Best Response Bayesian Reinforcement Learning for Multiagent Systems with State Uncertainty

Frans A. Oliehoek
Informatics Institute, University of Amsterdam
DKE, Maastricht University
f.a.oliehoek@uva.nl

Christopher Amato
CSAIL
MIT
camato@csail.mit.edu

ABSTRACT

It is often assumed that agents in multiagent systems with state uncertainty have full knowledge of the model of dynamics and sensors, but in many cases this is not feasible. A more realistic assumption is that agents must learn about the environment and other agents while acting. Bayesian methods for reinforcement learning are promising for this type of learning because they allow model uncertainty to be considered explicitly and offer a principled way of dealing with the exploration/exploitation tradeoff. In this paper, we propose a Bayesian RL framework for best response learning in which an agent has uncertainty over the environment and the policies of the other agents. This is a very general model that can incorporate different assumptions about the form of other policies. We seek to maximize performance and learn the appropriate models while acting in an online fashion by using sample-based planning built from powerful Monte-Carlo tree search methods. We discuss the theoretical properties of this approach and experimental results show that the learning approaches can significantly increase value when compared to initial models and policies.

1. INTRODUCTION

While there has been a large amount of recent work and success in planning problems for multiagent systems with state uncertainty [6, 14, 7, 26, 20], in many domains, agents will not have access to a full model of the domain or the ability to coordinate to compute a policy. Instead, more researchers have begun to consider agents that can adapt to an environment and the actions of the other agents (e.g., in ad hoc teamwork [30]). For example, if a robot is sent to Mars for exploration or construction, factors such as gravity and soil composition may have effects that are hard to predict and older robots may already be working on the mission. As such, our goal is to endow our agent with the capability to learn about both the dynamics of its environment, as well as the behavior of possible teammates. In addition, it should optimize its behavior with respect to prior knowledge and partial (and potentially noisy) sensor information it receives.

Bayesian reinforcement learning methods are a promising manner to conduct this type or learning because they allow us to incorporate prior knowledge and, in principle, give an optimal exploration/exploitation trade-off with respect to this prior belief. In many real-world situations, the true model may not be known, but a prior can be expressed over a class of possible models. This belief over modes can be used to choose actions that will maximize expected value, reducing uncertainty as needed to improve performance.

Unfortunately, in multiagent systems, only a few Bayesian RL methods have been considered. For example, the Bayesian RL framework has been used in stochastic games [9] and factored Markov decision processes (MDPs) [32]. While either model is intractable to solve optimally, both approaches generate approximate solutions (based on the value of perfect information) which perform well in practice. Both approaches also assume the state of the problem is fully observable (or can be decomposed into fully observable components). This is a common assumption to make, but many real-world problems have partial observability due to noisy or inadequate sensors as well as a lack of communication with the other agents. In fact, very few multiagent RL approaches of any kind consider partially observable domains (notable exceptions, e.g., [1, 10, 21]), and only our previous work [3] falls in the category of Bayesian RL.

In this previous paper we proposed to fill this void by proposing two approaches for Bayesian RL for multiagent systems with state uncertainty [3]: one approach considers a team of communicating agents, the other approach models the problem from the perspective of a single agent that tries to learn a best response in a team of other agents. In this work, we significantly generalize the latter approach by presenting a much more general framework of best-response models (BRMs). Using these general best-response models, we explore uncertainty about the environment with either 1) all other agents' models being fixed and known, or 2) all other agents' policies being unknown. To capture these uncertainties, we build upon the Bayes Adaptive partially observable Markov decision process (BA-POMDP) framework [23, 24]. We discuss how the resulting BA-BRM has the earlier history-based representation of [3] as its special case, and how it can directly also work on policies represented as finite-state controllers. As any BA-POMDP, the BA-BRM can be characterized and solved as a (possibly infinite state) POMDP, but the resulting problem is typically intractable to solve optimally. As an alternative, we present a novel sample-based planning method, called root-sampling Bayes-adaptive POMCP (RS-BA-POMCP), based on Monte-Carlo tree search which is much more scalable while retaining convergence guarantees. This framework represents a general method for reasoning about other agents and learning in multiagent domains with state uncertainty, incorporating available knowledge in a wide range of scenarios.

Appears in *The Ninth Annual Workshop on Multiagent Sequential Decision-Making Under Uncertainty* (*MSDM-2014*), held in conjunction with *AAMAS*, May 2014, Paris, France.

2. BACKGROUND

This section provides a concise description of POMDPs, which we will use the basis for our best-response methods, as well as previous work on Bayesian RL for POMDPs.

2.1 POMDPs

POMDPs represent a framework for planning under uncertainty and partial observability [18]. Formally, a POMDP is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{Z}, D, R, L \rangle$ with: $\mathcal{S}$, a finite set of states with designated initial state distribution b_0; $\mathcal{A}$, a finite set of actions; $\mathcal{Z}$, a finite set of observations; D the dynamics function specifying $D(s',z|s,a)$ the probability of a next state s' and observation, given a current state s and action a; R, a reward function: $R(s,a)$, the immediate reward for being in state s and taking action a; L, the horizon.

We note that we deviate from the standard formulation of a POMDP [18], which specifies D in a factored way, using a transition and observation model: $D(s',z|s,a) = T(s'|s,a)O(z|a,s')$. We point out the our formulation is a strict generalization: any POMDP can be represented using the D formulation. Moreover, we will encounter POMDPs that can not be represented in a factored fashion.

Most research concerning POMDPs has considered the task of *planning*: given a full specification of the model, determine an optimal policy (e.g., [18, 27]). However, in many real-world applications, the model is not (perfectly) known in advance, which means that the agents have to learn about their environment during execution. This is the task considered in *reinforcement learning (RL)* [31].

2.2 Bayesian RL for POMDPs

A fundamental problem in RL is that it is difficult to decide whether to try new actions in order to learn about the environment, or to exploit the current knowledge about the rewards and effects of different actions. In recent years, Bayesian RL methods have become popular because they potentially can provide a principled solution to this exploration/exploitation trade-off [12, 13, 22, 34].

In particular, we consider the framework of Bayes-Adaptive POMDPs [23, 24]. Due to lack of space, we are forced to give only a very concise overview; we refer to the original papers and [3] for more detail. This framework utilizes Dirichlet distributions to model uncertainty over transitions and observations[1] (typically assuming the reward function is chosen by the designer and thus known). In particular, if the agent could observe both states and observations, it could maintain a vector χ with the counts of the occurrences for all $\langle s, a, s', z \rangle$ tuples. We write $\chi_{sa}^{s'z}$ for the number of times that s,a is followed by s',z.

While the agent cannot observe the states and has uncertainty about the actual count vector, this uncertainty can be represented using the regular POMDP formalism. That is, the count vector is included as part of the hidden state of a special POMDP, called BA-POMDP. Formally, a BA-POMDP is a tuple $\langle \mathcal{S}_{BP}, \mathcal{A}, \mathcal{Z}, D_{BP}, R_{BP}, L \rangle$ with some modified components in comparison to the POMDP.

First, we point out that actions and observations remain the same as the case where there is no uncertainty about the transition and observation function (i.e., the same as a regular POMDP). However, the state of the BA-POMDP now includes Dirichlet parameters: $s_{BP} = \langle s, \chi \rangle$. The reward model remains the same (since it is assumed to be known), $R_{BP}(\langle s,\chi \rangle,a) = R(s,a)$. The dynamics functions, D_{BP}, however, are modified when compared to a POMDP. Given χ we can define the expected transition as $D_\chi(s',z|s,a) = \mathbf{E}[D(s',z|s,a)|\chi] = \frac{\chi_{sa}^{s'z}}{\sum_{s'z} \chi_{sa}^{s'z}}$. These expectation can now be used to define the transitions for the BA-POMDP. If we let $\delta_{sa}^{s'z}$ denote a vector of the length of χ containing all zeros except for the position corresponding to $\langle s,a,s',z \rangle$ (where it has a one), and if we let $\mathbb{I}_{a,b}$ denote the Kronecker delta that indicates (is 1 when) $a = b$, then we can define D_{BP} as $D_{BP}(s',\chi',z|s,\chi,a) = D_\chi(s',z|s,a)\mathbb{I}_{\chi',\chi+\delta_{sa}^{s'z}}$.

Remember that these counts are not observed by the agent, since that would require observations of the state. The agent can only maintain belief over these count vectors. Still, when interacting with the environment, *the ratio of the true—but unknown—count vectors will converge to coincide with the true transition and observation probabilities in expectation.* It is important to realize, however, that this convergence of count vector ratios does not directly imply learnability by the agent: even though the ratio of the count vectors of the true hidden state will converge, *the agent's belief over count vectors might not.*

3. BA-BRM : A GENERAL FORMULATION

In this paper, we are interested in problems with multiple agents that each receive their own observations. In some such settings, agents can communicate to share their experiences about the environment around them, which means that the problem of deciding how to act can be reduced to a (larger) single-agent problem. This situation is captured by the multiagent POMDP (MPOMDP) model. Due to the reduction to a single-agent problem, it is possible to extend the BA-POMDP to such settings, yielding the BA-MPOMDP which can deal with unknown environment in team settings [3, 4]. In many real-world scenarios, however, agents must learn about the environment and any other agents during execution based solely on their own local information. This is true in any situation where instantaneous, noise-and cost-free communication is not possible, practical or sensible. For instance, in competitive settings, this type of communication often does not make sense, and even in strictly cooperative settings there are often communication limitations.

In this section, we describe a different way of applying Bayesian RL techniques in multiagent systems by giving a subjective description of the problem. That is, we describe the problem from a single agent's perspective by defining its best-response model (BRM). We propose a Bayesian approach to online learning which represents the combined effect of the initial model and policies for the other agents using priors and updates probability distributions over these models as the agent acts in the real world.

Throughout the remainder of the paper, we consider an interactive setting from the perspective of a single agent i. This agent will interact with a number of other agents j (collectively also indicated as $-i$). We describe a very general model of other agents, and how that in general can be used to compute a best response. Then we extend this framework to learning settings where the protagonist agent is uncertain about the working of the environment or the other agents.

3.1 The Behavior of Other Agents

In order to describe other agents j, we introduce a very

[1] [23, 24] follow the standard T,O POMDP formalism, but we will give a description that matches with our D formalism.

general model for their behavior: A *model* m_j for agent j is a tuple $m_j = \langle \mathcal{A}_j, \Omega_j, \mathcal{I}_j, \pi_j, \beta_j, I_j \rangle$ with $\mathcal{A}_j$ the set of actions, Ω_j the set of observations, $\mathcal{I}_j$ a set of internal states, a policy $\pi_j : \mathcal{I}_j \to \Delta(\mathcal{A}_j)$, a belief update function $\beta_j : \mathcal{I}_j \times \mathcal{A}_j \times \Omega_j \to \Delta(\mathcal{I}_j)$ and I_j the current internal state of the agent. We will also refer to $f_j = \langle \mathcal{A}_j, \Omega_j, \mathcal{I}_j, \pi_j, \beta_j \rangle$ as a frame for agent j, such that a model is given by $m_j = \langle f_j, I_j \rangle$.

Notice that this definition does not restrict the set of internal states. Moreover, policies and belief update functions can be represented as look-up tables, but also be computational procedures themselves. Therefore, the formulation is extremely general, including most notions of agent, and in particular types such as MDP- or POMDP-based agents.

Also note that we treat the models of different agent as 'independent', i.e., each other agent j is supposed to have a model m_j that does not depend on the model of others. The resulting behavior is decentralized, but not necessarily independent. In contrast, by using rich internal state spaces and by making use of the correlations between observations agents receive, very complicated team behaviors can be captured. We believe that this is not different from teamwork as it would arise in a human team of, e.g., soccer players.

3.2 Computing a Best-Response

Models such as the ones introduced above can be used by agents in all kinds of environments. In the remainder of this paper, we will focus on a protagonist agent i that is situated in a POMDP-like environment. In particular, we define a *multiagent environment (MAE)* for agent i as a tuple $MAE_i = \langle \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^n, T, \{\mathcal{Z}_i\}_{i=1}^n, O, R_i \rangle$ where: $\mathcal{S}$ is the set of states of the environment; $\{\mathcal{A}_i\}_{i=1}^n$ is the collection of the sets of individual actions of the agents; T is the transition function that specifies $T(s'|s,a_i,a_{-i})$; $\{\mathcal{Z}_i\}_{i=1}^n$ is the collection of the sets of individual observations of the agents; O is the observation function that specifies $\Pr(z_i,z_{-i}|a_i,a_{-i},s')$; R_i is the reward function that specifies $R_i(s,a_i,a_{-i})$. That is, a MAE is a partially observable stochastic game (POSG) [16] that only specifies the reward component for the protagonist agent i.

In this setting, we will assume that agent i will have knowledge about the frame of the other agents f_{-i} although it is still uncertain about their internal states I_{-i}. Let us write $\mathcal{M}_{-i} = \{f_{-i}\} \times \mathcal{I}_{-i}$ for the set of models consistent with f_{-i}. Given $\mathcal{M}_{-i}$ and given a MAE_i, agent i can compute how to act optimally by constructing and solving an augmented POMDP [19]. Here we give a slight generalization of the formulation of [19] that extends to our general notion of agent. We will refer to this as a *best-response model (BRM)*. Formally, a best-response model for agent i is a tuple $BRM_i(MAE_i, \mathcal{M}_{-i}) = \left\langle \bar{\mathcal{S}}, \mathcal{A}_i, \mathcal{Z}_i, \bar{D}_i, \bar{R}_i \right\rangle$ that consists of a set of states $\bar{s} = \langle s, I_{-i} \rangle$, such that $\bar{\mathcal{S}} = \mathcal{S} \times \mathcal{I}_{-i}$; the sets of actions $\mathcal{A}_i$ and observations $\mathcal{Z}_i$; $\bar{D}_i$, the dynamics function $\bar{D}_i(\bar{s},a_i,\bar{s}',z_i)$, a combined transition/observation function that specifies:

$$\bar{D}_i(\bar{s}', z_i|\bar{s}, a_i) = \sum_{a_{-i}} \sum_{z_{-i}} T(s'|s,a) O(z|a,s') \prod_{j \neq i} \beta_j(I'_j|I_j,a_j,z_j) \pi_j(a_j|I_j); \quad (3.1)$$

$\bar{R}_i$, a reward function

$$\bar{R}_i(\bar{s},a_i) = \bar{R}_i(s,I_{-i},a_i) = \sum_{a_{-i}} R_i(s,a) \prod_j \pi_j(a_j|I_j). \quad (3.2)$$

Note that (3.1) shows that $\bar{D}_i$ in general cannot be factored.[2]

Since a BRM is a POMDP, we can define beliefs $\bar{b}_i(\bar{s})$ and value functions $V_i(\bar{b}_i)$, in the usual way. As such, the BRM may be solved with standard POMDP methods [18, 27, 29].

3.3 Transition & Observation Uncertainty

Here we consider the setting where agent i is uncertain about the transition and observation function, but knows the model for all other agents. That is, T,O in MAE_i are unknown, but $\mathcal{M}_{-i}$ (the set of models consistent with frames f_{-i}) is given. The direct consequence is that BRM_i is also unknown. However, since the BRM is a POMDP, we can incorporate uncertainty about the transition and observation model by transforming it to its Bayes' adaptive variant.

We define a *BA-BRM* as a tuple $\breve{M}_i = \left\langle \breve{\mathcal{S}}, \mathcal{A}_i, \mathcal{Z}_i, \breve{D}_i, \breve{R}_i, h \right\rangle$ where $\mathcal{A}_i$ and $\mathcal{Z}_i$ are the sets of actions and observations of agent i; $\breve{\mathcal{S}}$ is the set of states $\breve{s} = \langle \bar{s}, \chi \rangle = \langle s, I_{-i}, \chi \rangle$ where χ is the vector of counts $\chi_{\bar{s}a}^{\bar{s}'z}$ counting how often $\bar{s},a_i$ was followed by $\bar{s}',z_i$; $\breve{D}$ is the combined transition and observation function (see below); $\breve{R}$ is the reward function defined as $\breve{R}_i(\breve{s},a_i) = \bar{R}_i(\bar{s},a_i)$ via (3.2); h, the horizon. The dynamics function of the BRM (3.1) is transformed to a variant that takes into account the uncertainty as follows:

$$\breve{D}(\breve{s}', z_i|\breve{s}, a_i) = D_\chi(\bar{s}', z_i|\bar{s}, a_i) \mathbb{I}_{\chi', \chi + \delta_{\bar{s}a}^{\bar{s}'z}} \quad (3.3)$$

where $\delta_{\bar{s}a}^{\bar{s}'z}$ is a vector with entries for all possible transitions (all of these are 0 except for the entry for $(\bar{s},a_i) \to (\bar{s}',z_i)$), and where

$$D_\chi(\bar{s}', z|\bar{s}, a) \triangleq \frac{\chi_{\bar{s}a}^{\bar{s}'z}}{\sum_{\bar{s}',z} \chi_{\bar{s}a}^{\bar{s}'z}}. \quad (3.4)$$

is the expected transition-observation function induced by χ.

Remember that in a partially observable environment, χ usually cannot be observed directly. That is why χ is part of the hidden state. Nevertheless, we can think about how we would expect the true (unobserved) count vector to evolve over time. Let us denote the true count vector at stage t as $X(t)$, and note that $X(t)$ is a random variable that depends on the actual dynamics and behavior of the other agents. Even though the counts will continue to grow with time, it is easy to show that the induced ratio $D_\chi(\bar{s}',z|\bar{s},a)$ converges in probability to the true probability as defined by (3.1):

$$\forall_{\bar{s},a,\bar{s}',z} \quad D_{X(t)}(\bar{s}', z|\bar{s}, a) \xrightarrow{p} \bar{D}_i(\bar{s}', z|\bar{s}, a). \quad (3.5)$$

As was the case for the BRM, the BA-BRM is just a POMDP, which means that also in this case, the usual POMDP theory holds. The difficulty, however, is that the number of states $\breve{s}$ is (countably) infinite, which means that exactly representing the beliefs $\breve{b}_i$ and value function becomes impossible. Fortunately, since a BA-BRM is a special case of BA-POMDP, all the BA-POMDP theory holds even with the inclusion of other agent histories as part of the state information. Therefore, as for the BA-POMDPs, there are two ways to overcome this difficulty: constructing a finite ϵ-optimal approximate POMDP, or using sample-based (i.e., 'particle-based') representations of the belief in combination with sample-based planning. Note that, given $\breve{s} = (\bar{s}, \chi)$ and

[2] However, in case of independent observations, $O(z|a,s') = \prod_{i=1}^n O_i(z_i|a_i,s')$, it can be split in the more common formulation that specifies a separate transition and observation model: $\bar{O}(a_i,\bar{s}',z_i) = O(z_i|a_i,s')$, $\bar{T}(\bar{s},a_i,\bar{s}') = \sum_{a_{-i}} \sum_{z_{-i}} T(s'|s,a) \prod_{j \neq i} O(z_j|a_j,s') \beta_j(I'_j|I_j,a_j,z_j) \pi_j(a_j|I_j)$.

a_i, it is trivial to sample a next state $\breve{s}' = (\bar{s}', \chi')$ and observation z_i by sampling from (3.4):

$$\langle \bar{s}', z_i \rangle \sim D_\chi(\cdot | \bar{s}, a_i) \tag{3.6}$$

and setting $\chi' = \chi + \delta^{\bar{s}' z}_{\bar{s} a}$.

Prior distributions over environment and agent models can be represented as initial count vectors. As is clear from 3.5, the count ratios correspond to (should converge to) the true probability $\bar{D}_i(\bar{s}', z | \bar{s}, a)$. If these probabilities can be estimated, the count vectors can be set to ratios representing this quantity. Then, the confidence in this estimation can be reflected in a scaling factor of the various counts. In this way, different aspects of the agent and environment models can have different parameters and confidence based on knowledge of the problem. In the absence of domain knowledge a uniform prior with small counts can be utilized.

3.4 Policy Uncertainty

In the above, we assumed that the policies (i.e., the frames f_{-i}) of other agents are fixed and known. However, a key observation is that *in the construction of the dynamics function of the BA-BRM the knowledge about π_{-i} and β_{-i} is never employed*: the transition-observation $\breve{D}_i$ only depends on the counts via (3.4), but *not* on π_{-i} or β_{-i}.

This does not make the BA-BRM completely independent of π_{-i}, because the reward function $\breve{R}_i$ still depends on π_{-i} via (3.2). However, in many cases, the rewards of the agent might be independent (e.g., if the agents are only coupled through their transitions [5]). Moreover, even in cases where R_i does depend on the actions of other agents, it is still possible to find a new representation of the internal state of the other agents that renders $\bar{R}_i$ independent of π_{-i}. In particular, if we define $\sigma_j \triangleq \langle I_j, a_j \rangle$ as the new internal state, then $\bar{s} = \langle s, \sigma_{-i} \rangle$, and we can directly use that stored action to retrieve the right reward:

$$\bar{R}_i(\bar{s}, a_i) = \bar{R}_i(s, \langle I_{-i}, a_{-i} \rangle, a_i) = R_i(s, a_i, a_{-i}). \tag{3.7}$$

This means that we can have a description of the BA-BRM that is completely independent of π_{-i} or β_{-i}. Of course, knowledge about the aggregate of these functions can still be incorporated through the initial count vectors, but other than that we have no dependence on π_{-i} and β_{-i}. This means that the BA-BRM in fact *can deal with uncertainty regarding the policy of other agents.*

To clarify, consider how this would work for sample-based planning. The first step would make sure that the reward function $\bar{R}_i$ is independent of the other agents (potentially by converting to sequence-form internal states σ). Second, we specify an initial count vector χ_0 that represents our initial belief about the situation. This initial count vector will induce a distribution D_{χ_0} over BRM transitions as specified by (3.4) and therefore aggregates our belief about T, O, π_{-i} and β_{-i} (if our beliefs are accurate, we can specify the counts such that they induce precisely the distribution $\bar{D}_i$ given by (3.1)). Finally, we sample a state $\bar{s}_0$ from our initial belief, form $\breve{s}_0 = \langle \bar{s}_0, \chi_0 \rangle$ and use this to sample an episode using (3.6) and a rollout policy π_i. Repeating this procedure gives us a Monte Carlo estimate of $\breve{V}_i(\pi_i)$.

A different way of including uncertainty about the policy of the other agent is by explicitly absorbing π_{-i}, β_{-i} in the BRM state: $\bar{s} = \langle s, \pi_{-i}, \beta_{-i}, I_{-i} \rangle$. This approach is taken in I-POMDPs [14]. A downside of this approach, however, is that the number of BRM states immediately becomes infinite, making formulations of expected transitions and extensions of BA-POMDPs difficult. Moreover, our analysis here shows that such explicit modeling may not be necessary.

Finally, we point out that, when other agents are adaptive, the assumption of a unknown, but fixed policy is violated. This is no fundamental limitation, though; any adaptive algorithm can be cast as a model m_j provided that we have enough computational power to deal with all the required internal states I_j. Also, in practice methods such as Q-learning have been shown to be effective in such adaptive domains [25, 33]. Therefore, we expect that it might be possible to deal with this issue by, for instance, performing discounting of counts while learning during execution.

3.5 Bounded Loss for Policy Uncertainty

The BA-BRM framework brings another nice insight: it can be used to bound the loss of computing a best response to one particular policy while in fact the agent uses a different one. To show this, we assume that there is a single other agent and that for two policies π^x_j, π^y_j of agent j we have that $\forall_{a_j I_j} \left| \pi^x_j(a_j | I_j) - \pi^y_j(a_j | I_j) \right| \leq \epsilon$. Assume that π^x_j is the true policy of agent j, in that case the count ratios will converge to $\frac{\chi^{\bar{s}' z_i}_{\bar{s} a}(x)}{\sum_{\bar{s}' z} \chi^{\bar{s}' z}_{\bar{s} a}(x)} \xrightarrow{p} \sum_{a_j} \sum_{z_j} D(s', z | s, a_i, a_j) \beta_j(I'_j | I_j, a_j z_j) \pi^x_j(a_j | I_j)$ and similar for π^y_j. If this has happened, we loosely refer to these as converged count vectors χ^*_x and χ^*_y. As such, we can bound

$$\left| \frac{\chi^{\bar{s}' z}_{\bar{s} a}(x)}{\sum_{\bar{s}' z} \chi^{\bar{s}' z}_{\bar{s} a}(x)} - \frac{\chi^{\bar{s}' z}_{\bar{s} a}(y)}{\sum_{\bar{s}' z} \chi^{\bar{s}' z}_{\bar{s} a}(y)} \right| \leq \sum_{a_j} \sum_{z_j} D(s', z | s, a)$$

$$\beta_j(I'_j | I_j, a_j z_j) \left| \pi^x_j(a_j | I_j) - \pi^y_j(a_j | I_j) \right| \leq \epsilon \sum_{a_j} P(\bar{s}', z_i | \bar{s}, a_i, a_j)$$

and therefore

$$\sum_{\bar{s}'} \sum_{z_i} \left| \frac{\chi^{\bar{s}' z_i}_{\bar{s} a}(x)}{\sum_{\bar{s}' z} \chi^{\bar{s}' z}_{\bar{s} a}(x)} - \frac{\chi^{\bar{s}' z_i}_{\bar{s} a}(y)}{\sum_{\bar{s}' z} \chi^{\bar{s}' z}_{\bar{s} a}(y)} \right| \leq = \epsilon \left| \mathcal{A}_j \right| \tag{3.8}$$

THEOREM 1. *Given χ^*_x and χ^*_y, the converged count vectors as described above, for all stages-to-go t, then for any t-steps-to-go policy for agent i, the associated values are bounded:*

$$\max_{s \in S} \left| \alpha_t(s, \chi^*_x,) - \alpha_t(s, \chi^*_y) \right| \leq \frac{\epsilon \left| \mathcal{A}_j \right| (\gamma - \gamma^t) \left\| R \right\|_\infty}{(1 - \gamma)^2} \tag{3.9}$$

PROOF. The proof is analogous to the proof in [3], but makes use of the modified (3.8). □

The implication of this theorem is that if we compute a best response against some policy π^x_j which differs from π^y_j, the true policy used by agent j, by at most ϵ, then the loss in value is bounded by (3.9). This generalizes our previous [3] from history-based best-response representations to the general BRM formulation from Sec. 3. The difference is that the bound here (3.9) additionally depends on the size of the other agent's action set. Also note that this bound, while inspired by the Bayes-adaptive formulation, is a standalone result that only requires that the other agent's behavior can be represented by our general notion of a model, as defined in Sec. 3.1. This stands in contrast to results that appear similar, but pose sharp restrictions on the class of policies that other agent can use [17].

4. SPECIALIZED REPRESENTATIONS

In this section we discuss several specific instantiations of our general BA-BRM formulation. In particular, we will first treat the case where policies of the other agents are specified as mappings from histories to observations. Next, we consider policies represented as finite-state controllers.

4.1 History-Based Policies

A special case of our framework, also investigated in [3], is when other agents remember their full histories of actions and observations and use those for action selection. That is, when the internal state is the action-observation history: $I_j = h_j$. This case is special in that we have that $\bar{s}' = \langle s', h'_{-i} \rangle$ specifies the actions and observations for the other agents. Therefore (3.1) can be written as

$$\begin{aligned}\bar{D}_i(\bar{s}',z_i|\bar{s},a_i) &= T(s'|s,a)O(z|a,s')\pi_{-i}(a_{-i}|h_{-i}) \\ &= O(z_i|a_i,s',h'_{-i})[T(s'|s,a)O(z_{-i}|a,s')\pi_{-i}(a_{-i}|h_{-i})]\end{aligned}$$

where $\pi_{-i}(a_{-i}|h_{-i}) = \prod_{j \neq i} \pi_j(a_j|h_j)$. As a result, it is possible to maintain the count vectors in a factored form. In particular we have counts $\chi^{\bar{s}'}_{\bar{s}a}$ and $\chi^{z'}_{\bar{s}'a}$ (denoted ϕ and ψ in [3]) counting the number of $(\bar{s},a_i,\bar{s}')$ and $(\bar{s}',a_i,z_i)$ separately.

While very general, using action-observation histories as the basis for the policies of other agents has disadvantages. In particular, when the other agents use a deterministic policy, these can be more compactly represented as mappings from observation histories (OHs) to actions. While i is case, $\bar{D}_i$ does not factor, it is still likely that a factored representation of the counts allows for a more compact description, and thus faster learning, that is a good approximation.

4.2 Finite-State Controller Policies

The internal states of the agents can be formalized as nodes in a finite-state controller. In a (Moore) controller, for each agent, the policy π_j, is a mapping from nodes in the controller to actions and the belief update, β_j, is given by the transitions in the controller in the same way as defined in Section 3.1. This controller-based representation is potentially more concise than the history-based representation (which is exponential in the horizon) as it incorporates a bound on memory of the other agents. The approaches discussed above can be directly applied to this model.

5. SOLVING BA-POMDPS

BA-BRMs are special cases of BA-POMDPs, and as such have the same computational difficulties associated with them. While sample-based planners have provided some leverage [24], the typical BA-BRM is a very large BA-POMDP, and further improvements are required. In this section, we propose a novel sample-based planning method, RS-BA-POMCP, that is particularly aimed at solving BA-POMDPs. It is based on POMCP, but performs an even more aggressive form of 'root sampling'. We also prove its convergence to an ϵ-optimal value function.

POMCP [28] is a recent Monte Carlo tree search method for POMDPs that constructs a tree of action-observation histories h, each of which have a particle-based representation of the belief at that history. A key innovation is that it 'root samples' a hidden state s_0, which is subsequently used to sample a trajectory of states. This way, it is possible to incrementally build up the particle-based representations at the nodes, h, and thus it avoids doing expensive belief updates during the Monte Carlo simulations.

Since a BA-BRM is a POMDP, POMCP directly applies, yielding **BA-POMCP**. This method 'root-samples' a full augmented state $\check{s}_0 = \langle s_0,\chi_0 \rangle$, and subsequently, maintains such states $\check{s}_d = \langle s_d,\chi_d \rangle$ throughout the simulation. At every step it samples from the expected dynamic (3.6).

Sampling from (3.6), however, is expensive since at every step of a Monte Carlo rollout, we need to create a new expected dynamic and sample from it. As such, this forms a bottleneck in BA-POMCP. To overcome this problem, we propose **RS-BA-POMCP**. This method performs the normal root sampling of a hidden state $\check{s}_0 = \langle s_0,\chi_0 \rangle$, but at the start of each simulation, it *additionally root samples a single dynamics function D_{root}* from which is sampled for the remainder of the entire simulation, thus avoiding the construction of a new model to sample from at every step.

Remark 1. We point out that while this seems superficially similar to BAMCP [15]—which is POMCP applied to a BA-MDP [12]—it is substantially different. BAMCP root samples a transition model T and uses that throughout, *because a BA-MDP is a POMDP with hidden states $\langle s, T \rangle$ of which the hidden transition model component is stationary.* In a BA-POMDP, however, the hidden state is $\check{s} = \langle s, \chi \rangle$ where χ is not stationary, which means that the transition (dynamics) function is different for all states sampled in a state trajectory.

While RS-BA-POMCP is more efficient, it is not directly clear that the method is still sound, i.e., whether it still converges to an ϵ-optimal value function. Here we show that it is. The main steps in this proof are similar to the proof in POMCP. We point out however, that the technicalities of proving the components are far more involved. Due to lack of space we will defer a detailed presentation to an extended version of this paper.

We use the following notation: h_d is an action-observation history at depth d of a simulation, $h_d = (a_0,z_1,\ldots,a_{d-1},z_d)$. H_d is a *full* history at depth d, $H_d = \langle s_{0:d},h_d \rangle$. $H_d^{(i)}$ the full history at depth d corresponding to simulation i. $\chi(H_d)$ denotes the vector of counts at simulated full history H_d. Note that if χ_0 is the count vector at the root of simulation, we have that $\chi(H_d) = \chi_0 + \Delta(H_d)$, with $\Delta(H_d)$ the vector of counts of all (s,a,s',z) quadruples occurring in H_d.

The *full-history BA-POMDP rollout distribution* is the distribution over full histories of a BA-POMDP, when performing Monte-Carlo simulations according to a policy π is

$$P^\pi(H_{d+1}) = D_{\chi(H_d)}(s_{d+1},z_{d+1}|a_s,s_{d+1})\pi(a_d|h_d)P^\pi(H_d) \tag{5.1}$$

with $P^\pi(H_0) = b_0(\langle s_0,\chi_0 \rangle)$ the belief at the root.

The *full-history RS-BA-POMDP rollout distribution* is the distribution over full histories of a BA-POMDP, when performing Monte-Carlo simulations according to a policy π in combination with root sampling of the transition and observation models. This distribution, for a particular stage d, is given by $\tilde{P}^\pi_K(H_d) \triangleq \frac{1}{K_d} \sum_{i=1}^{K_d} \mathbb{I}_{H_d H_d^{(i)}}$, where K is the number of simulations in the empirical distribution, $H_d^{(i)}$ is the history specified by the i-th simulation at stage d.

LEMMA 1. *The full-history RS-BA-POMDP rollout distribution converges in probability to full-history BA-POMDP*

rollout distribution:

$$\forall_{H_d} \quad \tilde{P}^{\pi}_{K_d}(H_d) \xrightarrow{p} P^{\pi}(H_d). \tag{5.2}$$

PROOF. The proof of this lemma is substantialy more complex than the corresponding lemmas in [28, 15]. A proof sketch is provided in the appendix. □

THEOREM 2. *Given a suitable exploration constant (e.g., $c > \frac{Rmax}{1-\gamma}$), the RS-BA-POMCP converges in probability to an ϵ-optimal value function: $V(\langle s,\chi\rangle,h) \xrightarrow{p} V^*_\epsilon(\langle s,\chi\rangle,h)$.*

PROOF. This proof is identical to the theorems in [28, 15]. We focus on [28]: since a BA-POMDP is a POMDP, the guarantees of the theorem hold if the same requirements are met. Lemma 1 from [28] holds for all POMDPs, Lemma 2 from [28] corresponds to our Lemma 1. □

Intuitively, given the same rollout policy, the RS-BA-POMCP simulations converge to the same rollout distribution as does BA-POMCP, which means that the methods evaluate nodes in the same way and therefore the theoretical guarantees extend to RS-BA-POMCP.

6. EXPERIMENTAL EVALUATION

To empirically verify the effectiveness of different versions of the BA-BRM model in dealing with uncertain interactive environments, we implemented a version of RS-BA-POMCP where the protagonist agent interacts with an environment over a number of episodes ($N_{episodes}$), each of a finite length L. At the end of each episode, the protagonist's belief over states is reset to the initial belief, but the belief over count vectors is maintained. That way, the agent learns across all the episodes. In each stage of each episode, RS-BA-POMCP performs a number of simulations ($N_{simulations}$) in order to estimate the values and choose an action. For illustration, we show the value achieved as the number of learning episodes increases. The value is averaged over a number of runs (N_{runs}). Note that at the start of each run the count vectors are reset, so there is no learning across runs — these are only to determine average values. These experiments were run on a 2.5 GHz Intel i7 using a maximum of 8GB of memory. We evaluate our approach by performing online learning using the common decentralized tiger benchmark [19] and a meeting in a grid domain where the agent must find and stay in the same grid square as another agent [8].

Two-agent tiger problem. In order to convert the tiger problem to a best-response problem, we fix the policy of one agent to an observation-history based policy. We assumed history-based policies for the other agent and set initial count vectors to reflect high confidence in the true transition function and low confidence in a near uniform observation function. The observation prior counts were 100 for any action other than listening, and when listening, 6 for both agents observing the correct door, 5 for only one agent observing the correct door and 4 for both agents observing the incorrect door. We performed two experiments: one in which the protagonist agents is uncertain about the dynamics of the problem, but knows what the model of the other agent is, and one in which it also has uncertainty about that model. When the other agent policy was not assumed to be known, a uniform prior (all 1's) was used over all history-based policies of the given horizon.

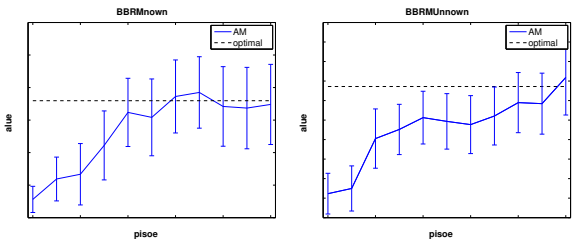

Figure 1: Value per episode in two-agent tiger problem for the BA-BRM with known and unknown other agent policies

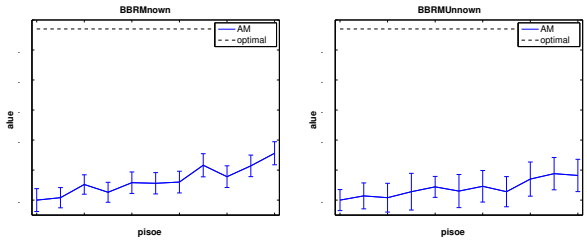

Figure 2: Value per episode in the grid problem for the BA-BRM method with known and unknown controllers for other agent policies

In the first setting—only uncertainty about the dynamics, called 'BRM-known'— the other agent uses an individual policy that corresponds to the optimal joint policy (it listens at each step until the same observation had been heard twice in a row). Since there is no policy uncertainty, an internal state of OHs $I_j = \vec{o}_j$ suffices. For this setting, we have the agents interacting over 250 episodes, each of $L = 3$. In order to take an action, the protagonist agent performs RS-BA-POMCP using $N_{simulations} = 10000$ simulations. The results are averaged over $N_{runs} = 100$ runs of this experiment. The results for this setting (mean return for a particular episode, with error bars based on standard error) are shown in Figure 1(left). Due to the initial prior, the first policy involves listening in most cases, but a much better policy is quickly learned. While significant noise remains in the values due to the large differences in rewards that are possible, the value converges to be near the optimal value in this version of the problem, 5.19.

In the second setting—an unknown model of the other agent— the other agent uses a policy of listening no matter what observation was seen. In this setting, the protagonist agent models the internal state of the other agent using its action-observation history: $I_j = \vec{\theta}_j$, since that includes the action as discussed in Sec. 3.4.[3] For this setting $N_{simulations} = 500$ simulations were used and averaged over $N_{runs} = 50$ runs. The value determined by the initial prior reflects listening on every step and after 250 episodes it is near the optimal value, which in this case is -0.28. Again, we see that the protagonist agent using RS-BA-POMCP is able to learn a near optimal policy in relatively few episodes.

Grid problem. We also consider a 2×2 grid with two agents, similar to the problem described in [2], but with more observations: the agents can observe their own position and whether the other agent is also there. Rewards in this problem are zero unless both agents share the same square,

[3]This follows the description from [3]. However, we point out that in fact the analysis here indicates that a more compact description $I_j = \langle \vec{o}_j, a_j \rangle$ also suffices.

so the goal for our agent is to navigate to the other agent's location as quickly as possible and stay in the same square as the other agent. We consider episodes of length $L = 5$. In order to convert to a best-response problem, we assume the other agent uses a three node controller that specifies the following behavior: the agent moves up until in the top row, then moves left until in the top-left corner, then stays.

In this problem, we set the prior so the agent had high confidence in its own (true) transitions and observations of its own location (1000 times the true probabilities), but was uniform over the other agent transition and observation function (all 1's) and with regard to the probability of observing the other agent. These priors reflect the idea of coordinating with an unknown agent without much prior knowledge of that agent.

Again we consider the same two settings (known and unknown policy of the other agent). For the known policy case, the internal state is the controller node: $I_j = n_j$. For the unknown policy setting we expanded the internal state description with the last action and observation of the other agent: $I_j = \langle n_j, z_j, a_j \rangle$.[4] For the unknown case a uniform prior was used over three node controllers for the other agent. For both settings RS-BA-POMCP used $N_{simulations} = 2000$ simulations for action selection. Results are averaged over $N_{runs} = 10$ runs. Fig. 2 shows the results for this domain. Because the prior had very little information about the other agent, the policy began very far from the optimal value of 1.94. As expected, learning is faster in the case of a known other agent policy, but in both cases, learning is observed. While the value after 250 episodes is not close to the optimal value, it is promising that agents are able to learn in such domains with limited initial knowledge and a noisy and partial learning signal.

7. RELATED MODELS

Work on POSGs [16] typically takes a game-theoretic (i.e., equilibrium) perspective: both the POSG, as well as the fact that all the agents are rational are assumed common knowledge. In contrast, while the MAE builds upon the same way of specifying the components of the problem, it does not impose any assumptions of rationality.

The BRM is closely related to interactive POMDPs [14]. In particular, $\bar{\mathcal{S}}$ is referred to as the 'interactive state space' in the context of I-POMDPs, and the optimal value function for I-POMDPs [14] is essentially the same as $V_i(\bar{b}_i)$. A difference in formulation is that the BRM does not include the set of *joint* actions, but is strictly a POMDP. A bigger difference is that, as with POSGs, work on I-POMDPs tends to focus on other agents that are assumed to be (approximately) rational. For instance, most work on I-POMDPs concerns finitely nested I-POMDPs that assume that the other agent is rational up to a certain strategy level (i.e., it is a form of level-k model or cognitive hierarchy [35]), and thus is complementary to our approach.

Finally, as explained in Section 3.5, I-POMDPs implement a different type of learning: they take the perspective that the dynamics of the environment (i.e., the MAE) is known, as well as the set of fixed candidate models $\mathcal{M}_j$. Bayesian updating is the mechanism by which the true model m_j will be inferred. In contrast, we consider the setting where the MAE is not known. Additionally, our formulation decomposes the uncertainty over the behavior of the other agent into uncertainty about the behavior at its internal states; that is rather than having an explicit set of known candidate models, we consider that there is one true model that we try to *estimate* by estimating the behavior at internal states. We point out that while the latter difference can be interpreted as merely a shift in perspective, it is this shift that enables the application of Bayesian reinforcement learning techniques that integrate learning about the environment as well as the other agents.

[4]Inclusion of z_j was convenient in our implementation, but $\langle n_j, a_j \rangle$ also suffices.

8. CONCLUSIONS

In this paper, we presented a first framework for Bayesian Reinforcement learning of best-responses in multiagent settings under state uncertainty. While approaches exist for acting optimally with respect to a belief over models of other agents [14], these cannot deal with uncertainty in the environment. In contrast, by building on BA-POMDPs [24], our BA-BRM framework can deal with both these types of uncertainty. Moreover, we show that by reasoning over the *aggregate influence* of environment and other agents, we can avoid explicit reasoning over infinitely many policies of other agents. To perform learning in these best-response models we developed RS-BA-POMCP, a scalable sample-based planning method based on Monte Carlo tree search, and prove its convergence. Results show significant learning can take place after a small number of episodes assuming different forms of other agent policies.

These approaches can serve as a framework for many future research directions in multiagent learning. Our proof-of-concept experiments clearly indicate the potential to learn, but we expect that more efficient methods that exploit all of our insights could greatly improve performance. Additionally, analytical results regarding the expected/worst-case/best-case time to convergence could be useful to better understand differences between problems such as the two used in our evaluation. A promising direction is to investigate settings that additionally have uncertainty regarding $\mathcal{I}_j$, the set of internal states. We expect that this can be tackled with tools from non-parametric Bayesian inference. In particular, we expect that it may be possible to map the BRM to an infinite POMDP [11]. Such an extension would be important for many applications, since it may be difficult to specify the number of internal states in practise.

Acknowledgements

Research supported in part by AFOSR MURI project #FA9550-091-0538 and in part by NWO Innovational Research Incentives Scheme Veni #639.021.336.

APPENDIX

At the start of every simulation RS-BA-POMCP root samples a D_{root}. Let us write $\tilde{P}^{\pi}(H_d)$ for the probability that RS-BA-POMCP generates H_d at depth d. Clearly $\tilde{P}^{\pi}_{K_d}(H_d) \xrightarrow{p} \tilde{P}^{\pi}(H_d)$. Moreover, it is possible to show that $\tilde{P}^{\pi}(H_d)$

$$= \int \tilde{P}^{\pi}(H_d|D_{root})\, Dir(D_{root}|\chi_{root}) dD_{root}$$

$$= b_0(s_0) \left[\prod_{t=1}^{d} \pi(a_{t-1}|h_{t-0}) \right] \left[\prod_{\langle s,a \rangle} \frac{B(\chi_{sa}(H_0))}{B(\chi_{sa}(H_d))} \right], \qquad (.1)$$

with $B(\alpha) = \frac{\Gamma(\alpha_1 + \ldots + \alpha_k)}{\Gamma(\alpha_1) \cdot \ldots \cdot \Gamma(\alpha_k)}$ the normalization term of a Dirichlet distribution with parametric vector α. Note that a history

$H_{d+1} = (H_d,a_d,s_{d+1},z_{d+1})$ only differs from H_d in the counts for $s_d a_d$. Therefore (.1) can be written in recursive form as $\tilde{P}^\pi(H_{d+1}) = \tilde{P}^\pi(H_d)\pi(a_d|h_d)\frac{B(\chi_{s_d a_d}(H_d))}{B(\chi_{s_d a_d}(H_{d+1}))}$ where the last term is the result of dividing out the contribution of the old counts for $s_d a_d$ and multiplying in the new contribution.

Note that $H_{d+1} = (H_d,a_d,s_{d+1},z_{d+1})$, as far as transitions are concerned, *only* differs from H_d in that it has one extra transition for the $(s_d,a_d,s_{d+1},z_{d+1})$ quadruple. Let us write $T = \sum_{sz} \chi^{sz}_{s_d a_d}(H_d)$ and $N = \chi^{s_{d+1} z_{d+1}}_{s_d a_d}(H_d)$. We also know that $\sum_{sz} \chi^{sz}_{s_d a_d}(H_{d+1}) = T+1$ and $\chi^{s_{d+1} z_{d+1}}_{s_d a_d}(H_{d+1}) = N+1$. Therefore we can write

$$\frac{B(\chi_{s_d a_d}(H_d))}{B(\chi_{s_d a_d}(H_{d+1}))} = \frac{\Gamma(T)/\prod_{s'z}\Gamma(\chi^{s'z}_{s_d a_d}(H_d))}{\Gamma(T+1)/\prod_{s'z}\Gamma(\chi^{s'z}_{s_d a_d}(H_{d+1}))}$$
$$= \cdots = \frac{\Gamma(T)}{\Gamma(T+1)}\frac{\Gamma(N+1)}{\Gamma(N)}$$

Since the gamma function has the property that $\Gamma(x+1) = x\Gamma(x)$ this equals $\frac{\Gamma(T)}{T\Gamma(T)}\frac{N\Gamma(N)}{\Gamma(N)} = \frac{N}{T}$. Therefore we get

$$\tilde{P}^\pi(H_{d+1}) = \tilde{P}^\pi(H_d)\pi(a_d|h_d)\frac{\chi^{s_{d+1} z_{d+1}}_{s_d a_d}(H_d)}{\sum_{sz}\chi^{sz}_{s_d a_d}(H_d)}. \qquad (.2)$$

the r.h.s. of this equation is identical to (5.1) except for the difference in between $\tilde{P}^\pi(H_d)$ and $P^\pi(H_d)$. This can be resolved by forward induction with base step: $\tilde{P}^\pi(H_0) = b_0(\langle s_0,\chi_0,\psi_0\rangle) = P^\pi(H_0)$, and the induction step directly following from (5.1) and (.2). Therefore we conclude that $\tilde{P}^\pi(H_{d+1}) = P^\pi(H_{d+1})$, and that $\forall_{H_{d+1}}\ \tilde{P}^\pi_{K_{d+1}}(H_{d+1}) \xrightarrow{p} P^\pi(H_d)$, thus proving the result. □

A. REFERENCES

[1] S. Abdallah and V. Lesser. Multiagent reinforcement learning and self-organization in a network of agents. In *AAMAS*, pages 172–179, 2007.

[2] C. Amato, J. S. Dibangoye, and S. Zilberstein. Incremental policy generation for finite-horizon DEC-POMDPs. In *ICAPS*, pages 2–9, 2009.

[3] C. Amato and F. A. Oliehoek. Bayesian reinforcement learning for multiagent systems with state uncertainty. In *MSDM*, pages 76–83, 2013.

[4] C. Amato, F. A. Oliehoek, and E. Shyu. Scalable bayesian reinforcement learning for multiagent POMDPs. In *Proc. of the First Multidisciplinary Conference on Reinforcement Learning and Decision Making (RLDM2013)*, 2013.

[5] R. Becker, V. Lesser, and S. Zilberstein. Decentralized Markov Decision Processes with Event-Driven Interactions. In *AAMAS*, pages 302–309, 2004.

[6] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman. Solving transition-independent decentralized Markov decision processes. *JAIR*, 22:423–455, 2004.

[7] D. S. Bernstein, C. Amato, E. A. Hansen, and S. Zilberstein. Policy iteration for decentralized control of Markov decision processes. *JAIR*, 34:89–132, 2009.

[8] D. S. Bernstein, E. A. Hansen, and S. Zilberstein. Bounded policy iteration for decentralized POMDPs. In *IJCAI*, pages 1287–1292, 2005.

[9] G. Chalkiadakis and C. Boutilier. Coordination in multiagent reinforcement learning: A Bayesian approach. In *AAMAS*, pages 709–716, 2003.

[10] Y.-H. Chang, T. Ho, and L. P. Kaelbling. All learning is local: Multi-agent learning in global reward games. In *NIPS 16*, 2004.

[11] F. Doshi-Velez. The infinite partially observable Markov decision process. In *NIPS 22*, pages 477–485, 2009.

[12] M. Duff. *Optimal learning: Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massassachusetts Amherst, 2002.

[13] Y. Engel, S. Mannor, and R. Meir. Reinforcement learning with gaussian processes. In *ICML*, pages 201–208, 2005.

[14] P. J. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multi-agent settings. *JAIR*, 24:24–49, 2005.

[15] A. Guez, D. Silver, and P. Dayan. Efficient Bayes-adaptive reinforcement learning using sample-based search. In *Advances in Neural Information Processing Systems*, 25, pages 1034–1042. 2012.

[16] E. A. Hansen, D. S. Bernstein, and S. Zilberstein. Dynamic programming for partially observable stochastic games. In *AAAI*, pages 709–715, 2004.

[17] T. N. Hoang and K. H. Low. Interactive POMDP lite: Towards practical planning to predict and exploit intentions for interacting with self-interested agents. In *IJCAI*, pages 2298–2305, 2013.

[18] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:1–45, 1998.

[19] R. Nair, D. Pynadath, M. Yokoo, M. Tambe, and S. Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *IJCAI*, pages 705–711, 2003.

[20] F. A. Oliehoek. Decentralized POMDPs. In M. Wiering and M. van Otterlo, editors, *Reinforcement Learning: State of the Art*, volume 12, pages 471–503. Springer, 2012.

[21] L. Peshkin, K.-E. Kim, N. Meuleau, and L. P. Kaelbling. Learning to cooperate via policy search. In *UAI*, pages 489–496, 2000.

[22] P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete bayesian reinforcement learning. In *ICML*, pages 697–704, 2006.

[23] S. Ross, B. Chaib-draa, and J. Pineau. Bayes-adaptive POMDPs. In *NIPS 19*, 2007.

[24] S. Ross, J. Pineau, B. Chaib-draa, and P. Kreitmann. A Bayesian approach for learning and planning in partially observable Markov decision processes. *JMLR*, 12:1729–1770, 2011.

[25] T. W. Sandholm and R. H. Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, 37:147–166, 1995.

[26] S. Seuken and S. Zilberstein. Formal models and algorithms for decentralized control of multiple agents. *JAAMAS*, 17(2):190–250, 2008.

[27] G. Shani, J. Pineau, and R. Kaplow. A survey of point-based POMDP solvers. *Autonomous Agents and Multi-Agent Systems*, pages 1–51, 2012.

[28] D. Silver and J. Veness. Monte-carlo planning in large POMDPs. In *NIPS 23*, pages 2164–2172, 2010.

[29] M. T. J. Spaan and N. Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *JAIR*, 24:195–220, 2005.

[30] P. Stone, G. A. Kaminka, S. Kraus, J. R. Rosenschein, and N. Agmon. Teaching and leading an ad hoc teammate: Collaboration without pre-coordination. *Artificial Intelligence*, 203:35–65, 2013.

[31] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[32] W. T. L. Teacy, G. Chalkiadakis, A. Farinelli, A. Rogers, N. R. Jennings, S. McClean, and G. Parr. Decentralized Bayesian reinforcement learning for online agent collaboration. In *AAMAS*, pages 417–424, 2012.

[33] G. Tesauro and J. O. Kephart. Pricing in agent economies using multi-agent Q-learning. *JAAMAS*, 5(3):289–304, 2002.

[34] N. Vlassis, M. Ghavamzadeh, S. Mannor, and P. Poupart. Bayesian reinforcement learning. In M. Wiering and M. van Otterlo, editors, *Reinforcement Learning: State of the Art*, volume 12. Springer, 2012.

[35] J. R. Wright and K. Leyton-Brown. Beyond equilibrium: Predicting human behavior in normal-form games. In *AAAI*, pages 901–907, 2010.