Approximate Solutions of Interactive POMDPs Using Point Based Value Iteration

Dennis D. Perez
AI Center
University of Georgia
Athens, GA 30602
dperez@uga.edu

Prashant Doshi
Dept. of Computer Science and AI Center
University of Georgia
Athens, GA 30602
pdoshi@cs.uga.edu

Abstract

We develop a point based method for solving finitely nested interactive POMDPs approximately. Analogously to point based value iteration (PBVI) in POMDPs, we maintain a set of belief points and form value functions composed of only those value vectors that are optimal at these points. However, as we focus on multiagent settings, the beliefs are nested and the computation of the value vectors relies on predicted actions of others. Consequently, we develop an *interactive* generalization of PBVI applicable to multiagent settings. We bound the error theoretically and provide empirical results using multiple domains.

1 Introduction

Interactive partially observable Markov decision processes (I-POMDPs; (Gmytrasiewicz & Doshi 2005)) are a framework for sequential decision-making in uncertain, multiagent environments. I-POMDPs facilitate planning and problem-solving in multiagent settings at an agent's own individual level, and in the absence of any centralized controllers (c.f. (Nair *et al.* 2003)) and knowledge about the beliefs of other agents (c.f. (Hansen, Bernstein, & Zilberstein 2004; Nair *et al.* 2003)). Analogous to POMDPs (Kaelbling, Littman, & Cassandra 1998), solutions of I-POMDPs are disproportionately affected not only by growing dimensionalities of the state space (curse of dimensionality), but also by a large policy space that grows exponentially with the number of actions and observations (curse of history).

Because I-POMDPs include models of other agents in the state space as well, the curses of dimensionality and history are particularly potent. First, if models of others encompass their beliefs (sometimes called *intentional models*), the state space is nested representing the beliefs over others' beliefs and their beliefs over others. Second, as the agents act and observe, their beliefs evolve over time. Thus, solutions of I-POMDPs are affected by not only the curse of history afflicting the modeling agent but also that exhibited by the modeled agents.

Previous techniques for approximating solutions of finitely nested I-POMDPs have focused on mitigating the impact of the curse of dimensionality. One approach is to form a sampled representation of the agent's prior nested belief. The samples are then propagated recursively over time using a process called the interactive particle filter (Doshi & Gmytrasiewicz 2005), that generalizes the particle filter to multiagent settings. Because the approach maintains a fixed set of N samples of the interactive state space, it saves on computations though at the expense of solution quality. However, the approach does not address the curse of history and is suited to solving I-POMDPs *online*, when an agent's prior belief is known.

In this paper, we focus on offline solutions of finitely nested I-POMDPs that do not assume a particular initial belief of the agent. In the context of POMDPs, point based solution techniques (for e.g. (Pineau, Gordon, & Thrun 2006; Spaan & Vlassis 2005)) provide effective offline approximations that reduce the impact of the curse of history and subsequently scale well to relatively large problems. This has motivated their use in approximating multiagent decision making. Szer and Charpillet (Szer & Charpillet 2006) improving on (Hansen, Bernstein, & Zilberstein 2004) develop a point based dynamic programming technique to approximate DEC-POMDPs. Seuken and Zilberstein (Seuken & Zilberstein 2007) adopt a memory bounded and point based technique to compute approximately optimal joint policy trees for DEC-POMDPs. While its application in DEC - POMDPs is somewhat straightforward because of the assumption of common knowledge of initial beliefs of the agents and a focus on team settings, we confront multiple challenges in doing so: (i) As point based techniques utilize a set of initial belief points, we need computational representations of the nested beliefs in order to select the initial belief points. (ii) Because there could be infinitely many computable models of other agents, the state space is prohibitively large. Finally, (iii) actions of an agent in a multiagent setting depend on others' actions as well. Therefore, solutions of others' models are required which suggests a recursive implementation of the point based technique.

We provide ways to address these challenges. We show that computational representations of multiply nested beliefs are non-trivial and restrictive assumptions are necessary to facilitate their representations. In this context, we limit the interactive state space by including a finite set of initial models of other agents and those models that are reachable from the initial set over time. Here, we make the assumption that the initial beliefs of the agent are *absolutely continuous* with the true models of all agents as defined in (Doshi & Gmytrasiewicz 2006; Kalai & Lehrer 1993). Finally, we present a

Copyright © 2007, authors listed above. All rights reserved.

generalized *point based value iteration* (PBVI; (Pineau, Gordon, & Thrun 2006)) for finitely nested I-POMDPs that recurses down the nesting, approximately solving the models at each level. We theoretically bound the error and provide evaluations of the performance of the approach on multiple problem domains.

2 Finitely Nested I-POMDPs

Interactive POMDPs generalize POMDPs to multiagent settings by including other agents' models as part of the state space (Gmytrasiewicz & Doshi 2005). Since other agents may also reason about others, the interactive state space is strategically nested; it contains beliefs about other agents' models and their beliefs about others. For simplicity of presentation we consider an agent, i, that is interacting with one other agent, j. A finitely nested I-POMDP of agent i with a strategy level l is defined as the tuple:

$$\text{I-POMDP}_{i,l} = \langle IS_{i,l}, A, T_i, \Omega_i, O_i, R_i \rangle$$

where: • $IS_{i,l}$ denotes a set of interactive states defined as, $IS_{i,l} = S \times M_{j,l-1}$, where $M_{j,l-1} = \{\Theta_{j,l-1} \cup SM_j\}$, for $l \geq 1$, and $IS_{i,0} = S$, where S is the set of states of the physical environment. $\Theta_{j,l-1}$ is the set of computable *intentional models* of agent j: $\theta_{j,l-1} = \langle b_{j,l-1}, \hat{\theta}_j \rangle$ where the *frame*, $\hat{\theta}_j = \langle A,\ \Omega_j,\ T_j,\ O_j,\ R_j,\ OC_j \rangle$. Here, j is Bayes rational and OC_j is j's optimality criterion. SM_j is the set of subintentional models of j. Simple examples of subintentional models include a no-information model and a fictitious play model (Fudenberg & Levine 1998), both of which are history independent. In this paper, we focus on intentional models only. We give a recursive bottom-up construction of the interactive state space:

$$\begin{array}{ll}
IS_{i,0} = S, & \Theta_{j,0} = \{\langle b_{j,0}, \hat{\theta}_j \rangle \mid b_{j,0} \in \Delta(IS_{j,0})\} \\
IS_{i,1} = S \times \Theta_{j,0}, & \Theta_{j,1} = \{\langle b_{j,1}, \hat{\theta}_j \rangle \mid b_{j,1} \in \Delta(IS_{j,1})\} \\
\vdots & \vdots \\
IS_{i,l} = S \times \Theta_{j,l-1}, & \Theta_{j,l} = \{\langle b_{j,l}, \hat{\theta}_j \rangle \mid b_{j,l} \in \Delta(IS_{j,l})\}
\end{array}$$

Similar formulations of nested spaces have appeared in (Aumann 1999; Brandenburger & Dekel 1993).

• $A = A_i \times A_j$ is the set of joint actions of all agents in the environment; • $T_i : S \times A \times S \rightarrow [0,1]$, describes the effect of the joint actions on the physical states of the environment; • Ω_i is the set of observations of agent i; • $O_i : S \times A \times \Omega_i \rightarrow [0,1]$ gives the likelihood of the observations given the physical state and joint action; • $R_i : IS_i \times A \rightarrow \mathbb{R}$ describes agent i's preferences over its interactive states. Usually only the physical states will matter.

Agent i's policy is the mapping, $\Omega_i^* \rightarrow \Delta(A_i)$, where Ω_i^* is the set of all observation histories of agent i. Since belief over the interactive states forms a sufficient statistic (Gmytrasiewicz & Doshi 2005), the policy can also be represented as a mapping from the set of all beliefs of agent i to a distribution over its actions, $\Delta(IS_i) \rightarrow \Delta(A_i)$.

2.1 Belief Update

Analogous to POMDPs, an agent within the I-POMDP framework updates its belief as it acts and observes. First, since the state of the physical environment depends on the actions of both agents, i's prediction of how the physical state changes has to be made based on its prediction of j's actions. Second, changes in j's models have to be included in i's belief update. Specifically, as j is intentional an update of j's beliefs due to its action and observation has to be included. In other words, i has to update its belief based on its prediction of what j would observe and how j would update its belief.

$$\begin{array}{l}
Pr(is^t | a_i^{t-1}, b_{i,l}^{t-1}) = \beta \sum_{IS^{t-1}: \widehat{m}_j^{t-1} = \widehat{\theta}_j^t} b_{i,l}^{t-1}(is^{t-1}) \\
\times \sum_{a_j^{t-1}} Pr(a_j^{t-1} | \theta_{j,l-1}^{t-1}) O_i(s^t, a_i^{t-1}, a_j^{t-1}, o_i^t) \\
\times T_i(s^{t-1}, a_i^{t-1}, a_j^{t-1}, s^t) \sum_{o_j^t} O_j(s^t, a_i^{t-1}, a_j^{t-1}, o_j^t) \\
\times \delta_D(SE_{\widehat{\theta}_j^t}(b_{j,l-1}^{t-1}, a_j^{t-1}, o_j^t) - b_{j,l-1}^t)
\end{array} \tag{1}$$

where β is the normalizing constant, δ_D is 1 if its argument is 0 otherwise it is 0, $Pr(a_j^{t-1} | \theta_{j,l-1}^{t-1})$ is the probability that a_j^{t-1} is Bayes rational for the agent described by model $\theta_{j,l-1}^{t-1}$, and $SE(\cdot)$ is an abbreviation for the belief update.

If agent j is modeled as an I-POMDP, then i's belief update invokes j's belief update (via the term $SE_{\widehat{\theta}_j^t}(b_{j,l-1}^{t-1},\ a_j^{t-1},\ o_j^t)$), which in turn invokes i's belief update and so on. This recursion in belief nesting bottoms out at the 0^{th} level. At this level, the belief update of the agent reduces to a POMDP belief update. [1] For illustrations of the belief update, additional details on I-POMDPs and how they compare with other multiagent frameworks, see (Gmytrasiewicz & Doshi 2005).

2.2 Value Iteration

Each belief state in a finitely nested I-POMDP has an associated value reflecting the maximum payoff the agent can expect in this belief state:

$$\begin{array}{l}
U^t(\langle b_{i,l}, \widehat{\theta}_i \rangle) = \underset{a_i \in A_i}{max} \Bigg\{ \sum\limits_{is \in IS_{i,l}} ER_i(is, a_i) b_{i,l}(is) + \\
\gamma \sum\limits_{o_i \in \Omega_i} Pr(o_i | a_i, b_{i,l}) U^{t+1}(\langle SE_{\widehat{\theta}_i}(b_{i,l}, a_i, o_i), \widehat{\theta}_i \rangle) \Bigg\}
\end{array} \tag{2}$$

where, $ER_i(is, a_i) = \sum_{a_j} R_i(is, a_i, a_j) Pr(a_j | \theta_{j,l-1})$. Eq. 2 is a basis for value iteration in I-POMDPs.

Agent i's optimal action, a_i^*, for the case of finite horizon with discounting, is an element of the set of optimal actions for the belief state, $OPT(\theta_{i,l})$, defined as:

$$\begin{array}{l}
OPT(\langle b_{i,l}, \widehat{\theta}_i \rangle) = \underset{a_i \in A_i}{argmax} \Bigg\{ \sum\limits_{is \in IS_{i,l}} ER_i(is, a_i) b_{i,l}(is) \\
+ \gamma \sum\limits_{o_i \in \Omega_i} Pr(o_i | a_i, b_{i,l}) U^{t+1}(\langle SE_{\widehat{\theta}_i}(b_{i,l}, a_i, o_i), \widehat{\theta}_i \rangle) \Bigg\}
\end{array}$$

2.3 Exact Solution

Notice that the value function, U^t, maps $\Theta_{i,l} \rightarrow \mathbb{R}$. Because $\Theta_{i,l}$ is a continuous space (countably infinite if we limit to computable beliefs), we cannot iterate over all the models of i to compute their values. Instead, analogous to POMDPs, we may decompose the value function into its components:

$$U^t(\langle b_{i,l}, \widehat{\theta}_i \rangle) = \sum_{is \in IS_{i,l}} \alpha^t(is) \times b_{i,l}(is) \tag{3}$$

[1] The 0^{th} level model is a POMDP: Other agent's actions are treated as exogenous events and folded into the T, O, and R functions.

where,

$$\alpha^t(is) = \max_{a_i \in A_i} \Bigg\{ ER_i(is, a_i) + \gamma \sum_{o_i} \sum_{is' \in IS_{i,l}} \Bigg\{ \sum_{a_j} Pr(a_j|\theta_{j,l-1}) \Bigg[T_i(s, a_i, a_j, s') O_i(s', a_i, a_j, o_i) \textstyle\sum_{o_j} O_j(s', a_i, a_j, o_j) \delta_D(SE_{\widehat{\theta}_j}(b_{j,l-1}, a_j, o_j) - b'_{j,l-1}) \Bigg] \Bigg\} \alpha^{t+1}(is') \Bigg\}$$

The proof for Eq. 3 is given in the Appendix of (Gmytrasiewicz & Doshi 2005). For small problems and finite sets of models, we may compute α^t (called the *alpha vector*) exactly, using methods that are analogous to those of POMDPs (Cassandra, Littman, & Zhang 1997; Monahan 1982). We outline a simple way below:

Let $\mathcal{V}^{t+1}$ be the set of time $t+1$ alpha vectors, then $\forall a_i \in A_i$, and $o_i \in \Omega_i$:

$$\Gamma^{a_i,*} \leftarrow \alpha^{a_i,*}(is) = \sum_{a_j \in A_j} R(s, a_i, a_j) Pr(a_j|\theta_{j,l-1}) \quad (4)$$

$$\Gamma^{a_i,o_i} \overset{\cup}{\leftarrow} \alpha^{a_i,o_i}(is) = \gamma \sum_{is'} \sum_{a_j} Pr(a_j|\theta_{j,l-1}) T_i(s, a_i, a_j, s') O_i(s', a_i, a_j, o_i) \sum_{o_j} O_j(s', a_i, a_j, o_j) \delta_D(SE_{\widehat{\theta}_j}(b_{j,l-1}, a_j, o_j) - b'_{j,l-1})\, \alpha^{t+1}(is') \qquad \forall \alpha^{t+1} \in \mathcal{V}^{t+1} \quad (5)$$

Thus we generate $\mathcal{O}(|A_i||\Omega_i|)$ sets of $|\mathcal{V}^{t+1}|$ vectors each. Each vector is of length $|IS_{i,l}|$. Next, we formulate Γ^{a_i} by taking the cross-sum of the previously computed sets of alpha vectors:

$$\Gamma^{a_i} \leftarrow \Gamma^{a_i,*} \oplus \Gamma^{a_i,o_i^1} \oplus \Gamma^{a_i,o_i^2} \oplus \cdots \oplus \Gamma^{a_i,o_i^{|\Omega_i|}} \quad (6)$$

We may generate $\mathcal{O}(|A_i||\mathcal{V}^{t+1}|^{|\Omega_i|})$ many distinct intermediate alpha vectors, and we utilize a linear program (LP) to pick those that are optimal for at least one belief point.

$$\mathcal{V}^t = \underset{\alpha^t}{\textsf{prune}} \ (\bigcup_{a_i} \Gamma^{a_i})$$

Notice that Eqs. 4 and 5 require $Pr(a_j|\theta_{j,l-1})$, which involves solving (exactly) the level $l-1$ intentional models of agent j. Thus, we carry out the above mentioned procedure recursively for solving models at all levels.

3 Computational Representation of Nested Beliefs

While Section 2 presents a mathematical definition of nested belief structures, their computational representations are also needed to facilitate implementations utilizing nested beliefs. However, as we show next, developing these representations is not a trivial task.

3.1 Complexity of Representation

To promote understanding, we assume that j's frame is known and agent i is uncertain about the physical states and j's beliefs only. We explore the representations bottom-up.

Agent i's level 0 belief, $b_{i,0} \in \Delta(S)$, is a vector of probabilities over each physical state: $b_{i,0} \overset{def}{=} \langle\ p_{i,0}(s_1), p_{i,0}(s_2), \ldots, p_{i,0}(s_{|S|})\ \rangle$. Since belief is a probability distribution, $\sum_{q=1}^{|S|} p_{i,0}(s_q) = 1$. We refer to this constraint as the simplex constraint. As we may write, $p_{i,0}(s_{|S|}) = 1 - \sum_{q=1}^{|S|-1} p_{i,0}(s_q)$, subsequently, only $|S|-1$ probabilities are needed to specify a level 0 belief.

Agent i's level 1 belief, $b_{i,1} \in \Delta(S \times \Theta_{j,0})$, may be rewritten as, $b_{i,1}(s, \theta_{j,0}) = p_{i,1}(s) p_{i,1}(\theta_{j,0}|s)$. Therefore, i's level 1 belief is a vector: $b_{i,1} \overset{def}{=} \langle (p_{i,1}(s_1), p_{i,1}(\Theta_{j,0}|s_1)), (p_{i,1}(s_2), p_{i,1}(\Theta_{j,0}|s_2)), \ldots, (p_{i,1}(s_{|S|}), p_{i,1}(\Theta_{j,0}|s_{|S|})) \rangle$. Here, the discrete distribution, $\langle\ p_{i,1}(s_1),\ p_{i,1}(s_2), \ldots, p_{i,1}(s_{|S|})\ \rangle$ satisfies the simplex constraint, and each $p_{i,1}(\Theta_{j,0}|s_q)$ is a single density function over j's level 0 beliefs. [2] We note that $p_{i,1}(\Theta_{j,0}|s_q)$ integrates to 1 over all level 0 models of j.

An example representation of a level 1 belief is to model each density, $p_{i,1}(\Theta_{j,0}|s_q)$, as a weighted mixture of K^q Gaussians. If the frame is known, $p_{i,1}(b_{j,0}|s_q) = \sum_{k=1}^{K^q} w^k\ \mathcal{N}\ (\mu_{i,1}^{k,q}, \Sigma_{i,1}^{k,q})\ (b_{j,0})$, where $b_{j,0}$ is a vector of probabilities as defined previously, $\mu_{i,1}^{k,q}$ and $\Sigma_{i,1}^{k,q}$ are the mean and covariance respectively, of the k^{th} Gaussian component of the mixture. The representation is general because for a sufficiently large K^q, Gaussian mixtures approximate any density to an arbitrary accuracy (McLachlan & Basford 1988).

Agent i's second level belief, $b_{i,2} \in \Delta(S \times \Theta_{j,1})$, analogous to level 1 beliefs, is a vector: $b_{i,2} \overset{def}{=} \langle\ (p_{i,2}(s_1), p_{i,2}(\Theta_{j,1}|s_1)),\ (p_{i,2}(s_2), p_{i,2}(\Theta_{j,1}|s_2)), \ldots, (p_{i,2}(s_{|S|}), p_{i,2}(\Theta_{j,1}|s_{|S|}))\ \rangle$. In comparison to level 0 and level 1 beliefs, representing doubly-nested beliefs and beliefs with deeper nestings is difficult. This is because these are distributions over density functions whose representations need not be finite. For example, let j's singly-nested belief densities be represented using a mixture of Gaussians as shown before. Then, i's doubly nested belief over j's densities is in part a vector of normalized mathematical functions of variables where the variables are the parameters of lower-level densities. Because the lower-level densities are Gaussian mixtures which could have *any* number of components and therefore an arbitrary number of means and covariances, the functions that represent doubly nested beliefs may have an infinite number of variables. Thus computational representations of i's level 2 beliefs are not trivial. We formalize this observation using Proposition 1, which states that multiply-nested beliefs are necessarily partial functions that fail to assign a probability to some elements (lower level beliefs) in their domains.

Proposition 1. *Agent i's multiply nested belief, $b_{i,l}$, $l \geq 2$, is strictly a partial (Turing-)recursive function.*

Proof. We briefly revisit the definition of nested beliefs: $b_{i,l} \in \Delta(IS_{i,l}) = \Delta(S \times \Theta_{j,l-1})$. For simplicity, we assume that j's frame is known (the proof is not contingent on this assumption). As the *basis* case, let $l = 2$, then $b_{i,2} \in \Delta(S \times \langle B_{j,1} \rangle)$. Because the physical state space is discrete, $b_{i,2}$ may be represented, in part using a collection of density functions on j's beliefs, one for each discrete state, $p_{i,2}(b_{j,1}|s_q)$, where $b_{j,1} \in B_{j,1}$ – the set of j's level 1 beliefs. Notice that, $b_{j,1}$

[2] If j's frame is not known, $p_{i,1}(\Theta_{j,0}|s_q)$ is a collection of densities over j's level 0 beliefs, one for each frame.

being singly-nested, is itself, in part, a collection of densities over j's level 0 beliefs, one for each state.

Recall from Section 2 that the models and therefore the belief density functions are assumed computable. Let x be the program of length in bits, $len(x)$, that encodes, say $p_{j,1}(b_{j,0}|s_1)$, in the language g. Then define the *complexity* of the density function, $p_{j,1}(b_{j,0}|s_1)$, as: $C_g(p_{j,1}) = \min\ \{len(x) : g(x) = p_{j,1}(\cdot|s_1)\}$. $C_g(\cdot)$ is the minimum length program in language g that computes the argument.[3] We observe that $len(x)$ is proportional to the number of parameters that describe $p_{j,1}(\cdot|s_1)$. Because the number of parameters of a density need not be bounded, $len(x)$ and consequently the complexity of the density may not be finite. Intuitively, this is equivalent to saying that the density could have "any shape".

Assume, by way of contradiction, that the level 2 density function, $p_{i,2}(b_{j,1}|s_1)$ is a total recursive function. Construct a Turing machine, T, that computes it. Because $p_{i,2}$ is total, T will halt on all inputs. Specifically, T will read the set of symbols on its input tape that describe the level 1 density function (the program, x), and once it has finished reading it halts and leaves a number between 0 and 1 on the output tape. This number is the output of the density function encoded by T. Note that T does not execute the input program x, but simply parses it to enable identification. Thus T is not a universal Turing machine. As we mentioned previously, the minimum length program (and hence the complexity) that encodes the level 1 density function may be infinite. Thus the size of the set of symbols on the input tape of T, $len(x)$, may be infinite, and T may not halt. But this is a contradiction. Thus, $p_{i,2}$ is a partial recursive function.

The argument may be extended inductively to further levels of nesting. ∎

As multiply-nested beliefs in their general form are partial recursive functions that are not defined for every possible lower level belief in their domains, restrictions on the complexity of nested beliefs are needed to allow for computability and so that they are well-defined. One sufficient way is to focus our attention on a limited set of other's models.

3.2 Absolute Continuity Condition

Let $\tilde{\Theta}_{j,0}$ be a *finite* set of j's computable level 0 models. Then, define $\widetilde{IS}_{i,1} = S \times \tilde{\Theta}_{j,0}$ and agent i's belief, $\tilde{b}_{i,1} \in \Delta(\widetilde{IS}_{i,1})$. As we mentioned before, i's level 1 belief may be rewritten as: $\tilde{b}_{i,1}(\tilde{is}) = p_{i,1}(s)p_{i,1}(\tilde{\theta}_{j,0}|s)$. Therefore, i's level 1 belief is a vector: $\tilde{b}_{i,1} \overset{def}{=} \langle(p_{i,1}(s_1),p_{i,1}(\tilde{\Theta}_{j,0}|s_1)), (p_{i,1}(s_2),p_{i,1}(\tilde{\Theta}_{j,0}|s_2)),\ldots, (p_{i,1}(s_{|S|}),p_{i,1}(\tilde{\Theta}_{j,0}|s_{|S|}))\ \rangle$. Here, the discrete distribution, $\langle\ p_{i,1}(s_1),\ p_{i,1}(s_2),\ldots,p_{i,1}(s_{|S|})\ \rangle$ satisfies the simplex constraint. Additionally, each $p_{i,1}(\tilde{\Theta}_{j,0}|s_1)$ is also a discrete distribution that satisfies the simplex constraint. We generalize to level l in a straightforward manner: Let $\tilde{\Theta}_{j,l-1}$ be a finite set of j's computable level $l-1$ models. Then, define $\widetilde{IS}_{i,l} = S \times \tilde{\Theta}_{j,l-1}$ and agent i's belief, $\tilde{b}_{i,l} \in \Delta(\widetilde{IS}_{i,l})$. Here, analogous to a level 1 belief, $b_{i,l} \overset{def}{=} \langle\ (p_{i,l}(s_1),\ p_{i,l}(\tilde{\Theta}_{j,l-1}|s_1)),\ (p_{i,l}(s_2),\ p_{i,l}(\tilde{\Theta}_{j,l-1}|s_2)),\ldots,(p_{i,l}(s_{|S|}),\ p_{i,l}(\tilde{\Theta}_{j,l-1}|s_{|S|}))\ \rangle$.

Notice that i's belief over the physical states and other's candidate models, together with its perfect information about its own model induces a predictive probability distribution over the joint future observations in the interaction (Doshi & Gmytrasiewicz 2006). Because we limit the support of i's belief to a finite set of models, the actual sequence of observations may not proceed along a path that is assigned some non-zero predictive probability by i's belief. In this case, i's observations may contradict its belief and a Bayesian belief update may not be possible.

Therefore, it is desirable that agent i's belief, $\tilde{b}_{i,l}$, assign a non-zero probability to each potentially realizable observation path in the interaction – this condition has also been called the truth compatibility condition (Kalai & Lehrer 1993). We formalize this condition mathematically using the notion of *absolute continuity* of two probability measures:

Definition 1 (Absolute Continuity)**.** *A probability measure p_1 is absolutely continuous with p_2, denoted as $p_1 \ll p_2$, if $p_2(E) = 0$ implies $p_1(E) = 0$, for any measurable set E.*

In order to formally define the condition, let ρ_0 be the true distribution over the possible observation paths induced by perfectly knowing the true models of i and j. Let $\rho_{b_{i,l}}$ be the distribution over the observation paths induced by i's initial belief, $\tilde{b}_{i,l}$. Then,

Definition 2 (Absolute Continuity Condition (ACC))**.** *ACC holds for an agent, say i, if $\rho_0 \ll \rho_{b_{i,l}}$.*

Of course, a sufficient but not necessary way to satisfy the ACC is for agent i to include each possible model of j in the support of its belief. However, as Proposition 1 precludes this, we select a finite set of j's candidate models with the partial (domain) knowledge that the true model of j is one of them.

4 Interactive PBVI

Because I-POMDPs include possible models of other agents that are solved, their solution complexity additionally suffers from the curse of history that afflicts the modeled agents. This curse manifests itself in the $|A_j||\mathcal{V}_j^{t+1}|^{|\Omega_j|}$ alpha vectors that are generated at time t (Eq. 5) and the subsequent application of LPs to select the optimal vectors, to solve the models of agent j. Point based approaches (for e.g., see (Pineau, Gordon, & Thrun 2006; Spaan & Vlassis 2005) for POMDPs, and (Szer & Charpillet 2006) for DEC-POMDPs) utilize a finite set of belief points to decide which alpha vectors to retain, and thereby do not utilize the LPs.

4.1 Bounded IS and Initial Beliefs

As we mentioned in Section 3, we limit the space of j's candidate initial models to a finite set, $\tilde{\Theta}_{j,l-1}$. However, because the models of j may grow as it acts and observes, agent i must track these models over time in order to act rationally. Let $\mathsf{Reach}(\tilde{\Theta}_{j,l-1}, H)$ be the set of level $l-1$

[3] Note that the complexity, C_g, is within a constant of the Kolmogorov complexity (Li & Vitanyi 1997) of the density function, $p_{j,1}$.

models that j could have in the course of H steps. Note that $\mathsf{Reach}(\tilde{\Theta}_{j,l-1}, 0) = \tilde{\Theta}_{j,l-1}$. In computing $\mathsf{Reach}(\cdot)$, we repeatedly update j's beliefs in the models contained in $\tilde{\Theta}_{j,l-1}$ using Eq. 1. We define a bounded interactive state space as follows:

$$\begin{array}{ll} \tilde{IS}_{i,0} = S, & \tilde{\Theta}_{j,0} = \{\langle \tilde{b}_{j,0}, \hat{\theta}_j \rangle \mid \tilde{b}_{j,0} \in \Delta(\tilde{IS}_{j,0})\} \\ \tilde{IS}_{i,1} = S \times \mathsf{Reach}(\tilde{\Theta}_{j,0}, H), & \tilde{\Theta}_{j,1} = \{\langle \tilde{b}_{j,1}, \hat{\theta}_j \rangle \mid \tilde{b}_{j,1} \in \Delta(\tilde{IS}_{j,1})\} \\ \vdots & \vdots \\ \tilde{IS}_{i,l} = S \times \mathsf{Reach}(\tilde{\Theta}_{j,l-1}, H), & \tilde{\Theta}_{j,l} = \{\langle \tilde{b}_{j,l}, \hat{\theta}_j \rangle \mid \tilde{b}_{j,l} \in \Delta(\tilde{IS}_{j,l})\} \end{array}$$

For each level of the nesting, we select an initial set of beliefs for the corresponding agent randomly. We show the procedure for performing this selection in Fig. 1.

```
RANDOM-SELECT(Strategy level:$l \geq 0$, $\tilde{IS}_{k,l}$, # beliefs:$N$)

if $l = 0$ then
    for $n$ from 1 to $N$ do
        Select a distribution, $\tilde{b}_{k,0} \in \Delta(S)$, randomly
        $\tilde{B}^N_{k,0} \overset{\cup}{\leftarrow} \tilde{b}_{k,0}$
    end for
else
    Call RANDOM-SELECT ($l-1$, $\tilde{IS}_{-k,l-1}$, $N$)
    for $n$ from 1 to $N$ do
        Select a distribution, $p_{k,l}(S) \in \Delta(S)$, randomly
        for all $s \in S$ do
            Select a distribution, $p_{k,l}(\tilde{\Theta}_{-k,l-1}|s) \in \Delta(\tilde{\Theta}_{-k,l-1})$, randomly
        end for
        for all $s \in S$, $\tilde{\theta}_{-k,l-1} \in \tilde{\Theta}_{-k,l-1}$ do
            $\tilde{b}_{k,l}(s, \tilde{\theta}_{-k,l-1}) \leftarrow p_{k,l}(s) \times p_{k,l}(\tilde{\Theta}_{-k,l-1}|s)$
        end for
        Normalize $\tilde{b}_{k,l}$, $\tilde{B}^N_{k,l} \overset{\cup}{\leftarrow} \tilde{b}_{k,l}$
    end for
end if
Return the belief sets, $\tilde{B}^N_{k,l}, \ldots, \tilde{B}^N_{k,0}$
```

Figure 1: A recursive algorithm for randomly selecting an initial set of N beliefs at all levels of the nesting. Here, k (and $-k$) assumes agent i (and j) or j (and i) as appropriate.

4.2 Point Based Back Projections

Given the bounded interactive state space defined previously, Eqs. 4 and 5 may be rewritten. $\forall a_i \in A_i$ and $o_i \in \Omega_i$:

$$\tilde{\Gamma}^{a_i,*} \leftarrow \alpha^{a_i,*}(\tilde{is}) = \sum_{a_j \in A_j} R(s, a_i, a_j) Pr(a_j|\tilde{\theta}_{j,l-1}) \quad (7)$$

$$\tilde{\Gamma}^{a_i,o_i} \overset{\cup}{\leftarrow} \alpha^{a_i,o_i}(\tilde{is}) = \gamma \sum_{\tilde{is}'} \sum_{a_j} Pr(a_j|\tilde{\theta}_{j,l-1}) T_i(s, a_i, a_j, s') \\ O_i(s', a_i, a_j, o_i) \sum_{o_j} O_j(s', a_i, a_j, o_j) \delta_D(SE_{\hat{\theta}_j}(\tilde{b}_{j,l-1}, a_j, o_j) \\ -\tilde{b}'_{j,l-1})\ \alpha^{t+1}(\tilde{is}') \qquad \forall \alpha^{t+1} \in \mathcal{V}^{t+1} \quad (8)$$

where $\tilde{is}, \tilde{is}' \in \tilde{IS}_{i,l}$ and $\tilde{is} = \langle s, \tilde{\theta}_{j,l-1} \rangle$.

Let $\tilde{B}_{i,l}$ be a finite set of level l belief points at some time t. As we seek alpha vectors of agent i that are optimal at the beliefs in $\tilde{B}_{i,l}$, we may simplify the cross-sum computations shown in Eq. 6. In particular, we need not consider all the vectors in a set, say Γ^{a_i,o_i^1}, but only those that are optimal at some belief point, $\tilde{b}_{i,l} \in \tilde{B}_{i,l}$. Formally,

$$\tilde{\Gamma}^{a_i} \leftarrow \tilde{\Gamma}^{a_i,*} \bigoplus_{o_i \in \Omega_i} \underset{\tilde{\Gamma}^{a_i,o_i}}{argmax}\, (\alpha^{a_i,o_i} \cdot \tilde{b}_{i,l}) \quad \forall \tilde{b}_{i,l} \in \tilde{B}_{i,l} \quad (9)$$

We again utilize $\tilde{B}_{i,l}$ to finally select the alpha vectors that form the set $\mathcal{V}^t$:

$$\mathcal{V}^t \leftarrow \underset{\alpha^t \in \bigcup_{a_i} \Gamma^{a_i}}{argmax}\ (\alpha^t \cdot \tilde{b}_{i,l}) \qquad \forall \tilde{b}_{i,l} \in \tilde{B}_{i,l}$$

Notice that in Eq. 9, we generate at most $\mathcal{O}(|A_i||\mathcal{V}^{t+1}|^{|\Omega_i|})$ alpha vectors, typically less, and do not require a LP to select the optimal ones. The set $\mathcal{V}^t$ contains unique alpha vectors that are optimal for at least one of the belief points in $\tilde{B}_{i,l}$. Hence, $\mathcal{V}^t$ contains at most $|\tilde{B}_{i,l}|$ many alpha vectors, typically less in practice. Because the number of alpha vectors depends on the set of belief points, we may limit the latter to a constant size.

What remains is how we compute the term $Pr(a_j|\tilde{\theta}_{j,l-1})$ in Eqs. 7 and 8. We may solve agent j's I-POMDP of level $l-1$ or POMDP of level 0 in an analogous manner using a finite set of belief points of j. Consequently, we recurse through the levels of nesting, utilizing a pre-computed finite set of belief points at each level to generate the alpha vectors that are optimal at those points.

4.3 Top Down Expansion of Belief Points

We point out three issues that may guide the expansions of finite sets of belief points for the agents. First, rather than being distributed over the entire space, an agent's beliefs often follows certain trajectories. Thus, selecting belief points that lie on the trajectories may result in solutions that offer good performance quality. Second, selecting a belief point that is in close spatial proximity to another may not result in a new alpha vector, thereby making the belief point redundant. Finally, in comparison to single agent settings, generating beliefs in a setting populated by other agents may require predicting their actions as well.

In the context of POMDPs, Spaan and Vlassis (Spaan & Vlassis 2005) utilize a fixed set of beliefs obtained by randomly exploring the environment. During the back projections, they progressively filter out the belief points considering only those for which the previously back projected alpha vectors are not a better policy. In (James, Samples, & Dolgov 2007), James et al. incrementally introduce belief points that have the potential of providing the largest gain, where gain is the difference between the current value of the policy at that point as obtained from previously selected alpha vectors and a minimal upper bound. However, as the authors conclude, finding the minimal upper bound is computationally expensive and for large belief spaces (as is the case in multiagent settings) may offset the runtime savings provided by point based approaches.

We utilize two approaches to expand the sets of belief points over time that are used to select the alpha vectors:

- **Stochastic trajectory simulation** For each belief in a belief set, $\tilde{B}_{i,l}$, we sample a physical state and the other agent's model. We then uniformly sample i's action, a_i, and in combination with the sampled physical state and j's action obtained from solving j's model, we sample the next physical state using the transition function. Given the updated physical state and joint actions, we sample an observation of i, o_i, from the observation function. Agent i's belief is then updated given its action, a_i, and observation, o_i, using the belief update (Eq. 1).

- **Error minimization** The approximation error in point based approaches, in part, depends on the density of the set of belief points. We prefer to generate a new belief point, $b_{i,l}^{t+1}$, such that the optimal alpha vector at that point is furthest in value from the alpha vector at an existing belief that is the closest to the generated belief. This is because in the absence of such a point, a large error would be incurred at that point. As the optimal alpha vector at $b_{i,l}^{t+1}$ is not known, we may utilize the maximum (or minimum) value, $\frac{R_{max}}{1-\gamma}$ for each is, in its place. Consequently, we first select a belief point, $b_{i,l}^{t}$ from the set $\tilde{B}_{i,l}$, which when updated will result in $b_{i,l}^{t+1}$.

Similar approaches were used in (Pineau, Gordon, & Thrun 2006) for expanding beliefs in point based approaches in the context of single agent POMDPs, where they demonstrated good results. [4] For each of the expansion techniques, beliefs at all strategy levels are recursively generated in an analogous manner.

5 Algorithm

We show the main procedure for performing the interactive PBVI (I-PBVI) in Fig. 2. We generate the initial belief points, $\langle \tilde{B}_{k,l}^N$, $\tilde{B}_{-k,l-1}^N$,. . ., $\tilde{B}_{k,0}^N \rangle$, using the RANDOM-SELECT algorithm in Fig. 1, though other ways, for example utilizing prior knowledge about probable beliefs, may be used. If the I-POMDP is not strategically nested, we back project the time $t + 1$ vectors using a standard backup technique for single agent POMDPs, as given in, say (Pineau, Gordon, & Thrun 2006). However, if the I-POMDP is nested, a more sophisticated approach is needed for the backup (lines 2-7). The alpha vectors at time H (horizon 1) are initialized to their lower bounds, $\frac{R_{min}}{1-\gamma}$ (line 1). This is sufficient to ensure that the repeated back projections will gradually improve the value function. Though in Fig. 2, we recursively expand the set of beliefs, $\langle$ $\tilde{B}_{k,l}^N$, $\tilde{B}_{-k,l-1}^N$,. . .,$\tilde{B}_{k,0}^N$ $\rangle$, after each backup, we may reduce our computational overhead by performing the expansions more sparsely. Here, we utilize the techniques in Section 4.3 for carrying out the expansions (lines 8-9).

We show the procedure for back projecting the vectors for the case where the I-POMDP is nested to a level $l > 0$, in Fig. 3. In a nutshell, we utilize the steps outlined in Section 4.2 to identify the projected alpha vectors that are optimal at the belief points in the set, $\tilde{B}_{k,l}$ (lines 2-17). However, in doing so we need to predict the other agent's actions as well which is obtained by solving its models. Therefore, in performing the backup, we descend through the nesting solving the models at each level by recursively performing the I-PBVI (note the recursive invocation of I-PBVI in line 1).

```
I-PBVI (Initial beliefs: $\langle \tilde{B}_{k,l}^N, \tilde{B}_{-k,l-1}^N, \ldots, \tilde{B}_{k,0}^N \rangle$, Horizons: $H > 0$, Strategy level: $l \geq 0$)
 1: $\tilde{\Gamma}^H \leftarrow$ INITIAL-ALPHAVECTORS ()
 2: for $t \leftarrow H - 1$ to $0$ do
 3:     if $l = 0$ then
 4:         $\tilde{\Gamma}^t \leftarrow$ PBVI BACKUP($\tilde{B}_{k,0}^N$, $\tilde{\Gamma}^{t+1}$, $H - t$)
 5:     else
 6:         $\tilde{\Gamma}^t \leftarrow$ I-PBVI BACKUP($\tilde{B}_{k,l}^N, \ldots, \tilde{B}_{k,0}^N$, $\tilde{\Gamma}^{t+1}$, $H -$
            $t$, $l$)
 7:     end if
 8:     Expand the previous set of beliefs at all levels using
        techniques from Section 4.3
 9:     Add the expanded beliefs to the existing sets
10: end for
11: return $\tilde{\Gamma}^0$
```

Figure 2: The interactive PBVI procedure for generating the alpha vectors at horizon, H. Note that when $l = 0$, the vector projection is analogous to that for POMDPs. Here, k (and $-k$) assumes agent i (and j) or j (and i) as appropriate.

```
I-PBVI BACKUP ($\langle \tilde{B}_{k,l}, \ldots, \tilde{B}_{k,0} \rangle$, $\tilde{\Gamma}_k^{t+1}$, $h$, $l$)

 1: $\tilde{\Gamma}_{-k}^t \leftarrow$ I-PBVI ($\langle \tilde{B}_{-k,l-1}, \ldots, \tilde{B}_{k,0} \rangle$, $h$, $l - 1$)
 2: for all $a_k \in A_k$ do
 3:     Compute $\alpha_k^{a_i,*}$ (Eq. 7) where $Pr(a_{-k}|\tilde{\theta}_{-k,l-1}) \leftarrow$
        GETACTION ($\tilde{\theta}_{-k,l-1}$, $\tilde{\Gamma}_{-k}^{t+1}$) and add $\alpha_k^{a_i,*}$ to $\tilde{\Gamma}^{a_i,*}$
 4:     for all $o_k \in \Omega_k$ do
 5:         Compute $\alpha_k^{a_i,o_i}$ (Eq. 8), where $Pr(a_{-k}|\tilde{\theta}_{-k,l-1}) \leftarrow$
            GETACTION ($\tilde{\theta}_{-k,l-1}$, $\tilde{\Gamma}_{-k}^{t+1}$), add $\alpha_k^{a_i,o_i}$ to $\tilde{\Gamma}^{a_i,o_i}$
 6:     end for
 7: end for
 8: for all $\tilde{b}_{k,l} \in \tilde{B}_{k,l}$ do
 9:     Compute $\alpha_k^{a_i}$ (Eq. 9) and add $\alpha_k^{a_i}$ to $\tilde{\Gamma}^{a_i}$
10: end for
11: $\tilde{\Gamma}^t \leftarrow \bigcup_{a_i} \tilde{\Gamma}^{a_i}$
12: for all $\tilde{b}_{k,l} \in \tilde{B}_{k,l}$ do
13:     $\alpha_k^* \leftarrow \underset{\alpha_k \in \tilde{\Gamma}^n}{argmax}\ \alpha_k \cdot \tilde{b}_{i,l}$
14:     if $\alpha_k^* \notin \tilde{\Gamma}_*^t$ then
15:         Add $\alpha_k^*$ to $\tilde{\Gamma}_*^t$
16:     end if
17: end for
18: return $\tilde{\Gamma}_*^t$
```

Figure 3: Procedure for backing up the alpha vectors when strategy level $l > 0$. Note the recursive call to I-PBVI on line 1 for solving the models of the other agent.

6 Computational Savings and Error Bounds

If the strategy level is 0, the I-POMDP$_i$ collapses into a POMDP and we generate in the worst case $\mathcal{O}(|A_i||\mathcal{V}^{t+1}|^{|\Omega_i|})$

[4] For POMDPs, the error minimization showed the best performance (Pineau, Gordon, & Thrun 2006), improving on (Spaan & Vlassis 2005) as well.

many alpha vectors at time t in order to solve the POMDP exactly. Let $M_{j,l-1} = \mathsf{Reach}(\tilde{\Theta}_{j,l-1}, H)$. We first consider solving the I-POMDP of i at level 1. Because we include $|M_{j,0}|$ many models of j in the state space, we need obtain $|M_{j,0}|$ alpha vectors assuming j's frame is known. These are used in solving the I-POMDP of i, which in the worst case generates $\mathcal{O}(|A_i||\mathcal{V}^{t+1}|^{|\Omega_i|})$ vectors. [5] Thus, a total of $\mathcal{O}(|A_i||\mathcal{V}^{t+1}|^{|\Omega_i|}+|M_{j,0}|)$ alpha vectors are obtained at level 1. Generalizing to level l and assuming, for the sake of simplicity, that the same number of models of the other agent are included at any level, $|M|$, we need $\mathcal{O}(|A_i||\mathcal{V}^{t+1}|^{|\Omega_i|}+|M|l)$ alpha vectors to solve the I-POMDP$_{i,l}$ exactly. In the context of the I-PBVI, if at most N belief points are used at any level, the approximate solution of a level 0 I-POMDP generates $\mathcal{O}(N)$ alpha vectors. For level 1, because solutions of $|M|$ models are obtained approximately using N belief points, we need obtain only $\mathcal{O}(N)$ vectors for j and another $\mathcal{O}(N)$ vectors to solve the I-POMDP of i at level 1 approximately. Generalizing to level l, we generate at most $\mathcal{O}(N(l+1))$ many alpha vectors. For the case where $N << |M|$, significant computational savings are obtained. Of course, for more than two agents, the number of alpha vectors are exponential in the number of agents.

The loss in optimality or error due to approximately solving the I-POMDP using I-PBVI is due to two reasons: (i) The alpha vectors that are optimal at selected belief points may be suboptimal at other points; and (ii) Models of the other agent are solved approximately as well. We begin a characterization of the error by noting that Eq. 3 may be rewritten as:

$$\begin{array}{l}\alpha^t(is) = \sum_{a_j \in A_j} Pr(a_j|\theta_{j,l-1}) \times \underset{a_i \in A_i}{max} \Bigg\{ R_i(s, a_i, a_j) + \\ \gamma \sum\limits_{o_i} \sum\limits_{is' \in IS_{i,l}} \Bigg\{ \Bigg[T_i(s, a_i, a_j, s') O_i(s', a_i, a_j, o_i) \sum_{o_j} O_j(s', a_i, \\ a_j, o_j) \delta_D(SE_{\widehat{\theta}_j}(b_{j,l-1}, a_j, o_j) - b'_{j,l-1}) \Bigg] \Bigg\} \alpha^{t+1}(is') \Bigg\} \\ \qquad = Pr(a_j|\theta_{j,l-1}) \cdot \alpha^t_{a_j}\end{array} \tag{10}$$

Let $\tilde{b}'_{i,l}$ be the belief point where the maximum error occurs, and α'' be the exact alpha vector that is optimal at this belief point. Let α be the approximate vector that is instead utilized at $\tilde{b}'_{i,l}$ for computing the policy. Note that in using α the solution suffers from both the sources of error mentioned previously, while α'' induces no error. Let α' be optimal at $\tilde{b}'_{i,l}$ while still exhibiting an error due to the approximate solution of j's models. We may define the worst case error as:

$$\begin{aligned}\mathcal{E} &= \alpha'' \cdot \tilde{b}'_{i,l} - \alpha \cdot \tilde{b}'_{i,l} \\ &= \alpha'' \cdot \tilde{b}'_{i,l} - \alpha \cdot \tilde{b}'_{i,l} + (\alpha' \cdot \tilde{b}'_{i,l} - \alpha' \cdot \tilde{b}'_{i,l}) \\ &= (\alpha'' \cdot \tilde{b}'_{i,l} - \alpha' \cdot \tilde{b}'_{i,l}) + (\alpha' \cdot \tilde{b}'_{i,l} - \alpha \cdot \tilde{b}'_{i,l})\end{aligned} \tag{11}$$

We first focus on the second term, $\alpha' \cdot \tilde{b}'_{i,l} - \alpha \cdot \tilde{b}'_{i,l}$. Here, the error is only due to the limited set of belief points, as both α' and α utilize the same approximate solution of j's models. Define $d_{\tilde{B}}$ as the largest of the distances between the pruned belief, $\tilde{b}'_{i,l}$, and the closest belief, $\tilde{b}_{i,l}$, among the selected points: $d_{\tilde{B}} = max_{\tilde{b}'_{i,l} \in \Delta_{i,l}}\ min_{\tilde{b}_{i,l} \in \tilde{B}_{i,l}}\ |\tilde{b}'_{i,l} - \tilde{b}_{i,l}|$. Note that $d_{\tilde{B}}$ reflects the density of the selected belief points within the belief simplex. The derivation of the error for this case proceeds in a manner analogous to that of PBVI (Pineau, Gordon, & Thrun 2006). Subsequently, we get the following worst-case error bound:

$$\alpha' \cdot \tilde{b}'_{i,l} - \alpha \cdot \tilde{b}'_{i,l} \leq \frac{R_i^{max} - R_i^{min}}{1-\gamma} d_{\tilde{B}}$$

Next, we turn our attention to the first term, $\alpha'' \cdot \tilde{b}'_{i,l} - \alpha' \cdot \tilde{b}'_{i,l}$, of Eq. 11. This term represents the error due to the approximate solution of the other agent's models obtained by using PBVI recursively. We may write it as:

$$\begin{array}{l}\alpha'' \cdot \tilde{b}'_{i,l} - \alpha' \cdot \tilde{b}'_{i,l} = \tilde{b}'_{i,l} \cdot (\alpha'' - \alpha') \\ = \tilde{b}'_{i,l} \cdot (\alpha''_{a_j} \cdot Pr(a_j|\cdot) - \alpha''_{a_j} \cdot Pr'(a_j|\cdot)) \quad \text{(Using Eq. 10)} \\ = \tilde{b}'_{i,l} \cdot (\alpha''_{a_j} \cdot (Pr(a_j|\cdot) - Pr'(a_j|\cdot)))\end{array}$$

The inner dot product is over j's actions. $Pr'(a_j|\cdot)$ represents the suboptimal probability due to the approximation. Consider the case where $Pr'(a_j|\cdot)$ prescribes an action, a'_j, different from that by $Pr(a_j|\cdot)$. Then the worst error is loosely bounded by, $\alpha''_{a_j} - \alpha''_{a'_j} \leq \frac{R_i^{max} - R_j^{min}}{1-\gamma}$. Therefore,

$$\alpha'' \cdot \tilde{b}'_{i,l} - \alpha' \cdot \tilde{b}'_{i,l} \leq \tilde{b}'_{i,l} \cdot \tfrac{R_i^{max} - R_i^{min}}{1-\gamma} = \tfrac{R_i^{max} - R_i^{min}}{1-\gamma}$$

Thus, although the error due to pruning the belief points is bounded and depends on the density of the selected belief points, we are unable to usefully bound the error due to approximately solving other's models.

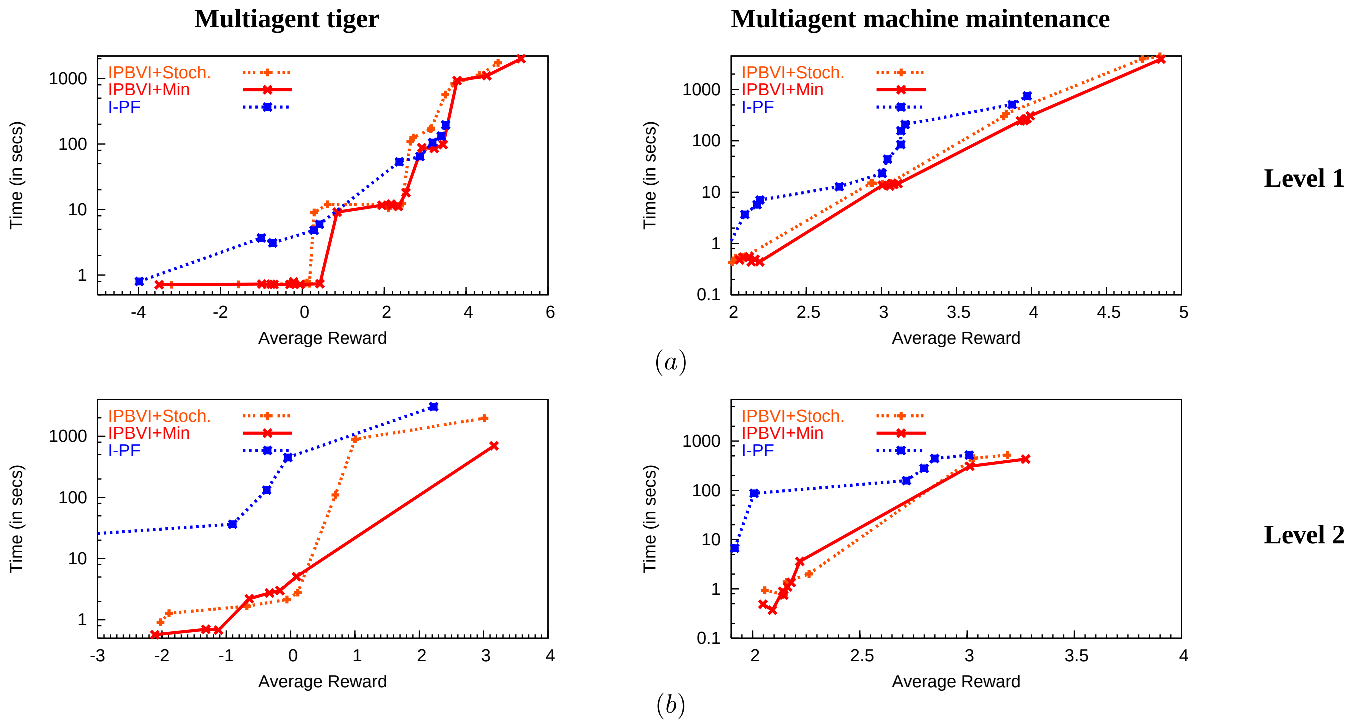
7 Performance Evaluation

We implemented the algorithm in Section 5 and evaluated its performance on the well known multiagent *tiger* problem (Gmytrasiewicz & Doshi 2005) and a multiagent version of the machine maintenance (MM) problem (Smallwood & Sondik 1973).

Although the two problems have a small physical state space (tiger: 2 physical states, MM: 3 physical states), the interactive state space, $IS_{i,l}$, is large because we include the models of the other agent as well (for example, MM: approx. 60 interactive states). For both the problems, we provide the time taken in reaching a particular performance in terms of the rewards gathered by agent i. The time consumed is a function of the number of belief points used during I-PBVI, the horizons of the policy and the number of j's models. We gradually increased the number of belief points, horizons and models and simulated the performance of the resulting policies over 10 trials with 50 runs each. In each trial, we selected a different initial belief of agent i, and sampled the starting state and belief of j from this belief. In solving the I-POMDP of agent i, j's models must be solved as well. We compare the results across both the expansion strategies mentioned in Section 4.3.

We show the level 1 and 2 plots for the two problems in Figs. 4(a) and (b), respectively. Lower values on y-axis indicate better performance. Notice that for level 1 the error

[5] Note that these vectors are of size $|\tilde{IS}_{i,l}|$ compared to size $|S|$ of the vectors for POMDPs.



(a)

(b)

Figure 4: Level 1 (j's models are POMDPs) and level 2 (j's models are level 1 I-POMDPs) plot of time consumed in achieving a desired performance. Note that the y-axis is in log scale. The I-PBVI significantly improves on the I-PF, a previous approximation technique for I-POMDPs. All experiments are run on a Linux platform with dual processor Xeon 3.4GHz with 4GB memory.

minimization expansion approach (denoted as IPBVI+Min) performs better than the approach of using the stochastic trajectory (denoted as IPBVI+Stoch) to expand the belief points, in both domains. Specifically, IPBVI+Min takes less time in providing an identical performance as when the IPBVI+Stoch is used. However, the distinction is less evident at level 2 where the greater computations incurred in using the minimization approach assume significance. These observations are analogous to the mixed performance of the different expansion techniques in POMDPs (Pineau, Gordon, & Thrun 2006). One way to assess the impact of deeper modeling is to measure the average rewards obtained by i across levels for the same number of belief points. Our experiments do not reveal a significant overall improvement when agent i's beliefs are doubly nested, although level 2 solutions are computationally more intensive as evident from Fig. 4. However, there is evidence in the tiger problem that modeling at level 1 results in better performance in comparison to naively treating the other agent as noise (Gmytrasiewicz & Doshi 2005).

Due to an absence of other offline approximation techniques for I-POMDPs, we compare the performance of I-PBVI with the interactive particle filter (I-PF) based approximation (Doshi & Gmytrasiewicz 2005). We generate policy trees for as many initial beliefs of i as the number of belief points used in I-PBVI. Although the I-PF is able to mitigate the curse of dimensionality, it must generate the full reachability tree to compute the policy and therefore it continues to suffer from the curse of history that affects I-POMDPs. The better performance of I-PBVI in comparison to I-PF demonstrates that point based value iteration is able to mitigate the impact of the curse of history that affects the solutions of both the agents' decision processes. Furthermore, we were unable to run the I-PF beyond a few time horizons due to excessive memory consumption.

8 Discussion

We presented a generalization of point based value iteration applicable to interactive settings where agents model others. The approximation technique is *anytime* and exhibits improved performance and scalability in comparison to previous approximations of I-POMDPs. While it mitigates the impact of having to maintain the history of interaction, nevertheless we maintain the set of reachable models of the other agent that could quickly grow over time. Further improvement is possible by carefully limiting the set of candidate models of others that are considered.

Acknowledgments This research is supported, in part, by a grant from UGA Research Foundation. The authors also acknowledge the comments of the anonymous reviewers.

References

Aumann, R. J. 1999. Interactive epistemology i: Knowledge. *International Journal of Game Theory* 28:263–300.

Brandenburger, A., and Dekel, E. 1993. Hierarchies of beliefs and common knowledge. *Journal of Economic Theory* 59:189–198.

Cassandra, A. R.; Littman, M. L.; and Zhang, N. L. 1997. Incremental pruning: A simple, fast, exact method for partially observable markov decision processes. In *Uncertainty in Artificial Intelligence*.

Doshi, P., and Gmytrasiewicz, P. J. 2005. Approximating state estimation in multiagent settings using particle filters. In *Autonomous Agents and Multi-agent Systems Conference (AAMAS)*, 320–327.

Doshi, P., and Gmytrasiewicz, P. J. 2006. On the difficulty of achieving equilibrium in interactive pomdps. In *National Conference on Artificial Intelligence (AAAI)*.

Fudenberg, D., and Levine, D. K. 1998. *The Theory of Learning in Games*. MIT Press.

Gmytrasiewicz, P., and Doshi, P. 2005. A framework for sequential planning in multiagent settings. *Journal of Artificial Intelligence Research (JAIR)* 24:49–79.

Hansen, E.; Bernstein, D.; and Zilberstein, S. 2004. Dynamic programming for partially observable stochastic games. In *National Conference on AI (AAAI)*.

James, M.; Samples, M.; and Dolgov, D. 2007. Improving anytime point based value iteration using principled point selections. In *International Joint Conference on AI (IJCAI)*.

Kaelbling, L.; Littman, M.; and Cassandra, A. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence Journal* 2.

Kalai, E., and Lehrer, E. 1993. Rational learning leads to nash equilibrium. *Econometrica* 61(5):1019–1045.

Li, M., and Vitanyi, P. 1997. *An Introduction to Kolmogorov Complexity and its Applications*. Springer.

McLachlan, G. J., and Basford, K. E. 1988. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York.

Monahan, G. E. 1982. A survey of partially observable markov decision processes: Theory, models, and algorithms. *Management Science* 28(1):1–16.

Nair, R.; Tambe, M.; Yokoo, M.; Pynadath, D.; and Marsella, S. 2003. Taming decentralized pomdps : Towards efficient policy computation for multiagent settings. In *International Joint Conference on AI*.

Pineau, J.; Gordon, G.; and Thrun, S. 2006. Anytime point-based approximations for pomdps. *Journal of AI Research (JAIR)* 27:335–380.

Seuken, S., and Zilberstein, S. 2007. Memory bounded dynamic programming for dec-pomdps. In *International Joint Conference on AI (IJCAI)*.

Smallwood, R., and Sondik, E. 1973. The optimal control of partially observable markov decision processes over a finite horizon. *Operations Research* 21:1071–1088.

Spaan, M., and Vlassis, N. 2005. Perseus: Randomized point-based value iteration for pomdps. *Journal of Artificial Intelligence Research (JAIR)* 24:195–220.

Szer, D., and Charpillet, F. 2006. Point based dynamic programming for dec-pomdps. In *Twenty First Conference on AI (AAAI)*.