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.


Copyright © 2015, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.
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, $M_{j,t-1}$, models how other agent $j$ makes its decisions simultaneously at level $t-1$. 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 $M_{j,t-1}$ and $A_{j}$. Each model, $m_{j,t-1}$, could be either a level $t-1$ 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 that 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 $i$ 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 $M_{j,t-1}$ to $M_{j,t}^{+1}$ 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 $|M_{j,t-1}| |A_{j}| |\Omega_{j}|$ models. Here, $|M_{j,t-1}|$ 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, $M_{j,0}^{1,i}$ and $M_{j,0}^{2,i}$, differ in $j$’s belief about the tiger’s location and are included in the model node $M_{j,0}^{i}$. 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 $m_{j,0}^{1,i}$ and $m_{j,0}^{2,i}$ 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 $M_{j,0}^{i}$. Figure 3: Details of the model update link where two models are expanded into four models in the new time step.

The CPT of $\text{Mod}[M_{j,0}^{i+1}]$ is shown in Fig. 4. For example, the first row of the CPT shows that $m_{j,0}^{1,i}$ is updated into the model $m_{j,0}^{2,i}$ 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 $m_{j,0}^{1,i}$, we assign a uniform distribution to indicate that $m_{j,0}^{1,i}$ does not transform into any of the new models for these actions.

Figure 4: The CPTs of the chance nodes $A_{j}^{i}$ and $\text{Mod}[M_{j,0}^{i+1}]$. 

<table>
<thead>
<tr>
<th>Decisions($A_{j}^{i}$)</th>
<th>OL: Open the left door</th>
<th>OR: Open the right door</th>
<th>L: Listen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observations($ Growlj$)</td>
<td>GL: Growl from the left door</td>
<td>GR: Growl from the right door</td>
<td></td>
</tr>
<tr>
<td>[OL, GL]</td>
<td>$m_{j,0}^{1,i}$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[OL, GR]</td>
<td>$m_{j,0}^{2,i}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[L, GL]</td>
<td>$m_{j,0}^{3,i}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[L, GR]</td>
<td>$m_{j,0}^{4,i}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[OR, *]</td>
<td>$m_{j,0}^{5,i}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>[OR, *]</td>
<td>$m_{j,0}^{6,i}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CPT of $\text{Mod}[M_{j,0}^{i+1}]$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 ($M_{j,l-1}$), 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 e-BE [28] solve I-DIDs by exploiting equivalences between models. In particular, DMU exactly solves I-DIDs while e-BE compromises the solution quality to achieve greater efficiency.

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

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, $b_{l,i}(M_{j,l-1})$, previous algorithms [26] solve $i$’s I-DID with the complete set of $j$'s models. Let $M_{j,l-1}$ denote the set of $j$'s models included in the model node of the I-DID initially. Then, in these approaches $M_{j,l-1} = M_{j,l-1}$. Subsequently, agent $i$ interacts with $j$ using the policy obtained by solving the I-DID and empirically updates $b_{l,i}(M_{j,l-1})$ accordingly. Doshi and Gmytrasiewicz show that $b_{l,i}(M_{j,l-1})$ converges to a probability distribution, denoted by $b_{l,i}(M_{j,l-1})$, 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: $M_{j,l-1} \subset M_{j,l-1}$. Agent $i$ then uses the solved policy to interact with $j$ for some time and updates $b_{l,i}(M_{j,l-1})$ using the trajectories of its actions and observations obtained from the interactions. Subsequently, OPIAM modifies the I-DID by partially replacing $M_{j,l-1}$ with models from the unexplored set ($M_{j,l-1} - M_{j,l-1}$).

We present OPIAM in Fig. 6. Given a large set of models of agent $j$, $M_{j,l-1}$, we randomly select a small set of candidate models, $\mathcal{M}_{j,l-1}$, 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 $\pi_{m,i}$ 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, $h_{i}^{T,(k)}$, is an action-observation sequence of $T$ steps: $h_{i}^{T,(k)} = \{a_{1}^{T}, o_{1}^{T+1}, \ldots, a_{T}^{T}, o_{T}^{T+1}\}$, where $o_{T}^{T+1}$ is null.

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

$$Pr(m_{j,l-1}^{T,(k)}| h_{i}^{T,(k)}) \propto \sum_{s=0}^{N} Pr(m_{j,l-1}^{T,(k)}|s) Pr(h_{i}^{T,(k)}|m_{j,l-1}^{T,(k)}$$

where $Pr(m_{j,l-1}^{T,(k)}|s)$ is the prior weight of $j$’s model given the state and $Pr(h_{i}^{T,(k)}|m_{j,l-1}^{T,(k)})$ the likelihood of $i$’s policy path. We compute this by inserting $h_{i}^{T,(k)}$ 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, $\sum_{k=1}^{N} Pr(m_{j,l-1}^{T,(k)})/N$ (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 $b_{l,i}(\mathcal{M}_{j,l-1})$. This is done by redistributing the probability mass in $i$’s belief for the selected models in $\mathcal{M}_{j,l-1}$ over all $j$’s models in $\mathcal{M}_{j,l-1}$ proportionally to their average weights obtained in the previous step. This is shown in Eq. 1:

$$b_{l,i}(m_{j,l-1}) = \frac{1}{N} \sum_{k=1}^{N} Pr(m_{j,l-1}|h_{i}^{T,(k)})$$

$$\times \sum_{m_{j,l-1} \in \mathcal{M}_{j,l-1}} b_{i,l}^{T-1}(m_{j,l-1})$$

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

1163
we focus on the challenge of selecting the most probable model of \( m \). Formally, we define the most probable path below.

**Definition 2 (Most Probable Path).** Given agent \( i \)'s model, \( m_{i,j} \), and \( i \)'s sequence of actions and observations, \( h_i^{T,(k)} \), define the most probable path for agent \( j \) as:

\[
\xi_j^{T,(k)} = \arg \max_{h_j^{T,(k)}} Pr(h_j^{T,(k)})
\]

where

\[
Pr(h_j^{T,(k)}) = \arg \max_{h_j^{T,(k)}} \prod_{t=1}^{T} Pr(a_j^t | h_i^{T,(k)}), h_j^{-1}, o_j^t) Pr(a_j^t | h_i^{T,(k)}), h_j^{-2}
\]

\[
\times Pr(a_j^t | h_i^{T,(k)})
\]

The computation factorizes the joint probability \( Pr(h_j^{T,(k)}) \) 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, \( H_j^T = A_j \times \prod_{k=1}^{T-1} (Q_j \times A_j) \) 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 \( Pr(a_j^t | h_i^{T,(k)}) \) at time \( t = 0 \), and subsequently sample the most probable observation and actions over time.

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

Consider a candidate model, \( m_{j,l-1} \in M_{j,l-1} \), whose solution is a \( T \)-step policy tree, \( \pi_j^{T,(k)} \). Let \( Pr(a_j^t | \pi_j^{T,(k)}) \) be the proportion among all actions that action \( a_j^t \), 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 \( \delta^T \) denote this measure and we define it as:

\[
\delta^T(m_{j,l-1}, H_j^T) = \sum_{t} \sum_{a_j \in A_j} |Pr(a_j^t | f_{m_{j,l-1}}) - Pr(a_j^t | h_j^T)| \quad (2)
\]

where \( Pr(a_j^t | H_j^T) = \sum_{\xi_j^{T,(k)} \in H_j^T} \omega_{\xi_j^{T,(k)}} \cdot Pr(a_j^t | \xi_j^{T,(k)}) \). Here, \( Pr(a_j^t | \xi_j^{T,(k)}) \) is 1 if action \( a_j^t \) appears in the most probable path, \( \xi_j^{T,(k)} \), at time step \( t \) and weight \( \omega_{\xi_j^{T,(k)}} \) 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, \( M_{j,l-1} \), and most probable paths, \( H_j^T \), define the most probable model, \( m_{j,l-1} = \arg \min_{m_{j,l-1} \in M_{j,l-1}} \delta^T(m_{j,l-1}, H_j^T) \)

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, \( M_{j,l-1} \), comprises of three models, \( \{m_1, m_2, m_3\} \), and another candidate model, \( m_4 \), is available to be selected for updating \( M_{j,l-1} \). Figure 7 shows a combination of policy trees (\( T = 3 \)) obtained by solving every model in \( M_{j,l-1} \) (on the left), and the policy tree of \( m_4 \) (on the right). After \( N = 6 \) interactions, \( H_j \) consists of three policy paths: \( \xi_1 = (OL, GR, L, GR, OL) \), \( \xi_2 = (L, GL, L, GR, L) \).
At time step \( t = 0 \), the model \( M_{j,t-1} \) is part of \( M_{j,t-1} \). As indicated in Def. 3, the model selected when its predicted actions, \( \Pr(a_j|m_{j,t-1}) \), are similar to those found in the most probable paths, \( \Pr(a_j|H_j^t) \), 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 \( m^*_j \) is part of \( M_{j,t-1} \). Let \( \eta \) be the worst \( L_1 \)-norm error in the prediction of \( j \)'s actions at any time step due to the model replacement, \( \eta = \max_{t \in T} \sum_{a_j \in A_j} \Pr(a_j|m_{j,t-1}) - \Pr(a_j|m^*_j) \). The expected value of agent \( i \)'s optimal policy given by the I-DID is:

\[
V^T(m_{i,t}) = \rho(b_{i,t}, a^*_i) + \sum_{s,m_{j,t-1}} \Pr(a_j|s,m_{j,t-1}) \sum_{a_j} R_i(s, a^*_i, a_j) \]

where \( \rho(b_{i,t}, a^*_i) = \sum_{s,m_{j,t-1}} b_{i,t}(s, m_{j,t-1}) \sum_{a_j} R_i(s, a^*_i, a_j) \) is \( i \)'s optimal action and \( m^*_j \) 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 \( j \)'s optimal policy when \( j \)'s predicted behavior is \( \Pr(a_j|m_{j,t-1}) \), instead of \( \Pr(a_j|m^*_j) \), as \( V^T(m_{i,t}) \) and \( \rho(b_{i,t}, a^*_i) \), respectively. Denote the maximal reward value in the reward function as \( R_i^{\text{max}} \).

Proposition 1 with help from Lemmas 1 and 2 gives the difference in the value functions due to the misprediction, \( \eta \).

**LEMA 1.** For any \( i \)'s belief, \( b_{i,t} \), and action, \( a_i \),

\[
|\rho(b_{i,t}, a_i) - \rho(b_{i,t}, a_i)| \leq \eta R_i^{\text{max}}
\]

**LEMA 2.** For any pair of updated models of agent \( i \), \( m_{i,t} \) and \( m_{i,t}^* \), where the latter is due to differing predictions of \( j \)'s actions,

\[
|\Pr(a_j|b_{i,t}, a_i) V^T(m_{i,t}) - \Pr(a_j|b_{i,t}, a_i) V^T(m_{i,t}^*)| \leq 3\eta(T - 1) R_i^{\text{max}} |\Omega_i|
\]

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 \( i \), \( m_{i,t} \), let \( V^T(m_{i,t}) \) be the expected reward from solving a \( T \)-horizon I-DID optimally, and \( V^T(m_{i,t}^*) \) 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 \( E^T = |V^T(m_{i,t}) - V^T(m_{i,t}^*)| \). Then,

\[
E^T \leq \eta R_i^{\text{max}} ((T - 1)(1 + 3(T - 1)|\Omega_i| + 1)
\]

Proposition 1 does not as yet bound the error because a non-trivial bound for \( \eta \) is not established. Proposition 2 provides the crucial missing piece probabilistically bounding \( \eta \) using \( \delta^T \) and an error term, \( \epsilon \), that is flexible.

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

Together, Propositions 1 and 2 establish a PAC bound on the worst error incurred by \( \text{OPIAM} \) 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 \( \text{OPIAM} \) is loose. For the pathological condition when \( m_{i,t}^* \notin M_{j,t-1} \), the probability is not guaranteed to improve with increasing samples unless \( b_{i,t} \) continues to satisfy the absolute continuity condition [12].

**5. EXPERIMENTAL RESULTS**

We implemented \( \text{OPIAM} \) (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
$\mathcal{M}_{j,t-1}/\mathcal{M}_{j,t-1}^0$, 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 $\epsilon$-BE, compacts the complete model space, $\mathcal{M}_{j,t-1}$; 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 ($|S|=2$, $|\mathcal{A}_i|\in\mathcal{A}_j|3$, $|\mathcal{Q}_i|\in\mathcal{Q}_j|6$, $|\Omega_i|\in\Omega_j|3$, $T=6$) and the larger multi-unmanned aerial vehicle (multiUAV) reconnaissance problem ($|S|=25$, $|\mathcal{A}_i|\in\mathcal{A}_j|5$, $|\mathcal{Q}_i|\in\mathcal{Q}_j|5$, $T=4$) [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 ($\mathcal{M}_j^0$). This allowed us to experiment with settings where the true models of others are not in $\mathcal{M}_j$. The parameter $\rho$ 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 $T$, $\epsilon$ and least probability. This resulted in $N \approx 100$ ($T=6$, $|\mathcal{A}_j|=3$, $\epsilon=0.15$ and least probability 0.9) and it varies with $T$.

We evaluate OPIAM’s performance under two conditions: (i) A typical case when the true model, $m^*_j\in\mathcal{M}_{j,t-1}$, in the multiagent tiger problem. The horizontal line denotes the average reward from the exact DMU method given a uniform prior over $\mathcal{M}_{j,t-1}$.

We evaluate OPIAM’s performance under two conditions: (i) A typical case when the true model, $m^*_j\in\mathcal{M}_{j,t-1}$, in the multiagent tiger problem. The horizontal line denotes the average reward from the exact DMU method given a uniform prior over $\mathcal{M}_{j,t-1}$.

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

The 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 $\epsilon$-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 $\epsilon$-BE). Generally fewer points for $\epsilon$-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 $\epsilon$-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 $\epsilon$-BE and ROnline. In this larger problem domain, $\epsilon$-BE carried out much less rounds of prior update in the given time period. In some cases, the interaction based on $\epsilon$-BE takes too much time to be shown. Importantly, the true model enters $\mathcal{M}_{j,t-1}$ when it’s included in the larger set. But, in its absence, the model weight
Table 1: OPIAM exhibits significant speed up compared to c-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 \( R_{\text{max}} \).

PROOF OF LEMMA 4. Lemmas 2 and 3 combined to obtain Lemma 4 in a straightforward way.

\[ |Q_T(m_{i,l,1}) - Q_T(m_{i,l,1})| \leq |\hat{\rho}(b_{i,t,1}, a_i) - \hat{\rho}(b_{i,t,1}, a_i)| + \sum_{a_i \in A_i} \text{Pr}(a_i|b_{i,t,1}, a_i)V^{T-1}(\tilde{m}_{i,l,1}) \]

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

\[ \mathcal{E}^T \leq \eta R_{\text{max}}^T (1 + 3(T - 1)\|\Omega_i\|) + \mathcal{E}^{T-1} \]

Subsequently, we may rewrite the above as.

\[ \sum_{a_i \in A_i} \text{Pr}(a_i|b_{i,t,1}, a_i)V^{T-1}(\tilde{m}_{i,l,1}) \]

Let \( Q_T(m_{i,l,1}) \) be the particular action that maximizes the above difference. Construct an intermediate action-value function, \( \tilde{Q}_T(m_{i,l,1}) \).

\[ \tilde{Q}_T(m_{i,l,1}) = \hat{\rho}(b_{i,t,1}, a_i) + \sum_{a_i \in A_i} \text{Pr}(a_i|b_{i,t,1}, a_i)V^{T-1}(\tilde{m}_{i,l,1}) \]

Subsequently, we may rewrite the above as.

\[ \tilde{Q}_T(m_{i,l,1}) - Q_T(m_{i,l,1}) \leq |\hat{\rho}(b_{i,t,1}, a_i) - \hat{\rho}(b_{i,t,1}, a_i)| + \sum_{a_i \in A_i} \text{Pr}(a_i|b_{i,t,1}, a_i)V^{T-1}(\tilde{m}_{i,l,1}) \]
REFERENCES


