Approximate Solutions of Interactive Dynamic Influence Diagrams Using Model Clustering

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

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

Qiongyu Chen
Dept. of Computer Science
National Univ. of Singapore
117543, Singapore
chenqy@comp.nus.edu.sg

Abstract

Interactive dynamic influence diagrams (I-DIDs) offer a transparent and semantically clear representation for the sequential decision-making problem over multiple time steps in the presence of other interacting agents. Solving I-DIDs exactly involves knowing the solutions of possible models of the other agents, which increase *exponentially* with the number of time steps. We present a method of solving I-DIDs approximately by limiting the number of other agents' candidate models at each time step to a constant. We do this by clustering the models and selecting a representative set from the clusters. We discuss the error bound of the approximation technique and demonstrate its empirical performance.

Introduction

Interactive dynamic influence diagrams (I-DIDs) (Doshi, Zeng, & Chen 2007) are graphical models of sequential decision-making in uncertain multi-agent settings. I-DIDs may be viewed as computational counterparts of I-POMDPs (Gmytrasiewicz & Doshi 2005) providing a way to solve I-POMDPs *online*. They generalize DIDs (Tatman & Shachter 1990), which may be viewed as computational counterparts of POMDPs, to multi-agent settings in the same way that I-POMDPs generalize POMDPs. I-DIDs contribute to a growing line of work that includes multi-agent influence diagrams (MAIDs) (Koller & Milch 2001), and more recently, networks of influence diagrams (NIDs) (Gal & Pfeffer 2003). All of 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 provide an alternative to normal and extensive forms of games, using a graphical formalism to represent games of imperfect information. 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. However, both MAIDs and NIDs provide an analysis of the game from an external viewpoint and their applicability is limited to static single play games. Matters are more complex when we consider interactions that 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 aim to address this gap by offering an intuitive way to extend sequential decision-making as formalized by DIDs to multi-agent settings.

As we may expect, I-DIDs acutely suffer from both the curses of dimensionality and history. 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 other agents' beliefs, 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 also from those exhibited by the modeled agents. This is further complicated by the nested nature of the state space.

In this paper, we present methods that reduce the dimensionality of the state space and mitigate the impact of the curse of history that afflicts the other modeled agents. The basic idea, motivated by the point based approaches for POMDPs (Pineau, Gordon, & Thrun 2003), is to limit and hold constant the number of models, $0 < K \ll M$, where M is the possibly large number of candidate models, of the other agents included in the state space at the first time step in the sequential interaction. Using the insight that beliefs that are spatially close are likely to be behaviorally equivalent (Rathnas., Doshi, & Gmytrasiewicz 2006), our approach is to *cluster* the models of the other agents and select representative models from each cluster. In this regard, we utilize the popular k-means clustering method (MacQueen 1967), which gives an iterative way to generate the clusters. Intuitively, the clusters contain 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. We select K representative models from the clusters and update them over time.

At each time step, we begin the clustering by identifying those models that lie on the boundary of the equivalence regions, and use these models as the initial means. Models on each side of the boundary points are expected to exhibit similar behaviors. For two-agent settings, we theoretically

bound the worst case error introduced by the approach in the policy of the other agent and empirically measure its impact on the quality of the policies pursued by the original agent. Our empirical results demonstrate the computational savings incurred in solving the I-DIDs and the favorable performance of the approach.

Overview of I-DIDs

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

Syntax

In addition to the usual chance, decision, and utility nodes, I-IDs include a new type of node called the *model* node (hexagon 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*.

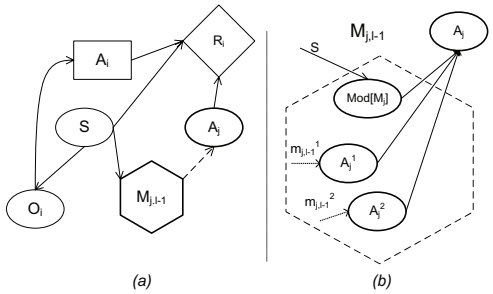

Figure 1: (a) A generic level $l > 0$ I-ID for agent i situated with one other agent j. The hexagon is the model node whose structure we show in (b). Members of the model node are I-IDs themselves ($m^1_{j,l-1}, m^2_{j,l-1}$; not shown here for simplicity) whose decision nodes are mapped to the corresponding chance nodes (A^1_j, A^2_j).

The model node contains as its values the alternative computational models ascribed by i to the other agent. 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, $m_{j,l-1} = \langle b_{j,l-1}, \hat{\theta}_{j,l-1} \rangle$, where $b_{j,l-1}$ is the level $l-1$ belief, and $\hat{\theta}_{j,l-1}$ 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 $Pr(a_j \in A^1_j) = \frac{1}{|OPT|}$ if $a_j \in OPT$, 0 otherwise. The conditional probability table of the chance node, A_j, is a *multiplexer*, that assumes the distribution of each of the action nodes (A^1_j, A^2_j) depending on the value of $Mod[M_j]$. In other words, when $Mod[M_j]$ has the value $m^1_{j,l-1}$, the chance node A_j assumes the distribution of the node A^1_j, and A_j assumes the distribution of A^2_j when $Mod[M_j]$ has the value $m^2_{j,l-1}$. The distribution over $Mod[M_j]$, is i's belief over j's models given the state.

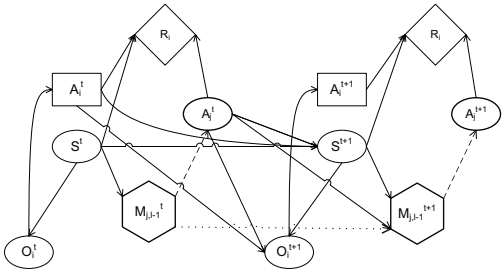

Figure 2: A generic two time-slice level l I-DID for agent i. Notice the dotted model update link that denotes the update of the models of j and the distribution over the models, over time.

I-DIDs extend I-IDs to allow sequential decision-making over several time steps.We depict a general two time-slice I-DID in 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.

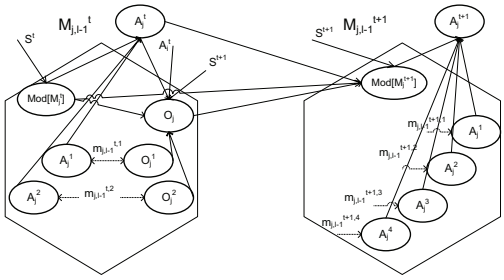

Figure 3: The semantics of the model update link. Notice the growth in the number of models in the model node at $t+1$.

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 at most $|M^t_j||A_j||\Omega_j|$ models. Here, $|M^t_j|$ 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. 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 may be implemented in the I-DID using the standard dependency links and chance nodes, as shown in Fig. 3.

Solution

The solution of an I-DID (and I-ID) proceeds in a bottom-up manner, and is implemented recursively. We start by solv-

ing the level 0 models, which may be traditional IDs. 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. Beliefs over these updated set of candidate models are calculated using the standard inference methods through the dependency links between the model nodes.

Model Clustering and Selection

Because models of the other agent, j, are included as part of the model node in i's I-DID, solution of the I-DID suffers from not only the high dimensionality due to the possibly large number of models of j, M, but also the curse of history responsible for an exponential number of candidate models of j over time. We mitigate the impact of these factors by holding constant the number of candidate models of j in the model node of the I-DID, at each time step.

Initial Means

For illustration, we assume that models of j differ only in their beliefs. Our arguments may be extended to models that differ in their frames as well. In order to selectively pick $0 < K \ll M$ models of j, we begin by identifying the *behaviorally equivalent* regions of j's belief space (Rathnas., Doshi, & Gmytrasiewicz 2006). These are regions of j's belief simplex in which the beliefs lead to an identical optimal policy. As an example, we show in Fig. 4 the behaviorally equivalent regions of j's level 0 belief simplex for the well-known tiger problem (Kaelbling, Littman, & Cassandra 1998). The agent opens the right door (OR) if it believes the probability that the tiger is behind the right door, *P(TR)*, is less than 0.1. It will listen (L) if $0.1 <$ *P(TR)* $<$ 0.9 and open left door (OL) if *P(TR)* > 0.9. Therefore, each of the optimal policies spans over multiple belief points. For example, OR is the optimal action for all beliefs in the set [0–0.1). Thus, beliefs in [0–0.1) are equivalent to each other in that they *induce the same optimal behavior*. Notice that at *P(TR)* = 0.1, the agent is indifferent between OR and L.

We select the initial means as those that lie on the intersections of the behaviorally equivalent regions. This allows models that are likely to be behaviorally equivalent to be grouped on each side of the means.[1] We label these as *sensitivity points* (SPs) and define them below:

[1]Another option could be the centers of the behaviorally equivalent regions. However, for small regions many models that do not belong to the region may also be grouped together.

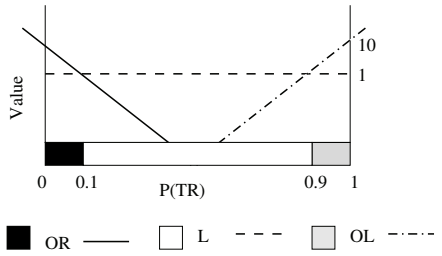

Figure 4: Horizon 1 solution of j's level 0 models in tiger problem. Note the belief ranges corresponding to the optimal actions.

Definition 1 (SP) *Let $b_{j,l-1}$ be a level $l-1$ belief of agent j and OPT($b_{j,l-1}$) be the optimal policy for this belief. Then $b_{j,l-1}$ is a sensitivity point (SP), if for any $\vec{\epsilon}$ s.t. $|\vec{\epsilon}| > 0$, OPT($b_{j,l-1} \pm \vec{\epsilon}$) $\neq$ OPT($b_{j,l-1}$).*

Referring to Fig. 4, *P(TR)* = 0.1 is an SP because slight deviations from 0.1 lead to either OR or L as the optimal action, while at 0.1 the agent is indifferent between the two. In order to compute the SPs, we observe that they are the beliefs at the non-dominated intersection points between the value functions of pairs of policy trees. The following linear program (LP) provides a straightforward way of computing the SPs. For each pair of possible policies of j, π'_j and π''_j as input, we solve:

LP_SP (π'_j,π''_j,Π_j)

Objective:	**Constraints:**
maximize δ	$\forall \pi_j \in \Pi_j / \{\pi'_j, \pi''_j\}$
Variable:	$b_{j,l-1} \cdot Val_{j,l-1}(\pi'_j) - b_{j,l-1} \cdot Val_{j,l-1}(\pi_j) \geq \delta$
$b_{j,l-1}$	$b_{j,l-1} \cdot Val_{j,l-1}(\pi'_j) - b_{j,l-1} \cdot Val_{j,l-1}(\pi''_j) = 0$
	$b_{j,l-1} \cdot 1 = 1$

Table 1: LP for exact computation of SPs.

If $\delta \geq 0$, then the belief, $b_{j,l-1}$, is a SP. Here, Π_j is the space of all horizon T policy trees, which has the cardinality $\mathcal{O}(|A_j|^{2|\Omega_j|^T})$. The computation of the value function, $Val_{j,l-1}(\cdot)$, requires solutions of agent i's level $l-2$ I-DIDs. These may be obtained exactly or approximately; we may recursively perform the model clustering and selection to approximately solve the I-DIDs, as outlined in this paper. The recursion bottoms out at the 0^{th} level where the DIDs may be solved exactly.

The above LP needs to be solved $\mathcal{O}(|A_j|^{2|\Omega_j|^T})$ times to find the SPs exactly, which is computationally expensive. We approximate this computation by randomly selecting K policy trees from the space of policies and invoking **LP_SP** (π'_j, π''_j, Π_j^K), where Π_j^K is the reduced space of policy trees, and $\pi'_j, \pi''_j \in \Pi_j^K$. Computation of the new set of SPs, denoted by SP_K, requires the solution of $\mathcal{O}(K^2)$ reduced LPs allowing computational savings.

In addition to the sensitivity points, we may also designate the vertices of the belief simplex as the initial means. This allows models with beliefs near the periphery of the simplex and away from the SPs, to be grouped together.

With each mean, say the n^{th} SP_K, we associate a cluster, $\mathcal{M}^n_{j,l-1}$, of j's models. The models in $\mathcal{M}^n_{j,l-1}$ are those with beliefs that are closer to the n^{th} SP_K than any other, with ties broken randomly. One measure of distance between belief points is the Euclidean distance, though other metrics such as the L1 may also be used.

Iterative Clustering

The initial clusters group together models of the other agent possibly belonging to multiple behaviorally equivalent regions. Additionally, some of the SP_K may not be candidate models of j as believed by i. In order to promote clusters of behaviorally equivalent models and segregate the non-behaviorally equivalent ones, we update the means using an iterative method often utilized by the k-means clustering approach (MacQueen 1967).

For each cluster, $\mathcal{M}^n_{j,l-1}$, we recompute the mean belief of the cluster and discard the initial mean, SP_K, if it is not in the support of i's belief. The new mean belief of the cluster, $\bar{b}_{j,l-1}$, is:

$$\bar{b}_{j,l-1} = \frac{\sum_{b_{j,l-1} \in \mathcal{B}^n_{j,l-1}} b_{j,l-1}}{|\mathcal{M}^n_{j,l-1}|} \quad (1)$$

Here, the summation denotes additions of the belief vectors, $\mathcal{B}^n_{j,l-1}$ is the set of beliefs in the n^{th} cluster, and $|\mathcal{M}^n_{j,l-1}|$ is the number of models in the n^{th} cluster.

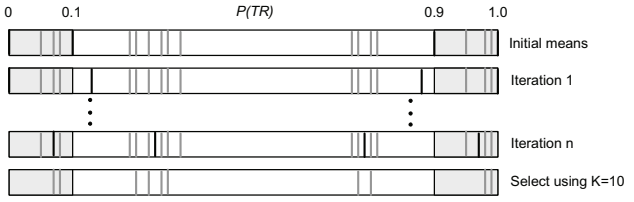

Figure 5: An illustration of the iterative clustering method. The gray vertical lines are the belief points while the black ones are the means. The SPs and the vertices of the belief simplex form the initial means. Notice the movement of the means over the iterations. Once the means have converged, we select K=10 models.

Next, we recluster the models according to the proximity of their beliefs to the revised means. Specifically, models are grouped with the mean to which their respective beliefs are the closest, and all ties are broken randomly. The steps of recomputing the means (Eq. 1) and reclustering using the revised means are repeated until *convergence* ie. the means no longer change. Intuitively, this iterative technique converges because over increasing iterations less new models will be added to a cluster, thereby making the means gradually invariant. We illustrate example movements of the means and clusters of beliefs over multiple iterations in Fig. 5.

Selecting K Models

Given the stable clusters, we select a total of K representative models from them. Depending on its population, a cluster, n, contributes, $k_n = \frac{|\mathcal{M}^n_{j,l-1}|}{M} \times K$ (rounded off to the floor integer) models to the set. The k_n models whose beliefs are the closest to the mean of the cluster are selected for inclusion in the set of models that are retained. Remaining models in the cluster are discarded. The selected models provide representative behaviors for the original set of models included in the cluster.

The models in the model node of i's I-DID, $M^{t+1}_{j,l-1}$, are pruned to include just the K models. These models form the values of the chance node, $Mod[M_j]$ in time step $t+1$. We note that our approach is more suited to situations where agent i has some prior knowledge about the possible models of others, thereby facilitating the clustering and selection.

Algorithm

We show the algorithm, **APPROX_I-DID**, for approximately solving I-DIDs in Fig. 6. The algorithm is a slight variation of the one in (Doshi, Zeng, & Chen 2007) that is used for solving I-DIDs exactly. In particular, on generating the candidate models in the model node, $M^{t+1}_{j,l-1}$, during the *expansion* phase (lines 3-9), we cluster and select K models of these using the procedure **KModelSelection**. We note that models at all levels will be clustered and pruned.

APPROX_I-DID (level $l \geq 1$ I-ID or level 0 ID, T, K)
Expansion Phase
1. **For** t **from** 1 **to** $T-1$ **do**
2. **If** $l \geq 1$ **then**
Populate $M^{t+1}_{j,l-1}$
3. **For each** m^t_j **in** Range($M^t_{j,l-1}$) **do**
4. Recursively call algorithm with $l-1$ I-ID
5. Map the decision node of the solved I-ID (or ID), $OPT(m^t_j)$, to a chance node A_j
6. **For each** a_j **in** $OPT(m^t_j)$, o_j **in** O_j **do**
7. Update j's belief, $b^{t+1}_j \leftarrow SE(b^t_j, a_j, o_j)$
8. $m^{t+1}_j \leftarrow$ New I-ID (or ID) with b^{t+1}_j as the initial belief
9. Range($M^{t+1}_{j,l-1}$) $\overset{\cup}{\leftarrow} \{m^{t+1}_j\}$
Approximate Model Space
10. Range($M^{t+1}_{j,l-1}$) $\leftarrow$ **KModelSelection**(Range($M^{t+1}_{j,l-1}$, $T-t, K$))
11. Add the model node, $M^{t+1}_{j,l-1}$, and the dependency links between $M^t_{j,l-1}$ and $M^{t+1}_{j,l-1}$ (shown in Fig. 3)
12. Add the chance, decision, and utility nodes for $t+1$ time slice and the dependency links
13. Establish the CPTs for each chance node and utility node
Look-Ahead Phase
14. Apply the standard look-ahead and backup method to solve the expanded I-DID

Figure 6: Approximate solution of a level $l \geq 0$ I-DID.

The algorithm for **KModelSelection** (Fig. 7) takes as input the set of models to be pruned, $\mathcal{M}_{j,l-1}$, current horizon H of the I-DID, and the parameter K. We compute the initial means – these are the sensitivity points, SP_K, obtained by solving the reduced LP of Table 1 (line 1; vertices of the belief simplex may also be added). Each model in $\mathcal{M}_{j,l-1}$ is assigned to a cluster based on the distance of its belief to a mean (lines 2-9). The algorithm then iteratively recalculates the means of the clusters and reassigns the models to a

cluster based on their proximity to the mean of the cluster. These steps (lines 10-16) are carried out until the means of the clusters no longer change. Given the stabilised clusters, we calculate the contribution, k_n, of the n^{th} cluster to the set K of models (line 18), and pick the k_n models from the cluster that are the closest to the mean (lines 19-20).

KModelSelection ($\mathcal{M}_{j,l-1}$, H, K)
Initial Means
1. Invoke **LP_SP** on K horizon H policy trees
2. Means$_0 \leftarrow \{SP_K^1, SP_K^2, \ldots, SP_K^n\}$
3. **For** i **from** 1 **to** n **do**
4. $\quad \mathcal{M}_{j,l-1}^i \leftarrow \{SP_K^i\}$ */* Initialize clusters*/*
5. **For each** $m_{j,l-1} = \langle b_{j,l-1}, \hat{\theta}_{j,l-1} \rangle$ **in** $\mathcal{M}_{j,l-1}$ **do**
6. $\quad SP_K^i \leftarrow \text{argmin}_{SP_K \in \text{Means}_0} |SP_K - b_{j,l-1}|$
7. $\quad \mathcal{M}_{j,l-1}^i \overset{\cup}{\leftarrow} m_{j,l-1}$
8. **For** i **from** 1 **to** n **do**
9. $\quad \mathcal{M}_{j,l-1}^i \overset{-}{\leftarrow} \{SP_K^i\}$ if SP_K^i is not in $\mathcal{M}_{j,l-1}$
Iteration
10. **Repeat**
11. $\quad$ **For** i from 1 to n **do**
12 $\quad\quad$ Recompute the mean of each cluster (Eq. 1)
13. $\quad$ **For each** $m_{j,l-1} = \langle b_{j,l-1}, \hat{\theta}_{j,l-1} \rangle$ **in** $\mathcal{M}_{j,l-1}$ **do**
14. $\quad\quad \bar{b}_{j,l-1}^i \leftarrow \text{argmin}_{\bar{b}_{j,l-1}} |\bar{b}_{j,l-1} - b_{j,l-1}|$
15. $\quad\quad \mathcal{M}_{j,l-1}^i \overset{\cup}{\leftarrow} m_{j,l-1}$
16. **Until** no change in the means
Selection
17. **For** i **from** 1 **to** n **do**
18. $\quad k_i \leftarrow \frac{|\mathcal{M}_{j,l-1}^i|}{|\mathcal{M}_{j,l-1}|} \times K$
19. $\quad$ Sort the models in cluster i using distance from mean
20. $\quad \mathcal{M}_K \overset{\cup}{\leftarrow}$ top k_i models
21. **Return** $\mathcal{M}_K$

Figure 7: Algorithm for clustering and selecting K models.

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 $M^0(|A_j||\Omega_j|)^t$ many distinct models of the other agent j, where M^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 M, then solving an I-DID at level l requires the solutions of $\mathcal{O}((NM)^l)$ many models. The **KModelSelection** algorithm reduces the number of agent's models at each level to K representative models while incurring the worst case complexity of $\mathcal{O}(M^2)$. Consequently, we need to solve $\mathcal{O}((NK)^l)$ number of models at each time step in comparison to $\mathcal{O}((NM)^l)$, where M grows exponentially over time. In general, $K \ll M$, resulting in a substantial reduction in the computation.

We bound the error introduced in j's behavior due to excluding all but K models. Recall that for some cluster n, we retain the k_n models closest to the mean. If $K = M$, then we retain all the models and the error is zero. Let $\mathcal{M}_K$ denote the set of K models and $\mathcal{M}_{/K}$ denote the set of the $M-K$ models that are pruned. The error may be bounded by finding the model among the K retained models that is the closest to the discarded one. Define d_K as the largest of the distances between a pruned model, $m_{j,l-1}$, and the closest model among the K selected models: $d_K = max_{m_{j,l-1} \in \mathcal{M}_{/K}} \; min_{m'_{j,l-1} \in \mathcal{M}_K} |b_{j,l-1} - b'_{j,l-1}|$, where $b_{j,l-1}$ and $b'_{j,l-1}$ are the beliefs in $m_{j,l-1}$ and $m'_{j,l-1}$, respectively. Given d_K, the derivation of the error bound proceeds in a manner analogous to that of PBVI (Pineau, Gordon, & Thrun 2003), though over finite horizon, H, of the I-DID. Thus, the worst-case error bound for the set K is:

$$\epsilon_K^H = (R_j^{max} - R_j^{min}) H^2 d_K \quad (2)$$

We may go a step further and gauge the impact of this error on agent i who has a belief over j's models. Using these beliefs, we may compute the expected impact of the error bound in Eq. 2. Let $Pr_i(M_{/K})$ be the probability mass of i's belief, $b_{i,l}$, on the space of pruned models. Thus the worst-case error bounded by Eq. 2 may occur with at most a probability of $Pr_i(M_{/K})$, while no error is incurred with the remaining probability. Consequently, the expected error bound of our approach is:

$$\mathcal{E} = Pr_i(M_{/K}) \times \epsilon_K^H + (1 - Pr_i(M_{/K})) \times 0 = Pr_i(M_{/K})\, \epsilon_K^H \quad (3)$$

For the example case where i's belief is a uniform distribution over the finite set of j's models, Eq. 3 becomes:
$\mathcal{E} = \frac{|M_{/K}|}{M} (R_j^{max} - R_j^{min}) H^2 d_K$

Equations 2 and 3 measure the errors introduced by **KModelSelection** at some nesting level l. These equations assume that the I-DIDs at the lower levels have been solved exactly. However, as we mentioned previously, we may use the model clustering and selection at all levels of nesting to approximately solve the I-DIDs. Deriving error bounds for this more general case is one avenue of future work.

Experiments

We implemented the algorithms in Figs. 6 and 7 and demonstrate the empirical performance of the model clustering approach on two problem domains: the multi-agent tiger (Doshi & Gmytrasiewicz 2005) and a multi-agent version of the machine maintenance problem (Smallwood & Sondik 1973). We also compare the performance with an implementation of the interactive particle filter (I-PF) (Doshi & Gmytrasiewicz 2005) in the context of I-DIDs. In particular, we show that the quality of the policies generated using our method approaches that of the exact policy as K increases. As there are infinitely many computable models, we obtain the exact policy by *exactly* solving the I-DID given a finite set of M models of the other agent. In addition, the approach performs better than the I-PF when both are allocated low numbers of models and consequently, less computational resources. Furthermore, we obtain significant computational savings from using the approximation technique as indicated by the low run times.

In Fig. 8, we show the average rewards gathered by executing the policies obtained from solving the level 1 I-DIDs

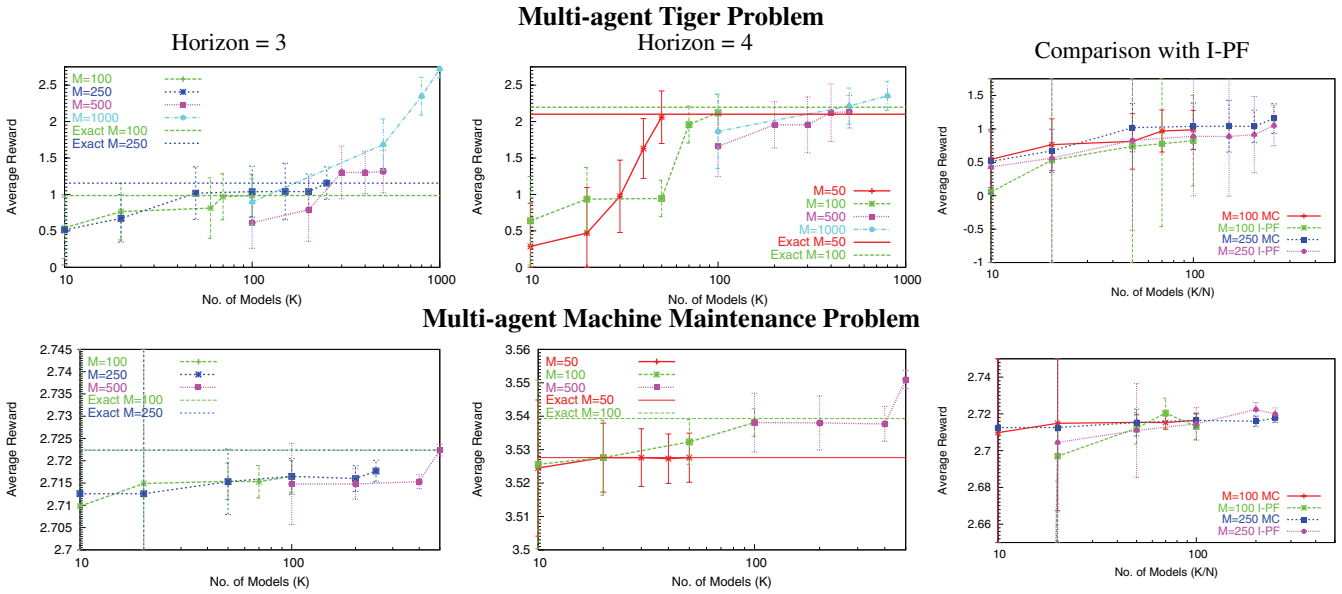

Figure 8: Performance profiles for the multi-agent tiger and machine maintenance problems generated by executing the policies obtained using model clustering and selection (MC). As the number of allocated models, K, increases, the performance approaches that of the exact for given M (shown as the straight line). We show this for different numbers, M, of candidate models of j, and compare with the I-PF.

approximately. Each data point here is the average of 50 runs where the true model of the other agent, j, is randomly picked according to i's belief distribution over j's models. Each curve within a plot is for a particular M, where M denotes the *total* number of candidate models of j. As we increase the number of models selected, K, from the clusters, the policies improve and converge toward the exact. This remains true for increasing M and for both, the multi-agent tiger and machine maintenance problem domains.

We observe from Fig. 8 that our approach obtains better average rewards with reduced variance than I-PF for small numbers, K, of selected models. This is due to the large variance in performances of PFs for small numbers of samples, which is a well-known problem. For larger numbers of models, both approaches exhibit similar performances.

Prob.	**Tiger**		**Machine**	
Exact	83.6s		99.2s	
	K = 20	K=50	K=20	K=50
MC	3.8s	10.5s	6.2s	18.7s
I-PF	3.9s	9.5s	4.3s	10.8s

Table 2: Run times of exactly and approximately solving the I-DID for a horizon of 4 and M=100 (3.0GHz, 1GB RAM, WinXP).

Finally, the run times in Table 2 are indicative of the computational savings incurred by pruning the model space to a fixed number of models at each time step in the I-DID. However, the approach is somewhat slower than the I-PF because of the convergence step, though its performance is significantly better as shown in Fig. 8. Using the clustering approach we were able to solve I-DIDs up to 8 horizons, while the exact solutions could not be obtained beyond 4 horizons. We expect similar results as we evaluate for deeper levels of strategic nesting of the models.

References

Doshi, P., and Gmytrasiewicz, P. J. 2005. A particle filtering based approach to approximating interactive pomdps. In *AAAI*.

Doshi, P.; Zeng, Y.; and Chen, Q. 2007. Graphical models for online solutions to interactive pomdps. In *AAMAS*.

Gal, Y., and Pfeffer, A. 2003. A language for modeling agent's decision-making processes in games. In *AAMAS*.

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

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

Koller, D., and Milch, B. 2001. Multi-agent influence diagrams for representing and solving games. In *IJCAI*, 1027–1034.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Berkeley Symposium on Mathematics, Statistics, and Probablity*. 281–297.

Pineau, J.; Gordon, G.; and Thrun, S. 2003. Point-based value iteration: An anytime algorithm for pomdps. In *IJCAI*.

Rathnas., B.; Doshi, P.; and Gmytrasiewicz, P. J. 2006. Exact solutions to interactive pomdps using behavioral equivalence. In *AAMAS*.

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

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