Dec-POMDPs with delayed communication

Frans A. Oliehoek
Informatics Institute
University of Amsterdam
The Netherlands
faolieho@science.uva.nl

Matthijs T.J. Spaan
Institute for Systems and Robotics
Instituto Superior Técnico
Lisbon, Portugal
mtjspaan@isr.ist.utl.pt

Nikos Vlassis
Department of Production Engineering and Management
Technical University of Crete
Chania, Greece
vlassis@dpem.tuc.gr

ABSTRACT

In this work we consider the problem of multiagent planning under sensing and acting uncertainty with a one time-step delay in communication. We adopt decentralized partially observable Markov processes (Dec-POMDPs) as our planning framework. When instantaneous and noise-free communication is available, agents can instantly share local observations. This effectively reduces the decentralized planning problem to a centralized one, with a significant decrease in planning complexity. However, instantaneous communication is a strong assumption, as it requires the agents to synchronize at every time step. Therefore, we explore planning in Dec-POMDP settings in which communication is delayed by one time step. We show that such situations can be modeled by Bayesian games in which the types of the agents are defined by their last private observation. We will apply Bayesian games to define a value function Q_{BG} on the joint belief space, and we will show that it is the optimal payoff function for our Dec-POMDP setting with one time-step delayed communication. The Q_{BG}-value function is piecewise linear and convex over the joint belief space, which we will use to define Q_{BG}-value iteration. Finally, we will adapt PERSEUS, an approximate POMDP solver, to compute Q_{BG}-value functions, and we will use it to perform some proof-of-concept experiments.

1. INTRODUCTION

In this work we consider the problem of multiagent planning under sensing and acting uncertainty with a one time-step delay in communication. We adopt a decision-theoretic perspective, and we employ decentralized partially observable Markov decision processes (Dec-POMDPs) as our planning paradigm. The problem of computing optimal plans in Dec-POMDPs is provably intractable (NEXP-complete [2]). When instantaneous and noise-free communication is available, agents can instantly share local observations. This effectively reduces the decentralized planning problem to a centralized single-agent POMDP [11], with a significant decrease in planning complexity. A large body of literature exists on exact and approximate POMDP solving exists, which has been exploited by several Dec-POMDP planning methods [3, 12, 15].

However, instantaneous communication is a strong assumption, as it requires the agents to synchronize at every time step. Each agent broadcasts its local observation to the team, and waits for incoming messages. Only when all observations from the other agents have arrived, an agent can decide upon its action. As such, relying on instant observation sharing can slow down task execution, and we will explore planning for agents that share local observations with a one time-step delay. For example, a team of robots are often linked using a wireless network. When wireless connectivity is low, messages may have to be retransmitted several times, leading to significant delays in the online decision making. Basically, we assume that the time between two time steps is sufficient for communication to complete.

In particular, when an agent receives an observation at time t, it will send it to its team members, but it immediately starts executing an action, without waiting for incoming messages. However, we assume that at time step $t+1$ all local observations from t have been received. Each agent can use the joint observation history up to time t and its local observation at $t+1$ to select its action at time $t+1$. An agent does not know the joint observation received at time $t+1$, but still has to choose an action. Such a situation can be modeled by using a *Bayesian game* [10], which is a strategic game with imperfect information. It allows us to compute the optimal action for an agent, given its private knowledge of its local observation at time $t+1$, and the joint history up to time t. In this way, we relax the assumption of instantaneous communication for observation sharing, and allow for better response times.

Oliehoek and Vlassis [8] have presented a value function Q_{BG} which is based on Bayesian game solutions. It assumes that agents know the joint action-observation history up to one time step ago, and is used as an upper bound to the optimal Dec-POMDP value function Q^*. We have also shown that we can define a Q_{BG}-value function over joint beliefs, and that it is piecewise linear and convex (PWLC) over this joint belief space [9]. Here we will show how it can be used for planning in Dec-POMDPs with delayed communication. In particular, we will show that the Q_{BG}-value function is the optimal payoff function for the Bayesian games, and we show how to perform exact Q_{BG}-value iteration. We will adapt PERSEUS [14], an approximate POMDP value iteration method, to compute approximate Q_{BG}-value functions

MSDM 2007 May 15, 2007, Honolulu, Hawai'i, USA.

more efficiently. We will apply it to perform some experiments that show the viability of Q_{BG}-value functions. Our theoretical results are in accordance with earlier results in the control literature, and the proposed Bayesian game formulation provides an alternative view of existing solution techniques [16, 5, 1].

In contrast with most current algorithms for planning in Dec-POMDP settings, which are based on policy search [7, 3, 4, 15], we consider value-iteration techniques. The first application of Bayesian games to Dec-POMDPs has been an approximate policy search technique [3]. However, there are two main differences with our work: we consider a Dec-POMDP model with delayed communication, while Emery-Montemerlo et al. [3] do not consider communication. Furthermore, a heuristic forward search in the joint belief tree is performed, while we use Bayesian games to perform value iteration. As such, the Q_{BG}-value function has merits beyond the delayed-communication setting we consider in this paper. In particular, several Dec-POMDP methods employ value functions over joint beliefs: as a heuristic to focus approximate policy search [3] for instance, or to provide an admissible heuristic for optimal methods based on A* search [15, 8].

The rest of the paper is structured as follows. In Section 2 we will briefly present the Dec-POMDP model. In Section 3 we will define value functions for Dec-POMDPs with instantaneous or delayed communication, and we will introduce Bayesian games. Section 4 shows how we can define finite-horizon and infinite-horizon Q_{BG}-value iteration, and we extend an approximate POMDP solver to compute approximate Q_{BG}-value functions. In Section 5 we will show a number of experiments, and in Section 6 we discuss our work.

2. BACKGROUND

We base our work on the *decentralized partially observable Markov decision process (Dec-POMDP)* framework [2]. Formally, a Dec-POMDP with m agents is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, T, R, \mathcal{O}, O, b^0 \rangle$, where

- $\mathcal{S}$ is a finite set of states;
- $\mathcal{A} = \times_i \mathcal{A}_i$ is the set of *joint actions*, where $\mathcal{A}_i$ is the set of actions available to agent i. Every time step, the agents take one joint action $\mathbf{a} = \langle a_1, ..., a_m \rangle$, but agents do not observe each other's actions.
- The transition probabilities $P(s'|s, \mathbf{a})$ are specified by the transition function T.
- Similar to the action component of the model $\mathcal{O} = \times_i \mathcal{O}_i$ is the set of joint observations. Again, every time step one joint observation $\mathbf{o} = \langle o_1, ..., o_m \rangle$ is received, from which each agent i only observes its own component o_i.
- O is the observation function, which specifies the probability of joint observations: $P(\mathbf{o}|\mathbf{a}, s')$.
- R is the immediate reward function, which maps states and joint actions to reals: $R(s, \mathbf{a})$.
- $b^0 \in \mathcal{P}(\mathcal{S})$ is the initial state distribution at time $t = 0$, where $\mathcal{P}(S)$ denotes the infinite set of probability distributions over the finite set S.

When there is only one agent in a Dec-POMDP, the model reduces to a POMDP [6].

A tuple of policies $\pi = \langle \pi_1, ..., \pi_m \rangle$ is referred to as a *joint policy.* In general, each individual deterministic (*pure*) policy π_i is a mapping from histories of observations to actions: $\pi_i(\big(o_i^1, ..., o_i^t\big)) = a_i$. Here, $\big(o_i^1, ..., o_i^t\big)$ is the sequence of observations received by agent i up to time step t, which we refer to as the *observation history* $\vec{o}_i^{\,t}$. We also use a different notion of history, namely the *action-observation history* $\vec{\theta}_i^{\,t}$, which consists of all observations received and actions taken up to time step t: $\vec{\theta}_i^{\,t} = \big(a_i^0, o_i^1, a_i^1 ..., a_i^{t-1}, o_i^t\big)$. The corresponding *joint* histories are denoted respectively as $\vec{\mathbf{o}}^{\,t}$ and $\vec{\theta}^{\,t}$.

3. DEC-POMDP MODELS WITH COMMUNICATION

We will now focus on computing optimal value functions for Dec-POMDPs with free and noiseless communication. First we will treat the case where this communication is instantaneous, followed by value functions for Dec-POMDPs with a one time-step delay in communication.

3.1 Instantaneous communication

A natural approach to alleviate the burden of the complexity result is to consider communication. Pynadath and Tambe proved that, in the case of free, instantaneous and noiseless communication, sharing local observations at each time step is optimal [11]. In this case the problem reduces to a POMDP: the agents can communicate their individual observations, which effectively means they observe the joint observation. This allows each agent to maintain a *joint belief* $b^{\vec{\theta}^{\,t}}$. The optimal value function for a POMDP is based on these joint beliefs and satisfies the following Bellman equation:

$$Q^*_{\mathrm{P}}(b^{\vec{\theta}^{\,t}}, \mathbf{a}) = R(b^{\vec{\theta}^{\,t}}, \mathbf{a}) + \sum_{\mathbf{o}} P(\mathbf{o}|b^{\vec{\theta}^{\,t}}, \mathbf{a}) \max_{\pi_{\mathrm{P}}^{t+1}(b^{\vec{\theta}^{\,t+1}})} Q^*_{\mathrm{P}}(b^{\vec{\theta}^{\,t+1}}, \pi_{\mathrm{P}}^{t+1}(b^{\vec{\theta}^{\,t+1}})), \quad (1)$$

where $R(b^{\vec{\theta}^{\,t}}, \mathbf{a}) = \sum_s R(s, \mathbf{a}) b^{\vec{\theta}^{\,t}}(s)$ is the expected immediate reward, and $b^{\vec{\theta}^{\,t+1}}$ is the joint belief resulting from $b^{\vec{\theta}^{\,t}}$ by action $\mathbf{a}$ and joint observation $\mathbf{o}$. This resulting joint belief can be calculated by Bayes' rule:

$$\forall_{s'} \quad b^{\vec{\theta}^{\,t+1}}(s') = P(s'|b^{\vec{\theta}^{\,t}}, \mathbf{a}, \mathbf{o}) = \frac{P(\mathbf{o}|\mathbf{a}, s')}{P(\mathbf{o}|b^{\vec{\theta}^{\,t}}, \mathbf{a})} \sum_s P(s'|s, \mathbf{a}) b^{\vec{\theta}^{\,t}}(s). \quad (2)$$

The joint policy at the next time step π_{P}^{t+1} is a mapping from beliefs to joint actions $\pi_{\mathrm{P}}^{t+1} : \mathcal{P}(\mathcal{S}) \rightarrow \mathcal{A}$. This means that the maximization operator in (1) selects the maximizing joint action. Solving (1) is not trivial as the space of beliefs is continuous. If one assumes that the initial belief state is given, one option is to generate all possible beliefs and then perform standard value iteration. Rather, most POMDP solvers exploit the fact that (1) is piecewise-linear and convex (PWLC) [13].

3.2 Delayed communication

In the previous section, we discussed how the Bellman equation for POMDPs characterizes the optimal Q-value

θ_1 \ θ_2		o_2: a_2	o_2: $\bar{a}_2$	$\bar{o}_2$: a_2	$\bar{o}_2$: $\bar{a}_2$
o_1	a_1	$+0.1$	$+2.2$	$+0.4$	-0.2
	$\bar{a}_1$	-0.5	$+2.0$	$+1.0$	$+2.0$
$\bar{o}_1$	a_1	$+0.4$	-0.2	$+0.7$	-2.6
	$\bar{a}_1$	$+1.0$	$+2.0$	$+2.5$	$+2.0$

Figure 1: A Bayesian game modeling the uncertainty regarding the last observation. Each of the 2 agents has 2 observations and 2 actions. The payoff function $Q(\theta, \mathbf{a})$ specifies the entries. The probability distribution over joint types here is uniform: $P(o_1, o_2) = P(o_1, \bar{o}_2) = ... = P(\bar{o}_1, \bar{o}_2) = 0.25$. Given this distribution, the highlighted entries indicate the optimal solution that has an expected value of $+2.0$.

function in the case of instantaneous communication of observations. In this section we will do the same for communicated observations that arrive with a delay of one time step. Receiving communicated observations with a one time-step delay means that, at a particular time step t, each agent i will know the previous joint observation history $\vec{\mathbf{o}}^{t-1}$ and its last individual observation o_i^t, but is uncertain regarding the last joint observation $\mathbf{o}^t$. The agents can also deduce $\vec{\theta}^{t-1}$, assuming that they know each other's pure policy.

The uncertainty regarding the last observation can be modeled using a *Bayesian game (BG)* [10]. A BG is a strategic game with imperfect information. In particular, each agent i has some individual information which defines the agent's *type* $\theta_i \in \Theta_i$. In this case the type of agent i corresponds to its last observation $\theta_i \equiv o_i^t$. $\Theta = \times_i \Theta_i$ is the set of joint types, in this case corresponding with the set of joint observations $\mathcal{O}$, over which a probability function $P(\Theta)$ is specified, in this case $P(\theta) \equiv P(\mathbf{o}^t | b^{\vec{\theta}^{t-1}}, \mathbf{a}^{t-1})$. Finally, the BG also specifies a payoff function $Q(\theta, \mathbf{a})$ that maps joint types and actions to rewards. In general each agent can have its own payoff function, but because in the Dec-POMDP model the agents receive the same rewards, the Bayesian game also specifies identical payoffs. Fig. 1 show an example of the BG in the case of two agents that both have two actions and two observations.

A policy in a BG for agent i maps its types, in this case individual observations, to its actions $\beta_i : \mathcal{O}_i \to \mathcal{A}_i$. E.g., in Fig. 1, the optimal BG-policy for agent 1 specifies $\{o_1 \to \bar{a}_1, \bar{o}_1 \to \bar{a}_1\}$. The solution of a BG with identical payoff is given by the optimal joint BG-policy β^*:

$$\beta^* = \arg\max_{\beta} \sum_{\theta \in \Theta} P(\theta) Q(\theta, \beta(\theta)), \quad (3)$$

where $\beta(\theta) = \langle \beta_1(\theta_1), ..., \beta_m(\theta_m) \rangle$ is the joint action specified by joint BG-policy β for joint type θ. Unfortunately, there is no other method known to optimally solve (3) than brute force search. For larger action and observation sets, approximate solution methods such as alternating maximization can be employed [7, 3].

In [8] the Q_{BG}-value function was introduced as an approximation for the communication-free Dec-POMDP setting. It is defined as

$$Q^*_{\mathrm{BG}}(\vec{\theta}^t, \mathbf{a}) = R(\vec{\theta}^t, \mathbf{a}) + \max_{\beta_{\langle \vec{\theta}^t, \mathbf{a} \rangle}} \sum_{\mathbf{o}} P(\mathbf{o} | \vec{\theta}^t, \mathbf{a}) Q^*_{\mathrm{BG}}(\vec{\theta}^{t+1}, \beta_{\langle \vec{\theta}^t, \mathbf{a} \rangle}(\mathbf{o})), \quad (4)$$

where $\beta_{\langle \vec{\theta}^t, \mathbf{a} \rangle} = \left\langle \beta_{\langle \vec{\theta}^t, \mathbf{a} \rangle, 1}(o_1), ..., \beta_{\langle \vec{\theta}^t, \mathbf{a} \rangle, m}(o_m) \right\rangle$ is a tuple of individual BG-policies $\beta_{\langle \vec{\theta}^t, \mathbf{a} \rangle, i} : \mathcal{O}_i \to \mathcal{A}_i$, and where

$$R(\vec{\theta}^t, \mathbf{a}) = \sum_s R(s, \mathbf{a}) P(s | \vec{\theta}^t) = \sum_s R(s, \mathbf{a}) b^{\vec{\theta}^t}(s). \quad (5)$$

Now we will demonstrate that the Q_{BG}-value function, is the optimal payoff function for the BGs introduced above for the delayed communication case. For the last time step $t = h-1$, Q should be based on the immediate reward only. Let us assume the action-observation history at the previous time step is $\vec{\theta}^{h-2}$. Because the agents know each other's policies, they know the last taken joint action and thus the joint observation probability $P(\mathbf{o} | \vec{\theta}^{h-2}, \mathbf{a}^{h-2})$. Also, for each joint observation $\mathbf{o}$, they know the resulting joint action-observation history $\vec{\theta}^{h-1}$ and can calculate the probability over states $b^{\vec{\theta}^{h-1}}$ it induces using (2). Consequently, the optimal Q-value function, $\mathrm{Q}^*_{\mathrm{BG}}$, for the last time step is

$$Q^*_{\mathrm{BG}}(\vec{\theta}^{h-1}, \mathbf{a}) = R(\vec{\theta}^{h-1}, \mathbf{a}). \quad (6)$$

Given these $\mathrm{Q}^*_{\mathrm{BG}}$-values, $\beta^*_{\langle \vec{\theta}^{h-2}, \mathbf{a} \rangle}$, the optimal policy for the last time step given the previous joint history $\vec{\theta}^{h-2}$ and action $\mathbf{a}$, is given using (3):

$$\beta^*_{\langle \vec{\theta}^{h-2}, \mathbf{a} \rangle} = \arg\max_{\beta_{\langle \vec{\theta}^{h-2}, \mathbf{a} \rangle}} \sum_{\mathbf{o} \in \mathcal{O}} P(\mathbf{o} | \vec{\theta}^{h-2}, \mathbf{a}) Q^*_{\mathrm{BG}}(\vec{\theta}^{h-1}, \beta(\mathbf{o})). \quad (7)$$

The optimal $\mathrm{Q}^*_{\mathrm{BG}}$ value for the before-last time step $h-2$ now can be defined as the expected immediate reward plus the expected future reward, which is the expectation of $\beta^*_{\langle \vec{\theta}^{h-2}, \mathbf{a} \rangle}$ as specified by the right side of (7)

$$Q^*_{\mathrm{BG}}(\vec{\theta}^{h-2}, \mathbf{a}) = R(\vec{\theta}^{h-2}, \mathbf{a}) + \max_{\beta_{\langle \vec{\theta}^{h-2}, \mathbf{a} \rangle}} \sum_{\mathbf{o} \in \mathcal{O}} P(\mathbf{o} | \vec{\theta}^{h-2}, \mathbf{a}) Q^*_{\mathrm{BG}}(\vec{\theta}^{h-1}, \beta^*_{\langle \vec{\theta}^{h-2}, \mathbf{a} \rangle}(\mathbf{o})). \quad (8)$$

Generalization of this equation yields (4). Given that the entire $\mathrm{Q}^*_{\mathrm{BG}}$ value function as given by (4) is calculated (i.e., for all $t = 0, ..., h-1$ for all $\vec{\theta}^t$ and $\mathbf{a}$), the optimal policy can be extracted and executed as follows. At a time step t, an arbitrary agent i has received the delayed communication meaning it can deduce the previous joint action-observation history $\vec{\theta}^{t-1}$ and joint action $\mathbf{a}^t$. With this information it can construct and solve the corresponding BG:

$$\beta^*_{\langle \vec{\theta}^{t-1}, \mathbf{a} \rangle} = \arg\max_{\beta} \sum_{\mathbf{o} \in \mathcal{O}} P(\mathbf{o} | \vec{\theta}^{t-1}, \mathbf{a}) Q^*_{\mathrm{BG}}(\vec{\theta}^t, \beta(\mathbf{o})). \quad (9)$$

Given $\beta^*_{\langle \vec{\theta}^{t-1}, \mathbf{a} \rangle}$ the agent can consider its own component $\beta^*_{\langle \vec{\theta}^{t-1}, \mathbf{a} \rangle, i}$ and execute the optimal individual action for the observation it actually received $a_i^* = \beta^*_{\langle \vec{\theta}^{t-1}, \mathbf{a} \rangle, i}(o_i)$.[1]

[1]Of course, (9) has to be evaluated in order to calculate the entire $\mathrm{Q}^*_{\mathrm{BG}}$-value function (it is part of (4)). Therefore, in

4. $\mathrm{Q_{BG}}$-VALUE ITERATION

The previous section explained how $\mathrm{Q_{BG}}$-value functions can be used for Dec-POMDPs with delays in communication. Here we will discuss how we the $\mathrm{Q_{BG}}$-value function can be computed more efficiently, by employing techniques used for POMDPs. This is possible because the $\mathrm{Q_{BG}}$-value function for a one time-step delay is piecewise-linear and convex (PWLC) [9].

4.1 Finite-horizon value iteration

The fact that the $\mathrm{Q_{BG}}$-value function is piecewise-linear and convex in the joint belief space allows to use many techniques from the POMDP literature [6]. Value iteration, for instance, is a method for solving POMDPs that builds a sequence of value functions which converge to the optimal value function for the current task [13]. Analogous to the POMDP case, we will define how to compute $\mathrm{Q}_{\mathrm{BG}}^{t-1}$ from $\mathrm{Q}_{\mathrm{BG}}^{t}$, i.e., how to extend a $\mathrm{Q_{BG}}$ value function for horizon h to horizon $h+1$. The main idea behind many value-iteration algorithms for POMDPs is that, given that $\forall_{\mathbf{a}}\ Q^t(\cdot,\mathbf{a})$ is represented by a set of vectors $\mathcal{V}_{\mathbf{a}}^t$, we can calculate the sets of vectors $\mathcal{V}_{\mathbf{a}}^{t-1}$ that represent $Q^{t-1}(\cdot,\mathbf{a})$. In more detail, for a particular belief b and joint action $\mathbf{a}$, we can compute its maximizing vector

$$\underset{v_{\mathbf{a}}^{t-1}\in\mathcal{V}_{\mathbf{a}}^{t-1}}{\arg\max}\ b\cdot v_{\mathbf{a}}^{t-1}, \tag{10}$$

by performing a so-called backup operation H. For the POMDP case, such a backup computes the optimal vector for a given belief b by back-projecting all vectors in the current horizon value function one step from the future and returning the vector that maximizes the value of b, see e.g., [14]. These back-projected vectors form the basis of both the POMDP and the $\mathrm{Q_{BG}}$ backup operators, and for a particular $\mathbf{a},\mathbf{o}^{t+1}$ and $v_{\mathbf{a}'}^{t+1}\in\mathcal{V}_{\mathbf{a}'}^{t+1}$ they are defined as

$$g_{\mathbf{a},\mathbf{o}}^{v_{\mathbf{a}'}^{t+1}}(s^t)=\sum_{s^{t+1}\in\mathcal{S}}P(\mathbf{o}|\mathbf{a},s^{t+1})P(s^{t+1}|s^t,\mathbf{a})v_{\mathbf{a}'}^{t+1}(s^{t+1}). \tag{11}$$

In the POMDP case, we can define the finite-horizon (not discounted) backup of a joint belief for a particular joint action $\mathbf{a}$ as

$$H_{\mathrm{P}}(b^{\vec{\theta}^t},\mathbf{a})=R_{\mathbf{a}}+\sum_{\mathbf{o}}\underset{\{g_{\mathbf{a},\mathbf{o}}^i\}_i}{\arg\max}\, b^{\vec{\theta}^t}\cdot g_{\mathbf{a},\mathbf{o}}^i, \tag{12}$$

where $(\cdot)$ denotes inner product, $R_{\mathbf{a}}$ is an $|S|$-dimensional vector, $R_{\mathbf{a}}(s)=R(s,\mathbf{a})$, and i is an index over all vectors in the next-horizon value function $\mathcal{V}_{\mathbf{a}'}^{t+1},\forall_{\mathbf{a}'}$. Now, in the $\mathrm{Q_{BG}}$ backup we use the same back-projected vectors, but instead of maximizing over all, we only maximize over those whose next time-step action $\mathbf{a}'$ is consistent with $\beta_{\langle\vec{\theta}^t,\mathbf{a}\rangle}$ a particular BG-policy for the BG for $\vec{\theta}^t,\mathbf{a}$. This set is defined as

$$\mathcal{G}_{\mathbf{a},\mathbf{o},\beta_{\langle\vec{\theta}^t,\mathbf{a}\rangle}}\equiv\left\{g_{\mathbf{a},\mathbf{o}}^{v_{\mathbf{a}'}^{t+1}}\mid v_{\mathbf{a}'}^{t+1}\in\mathcal{V}_{\mathbf{a}'}^{t+1}\wedge\beta_{\langle\vec{\theta}^t,\mathbf{a}\rangle}(\mathbf{o})=\mathbf{a}'\right\}. \tag{13}$$

The $\mathrm{Q_{BG}}$-backup is completed by maximizing over the BG-policies:

$$H_{\mathrm{B}}(b^{\vec{\theta}^t},\mathbf{a})=R_{\mathbf{a}}+\max_{\beta_{\langle\vec{\theta}^t,\mathbf{a}\rangle}}\sum_{\mathbf{o}^{t+1}}\underset{g_{\mathbf{a},\mathbf{o}}^{v_{\mathbf{a}'}^{t+1}}\in\mathcal{G}_{\mathbf{a},\mathbf{o},\beta_{\langle\vec{\theta}^t,\mathbf{a}\rangle}}}{\arg\max}\ b^{\vec{\theta}^t}\cdot g_{\mathbf{a},\mathbf{o}}^{v_{\mathbf{a}'}^{t+1}}. \tag{14}$$

At this point we can use H_{B} for value iteration in finite-horizon settings; next we will extend our work to the infinite-horizon case.

4.2 Infinite-horizon value iteration

In the infinite-horizon case, there generally is an infinite number of joint beliefs, therefore the $\mathrm{Q_{BG}}$ backup for finite horizon translates to the following backup operator for the infinite horizon:

$$H_{\mathrm{B}}Q(b,\mathbf{a})=R(b,\mathbf{a})+\gamma\max_{\beta_{\langle b,\mathbf{a}\rangle}}\sum_{\mathbf{o}}P(\mathbf{o}|b,\mathbf{a})Q(b^{\mathbf{ao}},\beta_{\langle b,\mathbf{a}\rangle}(\mathbf{o})), \tag{15}$$

where $b^{\mathbf{ao}}$ is the resulting joint belief after taking joint action $\mathbf{a}$ in b and observing $\mathbf{o}$.

This is a contraction mapping, which means that there is a fixed point, which is the optimal infinite-horizon $\mathrm{Q_{BG}}$-value function $Q^*(b,\mathbf{a})$ [9]. Together with the fact that Q^* for the finite horizon is PWLC, this means we can approximate $Q^*(b,\mathbf{a})$ with arbitrary accuracy using a PWLC value function. Note that when using the infinite-horizon value function, we will actually have to evaluate (9), as we cannot cache all maximizing $\beta^*_{\langle\vec{\theta}^{t-1},\mathbf{a}\rangle}$ as in the finite-horizon case.

4.3 Approximate $\mathrm{Q_{BG}}$-value iteration

Now that we have presented how we can backup joint beliefs with $\mathrm{Q_{BG}}$-value functions, we can adapt POMDP value-iteration algorithms to compute $\mathrm{Q_{BG}}$-value functions. A major cause of intractability of exact POMDP solution methods is their aim of computing the optimal action for every possible belief point in $P(S)$. A natural way to sidestep this intractability is to settle for computing an approximate solution by considering only a finite set of belief points. One such point-based POMDP method is PERSEUS [14], which operates on a large belief set sampled by simulating random trajectories through belief space. Approximate value iteration is performed on this belief set by applying a number of backup stages, ensuring that in each backup stage the value of each point in the belief set is improved; the key observation is that a single backup, whether H_{P} or H_{B}, may improve the value of many belief points. Contrary to other point-based methods, PERSEUS backs up only a (randomly selected) subset of points in the belief set, sufficient for improving the value of each belief point in the set. Adapting PERSEUS to compute $\mathrm{Q_{BG}}$-value functions requires replacing H_{P} with H_{B}.

5. EXPERIMENTS

We present experiments on two problem domains, as a proof of concept of our method. The first problem is Dec-Tiger, a standard test problem for Dec-POMDPs with 2 agents, 2 states, 4 joint observations and 9 joint actions. A detailed description of Dec-Tiger is provided by Nair et al. [7]. The second problem is called OneDoor and is depicted in Fig. 2(Right). The environment contains two simulated robots, one of which starts in location A and has to

---

practice, the $\beta^*_{\langle\vec{\theta}^{t-1},\mathbf{a}\rangle}$ will be cached and evaluation of (9) is not necessary anymore in the online phase.

Problem	h	$V_{\mathrm{P}}(b_0)$	$V_{\mathrm{QBG}}(b_0)$
Dec-Tiger	5	26.80	10.68
Dec-Tiger	10	60.29	34.59
Dec-Tiger	15	93.59	53.16
OneDoor	10	0.140	0.0796

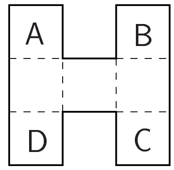

Figure 2: Left: values of initial belief b_0 for several problems with horizon h. V (b_0) indicates the PERSEUS POMDP value of b_0, and V (b_0) the value of b_0 for the Q -value function. Right: the OneDoor environment, with 7 locations for each agent.

move to B, and the other starts in location C and has to reach D. The system receives of reward 1 for each robot that reaches its goal location, but it is penalized with a reward of -10 if both robots occupy the same location. The robots have to learn that one should wait for the other, in order to avoid bumping into each other. They can move one square in each direction ($|\mathcal{A}| = 16$), but only with 80% accuracy. The remaining probability mass is distributed equally over the other three actions, and when a robot attempts to exit the environment it remains stationary instead. Each robot can observe whether there are one or less walls surrounding, or two or more ($|\mathcal{O}| = 4$). Its sensors are not noisy, but the environment is still partially observable, as many locations map to the same observation, and as each robot cannot observe the other robot.

We used PERSEUS to compute Q_{BG}-value functions for these problems, as well as POMDP value functions. In order to compute finite-horizon policies with PERSEUS, which essentially is an infinite-horizon technique, we extended the state description of each problem with the time step of the decision problem. We also added an absorbing state, to which the agents always transition when they reach the horizon. They cannot leave the absorbing state, and in which each joint action gathers zero reward. The discount rate γ is set to 1 for Dec-Tiger, and to 0.95 for OneDoor. We used unique beliefs in each belief set, and limited its size to 100.

Fig. 2(Left) shows values of the initial joint belief b_0 for both value functions, which indicates the expected control quality of the computed policy. As expected, the POMDP value $V_{\mathrm{P}}(b_0)$ is higher than the Q_{BG}-value $V_{\mathrm{QBG}}(b_0)$, as the POMDP solution uses instantaneous observation sharing, which requires the agents to synchronize at every time step. However, we can see that the proposed Q_{BG} solution produces adequate results, without requiring the agents to wait for incoming messages.

Note that given the one-step communication delay, the Bayesian game solution is optimal, but approximations will be introduced due to the approximate nature of PERSEUS. However, for the Dec-Tiger problem and a horizon of 5, we were able to compute the exact values of b_0 using the $\mathrm{Q}_{\mathrm{POMDP}}$ and Q_{BG} methods as described in [8]. They are 26.81 resp. 10.68, which are practically identical to the values reported in Fig. 2(Left). For larger horizons computing exact solutions quickly becomes impossible due to limited memory space. This highlights the appeal of approximate algorithms like PERSEUS, as they can exploit a sparse joint belief space.

6. CONCLUSIONS AND DISCUSSION

In this paper we considered planning under uncertainty for a multiagent system with delayed communication. In particular we focused on Dec-POMDP settings with a one time-step communication delay. It is known that under the assumption of instantaneous communication this model reduces to a standard POMDP, for which many exact and approximate solution techniques exist, a fact which has been exploited by several Dec-POMDP planning methods [3, 12, 15]. However, as we discussed, instantaneous communication is a strong assumption, as it requires the agents to synchronize at every time step. Therefore we explored planning in Dec-POMDP settings in which communication is delayed by one time step. It allows an agent to act immediately when receiving a local observation, and we assume that the duration of each action is sufficient for the other agents' observations to arrive. We showed how this situation can be modeled by Bayesian games in which the types of the agents are defined by their last private observation. This is in contrast to approaches that model regular (non-communicative) Dec-POMDPs using BGs [3, 8]: the BGs in those approaches have types that correspond to entire (action-) observation histories.

We showed that the Q_{BG}-value function is the optimal payoff function for the BGs resulting from the one time-step delayed communication. Using the PWLC property of the Q_{BG}-value function, we demonstrated how we can define value iteration for planning in this delayed communication Dec-POMDP setting. Next, we discussed that for the infinite-horizon case, we can approximate the optimal Q_{BG}-value function arbitrarily well with a PWLC value function. We extended an approximate POMDP solver to compute approximate Q_{BG}-value functions, and used it to demonstrate the viability of our approach in some experiments.

In [8] we proposed the Q_{BG}-value function as an upper bound to the optimal value function for communication-free Dec-POMDPs. In this paper, we proved that the Q_{BG}-value function also provides an optimal solution for Dec-POMDPs with a one time-step communication delay. However, it has been brought to our attention that in the automatic control literature the problem of decentralized control in a delayed-communication setting has also been tackled [16, 5, 1], where it is known as a one step delay sharing (OSDS) information pattern[2]. In particular, it was shown that under OSDS there is a PWLC value function [16] and that a dynamic programming formulation for the problem exists [5]. We were not aware of this work, however, we believe that the Bayesian game perspective presents a unified view for decentralized decision making with no communication and with one or more steps of delayed communication. Also, we anticipate that it allows for several promising extensions, as we will present shortly.

The techniques that we presented in this paper extend beyond the delayed-communication setting, as several Dec-POMDP solution techniques exist that employ a value function over joint beliefs as a heuristic. Instead of using $\mathrm{Q}_{\mathrm{MDP}}$ [3, 15, 8] or $\mathrm{Q}_{\mathrm{POMDP}}$ [12, 8], the Q_{BG}-value function will provide a better approximation of the true optimal Q-value function of the general Dec-POMDP, although at a higher computational cost [8].

Future work will consider approximate methods for solv-

[2] We thank the reviewers for pointing out this related work.

ing the BGs, as optimally solving these can be expensive for problems with many observations. For instance solving them using an alternating-maximization approach [7, 3] would be a logical step. We will also be considering situations in which the communication is delayed more than one time step, in which case each agent's type will be a local observation history instead of just the last received local observation.

Another direction of future research is the situation in which the agents decide online whether or not to wait for other the transmitted local observations of the other agents. By considering the expected value difference between the POMDP and the $\mathrm{Q_{BG}}$ value of a belief, each agent can determine at each time step the relative benefit of waiting for all other local observations. In particular, we can define a new backup operator that compares the resulting vectors from H_{P} and H_{B}, and selects which one to use based on their value difference and a task-dependent penalty for waiting. The benefit of such an approach would be that, when the other agents' observations are not very relevant at a particular time step (in terms of expected value loss), we act quickly according to the $\mathrm{Q_{BG}}$-value function. However, in situations that require tight coordination each agent will choose to wait, and act on the joint observation.

7. ACKNOWLEDGMENTS

We would like to thank the reviewers for their valuable comments. F.A. Oliehoek is supported by the Interactive Collaborative Information Systems (ICIS) project, funded by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024. This work was supported by Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the POS_Conhecimento Program that includes FEDER funds.

8. REFERENCES

[1] M. Aicardi, F. Davoli, and R. Minciardi. Decentralized optimal control of Markov chains with a common past information set. *IEEE Transactions on Automatic Control*, 32(11):1028–1031, 1987.

[2] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, 2002.

[3] R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun. Approximate solutions for partially observable stochastic games with common payoffs. In *Proc. of Int. Joint Conference on Autonomous Agents and Multi Agent Systems*, 2004.

[4] E. A. Hansen, D. Bernstein, and S. Zilberstein. Dynamic programming for partially observable stochastic games. In *Proc. of the National Conference on Artificial Intelligence*, 2004.

[5] K. Hsu and S. I. Marcus. Decentralized control of finite state Markov processes. *IEEE Transactions on Automatic Control*, 27(2):426–431, 1982.

[6] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[7] R. Nair, M. Tambe, M. Yokoo, D. Pynadath, and S. Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *Proc. Int. Joint Conf. on Artificial Intelligence*, 2003.

[8] F. A. Oliehoek and N. Vlassis. Q-value functions for decentralized POMDPs. In *Proc. of Int. Joint Conference on Autonomous Agents and Multi Agent Systems*, 2007.

[9] F. A. Oliehoek, N. Vlassis, and M. T. J. Spaan. Properties of the Q_{BG}-value function. Technical Report IAS-UVA-07-03, Informatics Institute, University of Amsterdam, 2007.

[10] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, July 1994.

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

[12] M. Roth, R. Simmons, and M. Veloso. Reasoning about joint beliefs for execution-time communication decisions. In *Proc. of Int. Joint Conference on Autonomous Agents and Multi Agent Systems*, 2005.

[13] E. J. Sondik. *The optimal control of partially observable Markov processes*. PhD thesis, Stanford University, 1971.

[14] M. T. J. Spaan and N. Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.

[15] D. Szer, F. Charpillet, and S. Zilberstein. MAA*: A heuristic search algorithm for solving decentralized POMDPs. In *Proc. of Uncertainty in Artificial Intelligence*, 2005.

[16] P. Varaiya and J. Walrand. On delayed sharing patterns. *IEEE Transactions on Automatic Control*, 23(3):443–445, 1978.