINT	5.763	6.616	*5.305*	7.606
MULTI-POMDP	*3.651*	*3.345*	5.083	3.345
SINGLE-POMDP	*3.546*	*3.411*	5.203	6.853
ALWAYS COMM	3.186	4.501	4.308	6.000
COMM $k = 2$	−0.137	3.837	4.806	*6.165*
COMM $k = 3$	0.189	2.097	4.980	5.142
COMM $k = 4$	0.007	*3.323*	4.816	*6.100*

The results of our proposed method correspond to those dubbed MULTI-POMDP.

The results in Table 1 prompt several interesting observations. First of all, out of all 4 environments, Map 1 is the one where coordination is more critical and CIT is the one where coordination is less critical. This can be observed by noticing the difference in value between the single and the joint sets of agents.

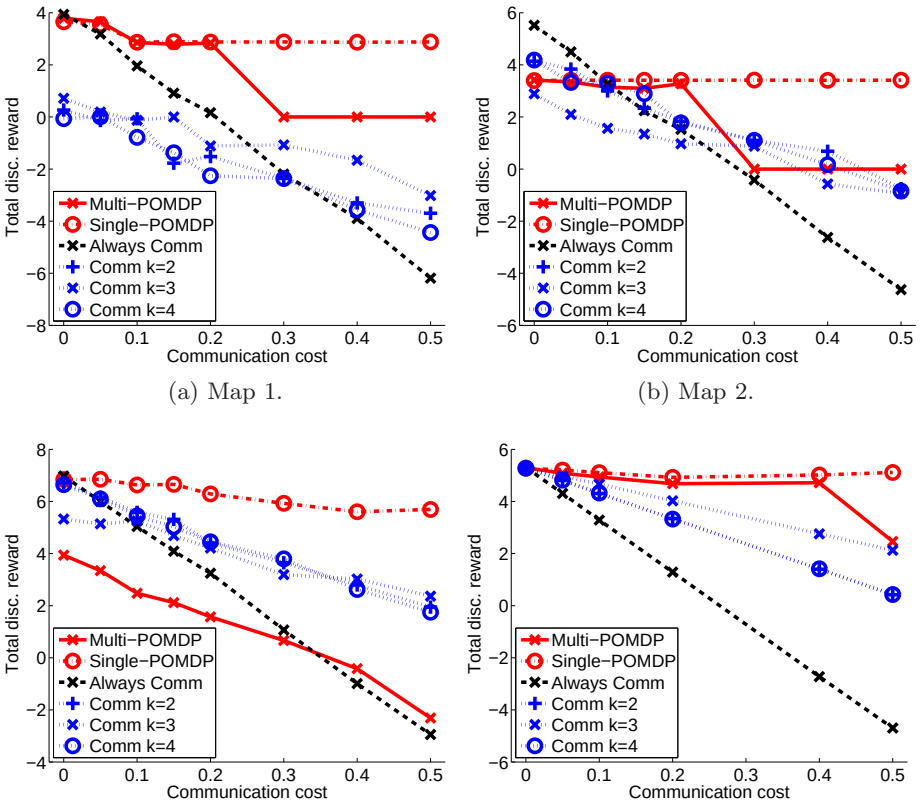
Secondly, we note that in all but the CIT environment, the fact that our approach (MULTI-POMDP) uses memoryless policies causes the agents to behave "cautiously", leading one of the agents to avoid crossing the critical areas. This is the reason why, in such scenarios, our method attains approximately 1/2 of the reward received by the JOINT agents.

Another interesting aspect is that in all but the PENTAGON scenario our agent is able to outperform the other communicating agents, indicating that even with the limitations arising from the adopted POMDP solution technique, our agents are still able to effectively balance the costs and benefits of communication. To further explore this aspect, we analyzed the performance of all methods with different communication costs in the range from 0 to 0.5. The results obtained are depicted in Fig. 4. The plotted results are in accordance with those in Table 1. It is worth noting, however, that as the communication cost increases, the agents eventually cease to communicate and, since the penalty for failing to coordinate is to high, they eventually opt by standing still.

Finally, we conclude by noting that, in the PENTAGON scenario, the performance of our agents is significantly below that of all other agents. This is due to the memoryless policy adopted, that greatly underestimates the value of the observations that the agents make, leading to an excessively "conservative" policy that avoids collisions at all costs.

6 Conclusions

In this paper we proposed the use of a POMDP model to analyze the communication needs of an agent in a Dec-POMDP scenario where the interaction



(a) Map 1. (b) Map 2.

(c) PENTAGON. (d) CIT.

Fig. 4: Total discounted reward for each set of agents in each of the test scenarios as the communication cost varies from 0 to 0.5. The results are averaged over 1,000 independent Monte-Carlo runs.

between the agents is sparse. We used this approach in a simple example, illustrating how the communication needs of an agent can be computed so as to optimize communication. For example, in the situation depicted in Fig. 1, our approach was successfully able to capture the intuition that the fundamental states for coordination are those around the doorway.

We also further explored the usefulness of this approach in computing policies for Dec-POMDPs where the agents must explicitly reason about communication. We used simple memoryless policies with which we can use the POMDP approach for communication in a round-robin fashion and compute the policy for each agent conditioned on the policies already derived for the other agents. We noticed that the use of memoryless policies renders the agents "cautious", in that they prefer not to receive any reward rather than risking a large penalty.

In the future we will explore other POMDP solution methods at the cost of violating requirement (2), exploring bounds on the loss of value to be potentially incurred.

Acknowledgments

This work was funded in part by Fundação para a Ciência e a Tecnologia (INESC-ID multiannual funding) through the PIDDAC Program funds and the project CMU-PT/SIA/0023/2009 under the Carnegie Mellon-Portugal Program. M.S. is funded by an EU Marie Curie Intra-European Fellowship.

References

1. Allen, M., Zilberstein, S.: Complexity of decentralized control: Special cases. In: Adv. Neural Information Proc. Systems. vol. 22, pp. 19–27 (2009)
2. Becker, R., Lesser, V., Zilberstein, S.: Decentralized Markov decision processes with event-driven interactions. In: Proc. Int. Conf. Auton. Agents and Multiagent Systems. pp. 302–309 (2004)
3. Becker, R., Zilberstein, S., Lesser, V., Goldman, C.: Solving transition independent decentralized Markov decision processes. J. Artificial Intelligence Research 22, 423–455 (2004)
4. Goldman, C., Zilberstein, S.: Decentralized control of cooperative systems: Categorization and complexity analysis. J. Artificial Intelligence Research 22, 143–174 (2004)
5. Goldmann, C., Allen, M., Zilberstein, S.: Learning to communicate in a decentralized environment. J. Auton. Agents and Multiagent Systems 15(1), 47–90 (2007)
6. McCallum, A.: Instance-based utile distinctions for reinforcement learning with hidden state. In: Proc. 12th Int. Conf. Machine Learning. pp. 387–396 (1995)
7. Ong, S., Png, S., Hsu, D., Lee, W.: POMDPs for robotic tasks with mixed observability. In: Proc. 2010 Conf. Robotics: Science and Systems (2010)
8. Poupart, P., Boutilier, C.: Bounded finite state controllers. In: Adv. Neural Information Proc. Systems. vol. 16 (2004)
9. Pynadath, D., Tambe, M.: The communicative multiagent team decision problem: Analyzing teamworktheories and models. J. Artificial Intelligence Research 16, 389–423 (2002)
10. Roth, M., Simmons, R., Veloso, M.: Exploiting factored representations for decentralized execution in multiagent teams. In: Proc. Int. Conf. Auton. Agents and Multiagent Systems. pp. 469–475 (2007)
11. Seuken, S., Zilberstein, S.: Formal models and algorithms for decentralized decision making under uncertainty. Autonomous Agents and Multi-Agent Systems (Feb 2008)
12. Singh, S., Jaakkola, T., Jordan, M.: Learning without state-estimation in partially observable Markovian decision processes . In: Proc. 11th Int. Conf. Machine Learning. pp. 284–292 (1994)
13. Spaan, M., Melo, F.: Interaction-driven Markov games for decentralized multiagent planning under uncertainty. In: Proc. Int. Conf. Auton. Agents and Multiagent Systems. pp. 525–532 (2008)

14. Varakantham, P., Kwak, J., Taylor, M., Marecki, J., Scerri, P., Tambe, M.: Exploiting coordination locales in distributed POMDPs via social model shaping. In: Proc. 19th Int. Conf. Automated Planning and Scheduling. pp. 313–320 (2009)
15. Witwicki, S.J., Durfee, E.H.: Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In: Int. Conf. on Automated Planning and Scheduling (2010)
16. Wu, F., Zilberstein, S., Chen, X.: Multi-agent online planning with communication. In: Proc. Int. Conf. Automated Planning and Scheduling. pp. 321–329 (2009)