Improved Use of Partial Policies for Identifying Behavioral Equivalence

Yifeng Zeng
Dept. of Computer Science Dept. of Automation
Aalborg University Xiamen University
Aalborg, Denmark Xiamen, China
yfzeng@cs.aau.edu yfzeng@xmu.edu.cn

Yinghui Pan
Dept. of Automation Information Management
Xiamen University Jiangxi Univ. of Fin.& Eco.
Xiamen, China Jiangxi, China
pyhui@xmu.edu.cn

Hua Mao
Dept. of Computer Science
Aalborg University
Aalborg, Denmark
huamao@cs.aau.edu

Jian Luo
Dept. of Automation
Xiamen University
Xiamen, China
jianluo@xmu.edu.cn

ABSTRACT

Interactive multiagent decision making often requires to predict actions of other agents by solving their behavioral models from the perspective of the modeling agent. Unfortunately, the general space of models in the absence of constraining assumptions tends to be very large thereby making multiagent decision making intractable. One approach that can reduce the model space is to cluster *behaviorally equivalent* models that exhibit identical policies over the whole planning horizon. Currently, the state of the art on identifying equivalence of behavioral models compares partial policy trees instead of entire trees. In this paper, we further improve the use of partial trees for the identification purpose and develop an incremental comparison strategy in order to efficiently ascertain the model equivalence. We investigate the improved approach in a well-defined probabilistic graphical model for sequential multiagent decision making - interactive dynamic influence diagrams, and evaluate its performance over multiple problem domains.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

Keywords

decision making, agent modeling, behavioral equivalence

1. INTRODUCTION

Decision making in interactive multiagent settings becomes complicated mainly due to unknown actions of other agents from the eyes of the modeling agent. A general solution is to model other agents using a specific representation and then solve the models to predict their actions. Unfortunately, the model space ascribed to other agents is often very large thereby making multiagent decision making intractable. A line of research has exploited the concept of *behavioral equivalence (BE)* to reduce the dimensionality of the model space [2, 12, 13]. A pair of models are behaviorally equivalent if the models have identical solutions that are normally represented as policy trees. We may consider to group a set of BE models and choose a representative model for each cluster. Clustering BE models to reduce the model space will not compromise the solution optimality since it is the prescriptive aspects of the models and not the descriptive that matter to the modeling agent. Recently, a well-defined probabilistic decision making framework - interactive dynamic influence diagram (I-DID) [6] - has intensively exploited BE models for achieving the solution scalability.

I-DIDs are probabilistic graphical models for sequential decision making in uncertain multiagent settings. They generalize dynamic influence diagrams (DIDs) [14] to multiagent settings analogously to the way that interactive partially observable Markov decision processes (I-POMDPs) [9] generalize POMDPs. As we may expect, solving I-DIDs is computationally very hard. This is because the state space in I-DIDs includes the models of other agents in addition to the traditional physical states. As the agents act, observe and update beliefs, I-DIDs must track the evolution of the models over time. The exponential growth in the number of models over time also further contributes to the dimensionality of the state space. This is further complicated by the nested nature of the state space.

Previous I-DID solutions, including both exact and approximate ones, mainly exploit the concept of BE to reduce the dimensionality of the state space. For example, the proposed technique in [5] updates only those models that lead to behaviorally distinct models at the next time step. It results in a minimal model space. A central component of this technique is the way of identifying equivalence of behavioral models ascribed to other agents. It firstly builds policy trees for the associated models and then checks the equality of every path in the entire trees. Since the size of the policy

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

tree increases exponentially as the horizon increases, the BE identification method becomes computationally intractable in the case of large horizons. Additionally, based on this identification technique, the current I-DID solution does not scale desirably to large horizons because it groups only exact BE models thereby still resulting in a large model space. One leading solution to further reduce the model space is to cluster models that are approximately BE.

Recently, one efficient way of identifying approximately BE between models is to compare their partial policy trees instead of entire ones [19]. The depth of the partial trees is determined by a given approximate measure of BE. This defines an approximately BE that could group more models together resulting in less numbers of BE classes. However, the proposed method still requires an expansion of a full size of the partial policy tree that has a symmetric structure with a uniform length on all paths. This may lead to a strict condition on approximating BE while the identification using partial trees is not executed efficiently.

In this paper, we present an improved version of using partial trees to identify approximately BE models. We make a general definition on a partial policy tree that allows different lengths for its paths. The maximum path length is calculated according to a predefined value on measuring the approximation between two BE models. The measurement value quantifies the allowed divergence between updated beliefs in the policy trees. To efficiently use partial policy trees to determine approximately BE, we propose an incremental identification approach: we expand the trees only when comparing the updated beliefs at the leaf nodes is not sufficient to ascertain the model equivalence. The comparison expects to be terminated before it reaches the maximum length for all paths. By doing this we maintain a rather small set of policy paths instead of all full paths in the partial trees. Specifically, the incremental method is applicable even when the maximum length can't be computed in some problem domains.

Furthermore, we may group more approximately BE models by comparing only a subset of policies in the partial trees. As the comparison of the policy paths may terminate before it reaches their maximum lengths, the error is introduced on predicting the future policies. We bound the prediction error due to the incomplete search of the partial trees on determining approximately BE. Finally, we evaluate the empirical performance of the proposed approach in the context of multiple problem domains, and demonstrate its scalability on solving I-DIDs of significantly large horizons.

2. BACKGROUND: INTERACTIVE DID AND BEHAVIORAL EQUIVALENCE

We start with a brief review on interactive dynamic influence diagram (I-DID) and then describe its solutions that are developed using the technique on clustering behaviorally equivalent models. More details could be found in this line of research [6, 5, 19].

2.1 Interactive Dynamic Influence Diagram

I-DIDs extend probabilistic graphical models - dynamic influence diagrams (DIDs) [14] - to represent how agents make a sequence of rational decisions while interacting with other agents over time in an uncertain environment. A regular DID models sequential decision making for a single agent by linking a set of chance, decision and utility nodes over multiple time steps. To consider multiagent interaction, I-DIDs introduce a new type of node called the *model node* (hexagonal node, $M_{j,l-1}$, in Fig. 1) that represent how another agent j acts simultaneously when the modeling agent i reasons its own decisions at level l. The model node contains a set of j's candidate models at level $l-1$ ascribed by i . A link from the chance node S to the model node $M_{j,l-1}$ represents agent i's beliefs over j's models. Specifically, it is a probability distribution in the conditional probability table (CPT) of the chance node $Mod[M_j]$ (in Fig. 2). Each model, $m_{j,l-1}$, could be either a level $l-1$ I-DID or a DID at level 0. Model solutions are the predicted behavior of j and are encoded into a chance node A_j through a dashed link, called a *policy link*. Connecting A_j with other nodes in an I-DID structures how agent j's actions are engaged in i's decision making process.

Expanding an I-DID involves the update of the model node over time as indicated in the *model update link* - a dotted arrow from $M^t_{j,l-1}$ to $M^{t+1}_{j,l-1}$ in Fig. 1. As agent j acts and receives observations over time, its models are updated to reflect their changed beliefs. For each model $m^t_{j,l-1}$ at time t, its optimal solutions may include all decision options and agent j may receive any of the possible observations. Consequently, the set of updated models at time $t+1$ will have up to $|\mathcal{M}^t_{j,l-1}||A_j||\Omega_j|$ models. Here, $|\mathcal{M}^t_{j,l-1}|$ is the number of models at time step t, $|A_j|$ and $|\Omega_j|$ are the largest spaces of actions and observations respectively. The models differ in their initial beliefs updated using a configuration of action and observation. The CPT of $Mod[M^{t+1}_{j,l-1}]$ specifies the function, $\tau(b^t_{j,l-1}, a^t_j, o^{t+1}_j, b^{t+1}_{j,l-1})$ which is 1 if the belief $b^t_{j,l-1}$ in the model $m^t_{j,l-1}$ using the action a^t_j and observation o^{t+1}_j updates to $b^{t+1}_{j,l-1}$ in a model $m^{t+1}_{j,l-1}$; otherwise it is 0. We may implement the model update link using standard dependency links and chance nodes, as shown in Fig. 2, and transform an I-DID into a regular DID. Consequently, any DID technique can be exploited to solve an I-DID. Details on algorithms for solving an I-DID are in [6].

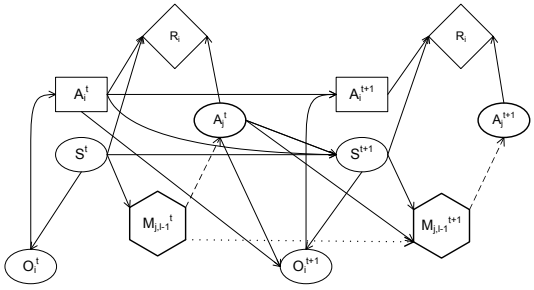


Figure 1: A generic two time-slice level l I-DID for agent i. Notice the dotted model update link that denotes the update of the models of j and of the distribution over the models, over time.

2.2 Behavioral Equivalence and Its Identification

As we may expect, the complexity of solving I-DIDs is mainly due to the growing space of possible models ascribed to other agents. It is computationally impossible if all models are considered in the model node. As the modeling agent

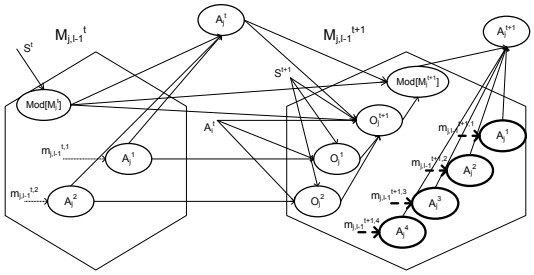


Figure 2: Implementation of the model update link using standard dependency link and chance nodes e.g. two models, $m^{t,1}_{j,l-1}$ and $m^{t,2}_{j,l-1}$, are updated into four models (shown in bold) at time $t+1$.

cares only about the predicted behavior, not the descriptive models, of the other agent, those models that have identical solutions need not be distinguished on solving I-DIDs. In other words, models that are *BE* [12] – whose behavioral predictions for the other agent are identical – could be pruned and a single representative model considered. Based on this strategy on reducing the model space, a set of algorithms have been developed with the purpose of scaling up solutions to I-DIDs over a large number of horizons [18, 5, 19]. All of the algorithms need to cope with the problem of identifying BE between a pair of models.

In the I-DID context, the other agent j's model, $m_{j,l-1}$ is a level $l-1$ I-DID or a DID if l equals to 1. Without loss of generality, we represent model solutions of T horizons as a *policy tree*, denoted by $OPT(m_{j,l-1}) \triangleq \pi^T_{m_{j,l-1}}$ where $OPT(\cdot)$ denotes the solution of the model that forms the argument. Two models, $m_{j,l-1}$ and $\hat{m}_{j,l-1}$, are BE if and only if $\pi^T_{m_{j,l-1}} = \pi^T_{\hat{m}_{j,l-1}}$. The BE identification requires to maintain and compare the entire policy trees each of which contains $(|\Omega_j|)^{T-1}$ possible paths. This is inefficient on both computational time and memory. To resolve this inefficiency, Zeng *et al.* [19] recently propose one technique to identify approximately BE by comparing depth-q ($q \leq T$) policies as well as updated beliefs, $b^{q,k}_{m_{j,l-1}}$, at the leaf nodes of the partial policy trees. Formally, let $D_{KL}[p||p']$ denote the KL divergence [11] between probability distributions, p and p'. The technique defines an approximately BE for a given measure ϵ (≥ 0).

DEFINITION 1 ((ϵ, q)-BE). *Two models of agent j, $m_{j,l-1}$ and $\hat{m}_{j,l-1}$, are (ϵ,q)-BE, $\epsilon \geq 0$, $q \leq T$, if their depth-q policy trees are identical, $\pi^q_{m_{j,l-1}} = \pi^q_{\hat{m}_{j,l-1}}$, and if $q < T$ then beliefs at the leaves of the two policy trees diverge by at most ϵ: $\max_{k=1\ldots|\Omega_j|^q} D_{KL}[b^{q,k}_{m_{j,l-1}}||b^{q,k}_{\hat{m}_{j,l-1}}] \leq \epsilon$.*

More importantly, it is found that the depth q of the partial tree can be determined given some ϵ. Eq. 1 shows the way of computing q, where γ_F is a minimal mixing rate in a stochastic transition and $b^{0,k}_{m_{j,l-1}}$ ($b^{0,k}_{\hat{m}_{j,l-1}}$)) initial beliefs in j's model $m_{j,l-1}$ ($\hat{m}_{j,l-1}$). The computation is based on the fact: the KL divergence between the distributions over the same space contacts with the rate $(1-\gamma_F)$ after one transition [1]. In the I-DID context, γ_F is the minimum probability mass on some state due to the transition, and is computed by multiplying the state transition probability and the likelihood of observation for j.

$$q = min\left\{T, max\{0, \lfloor \frac{ln\frac{\epsilon}{D_{KL}(b^{0,k}_{m_{j,l-1}}||b^{0,k}_{\hat{m}_{j,l-1}})}}{ln(1-\gamma_F)} \rfloor\}\right\} \quad (1)$$

Accordingly, the straightforward implementation for identifying (ϵ, q)-BE is to firstly build partial policy trees of depth-q (line 2) and then check the equality between them (line 3). It is called as a plain algorithm for identifying (ϵ, q)-BE of two models, ϵ-BE-P, as shown in Fig. 3.

ϵ-BE-P (Models, $m_{j,l-1}$ and $\hat{m}_{j,l-1}$, Horizon T, and parameters, γ_F and ϵ)

1. Compute q according to Eq. 1
2. Build depth-q partial trees: $\pi^q_{m_{j,l-1}}$ and $\pi^q_{\hat{m}_{j,l-1}}$
3. **If** $\pi^q_{m_{j,l-1}} = \pi^q_{\hat{m}_{j,l-1}}$
4. Return $True$; **Else**, Return $False$

Figure 3: A plain algorithm, ϵ-BE-P, for approximate BE identification by comparing the entire depth-q trees.

3. INCREMENTAL BE IDENTIFICATION

As discussed above, identifying the behaviorally equivalent models of the other agent j plays a central role in the I-DID solutions. Using partial policy trees provides a promising direction to scale BE to large horizons since it groups together more models that could be approximately BE and simplifies the complexity of identifying BE by comparing only a subset of the entire policy trees. A plain realization of this strategy is to compare the partial policy trees that are symmetric and are fully constructed using a uniform length for all policy paths. We aim to further enhance the use of partial policies to cluster more approximately BE models in a more efficient way. We firstly define approximately BE models using asymmetric policy trees and then propose an incremental technique to identify the models.

3.1 Approximate BE

A q-length policy path is an action-observation sequence describing what agent j acts and observes over q time steps. It is denoted by, $h^q_j = \{a^t_j, o^{t+1}_j\}^q_{t=1}$, where o^{T+1}_j is null for a T ($q \leq T-1$) horizon planning problem. If $a^t_j \in A_j$ and $o^{t+1}_j \in \Omega_j$, where A_j and Ω_j are agent j's action and observation sets respectively, then a depth-q policy tree is a set of all q-length paths: $\pi^q_j = \Pi^q_1(A_j \times \Omega_j)$ where o^q_j is null. As we may notice, the tree is symmetric since all paths have the same length q. For an asymmetric policy tree, we need to enumerate the set of policy paths and some paths may differ in the length. We index paths of the same length by imposing an order on the observations in the policy tree. Formally, let $\pi^{q_L,q_U}_j =< h^{q_1,1}_j, \cdots, h^{q_r,k}_j >$ be the asymmetric policy tree of depth-(q_L, q_U) where q_L is the minimum length, $q_L = Min(q_1, \cdots, q_r)$, q_U the maximum one, $q_U = Max(q_1, \cdots, q_r)$, and k an index number.

Notice that beliefs updated using an action-observation sequence in a partially observable stochastic process is a sufficient statistic for the history. Consequently, future policies are predicted only on the updated beliefs. If $b^{0,k}_{j,l-1}$ is

the initial belief in the model, $m_{j,l-1}$, then let $b_{j,l-1}^{q,k}$ be the new belief on updating it using the q-length policy path $h_j^{q,k}$. The policies, $\Pi_{q+1}^T(A_j \times \Omega_j)$, succeeding to the path $h_j^{q,k}$ can be predicted using the belief $b_{j,l-1}^{q,k}$. Using the partial trees and updated beliefs, we may re-write the full policy tree as follows: $\pi_{m_{j,l-1}}^T = < \pi_{m_{j,l-1}}^{q_L,q_U}, B_{m_{j,l-1}}^{q_L,q_U} >=< (h_j^{q_1,1}, b_{m_{j,l-1}}^{q_1,1}), \cdots, (h_j^{q_r,k}, b_{m_{j,l-1}}^{q_r,k}) >$, where $B_{m_{j,l-1}}^{q_L,q_U}$ is the set of updated beliefs. Consequently, comparing a small number of policy paths and beliefs is sufficient to identify BE. We modify Def. 1 to formulate an approximately BE, called (ϵ, q_L, q_U)-BE, between models as follows.

DEFINITION 2 ((ϵ, q_L, q_U)-BE). *Two models of agent j, $m_{j,l-1}$ and $\hat{m}_{j,l-1}$, are (ϵ, q_L, q_U)-BE, $\epsilon \geq 0$, $q_U \leq T$, if their depth-(q_L, q_U) policy trees are identical, $\pi_{m_{j,l-1}}^{q_L,q_U} = \pi_{\hat{m}_{j,l-1}}^{q_L,q_U}$, and if $q_U < T$ then updated beliefs for the two policy trees diverge by at most ϵ:*
$\max_{(q_1,1),\cdots,(q_r,k)} D_{KL}[b_{m_{j,l-1}}^{q_r,k} || b_{\hat{m}_{j,l-1}}^{q_r,k}] \leq \epsilon.$

Intuitively, two models are (ϵ, q_L, q_U)-BE if they have identical solutions of depth-(q_L, q_U) trees and the divergence of pairs of the updated beliefs at the leaves of the depth-(q_L, q_U) tree is not larger than ϵ. Two (ϵ, q_L, q_U)-BE models become exact BE as ϵ approaches zero. If the partial tree is symmetric in the setting of $q_L = q_U$, (ϵ, q_L, q_U)-BE is equivalent to the notion of approximately BE in Def. 1. Hence (ϵ, q_L, q_U)-BE provides a general definition of approximately BE using asymmetric partial trees. The remaining question is how to compute values for the parameters, q_L and q_U, given some ϵ.

As mentioned in Sec. 2.2, the mixing rate is computed as the minimal one for the transitions of all possible action-observation pairs. Eq. 1 provides a principled way of determining the maximum length q_U for all policy paths given the amount of approximation ϵ. Meanwhile, we observe that the divergence of updated beliefs using some paths may turn out to be much less than ϵ before the paths are fully extended into the length q_U. This may occur due to the fact that the KL divergence of belief distributions contracts monotonically over time [1]. The minimum length q_L is the earliest time when the belief divergence is known to be smaller than ϵ. Its value is found during the BE identification.

3.2 Incremental Comparison

(ϵ, q_L, q_U)-BE provides a novel way to identify approximately BE and compares partial trees with an asymmetric structure. This differs from (ϵ, q)-BE that needs to compare a full size of partial trees. Due to the unknown value for the minimum length, the size of asymmetric trees can't be decided given a single input of approximation measure ϵ. However, we are able to bound the tree size using the maximum path length, which avoids an arbitrary expansion on the tree. For the purpose of identifying (ϵ, q_L, q_U)-BE, we propose an incremental technique below.

We compare both partial trees and updated beliefs at the leaves of the trees when we expand the policy tree at every time step. We terminate the comparison once there is any unmatched behavior in the paths; otherwise, we expand the trees until the depth of the partial trees reaches the maximum value q_U. In addition, we do not further expand a partial tree at the end of a policy path if the path is identical and the divergence of updated beliefs is not larger than ϵ. This is because the equivalence of future behavior can be sufficiently determined without checking the unexpanded partial trees. The q_L value is the minimum length of all paths when the comparison terminates. The procedure is an incremental policy comparison for the (ϵ, q_L, q_U)-BE identification, called ϵ-BE-I. We illustrate the procedure using an example in Fig. 4. The example is constructed in the *Tiger* problem domain - well studied in the POMDP literature.

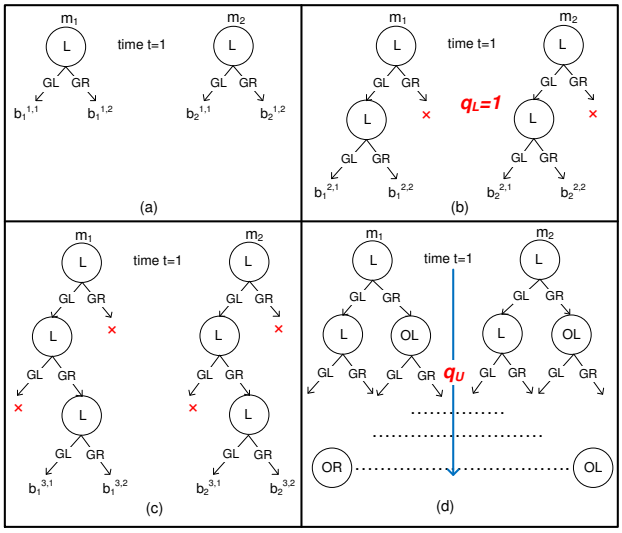


Figure 4: ϵ-BE-I, (a)-(c), and ϵ-BE-P, (d), for (ϵ, q_L, q_U)-BE identification.

Example: We are checking whether two models of agent j, m_1 and m_2, are (ϵ, q_L, q_U)-BE given the approximation amount ϵ. Assuming that b_1^0 and b_2^0 are their initial beliefs and the mixing rate γ_F is computed in the domain, we calculate the q_U value [1] in Eq. 1 that will serve as the upper bound for iterating the comparison. Since $D_{KL}(b_1^0||b_2^0)$ is larger than ϵ, we need to build the root nodes for the policy trees that are solutions of two models respectively. We then update their initial beliefs into new ones given possible observations (GL and GR) because both trees have identical actions L at time t=1 (Fig. 4(a)). We compute the divergences of each pair of new beliefs given the same observation like $D_{KL}(b_1^{1,1}||b_2^{1,1})$ and $D_{KL}(b_1^{1,2}||b_2^{1,2})$. Suppose that $D_{KL}(b_1^{1,1}||b_2^{1,1})$ is still larger than ϵ while $D_{KL}(b_1^{1,2}||b_2^{1,2})$ is less than ϵ. We must expand the policy tree following the path $\{L, GL\}$, but will not continue the expansion in the other path $\{L, GR\}$ at t=2 (Fig. 4(b)). We say that this path is *blocked* (denoted by $\times$) and will not be considered for a further expansion. The minimum path length q_L is now found and is equal to 1. We compare the policies following the path $\{L, GL\}$, and update their beliefs if the policies are equivalent at t=2. We repeat the same procedure at t=3 and so on until either the maximal depth q_U is approached or no policy paths can be further expanded (Fig. 4(c)). The incremental procedure may generate a small size of the partial policy trees for identifying the equivalence between m_1 and m_2. For the same identification purpose, the previous algorithm, ϵ-BE-P, needs to construct and compare the partial policy trees that expand all paths to the maximal length

[1]We may predefine the q_U value if it can't be computed in Eq. 1

q_U (Fig. 4(d)).

In addition, we observe that the depth value q can't be calculated in Eq. 1 if the mixing rate γ_F becomes zero. To run the ϵ-BE-P algorithm, we need to specify the q value even given the known approximation amount ϵ. This results in a partial policy tree that is arbitrarily large. We need to check the equality for all paths in the partial tree for the identification purpose. On the other hand, the incremental algorithm, ϵ-BE-I, employs ϵ as the threshold value to prune the partial trees while it performs the path comparison and expands the trees. The predefined depth q_U acts as an upper bound value to terminate the identification process if it is necessary. In summary, the incremental policy comparison algorithm becomes a universal approach for identifying approximately BE models.

3.3 Algorithm

We present the incremental policy comparison algorithm for identifying (ϵ, q_L, q_U)-BE between two models in Fig. 5. As mentioned in the previous section, the algorithm terminates the identification process when any of the following conditions is met: (a) Initial beliefs diverge at most ϵ and (ϵ, q_L, q_U)-BE of two models are immediately ascertained (line 6); (b) Any unmatched policy is detected and the models are not (ϵ, q_L, q_U)-BE (line 11); (c) (ϵ, q_L, q_U)-BE is confirmed for two models when either no path can be further expanded or the depth q_U is approached (lines 12-14). By doing this, we can avoid the expansion of entire depth-q_U trees while achieving the identification of (ϵ, q_L, q_U)-BE between two models.

ϵ-BE-I (Models, $m_{j,l-1}$ and $\hat{m}_{j,l-1}$, Horizon T, and parameters, γ_F and ϵ)

1. Case $\gamma_F \in (0, 1]$: Compute q_U according to Eq. 1
2. Case γ_F=0: Specify $q_U \leq T$
3. **For** t=1 to q_U **do**
4. **If** $D_{KL}(b^{t-1,k}_{m_{j,l-1}} || b^{t-1,k}_{\hat{m}_{j,l-1}}) \leq \epsilon$
5. **Case** $t>1$: Block the path $h^{t,k}_{m_{j,l-1}}$ ($h^{t,k}_{\hat{m}_{j,l-1}}$)
6. **Case** t=1: Return $True$ and **Break**
7. **else**
8. **If** $h^{t,k}_{m_{j,l-1}} = h^{t,k}_{\hat{m}_{j,l-1}}$
9. **Case** $t<T$: Expand the t-length paths and compute the updated belief $b^{t,k}_{m_{j,l-1}}$ ($b^{t,k}_{\hat{m}_{j,l-1}}$) given the path $h^{t,k}_{m_{j,l-1}}$ ($h^{t,k}_{\hat{m}_{j,l-1}}$)
10. **Case** t=T: Return $True$
11. **else** Return $False$ and **Break**
12. **If** All paths $h^{t,k}_{m_{j,l-1}}$ ($h^{t,k}_{\hat{m}_{j,l-1}}$) are blocked
13. Return $True$
14. Return $True$

Figure 5: An incremental algorithm, ϵ-BE-I, for determining the equivalence of two models given the approximate amount ϵ.

ϵ-BE-I differs from ϵ-BE-P since it compares only a subset of depth-q_U trees. It blocks the path $h^{q_r,k}_{m_{j,l-1}}$ ($h^{q_r,k}_{\hat{m}_{j,l-1}}$) for a further comparison when the divergence of beliefs, $D_{KL}(b^{q_r,k}_{m_{j,l-1}} || b^{q_r,k}_{\hat{m}_{j,l-1}})$, is smaller than ϵ at time step q_r. Notice that the partial trees succeeding to $h^{q_r,k}_{m_{j,l-1}}$ ($h^{q_r,k}_{\hat{m}_{j,l-1}}$) may not be identical although the updated beliefs have a small amount of divergence. Consequently, ϵ-BE-I may result in grouping more approximately BE than ϵ-BE-P.

4. COMPUTATIONAL SAVINGS AND ERROR BOUND

Algorithms for determining (ϵ, q_L, q_U)-BE of a pair of models mainly perform the path comparison in policy trees. The complexity is proportional to the number of comparisons required to approximately decide the equivalence. For the ϵ-BE-P algorithm, we need to compare every path in partial trees of depth-q. Since there are a maximum of $|\Omega_j|^{q_U-1}$ leaf nodes in a depth-q_U tree, the complexity of ϵ-BE-P is $\mathcal{O}(|\mathcal{M}_{j,l-1}|^2 |\Omega_j|^{q_U})$ where $|\mathcal{M}_{j,l-1}|$ is the number of candidate models. On the other hand, the ϵ-BE-I algorithm prunes the paths while it traverses a depth-q_U tree from the root. This may result in an asymmetric partial tree where the number of leaf nodes is N where $N \ll |\Omega_j|^{q_U-1}$. Meanwhile the ϵ-BE-I algorithm needs to compare beliefs for which the number is also bounded by N. Consequently, the complexity of ϵ-BE-I becomes $\mathcal{O}(2|\mathcal{M}_{j,l-1}|^2 N)$. In addition, ϵ-BE-I involves the belief calculation in the procedure that costs little on propagating beliefs in solved models.

Both algorithms preclude storing entire policy trees that contain $(|\Omega_j|)^{T-1}$ possible paths. For the ϵ-BE-P algorithm, we maintain at most $2(|\Omega_j|)^{q_U-1}$ paths ($q_U \leq T$) at each time step when a pair of models are under the identification. For the ϵ-BE-I algorithm, we need to store only $2N$ paths each of which has the length bounded by q_U. Hence ϵ-BE-I achieves much better memory efficiency compared to ϵ-BE-P.

We analyze the error in the value of j's predicted behavior. An error occurs when a behaviorally distinct model, $m_{j,l-1}$, is grouped with the model, $\hat{m}_{j,l-1}$, given an approximation amount ϵ. Let $m_{j,l-1}$ be the model associated with $\hat{m}_{j,l-1}$, resulting in the worst error. Let α_T and $\hat{\alpha}_T$ be the exact entire policy trees obtained by solving the two models, respectively. Then, the error is: $\rho = |\alpha^T \cdot b^0_{m_{j,l-1}} - \alpha^T \cdot b^0_{\hat{m}_{j,l-1}})|$. As ϵ-BE-I starts to prune the path at the length q_L. The error in the worst case becomes:

$$
\begin{aligned}
\rho &= |\alpha^{T-q_L} \cdot b^{q_L}_{m_{j,l-1}} - \alpha^{T-q_L} \cdot b^{q_L}_{\hat{m}_{j,l-1}}| \\
&= |\alpha^{T-q_L} \cdot b^{q_L}_{m_{j,l-1}} + \hat{\alpha}^{T-q_L} \cdot b^{q_L}_{m_{j,l-1}} - \hat{\alpha}^{T-q_L} \cdot b^{q_L}_{m_{j,l-1}} \\
&\quad -\alpha^{T-q_L} \cdot b^{q_L}_{\hat{m}_{j,l-1}}| \quad \text{(add zero)} \\
&\leq |\alpha^{T-q_L} \cdot b^{q_L}_{m_{j,l-1}} + \hat{\alpha}^{T-q_L} \cdot b^{q_L}_{\hat{m}_{j,l-1}} - \hat{\alpha}^{T-q_L} \cdot b^{q_L}_{m_{j,l-1}} \\
&\quad -\alpha^{T-q_L} \cdot b^{q_L}_{\hat{m}_{j,l-1}}| \quad (\hat{\alpha}^{T-q_L} \cdot b^{q_L}_{\hat{m}_{j,l-1}} \geq \hat{\alpha}^{T-q_L} \cdot b^{q_L}_{m_{j,l-1}}) \\
&= |(\alpha^{T-q_L} - \hat{\alpha}^{T-q_L}) \cdot (b^{q_L}_{m_{j,l-1}} - b^{q_L}_{\hat{m}_{j,l-1}})| \\
&\quad \text{(Hölder's ineq.)} \\
&\leq |\alpha^{T-q_L} - \hat{\alpha}^{T-q_L}|_\infty \cdot |(b^{q_L}_{m_{j,l-1}} - b^{q_L}_{\hat{m}_{j,l-1}})|_1 \\
&\quad \text{(Pinsker's ineq.)} \\
&\leq |\alpha^{T-q_L} - \hat{\alpha}^{T-q_L}|_\infty \cdot 2 D_{KL}(b^{q_L}_{m_{j,l-1}} || b^{q_L}_{\hat{m}_{j,l-1}}) \\
&\leq (R_j^{max} - R_j^{min})(T - q_L) \cdot 2\epsilon \quad \text{(by definition)}
\end{aligned}
$$

Here, R_j^{max} and R_j^{min} are the maximum and minimum rewards of j, respectively. This error bound is not tight as that of ϵ-BE-P which is $(R_j^{max} - R_j^{min})(T - q_U) \cdot 2\epsilon$ when $q_L < q_U$. The gap is due to the utilization of ϵ on approximating BE at different depths as previously mentioned. We expect that the subtle difference has a limited impact on the I-DID solutions by clustering approximately BE models.

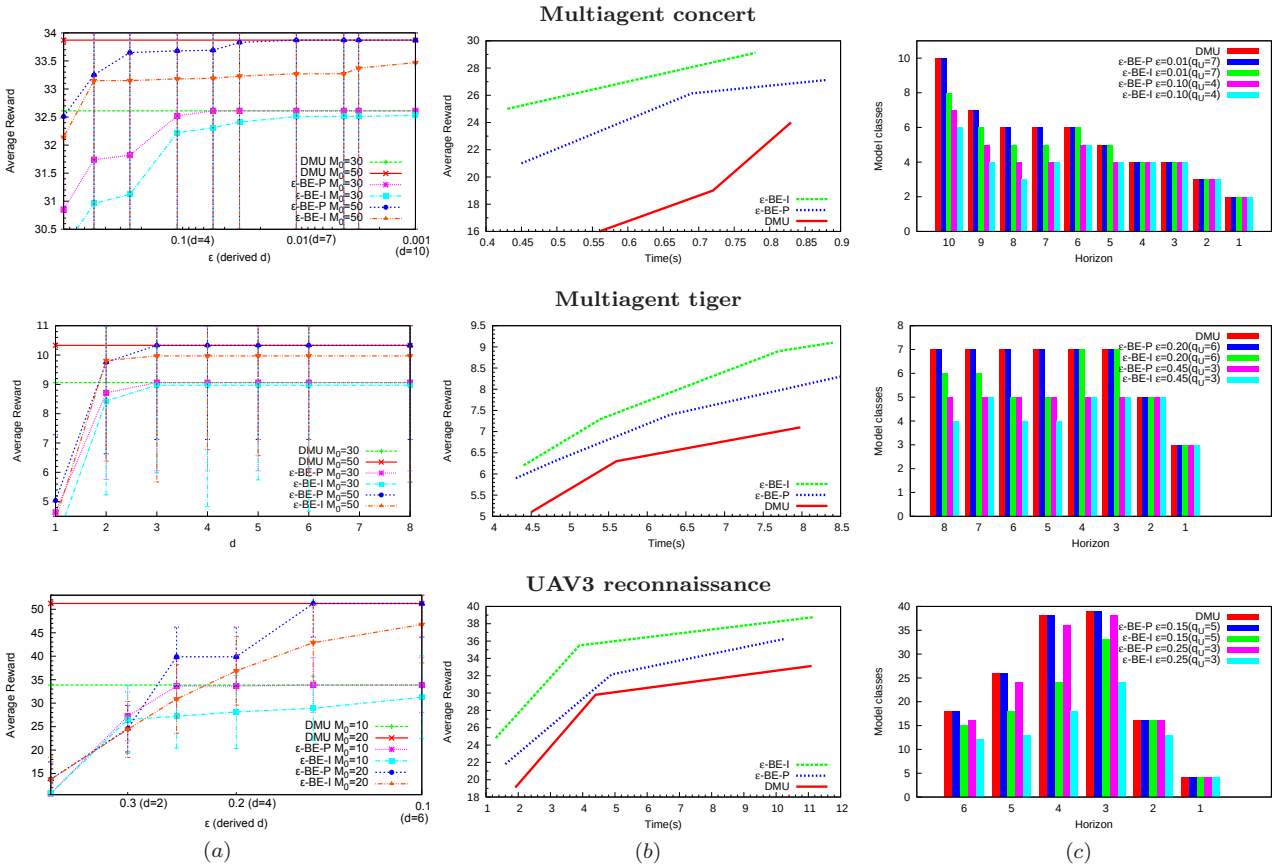


Figure 6: Performance profile obtained by solving level 1 I-DIDs for the different problem domains using ϵ-BE-I, ϵ-BE-P and DMU. (a) Average rewards; (b) Efficiency comparison; and (c) Reduced model space.

5. EXPERIMENTAL RESULTS

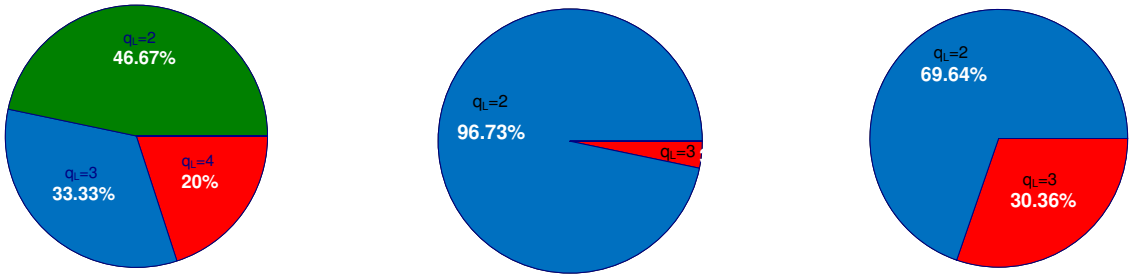
We implemented ϵ-BE-I algorithm for determining (ϵ, q_L, q_U)-BE of models and use it to group models into a class. We then select one representative model for each class while pruning others, similarly to using exact BE. We embed the procedure into the algorithm for solving I-DIDs. We also compare it with ϵ-BE-P algorithm (letting $q{=}q_U{=}q_L$) which serves as a baseline on approximating (ϵ, q_L, q_U)-BE. In addition, we compare both algorithms with one exact BE approach, called discriminative model update (DMU), previously proposed to solve I-DIDs [5]. DMU approach clusters BE models by comparing their entire policy trees and updates only those models that will be behaviorally distinct from existing ones. We evaluate all of these three approaches (namely ϵ-BE-I, ϵ-BE-P and DMU) when they are used to solve level 1 I-DIDs of increasing horizons over four problem domains. Relevant information on domain dimensions and minimal mixing rates are listed in Table 1. Note that UAV5 - an extended version of the two-agent unmanned aerial vehicle (UAV) problem [4, 19] - is the largest domain so far used to evaluate the I-DIDs.

We formulate level 1 I-DIDs of increasing horizons for the problems and solve them using the three approaches. We show that the quality of the policies generated by ϵ-BE-I approaches that of ϵ-BE-P given the same approximation measure. Meanwhile, the solution quality generated by both approximate techniques converges to that of the exact DMU as ϵ decreases (with the corresponding increase in q_U). We also show that in most cases ϵ-BE-I is able to identify approximately BE models without constructing the partial trees of a full size. This verifies the utility of using the incremental technique for the BE identification purpose. In addition, we demonstrate that ϵ-BE-I further reduces the model space and performs better than ϵ-BE-P on the issue of solution scalability.

Domains	γ_F	$\lvert S\rvert$	$\lvert A_i\rvert$	$\lvert A_j\rvert$	$\lvert\Omega_i\rvert$	$\lvert\Omega_j\rvert$
Tiger [9]	0	2	3	3	6	3
UAV3 [4, 19]	0.2	25	5	5	4	5
Concert [19]	0.5	2	3	3	4	2
UAV5	0.2	81	5	5	4	5

Table 1: Domains used to evaluate algorithms for solving I-DIDs.

In Fig. 6(a), we report the average rewards gathered by simulating the I-DID solutions over 1,000 runs. Each run of simulation is executed by randomly picking up the true model of j according to i's belief. We used a horizon of 10 for the *Concert* domain, 8 for the *Tiger* and 6 for the *UAV3*. For a given number of initial models $M_{j,0}$, ϵ-BE-I obtains similar average rewards in comparison to ϵ-BE-P. As expected, their solutions improve and converge toward the exact method DMU as ϵ reduces.



(*a*) **Multiagent concert (T=10)** (*b*) **Multiagent tiger (T=8)** (*c*) **UAV3 reconnaissance (T=6)**

Figure 7: Likelihood that ϵ-BE-I terminates at q_L in the setting of (*a*) q_U=5; (*b*) q_U=6; and (*c*) q_U=4.

Fig. 6(*b*) confirms our intuition on the favorable efficiency of ϵ-BE-I technique. For a given allocated time, ϵ-BE-I obtains larger rewards than other approaches including both DMU and ϵ-BE-P. As we show in Fig. 6(*c*), the number of models in a model node drops when ϵ-BE-I is employed to prune the model space. This is because ϵ-BE-I clusters more approximately BE models using a small set of policies.

In Fig. 7, we show the likelihood that ϵ-BE-I terminates the comparison before reaching the maximum length, q_U, of all paths in the partial policy trees. We compute this likelihood as the percentage of occurrences for each q_L value under a given ϵ. The cases, $q_L < q_U$, are often observed to terminate the policy comparison especially for a small ϵ value (with the corresponding large q_U value).

In Table 2, we show the running times of three techniques for solving problems of increasing horizons. In obtaining the run times for the approximations, we adjusted the corresponding parameters so that the quality of the solution by each approach was similar to each other. ϵ-BE-I achieves the reduced running times and improved scalability over all domains. In particular, for the large UAV5 domain, we were able to solve the I-DIDs for more than 8 time steps.

Level 1	**T**	**Time (s)**		
		DMU	ϵ-BE-I	ϵ-BE-P
Concert	6	0.29	0.11	0.31
	10	2.3	0.22	1.9
	25	*	**9.1**	**13.1**
Tiger	6	0.34	0.16	0.21
	8	1.3	0.21	0.37
	20	*	**2.49**	**3.1**
UAV3	6	13.1	8.1	8.9
	8	161	19	27
	10	*	48	55
	20	*	76	98
	25	*	**132**	*
UAV5	4	19.3	7.9	9.8
	6	*	16	31
	8	*	**60**	*

Table 2: ϵ-BE-I scales better than other approaches. Experiments were run on a Linux platform with Intel Core2 2.4GHz with 4GB of memory.

6. RELATED WORK

I-DIDs [6] emerge as an important framework on modeling multiagent decision making problems. Models for the similar purpose include multiagent influence diagrams (MAIDs) [10], and networks of influence diagrams (NIDs) [7, 8]. These formalisms structure the complex problem domains 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.

As we mentioned before, the complexity of I-DIDs is mainly due to 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, Zeng *et al.* [18] employed a k-means approach to cluster models together and select K representative models in the model node at each time step. This approach needs to expand all models before clustering is applied, which consumes a large amount of memory on storing the models. A recent approach [5] preemptively avoids expanding models that will turn out to be behaviorally equivalent to others in the new time step. By discriminating between model updates, the approach generates a minimal set of models in each non-initial model node. This line of work exploits the concept of BE, introduced earlier [13, 12]. The developed method quickly turns to be inefficient since it requires to maintain and compare the entire policy trees for identifying BE models. In parallel, Zeng *et al.* [15] attempted to cluster models using K most probable paths in the policy tree. However, the proposed technique is facing an unsolved problem on computing path probabilities. Similarly, the attempt using subjectivly equivalent models to cluster models requires the prediction on behavior of the modeling agent [3]. Another efficient way to reduce the model space is achieved by clustering models that are actionally equivalent [16]. Recently,

Zeng and Doshi [17] compare various I-DID solutions and demonstrate their utilities in more problem domains.

7. DISCUSSION

I-DIDs provide a graphical formalism for modeling the sequential decision making of an agent in an uncertain multiagent setting. The increased complexity of I-DIDs is predominantly due to the large space of candidate models and its exponential growth over time. Previous solutions to I-DIDs limit the model growth mainly by clustering BE models at each step. We presented an improved version of using partial policies to identify BE models. We defined approximately BE based on a partial policy tree that has an asymmetric structure and allows different lengths for its paths. Our definition avoids building a full size of the partial trees and clusters more models that are approximately BE. We showed that our new approach gains much computational savings and achieves better scalability over the state of the art approach. As we note that our approach is developed based on the contraction property of problem domains, we may further refine the approach by exploiting the relevant property.

8. ACKNOWLEDGMENT

The authors acknowledge the supports from NSFC (#60974089 and #60975052). Yifeng thanks Dr. Prashant Doshi (in the University of Georgia) for useful comments on the first draft.

9. REFERENCES

[1] X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *The 14th Conference on Uncertainty in Artificial Intelligence(UAI)*, pages 33–42, 1998.

[2] E. Dekel, D. Fudenberg, and S. Morris. Topologies on types. *Theoretical Economics*, 1:275–309, 2006.

[3] P. Doshi, M. Chandrasekaran, and Y. Zeng. Epsilon-subject equivalence of models for interactive dynamic influence diagrams. In *WIC/ACM/IEEE Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pages 165–172, 2010.

[4] P. Doshi and E. Sonu. Gatac: A scalable and realistic testbed for multiagent decision making. In *AAMAS 2010 Workshop on Multi-agent Sequential Decision-Making in Uncertain Domains*, pages 64–68, 2010.

[5] P. Doshi and Y. Zeng. Improved approximation of interactive dynamic influence diagrams using discriminative model updates. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 907–914, 2009.

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

[7] K. Gal and A. Pfeffer. Networks of influence diagrams: A formalism for representing agents' beliefs and decision-making processes. *Journal of Artificial Intelligence Research*, 33:109–147, 2008.

[8] Y. Gal and A. Pfeffer. A language for modeling agent's decision-making processes in games. In *Autonomous Agents and Multi-Agents Systems Conference (AAMAS)*, pages 265–272, 2003.

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

[10] D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1027–1034, 2001.

[11] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 1951.

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

[13] B. Rathnas., P. Doshi, and P. J. Gmytrasiewicz. Exact solutions to interactive pomdps using behavioral equivalence. In *Autonomous Agents and Multi-Agents Systems Conference (AAMAS)*, pages 1025–1032, 2006.

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

[15] Y. Zeng, Y. Chen, and P. Doshi. Approximating behavioral equivalence of models using top-k policy paths. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1229–1230, 2011.

[16] Y. Zeng and P. Doshi. Speeding up exact solutions of interactive influence diagrams using action equivalence. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1996–2001, 2009.

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

[18] Y. Zeng, P. Doshi, and Q. Chen. Approximate solutions of interactive dynamic influence diagrams using model clustering. In *Twenty Second Conference on Artificial Intelligence (AAAI)*, pages 782–787, Vancouver, Canada, 2007.

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