Improved Approximation of Interactive Dynamic Influence Diagrams Using Discriminative Model Updates

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

Yifeng Zeng
Dept. of Computer Science
Aalborg University
DK-9220 Aalborg, Denmark
yfzeng@cs.aau.dk

ABSTRACT

Interactive dynamic influence diagrams (I-DIDs) are graphical models for sequential decision making in uncertain settings shared by other agents. Algorithms for solving I-DIDs face the challenge of an exponentially growing space of candidate models ascribed to other agents, over time. We formalize the concept of a *minimal model set*, which facilitates qualitative comparisons between different approximation techniques. We then present a new approximation technique that minimizes the space of candidate models by discriminating between model updates. We empirically demonstrate that our approach improves significantly in performance on the previous clustering based approximation technique.

Categories and Subject Descriptors

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

General Terms

Theory, Performance

Keywords

decision making, agent modeling, behaviorally equivalent

1. INTRODUCTION

Interactive dynamic influence diagrams (I-DIDs) [1] are graphical models for sequential decision making in uncertain multiagent settings. I-DIDs concisely represent the problem of how an agent should act in an uncertain environment shared with others who may act in possibly similar ways. I-DIDs may be viewed as graphical counterparts of interactive POMDPs (I-POMDPs) [3], providing a way to model and exploit the embedded structure often present in real-world decision-making situations. They generalize DIDs [11], which are graphical representations of POMDPs, to multiagent settings in the same way that I-POMDPs generalize POMDPs.

As we may expect, I-DIDs acutely suffer from both the curses of dimensionality and history [5]. This is because the state space in I-DIDs includes the models of other agents in addition to the traditional physical states. These models encompass the agents' beliefs, action and sensory capabilities, and preferences, and may themselves be formalized as I-DIDs. The nesting is terminated at the 0^{th} level where the other agents are modeled using DIDs. As the agents act, observe, and update beliefs, I-DIDs must track the evolution of the models over time. Consequently, I-DIDs not only suffer from the curse of history that afflicts the modeling agent, but more so from that exhibited by the modeled agents. The exponential growth in the number of models over time also further contributes to the dimensionality of the state space. This is complicated by the nested nature of the space.

Previous approach for approximating I-DIDs [1] focuses on reducing the dimensionality of the state space by limiting and holding constant the number of models of the other agents. Using the insight that beliefs that are spatially close are likely to be behaviorally equivalent [7], the approach clusters the models of the other agents and selects representative models from each cluster. Intuitively, a cluster contains models that are likely to be behaviorally equivalent and hence may be replaced by a subset of representative models without a significant loss in the optimality of the decision maker. However, this approach first generates all possible models before reducing the space at each time step, and utilizes an iterative and often time-consuming k-means clustering method.

In this paper, we begin by formalizing a *minimal set* of models of others, a concept previously discussed in [6]. Then, we present a new approach for approximating I-DIDs that significantly reduces the space of possible models of other agents that we need consider by discriminating between model updates. Specifically, at each time step, we select only those models for updating which will result in predictive behaviors that are distinct from others in the updated model space. In other words, models that on update would result in predictions which are identical to those of existing models are not selected for updating. For these models, we simply transfer their revised probability masses to the existing behaviorally equivalent models. Intuitively, this approach improves on the previous one because it does not generate all possible models prior to selection at each time step; rather it results in minimal sets of models.

In order to avoid updating all models, we find the regions of the belief space so that models whose beliefs fall in these regions will be behaviorally equivalent on update. Note that these regions need not be in spatial proximity. Because obtaining the exact regions is computationally intensive, we approximately obtain these regions by solving a subset of the models and utilizing their combined policies. We theoretically analyze the error introduced by this approach in the optimality of the solution. More importantly, we experimentally evaluate our approach on I-DIDs formulated for two problem domains and show approximately an order of magnitude improvement in performance in comparison to the previous clustering approach.

2. BACKGROUND: INTERACTIVE DID

We briefly describe interactive influence diagrams (I-IDs) for two-agent interactions followed by their extensions to dynamic set-

Cite as: Improved Approximation of Interactive Dynamic Influence Diagrams Using Discriminative Model Updates, Prashant Doshi and Yifeng Zeng, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. XXX-XXX.

tings, I-DIDs, and refer the reader to [1] for more details.

2.1 Syntax

In addition to the usual chance, decision, and utility nodes, I-IDs include a new type of node called the *model node* (hexagonal node, $M_{j,l-1}$, in Fig. $1(a)$). We note that the probability distribution over the chance node, S, and the model node together represents agent i's belief over its *interactive state space*. In addition to the model node, I-IDs differ from IDs by having a chance node, A_j, that represents the distribution over the other agent's actions, and a dashed link, called a *policy link*.

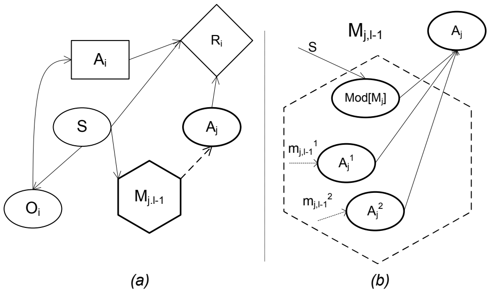


Figure 1: (a) **A generic level** $l > 0$ **I-ID for agent** i **situated with one other agent** j**. The hexagon is the model node (**$M_{j,l-1}$**) and the dashed arrow is the policy link.** (b) **Representing the model node and policy link using chance nodes and dependencies between them.**

The model node contains as its values the alternative computational models ascribed by i to the other agent. We denote the set of these models by $\mathcal{M}_{j,l-1}$. A model in the model node may itself be an I-ID or ID, and the recursion terminates when a model is an ID or a simple probability distribution over the actions. Formally, we denote a model of j as, $m_{j,l-1} = \langle b_{j,l-1}, \hat{\theta}_j \rangle$, where $b_{j,l-1}$ is the level $l-1$ belief, and $\hat{\theta}_j$ is the agent's *frame* encompassing the action, observation, and utility nodes. We observe that the model node and the dashed policy link that connects it to the chance node, A_j, could be represented as shown in Fig. $1(b)$. The decision node of each level $l-1$ I-ID is transformed into a chance node. Specifically, if OPT is the set of optimal actions obtained by solving the I-ID (or ID), then $Pr(a_j \in A_j^1) = \frac{1}{|OPT|}$ if $a_j \in OPT$, 0 otherwise. The conditional probability table (CPT) of the chance node, A_j, is a *multiplexer*, that assumes the distribution of each of the action nodes (A_j^1, A_j^2) depending on the value of $Mod[M_j]$. In other words, when $Mod[M_j]$ has the value $m_{j,l-1}^1$, the chance node A_j assumes the distribution of the node A_j^1, and A_j assumes the distribution of A_j^2 when $Mod[M_j]$ has the value $m_{j,l-1}^2$. The distribution over $Mod[M_j]$, is i's belief over j's models given the state. For more than two agents, we add a model node and a chance node representing the distribution over an agent's action linked together using a policy link, for each other agent.

I-DIDs extend I-IDs to allow sequential decision making over several time steps (see Fig. 2). In addition to the model nodes and the dashed policy link, what differentiates an I-DID from a DID is the *model update link* shown as a dotted arrow in Fig. 2. We briefly explain the semantics of the model update next.

The update of the model node over time involves two steps: First, given the models at time t, we identify the updated set of models that reside in the model node at time $t+1$. Because the agents act and receive observations, their models are updated to reflect their changed beliefs. Since the set of optimal actions for a model could include all the actions, and the agent may receive any one of $|\Omega_j|$ possible observations, the updated set at time step $t+1$ will have up to $|\mathcal{M}_{j,l-1}^t||A_j||\Omega_j|$ models. Here, $|\mathcal{M}_{j,l-1}^t|$ is the number of models at time step t, $|A_j|$ and $|\Omega_j|$ are the largest spaces of actions and observations respectively, among all the models. The CPT of $Mod[M_{j,l-1}^{t+1}]$ encodes the function, $\tau(b_{j,l-1}^t, a_j^t, o_j^{t+1}, b_{j,l-1}^{t+1})$ which is 1 if the belief $b_{j,l-1}^t$ in the model $m_{j,l-1}^t$ using the action a_j^t and observation o_j^{t+1} updates to $b_{j,l-1}^{t+1}$ in a model $m_{j,l-1}^{t+1}$; otherwise it is 0. Second, we compute the new distribution over the updated models, given the original distribution and the probability of the agent performing the action and receiving the observation that led to the updated model. The dotted model update link in the I-DID may be implemented using standard dependency links and chance nodes, as shown in Fig. 3 transforming it into a flat DID.

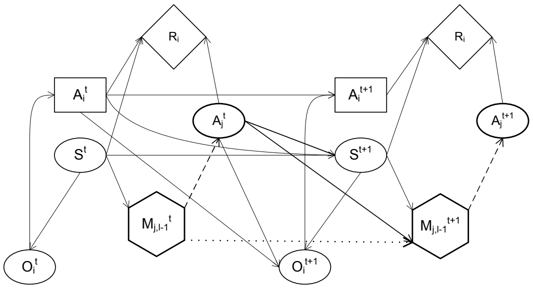


Figure 2: A generic two time-slice level l **I-DID for agent** i**.**

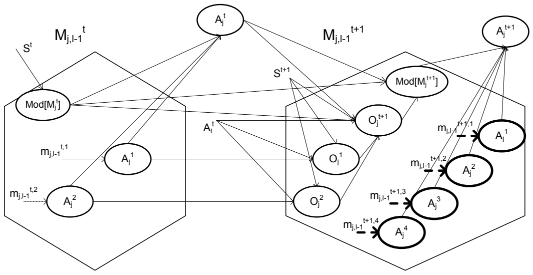


Figure 3: The semantics of the model update link. Notice the growth in the number of models at $t+1$ **shown in bold.**

2.2 Solution

The solution of an I-DID (and I-ID) proceeds in a bottom-up manner, and is implemented recursively as shown in Fig. 4. We start by solving the level 0 models, which may be traditional DIDs. Their solutions provide probability distributions which are entered in the corresponding action nodes found in the model node of the level 1 I-DID. The solution method uses the standard look-ahead technique, projecting the agent's action and observation sequences forward from the current belief state, and finding the possible beliefs that i could have in the next time step. Because agent i has a belief over j's models as well, the look-ahead includes finding out the possible models that j could have in the future. Consequently, each of j's level 0 models represented using a standard DID in the first time step must be solved to obtain its optimal set of actions. These actions are combined with the set of possible observations that j could make in that model, resulting in an updated set of candidate models (that include the updated beliefs) that could describe the behavior of j. $SE(b_j^t, a_j, o_j)$ is an abbreviation for the belief update. Beliefs over these updated set of candidate models are calculated using the standard inference methods through the dependency links between the model nodes (Fig. 3). The algorithm in Fig. 4 may be realized using the standard implementations of DIDs.

```
I-DID EXACT(level $l \geq 1$ I-DID or level 0 DID, $T$)
Expansion Phase
For $t$ from $0$ to $T-1$ do
    If $l \geq 1$ then
        Populate $M_{j,l-1}^{t+1}$
        For each $m_j^t$ in $\mathcal{M}_{j,l-1}^t$ do
            Recursively call algorithm with the $l-1$ I-DID (or DID)
            that represents $m_j^t$ and the horizon, $T-t$
            Map the decision node of the solved I-DID (or DID),
            $OPT(m_j^t)$, to the chance node $A_j^t$
            For each $a_j$ in $OPT(m_j^t)$ do
                For each $o_j$ in $O_j$ (part of $m_j^t$) do
                    Update $j$'s belief, $b_j^{t+1} \leftarrow SE(b_j^t, a_j, o_j)$
                    $m_j^{t+1} \leftarrow$ New I-DID (or DID) with $b_j^{t+1}$ as init. belief
                    $\mathcal{M}_{j,l-1}^{t+1} \overset{\cup}{\leftarrow} \{m_j^{t+1}\}$
        Add the model node, $M_{j,l-1}^{t+1}$, and the model update link
        between $M_{j,l-1}^t$ and $M_{j,l-1}^{t+1}$
    Add the chance, decision, and utility nodes for $t+1$ time slice
    and the dependency links between them
    Establish the CPTs for each chance node and utility node

Solution Phase
If $l \geq 1$ then
    Represent the model nodes and the model update link as in Fig. 3
    to obtain the DID
Apply the standard look-ahead and backup method to solve the
expanded DID (other solution approaches may also be used)
```

Figure 4: Algorithm for exactly solving a level $l \geq 1$ I-DID or level 0 DID expanded over T time steps.

3. DISCRIMINATIVE MODEL UPDATES

As we mentioned, the number of candidate models of the other agent in the model node grows exponentially over time. This exponential growth leads to a disproportionate increase in the size of the state space and the number of models that need to be solved. We begin by introducing a set of models that is *minimal* and describe a method for generating this set. A minimal set is analogous to the idea of minimal mental model space in [6]. For simplicity, we assume that models of the other agent differ only in their beliefs and that the other agent's frame is known.

3.1 Minimal Model Sets

Although the space of possible models is very large, not all models need to be considered in the model node. Models that are *behaviorally equivalent* [6, 7] – whose behavioral predictions for the other agent are identical – could be pruned and a single representative model considered. This is because the solution of the subject agent's I-DID is affected by the predicted behavior of the other agent only; thus we need not distinguish between behaviorally equivalent models.

Given the set of models of the other agent, j, in a model node, $\mathcal{M}_{j,l-1}$, we define a corresponding *minimal* set of models:

DEFINITION 1 (MINIMAL SET). *Define a minimal set of models, $\hat{\mathcal{M}}_{j,l-1}$, as the largest subset of $\mathcal{M}_{j,l-1}$, such that for each model, $m_{j,l-1} \in \hat{\mathcal{M}}_{j,l-1}$, there exists no other model, $m'_{j,l-1} \in \hat{\mathcal{M}}_{j,l-1}/m_{j,l-1}$ for which $OPT(m_{j,l-1}) = OPT(m'_{j,l-1})$, where $OPT(\cdot)$ denotes the solution of the model that forms the argument.*

We say that $\hat{\mathcal{M}}_{j,l-1}$ *minimizes* $\mathcal{M}_{j,l-1}$. As we illustrate in Fig. 5 using the well-known tiger problem [3], the set $\hat{\mathcal{M}}_{j,l-1}$ that minimizes $\mathcal{M}_{j,l-1}$ comprises of all the behaviorally *distinct* representatives of the models in $\mathcal{M}_{j,l-1}$ and only these models. Because any model from a group of behaviorally equivalent models may be selected as the representative in $\hat{\mathcal{M}}_{j,l-1}$, a minimal set corresponding to $\mathcal{M}_{j,l-1}$ is not unique, although its cardinality remains fixed.

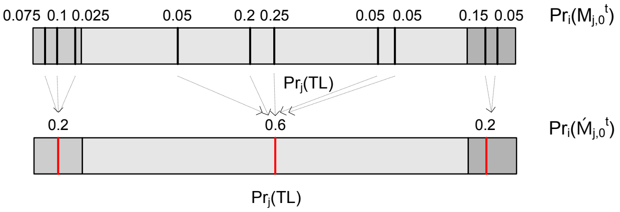


Figure 5: Illustration of a minimal set using the tiger problem. Black vertical lines denote the beliefs contained in different models of agent j included in model node, $M_{j,0}$. Decimals on top indicate i's probability distribution over j's models. In order to form a minimal set, $\hat{\mathcal{M}}_{j,0}$, we select a representative model from each behaviorally equivalent group of models (models in differently shaded regions). Agent i's distribution over the models in $\hat{\mathcal{M}}_{j,0}$ is obtained by summing the probability mass assigned to the individual models in each region. Note that $\hat{\mathcal{M}}_{j,0}$ is not unique because any model within a shaded region could be selected for inclusion in it.

Agent i's probability distribution over the minimal set, $\hat{\mathcal{M}}_{j,l-1}$, conditioned on the physical state is obtained by summing the probability mass over behaviorally equivalent models in $\mathcal{M}_{j,l-1}$ and assigning the accumulated probability to the representative model in $\hat{\mathcal{M}}_{j,l-1}$. Formally, let $\hat{m}_{j,l-1} \in \hat{\mathcal{M}}_{j,l-1}$, then:

$$\hat{b}_i(\hat{m}_{j,l-1}|s) = \sum_{m_{j,l-1} \in \mathbb{M}_{j,l-1}} b_i(m_{j,l-1}|s) \qquad (1)$$

where $\mathbb{M}_{j,l-1} \subseteq \mathcal{M}_{j,l-1}$ is the set of behaviorally equivalent models to which the representative $\hat{m}_{j,l-1}$ belongs. Thus, if $\hat{\mathcal{M}}_{j,l-1}$ minimizes $\mathcal{M}_{j,l-1}$, then Eq. 1 shows how we may obtain the probability distribution over $\hat{\mathcal{M}}_{j,l-1}$ at some time step, given i's belief distribution over models in the model node at that step (see Fig. 5).

The minimal set together with the probability distribution over it has an important property: Solution of an I-DID remains unchanged when the models in a model node and the distribution over the models are replaced by the corresponding minimal set and the distribution over it, respectively. In other words, transforming the set of models in the model node into its minimal set preserves the solution. Proposition 1 states this formally:

PROPOSITION 1. *Let $X : \Delta(\mathcal{M}_{j,l-1}) \rightarrow \Delta(\hat{\mathcal{M}}_{j,l-1})$ be a mapping defined by Eq. 1, where $\mathcal{M}_{j,l-1}$ is the space of models in a model node and $\hat{\mathcal{M}}_{j,l-1}$ minimizes it. Then, applying X preserves the solution.*

The proof of Proposition 1 is given in Appendix A. Proposition 1 allows us to show that $\hat{\mathcal{M}}_{j,l-1}$ is indeed minimal given $\mathcal{M}_{j,l-1}$ with respect to the solution of the I-DID.

COROLLARY 1. *$\hat{\mathcal{M}}_{j,l-1}$ in conjunction with X is a sufficient solution-preserving subset of models found in $\mathcal{M}_{j,l-1}$.*

Proof of this corollary follows directly from Proposition 1. Notice that the subset continues to be solution preserving when we additionally augment $\hat{\mathcal{M}}_{j,l-1}$ with models from $\mathcal{M}_{j,l-1}$.

As the number of models in the minimal set is, of course, no more than in the original set and typically much less, solution of the I-DID is often computationally less intensive with minimal sets.

3.2 Discriminating Using Policy Graph

A straightforward way of obtaining $\hat{\mathcal{M}}_{j,l-1}$ *exactly* at any time step is to first ascertain the behaviorally equivalent groups of models. This requires us to solve the I-DIDs or DIDs representing the models, select a representative model from each behaviorally equivalent group to include in $\hat{\mathcal{M}}_{j,l-1}$, and prune all others which have the same solution as the representative.

In order to avoid solving models, Doshi et al. [1] use the insight that models whose beliefs are spatially close are likely to be behaviorally equivalent. A k-means clustering approach is utilized, which clusters models based on their belief proximity and selects a pre-defined number of models from each cluster while pruning the models on the fringes of each cluster. This approach is not guaranteed to generate $\hat{\mathcal{M}}_{j,l-1}$ exactly – several behaviorally equivalent models often remain in the reduced model space. Further, the full set of models must be generated in subsequent time steps before clustering. This leaves room for further improvement.

3.2.1 *Approach*

Given the set of j's models, $\mathcal{M}_{j,l-1}$, at time t(=0), we present a technique for generating the minimal sets at subsequent time steps in the I-DID. We first observe that behaviorally distinct models at time t may result in updated models at $t+1$ that are behaviorally equivalent. Hence, our approach is to select at time step t only those models for updating which will result in predictive behaviors that are distinct from others in the updated model space at $t+1$. Models that will result in predictions on update which are identical to those of other existing models at $t+1$ are not selected for updating. Consequently, the resulting model set at $t+1$ is minimal.

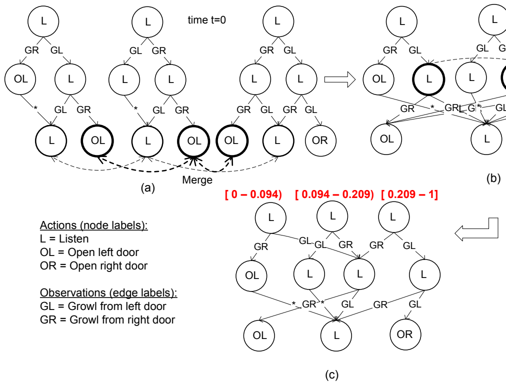


Figure 6: (a) **Example policy trees obtained by solving three models of** j **for the tiger problem setting. We may merge the three L nodes and OL nodes respectively to obtain the graph in** (b)**. Because the three policy trees of two steps rooted at L are identical, we may merge them to obtain the policy graph in** (c)**. Nodes at** $t=0$ **are annotated with ranges of** $Pr_j(TL)$**.**

We do this by solving the individual I-DIDs or DIDs in $\mathcal{M}^t_{j,l-1}$. Solutions to DIDs or I-DIDs are policy trees, which may be merged bottom up to obtain a *policy graph*, as we demonstrate in Fig. 6. The following proposition gives the complexity of merging the policy trees to obtain the policy graph.

PROPOSITION 2 (COMPLEXITY OF TREE MERGE). *The worst-case complexity of the procedure for merging policy trees to form a policy graph is* $\mathcal{O}((|\Omega_j|^{T-1})^{|\hat{\mathcal{M}}_j|})$*, where* T *is the horizon.*

PROOF. Complexity of the policy tree merge procedure is proportional to the number of comparisons that are made between parts of policy trees to ascertain their similarity. As the procedure follows a bottom-up approach, the maximum number of comparisons are made between leaf nodes and the worst case occurs when none of the leaf nodes of the different policy trees can be merged. Note that this precludes the merger of upper parts of the policy trees as well. Each policy tree may contain up to $|\Omega_j|^{T-1}$ leaf nodes, where T is the horizon. The case when none of the leaf nodes merge must occur when the models are behaviorally distinct – they form a minimal set, $\hat{\mathcal{M}}_j$. Hence, at most $\mathcal{O}((|\Omega_j|^{T-1})^{|\hat{\mathcal{M}}_j|})$ comparisons are made. ∎

Each node in the policy graph represents an action to be performed by the agent and edges represent the agent's observations. As is common with policy graphs in POMDPs, we associate with each node at time $t=0$, a range of beliefs for which the corresponding action is optimal (see Fig. 6(c)). This range may be obtained by computing the value of executing the policy tree rooted at each node at $t=0$ and starting from each physical state. This results in a vector of values for each policy tree, typically called the α-vector. Intersecting the α-vectors and projecting the intersections on the belief simplex provides us with the boundaries of the needed belief ranges.

We utilize the policy graph to discriminate between model updates. For clarity, we formally define a policy graph next.

DEFINITION 2 (POLICY GRAPH). *Define a policy graph as:*

$$PG = \langle \mathcal{V}, \mathcal{E}, \mathcal{L}_v, \mathcal{L}_e \rangle$$

where $\mathcal{V}$ *is the set of vertices (nodes);* $\mathcal{E}$ *is the set of ordered pairs of vertices (edges);* $\mathcal{L}_v : \mathcal{V} \rightarrow A$ *assigns to each vertex an action from the set of actions,* A *(node label); and* $\mathcal{L}_e : \mathcal{E} \rightarrow \Omega$ *assigns to each edge an observation from the set of observations,* Ω *(edge label).* $\mathcal{L}_e$ *observes the property that no two edges whose first elements are identical (begin at the same vertex) are assigned the same observation.*

Notice that a policy graph augments a regular graph with meaningful node and edge labels. For a policy graph, PG, we also define the transition function, $\mathcal{T}_p : \mathcal{V} \times \Omega \rightarrow \mathcal{V}$. $\mathcal{T}_p(v,o)$ returns the vertex, v', such that $\{v, v'\} \in \mathcal{E}$ and $\mathcal{L}_e(\{v,v'\}) = o$.

Our simple insight is that $\mathcal{T}_p(v,o)$ *is the root node of a policy tree that represents the predictive behavior for the model updated using the action* $\mathcal{L}_v(v)$ *and observation* o*.* As we iterate over j's models in the model node at time t in the expansion phase while solving the I-DID, we utilize $\mathcal{T}_p$ in deciding whether to update a model, $m_{j,l-1} \in \mathcal{M}^t_{j,l-1}$. We first combine the policy trees obtained by solving the models in node $M^t_{j,l-1}$ to obtain the policy graph, PG, as shown in Fig. 6. Let v be the vertex in PG whose action label, $\mathcal{L}_v(v)$, represents the rational action for $m_{j,l-1}$. We can ascertain this by simply checking whether the belief in $m_{j,l-1}$ falls within the belief range associated with the node. For every observation $o \in \mathcal{L}_e(\{v,\cdot\})$, we update the model, $m_{j,l-1}$, using action $\mathcal{L}_v(v)$ and observation o, if $v' = \mathcal{T}_p(v,o)$ has not been generated previously for this or any other model. We illustrate below:

EXAMPLE 1 (MODEL UPDATE). *Consider the level 0 models of* j *in the model node at time* t, $\mathcal{M}^t_{j,0} = \{\langle 0.01, \hat{\theta}_j\rangle, \langle 0.5, \hat{\theta}_j\rangle, \langle 0.05, \hat{\theta}_j\rangle\}$*, for the multiagent tiger problem. Recall that in a model of* j*, such as* $\langle 0.01, \hat{\theta}_j\rangle$*, 0.01 is* j*'s belief and* $\hat{\theta}_j$ *is its frame. From the PG in Fig. 6(*c*), the leftmost node prescribing the action* L *is optimal for the first and third models, while the rightmost node also prescribing* L *is optimal for the second model. Beginning with model,* $\langle 0.01, \hat{\theta}_j\rangle$*,* $\mathcal{T}_p(v, GL) = v_1$ *(where* $\mathcal{L}_v(v_1) = L$*) and* $\mathcal{T}_p(v, GR) = v_2$ *(*$\mathcal{L}_v(v_2) = OL$*). Since this is the first model we* *consider, it will be updated using L and both observations resulting in two models in $\mathcal{M}_{j,0}^{t+1}$. For the model, $\langle 0.5, \hat{\theta}_j \rangle$, if v' is the optimal node ($\mathcal{L}_v(v') = L$), $\mathcal{T}_p(v', GR) = v_1$, which has been encountered previously. Hence, the model will not be updated using L and GR, although it will be updated using L, GL.*

Intuitively, for a model, $m_{j,l-1}$, if node $v' = \mathcal{T}_p(v, o)$ has been obtained previously for this or some other model and action-observation combination, then the update of $m_{j,l-1}$ will be behaviorally equivalent to the previously updated model (both will have policy trees rooted at v'). Hence, $m_{j,l-1}$ need not be updated using the observation o. Because we do not permit updates that will lead to behaviorally equivalent models, the set of models obtained at $t + 1$ is minimal. Applying this process analogously to models in the following time steps will lead to minimal sets at all subsequent steps and nesting levels.

3.2.2 *Approximation*

We may gain further efficiency by avoiding the solution of all models in the model node at the initial time step. A simple way of doing this is to randomly select K models of j, such that $K \ll |\mathcal{M}_{j,l-1}^0|$. Solution of the models will result in K policy trees, which could be combined as shown in Fig. 6 to form a policy graph. This policy graph is utilized to discriminate between the model updates. Notice that the approach becomes exact if the optimal solution of each model in $\mathcal{M}_{j,l-1}^0$ is identical to that of one of the K models. Because the K models are selected randomly, this assumption is implausible and the approach is likely to result in a substantial loss of optimality.

We propose a simple refinement that mitigates the loss. Recall that models whose beliefs are spatially close are likely to be behaviorally equivalent [7]. Each of the remaining $|\mathcal{M}_{j,l-1}^0| - K$ models whose belief is not within $\epsilon \geq 0$ of the belief of any of the K models will also be solved. This additional step makes it more likely that all the behaviorally distinct solutions will be generated and included in forming the policy graph. If $\epsilon = 0$, all models in the model node will be solved, while increasing ϵ reduces the number of solved models beyond K. One measure of distance between belief points is the Euclidean distance, though other metrics such as the L1 may also be used.

3.3 Transfer of Probability Mass

Notice that a consequence of not updating models using some action-observation combination is that the probability mass that would have been assigned to the updated model in the model node at $t + 1$ is lost. Disregarding this probability mass may introduce error in the optimality of the solution.

We did not perform the update because a model that is behaviorally equivalent to the updated model already exists in the model node at time $t + 1$. *We could avoid the error by transferring the probability mass that would have been assigned to the updated model on to the behaviorally equivalent model.*

As we mentioned previously, the node $Mod[M_{j,l-1}^{t+1}]$ in the model node $M_{j,l-1}^{t+1}$, has as its values the different models ascribed to agent j at time $t + 1$. The CPT of $Mod[M_{j,l-1}^{t+1}]$ implements the function $\tau(b_{j,l-1}^t, a_j^t, o_j^{t+1}, b_{j,l-1}^{t+1})$, which is 1 if $b_{j,l-1}^t$ in the model $m_{j,l-1}^t$ updates to $b_{j,l-1}^{t+1}$ in model $m_{j,l-1}^{t+1}$ using the action-observation combination, otherwise it is 0. Let $m_{j,l-1}^{t+1'} = \langle b_{j,l-1}^{t+1'}, \hat{\theta}_j \rangle$ be the model that is behaviorally equivalent to $m_{j,l-1}^{t+1}$. In order to transfer the probability mass to this model if the update is pruned, we modify the CPT of $Mod[M_{j,l-1}^{t+1}]$ to indicate that $m_{j,l-1}^{t+1'}$ is the model that results from updating $b_{j,l-1}^t$ with action, a_j^t and observation o_j^{t+1}. This has the desired effect of transferring the probability that would have been assigned to the updated model (Fig. 3) on to $m_{j,l-1}^{t+1'}$ in the model node at time $t + 1$.

4. ALGORITHM

We present the algorithm for solving a level $l \geq 1$ I-DID approximately (as well as a level 0 DID) in Fig. 7. The algorithm differs from the exact approach (Fig. 4) in the presence of an initial approximation step and in the expansion phase. In addition to a two time-slice level l I-DID and horizon T, the algorithm takes as input the number of random models to be solved initially, K, and the distance, ϵ. Following Section 3.2, we begin by randomly selecting K models to solve (lines 2-5). For each of the remaining models, we identify one of the K solved model whose belief is spatially the closest (ties broken randomly). If the proximity is within ϵ, the model is not solved – instead, the previously computed solution is assigned to the corresponding action node of the model in the model node, $M_{j,l-1}^0$ (lines 6-12). Subsequently, all models in the model node are associated with their respective solutions (policy trees), which are merged to obtain the policy graph (line 13).

In order to populate the model node of the next time step, we identify the node v in PG that represents the optimal action for a model at time t. The model is updated using the optimal action a_j ($= \mathcal{L}_v(v)$) and each observation o_j only if the node, $v' = \mathcal{T}_p(v, o_j)$ has not been encountered in previous updates (lines 16-22). Given a policy graph, evaluating $\mathcal{T}_p(v, o_j)$ is a constant time operation. Otherwise, as mentioned in Section 3.3, we modify the CPT of node, $Mod[M_{j,l-1}^{t+1}]$, to transfer the probability mass to the behaviorally equivalent model (line 24). Consequently, model nodes at subsequent time steps in the expanded I-DID are likely populated with minimal sets. Given the expanded I-DID, its solution may proceed analogously to the exact approach.

5. COMPUTATIONAL SAVINGS AND ERROR BOUND

The primary complexity of solving I-DIDs is due to the large number of models that must be solved over T time steps. At some time step t, there could be $|\mathcal{M}_j^0|(|A_j||\Omega_j|)^t$ many models of the other agent j, where $|\mathcal{M}_j^0|$ is the number of models considered initially. The nested modeling further contributes to the complexity since solutions of each model at level $l - 1$ requires solving the lower level $l - 2$ models, and so on recursively up to level 0. In an N+1 agent setting, if the number of models considered at each level for an agent is bound by $|\mathcal{M}|$, then solving an I-DID at level l requires the solutions of $\mathcal{O}((N|\mathcal{M}|)^l)$ many models. Discriminating between model updates reduces the number of agent models at each level to at most the size of the minimal set, $|\hat{\mathcal{M}}^t|$, while solving at least K models initially and incurring the worst-case complexity of $\mathcal{O}((|\Omega|^{T-1})^{|\hat{\mathcal{M}}|})$ in forming the policy graph (Proposition 2). Consequently, we need to solve at most $\mathcal{O}((N|\hat{\mathcal{M}}^*|)^l)$ number of models at each non-initial time step, where $\hat{\mathcal{M}}^*$ is the largest of the minimal sets, in comparison to $\mathcal{O}((N|\mathcal{M}|)^l)$. Here $\mathcal{M}$ grows exponentially over time. In general, $|\hat{\mathcal{M}}| \ll |\mathcal{M}|$, resulting in a substantial reduction in the computation. Additionally, a reduction in the number of models in the model node also reduces the size of the interactive state space, which makes solving the upper-level I-DID more efficient.

If we choose to solve all models in the initial model node, $M_{j,l-1}^0$, in order to form the policy graph, all subsequent sets of models will indeed be minimal. Consequently, there is no loss in the optimality of the solution of agent i's level l I-DID.

```
I-DID APPROX(level $l \geq 1$ I-DID or level 0 DID, $T$, $K$, $\epsilon$)
If $l \geq 1$ then
    Selectively solve $\mathcal{M}^0_{j,l-1}$
    Randomly select $K$ models from $\mathcal{M}^0_{j,l-1}$
    For each $m^k_j$ in the $K$ models do
        Recursively call algorithm with the $l-1$ I-DID (or DID)
        that represents $m^k_j$, the horizon $T$, and $K$, $\epsilon$
        Map the decision node of the solved I-DID (or DID),
        $OPT(m^k_j)$, to the chance node $A^k_j$
    For each $m^{\bar{k}}_j$ in the $|\mathcal{M}^0_{j,l-1}| - K$ models do
        Find model among $K$ whose belief, $b^k_j$, is closest to $b^{\bar{k}}_j$ in $m^{\bar{k}}_j$
        If $||b^k_j - b^{\bar{k}}_j||_1 \leq \epsilon$ then
            Map the decision node, $OPT(m^k_j)$, to the chance node, $A^{\bar{k}}_j$
        else
            Recursively call algorithm with the $l-1$ I-DID (or DID)
            that represents $m^{\bar{k}}_j$, the horizon $T$, $K$, and $\epsilon$
            Map the decision node of the solved I-DID (or DID),
            $OPT(m^{\bar{k}}_j)$, to the chance node $A^{\bar{k}}_j$
    Merge the solutions (policy trees) of all models bottom up
    to obtain the policy graph, $PG$
Expansion Phase
For $t$ from 0 to $T-1$ do
    If $l \geq 1$ then
        Populate $M^{t+1}_{j,l-1}$ minimally
        For each $m^t_j$ in $\mathcal{M}^t_{j,l-1}$ do
            For each $a_j$ in $OPT(m^t_j)$ do
                For each $o_j$ in $\Omega_j$ (part of $m^t_j$) do
                    If $\mathcal{T}_p(v, o_j)$ not been encountered previously then
                        Update $j$'s belief, $b^{t+1}_j \leftarrow SE(b^t_j, a_j, o_j)$
                        $m^{t+1}_j \leftarrow$ New I-DID (or DID) with $b^{t+1}_j$ as belief
                        $\mathcal{M}^{t+1}_{j,l-1} \overset{\cup}{\leftarrow} \{m^{t+1}_j\}$
                    else
                        Update CPT of $Mod[M^{t+1}_{j,l-1}]$ s. t. row $m^t_j$, $a_j$, $o_j$
                        has a 1 in column of behaviorally equivalent model
        Add the model node, $M^{t+1}_{j,l-1}$, and the model update link
        between $M^t_{j,l-1}$ and $M^{t+1}_{j,l-1}$
    Add the chance, decision, and utility nodes for $t+1$ time slice
    and the dependency links between them
    Establish the CPTs for each chance node and utility node

The solution phase proceeds analogously as in Fig. 4
```

Figure 7: Algorithm for approximately solving a $l \geq 1$ I-DID expanded over T steps using discriminative model updates.

For the case where we select $K < |\mathcal{M}^0_{j,l-1}|$ models to solve, if ϵ is infinitesimally small, we will eventually solve all models resulting in no error. With increasing values of ϵ, larger numbers of models remain unsolved and could be erroneously associated with existing solutions. In the worst case, some of these models may be behaviorally distinct from all of the K solved models. Therefore, the policy graph is a subgraph of the one in the exact case, and leads to sets of models that are subsets of the minimal sets. Additionally, lower level models are solved approximately as well. Let $m_{j,l-1}$ be the model associated with a solved model, $m'_{j,l-1}$, resulting in the worst error. Let α be the exact policy tree obtained by solving $m_{j,l-1}$ optimally and α' be the policy tree for $m'_{j,l-1}$ obtained in Fig. 7. As $m'_{j,l-1}$ is itself solved approximately, let α'' be the exact policy tree that is optimal for $m'_{j,l-1}$. If $b_{j,l-1}$ is the belief in $m_{j,l-1}$ and $b'_{j,l-1}$ in $m'_{j,l-1}$, then the error is:

$$\begin{aligned} E &= |\alpha \cdot b_{j,l-1} - \alpha' \cdot b_{j,l-1}| \\ &= |\alpha \cdot b_{j,l-1} - \alpha' \cdot b_{j,l-1} + (\alpha'' \cdot b_{j,l-1} - \alpha'' \cdot b_{j,l-1})| \\ &\leq |(\alpha \cdot b_{j,l-1} - \alpha'' \cdot b_{j,l-1})| + |(\alpha'' \cdot b_{j,l-1} - \alpha' \cdot b_{j,l-1})| \end{aligned} \quad (2)$$

For the first term, $|\alpha \cdot b_{j,l-1} - \alpha'' \cdot b_{j,l-1}|$, which we denote by ρ, the error is only due to associating $m_{j,l-1}$ with $m'_{j,l-1}$ – both are solved exactly. We analyze this error below:

$$\begin{aligned} \rho &= |\alpha \cdot b_{j,l-1} - \alpha'' \cdot b_{j,l-1}| \\ &= |\alpha \cdot b_{j,l-1} - \alpha'' \cdot b'_{j,l-1} + \alpha'' \cdot b'_{j,l-1} - \alpha'' \cdot b_{j,l-1}| \\ &\leq |\alpha \cdot b_{j,l-1} - \alpha \cdot b'_{j,l-1} + \alpha'' \cdot b'_{j,l-1} - \alpha'' \cdot b_{j,l-1}| \; (\alpha'' \cdot b'_{j,l-1} \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \geq \alpha \cdot b'_{j,l-1}) \\ &= |\alpha \cdot (b_{j,l-1} - b'_{j,l-1}) - \alpha'' \cdot (b_{j,l-1} - b'_{j,l-1})| \\ &= |(\alpha - \alpha'') \cdot (b_{j,l-1} - b'_{j,l-1}) \\ &\leq ||\alpha - \alpha''||_\infty \times ||b_{j,l-1} - b'_{j,l-1}||_1 \qquad \text{(Hölder's inequality)} \\ &\leq (R^{max}_j - R^{min}_j)T \times \epsilon \end{aligned} \quad (3)$$

In subsequent time steps, because the sets of models could be subsets of the minimal sets, the updated probabilities could be transferred to incorrect models. In the worst case, the error incurred is bounded analogously to Eq. 3. Hence, the cumulative error in j's behavior over T steps is $T \times \rho$, which is similar to that in [1]:

$$\rho^T \leq (R^{max}_j - R^{min}_j)T^2\epsilon$$

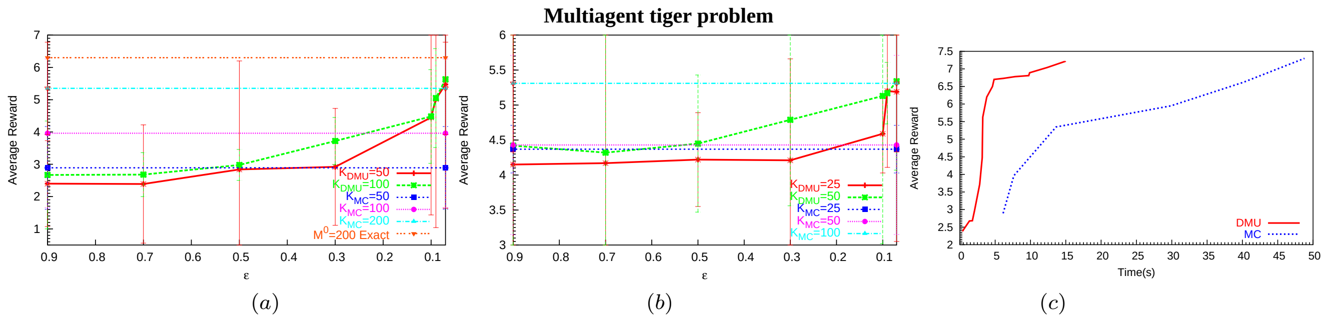
The second term, $|(\alpha'' \cdot b_{j,l-1} - \alpha' \cdot b_{j,l-1})|$, in Eq. 2 represents the error due to the approximate solutions of models further down in level. Since j's behavior depends, in part, on the actions of the other (and not the value of its solution), even a slight deviation by j from the exact prediction could lead to j's behavior that is worst in value. Hence, it seems difficult to derive bounds for the second term that are tighter than the usual, $(R^{max}_j - R^{min}_j)T$.

In summary, for any K, error in j's predicted behavior due to mapping models within ϵ is bounded, but we are unable to usefully bound the error due to approximately solving lower level models.

6. EXPERIMENTAL RESULTS

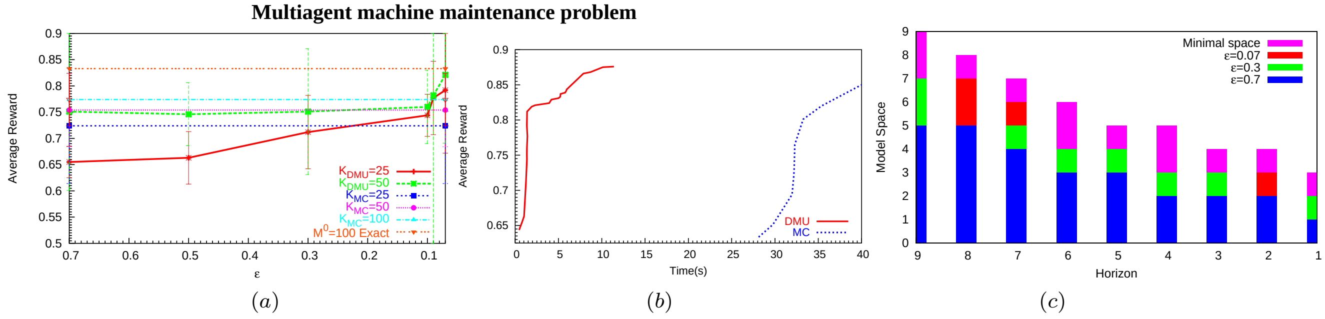
We implemented the algorithm in Fig. 7 (utilizing Hugin Expert) and demonstrate the empirical performance of discriminative model updates (DMU) on level 1 I-DIDs for two well-studied problem domains: the multiagent tiger [3] (this formulation is different from the one in [8] having more observations) and a multiagent version of the machine maintenance problem [9]. We also compare the performance with an implementation of model clustering (MC) [1], previously proposed to approximate I-DIDs. In particular, we show that the quality of the policies generated by discriminating between model updates while solving I-DIDs approaches that of the exact policy as K (which we now refer to as K_{DMU} to distinguish it from the K_{MC} in MC) is increased and ϵ decreased. As there are infinitely many computable models, we obtain the exact policy by *exactly* solving the I-DID given an initial finite set of M^0 models of the other agent. In addition, we show that DMU performs significantly (sometimes an order of magnitude) better than MC by comparing the time taken in achieving a level of expected reward. As we illustrate, this could be attributed to the low numbers of models retained by DMU, which approaches $|\hat{\mathcal{M}}^t_{j,0}|$.

In Figs. $8(a, b)$, we show the average rewards gathered by executing the policy trees obtained from approximately solving the level 1 I-DIDs for the multiagent tiger problem. Each data point is the average of 50 runs of executing the policies, where the true model of the other agent, j, is randomly picked according to i's belief distribution over j's models. Each plot is for a particular M^0, where M^0 denotes the *total* number of candidate models ascribed to j initially. For a given K_{DMU}, the policies improve and converge toward the exact as we reduce distance, ϵ. Increasing K_{DMU} lifts the average rewards. Notice that DMU significantly improves on the average reward of MC as we reduce ϵ, for $K_{DMU} = K_{MC}$. This behavior remains true for the multiagent machine maintenance



(a) (b) (c)

Figure 8: Performance profiles for the multiagent tiger problem generated by executing policies obtained using DMU on an I-DID of (a) horizon T=4; M^0=200, and (b) T=8; M^0=100. As K_{DMU} increases and ϵ reduces, the performance approaches that of the exact for given M^0. We compare with MC for varying K_{MC} as well. Vertical bars represent the standard deviations. (c) Notice that an I-DID solved using DMU requires approximately an order of magnitude less time as the MC to produce comparable solutions.



(a) (b) (c)

Figure 9: Performance profiles for multiagent MM problem. (a) T=4; M^0=100. As K_{DMU} increases and ϵ decreases, the performance approaches that of the exact for given M^0. (b) I-DID solved using DMU generates comparable average rewards in time that is approximately an order of magnitude less than used by MC. (c) Number of models generated by DMU in model node for different horizons in a T=10 I-DID for the multiagent tiger problem (M^0 = 200). As ϵ reduces, the model space approaches the minimal set.

problem as well (see Fig. 9(a)). Importantly, DMU results in solutions comparable to those using MC but, for many cases, in an order of magnitude less time (Figs. 8(c) and 9(b)). The time consumed is a function of K_{DMU}, ϵ and the horizon, which are varied.

Utilizing DMU while solving I-DIDs exhibits improved efficiency because it maintains few models in the model node at each time step. As we show in Fig. 9(c), the number of models in a model node is very close to the minimal model set for low values of ϵ. This is in contrast to MC, which, although less than M^0, still keeps a relatively high number of models to obtain comparable solution quality. Many of these models are behaviorally equivalent and could have been pruned out. We obtain slightly less models than the minimal set for low ϵ because the lower level DIDs are also solved approximately. The minimal sets were computed using a linear program analogous to the one in [1] for finding sensitivity points.

Level 1	T	Time (s)	
		DMU	MC
Tiger	6	2.53	19.86
	10	92.33	*
	17	488.12	*
MM	4	0.578	29.77
	10	95.31	*
	15	823.42	*

Table 1: DMU scales significantly better than MC to larger horizons. All experiments are run on a WinXP platform with a dual processor Xeon 2.0GHz with 2GB memory.

Finally, as we show in Table 1 we were able to solve I-DIDs over more than 15 horizons using DMU (M^0=25), improving significantly over the previous approach which could comparably solve only up to 6 horizons.

7. RELATED WORK

Suryadi and Gmytrasiewicz [10] in an early piece of related work, proposed modeling other agents using IDs. The approach proposed ways to modify the IDs to better reflect the observed behavior. However, unlike I-DIDs, other agents did not model the original agent and the distribution over the models was not updated based on the actions and observations.

I-DIDs contribute to a growing line of work that includes multiagent influence diagrams (MAIDs) [4], and more recently, networks of influence diagrams (NIDs) [2]. These formalisms seek to explicitly and transparently model the structure that is often present in real-world problems by decomposing the situation into chance and decision variables, and the dependencies between the variables. MAIDs objectively analyze the game, efficiently computing the Nash equilibrium profile by exploiting the independence structure. NIDs extend MAIDs to include agents' uncertainty over the game being played and over models of the other agents.

Both MAIDs and NIDs provide an analysis of the game from an external viewpoint, and adopt Nash equilibrium as the solution concept. However, equilibrium is not unique – there could be many joint solutions in equilibrium with no clear way to choose between them – and incomplete – the solution does not prescribe a policy when the policy followed by the other agent is not part of the equilibrium. Specifically, MAIDs do not allow us to define a distribution over non-equilibrium behaviors of other agents. Furthermore, their applicability is limited to static single play games. Interactions are more complex when they are extended over time, where predictions about others' future actions must be made using models that change as the agents act and observe. I-DIDs seek to address this gap by offering an intuitive way to extend sequential decision making as formalized by DIDs to multiagent settings. They allow the explicit representation of other agents' models as the values of a special *model node*. Other agents' models and the original agent's beliefs over these models are then updated over time.

As we mentioned, a dominating cause of the complexity of I-DIDs is the exponential growth in the candidate models over time. Using the insight that models whose beliefs are spatially close are likely to be behaviorally equivalent, Doshi et al. [1] utilized a k-means approach to cluster models together and select K models closest to the means in the model node at each time step. While this approach requires all models to be expanded before clustering is applied, in this paper we preemptively avoid expanding models that will turn out to be behaviorally equivalent to others.

Minimal sets of models were previously discussed by Pynadath and Marsella in [6], which used the concept of behavioral equivalence, introduced earlier in [7], to form the space. In addition to a formal treatment, we contextualize minimal sets within the framework of I-DIDs and utilize them to compare across approximations.

8. DISCUSSION

I-DIDs provide a general and graphical formalism for sequential decision making in the presence of other agents. The increased complexity of I-DIDs is predominantly due to the exponential growth in the number of candidate models of others, over time. These models may themselves be represented as I-DIDs or DIDs. Many of these models may be behaviorally equivalent or may become equivalent on update. We introduced the concept of a minimal model set that may be used to qualitatively compare between approximation techniques that reduce the space of models. One such approach is to discriminatively update models only if the resulting models are not behaviorally equivalent to previously updated ones. We showed an efficient way to gauge whether a model should be updated. The empirical performance demonstrates the computational savings provided by this approach and its significant improvement over the previous approximation technique. Although we focused on level 1 I-DIDs, we expect similar results as we evaluate for deeper levels of strategic nesting of the models.

Acknowledgments

This research is supported in part by grant #FA9550-08-1-0429 from AFOSR to Prashant Doshi.

9. REFERENCES

[1] P. Doshi, Y. Zeng, and Q. Chen. Graphical models for interactive pomdps: representations and solutions. *JAAMAS*, DOI:10.1007/s10458-008-9064-7, 2008.

[2] Y. Gal and A. Pfeffer. A language for modeling agent's decision-making processes in games. In *AAMAS*, pages 265–272, 2003.

[3] P. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multiagent settings. *JAIR*, 24:49–79, 2005.

[4] D. Koller and B. Milch. Multi-agent IDs for representing and solving games. In *IJCAI*, pages 1027–1034, 2001.

[5] J. Pineau, G. Gordon, and S. Thrun. Anytime point-based value iteration for large pomdps. *JAIR*, 27:335–380, 2006.

[6] D. Pynadath and S. Marsella. Minimal mental models. In *AAAI*, pages 1038–1044, 2007.

[7] B. Rathnas., P. Doshi, and P. J. Gmytrasiewicz. Exact solutions to interactive pomdps using behavioral equivalence. In *AAMAS*, pages 1025–1032, 2006.

[8] S. Seuken and S. Zilberstein. Improved memory bounded dynamic programming for decentralized pomdps. In *UAI*, pages 2009–2015, 2007.

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

[10] D. Suryadi and P. Gmytrasiewicz. Learning models of other agents using IDs. In *UM*, pages 223–232, 1999.

[11] J. A. Tatman and R. D. Shachter. Dynamic programming and influence diagrams. *IEEE Trans. SMC*, 20(2):365–379, 1990.

APPENDIX

A. PROOF OF PROPOSITION 1

PROOF. We prove by induction on the horizon. Let $\{\mathbb{M}^1_{j,l-1}, \ldots, \mathbb{M}^q_{j,l-1}\}$ be the collection of behaviorally equivalent sets of models in $\mathcal{M}_{j,l-1}$. We aim to show that the value of each of i's actions in the decision nodes at each time step remains unchanged on application of the transformation, X. This implies that the solution of the I-DID is preserved. Let $Q^n(b_{i,l}, a_i)$ give the action value at horizon n. It's computation in the I-DID could be modeled using the standard dynamic programming approach. Let $ER_i(s, m_{j,l-1}, a_i)$ be the expected immediate reward for agent i averaged over j's predicted actions. Then, $\forall_{m^q_{j,l-1} \in \mathbb{M}^q_{j,l-1}}$ $ER_i(s, m^q_{j,l-1}, a_i) = \sum_{a_j} R_i(s, a_i, a_j)\ Pr(a_j|m^q_{j,l-1}) = R_i(s, a_i, a^q_j)$, because a^q_j is optimal for all $m^q_{j,l-1} \in \mathbb{M}^q_{j,l-1}$.

Basis step: $Q^1(b_{i,l}, a_i) = \sum_{s,m_{j,l-1}} b_{i,l}(s, m_{j,l-1}) ER_i(s, m_{j,l-1}, a_i) = \sum_{s,q} b_{i,l}(s) \sum_{m^q_{j,l-1} \in \mathbb{M}^q_{j,l-1}} b_{i,l}(m^q_{j,l-1}|s) R_i(s, a_i, a^q_j)$

(a^q_j is optimal for all behaviorally equivalent models in $\mathbb{M}^q_{j,l-1}$)

$= \sum_{s,q} b_{i,l}(s) R_i(s, a_i, a^q_j) \sum_{m^q_{j,l-1} \in \mathbb{M}^q_{j,l-1}} b_{i,l}(m^q_{j,l-1}|s)$

$= \sum_{s,q} b_{i,l}(s) R_i(s, a_i, a^q_j) \hat{b}_{i,l}(\hat{m}^q_{j,l-1}|s)$ (from Eq. 1)

$= \sum_{s,q} \hat{b}_{i,l}(s, \hat{m}^q_{j,l-1}) ER_i(s, \hat{m}^q_{j,l-1}, a_i)$ (a^q_j is optimal for $\hat{m}^q_{j,l-1}$)

$= \hat{Q}^1_i(\hat{b}_{i,l}, a_i)$

Inductive hypothesis: Let, $\forall_{a_i, b_{i,l}}\ Q^n(b_{i,l}, a_i) = \hat{Q}^n(\hat{b}_{i,l}, a_i)$, where $\hat{b}_{i,l}$ relates to $b_{i,l}$ using Eq. 1. Therefore, $U^n(b_{i,l}) = \hat{U}^n(\hat{b}_{i,l})$ where $U^n(b_{i,l})$ is the expected utility of $b_{i,l}$ for horizon n.

Inductive proof: $Q^{n+1}(b_{i,l}, a_i) = \hat{Q}^1(\hat{b}_{i,l}, a_i) + \sum_{o_i, s, m_{j,l-1}, a_j} Pr(o_i|s, a_i, a_j) Pr(a_j|m_{j,l-1}) b_{i,l}(s, m_{j,l-1}) U^n(b'_{i,l})$ (basis step)

$= \hat{Q}^1(\hat{b}_{i,l}, a_i) + \sum_{o_i, s, q} Pr(o_i|s, a_i, a^q_j)\ b_{i,l}(s) \sum_{m^q_{j,l-1} \in \mathbb{M}^q_{j,l-1}} b_{i,l}(m^q_{j,l-1}|s)\ U^n(b'_{i,l})$ (a^q_j is optimal for model in $\mathbb{M}^q_{j,l-1}$)

$= \hat{Q}^1(\hat{b}_{i,l}, a_i) + \sum_{o_i, s, q} Pr(o_i|s, a_i, a^q_j)\ b_{i,l}(s) \sum_{m^q_{j,l-1} \in \mathbb{M}^q_{j,l-1}} b_{i,l}(m^q_{j,l-1}|s)\ \hat{U}^n(\hat{b}'_{i,l})$ (using the inductive hypothesis)

$= \hat{Q}^1(\hat{b}_{i,l}, a_i) + \sum_{o_i, s, q} Pr(o_i|s, a_i, a^q_j)\ b_{i,l}(s)\ \hat{b}_{i,l}(\hat{m}^q_{j,l-1}|s)\ \hat{U}^n(\hat{b}'_{i,l})$ (from Eq. 1)

$= \hat{Q}^1(\hat{b}_{i,l}, a_i) + \sum_{o_i, s, q} Pr(o_i|s, a_i, a^q_j)\ \hat{b}_{i,l}(s, \hat{m}^q_{j,l-1})\ \hat{U}^n(\hat{b}'_{i,l})$

$= \hat{Q}^{n+1}(\hat{b}_{i,l}, a_i)$ ∎