Iterative Online Planning in Multiagent Settings with Limited Model Spaces and PAC Guarantees

Yingke Chen
Dept. of Computer Science
University of Georgia
Athens, GA, USA
ykchen@uga.edu

Prashant Doshi
Dept. of Computer Science
University of Georgia
Athens, GA, USA
pdoshi@cs.uga.edu

Yifeng Zeng
School of Computing
Teesside University
Middlesborough, UK
y.zeng@tees.ac.uk

ABSTRACT

Methods for planning in multiagent settings often model other agents' possible behaviors. However, the space of these models – whether these are policy trees, finite-state controllers or intentional models – is very large and thus arbitrarily bounded. This may *exclude* the true model or the optimal model. In this paper, we present a novel iterative algorithm for *online* planning that considers a limited model space, updates it dynamically using data from interactions, and provides a provable and probabilistic bound on the approximation error. We ground this approach in the context of graphical models for planning in partially observable multiagent settings – interactive dynamic influence diagrams. We empirically demonstrate that the limited model space facilitates fast solutions and that the true model often enters the limited model space.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms, Experimentation

Keywords

online planning; multiple agents; influence diagram; mental models

1. INTRODUCTION

Approaches for planning and plan recognition in cooperative and noncooperative multiagent settings [3, 9, 22] often model other agents' possible behaviors. These models could be policy trees [19], finite-state controllers [15], or intentional models with beliefs, preferences and capabilities [8]. Because the space of such models is very large – theoretically, it is countably infinite – a small subset of models is typically handpicked or arbitrarily selected. However, this precludes any guarantees that the true model or the policy tree that is part of an optimal joint plan is included.

Observing that the true behaviors of other agents are revealed only when the agents interact, we target the following problem setting in this paper: a subject agent repeatedly interacts with another agent whose behavior (guided by a plan or a policy) is fixed. Starting with a simple "baseline" planning model, how should the subject agent adapt its model to the beliefs and preferences of others in order to improve its planning? Algorithms for this problem setting enjoy many applications. They could help build new and smarter AI embedded in real-time strategy games that repeatedly interacts with the existing AI (whose programmed behavior is usually fixed) learning its strategy and using it in its own planning. The methods also find application in building a smart robotic soccer player that joins an ad hoc team of other soccer players with differently programmed play [2, 14, 21]. On repeatedly interacting with a team mate, subject robot adapts its play to the strengths of the team mate.

We focus on individual planning in multiagent settings as formalized by the graphical interactive dynamic influence diagrams (I-DIDs) [8]. Extending single-agent DIDs [11], I-DIDs are a general and *graphical* framework for sequential decision making (planning) under uncertainty from an individual agent's perspective in both competitive and cooperative settings. Emerging applications in automated vehicles that communicate [13], integration with the belief-desire-intention framework [4], and toward ad hoc teamwork [3] motivate advances for I-DIDs. Previous efforts focus on identifying behaviorally equivalent models thereby leading to a suite of techniques for lossless and lossy compressions of the model space in I-DIDs [28]. As a large space of distinct behaviors exists, these algorithms must still maintain a significantly large model space and are unable to plan for large horizons. In this context, this paper makes the following contributions:

1. We present an approach that ascribes an arbitrary size-limited set of models to other agents and *adapts* this set using trajectories from online interactions. The approach leads to a novel iterative algorithm, OPIAM, for online planning in settings shared with other agents as delineated above. The approach is also useful in other algorithms that consider models such as point-based methods for solving interactive POMDPs [7], memory-bounded dynamic programming for decentralized POMDPs [19], and online planning for ad hoc teamwork [24].
2. In a first for online planning in multiagent settings, OPIAM exhibits a provable probabilistic guarantee that the approximation error is bounded. The probabilistic error bound improves as more trajectories are obtained.
3. On two problem domains with up to 5 agents, we empirically demonstrate that by considering a limited but adaptive model space, the online planning is faster compared to the previous best compression of model spaces [27], and simultaneously obtains high average rewards. This makes OPIAM outperform current I-DID algorithms, although it has the benefit of online interactions. We observe in our experiments that the true model or its observationally equivalent counterpart almost always gets included in the limited model space as the interactions progress and the space is updated. This provides an explanation for the good quality behavior despite considering a small model space.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

2. BACKGROUND: INTERACTIVE DID

When there are multiple agents operating in a stochastic and partially observable environment, the corresponding decision making processes could be formalized as decentralized partially observable Markov decision processes (Dec-POMDPs), interactive POMDPs (I-POMDPs) or others [5]. Dec-POMDPs model cooperative agents as a team who share joint beliefs over states and a local policy is provided to each agent. I-POMDPs take the perspective of an individual agent operating in presence of other self-interested agents. They are suitable for both cooperative and competitive settings. As a graphical counterpart of I-POMDPs, I-DIDs explicitly model the problem structure and show computational advantages in complex problem domains [8]. We briefly review I-DIDs below.

2.1 Representation

I-DIDs formalize how a subject agent i optimizes its decisions while interacting with another agent j whose actions impact their common states S and rewards R. In addition to regular chance, decision and utility nodes in DID [23], a new type of node called the *model node*, $M_{j,l-1}$, models how other agent j makes its decisions simultaneously at level $l-1$. More explicitly, it contains all possible j's models whose solutions give the predicted behavior A_j, which is represented by a *policy link* (the dashed line) connecting $M_{j,l-1}$ and A_j. Each model, $m_{j,l-1}$, could be either a level $l-1$ I-DID or a DID at level 0 where the other agent is not further modeled. Here, level pertains to the recursive reasoning between agents and agents at a low level do not model agents at a higher level. For example, a level 1 agent considers agents at level 0 while level 0 agents only model the environment.

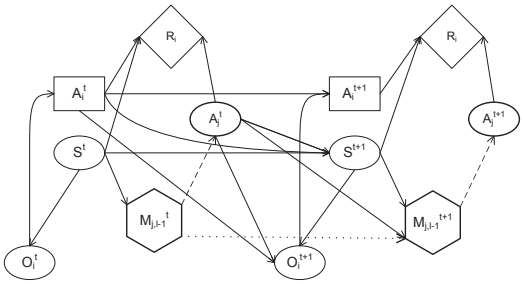

Figure 1: A generic two time-slice level l I-DID for agent i. Policy links are marked as dash lines, while the model update link is marked as a dotted lines.

Besides the physical state, agent i's belief is over possible models of agent j. Moreover, as j acts and receives observations over time, its models are updated to reflect its changed beliefs. The *model update link*, a dotted arrow from $M^t_{j,l-1}$ to $M^{t+1}_{j,l-1}$ in Fig. 1, represents the update of j's models over time. The updated models differ in the beliefs that obtain for a pair of j's actions and observations. Consequently, the set of updated models at time $t+1$ will have up to $|\mathcal{M}^t_{j,l-1}||A_j||\Omega_j|$ models. Here, $|\mathcal{M}^t_{j,l-1}|$ is the number of models at time step t, and $|A_j|$ and $|\Omega_j|$ are the largest spaces of actions and observations respectively. I-DID becomes a regular DID when the model update link is replaced with regular dependency links and chance nodes. We may employ any DID solution technique to solve an I-DID. For clarity, we elaborate an example I-DID for the well-studied multiagent tiger problem [9].

EXAMPLE 1 (LEVEL 1 I-DID). *Figure 2 shows the I-DID for a level 1 agent i who considers two models of j at level 0 in the two-agent tiger problem. The two models, $m^{t,1}_{j,0}$ and $m^{t,2}_{j,0}$, differ in j's belief about the tiger's location and are included in the model node $M^t_{j,0}$. As indicated by the conditional probability table (CPT) in Fig. 4, solving the models results in j's optimal decisions, OL and L, respectively.*

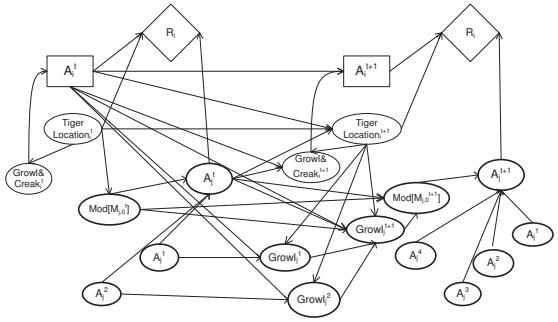

Figure 2: A two time-slice level 1 I-DID for agent i in the tiger problem. The model update link has been replaced by regular arcs in DIDs.

We show the update of $m^{t,1}_{j,0}$ and $m^{t,2}_{j,0}$ in Fig. 3. As agent j may receive one of two observations (either GL or GR), four new models are generated in the model node $M^{t+1}_{j,0}$.

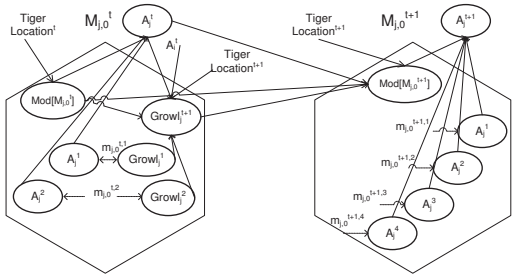

Figure 3: Details of the model update link where two models are expanded into four models in the new time step.

The CPT of $Mod[M^{t+1}_{j,0}]$ is shown in Fig. 4. For example, the first row of the CPT shows that $m^{t,1}_{j,0}$ is updated into the model $m^{t+1,1}_{j,0}$ when agent j takes the action OL and observes GL in the next time step. As neither OR nor L is the optimal decision for $m^{t,1}_{j,0}$, we assign a uniform distribution to indicate that $m^{t,1}_{j,0}$ does not transform into any of the new models for these actions.

Mod[$M_{j,0}^t$]	OL	OR	L
$m_{j,0}^{t,1}$	1	0	0
$m_{j,0}^{t,2}$	0	0	1

CPT of A_j^t

Decisions(A_j)
OL: Open the left door
OR: Open the right door
L: Listen
Observations(Growl$_j$)
GL: Growl from the left door
GR: Growl from the right door

<A_j^t, Growl$_j^{t+1}$>	Mod[$M_{j,0}^t$]	$m_{j,0}^{t+1,1}$	$m_{j,0}^{t+1,2}$	$m_{j,0}^{t+1,3}$	$m_{j,0}^{t+1,4}$
<OL, GL>	$m_{j,0}^{t,1}$	1	0	0	0
<OL, GR>	$m_{j,0}^{t,1}$	0	1	0	0
<L, GL>	$m_{j,0}^{t,2}$	0	0	1	0
<L, GR>	$m_{j,0}^{t,2}$	0	0	0	1
<OR, * >	*	1/4	1/4	1/4	1/4
<L, * >	$m_{j,0}^{t,1}$	1/4	1/4	1/4	1/4
<OL, * >	$m_{j,0}^{t,2}$	1/4	1/4	1/4	1/4

CPT of Mod[$M_{j,0}^{t+1}$]

Figure 4: The CPTs of the chance nodes A^t_j and $Mod[M^{t+1}_{j,0}]$.

2.2 Solutions

Solving a level l I-DID requires the expansion and solution of j's models at level $l-1$. We outline the exact I-DID algorithm in Fig. 5. Lines 4-5 solve j's models to obtain the policy link. Line 6 invokes techniques for compression of the model space based on *behavioral equivalence* [17], **PruneBehavioralEq** ($\mathcal{M}_{j,l-1}$), and returns representative models of j. Lines 7-15 implement the model update link in an I-DID. Finally, lines 17-18 solve the transformed I-DID using standard DID algorithms. Previous offline techniques such as DMU [26] and ϵ-BE [28] solve I-DIDs by exploiting equivalences between models. In particular, DMU exactly solves I-DIDs while ϵ-BE compromises the solution quality to achieve greater efficiency.

I-DID EXACT(level $l \geq 1$ I-DID or level 0 DID, horizon T)
Expansion Phase
1. **For** t **from** 0 **to** $T-1$ **do**
2. **If** $l \geq 1$ **then**
Populate $M^{t+1}_{j,l-1}$
3. **For each** m^t_j **in** $\mathcal{M}^t_{j,l-1}$ **do**
4. Recursively call algorithm with the $l-1$ I-DID (or DID) that represents m^t_j and horizon, $T-t$
5. Map the decision node of the solved I-DID (or DID), $OPT(m^t_j)$, to the corresponding chance node A_j
6. $\mathcal{M}^t_{j,l-1} \leftarrow$ **PruneBehavioralEq**($\mathcal{M}^t_{j,l-1}$)
7. **For each** m^t_j **in** $\mathcal{M}^t_{j,l-1}$ **do**
8. **For each** a_j **in** $OPT(m^t_j)$ **do**
9. **For each** o_j **in** O_j (part of m^t_j) **do**
10. Update j's belief, $b^{t+1}_j \leftarrow SE(b^t_j, a_j, o_j)$
11. $m^{t+1}_j \leftarrow$ New I-DID (or DID) with b^{t+1}_j
12. $\mathcal{M}^{t+1}_{j,l-1} \overset{\cup}{\leftarrow} \{m^{t+1}_j\}$
13. Add the model node, $M^{t+1}_{j,l-1}$, and the model update link
14. Add the chance, decision, and utility nodes for $t+1$ time slice and the dependency links between them
15. Establish the conditional probability tables (CPTs) for each chance and utility node
Solution Phase
16. **If** $l \geq 1$ **then**
17. Represent the model nodes, policy links and the model update links to obtain the DID
18. Use standard look-ahead and backup to solve the expanded DID

Figure 5: Algorithm for exactly solving a level $l \geq 1$ I-DID or level 0 DID expanded over T time steps.

3. ONLINE PLANNING WITH LIMITED MODEL SPACE

I-DIDs generally maintain a sufficiently large set of candidate models to capture possible behavior of other agents. Previous research focuses on reducing the expanding space of models at every time step [26]. Nevertheless, the methods are challenged by the *exponential growth* in the number of models over time. Moreover, the existing *de-facto* method for updating distributions over the models is to use a Bayesian update that does not adapt the model space – prune or replace candidate models.

Using the insight that observations received by agent i can reveal the actions performed by agent j which depend on its model, we propose a new algorithm that exploits agents' interactions to improve I-DID solutions by focusing on identifying agent j's true behavioral model. We outline the algorithm and discuss the steps.

3.1 Algorithm Outline

Methods that model others operate in the context of *two* sets of models: a large universal set of models and the model set considered by the method. Given agent i's belief distribution over a set of j's models, $b_{i,l}(\mathcal{M}_{j,l-1})$, previous algorithms [26] solve i's I-DID with the *complete* set of j's models. Let $\mathcal{M}^0_{j,l-1}$ denote the set of j's models included in the model node of the I-DID initially. Then, in these approaches $\mathcal{M}^0_{j,l-1} = \mathcal{M}_{j,l-1}$. Subsequently, agent i interacts with j using the policy obtained by solving the I-DID and empirically updates $b_{i,l}(\mathcal{M}_{j,l-1})$ accordingly. Doshi and Gmytrasiewicz show that $b_{i,l}(\mathcal{M}_{j,l-1})$ converges to a probability distribution, denoted by $b^*_{i,l}(\mathcal{M}_{j,l-1})$, after a sufficiently large number of interactions if the initial belief satisfies the absolute continuity condition [6].

In contrast, our new algorithm, OPIAM (**Online plan, interact and adapt models**), solves an initial "baseline" I-DID using a *partial* set of j's models: $\mathcal{M}^0_{j,l-1} \subset \mathcal{M}_{j,l-1}$. Agent i then uses the solved policy to interact with j for some time and updates $b_{i,l}(\mathcal{M}^0_{j,l-1})$ using the trajectories of its actions and observations obtained from the interactions. Subsequently, OPIAM modifies the I-DID by partially replacing $\mathcal{M}^0_{j,l-1}$ with models from the unexplored set ($\mathcal{M}_{j,l-1}/\mathcal{M}^0_{j,l-1}$).

We present OPIAM in Fig. 6. Given a large set of models of agent j, $\mathcal{M}_{j,l-1}$, we randomly select a *small* set of candidate models, $\mathcal{M}^0_{j,l-1}$, and initialize the I-DID, $m_{i,l}$, with a uniform distribution over the models (lines 1-2). Solving $m_{i,l}$ provides an initial policy that agent i executes to play with agent j online (line 6). Formally, we denote a T-horizon policy as $\pi^T_{m_{i,l}}$ that is represented using a tree and contains a set of policy paths from the root node to the leaf.

DEFINITION 1 (POLICY PATH). *A policy path, $h^{T,(k)}_i$, is an action-observation sequence of T steps: $h^{T,(k)}_i = \{a^t_i, o^{t+1}_i\}^{T-1}_{t=0}$, where o^T_i is null.*

After an interaction, agent i weights each of j's candidate model, $Pr(m_{j,l-1}|h^{T,(k)}_i)$, given i's observations and executed actions up to T steps (line 7). This computation uses Bayes rule in a straightforward way:

$$Pr(m_{j,l-1}|h^{T,(k)}_i) \propto \sum\nolimits_{s^0} Pr(m_{j,l-1}|s^0) Pr(h^{T,(k)}_i|m_{j,l-1}, s^0)$$

where $Pr(m_{j,l-1}|s^0)$ is the prior weight of j's model given the state and $Pr(h^{T,(k)}_i|m_{j,l-1}, s^0)$ the likelihood of i's policy path. We compute this by inserting $h^{T,(k)}_i$ as evidence into the corresponding decision and chance nodes, and $m_{j,l-1}$ in the model node of the level l I-DID, $m_{i,l}$.

After N interactions, the average weight of each model is obtained, $\sum^N_{k=1} Pr(m_{j,l-1}|h^{T,(k)}_i)/N$ (line 10). As not all of j's models are included in the current I-DID, OPIAM next updates the belief over the partial set of models in the larger model space $\mathcal{M}_{j,l-1}$, denoted by $b^\tau_{i,l}(\mathcal{M}^0_{j,l-1})$. This is done by redistributing the probability mass in i's belief for the selected models in $\mathcal{M}^0_{j,l-1}$ over all j's models in $\mathcal{M}^0_{j,l-1}$ proportionally to their average weights obtained in the previous step. This is shown in Eq. 1:

$$b^\tau_{i,l}(m_{j,l-1}) = \frac{1}{N}\sum_{k=1}^{N} Pr(m_{j,l-1}|h^{T,(k)}_i) \times \sum_{m_{j,l-1} \in \mathcal{M}^0_{j,l-1}} b^{\tau-1}_{i,l}(m_{j,l-1}) \quad (1)$$

where the second factor that sums $b^{\tau-1}_{i,l}(m_{j,l-1})$ over all models in $\mathcal{M}^0_{j,l-1}$ is the total probability mass over models included in the I-DID, and $Pr(m_{j,l-1}|h^{T,(k)}_i)$ is the updated weight of model $m_{j,l-1}$ given i's policy path, $h^{T,(k)}_i$.

OPIAM(level $l \geq 1$ I-DID of agent i, $m_{i,l}$; candidate models of agent j at level $l-1$ or level 0, $\mathcal{M}_{j,l-1}$; horizon T; ρ)

1. Weight $\mathcal{M}_{j,l-1}$ equally and set the counter, $\tau = 1$
2. Select a subset of j's candidate models, $\mathcal{M}^0_{j,l-1} \subset \mathcal{M}_{j,l-1}$
3. **Do**
4. Solve agent i's model, $m_{i,l}$ with $\mathcal{M}^0_{j,l-1}$, using **I-DID EXACT** in Fig. 5, and output i's policy $\pi_{m_{i,l}}$
   Interaction Phase
5. **For** k = 1 **to** N interactions of length T **do**
6. Agent i plays with j according to the policy $\pi_{m_{i,l}}$
7. Compute the posterior weights of selected j's models $Pr(m_{j,l-1}|h_i^{T,(k)})$
8. Compute j's most probable path ξ_j^T and update its occurrence ω_{ξ_j}
9. **For each** $m_{j,l-1}$ **in** $\mathcal{M}^0_{j,l-1}$ **do**
10. Average the posterior weight: $\sum_{k=1}^{N} Pr(m_{j,l-1}|h_i^{T,(k)})/N$
11. Update beliefs about the selected models $b^{\tau}_{i,l}(\mathcal{M}^0_{j,l-1})$ using Eq. 1
12. Calculate Δ= $\|b^{\tau}_{i,l}(\mathcal{M}_{j,l-1})$-$b^{\tau-1}_{i,l}(\mathcal{M}_{j,l-1})\|_2$
13. **If** $\Delta \leq \rho$ **then**
14. Set $\tau = 0$
15. **else**
16. Set $\tau = \tau + 1$
    Model Adaptation Phase
17. Aggregate most probable paths ξ_j^T into $\mathcal{H}_j^T$
18. Select the most probable model, $\dot{m}_{j,l-1} \in \mathcal{M}_{j,l-1}/\mathcal{M}^0_{j,l-1}$
19. Select model, $m_{j,l-1} \in \mathcal{M}^0_{j,l-1}$, with the least average weight
20. Replace $m_{j,l-1}$ with $\dot{m}_{j,l-1}$ in $\mathcal{M}^0_{j,l-1}$
21. **While** $\tau > 0$

Figure 6: OPIAM **in combination with a level l I-DID expanded over T steps allows online planning with a limited model space.**

Subsequently, OPIAM prunes the model in $\mathcal{M}^0_{j,l-1}$ with the lowest weight and replaces it with one from the remaining models (lines 19-20). The initial belief in the I-DID for the next round of interactions is a normalized distribution over the updated $\mathcal{M}^0_{j,l-1}$. We terminate model adaptation when the change in the belief distribution is less than a small value, ρ, during two consecutive iterations (lines 12-14).

In principle, the online algorithm allows agent i to explore j's possible models with the overall goal of identifying j's true behavior. Next, we present a method for selecting a model in $\mathcal{M}_{j,l-1}$ as replacement in the *model adaptation* phase (lines 17-18).

3.2 Most Probable Model Selection

A key observation is that while the true model of agent j cannot be directly observed, j's behavior induces i's observations, which in turn provide inferential information about j's models. Hence, we focus on the challenge of selecting the most probable model of j based on i's action-observation history for adapting model space, $\mathcal{M}^0_{j,l-1}$.

We point out that every interaction involves only one particular policy path of j. After every interaction, agent i infers the most probable policy path for agent j given i's actions and observations. Formally, we define the most probable path below.

DEFINITION 2 (MOST PROBABLE PATH). *Given agent i's model, $m_{i,l}$, and i's sequence of actions and observations, $h_i^{T,(k)}$, define the most probable path for agent j as:*

$$\xi_j^{T,(k)} = \arg\max_{h_j^T \in H_j^T} Pr(h_j^T|h_i^{T,(k)})$$
$$= \arg\max_{h_j^T \in H_j^T} \prod_{t=1}^{T-1} Pr(a_j^t|h_i^{T,(k)}, h_j^{t-1}, o_j^t)\ Pr(o_j^t|h_i^{T,(k)}, h_j^{t-2})$$
$$\times\ Pr(a_j^0|h_i^{T,(k)})$$

The computation factorizes the joint probability $Pr(h_j^T, h_i^{T,(k)})$ given the graphical structure of the I-DID, and is carried out through a usual evidence propagation in the level l I-DID. The value of each term above is obtained from the distributions in nodes in the I-DID. Here, $H_j^T = A_j \times \prod_{t=1}^{T-1}(\Omega_j \times A_j)$ are all possible policy paths of agent j of T steps as in Def. 1.

Because agent j's set of possible paths is large, we may select the most probable path approximately by sampling the most probable action and observation at every time step. Specifically, we compute the most probable action by maximizing $Pr(a_j^0|h_i^{T,(k)})$ at time $t = 0$, and subsequently sample the most probable observation and actions over time.

Let $\mathcal{H}_j^T$ be the set of the most probable paths after N interactions, $\mathcal{H}_j^T = \bigcup_{k=1}^{N} \xi_j^{T,(k)}$. Subsequently, $\mathcal{H}_j^T$ will compose the entire policy tree that j is using, $\pi^T_{m_j^*}$, if agent i interacts with j for a sufficiently long time, i.e., $N \to \infty$.

Consider a candidate model, $m_{j,l-1} \in \mathcal{M}_{j,l-1}$, whose solution is a T-step policy tree, $\pi^T_{m_j}$. Let $Pr(a_j^t|\pi^T_{m_j})$ be the proportion among all actions that action, a_j, appears at time step t in all paths of the policy tree. *We seek a measure of how well the candidate model, $m_{j,l-1}$, fits the inferred most probable action-observation histories of j.* Let δ^T denote this measure and we define it as:

$$\delta^T(m_{j,l-1}, \mathcal{H}_j^T) \triangleq \sum_t \sum_{a_j \in A_j} |Pr(a_j^t|\pi^T_{m_j}) - Pr(a_j^t|\mathcal{H}_j^T)| \quad (2)$$

where $Pr(a_j^t|\mathcal{H}_j^T) \triangleq \sum_{\xi_j^{T,(k)} \in \mathcal{H}_j^T} \omega_{\xi_j^T} \cdot Pr(a_j^t|\xi_j^{T,(k)})$. Here, $Pr(a_j^t|\xi_j^{T,(k)})$ is 1 if action a_j appears in the most probable path, $\xi_j^{T,(k)}$, at time step t and weight $\omega_{\xi_j^T}$ is the normalized occurrence of $\xi_j^{T,(k)}$ over N interactions.

Subsequently, we obtain the most probable model for replacement as shown below.

DEFINITION 3 (MOST PROBABLE MODEL). *Given the set of agent j's models, $\mathcal{M}_{j,l-1}$, and most probable paths, $\mathcal{H}_j^T$, define the most probable model, $\dot{m}_{j,l-1}$, for the level $l-1$ agent j as:*

$$\dot{m}_{j,l-1} = \arg\min_{m_{j,l-1} \in \mathcal{M}_{j,l-1}/\mathcal{M}^0_{j,l-1}} \delta^T(m_{j,l-1}, \mathcal{H}_j^T)$$

We elaborate the computation of the most probable model using an example in the multiagent tiger problem [9] below.

EXAMPLE 2. *Assume that the initial model set, $\mathcal{M}^0_{j,l-1}$, comprises of three models, $\{m_j^1, m_j^2, m_j^3\}$, and another candidate model, m_j^4, is available to be selected for updating $\mathcal{M}^0_{j,l-1}$. Figure 7 shows a combination of policy trees ($T = 3$) obtained by solving every model in $\mathcal{M}^0_{j,l-1}$ (on the left), and the policy tree of m_j^4 (on the right). After $N = 6$ interactions, $\mathcal{H}_j$ consists of three policy paths:* ξ_1=$\langle OL, GR, L, GR, OL\rangle$, ξ_2=$\langle L, GL, L, GR, L\rangle$

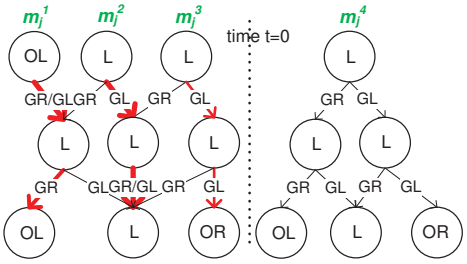

Figure 7: Policy trees of initial models, $\mathcal{M}^0_{j,l-1}$ (left), and the candidate model, m^4_j (right). Node and edge labels are j's actions and observations respectively. Darker (red) edges indicate the occurrence of most probable paths over interactions.

and ξ_3=$\langle L, GL, L, GL, OR\rangle$. These are inferred with the (normalized) frequencies, $\omega_{\xi_1} = 1/6$, $\omega_{\xi_2} = 2/3$ and $\omega_{\xi_3} = 1/6$, respectively.

$Pr(a^t_j|\pi_{m^4_j})$ *at the different time steps is given below. The superscripts on actions denote the time step.*

$$Pr(L^1|\pi_{m^4_j}) = 1;\ Pr(OL^1|\pi_{m^4_j}) = 0;\ Pr(OR^1|\pi_{m^4_j}) = 0;$$
$$Pr(L^2|\pi_{m^4_j}) = 1;\ Pr(OL^2|\pi_{m^4_j}) = 0;\ Pr(OR^2|\pi_{m^4_j}) = 0;$$
$$Pr(L^3|\pi_{m^4_j}) = 1/2;\ Pr(OL^3|\pi_{m^4_j}) = 1/4;\ Pr(OR^3|\pi_{m^4_j}) = 1/4;$$

Next, we compute $Pr(a_j|\mathcal{H}_j)$ for each action at each time step.

$$Pr(L^1|\mathcal{H}_j) = 5/6;\ Pr(OL^1|\mathcal{H}_j) = 1/6;\ Pr(OR^1|\mathcal{H}_j) = 0;$$
$$Pr(L^2|\mathcal{H}_j) = 1;\ Pr(OL^2|\mathcal{H}_j) = 0;\ Pr(OR^2|\mathcal{H}_j) = 0;$$
$$Pr(L^3|\mathcal{H}_j) = 4/6;\ Pr(OL^3|\mathcal{H}_j) = 1/6;\ Pr(OR^3|\mathcal{H}_j) = 1/6;$$

Finally, the distance, δ, between the two distributions gives a measure of closeness of m^4_j to the observed trajectories:

$$\delta = (|1 - 5/6| + |0 - 1/6|) + (|1 - 1|) + (|1/2 - 2/3| + |1/4 - 1/6| + |1/4 - 1/6|) = 2/3$$

4. SAVINGS AND PAC BOUND

It is well known that the primary complexity of solving I-DIDs is due to the exponentially growing number of j's models over time. At time step t, there could be $|\mathcal{M}^0_{j,l-1}|(|A_j||\Omega_j|)^t$ many models of the other agent j, where $|\mathcal{M}^0_{j,l-1}|$ is the number of models considered initially. Previous approaches consider the entire candidate set of j's models, $\mathcal{M}^0_{j,l-1} = \mathcal{M}_{j,l-1}$, where $\mathcal{M}_{j,l-1}$ is generally large as it seeks to cover as much as feasible possible models of j.

In contrast, OPIAM considers a relatively small set of j's models initially and iteratively explores the rest online. While the adaptation phase eventually solves all models in the initial $\mathcal{M}_{j,l-1}$ (Eq. 2), the primary computational savings arise in the state space during the look ahead expansion: $|\mathcal{M}^0_{j,l-1}|(|A_j||\Omega_j|)^t \ll |\mathcal{M}_{j,l-1}|(|A_j||\Omega_j|)^t$, because $|\mathcal{M}^0_{j,l-1}| \ll |\mathcal{M}_{j,l-1}|$. This makes solving I-DIDs faster – facilitating real-time solution constraints – and scales up the solution in the planning horizon.

Recall that we adapt the model space by including the most probable model. As indicated in Def. 3, the model $\dot{m}_{j,l-1}$ is likely to be selected when its predicted actions, $Pr(a_j|\dot{m}_{j,l-1})$, are similar to those found in the most probable paths, $Pr(a_j|\mathcal{H}^T_j)$, in the comparison between policy paths. Hence, exploring j's model space online and including the most probable model reduces agent i's loss in value due to misprediction. We take further steps to bound such loss after each iteration of N interactions.

We consider the (typical) condition that agent j's true model $m^*_{j,l-1}$ is part of $\mathcal{M}_{j,l-1}$. Let η be the worst L_1-norm error in the prediction of j's actions at any time step due to the model replacement, $\eta = \max_{t\in T} \sum_{a_j\in A_j} |Pr(a^t_j|\dot{m}_{j,l-1}) - Pr(a^t_j|m^*_{j,l-1})|$. The expected value of agent i's optimal policy given by the I-DID is:

$$V^T(m_{i,l}) \quad = \rho(b_{i,l}, a^*_i) + \textstyle\sum_{o_i} Pr(o_i|b_{i,l}, a^*_i) V^{T-1}(m'_{i,l})$$

where $\rho(b_{i,l}, a^*_i) = \sum_{s,m_{j,l-1}} b_{i,l}(s, m_{j,l-1}) \sum_{a_j} R_i(s, a^*_i, a_j) \times Pr(a_j|m_{j,l-1})$. Here, a^*_i is i's optimal action and $m'_{i,l}$ is the updated model of agent i containing the updated belief at the next time step. We denote the expected value and the immediate expected reward of agent i's optimal policy when j's predicted behavior is $Pr(a_j|\dot{m}_{j,l-1})$, instead of $Pr(a_j|m^*_{j,l-1})$, as $\dot{V}^T(m_{i,l})$ and $\dot{\rho}(b_{i,l}, a_i)$, respectively. Denote the maximal reward value in the reward function as R^{max}_i.

Proposition 1 with help from Lemmas 1 and 2 gives the difference in the value functions due to the misprediction, η.

LEMMA 1. *For any i's belief, $b_{i,l}$, and action, a_i,*

$$|\rho(b_{i,l}, a_i) - \dot{\rho}(b_{i,l}, a_i)| \leq \eta R^{max}_i$$

LEMMA 2. *For any pair of updated models of agent i, $m'_{i,l}$, and $\dot{m}'_{i,l}$, where the latter is due to differing predictions of j's actions,*

$$|Pr(o_i|b_{i,l}, a_i)\ V^{T-1}(m'_{i,l}) - \dot{P}r(o_i|b_{i,l}, a_i)\ V^{T-1}(\dot{m}'_{i,l})| \leq 3\eta(T-1)\ R^{max}_i|\Omega_j|$$

We use Lemmas 1 and 2 to establish Proposition 1. Brief proofs of the lemmas and propositions below are given in the Appendix.

PROPOSITION 1. *For a given I-DID with initial belief of agent i, $m_{i,l}$, let $V^T(m_{i,l})$ be the expected reward from solving a T-horizon I-DID optimally, and $\dot{V}^T(\dot{m}_{i,l})$ be the expected reward from solving a T-horizon I-DID by considering the most probable model instead of the true model in the set. Let $\mathcal{E}^T \triangleq |V^T(m_{i,l}) - \dot{V}^T(\dot{m}_{i,l})|$. Then,*

$$\mathcal{E}^T \leq \eta R^{max}_i((T-1)(1 + 3(T-1)|\Omega_i||\Omega_j|) + 1)$$

Proposition 1 does not as yet bound the error because a non-trivial bound for η is not established. Proposition 2 provides the crucial missing piece probabilistically bounding η using δ^T and an error term, ϵ, that is flexible.

PROPOSITION 2. $\eta \leq (\delta^T + \epsilon)$ *with probability at least* $1 - \frac{|A_j|^T}{e^{2NT(\epsilon/|A_j|^T)^2}}$.

Together, Propositions 1 and 2 establish a PAC bound on the worst error incurred by OPIAM after a single iteration of N interactions between agents i and j. The error accumulates with more iterations until the algorithm terminates due to which the overall bound on the error of OPIAM is loose. For the pathological condition when $m^*_{j,l-1} \notin \mathcal{M}_{j,l-1}$, the probability is not guaranteed to improve with increasing samples unless $b_{i,l}(\mathcal{M}_{j,l-1})$ continues to satisfy the absolute continuity condition [12].

5. EXPERIMENTAL RESULTS

We implemented OPIAM (Fig. 6) to improve planning in level 1 I-DIDs. As a baseline, another model replacement technique (replacing line 18 in Fig. 6) that *randomly* selects a new model from

$\mathcal{M}_{j,l-1}/\mathcal{M}^0_{j,l-1}$, labeled as ROnline is used. Previous best approach for solving I-DIDs compares partial policy trees and belief distributions at the leaf nodes for approximate equivalence and clustering models [28]. This method, labeled as ϵ-BE, compacts the complete model space, $\mathcal{M}_{j,l-1}$; it is also allowed to interact and update priors. This approach serves as a high quality benchmark. A non-adaptive exact method, DMU [26], serves to demonstrate the benefits of adaptation. All experiments are run on a Windows platform with 1.9 GHz i3 processor and 6 GB memory.

We compare their performances on two noncooperative problem domains: the small multiagent tiger ($|S|$=2, $|A_i|$=$|A_j|$=3, $|\Omega_i|$=6, $|\Omega_j|$=3, T=6) and the larger multi-unmanned aerial vehicle (multiUAV) reconnaissance problem ($|S|$=25, $|A_i|$=$|A_j|$=5,$|\Omega_i|$=$|\Omega_j|$=5, T=4) [26]. Models of other agents are IDs that encode the problem and differ in their initial beliefs. Though this space is continuous, BE offers a way to make this space discrete and include the true model. Instead, we evaluated all algorithms for two arbitrary sets of candidate models – $\mathcal{M}_j$ = 50 or 100 – from which we select different smaller sets of initial models ($\mathcal{M}^0_j$). This allowed us to experiment with settings where the true models of others are not in $\mathcal{M}_j$. The parameter ρ that guides the termination of the algorithm (Fig.6, line 13) is set as 0.01. Proposition 2 provides a way to obtain N given T, ϵ and least probability. This resulted in $N \approx 100$ (T = 6, $|A_j|$=3, ϵ=0.15 and least probability 0.9) and it varies with T.

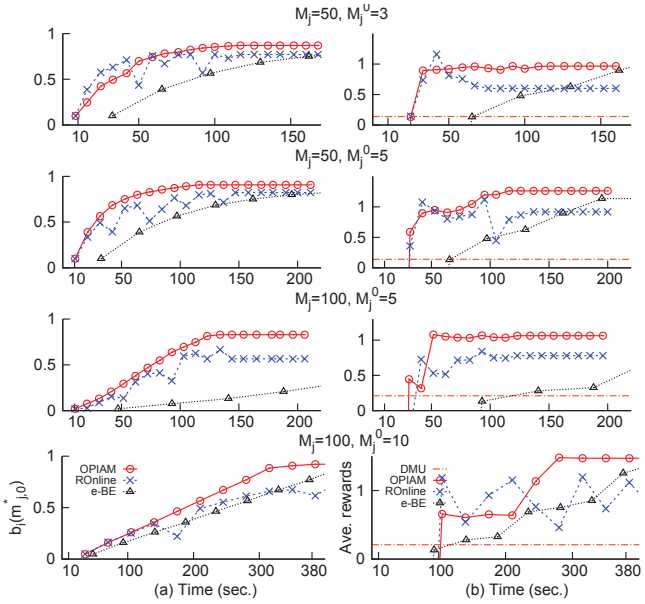

Figure 8: Increasing model weights on the true model (left) and improving average rewards (right) over time for the case: $m^*_{j,l-1} \in \mathcal{M}_{j,l-1}$, in the multiagent tiger problem. The horizontal line denotes the average reward from the exact DMU method given a uniform prior over $\mathcal{M}_{j,l-1}$.

We evaluate OPIAM's performance under two conditions: (i) A typical case when the true model, $m^*_{j,l-1} \in \mathcal{M}_{j,l-1}$; and (ii) Because $\mathcal{M}_{j,l-1}$ is itself finite in practice, a pathological and bold case where $m^*_{j,l-1} \notin \mathcal{M}_{j,l-1}$. The latter is often utilized to test methods for ad hoc cooperation.

For the former case, Fig. 8 (*left*) demonstrates the probability mass on the true model with time for different configurations of $\mathcal{M}^0_{j,l-1}$ and $\mathcal{M}_{j,l-1}$ in the tiger problem. Increasing weights indicate that $m^*_{j,l-1}$ enters the smaller model set. More importantly,

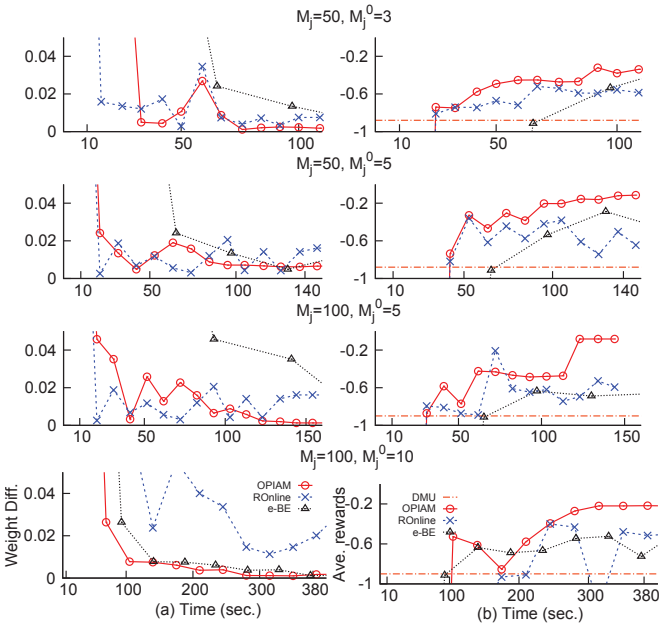

Figure 9: Model weight differences (left) and average rewards (right) for the case: $m^*_j \notin \mathcal{M}_j$, in tiger. The weight difference, Δ in line 12 of Fig. 6 reaches zero.

this model receives much more attention from agent i over time than its initial weight. Here, time is a function of increasing iterations and interaction lengths. Observe that the model weight stabilizes and the difference in distributions approaches zero with the algorithm terminating.

Simultaneously, Fig. 8(*right*) shows the reward obtained over T steps averaged over N. This increases with time indicating that OPIAM is progressively generating higher-valued policies that better predict agent j's actions. ROnline's performance is uneven on both the true model probability and average rewards. Importantly, OPIAM is faster than ϵ-BE whose performance also improves with time due to improved priors, but holding the entire model space slows it down considerably despite its aggressive compression. Each point on the curves indicates a model space adaptation (by OPIAM and ROnline) and prior update (by OPIAM, ROnline and ϵ-BE). Generally fewer points for ϵ-BE suggests that it is slow, despite its compression, in generating a new plan with updated priors compared to others.

For the latter case, Fig. 9 (*left*) shows the differences in model weights over time, as mentioned in line 12 of Fig. 6. The difference eventually approaches zero and stays there. The average rewards do not improve as steadily as they do in the previous case, and they reach values that are close to positive but lower than those when the true model is present in the space. However, OPIAM's performance remains significantly better than ϵ-BE, ROnline and DMU. This is insightful revealing that a combination of models, which can be seen as an approximation, may partly compensate for the absence of the true model from the considered space.

In Figs. 10 and 11, we show the performance of OPIAM in the context of the larger multiUAV problem. While the time duration has increased, OPIAM continues to exhibit average rewards that improve on both ϵ-BE and ROnline. In this larger problem domain, ϵ-BE carried out much less rounds of prior update in the given time period. In some cases, the interaction based on ϵ-BE takes too much time to be shown. Importantly, the true model enters $\mathcal{M}^0_{j,l-1}$ when it's included in the larger set. But, in its absence, the model weight

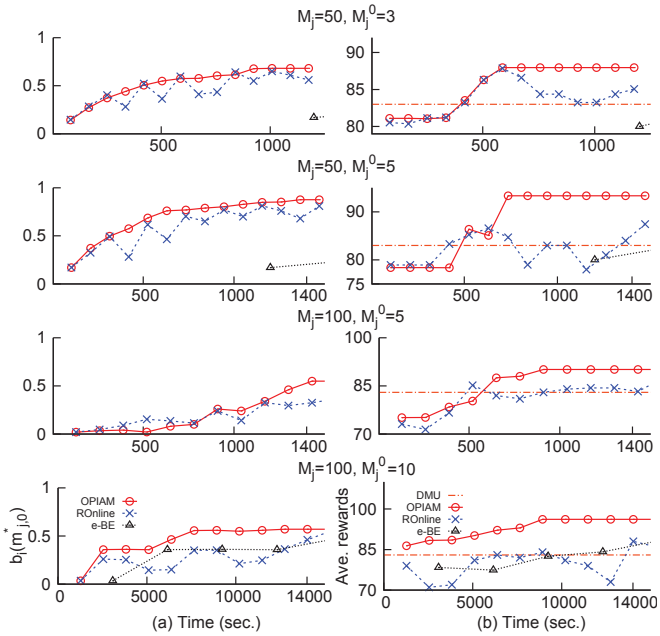

Figure 10: Most probable model weights and average rewards for the case: $m_j^* \in \mathcal{M}_j$, in the larger multiUAV problem.

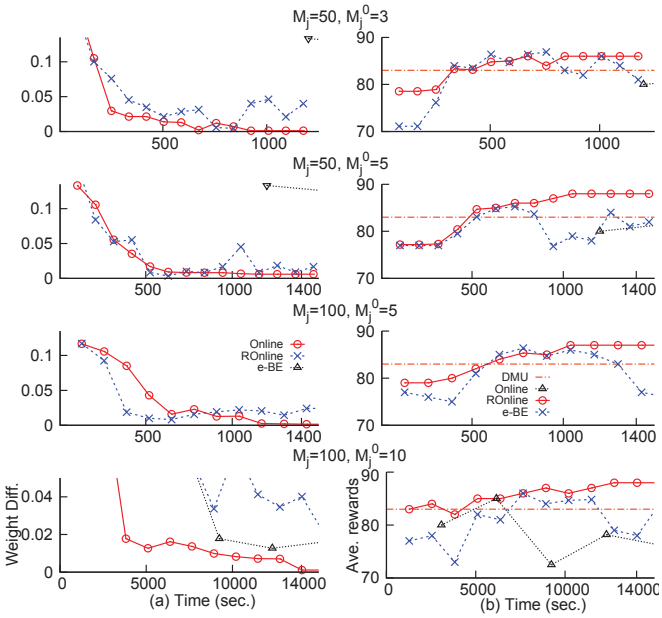

Figure 11: Weight differences over time and average rewards for the case: $m_j^* \notin \mathcal{M}_j$, in UAV problem.

difference approaches zero and stays there. Helped by the online interactions, both OPIAM and ϵ-BE reach rewards that improve on the exact DMU indicated by the flat line.

Finally, we demonstrate the speed up that OPIAM brings in problems with longer planning horizons and when more agents are sharing the setting. Table 1 shows significant speed up compared to ϵ-BE; speed up that increases with problem sizes. ϵ-BE exhibits poor scalability with more agents and the corresponding speed ups improve to multiple orders of magnitude. Such speed ups gradually bring online planning in partially observable multiagent settings to the realistic time requirements imposed on these algorithms.

2-agent Tiger (sec)				**2-agent UAV** (sec)			
T	ϵ-BE	OPIAM	Speedup	T	ϵ-BE	OPIAM	Speedup
8	65	11.7	5.6	6	2,973	106	28
15	217	67	3.2	8	*	836	n.a.
5-agent Tiger (sec)				**4-agent UAV** (sec)			
T	ϵ-BE	OPIAM	Speedup	T	ϵ-BE	OPIAM	Speedup
3	13,818	590	23	2	29,638	840	35
6	*	8,394	n.a.	4	*	7,840	n.a.

Table 1: OPIAM exhibits significant speed up compared to ϵ-BE with comparable average rewards. These times reflect multiple rounds until convergence.

6. RELATED WORK

In settings where others' actions are not directly observable but may be inferred through state changes, Sonu and Doshi [20] present a bimodal approach in which the subject agent initially uses a single-agent POMDP controller, and switches online to a multiagent controller when it is sufficiently certain of the physical state. Unlike OPIAM, planning is offline and model space is not updated based on data.

While approaches such as OPIAM that target settings shared with other agents with possibly conflicting preferences have been sparse, multiple approaches of online planning for cooperation exist. Wu et al. [25] use policy equivalence to reduce the size of histories so that agents can continue to coordinate under the condition of limited communication. A second approach, OPAT, by Wu et al. [24] performs online planning for ad hoc teams. It targets a simpler problem setting where the state and joint actions are fully observable. However, Chandrasekaran et al. [3] show that extending OPAT to partial observability did not demonstrate better performance than an I-DID based solution when the type of the other agent is not known a'priori. Harsanyi-Bellman Ad Hoc Coordination (HBA) [1] provides a generalized stochastic game framework that maintains a distribution over a set of user-defined teammate types. Reinforcement learning is utilized to learn the agent's optimal actions online. However, similar to OPAT, HBA targets settings where the states and joint actions of others are perfectly observed.

Different from existing online POMDP-based methods [18], OPIAM targets multiagent settings and *adapts* the model space thereby repeatedly changing the state space of the problem. OPIAM's focus on a limited set of others' models is reminiscent of point-based value iteration for POMDPs [16] and interactive POMDPs [7]. In these techniques, limited belief points are generated from simulated trajectories of the subject agent. However, a key difference is that OPIAM selects models based on probabilistic observations by the subject agent of *others'* trajectories, which help build the most probable paths of others. As we mentioned previously, OPIAM's approach could also be utilized in other algorithms that consider models such as memory-bounded dynamic programming for decentralized POMDPs [19].

7. CONCLUDING REMARKS

A key contribution of OPIAM is its capability to work with a small model space and adapt it in order to perform fast online planning. Empirical evaluations demonstrate that average rewards improve with time. In particular, the performance reveals an important insight: when the true model is absent even from the larger model space, combinations of models could approximately fit observed trajectories in a form of equivalence. This is evident from improving rewards in both the problems for this pathological case. Consequently, OPIAM represents a significant pragmatic step toward individual online planning in multiagent settings with applications

in ad hoc cooperation and other methods that maintain models. Our future work involves exploring other ways of adapting the model set that exhibits improved accuracy and demonstrating the benefit of this approach in cooperative algorithms as well that ascribe models to others. The greater speedups in larger domains also encourage further investigation into representation and solution techniques for I-DIDs.

Acknowledgments

We gratefully acknowledge support from an ONR grant, N000141310870, and a NSF CAREER grant, IIS-0845036.

APPENDIX

Denote the maximal reward value in the utility function as R_i^{max}.

PROOF OF LEMMA 1. Since both $m^*_{j,l-1}$ and $m_{j,l-1}$ are considered as behaviorally equivalent models and are assigned with the same beliefs in the I-DID, we proceed:

$$|\rho(b_{i,l},a_i) - \dot{\rho}(b_{i,l},a_i)| = |\sum_{s,m_{j,l-1}} b_{i,l}(s,m_{j,l-1}) \sum_{a_j} R_i(s,a_i,a_j)(Pr(a_j|m^*_{j,l-1}) - Pr(a_j|\dot{m}_{j,l-1}))|$$
$$\leq \eta \sum_{s,m_{j,l-1}} b_{i,l}(s,m_{j,l-1}) \sum_{a_j} R_i^{max} \leq \eta R_i^{max}$$

□

Next, we give the proof of Lemma 2 in Section 4, which corresponds to Lemma 4 here. First, we establish Lemmas 1 and 2, which will be needed by Lemma 4.

LEMMA 2. *For any, $b_{i,l}$, a_i, o_i, and a_j, $|Pr(o_i|b_{i,l},a_i) - \dot{Pr}(o_i|b_{i,l},a_i)| \leq \eta|\Omega_j|$.*

LEMMA 3. *For any pair of beliefs, $b'_{i,l}$ and $\dot{b}'_{i,l}$, obtained by updating the same initial belief and the latter obtained by considering j's most probable models for predicting its actions,*

$$Pr(o_i|b_{i,l},a_i)|b'_{i,l} - \dot{b}'_{i,l}|_1 \leq 2\eta|\Omega_j|$$

PROOF SKETCH OF LEMMA 4. Lemmas 2 and 3 combined together result in Lemma 4 in a straightforward way. □

PROOF OF PROPOSITION 1. Note that, $V^T(m_{i,l}) = \max_{a_i \in A_i} Q^T(m_{i,l},a_i)$. We get

$$|V^T(m_{i,l}) - \dot{V}^T(m_{i,l})| \leq \max_{a_i \in A_i} |Q^T(m_{i,l},a_i) - \dot{Q}^T(m_{i,l},a_i)|$$

Let a_i be the particular action that maximizes the above difference. Construct an intermediate action-value function,

$$\ddot{Q}^T(m_{i,l},a_i) = \dot{\rho}(b_{i,l},a_i) + \sum_{o_i} \dot{Pr}(o_i|b_{i,l},a_i)V^{T-1}(\dot{m}'_{i,l})$$

Note that $\ddot{Q}^T(m_{i,l},a_i)$ differs from $\dot{Q}^T(m_{i,l},a_i)$ in using the value function, $V^{T-1}(\dot{m}'_{i,l})$, instead of $\dot{V}^{T-1}(\dot{m}'_{i,l})$. In other words, it uses the exact value function at the next time step.

Subsequently, we may rewrite the above as,

$$|Q^T(m_{i,l},a_i) - \dot{Q}^T(m_{i,l},a_i)| \leq |Q^T(m_{i,l},a_i) - \ddot{Q}^T(m_{i,l},a_i)| + |\ddot{Q}^T(m_{i,l},a_i) - \dot{Q}^T(m_{i,l},a_i)| \quad (3)$$

Focusing on the first term of the right side of (3),

$$|Q^T(m_{i,l},a_i) - \ddot{Q}^T(m_{i,l},a_i)| \leq |(\rho(b_{i,l},a_i) - \dot{\rho}(b_{i,l},a_i)| + \sum_{o_i} |Pr(o_i|b_{i,l},a_i)V^{T-1}(m'_{i,l}) \dot{Pr}(o_i|b_{i,l},a_i)V^{T-1}(\dot{m}'_{i,l})| \quad (4)$$
$$\leq \eta R_i^{max}(1 + 3(T-1)|\Omega_i||\Omega_j|)$$

Focusing on the second term in the inequality of (3),

$$|\ddot{Q}^T(m_{i,l},a_i) - \dot{Q}^T(m_{i,l},a_i)| \leq |(\dot{\rho}(b_{i,l},a_i) - \dot{\rho}(b_{i,l},a_i)| + |\sum_{o_i} \dot{Pr}(o_i|b_{i,l},a_i)(V^{T-1}(\dot{m}'_{i,l}) - \dot{V}^{T-1}(\dot{m}'_{i,l}))|$$
$$\leq \sum_{o_i} \dot{Pr}(o_i|b_{i,l},a_i)|V^{T-1}(m'_{i,l}) - \dot{V}^{T-1}(\dot{m}'_{i,l})| \quad (5)$$
$$= \mathcal{E}^{T-1} \sum_{o_i} \dot{Pr}(o_i|b_{i,l},a_i) = \mathcal{E}^{T-1}$$

Substituting (4) and (5) into the inequality of (3) we get,

$$\begin{aligned} \mathcal{E}^T &\leq \eta R_i^{max}(1 + 3(T-1)|\Omega_i||\Omega_j|) + \mathcal{E}^{T-1} \\ &\leq \eta R_i^{max}(T-1)(1 + 3(T-1)|\Omega_i||\Omega_j|) + \mathcal{E}^1 \\ &\leq \eta R_i^{max}(T-1)(1 + 3(T-1)|\Omega_i||\Omega_j|) + \eta R_i^{max} \\ &= \eta R_i^{max}((T-1)(1 + 3(T-1)|\Omega_i||\Omega_j|) + 1) \end{aligned}$$

Observe that the difference in value of agent i's policy is linearly bounded by the prediction error, η. The selection of most probable models, $\dot{m}_{j,l-1}$, reduces this error, which optimizes the performance of our algorithm. □

PROOF OF PROPOSITION 2. Let $\mathcal{H}_j^T = \xi_j^{T,(1)}, \xi_j^{T,(2)}, \ldots, \xi_j^{T,(N)}$ be the N most probable paths of agent j obtained from the interactions before the adaptation phase in OPIAM. These paths are likely to be simulations of j's true model, $m^*_{j,l-1}$. Then,

$$Pr(a_j|\mathcal{H}_j^T) \triangleq \sum\nolimits_{\xi_j^{T,(k)} \in \mathcal{H}_j^T} \omega_{\xi_j^T} Pr(a_j|\xi_j^{T,(k)})$$

We may view $Pr(a_j|\mathcal{H}_j^T)$ as the sample mean and $Pr(a_j|m^*_{j,l-1})$ as the true mean. Hoeffding's inequality [10] provides a bound on the probable rate of convergence of the sample mean to true mean:

$$Pr(|Pr(a_j^t|\mathcal{H}_j^T) - Pr(a_j^t|m^*_{j,l-1})| > \hat{\epsilon}) \leq \frac{1}{e^{2NT\hat{\epsilon}^2}}$$

$$Pr(\sum_t \sum_{a_j} |Pr(a_j^t|\mathcal{H}_j^T) - Pr(a_j^t|m^*_{j,l-1})| > |A_j|T\hat{\epsilon}) \leq |A_j|T\frac{1}{e^{2NT\hat{\epsilon}^2}}$$

$$Pr(\sum_t \sum_{a_j} |Pr(a_j^t|\mathcal{H}_j^T) - Pr(a_j^t|m^*_{j,l-1})| > \epsilon) \leq |A_j|T\frac{1}{e^{2NT(\epsilon/|A_j|T)^2}}$$

where $\epsilon = |A_j|\hat{\epsilon}$.

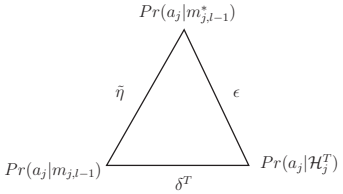

Figure 12: $\tilde{\eta}$, δ^T, and ϵ form a triangle.

Subsequently,

$$Pr(\sum_t \sum_{a_j} |Pr(a_j^t|\mathcal{H}_j^T) - Pr(a_j^t|m^*_{j,l-1})| \leq \epsilon) \geq 1 - |A_j|T\frac{1}{e^{2NT(\epsilon/|A_j|T)^2}} \quad (6)$$

Recall that, $\eta = \max_t \sum_{a_j} |Pr(a_j^t|\dot{m}_{j,l-1}) - Pr(a_j^t|m^*_{j,l-1})|$ and $\delta^T = \sum_t \sum_{a_j \in A_j} |Pr(a_j^t|\pi^T_{\dot{m}_j}) - Pr(a_j^t|\mathcal{H}_j^T))|$, where $\dot{m}_{j,l-1}$ is the most probable model selected for inclusion in $\mathcal{M}^0_{j,l-1}$ per Def. 3. Let $\tilde{\eta} = \sum_t \sum_{a_j} |Pr(a_j^t|\dot{m}_{j,l-1}) - Pr(a_j^t|m^*_{j,l-1})|$ where $\eta \leq \tilde{\eta}$.

Figure 12 shows the relationships between the three differences. Subsequently, from the law of triangle inequality, $\eta \leq \tilde{\eta} \leq (\delta^T + \epsilon)$ where upper bound ϵ obtains with probability at least $1 - \frac{|A_j|T}{e^{2NT(\epsilon/|A_j|T)^2}}$. □

REFERENCES

[1] S. Albrecht and S. Ramamoorthy. A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems. Technical report, School of Informatics, The University of Edinburgh, United Kingdom, 2013.

[2] S. V. Albrecht and S. Ramamoorthy. Ad hoc coordination in multiagent systems with applications to human-machine interaction. In *International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, pages 1415–1416, 2013.

[3] M. Chandrasekaran, P. Doshi, Y. Zeng, and Y. Chen. Team behavior in interactive dynamic influence diagrams with applications to ad hoc teams (extended abstract). In *Autonomous Agents and Multi-Agent Systems Conference (AAMAS)*, pages 1559–1560, 2014.

[4] Y. Chen, J. Hong, W. Liu, L. Godo, C. Sierra, and M. Loughlin. Incorporating PGMs into a BDI architecture. In *16th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA)*, pages 54–69, 2013.

[5] P. Doshi. Decision making in complex mulitiagent contexts: A tale of two frameworks. *AI Magazine*, 33(4):82–95, 2012.

[6] P. Doshi and P. J. Gmytrasiewicz. On the difficulty of achieving equilibrium in interactive POMDPs. In *Twenty-First Conference on Artificial Intelligence (AAAI)*, pages 1131–1136, 2006.

[7] P. Doshi and D. Perez. Generalized point based value iteration for interactive pomdps. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI*, pages 63–68, 2008.

[8] P. Doshi, Y. Zeng, and Q. Chen. Graphical models for interactive pomdps: Representations and solutions. *Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS)*, 18(3):376–416, 2009.

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

[10] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[11] R. A. Howard and J. E. Matheson. Influence diagrams. In *Readings on the Principles and Applications of Decision Analysis*, pages 721–762, 1984.

[12] E. Kalai and E. Lehrer. Rational learning leads to nash equilibrium. *Econometrica*, pages 1019–1045, 1993.

[13] J. Luo, H. Yin, B. Li, and C. Wu. Path planning for automated guided vehicles system via interactive dynamic influence diagrams with communication. In *9th IEEE International Conference on Control and Automation (ICCA)*, pages 755 –759, 2011.

[14] P. MacAlpine, K. Genter, S. Barrett, and P. Stone. The RoboCup 2013 drop-in player challenges: A testbed for ad hoc teamwork. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2014.

[15] J. Marecki, T. Gupta, P. Varakantham, M. Tambe, and M. Yokoo. Not all agents are equal: Scaling up distributed POMDPs for agent networks. In *International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 485–492, 2008.

[16] J. Pineau, G. Gordon, and S. Thrun. Anytime point-based value iteration for large pomdps. *Journal of Artificial Intelligence Research*, 27:335–380, 2006.

[17] D. Pynadath and S. Marsella. Minimal mental models. In *Twenty-Second Conference on Artificial Intelligence (AAAI)*, pages 1038–1044, Vancouver, Canada, 2007.

[18] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa. Online planning algorithms for pomdps. *Journal Artificial Intelligence Research (JAIR)*, pages 663–704, 2008.

[19] S. Seuken and S. Zilberstein. Memory bounded dynamic programming for decentralized POMDPs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2009–2015, 2007.

[20] E. Sonu and P. Doshi. Bimodal switching for online planning in multiagent settings. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 360–366, 2013.

[21] P. Stone, G. A. Kaminka, S. Kraus, J. S. Rosenschein, et al. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *AAAI*, 2010.

[22] G. Sukthankar, C. Geib, H. Bui, D. Pynadath, and R. Goldman, editors. *Plan, Activity and Intent Recognition*. Springer, 2014.

[23] J. A. Tatman and R. D. Shachter. Dynamic programming and influence diagrams. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2):365–379, 1990.

[24] F. Wu, S. Zilberstein, and X. Chen. Online planning for ad hoc autonomous agent teams. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 439–445, 2011.

[25] F. Wu, S. Zilbersteinb, and X. Chen. Online planning for multi-agent systems with bounded communication. *Artificial Intelligence*, 175(2):487–511, 2011.

[26] Y. Zeng and P. Doshi. Exploiting model equivalences for solving interactive dynamic influence diagrams. *Journal of Artificial Intelligence Research (JAIR)*, 43:211–255, 2012.

[27] Y. Zeng, P. Doshi, Y. Pan, H. Mao, M. Chandrasekaran, and J. Luo. Utilizing partial policies for identifying equivalence of behavioral models. In *Twenty-Fifth Conference on Artificial Intelligence (AAAI)*, pages 1083–1088, 2011.

[28] Y. Zeng, H. Mao, Y. Pan, and J. Luo. Improved use of partial policies for identifying behavioral equivalence. In *Autonomous Agents and Multi-Agent Systems Conference (AAMAS)*, pages 1015–1022, 2012.