Approximating behavioral equivalence for scaling solutions of I-DIDs

Yifeng Zeng[1,2] · Prashant Doshi[3] · Yingke Chen[5] · Yinghui Pan[4] · Hua Mao[5] · Muthukumaran Chandrasekaran[3]

Abstract Interactive dynamic influence diagram (I-DID) is a recognized graphical framework for sequential multiagent decision making under uncertainty. I-DIDs concisely represent the problem of how an individual agent should act in an uncertain environment shared with others of unknown types. I-DIDs face the challenge of solving a large number of models that are ascribed to other agents. A known method for solving I-DIDs is to group models of other agents that are behaviorally equivalent. Identifying model equivalence requires solving models and comparing their solutions generally represented as policy trees. Because the trees grow exponentially with the number of decision time steps, comparing entire policy trees becomes intractable, thereby limiting the scalability of previous I-DID techniques. In this article, our specific approaches focus on utilizing partial policy trees for comparison and determining the distance between updated beliefs at the leaves of the trees. We propose a principled way to determine how much of the policy trees to consider, which trades off solu-tion quality for efficiency. We further improve on this technique by allowing the partial policy trees to have paths of differing lengths. We evaluate these approaches in multiple problem domains and demonstrate significantly improved scalability over previous approaches.

Keywords Multiagent systems · Decision making · Influence diagrams · Behavioral equivalence

✉ Yifeng Zeng
y.zeng@tees.ac.uk

Yingke Chen
yke.chen@gmail.com

1 Department of Automation, Xiamen University, Xiamen, China

2 School of Computing, Teesside University, Middlesbrough, UK

3 Department of Computer Science, University of Georgia, Athens, GA, USA

4 Department of Information Management, Jiangxi University of Finance and Economics, Nanchang, China

5 Department of Computer Science, Sichuan University, Chengdu, China

1 Introduction

Interactive dynamic influence diagram (I-DID) [17,49] is a probabilistic graphical model for sequential decision making in uncertain multiagent settings. It concisely represents the problem of how an agent should act in an uncertain environment shared with others of unknown types. I-DIDs generalize the standard DIDs [44] to multiagent settings providing a way to model and exploit the embedded structure often present in real-world decision making.

Differing from other frameworks such as decentralized POMDPs [39] and multiagent influence diagrams [23], I-DIDs take the perspective of an individual agent to sequential decision making in multiagent settings and do not assume the common knowledge of beliefs between multiple agents. I-DIDs provide a graphical and naturally factored representation for interactive partially observable Markov decision processes (I-POMDPs) [20]. The individual agent perspective taken by I-DIDs (and I-POMDPs) makes it a general framework applicable in both cooperative and competitive agent settings. For example, the I-DID-based framework has been implemented in emerging applications of automated guided vehicles that communicate [27] in real-time operations. Recently, I-DID is integrated with the belief–desire–intention (BDI) framework [9], which enhances the BDI agent's reasoning and planning capability under uncertainty. It has been coined as one principle way for dealing with ad hoc agent teamwork problems [7].

Sequential decision making in partially observable multiagent settings is a very hard computational problem in general [4]. A solution to the decision-making problem is a sequence of actions and observations over multiple time steps, namely a policy of an agent, which prescribes the agent's actions given specific observations. We aim to find an optimal solution/policy to sequential multiagent decision-making problems. Expectedly, solving I-DIDs tends to be computationally complex as they acutely suffer from both the curses of dimensionality and history [35]. This is because the state space in I-DIDs includes candidate behavioral models of other agents in addition to the traditional physical states. These models could be I-DIDs themselves thereby leading to a nested modeling. As the agents act, observe, and update beliefs, I-DIDs must track the evolution of the models over time. Consequently, I-DIDs suffer not only 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 exacerbated by the nested nature of the space.

Given this complexity, principled methods for solving I-DIDs of that scale are critically needed. Previous work in this regard has mainly exploited *behavioral equivalence (BE)* of models to reduce the dimensionality of the state space [49]. Models that are behaviorally equivalent (BE) [13,36,37] prescribe identical behavior, and these may be grouped because it is the prescriptive aspects of the models and not the descriptive that matter to the decision maker. Essentially, we cluster BE models of other agents and select a representative model for each equivalence class. For example, [16] minimize the model space by updating only those models that lead to behaviorally distinct models at the next time step. While this approach speeds up solutions of I-DIDs considerably, it does not scale desirably to large horizons. This is because: (a) models are compared for BE using their solutions which are policy trees. As the time period of decision making increases, the size of the policy tree increases exponentially; (b) the condition for BE is strict: entire policy trees of two models must match exactly. While this can be done bottom-up, the complexity of this operation depends on the size of the policy tree. Finally (c) the space of models that must be compared with each other grows exponentially over time, as we mentioned previously.

Significant progress could be made by efficiently determining if two models are BE and by grouping models that are *approximately* BE. We expect the latter to result in a fewer number of equivalence classes. Some of these classes contain more models, thereby producing fewer representatives at the cost of prediction error. In this article, we present new approaches that address both these issues. We determine BE between two models more efficiently by comparing their *partial* policy trees and the updated beliefs at the leaves of the policy trees. This leads to significant savings in memory as we do not store entire policy trees. Furthermore, we may group models whose partial policy trees are identical but the updated beliefs at the leaves diverge by small amounts. This defines the first principled approximate measure of BE to the best of our knowledge, which could group more models together.

In order to determine the partialness of the policy trees to compare, we use the insight that the divergence between the updated beliefs at the corresponding leaves of two identical policy trees will not be greater than the divergence between the initial beliefs. [5] show that the change in the divergence is a contraction controlled by a rate parameter, γ. We show how we may calculate γ in our context and use it to obtain the depth of the partial policy tree to use for a given approximate measure of BE.

As we compute a single rate parameter, γ, given a problem domain, the partial policy trees tend to be perfectly depth balanced. In other words, each branch of the tree consisting of an action–observation sequence is of the same length. Motivated by the fact that the rate parameter is a worst-case estimate, we improve the efficiency of our approach by allowing policy trees to have branches of differing lengths. Branches of trees may be limited to a shallower depth if the divergence at their leaves has already fallen within an approximation measure. We present an iterative method for determining the partial policy trees to compare, which need not be depth balanced, thereby further reducing the memory required to store the trees. On the other hand, we may expend additional time in computing multiple updated beliefs. This approach also allows us to group more models in a class because it relaxes the previous approximation, possibly resulting in a larger error.

We evaluate the empirical performances of these approaches on multiple problem domains within the framework of I-DIDs and demonstrate that they allow us to scale the solution of I-DIDs, particularly in the computing time, significantly more than previous techniques [49]. Additionally, we experiment with large problem domains, one of which pertains to countering money laundering as introduced by [29] and the other uses a multiagent simulation testbed called the *Georgia testbed for autonomous control of vehicles (GaTAC)* [15], which facilitates scalable and realistic problem domains pertaining to autonomous control of unmanned agents such as uninhabited aerial vehicles.

We list our main contributions as follows.

1. We present the first principled formulation of approximate BE that allows for larger clusters of models of the other agents resulting in solutions of I-DIDs that scale significantly better.
2. As full policy trees grow exponentially with longer look ahead, we present an approach that compares partial policy trees, which are obtained by solving the models. To further improve the efficiency of the method, we introduce a technique for incrementally comparing policy trees in order to identify models that are approximate BE.
3. We theoretically analyze the reduced computational complexity due to the approximation. More importantly, we evaluate the performance of the algorithms on multiple problems and demonstrate the scalability in two large pragmatic domains.

We organize the article as follows: We briefly review the concept of BE and the I-DID framework in Sect. 2. In order to facilitate a conceptualization of an approximate BE, we

develop a new technique to identify exact BE in Sect. 3, followed by a novel formulation of ϵ-BE in Sect. 4. We may avoid fully comparing partial policy trees as we show in Sect. 5 leading to a further approximation. The computational savings and associated error are theoretically analyzed in Sect. 6. We empirically analyze the impact of these approaches on the quality and scalability of the solution including experimentation on a scalable UAV simulation testbed in Sect. 7. We discuss related frameworks and models in Sect. 8 particularly focusing on other ID-based frameworks for multiagent settings. Finally, we conclude this article with a discussion of the limitations and future work in Sect. 9.

2 Background: interactive DID and behavioral equivalence

We begin with a brief overview of interactive DIDs and then proceed to describe the usage of BE for solving I-DIDs.

2.1 Interactive dynamic influence diagrams

Influence diagrams (IDs) [21,41] are probabilistic graphical models well suited to representing an agent's decision-making problem in an uncertain environment. They typically utilize chance nodes which model the uncertain aspects of the problem through random variables such as those for modeling the physical state, S, and the agent's observations, O_i. IDs additionally use decision nodes that model the agent's actions, A_i, and utility nodes that model the agent's reward function, R_i. Usually, action(s) with the largest expected utility is selected on evaluating the ID for each possible setting of the decision node.

Interactive IDs (I-IDs) [17] generalize the formalism of IDs to model decision making in multiagent settings. In addition to the nodes found in an ID, the I-ID for an agent i includes a new type of node called the *model node*. The model node contains as its values the alternative computational models ascribed by i to the other agent. We denote it as the hexagonal node, $M_{j,l-1}$, in Fig. 1, where j denotes the other agent. Subscript $l-1$ is the *strategy level*, which emphasizes the capability for a nested modeling of i by the other agent j. Agent j's level is one less than that of i, which is consistent with previous hierarchical formalizations in game theory [1,3] and decision theory [20]. A basis level 0 model is then an ID or a flat probability distribution. The *interactive state space* consists of the chances nodes comprising S together with the model node. Denote the set of models considered in the model node by $\mathcal{M}_{j,l-1}$, and an individual 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, which is a probability distribution over j's interactive state space, and $\hat{\theta}_j$ is the agent's *frame* encompassing the decision, observation, and utility nodes. 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 flat probability distribution over the actions.

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*, between the model and chance nodes. This link denotes that the distribution over A_j depends on the model selected in the model node.

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. 2a transforming the I-ID to a flat ID shown in Fig. 2b. In particular, the decision node of each level $l-1$ I-ID or level 0 ID is mapped into a chance node. Specifically, if $\mathsf{OPT}(m^1_{j,l-1})$ is the set of optimal actions obtained by solving the I-ID (or ID) denoted by $m^1_{j,l-1}$, then the corresponding distribution over the mapped chance

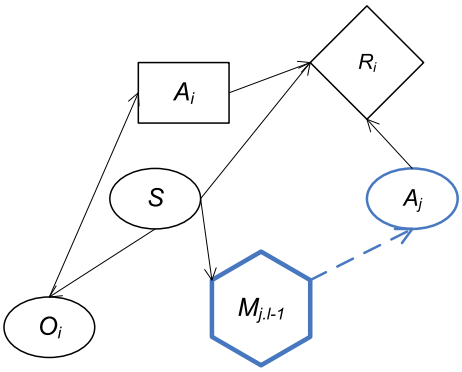


Fig. 1 A generic level $l > 0$ I-ID for agent i situated with one other agent j. The highlighted hexagon in *blue* is the model node, $M_{j,l-1}$, and the *dashed arrow* is the policy link to other's actions (color figure online)

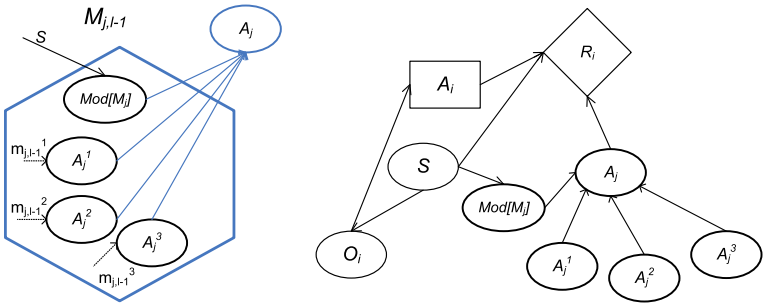


Fig. 2 **a** The model node and policy link in Fig. 1 may be represented using chance nodes and dependencies between the nodes. The decision nodes of the lower-level I-IDs or IDs ($m^1_{j,l-1}$, $m^2_{j,l-1}$, $m^3_{j,l-1}$ where the superscript numbers serve to distinguish the models) are mapped to the corresponding chance nodes (A^1_j, A^2_j, A^3_j), respectively, which is indicated by the *dotted arrows*. Depending on the value of node, $Mod[M_j]$, distribution of each of the action chance nodes is assigned to node A_j with some probability. **b** The I-ID of Fig. 1 transforms into the flat ID with the model node and policy link replaced as in **a**

node, A^1_j, is: $Pr(a_j \in A^1_j) = \frac{1}{|\text{OPT}(m^1_{j,l-1})|}$ if $a_j \in \text{OPT}(m^1_{j,l-1})$, 0 otherwise. The different chance nodes (A^1_j, A^2_j, A^3_j)—one for each model—and additionally, the chance node labeled $Mod[M_j]$ form the parents of the chance node, A_j. There are as many action nodes as the number of models in the support of agent i's beliefs. The distribution over $Mod[M_j]$ is i's belief over j's models given the state. The conditional probability table (CPT) of the chance node, A_j, is a *multiplexer* that selects the distribution of each of the action nodes (A^1_j, A^2_j, A^3_j) depending on the value of the selector, $Mod[M_j]$. 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; A_j assumes the distribution of A^2_j when $Mod[M_j]$ selects the value $m^2_{j,l-1}$, and analogously for $m^3_{j,l-1}$. 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.

Zeng and Doshi [49] illustrate the formalism in the context of the multiagent tiger problem [20]—a two-agent generalization of the well-known single-agent tiger problem [22]. We include it here as well in order to promote understanding of the framework and for completeness. The multiagent tiger problem also forms a running example throughout this article.

This problem domain involves two closed doors, one of which hides a tiger and the other hides a pot of gold, and two agents, i and j, which face the closed doors. Each agent may open either the left door (action denoted by OL), or the right door (OR), or listen (L). On listening, an agent may hear the tiger growling either from behind the left door (observation denoted by GL) or from behind the right door (GR). Additionally, the agent hears creaks emanating from the direction of the door that was possibly opened by the other agent. This includes creak from the left (CL), creak from the right (CR), or silence (S) if no door was opened. All observations are assumed to be noisy. If any door is opened by an agent, the tiger appears behind any of the two doors randomly in the next time step. An agent gets rewarded for opening the door that hides the gold but gets penalized for opening the door hiding the tiger. While the actions of the other agent do not directly affect the reward for an agent, they may potentially change the location of the tiger. This multiagent formulation of the problem differs from other formulations of this problem such as that of [28] in the presence of door creaks and that it is not cooperative.

We set up the I-ID for the multiagent tiger problem in Fig. 3. We discuss the CPTs of the various nodes in "Multiagent tiger problem" section of Appendix. While the I-ID illustrates two models of j, in practice there would be as many action nodes of j if there were more models.

I-DIDs generalize I-IDs to allow sequential decision making over multiple time steps. In addition to the model nodes and the dashed policy link, an I-DID differs from a traditional DID in its use of the *model update link* shown as a dotted arrow in Fig. 4. We briefly explain the semantics of the model update next.

Agents in a multiagent setting typically act and make observations, which changes their beliefs. Therefore, 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

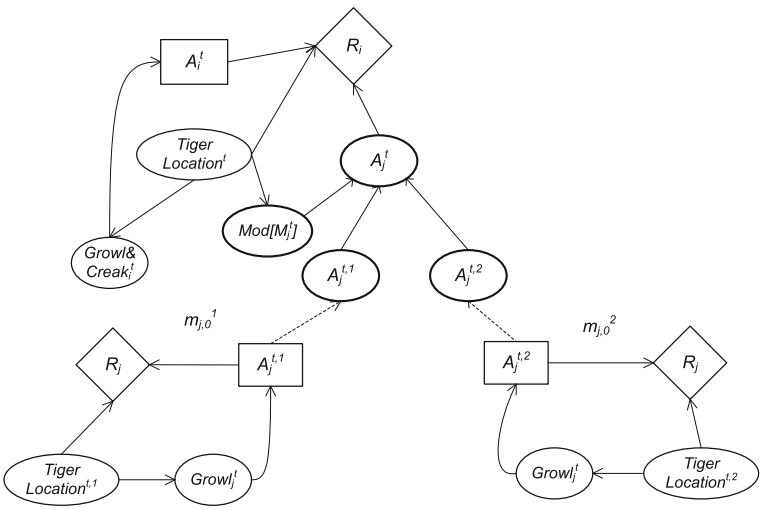


Fig. 3 Level 1 I-ID of i for the multiagent tiger problem. Solutions of two level 0 models (IDs) of j map to the chance nodes, $A_j^{t,1}$ and $A_j^{t,2}$, respectively (illustrated using *dotted arrows*), transforming the I-ID into a flat ID. The two models differ in the distribution over the chance node, TigerLocation^t

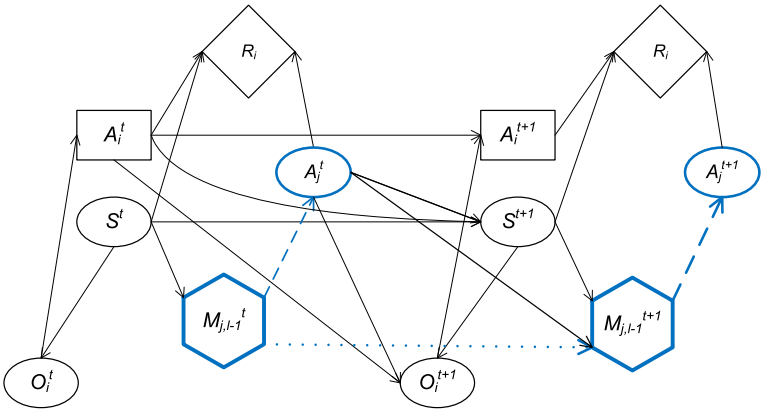


Fig. 4 A generic two time-slice level l I-DID for agent i. The *dotted arrow* is the model update link that denotes the update of the models of j and of the distribution over the models as both agents act and observe, over time

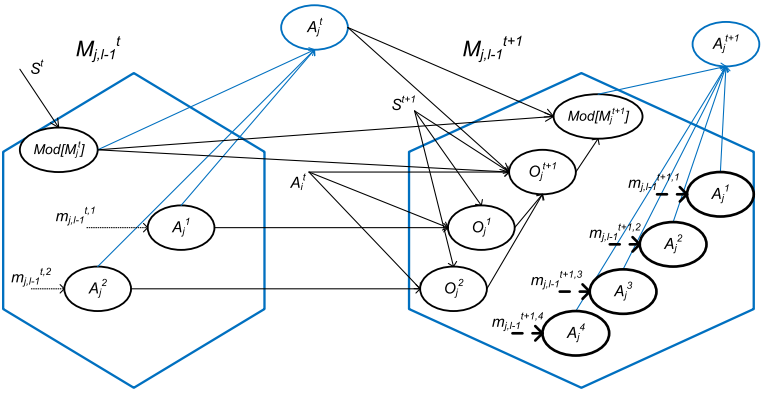


Fig. 5 The model update link may be represented using chance nodes and dependency links between them. Consider two models in the model node at time t. These grow *exponentially* in number to more models in the model node at $t+1$ as shown in *bold* (*superscript* numbers distinguish the different models). Models at $t+1$ reflect the updated beliefs of j, and their solutions provide the probability distributions for the corresponding action nodes

the actions, and the agent may receive any one of $|\Omega_j|$ possible observations where Ω_j is the set of j's 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 with a nonzero probability in the distribution over $Mod[M_{j,l-1}^t]$, $|A_j|$ and $|\Omega_j|$ are the largest spaces of actions and observations, respectively, among all the models.

The CPT of chance node, $Mod[M_{j,l-1}^{t+1}]$, encodes the indicator 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 a 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. 5, thereby transforming the I-DID into a flat DID. We show

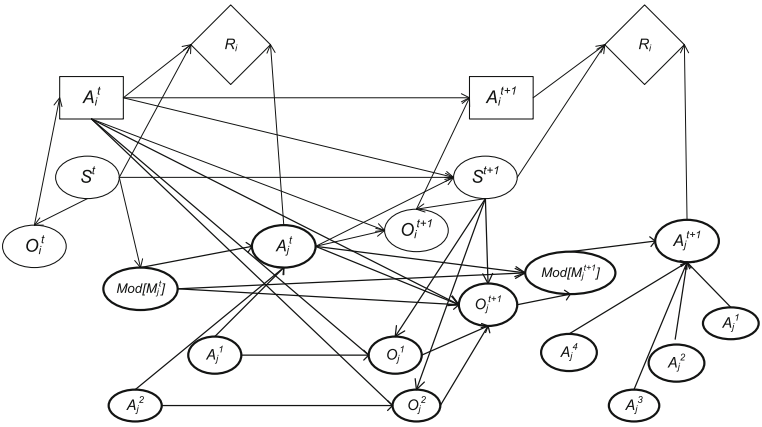


Fig. 6 A flat DID obtained by replacing the model nodes and model update link in the I-DID of Fig. 4 with the chance nodes and the relationships (in *bold*) as shown in Fig. 5. The lower-level models are solved to obtain the distributions for the action chance nodes

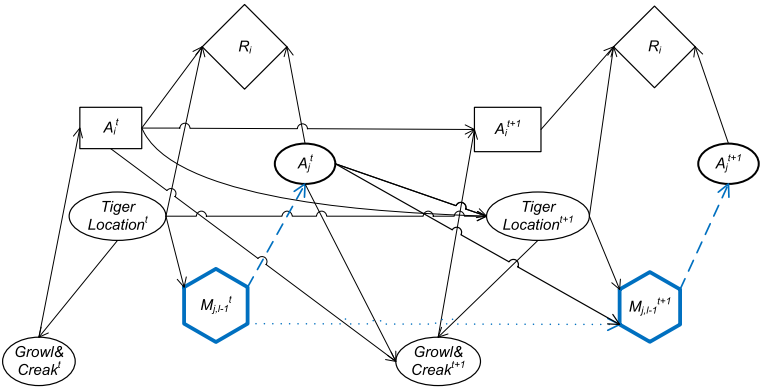


Fig. 7 Two time-slice level l I-DID of i for the multiagent tiger problem. Highlighted model nodes in *blue* contain the different models of j (color figure online)

the two time-slice flat DID with the model nodes and the model update link replaced by the chance nodes and the relationships between them, in Fig. 6. Chance nodes and dependency links not in bold are standard, usually found in single-agent DIDs.

We illustrate the two time-slice I-DID for the multiagent tiger problem in Fig. 7. The model update link not only updates the number of j's candidate models due to j's action and observations of growl and creak, but also updates the probability distribution over these models. In Fig. 8, we illustrate the update of a single model of j contained in the model node at time t over time.

2.2 Model solution using behavioral equivalence

As we mentioned above, the complexity on solving I-DID is due to the exponential growth number in candidate models of other agents over time. To reduce the model space, we may group together models whose solutions are identical into a single equivalence class and select a representative model from each class. As [36] note, it is the *behavior* of the other

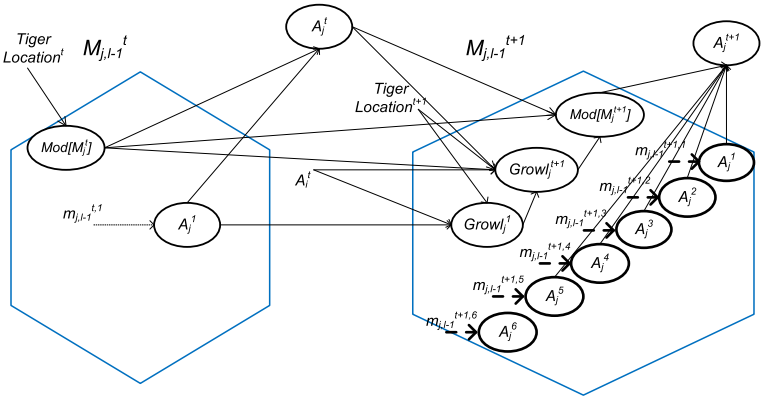


Fig. 8 Because agent j in the tiger problem may receive any one of the six possible observations given the action prescribed by its model, a single model in the model node at time t could lead to six distinct models at time $t+1$

agents, which impact the decision maker's actions regardless of how the agents are internally modeled. We define BE more formally below:

Definition 1 *(BE).* Two models, m_j and $\hat{m}_j$, of an agent, j, are behaviorally equivalent if $\mathsf{OPT}(m_j) = \mathsf{OPT}(\hat{m}_j)$, where $\mathsf{OPT}(\cdot)$ denotes the solution of the model.

Thus, BE models are those whose behavioral predictions for the agent are identical.

Models that are determined to be exactly BE by comparing their policy trees for equality are grouped, and a single representative is selected from each equivalence class, thereby reducing the set of candidate models, $\mathcal{M}_{j,l-1}$, in the model node to a *behaviorally minimal* subset. We may define this subset as the largest subset of $\mathcal{M}_{j,l-1}$ such that no two models in it are BE. Of course, the probability mass on each individual model in a class is summed and assigned to the representative.

$$\hat{b}_{i,l}(\hat{m}_{j,l-1}|s) = \sum_{m_{j,l-1}} b_{i,l}(m_{j,l-1}|s) \tag{1}$$

where $m_{j,l-1}$ is a model that is BE to $\hat{m}_{j,l-1}$.

The solution of an I-DID (and I-ID) is implemented recursively down the levels as shown in Fig. 9. In order to solve a level 1 I-DID of horizon T, we start by solving the base level 0 models of the other agent, which may be traditional DIDs of horizon T. Their solutions provide probability distributions over the other agent's actions, which are entered in the corresponding action nodes found in the model node of the level 1 I-DID at the corresponding time step (lines 3–5). Subsequently, the set of j's models is minimized by excluding the BE models (line 6).

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 [38]. Because agent i has a belief over j's models as well, the look-ahead technique 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 must be solved in the first time step up to horizon T 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

I-DID EXACT (level $l \geq 1$ I-DID or level 0 DID, horizon T)

Expansion Phase

1. **For** t **from** 0 **to** $T-1$ **do**
2. **If** $l \geq 1$ **then**
   Minimize $M^t_{j,l-1}$
3. **For each** m^t_j **in** $\mathcal{M}^t_{j,l-1}$ **do**
4. Recursively call algorithm with the $l-1$ I-DID (or DID) that represents m^t_j and the horizon, $T-t$
5. Map the decision node of solved I-DID (or DID), $\mathsf{OPT}(m^t_j)$, to the corresponding chance node A_j
6. $\mathcal{M}^t_{j,l-1} \leftarrow$ **PruneBEModels**($\mathcal{M}^t_{j,l-1}$)
7. **If** $t < T-1$ **then**
   Populate $M^{t+1}_{j,l-1}$
8. **For each** m^t_j **in** $\mathcal{M}^t_{j,l-1}$ **do**
9. **For each** a_j **in** $\mathsf{OPT}(m^t_j)$ **do**
10. **For each** o_j **in** O_j (part of m^t_j) **do**
11. Update j's belief, $b^{t+1}_j \leftarrow SE(b^t_j, a_j, o_j)$
12. $m^{t+1}_j \leftarrow$ New I-DID (or DID) with b^{t+1}_j as initial belief
13. $\mathcal{M}^{t+1}_{j,l-1} \overset{\cup}{\leftarrow} \{m^{t+1}_j\}$
14. Add the model node, $M^{t+1}_{j,l-1}$, and the model update link between $M^t_{j,l-1}$ and $M^{t+1}_{j,l-1}$
15. Add the chance, decision, and utility nodes for $t+1$ time slice and the dependency links between them
16. Establish the CPTs for each chance node and utility node

Solution Phase

17. **If** $l \geq 1$ **then**
18. Represent the model nodes, policy links and the model update links as in Fig. 5 to obtain the DID
19. Apply the standard look-ahead and backup method to solve the expanded DID (other solution approaches may also be used)

Fig. 9 Algorithm for exactly solving a level $l \geq 1$ I-DID or level 0 DID expanded over T time steps in a two-agent setting

could describe the behavior of j. $SE(b^t_j, a_j, o_j)$ is an abbreviation for the belief update (lines 8–13). Beliefs over this updated set of candidate models are calculated using the standard inference methods through the dependency links between the model nodes shown in Fig. 5 (lines 15–18). Agent i's I-DID is expanded across all time steps in this manner. Because I-DIDs are transformed into flat DIDs, we point out that the algorithm in Fig. 9 may be realized with the help of standard implementations of DIDs such as HUGIN EXPERT [2] and NETICA [46]. The solution is a *policy tree* that prescribes the optimal action(s) to perform for agent i initially given its belief, and the actions thereafter conditional on its observations up to time T.

3 Approximate behavioral equivalence

Although BE represents an effective exact criteria to group models, identifying BE models requires us to compare the entire solutions of models. Because the solutions are policy trees, all paths in the trees must be compared, which grow exponentially over time. This is further complicated by the number of candidate models of the other agents in the model node growing exponentially over time. In order to scale approaches utilizing BE, we seek to do the following:

1. Reduce the complexity of identifying BE by comparing partial policy trees; and
2. Group together more models that could be approximately BE resulting in fewer representatives and a smaller set of candidate models.

For the sake of clarity, we assume that the models of the other agent j have identical frames (possibly different from i's) and differ only in their beliefs. We discuss the implications of this assumption for our approach in Sect. 9. We focus on the general setting where a model, $m_{j,l-1}$, is itself a DID or an I-DID, in which case its solution could be represented as a policy tree. We denote the policy tree of horizon, T, as $\pi^T_{m_{j,l-1}}$ (also called a depth-T policy tree) in which all action-observation paths are of length T; therefore, $OPT(m_{j,l-1}) \triangleq \pi^T_{m_{j,l-1}}$, where $OPT(\cdot)$ denotes the solution of the model. We illustrate example policy trees below:

Example 1 (*Policy trees*) For the multiagent tiger problem introduced previously in Sect. 2.1, consider two level 0 models of agent j, $m^1_{j,0}$ and $m^2_{j,0}$, included in the initial model node of agent i's I-DID shown in Fig. 7, expanded to $T = 3$ time steps. Let $m^1_{j,0} = \langle 0.15, \hat{\theta}_j \rangle$, where 0.15 is agent j's belief that the tiger is behind the left door (TL) and $\hat{\theta}_j$ refers to the frame consisting of the chance nodes and the dependency links that constitute its DID. Let $m^2_{j,0} = \langle 0.5, \hat{\theta}_j \rangle$. We show the policy trees obtained by solving the two models below.

Recall from Definition 1 that two models of j are BE if they produce identical behaviors for j. Formally, models $m_{j,l-1}, \hat{m}_{j,l-1} \in \mathcal{M}_{j,l-1}$ are BE if $\pi^T_{m_{j,l-1}} = \pi^T_{\hat{m}_{j,l-1}}$.

Each path in the policy tree from the root to the leaf is an action–observation sequence denoted by $h^{T-1}_j = \{a^t_j, o^{t+1}_j\}^{T-1}_{t=0}$, where o^T_j is null. For example, the leftmost path in the policy tree in Fig. 10a is $h^2_j = \{\text{L}, \text{GR}, \text{L}, \text{GR}, \text{OL}\}$. If $a^t_j \in A_j$ and $o^{t+1}_j \in \Omega_j$, where A_j

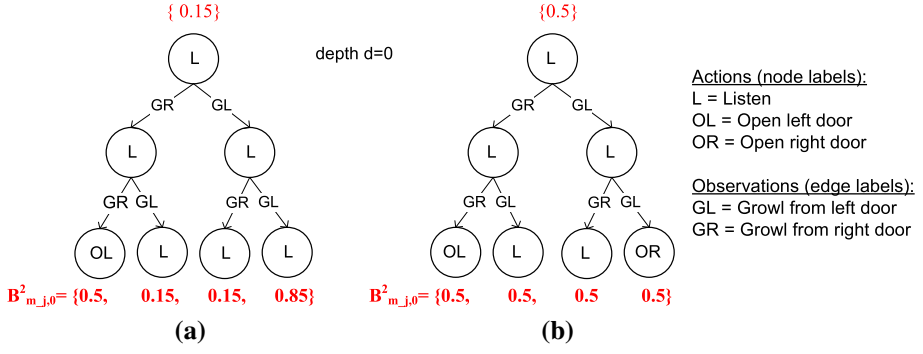


Fig. 10 Horizon, $T = 3$, policy trees obtained by solving, **a** $m^1_{j,0} = \langle 0.15, \hat{\theta}_j \rangle$ and **b** $m^2_{j,0} = \langle 0.5, \hat{\theta}_j \rangle$. The depth of these trees is $2 (= T - 1)$ with the root nodes at depth 0. We also show the sets of updated beliefs at the leaves for the two policy trees

and Ω_j are agent j's action and observation sets, respectively, then the set of all $T-1$-length paths is $H_j^{T-1} = A_j \times \Pi_1^{T-1}(\Omega_j \times A_j)$.

Without loss of generality, we may impose an ordering on a policy tree by assuming some order for the observations, which guard the arcs in the tree. An example ordering of observations may be GR followed by GL in the policy trees in Fig. 10. Furthermore, if $b^0_{j,l-1}$ is the initial belief in the model, $m_{j,l-1}$, then let $b^d_{j,l-1}$ be the belief on updating $b^0_{j,l-1}$ using the action-observation path of length d, h^d_j. Let $B^d_{m_{j,l-1}}$ be the ordered set of beliefs that obtain on updating the initial belief using all d-length paths in the ordered policy tree of model, $m_{j,l-1}$. Therefore, a belief in $B^d_{m_{j,l-1}}$ has an index, k, such that $k \leq |\Omega_j|^d$. These are the updated beliefs at the leaves of the ordered policy tree. For illustration, we show the sets of updated beliefs, $B^2_{m^1_{j,l-1}}$ and $B^2_{m^2_{j,l-1}}$, at the leaves of the two ordered policy trees, in Fig. 10. Finally, let $D_{KL}[p||q]$ denote the Kullback–Leibler (KL) divergence [11] or relative entropy between probability distributions, p and q.

We may identify BE between models by comparing partial policy trees such as depth-d policy trees, all of whose action-observation paths are of length d. Proposition 1 formally presents the result.

Proposition 1 (Revisiting BE) *Given two models of agent j, $m_{j,l-1}$ is BE to $\hat{m}_{j,l-1}$ if their depth-d policy trees, $d \leq T-1$, are identical, $\pi^d_{m_{j,l-1}} = \pi^d_{\hat{m}_{j,l-1}}$, and if $d < T-1$ then beliefs at the leaves of the two ordered policy trees do not diverge: $D_{KL}[b^{d,k}_{m_{j,l-1}}||b^{d,k}_{\hat{m}_{j,l-1}}] = 0$ $\forall k = 1 \ldots |\Omega_j|^d$, where $b^{d,k}_{m_{j,l-1}} \in B^d_{m_{j,l-1}}$, $b^{d,k}_{\hat{m}_{j,l-1}} \in B^d_{\hat{m}_{j,l-1}}$.*

Proof Proposition 1 holds because of the well-known fact that beliefs updated using an action–observation sequence in a partially observable Markov process in a single-agent [42] or a multiagent setting [20] is a sufficient statistic for the history of actions and observations. Consequently, future behavior is predicated on the beliefs only. Furthermore, $D_{KL}[b^{d,k}_{m_{j,l-1}}||b^{d,k}_{\hat{m}_{j,l-1}}] = 0$ if and only if the two distributions are equal. Therefore, pairs of models that satisfy the two conditions in Proposition 1 for some $d < T-1$ will necessarily conform to Definition 1. If $d = T-1$, then Proposition 1 requires the two policy trees to be completely identical, as in Definition 1. □

We point out that Proposition 1 is not particularly sensitive to the measure of divergence between distributions that we utilize. While the KL divergence between two distributions is appropriate because it is zero if and only if the two distributions are equal, the same is also true for, say, the L_1 distance. However, KL divergence has some desirable properties lacked by other norms, which we will exploit later.

Notice that the technique on identifying BE through Proposition 1 produces the same grouping of BE models as previously for the case $d = T-1$ because it collapses into Definition 1. For the case of $d < T-1$, it may group fewer models in a BE class because belief sets that do diverge could still result in an identical pair of complete policy trees. Hence, the new technique may lead to more BE classes than the minimal number.

The advantage offered by Proposition 1 is that we may elegantly generalize it to the notion of approximate BE:

Definition 2 ((ϵ, d)-*BE*) Given two models of agent j, $m_{j,l-1}$ is (ϵ,d)-BE to $\hat{m}_{j,l-1}$ if their depth-d policy trees are identical, $\pi^d_{m_{j,l-1}} = \pi^d_{\hat{m}_{j,l-1}}$, and beliefs at the leaves of the two ordered policy trees diverge by at most ϵ: $max_{k=1\ldots|\Omega_j|^d}\ D_{KL}[b^{d,k}_{m_{j,l-1}}||b^{d,k}_{\hat{m}_{j,l-1}}] \leq \epsilon$, for $\epsilon \geq 0$ and $d < T-1$.

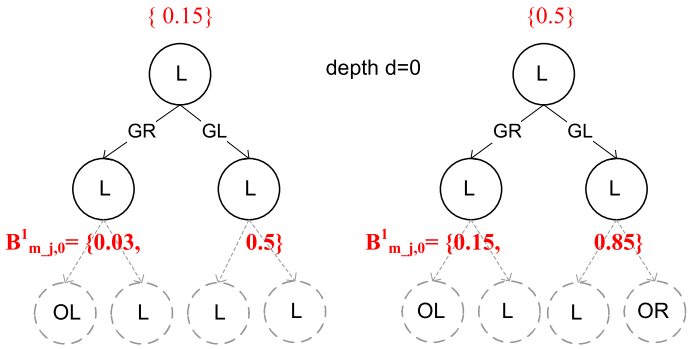


Fig. 11 Partial policy trees of depth 1 are identical, though the complete policy trees are not. The maximum KL divergence between the corresponding beliefs at the leaves is 0.33. Therefore, models $m^1_{j,0}$ and $m^2_{j,0}$ are (0.33, 1)-BE

Intuitively, two models are (ϵ, d)-BE if their solutions share an identical depth-d policy tree and the divergence of pairs of the ordered beliefs at the leaves of the depth-d trees is not larger than ϵ. As ϵ approaches zero, (ϵ, d)-BE converges to Proposition 1. We illustrate the notion of (ϵ, d)-BE using an example.

Example 2 (*(ϵ, d)-BE policy trees*) Consider partial policy trees of depth, $d = 1$, extracted from the policy trees in Fig. 10. We show these in Fig. 11 along with the sets of updated beliefs at the leaves, $B^1_{m^1_{j,0}}$ and $B^1_{m^2_{j,0}}$. Notice that the partial policy trees are identical. The KL divergence between the corresponding beliefs in sets $B^1_{m^1_{j,0}}$ and $B^1_{m^2_{j,0}}$ is $D_{KL}[\langle 0.03, 0.97\rangle || \langle 0.15, 0.85\rangle] = 0.08$ and $D_{KL}[\langle 0.5, 0.5\rangle || \langle 0.85, 0.15\rangle] = 0.33$, and the maximum KL divergence is 0.33. Therefore, models $m^1_{j,0}$ and $m^2_{j,0}$ are (0.33, 1)-BE.

While Definition 2 above is parameterized by the depth d of the policy trees and the measure of approximation, ϵ, we show in the next section that d may be determined given some ϵ.

4 Depth of the partial policy

Definition 2 introduces a measure of approximate BE between two models. It is parameterized by both the amount of approximation, ϵ, and the partialness of the comparison, d. However, we show that the depth d may be uniquely determined by the amount of approximation that is allowed in the equivalence of two models. We begin by reviewing an important result for a Markov stochastic process.

Consider a discrete Markov stochastic process with a general state space, $S = \{s_1, s_2, \ldots, s_n\}$, and transition function, Q. Let p^t and $\hat{p}^t$ be two arbitrary distributions over the state space. It is well known that a stochastic transition never increases the KL divergence between two distributions such as p^t and $\hat{p}^t$ over the same state space in a Markov stochastic process (e.g., see §4.4 in [11]). [5] take it a step further and show that the KL divergence between the distributions contracts at a geometric rate with time given a stochastic transition, and the rate of contraction is based on a *mixing rate*, γ_Q:

Lemma 1 (Contraction—[5]) *For a discrete Markov stochastic process with transition function, Q, and arbitrary distributions over the state space, p^t, $\hat{p}^t$, prior to a transition, and p^{t+1}, $\hat{p}^{t+1}$ post-transition:*

$$D_{KL}(p^{t+1}||\hat{p}^{t+1}) \leq (1-\gamma_Q) D_{KL}(p^t||\hat{p}^t)$$

where

$$\gamma_Q = \min_{s_1,s_2} \sum_{k=1}^{n} min(Q(s_k|s_1),\, Q(s_k|s_2))$$

Mixing rate, γ_Q, represents the minimal amount by which posterior distributions agree with each other after one transition.

In our context, we may apply Lemma 1 to bound the divergence between the beliefs of two models updated using an action–observation sequence. Let $F_{a_j,o_j}(s'|s)$ be the "stochastic transition" from state s to s' obtained by multiplying the state transition probability due to action, a_j, with the likelihood of observation, o_j, for j. If the models are level 0 DIDs,

$$F_{a_j,o_j}(s'|s) = \alpha\ Pr(o_j|s', a_j)\ Pr(s'|s, a_j) \tag{2}$$

where α is the normalization constant. $\gamma_{F_{a_j,o_j}}$ is the minimum fraction of the probability mass for which the updated beliefs of the two models agree, due to the transition, and is the minimal mixing rate:

$$\gamma_{F_{a_j,o_j}} = \min_{s_1,s_2} \sum_{s' \in S} min\{F_{a_j,o_j}(s'|s_1),\, F_{a_j,o_j}(s'|s_2)\} \tag{3}$$

Because F_{a_j,o_j} depends on the state transition probabilities due to action and the observation probabilities, the mixing rate is a property of the problem domain.

If the strategy level $l > 1$, the stochastic transition, $F_{a_j,o_j}(\cdot|\cdot)$, is over the interactive states of j where each such state is a pair of a physical state and model of i; Eq. 2 becomes:

$$F_{a_j,o_j}(is'|is) = \alpha \sum_{a_i \in A_j} Pr(a_i|m_{i,l-1})\ Pr(o_j|s', a_i, a_j)\ Pr(s'|s, a_i, a_j)$$
$$\sum_{o_i \in \Omega_i} Pr(o_i|s', a_i, a_j)\ \tau(b_{i,l-1}, a_i, o_i, b'_{i,l-1})$$

Each of the terms above may be obtained from the CPTs of the nodes in j's I-DID. For this case, we modify Eq. 3 as well by utilizing the transition function above in place of $F_{a_j,o_j}(s'|s)$.

Consequently, the proposition presented next shows how a pair of models may come closer after an action–observation sequence:

Proposition 2 (Contraction of model divergence) *Divergence between initial beliefs, $b^{0,k}_{m_{j,l-1}}$ and $b^{0,k}_{\hat{m}_{j,l-1}}$ in models $m_{j,l-1}$ and $\hat{m}_{j,l-1}$, respectively, reduces on performing action, a_j, and observing, o_j, as:*

$$D_{KL}(b^{1,k}_{m_{j,l-1}}||b^{1,k}_{\hat{m}_{j,l-1}}) \leq (1-\gamma_{F_{a_j,o_j}}) D_{KL}(b^{0,k}_{m_{j,l-1}}||b^{0,k}_{\hat{m}_{j,l-1}}) \tag{4}$$

where $\gamma_{F_{a_j,o_j}}$ is as defined in Eq. 3.

Next, we may iteratively apply Eq. 4 over an action–observation sequence of length d that corresponds to a path in a depth-d policy tree resulting in a geometric contraction of the initial KL divergence:

$$D_{KL}(b^{d,k}_{m_{j,l-1}}||b^{d,k}_{\hat{m}_{j,l-1}}) \leq (1-\gamma_F)^d D_{KL}(b^{0,k}_{m_{j,l-1}}||b^{0,k}_{\hat{m}_{j,l-1}}) \tag{5}$$

Here, because a path may involve different sequences of actions and observations,

$$\gamma_F = \min_{(a_j,o_j) \in A_j \times \Omega_j} \gamma_{F_{a_j,o_j}} \tag{6}$$

The definition of approximate BE in the previous section (Definition 2) limits the maximum divergence between any pair of beliefs at the leaves of the partial policy trees to at most ϵ. Because Eq. 5 bounds this divergence as well, we may equate the bound to ϵ and obtain the equation below.

$$(1-\gamma_F)^d D_{KL}(b^{0,k}_{m_{j,l-1}} || b^{0,k}_{\hat{m}_{j,l-1}}) = \epsilon \tag{7}$$

In the above equation, the only unknown is d because γ_F may be obtained as shown previously, ϵ is given, and $b^{0,k}_{m_{j,l-1}}$, $b^{0,k}_{\hat{m}_{j,l-1}}$ are the given initial beliefs in the models under comparison. We further drive d by applying logarithm on both sides of Eq. 7 and get the following:

$$\begin{aligned} & d\ ln(1-\gamma_F) + ln\ D_{KL}(b^{0,k}_{m_{j,l-1}} || b^{0,k}_{\hat{m}_{j,l-1}}) = ln\ \epsilon \\ \rightarrow\ & d\ ln(1-\gamma_F) = ln\ \epsilon - ln\ D_{KL}(b^{0,k}_{m_{j,l-1}} || b^{0,k}_{\hat{m}_{j,l-1}}) \\ \rightarrow\ & d = \frac{ln\ \epsilon - ln\ D_{KL}(b^{0,k}_{m_{j,l-1}} || b^{0,k}_{\hat{m}_{j,l-1}})}{ln(1-\gamma_F)} \end{aligned}$$

Since the depth d shall be a nonnegative integer not larger than the entire planning horizon T, Proposition 3 derives d for a given value of ϵ.

Proposition 3 (Depth of comparison). *For any $b^{0,k}_{m_{j,l-1}}$, $b^{0,k}_{\hat{m}_{j,l-1}}$, T, computed γ_F and a given ϵ as defined above, we may obtain the depth d as:*

$$d = min\left\{T-1, max\left\{0, \lfloor \frac{ln\ \epsilon - ln\ D_{KL}\left(b^{0,k}_{m_{j,l-1}} || b^{0,k}_{\hat{m}_{j,l-1}}\right)}{ln(1-\gamma_F)} \rfloor \right\}\right\} \tag{8}$$

where $\lfloor \cdot \rfloor$ gives the floor value, $\gamma_F \in (0,1)$, and $\epsilon > 0$.

Proposition 3 gives the smallest depth that we could use for comparing the policy trees. In general, as ϵ increases, d reduces for a model pair until it becomes zero when we compare just the initial beliefs in the models. Conversely, as ϵ reduces and we tolerate less approximation, we must compare larger parts of the policy trees.

We note that the minimal mixing rate depending on the function, F_{a_j,o_j}, may also assume two extreme values: $\gamma_F = 1$ and $\gamma_F = 0$. The former case implies that the updated beliefs are identical. For example, they have all probability mass in the same state, and the KL divergence of these distributions is zero after a transition. Hence, we set $d = 1$. For the latter case, there is at least one pair of states from which the updated beliefs do not agree at all: There is no overlap among the states receiving probability masses in the two distributions. For this null mixing rate, the KL divergence may not contract and d may not be derived. Thus, we may arbitrarily select $d \leq T-1$.

We illustrate the computation of the minimal mixing rate using an example.

Example 3 (*Computing γ_F*) Notice from Eq. 6 that the final mixing rate is the minimum of all mixing rates for different combinations of action and observation of agent j.

Let us compute $\gamma_{F_{a_j,o_j}}$ for the tiger problem when $a_j = L$ and $o_j = GL$. We use Eq. 2 to first compute $F_{L,GL}(s' = TL | s = TL)$ and $F_{L,GL}(s' = TR | s = TL)$ with values from "Multiagent tiger problem" section of Appendix:

$$\begin{aligned} F_{L,GL}(s' = TL | s = TL) &= \alpha Pr(GL|TL, L) \cdot Pr(TL|TL, L) \\ &= \alpha 0.85 \cdot 1 = 0.85\alpha \\ F_{L,GL}(s' = TR | s = TL) &= \alpha Pr(GL|TR, L) \cdot Pr(TL|TR, L) \\ &= \alpha 0.15 \cdot 0 = 0 \end{aligned}$$

where α is the normalization constant. Next, we compute $F_{L,GL}(s' = TL|s = TR)$ and $F_{L,GL}(s' = TR|s = TR)$:

$$\begin{aligned} F_{L,GL}(s' = TL|s = TR) &= \alpha Pr(GL|TL, L) \cdot Pr(TL|TR, L) \\ &= \alpha 0.85 \cdot 0 = 0 \\ F_{L,GL}(s' = TR|s = TR) &= \alpha Pr(GL|TR, L) \cdot Pr(TR|TR, L) \\ &= \alpha 0.15 \cdot 1 = 0.15\alpha \end{aligned}$$

In order to obtain $\gamma_{F_{L,GL}}$, we take the minimum of all $F_{L,GL}(\cdot|\cdot)$ values as shown in Eq. 3. In other words, $\gamma_{F_{L,GL}}$ may be obtained by summing the minimum values across the rows or columns of the matrix below:

	s	
	TL	TR
s'		
TL	$0.85\,\alpha$	0
TR	0	0.15α

Therefore, $\gamma_{F_{L,GL}} = 0 + 0 = 0$. Because γ_F is the minimum across all $\gamma_{F_{a_j,o_j}}$, which cannot be less than 0, the minimal mixing rate for the tiger problem is 0.

Because we may compute depth, d, analytically in most cases, we revise Definition 2 to obtain a definition of ϵ-BE for $\epsilon \geq 0$:

Definition 3 (ϵ-*BE*) Given two models of agent j, $m_{j,l-1}$ is ϵ-BE ($\epsilon > 0$) to $\hat{m}_{j,l-1}$ if their depth-d policy trees, where d is computed according to Eq. 8, are identical, $\pi^d_{m_{j,l-1}} = \pi^d_{\hat{m}_{j,l-1}}$, and beliefs at the leaves of the two ordered depth-d policy trees diverge by at most ϵ: $max_{k=1...|\Omega_j|^d} D_{KL}[b^{d,k}_{m_{j,l-1}} || b^{d,k}_{\hat{m}_{j,l-1}}] \leq \epsilon$. If $\epsilon = 0$, this definition collapses into Proposition 1 with an arbitrary depth, d.

Note that ϵ-BE is not symmetric and not necessarily transitive.

In summary, we may group together two models that are approximately BE by a measure of ϵ or less by first determining the depth d using Eq. 8 and then ensuring that the partial policy trees, which are the solutions of the models, are identical down to the depth d.

5 Incremental comparison of model solutions

Notice that the mixing rate, γ_F, as computed using Eqs. 3 and 6 is the minimal one, and a single rate is computed for a problem domain. This often leads to an overly large depth, d, with the divergence between the updated beliefs for pairs of policy trees reducing to smaller than ϵ before the depth is reached. We introduce further efficiency in memory usage by addressing this limitation.

One way to avoid comparing depth-d balanced trees fully for equality is to compare branches of increasing length incrementally until either the divergence between the updated beliefs at the corresponding nodes drops to ϵ or smaller, or the depth reaches d. Let d_L be the least depth at which further expansions from a node in the two trees for the purpose of comparison are *blocked*. Depth, d_U, is the greatest such depth and $d_U \leq d$.

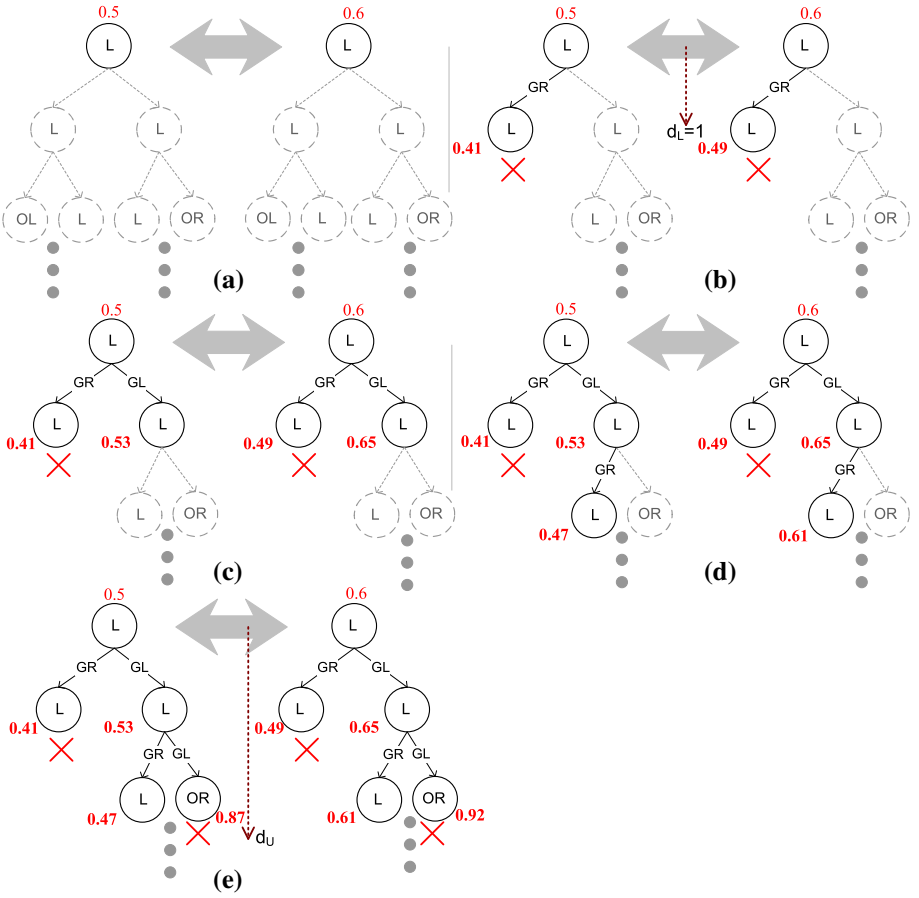


Fig. 12 Incrementally expanding and comparing policy tree solutions of two models for equality. A *times symbol* below a node indicates that the tree need not be expanded further from that node for comparison purposes because the KL divergence of the updated beliefs at the node pair is equal to ϵ or less

Example 4 (*Incremental comparison*) We illustrate this procedure using a pair of example policy trees for the multiagent tiger problem, in Fig. 12. Let $\epsilon = 0.015$ for this example. As we mentioned previously, the minimal mixing rate for the multiagent tiger problem, $\gamma_F = 0$. Therefore, we may arbitrarily select a large depth, d. We begin by comparing the initial beliefs of the two models as shown in Fig. 12a. As $D_{KL}\left(\langle 0.5, 0.5\rangle || \langle 0.6, 0.4\rangle\right) > \epsilon$, we expand the branches from the root node breadth-wise left to right computing the updated beliefs at the next set of nodes. Because $D_{KL}\left(\langle 0.41, 0.59\rangle || \langle 0.49, 0.51\rangle\right) < \epsilon$, we do not expand the tree further along this branch, as shown in Fig. 12b. For our example, $d_L = 1$. Because the KL divergence between $\langle 0.53, 0.47\rangle$ and $\langle 0.65, 0.35\rangle$ remains larger than ϵ, we continue to expand the trees along this branch. As we show in Fig. 12d, e, we continue comparing the KL divergences of the updated beliefs at the corresponding leaf nodes. The KL divergence between the updated beliefs, $\langle 0.87, 0.13\rangle$ and $\langle 0.92, 0.08\rangle$ at the leaf nodes being smaller than ϵ, we need not expand along this branch further. As we continue expanding the policy trees, the upper bound on the depth, d_U, is larger than 2 but does not exceed d.

The primary benefit of incrementally comparing the policy trees is now obvious: We need not store perfect depth-d balanced trees in memory; rather, we may obtain computational savings if d_L is significantly smaller than d. Notice that we need not precompute d if we are incrementally comparing the policy trees. Instead, further comparisons are blocked along a branch whenever the KL divergence reduces to ϵ or smaller. Recall that for extreme values of the mixing rate such as when $\gamma_F = 0$, Eq. 8 is unable to provide a value for d. Therefore, a secondary benefit of the incremental comparison approach is that we need not arbitrarily select d for pathological mixing rates.

Recall that ϵ-BE (as per Definition 3) can be ensured by comparing entire depth-d policy trees. Proposition 4 points out that if two models are ϵ-BE, the incremental comparison will identify the models as ϵ-BE as well.

Proposition 4 (Quality) *Given two models of agent j, $m_{j,l-1}$ is ϵ-BE to $\hat{m}_{j,l-1}$, $\epsilon > 0$. Then, incremental comparison will identify the two models as ϵ-BE.*

Proof Let $\pi^d_{m_{j,l-1}}$ and $\pi^d_{\hat{m}_{j,l-1}}$ be the depth-d policy trees obtained from models, $m_{j,l-1}$ and $\hat{m}_{j,l-1}$, respectively, where d is determined using Eq. 8 given ϵ. Because the models are ϵ-BE, the two policy trees are identical as per Definition 3. Proceeding in a breadth-wise manner through the policy trees, the incremental comparison updates beliefs at the nodes and measures the KL divergence between them. For each pair of compared branches of the two policy trees, the divergence either between beliefs at the intermediate nodes of the branches or between beliefs at the leaf nodes reduces to smaller than or equal to ϵ.

If the divergence at intermediate nodes falls below ϵ, we do not compare any further along those branches. Importantly, because the branches are identical, the divergence will continue to reduce or remain the same, thereby satisfying the property of the divergence being smaller than or equal to ϵ. In the worst case, updated beliefs at the leaf nodes are compared, whose divergence is guaranteed to be smaller than ϵ. This holds for all pairs of branches that are compared. Therefore, the incremental comparison will also identify the two models as ϵ-BE for a given $\epsilon > 0$. □

On the other hand, models deemed to be ϵ-BE by the incremental technique may not precisely be ϵ-BE (as defined in Definition 3). This is because the incremental comparison does not compare the branches from the least depth d_L to the depth d in the policy trees for being identical. Therefore, the clustering of models may now differ from the previous approach with additional models being possibly included in a cluster.

6 Computational savings and predictive error bound

Given that we may analytically determine d using Eq. 8, the complexity of identifying whether a pair of models are approximately BE is dominated by the complexity of comparing two depth-d trees. This is proportional to the number of comparisons made as we traverse the policy trees. As there are a maximum of $|\Omega_j|^d$ leaf nodes in a depth-d balanced tree, the following proposition gives the complexity of identifying BE classes in the model node of agent i's I-DID at some time step.

Proposition 5 (Complexity of BE). *The asymptotic complexity of the procedure for identifying all models that are ϵ-BE is $\mathcal{O}(|\mathcal{M}_{j,l-1}|^2|\Omega_j|^d)$, where $|\mathcal{M}_{j,l-1}|$ is the number of models in the model node.*

While the time complexity of comparing two partial policy trees is given by Proposition 5 (set $|M_{j,l-1}| = 2$), we maintain at most $2|\Omega_j|^d$ paths ($d \leq T-1$) at each time step for

each pair of models that are being compared, with each path occupying space proportional to d. This avoids storing entire policy trees containing $(|\Omega_j|)^{T-1}$ possible paths, leading to significant savings in memory when $d \ll T$.

Further computational savings are typically obtained when the partial policy trees are compared incrementally. Although in the worst case, $d_L = d_U = d$, and therefore the worst-case asymptotic complexity continues to be defined by Proposition 5, typically $d_L < d_U$. Consequently, the number of leaf nodes compared between two trees is significantly less than $2|\Omega_j|^d$. However, these additional savings in memory are moderated by the additional expense of updating beliefs after each observation-action branch and computing the KL divergence between two belief distributions.

We analyze the error in the value of j's predicted behavior. If $\epsilon = 0$, grouped models are exactly BE and there is no error. With increasing values of ϵ (resulting in small d values), model, $m_{j,l-1}$, may be approximately grouped with the model, $\hat{m}_{j,l-1}$, from which it is actually behaviorally distinct. Let $\hat{m}_{j,l-1}$ be the model, which when associated with $m_{j,l-1}$ results in the worst error. Let α^T and $\hat{\alpha}^T$ be the exact value vectors for the entire policy trees obtained by solving the two models, respectively. Then, if $\hat{m}_{j,l-1}$ is selected as the representative of the group, the error is: $\rho = |\alpha^T \cdot b^0_{m_{j,l-1}} - \alpha^T \cdot b^0_{\hat{m}_{j,l-1})}|$. Let d_{min} be the smallest depth of comparison across all pairs of policy trees in the model node. Because the depth-d_{min} policy trees of the two models are identical (Definition 2), the error arises from the remaining parts of the policy trees that are not compared, which could be different, and it becomes:

$$\rho = \max_k |\alpha^{T-d_{min}-1} \cdot b^{d_{min}+1,k}_{m_{j,l-1}} - \alpha^{T-d_{min}-1} \cdot b^{d_{min}+1,k}_{\hat{m}_{j,l-1}}|$$

Here, α^{T-d-1} is the value vector of the policy subtree, which is optimal for $b^{d+1,k}_{m_{j,l-1}}$, and $b^{d+1,k}_{m_{j,l-1}}$ and $b^{d+1,k}_{\hat{m}_{j,l-1}}$ are beliefs at which the error is maximum.

Proposition 6 *The worst-case error in the value of j's predicted behavior due to grouping models that are ϵ-BE is bounded as,*

$$\rho \leq (R_j^{max} - R_j^{min})(T - d_{min} - 1) \cdot 2\epsilon$$

where R_j^{max} and R_j^{min} are the maximum and minimum rewards of j over any state and action, respectively.

The proof of this proposition is given in "Appendix 1". On the other hand, if we incrementally compare policy trees for equality as described in Sect. 5, then the worst error occurs when all the branches of the two policy trees are blocked from further comparisons at depth, $d_L = d_U$, which is smaller than d_{min}. We may obtain the error by substituting d_{min} with d_L in the final inequality above due to which the error bound becomes:

$$\rho \leq (R_j^{max} - R_j^{min})(T - d_L - 1) \cdot 2\epsilon$$

Note that because $d_L < d$, this error bound is larger than the previous one, implying that the error could be worse.

Of course, these errors are tempered by the probability that agent i assigns to the model, $m_{j,l-1}$, in the model node at time step, d.

7 Experimental results

We implemented our methods of determining ϵ-BE between models and use them to group models into classes. Specifically, for a given value of ϵ, we group models together that are ϵ-BE with a representative thereby resulting in approximate equivalence classes, which partitions the model space. This is followed by retaining the representative for each class while pruning others, analogously to using exact BE. This procedure now implements **PruneBehaviorEq** (line 6) in Fig. 9. Flat DIDs obtained by transforming the I-DIDs are solved as limited memory DIDs [25] using the Hugin Expert API.

Because our approach is the first to formalize an approximation of BE (to the best of our knowledge), we compare it with the previous most efficient algorithm that exploits exact BE while solving I-DIDs. This technique [49] groups BE models using their entire policy trees and updates only those models that will be behaviorally distinct from existing ones; we label it as DMU. As a second baseline, we transform the I-DIDs into limited memory DIDs [25] at each level with the decision node in each time-slice remembering the previous decision only. Models are not grouped for pruning in the model nodes, and we label this approach as LIM I-DID.

We demonstrate the properties of the methods using two standard problem domains: the two-agent tiger problem ($|S| = 2, |A_i| = |A_j| = 3, |\Omega_i| = 6, |\Omega_j| = 3$) described previously in Sect. 2, and the multiagent version of the concert problem ($|S| = 2$, $|A_i| = |A_j| = 3$, $|\Omega_i| = 4$, $|\Omega_j| = 2$).[1] In particular, we hypothesize that (a) as we increase the measure of approximation, ϵ, fewer equivalence classes appear in the partition of the model space, $\mathcal{M}_{j,l-1}$, and (b) the quality of the I-DID's solution approximation approaches that of the exact DMU as ϵ reduces.

Equally importantly, we also evaluate the scalability of the methods by applying it to a larger domain related to money laundering and using a scalable multiagent testbed with practical implications called the Georgia testbed for autonomous control of vehicles (GaTAC) [15,43]. We simulate a much larger problem domain in GaTAC: the two-agent unmanned aerial vehicle (UAV) interception problem in a 5×5 grid ($|S| = 81$, $|A_i| = |A_j| = 5$, $|\Omega_i| = |\Omega_j| = 5$). I-DIDs for all problem domains are provided in "Appendix".

7.1 Anytime performance and equivalence classes

We report on the performance of the techniques—ϵ-BE, DMU, and LIM I-DID—when used for solving level 1 I-DIDs of increasing horizon in the context of the small domains. We show that ϵ-BE produces a better-quality solution than DMU and LIM I-DID in the same amount of allocated time and computational resources, and the quality of the solution generated by ϵ-BE converges to that of the exact DMU as ϵ decreases and more computational resources are available (with the corresponding increase in d). The multiagent tiger problem exhibits a minimal mixing rate of zero, due to which the partial depth, d, is selected arbitrarily. We increase d as ϵ reduces.

In Fig. 13a, we show the best solution possible on average for a given time allocation for the multiagent concert and tiger problems. Notice that ϵ-BE consistently produces better-quality solution than DMU and LIM I-DID. This is because it solves for a longer horizon than the latter techniques in the same time. A lack of any pruning of the models in LIM I-DID is predominantly responsible for its relatively poor performance and its inability to generate better-quality solutions. Figure 13b shows the average rewards gathered by simulating the

[1] We adapt the single-agent concert problem from the POMDP repository at http://www.cs.brown.edu/research/ai/pomdp/.

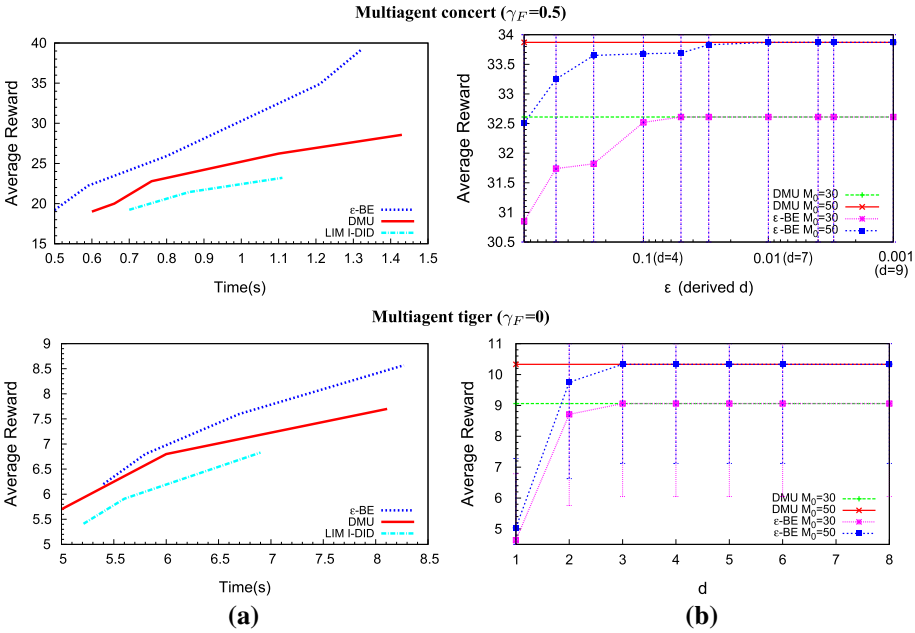


Fig. 13 **a** For a given allocation of time, ϵ-BE may produce solutions of significantly better quality than DMU and LIM I-DID by running for more horizons and greater number of initial models. This clearly shows the benefit of the approximation. Experiments were run using a level 1 I-DID on a Linux platform with Intel Core2 2.4 GHz with 4 GB of memory. **b** The approximations converge to the exact DMU as ϵ reduces and this is consistent for differing numbers of initial models, $M_{j,0}^0$

solutions obtained for decreasing ϵ for each of the two problem domains. We used a horizon of 10 for the small domains. Each data point is the average of 500 runs where a model of j is sampled according to i's initial belief and assigned to j in the simulations. For a given number of initial models, $M_{j,0}$, the solutions improve and converge toward the exact (DMU) as ϵ reduces. While the derived partial depths vary from 0 up to the horizon minus 1 for extremely small ϵ, we point out that the solutions converge to the exact for $d < T - 1$, including the tiger problem (at $d = 3$) despite the zero mixing rate.

Interestingly, Fig. 14 confirms our intuition that ϵ-BE leads to significantly fewer model classes for large ϵ, and therefore smaller d. However, the number of classes when $\epsilon = 0$ is more than DMU because diverging leaf beliefs may still lead to BE of models as mentioned

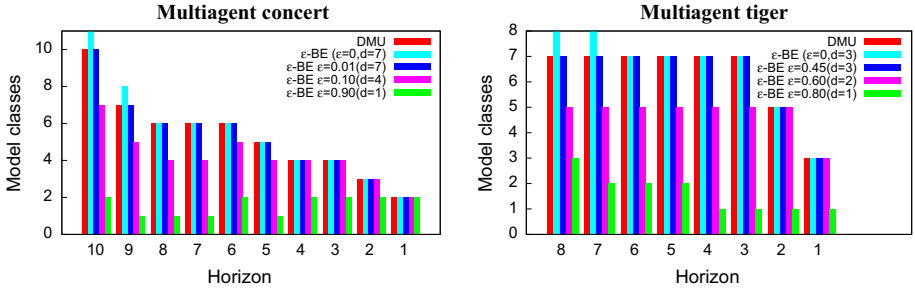


Fig. 14 Model space partitions by ϵ-BE and DMU across the different horizons while solving a level 1 I-DID. As ϵ increases, more models are grouped into a single class due to increased tolerance resulting in less classes in the partition

previously in Sect. 3. Importantly, comparing partial policy trees is sufficient to obtain almost the same model space as in the exact case, which is responsible for the early convergence to the exact reward we observe in Fig. 13b.

Next, we evaluate the benefit of incrementally comparing the model solutions. We label this approach as ϵ-BE-Inc and particularly compare it with ϵ-BE. In the context of the two problem domains we are using, ϵ-BE-Inc leads to solutions of the level 1 I-DIDs that are of better quality than those by ϵ-BE in the same amount of allocated time. Although incrementally comparing the trees often leads to a greater approximation, the associated computational savings allow us to solve I-DIDs of longer horizons and with more models in the given time. We show this in Fig. 15.

As we may expect, ϵ-BE-Inc groups more models together due to its comparison of smaller portions of the trees for equality, which leads to less equivalence classes in the partition of the model space. In Fig. 16, we see that ϵ-BE-Inc results in a fewer number of classes for about half the horizons.

In the context of fewer numbers of equivalence classes, what are the values of d_L that emerge as we compare models? We investigated this question and discovered that for a typical model space, d_L surprisingly remained smaller than d. In Fig. 17, we show the percentage of all model comparisons, which exhibited different values of d_L.

Finally, in Table 1, we compare the different techniques based on the time each takes to solve problems of increasing horizons. We additionally include a heuristic approach [47], labeled TopK, that samples K paths from a policy tree that are approximately most likely to occur and uses just these paths to compare for equivalence, and we include an additional problem domain involving UAV reconnaissance of a fugitive who intends to reach a safe

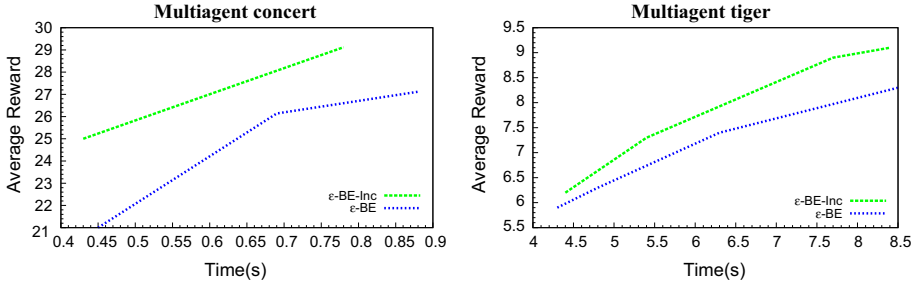


Fig. 15 Incrementally comparing policy trees as in ϵ-BE-Inc often leads to early terminations along some branches and is significantly more efficient in general than ϵ-BE

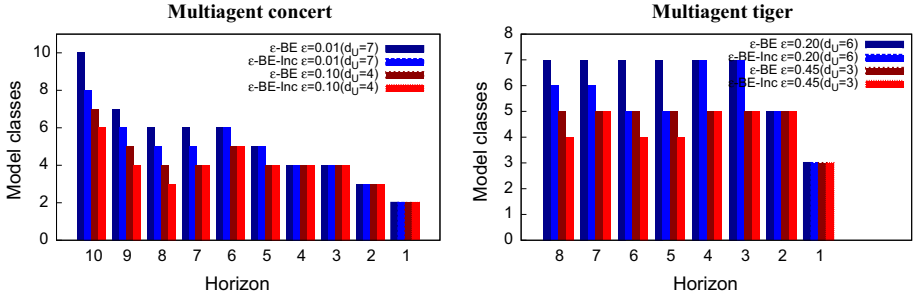


Fig. 16 Greater approximation in ϵ-BE-Inc leads to less classes and model representatives than ϵ-BE for longer horizons

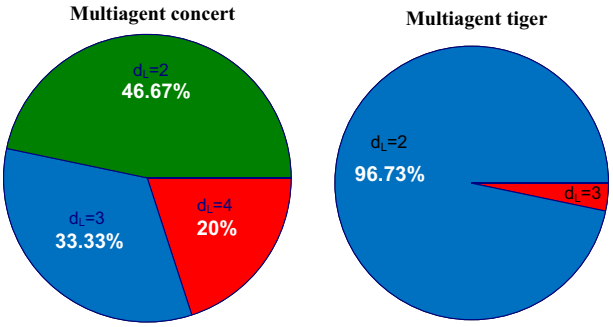


Fig. 17 Percentages of all model comparisons, which exhibited the different values of d_L. **a** $d_U = d = 5$, **b** $d_U = d = 6$. Interestingly, notice that d_L remained less than d for all comparisons

Table 1 ϵ-BE-based approaches show scalability to a very large horizon in the context of I-DIDs, in multiple problem domains

Level 1 I-DID	T	Time (s)			
		DMU	TopK	ϵ-BE	ϵ-BE-Inc
Concert	6	0.29	0.36	0.31	0.11
	10	2.3	2.4	1.9	0.22
	25	*	**336.24**	**13.1**	**9.1**
Tiger	6	0.34	0.31	0.21	0.16
	8	1.3	3.7	0.37	0.21
	20	*	**218**	**3.1**	**2.49**
UAV3	6	19.3	10.1	8.9	8.1
	8	186.7	111	27	19
	10	*	462	55	48
	20	*	*	**98**	**76**
	25	*	*	*	**132**

Experiments were run on a Linux platform with Intel Core2 2.4 GHz with 4 GB of memory The bold values highlight the scalability of the proposed BE methods over the planning horizons

sector, which is played out in a 3 × 3 grid ($|S| = 25$, $|A_i| = |A_j| = 5$, $|\Omega_i| = |\Omega_j| = 5$) [15]. Both ϵ-BE and ϵ-BE-Inc demonstrate significant scalability over DMU, solving for much longer horizons than exactly possible. They show significant run time speedup over TopK as well, which needs to maintain complete paths, though not all, that grow long. ϵ and K were varied to get the same reward as DMU if appropriate, otherwise until the model space stabilized. Additional savings obtained by ϵ-BE-Inc allow us to solve the UAV3 problem up to a horizon of 25 in a reasonable amount of time. We point out that expanding I-DIDs to 20 or more time steps surpasses previous horizon-centric scalability results for such domains by a large margin [49], and the resulting solutions are of quality that is sufficiently high for these problem domains, which would satisfy most requirements.

7.2 Scalable performance

We demonstrate scalability of the techniques by utilizing them to counter money laundering cast as an adversarial decision-making problem (Sect. 7.2.1) and to solve the sequential decision-making problem of a UAV I tasked with intercepting another hostile UAV J, which intends to raid an immobile allied military base (Sect. 7.2.2).

7.2.1 Countering money laundering

The money laundering problem, introduced by [29] and possessing realistic underpinnings, is a game between law enforcement (blue team) and the money launderers (red team) who aim to move their assets from a "dirty" pot to a "clean" one through a series of financial transactions while evading capture by the blue team. The blue team can place sensors at various locations such as bank accounts, trusts, and real estate to detect the presence of the "dirty" money. The physical state is defined by the joint location of the dirty money and that of the sensor. The possible locations of the red team's assets are: *dirty pot*, *bank accounts*, *insurance, securities*, *offshore accounts*, *shell companies*, *trusts*, *corporate loan*, *casino accounts*, *real estate*, and *clean pot*. The possible locations of the blue team's sensor are: *bank accounts*, *insurance, securities*, *shell companies*, *trusts*, *corporate loan*, *casino accounts*, *real estate*, and *none*. The red team may perform any of the three nondeterministic actions of placement, layering or integration to move its assets from one location to another or it could listen to gain noisy information about the location of the blue team's sensor. The blue team may place its sensors in one of the eight locations or it could confiscate the assets of the red team. As [29] mention, this problem is about 20 times larger than the previous small problem domains. It exhibits a physical state space of 99 states for the subject agent (blue team), 9 actions for the subject agent and 4 for the opponent, and 11 observations for the subject and 4 for the other. More details on this domain including the I-DID models are discussed in "Money laundering problems" section of Appendix.

Blue team's decision making in the money laundering problem is modeled using a level 1 I-DID, which ascribes level 0 models to the red team. The blue team ascribes 40 or more models to the other in the initial model node. Its initial belief is a uniform distribution over the physical states and models of the other team. As we notice from Fig. 18a, DMU's performance is close to that of ϵ-BE for small time allocations until ϵ-BE breaks away by solving I-DIDs of horizons up to 10 and considering more models. In comparison, DMU is unable to move past I-DIDs of horizon 8 for this large problem, while LIM I-DID does not solve past 4 horizons despite forgetting all but the previous decision. Each data point is the average of 150 runs of the blue team's I-DID. The blue team is able to account for more models on using ϵ-BE-Inc compared to ϵ-BE and solve I-DIDs of horizon 12, which is longer than that by ϵ-BE, as we show in Fig. 18b.

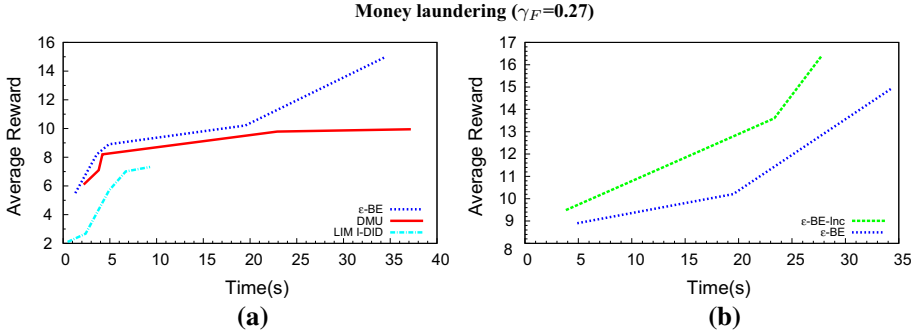


Fig. 18 **a** ϵ-BE continues to demonstrate improved value in comparison with DMU and LIM I-DID for the same allocation of time, for this large problem domain. We may solve a level 1 I-DID up to horizon 10 with more than 40 initial models in approximately 35 s. Experiments were run using a level 1 I-DID on a Linux platform with Intel Core2 2.4 GHz with 4 GB of memory. **b** Terminating comparisons early allows ϵ-BE-Inc to account for more models thereby improving on ϵ-BE

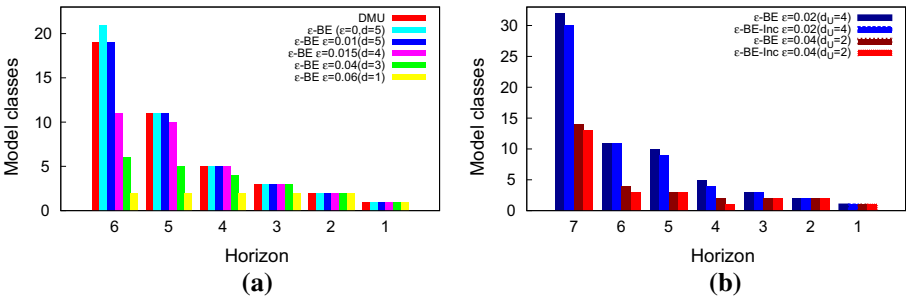
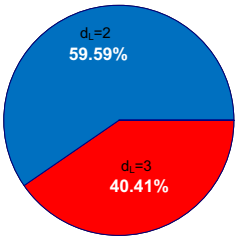


Fig. 19 The number of behaviorally distinct model classes drops for the initial time steps, as we relax ϵ allowing partial policy trees of smaller depth to be compared and more models to be grouped into a class, in comparison with DMU



Level 1 I-DID	T	Time (s)			
		DMU	TopK	ϵ-BE	ϵ-BE-Inc
Money Laundering	6	148.8	85.8	78.3	67.7
	8	787.6	582.5	220.1	208.6
	10	*	*	1320	1210
	12	*	*	*	2410

(b)

Fig. 20 **a** Proportion of all policy tree comparisons in the money laundering problem for which $d_L < d_U$ is high, where $d_U = d = 5$ for $\epsilon = 0.025$. **b** Run times of different approaches on a Linux platform with Intel Core2 2.4 GHz with 4 GB of memory. '*Asterisk*' indicates that the solution ran out of memory

In order to understand the reason behind the improved performance of ϵ-BE, we again find out the number of model classes at each time step in the model node of the blue team's I-DID. As we show in Fig. 19a, more than 40 initial models are grouped into 19 model classes by DMU, and just 11 classes by ϵ-BE for $\epsilon = 0.015$, which leads to partial policy trees of depth 4. This number of model classes steadily drops as we increase ϵ. The incremental comparison of ϵ-BE-Inc produces a slightly fewer number of model classes because more models get grouped as approximately BE due to the partial comparisons. As shown in Fig. 20a, about 60 % of the model pair comparisons exhibited a reduced comparison depth of, $d_L = 2$, along at least one branch of the policy trees, while the remaining exhibited, $d_L = 3$. This primarily results in the improved performance of ϵ-BE-Inc seen previously. Figure 20 shows the time taken to solve a level 1 I-DID using different approaches while attaining the same value as DMU. Partial policy comparisons compress the model space substantially allowing more models and taking less than half as much time as that by the exact approach. While [29] reported solving the money laundering problem with a maximum look ahead of 4, we may scale in horizon up to 12 using ϵ-BE-Inc with $|\mathcal{M}_{j,0}| = 40$. This scalability is in part due to a factored representation of the state space enabled by graphical models, as well.

7.2.2 UAV reconnaissance and interception in GaTAC

We assume that the UAV scenario is played out in a 5×5 grid of sectors as shown in Fig. 21a. We formulate the problem from the perspective of UAV I. The state space is modeled using

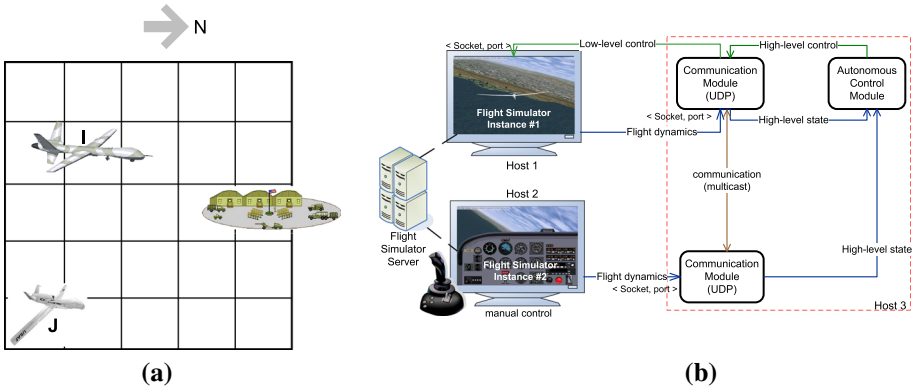


Fig. 21 **a** A larger problem domain involving UAV reconnaissance and interception in a 5×5 combat theater. UAV I is tasked with intercepting J before the latter raids the allied military base. **b** Architectural design of GaTAC showing two networked instances of a flight simulator one of which is autonomous and the other is manually controlled

the possible relative locations of UAV J with respect to I. In a 5×5 theater, this space consists of 81 physical states including *same*, *north*, *south*, *east*, *west*, *north-west*, *north-east*. J's physical state space consists of its absolute location in the grid. Both the UAVs may move in one of the four cardinal directions, or loop in their current locations in a full surveillance mode. Thus, the actions for I and J are *{move_north, move_south, move_east, move_west, listen}*. Typical UAVs such as the MQ-1B predator possess an infrared sensor and a color TV camera, which allows them to carry out ground and aerial reconnaissance. Consequently, we assume that both UAVs have the following observational capabilities: *{sense_north, sense_south, sense_level, sense_found}*, where UAV I's sensing target is J and J's target is the allied military base, and while *sense_north* and *sense_south* are self-explanatory, *sense_level* denotes that the target is in the same column and *sense_found* denotes that the target is in the same sector as the UAV. At a strategy level of 0, UAV J's transition function straightforwardly models the nondeterministic movements of J. However, I's transitions are contingent on the joint actions of both UAVs. The noise in determining the next state is not only due to a small amount of nondeterminism in its movement but also due to the state being relative. UAV J knows the location of the base in the theater. However, because it is not perfectly aware of its own location, it may sense the base noisy only when it is within a radius of 1 sector from it. On the other hand, UAV I senses J accurately as being north of it if J is in any sector that is north of I, and analogously for the other observations. The reward functions penalize excessive action taking by associating each action with a small cost and rewards the UAVs for performing an action and receiving an observation of *sense_found*, indicating that the UAVs may intercept their respective targets.

We modeled UAV I's decision-making problem using a level 1 I-DID modeling J using level 0 DIDs. Four distinct classes of models of J were included, which were grouped into approximate BE classes for differing values of ϵ using the approaches of ϵ-BE and ϵ-BE-Inc. These models differ in the initial beliefs that I thinks J has about its location. They imply that J is in the lower part of the grid and in the same column, or slightly to the left or right of UAV I. We show example policy trees for I computed using the two approaches of ϵ-BE and ϵ-BE-Inc, in Fig. 22. Here, UAV I believes that J is to its north-west. To permit illustration, we show policies for a horizon of 3.

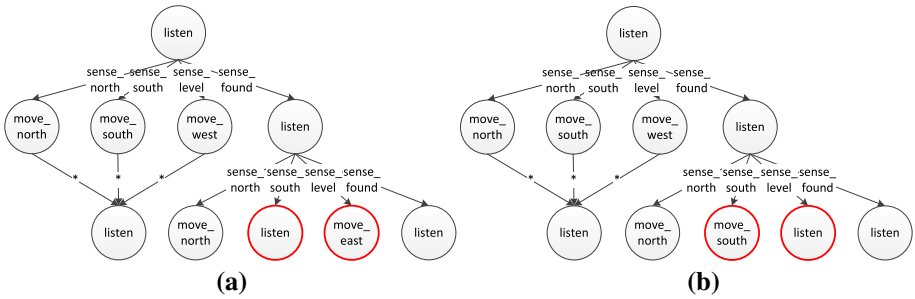


Fig. 22 Horizon 3 policy trees for UAV I computed using, **a** ϵ-BE, and **b** ϵ-BE-inc approaches. Here, $\epsilon = 0.42$ leading to $d = d_U = 2$. The distinctions between the two policies are highlighted

Policy computed using ϵ-BE in Fig. 22a improves on the policy in Fig. 22b by recommending a final action of *listen* on sensing UAV J south of I. This is because the game has reset previously as J was intercepted—indicated by the observation of *sense_found*—and is now to the north-west of I. Furthermore, its preferable to move west or east on observing *sense_level* thereby making another intercept possible, instead of listening.

Extended policies of UAV I were run in the theater of Fig. 21a simulated in GaTAC, which is a hyper-realistic simulation environment for evaluating control of autonomous robotic vehicles such as UAVs. GaTAC provides a low-cost and open-source (GNU Affero public license version 3) alternative to highly expensive simulation infrastructures for an academic laboratory setting. A simplified architectural design of GaTAC is shown in Fig. 21b where an autonomous UAV interacts with a manually controlled one. Briefly, GaTAC deploys multiple instances of an open-source flight simulator called FlightGear [34] with 3D scenery from TerraGear, possibly on different networked platforms which communicate through an external server. Multiple UAVs may communicate with each other using the multicast protocol. GaTAC provides a complete workflow to users which facilitates setting up cooperative or noncooperative environments and parsing policies produced by recognized decision-making frameworks including decentralized POMDPs.

The use of ϵ-BE allows us to solve UAV I's decision-making problems for horizons of up to 10. This represents a significant step forward toward meaningfully applying these sophisticated frameworks to real-world problem domains. We show the impact of ϵ-BE and ϵ-BE-Inc with identical ϵ and d_U of 6 and 8 on the space of models of the other UAV J in Fig. 23. Models whose numbers could reach greater than fifty thousands by the last time step may be grouped together to result in a tractable model space. As we may expect, ϵ-BE-Inc continues to group more models together compared to ϵ-BE, thereby resulting in a reduced number of model classes.

We simulated policies of horizons 8 and 10 for UAV I with three different initial beliefs, in GaTAC. Policies were generated using both ϵ-BE and ϵ-BE-Inc. Our procedure involved sampling a model of UAV J and the initial locations of UAVs I and J, all distributed according to I's particular initial belief. UAVs I and J then pursued trajectories in the theater of Fig. 21a guided by their policies.[2] We ran 25 simulations for each horizon and approach in GaTAC, resulting in a total of 100 simulations. Each run ends when either J is intercepted or it reaches the allied base.

We did not observe a significant difference in the average reward performance between the two policies of different horizons, as we show in Table 2. This is primarily because a horizon

[2] The starting locations of the UAVs may differ from those shown in Fig. 21a.

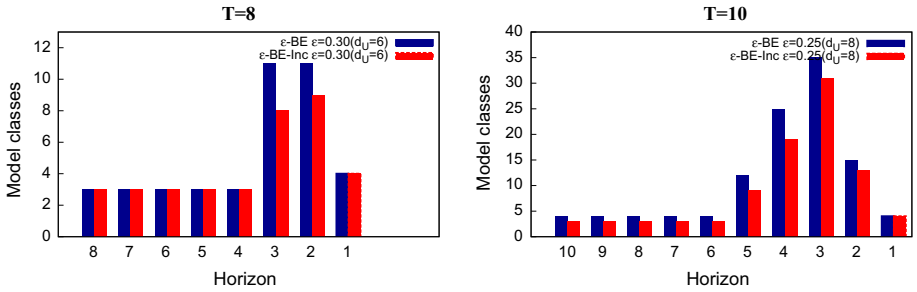


Fig. 23 Models are grouped into a tractable number of classes by the approaches, ϵ-BE and ϵ-BE-Inc, for the large UAV interception problem. Solving for $d = d_U$ results in a value of 6 for $\epsilon = 0.3$ and a value of 8 for $\epsilon = 0.25$ in this problem domain

Table 2 Average rewards and run times for expanding the large I-DIDs and solving them on a platform with a 2.4 GHz processor and 4GB of memory

Level 1 I-DID	T	ϵ-BE			ϵ-BE-Inc		
		Avg. Rwd	Time (s)		Avg. Rwd	Time (s)	
			Expansion	Solution		Expansion	Solution
UAV	8	38.20	202	0.98	37.32	199	0.91
5×5	10	43.64	1026	3.8	46.68	811	2.2

Notice that the policy obtained using ϵ-BE did not consistently obtain a better reward than that of ϵ-BE-Inc. We do not include the time it takes to extract and write out the policy trees, which becomes substantial for large horizon trees. Nevertheless, solutions of longer horizon I-DIDs are now possible

of 8 and consequently 10 are both sufficient in order to obtain a good quality policy for UAV I in our theater. Furthermore, both ϵ-BE and ϵ-BE-Inc generated policy trees that were similar for most parts although not identical due to which we did not observe a significant difference in performance between them. This could be an indication that the two approaches yield comparative results for large problems. However, Table 2 demonstrates a significant reduction of about 20 % between the two techniques in the time it takes to expand and solve the large I-DIDs for a horizon of 10. See Fig. 9 for details on the two phases of the algorithm. This is primarily due to the reduced number of model classes maintained by ϵ-BE-Inc at many of the time steps as demonstrated in Fig. 23b. We were able to further scale the horizon to 12 with ϵ-BE consuming a total of 2033 s and ϵ-BE-Inc taking 30 % less time, before the solution phase ran out of memory for longer horizons.

A closer look at our simulations reveals that out of the total 100 runs, UAV I intercepted J in 65 of the simulation runs. Among the remaining runs, J reached the allied base in 22 of them, while the remaining 13 did not yield any result. Consequently, J was intercepted about three times more than it reached the base despite the considerable amount of uncertainty in the problem. However, the overall capture rate was 65 % indicating that there is room for improvement. We show selected trajectories of UAVs I and J utilizing the two techniques for a horizon of 8 in Fig. 24. We select example trajectories where UAV J reaches the allied base and those where I intercepts J before it does so. We point out that I is unsure of the exact initial location of J due to which in some of the runs, J narrowly escapes being intercepted, as in the lower theater of Fig. 24b.

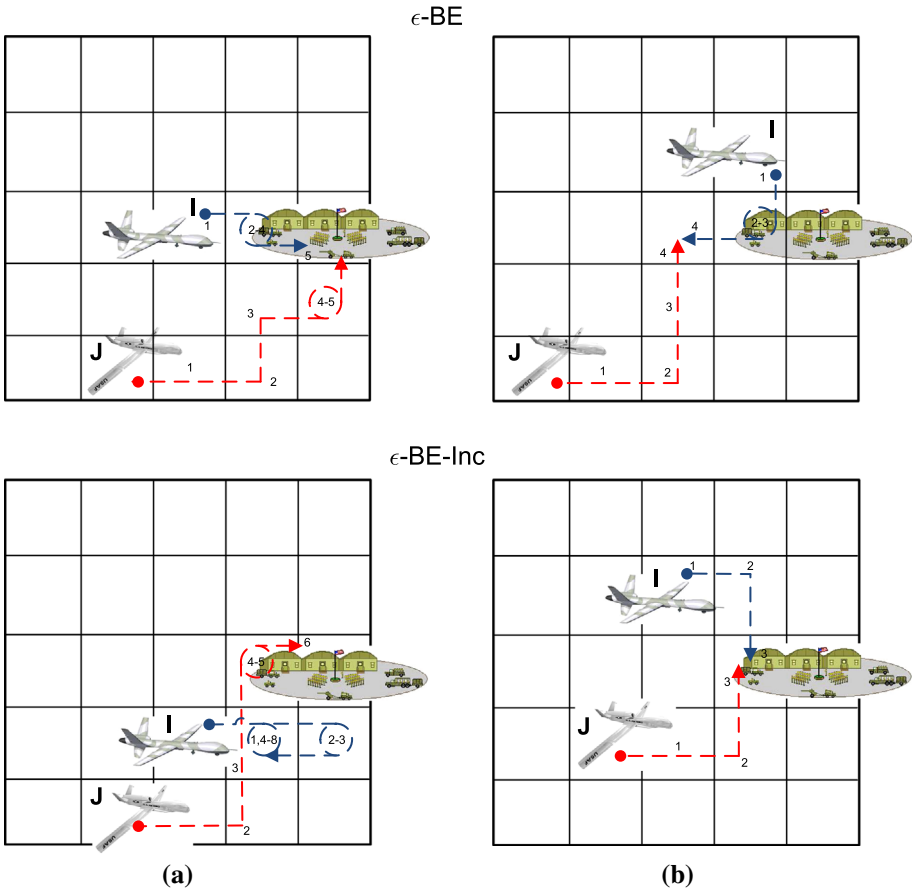


Fig. 24 Two example runs depicting UAV J, **a** reaching the base, and **b** being intercepted by I. The numbers on the trajectories denote the policy steps of the two UAVs; they move continuously and concurrently in the simulation. The dots indicate their starting locations. Notice that the policies produce trajectories for I that can get sophisticated

In summary, our empirical analysis using multiple problem domains shows that ϵ-BE-Inc results in quicker solutions compared to ϵ-BE though at the expense of greater approximation. However, the improved efficiency may be leveraged to solve for larger dimensions. Both generate solutions of flexible quality. In general, ϵ-BE-based approaches show improved scalability: We solved a 5×5 UAV theater for a horizon of 8 in time that is less by an order of magnitude than the time it took previously for a comparable domain [49], and scaled further to longer horizons. These results were obtained within the confines of a commercial, off-the-shelf software for IDs.

8 Related work

Dynamic influence diagrams [44] occupy an important place in the spectrum of formalisms available for modeling and solving sequential decision-making problems. I-DIDs generalize

the influence diagrams to multiagent settings facilitating decision making in the presence of other sophisticated decision makers of uncertain types. They may be viewed as graphical counterparts of finitely nested interactive partially observable Markov decision processes (POMDP) [20].

Other prominent generalizations of IDs to multiagent environments include multiagent influence diagrams (MAIDs) [23,24] and networks of influence diagrams (NIDs) [18,19]. All of these formalisms structure the complex problem domains by decomposing the situation into chance and decision random variables, and the dependencies between the variables. MAIDs objectively analyze the game, efficiently computing the Nash equilibrium profile by exploiting the conditional independence structure. NIDs extend MAIDs to include agents' uncertainty over the game being played and over models of the other agents allowing for nested modeling. Solving NIDs involves transforming them into MAIDs. [17] observe that 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 is incomplete—the solution does not prescribe a policy when the policy followed by the other agent is not part of the equilibrium. Furthermore, the process of equilibration in noncooperative settings continues to remain poorly understood. In comparison with I-DIDs, MAIDs do not allow us to define a distribution over nonequilibrium behaviors of other agents. This is especially problematic when the perspective is that of a decision maker in the presence of others. Furthermore, the applicability of MAIDs and NIDs is limited to static single-shot games. Of course, 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, a predominant factor in the complexity of I-DIDs is due to the exponential growth in the candidate models over time. Using the heuristic that models whose beliefs are spatially close are likely to be BE, [50] employed a k-means approach to cluster models together and selected K representative models in the model node at each time step. However, the K representatives are not guaranteed to be behaviorally distinct due to which the set is not *minimal*. Furthermore, all representatives are expanded to obtain the set of possible models at the next time step before clustering is applied, which may consume excessive memory for storing the models. We may preemptively avoid expanding models that will turn out to be BE to others in the next time step [16,49]. By discriminating between model updates, the approach generates a minimal set of models in each noninitial model node. Another attempt on approximating BE is to identify the BE by comparing probability distributions over the subject agent's action-observation paths, which, however, turns out to be internally contradictory [6,14]. As we mentioned previously, this line of investigation exploits the concept of BE, introduced previously [36,37].

While exploiting BE makes the general space of models parsimonious, identifying that two models are BE requires comparing their solutions, which are often policy trees. The trees grow exponentially in size with the horizon of decision making. In a different approach, [47] sought to cluster models by comparing the K-most probable paths in the policy trees. However, computing path probabilities becomes computationally hard as the paths become longer, and bounding the prediction error is not possible. We may further prune the model space by clustering models whose predicted actions at a particular time step are identical [48,49]. While the clusters may change at each time step, the benefit is that the number of clusters is bounded by the total number of actions of the agent, leading to a small model space. Various

I-DID solution techniques that exploit the notion of equivalence were recently compared [49] and their effectiveness demonstrated on several problem domains including in GaTAC. Building upon the current BE-based I-DID solutions [51,52], this paper formally develops the BE identification techniques and further improves the solutions by approximating the techniques. The solution quality is proved in a theoretical way. We conduct additional results in a larger testbed as well as in one new problem domain. Recently, Chen et al. [8] initiate the study of online I-DID solutions by developing true behavior of other agents during their interactions. Ross et al. [10] focus on learning agents behavior from available data, which provide prior knowledge on refining model space in I-DIDs.

While graphical models remain as yet unexplored in the context of cooperative decision making modeled using frameworks such as decentralized POMDPs [40], factored representations of the state space are becoming prevalent. [32] demonstrated that factored representations of the state space provided a speedup of about two orders of magnitude while solving small team problems exactly, because the representations facilitated exploiting conditional independence. Such factored representations also facilitate solutions to decentralized POMDPs with many agents by exploiting the interaction structure among the agents [30]. Another approach [33] utilizes factored representations in a dynamic Bayesian network to project agents' beliefs forward and utilizes expectation maximization to learn stochastic finite-state controllers. Factored representations for decentralized POMDPs operate on a common initial belief over the state space variables for all agents, and a common reward function. Meanwhile, [45] used influence-based abstraction to decouple local agents' interactions in decentralized POMDPs, which is further generalized to quantify the complexity of multiagent planning [31].

9 Discussion

We show how we may utilize partial solutions of models to determine approximate BE and applied it to significantly scale solutions of I-DIDs. Our insight is that comparing partial solutions of models is likely sufficient for grouping models similarly to using exact BE, as our experiments indicate. We use a principled technique to determine the partialness given the approximation measure, though not all problem domains may allow this.

While we demonstrate the utility of approximate BE in the context of a decision-making framework, it may serve to make the model space tractable for game playing, user modeling, and plan recognition as well. Approaches in these areas of multiagent systems confront large model spaces, which are often trimmed in an ad hoc manner using either data, domain knowledge, or restrictive assumptions. [36] demonstrated an application of BE in a social simulation setting related to class bullying. Here, both the teacher and the bully maintain several mental models of each other's possible behaviors.

I-DIDs allow the model frames to differ as well, thereby permitting the modeling of situations where the subject agent believes that the other agent may have a different model of the decision-making problem. The minimal mixing rate may then differ for models with differing frames and would need to be recomputed. Despite the multiple mixing rates and therefore multiple depths down to which the policy trees must be identical, we may continue to partition the model space using approximate BE.

Given its efficacy in two-agent settings, our immediate line of future work is to evaluate this approach for $N > 2$ agents. In general, the model space grows exponentially with the number of agents. While considerations of exact and approximate BE will continue to

compact the individual agents' model spaces as we show in this article, the minimization may be very effective for anonymous games [12]. These are decision-making contexts where the focus is on the actions performed by the other agents, without identifying which agent in particular or how many agents performed an action. In this case, we may apply BE to compact the collective model space of all agents.

Acknowledgments This research is in part supported by NSFC 61375070, 61562033, 61502322. Yinghui would like to thank the Grant 20151BAB207021 and 20151BDH80014 from Jiangxi Province, China. Prashant would like to thank the support from a NSF CAREER Grant IIS-0845036 and a Grant from ONR N000141310870.

Appendix 1: Proofs of propositions

We begin by proving the bound in Proposition 6. We may evaluate the error, ρ, as:

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

Here, R^{max}_j and R^{min}_j are the maximum and minimum rewards of j, respectively.

Appendix 2: Problem domains

Detailed descriptions of all the problem domains utilized in our evaluations, including their I-DID models, are given in "Multiagent tiger problem" section to "UAV reconnaissance and interception problem" section of Appendix.

Multiagent tiger problem

As we mentioned previously, our multiagent tiger problem is a noncooperative generalization of the well-known single-agent tiger problem [22] to the multiagent setting. It differs from other multiagent versions of the same problem [28] by assuming that the agents hear creaks as well as the growls and the reward function does not promote cooperation. Creaks are indicative of which door was opened by the other agent(s). While we described the problem in Sect. 2, we quantify the different uncertainties here. We assume that the accuracy of creaks is 90 %, while the accuracy of growls is 85 % as in the single-agent problem. The tiger location is chosen randomly in the next time step if any of the agents opened any doors in the current

Table 3 CPT of the chance node $TigerLocation^{t+1}$ in the I-DID of Fig. 7

$\langle a_i^t, a_j^t \rangle$	TigerLocation^t	TL	TR
$\langle OL, * \rangle$	*	0.5	0.5
$\langle OR, * \rangle$	*	0.5	0.5
$\langle *, OL \rangle$	*	0.5	0.5
$\langle *, OR \rangle$	*	0.5	0.5
$\langle L, L \rangle$	TL	1.0	0
$\langle L, L \rangle$	TR	0	1.0

Table 4 CPT of the chance node, $Growl\&Creak^{t+1}$, in agent i's I-DID

$\langle a_i^t, a_j^t \rangle$	TgrLoc^{t+1}	⟨GL, CL⟩	⟨GL, CR⟩	⟨GL, S⟩	⟨GR, CL⟩	⟨GR, CR⟩	⟨GR, S⟩
$\langle L, L \rangle$	TL	0.85*0.05	0.85*0.05	0.85*0.9	0.15*0.05	0.15*0.05	0.15*0.9
$\langle L, L \rangle$	TR	0.15*0.05	0.15*0.05	0.15*0.9	0.85*0.05	0.85*0.05	0.85*0.9
$\langle L, OL \rangle$	TL	0.85*0.9	0.85*0.05	0.85*0.05	0.15*0.9	0.15*0.05	0.15*0.05
$\langle L, OL \rangle$	TR	0.15*0.9	0.15*0.05	0.15*0.05	0.85*0.9	0.85*0.05	0.85*0.05
$\langle L, OR \rangle$	TL	0.85*0.05	0.85*0.9	0.85*0.05	0.15*0.05	0.15*0.9	0.15*0.05
$\langle L, OR \rangle$	TR	0.15*0.05	0.15*0.9	0.15*0.05	0.85*0.05	0.85*0.9	0.85*0.05
$\langle OL, * \rangle$	*	1/6	1/6	1/6	1/6	1/6	1/6
$\langle OR, * \rangle$	*	1/6	1/6	1/6	1/6	1/6	1/6

step. Figure 7 shows an I-DID unrolled over two time-slices for the multiagent tiger problem. We give the CPTs for the different nodes below:

We assign the marginal distribution over the tiger's location from agent i's initial belief to the chance node, $TigerLocation^t$. The CPT of $TigerLocation^{t+1}$ in the next time step conditioned on $TigerLocation^t$, A_i^t, and A_j^t is the transition function, shown in Table 3. The CPT of the observation node, $Growl\&Creak^{t+1}$, is shown in Table 4. CPTs of the observation nodes in level 0 DIDs are identical to the observation function in the single-agent tiger problem.

Decision nodes, A_i^t and A_i^{t+1}, contain possible actions of agent i such as L, OL, and OR. Model node, $M_{j,l-1}^t$, contains the different models of agent j which are DIDs if the I-DID is at level 0, otherwise they are I-DIDs themselves. The distribution over the associated $Mod[M_j^t]$ node (see Fig. 8) is the conditional distribution over j's models given physical state from agent i's initial belief. The CPT of the chance node, $Mod[M_j^{t+1}]$, in the model node, $M_{j,l-1}^{t+1}$, reflects which prior model, action and observation of j results in a model contained in the model node.

Finally, the utility node, R_i, in the I-DID relies on both agents' actions, A_i^t and A_j^t, and the physical states, $TigerLocation^t$. The utility table is shown in Table 5. These payoffs are analogous to the single-agent version, which assigns a reward of 10 if the correct door is opened, a penalty of 100 if the opened door is the one behind which is a tiger, and a penalty of 1 for listening. A result of this assumption is that the other agent's actions do not impact the original agent's payoffs directly, but rather indirectly by resulting in states that matter to the original agent. The utility tables for level 0 models are exactly identical to the reward function in the single-agent tiger problem.

Table 5 Utility table for node, R_i, in the I-DID

$\langle a_i, a_j \rangle$	TL	TR
$\langle OR, OR \rangle$	10	−100
$\langle OL, OL \rangle$	−100	10
$\langle OR, OL \rangle$	10	−100
$\langle OL, OR \rangle$	−100	10
$\langle L, L \rangle$	−1	−1
$\langle L, OR \rangle$	−1	−1
$\langle OR, L \rangle$	10	−100
$\langle L, OL \rangle$	−1	−1
$\langle OL, L \rangle$	−100	10

Utility table in the I-DID for agent j is the same with column label, $\langle a_i, a_j \rangle$, swapped

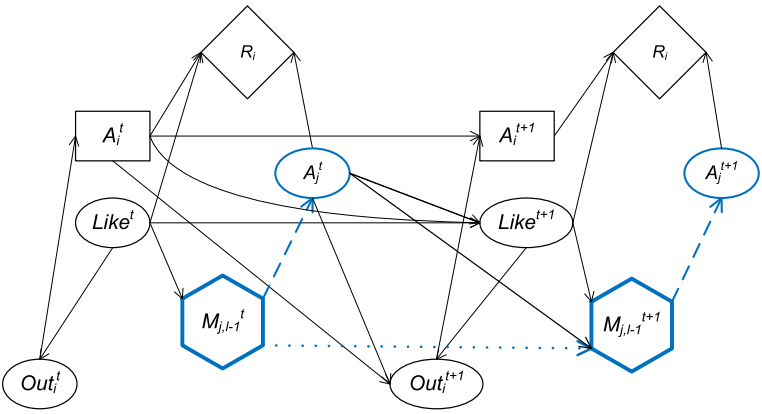


Fig. 25 Level l I-DID for concert i in the multiagent concert problem

Multiagent concert problem

We extend the single-agent concert problem available in the POMDP repository[3] to a two-agent setting. The problem involves a concert organizer who must decide whether to advertise the concert on TV, over the radio, or do nothing. The problem is inspired by real-world marketing problems involving multiple brands, changing attitudes about brands and the effect of advertising [26].

In the multiagent concert problem, two separate concerts are involved, each of which may be advertised on TV (we denote this action as TV), over the radio (denoted as $Radio$), or none (denoted as $Nothing$). The state of this problem is two different attitudes or predispositions that the target audience may have about both the concerts in general: They may be interested in them (denoted as I) or bored with them (denoted as B). The output of the actions could make the target audience definitely want to attend a particular concert (we denote this observation as Go), may attend the concert (denoted as $MayGo$), may not attend it (denoted as $MayNoGo$), or definitely not want to attend the concert ($NoGo$).

Figure 25 shows a level l I-DID unrolled over two time-slices for the multiagent concern domain.

[3] http://www.pomdp.org/pomdp/examples/index.shtml.

Table 6 CPT of the chance node, $Like^{t+1}$, in agent i's I-DID of Fig. 25

$\langle a_i^t, a_j^t \rangle$	Like^t	Interested	Bored
$\langle TV, TV \rangle$	I	0.90	0.10
$\langle TV, TV \rangle$	B	0.60	0.40
$\langle TV, Radio \rangle$	I	0.85	0.15
$\langle TV, Radio \rangle$	B	0.45	0.55
$\langle TV, Nothing \rangle$	I	0.70	0.30
$\langle TV, Nothing \rangle$	B	0.35	0.65
$\langle Radio, TV \rangle$	I	0.85	0.15
$\langle Radio, TV \rangle$	B	0.45	0.55
$\langle Radio, Radio \rangle$	I	0.80	0.20
$\langle Radio, Radio \rangle$	B	0.30	0.70
$\langle Radio, Nothing \rangle$	I	0.65	0.35
$\langle Radio, Nothing \rangle$	B	0.20	0.80
$\langle Nothing, TV \rangle$	I	0.70	0.30
$\langle Nothing, TV \rangle$	B	0.15	0.85
$\langle Nothing, Radio \rangle$	I	0.65	0.35
$\langle Nothing, Radio \rangle$	B	0.20	0.80
$\langle Nothing, Nothing \rangle$	I	0.50	0.50
$\langle Nothing, Nothing \rangle$	B	0.10	0.90

The decision node, A_i, contains the possible marketing actions for concert i, such as TV, $Radio$, or $Nothing$. The chance node, $Like$, represents audience attitude toward concerts. As attitudes vary in general, we begin with a uniform distribution over them as the initial belief. We show the CPT of $Like^{t+1}$, which is the transition function, in Table 6. It models the fact that a TV marketing campaign may change the attitudes with a higher probability or maintaining interest compared to a radio campaign, while doing nothing may have an adverse impact.

The observation node, Out_i, models the observed indications of the target audience toward going out to attend the concert, i, through the values, Go, $MayGo$, $MayNoGo$, and $NoGo$. The CPT of the node, Out_i^{t+1}, is shown below table 7 and models the notion that TV advertisements would be more effective in translating the predispositions and making the target audience want to attend the conference even if they are bored, compared to radio advertisements. On the other hand, doing nothing does not have much effect and would result in a direct translation of predispositions to wanting to attend the conference or not.

Finally, we show the reward function in the utility node, R_i, in the I-DID table 8. The rewards combine the cost of the different marketing campaigns with TV being most expensive, and a quantified efficacy of the different campaigns with TV being most effective. We show the reward function in Fig. 8.

Money laundering problems

As [29] mention, money laundering is a process of transferring "dirty" money to "clean" money through a series of criminal transactions. It normally contains three steps, namely $placement$, $layering$, and $integrating$. In the placement phase, money launderers introduce the dirty money into some common targets of financial systems like bank accounts, insurance, and securities. Then, in the layering phase, they transfer the money into some businesses like

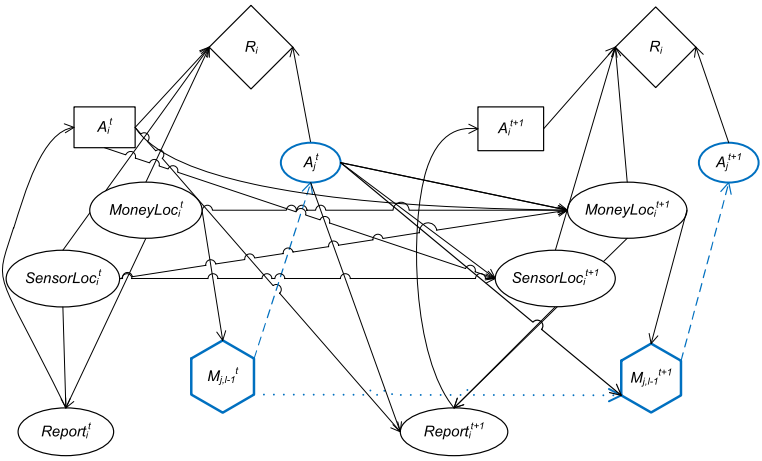


Fig. 26 Level l I-DID of agent i for the money laundering problem

trusts, offshore accounts and shell companies. The transactions may obscure the money source. Finally, in the integration phase, the money launderers involve the laundered money into more legitimate businesses including real estate, loans, and casinos. On the other hand, as an anti-money laundering body, law enforcement monitors the money laundering flow by placing physical sensors at each possible location of dirty money. It analyzes the received information and accordingly confiscates the dirty money once it correctly identifies the money location.

The law enforcement and money launderers are denoted as the *blue* team (agent i) and red team (agent j), respectively, in the problem domain. The blue team is represented as a level l I-DID shown in Fig. 26, while the red team is at level $l-1$. The joint state of level l I-DID contains both money locations of the red team (11 possible states), ML^t, and sensor locations installed by the blue team (9 possible states), SL^t. The blue team has 9 possible actions in the decision node, A_i^t, including the placement of possible sensors and the confiscation of the dirty money. The CPTs of chance nodes, $MoneyLoc^{t+1}$ and $SensorLoc^{t+1}$, encode the probabilities of the sensor installation in possible money locations. In particular, only when the blue team places the sensors in the same location as where the dirty money is transferred, it confiscates the dirty money and resumes its states.

The blue team receives observations in terms of reports generated from most of the installed sensors. The chance node, $Report_i^{t+1}$, has 9 states and its CPT provides the sensing capability of the blue team. On average, blue team correctly detects the real location of the dirty money 80 % of the times given a positive report on the location.

The utility node, R_i, is the reward assigned to the blue team when the agent acts at the joint state. The blue team gets 100 if it confiscates dirty money while it costs -10 for placing any sensor in the targeted location. The actual CPT tables are large and we do not show them here.

UAV reconnaissance and interception problem

We show a level l I-DID for the multiagent UAV problem in Fig. 27. Models of agent j, which may play the role of a fugitive or a hostile UAV J at the lower level differ in the probability that the fugitive assigns to its position in the grid. The UAV's (agent i) initial beliefs are

Table 7 CPT of the chance node, Out^{t+1}, in concert i's I-DID. The CPT of the corresponding node in concert j's I-DID is similar with the joint actions reversed

$\langle a_i^t, a_j^t \rangle$	Like^{t+1}	$\langle Go \rangle$	$\langle MayGo \rangle$	$\langle MayNoGo \rangle$	$\langle NoGo \rangle$
$\langle TV, TV \rangle$	I	0.64	0.16	0.16	0.04
$\langle TV, TV \rangle$	B	0.49	0.21	0.21	0.09
$\langle TV, Radio \rangle$	I	0.56	0.24	0.14	0.06
$\langle TV, Radio \rangle$	B	0.16	0.24	0.24	0.36
$\langle TV, Nothing \rangle$	I	0.72	0.08	0.18	0.02
$\langle TV, Nothing \rangle$	B	0.07	0.63	0.03	0.27
$\langle Radio, TV \rangle$	I	0.56	0.24	0.14	0.06
$\langle Radio, TV \rangle$	B	0.28	0.12	0.42	0.18
$\langle Radio, Radio \rangle$	I	0.49	0.21	0.21	0.09
$\langle Radio, Radio \rangle$	B	0.16	0.24	0.24	0.36
$\langle Radio, Nothing \rangle$	I	0.63	0.07	0.27	0.32
$\langle Radio, Nothing \rangle$	B	0.04	0.36	0.06	0.54
$\langle Nothing, TV \rangle$	I	0.72	0.18	0.08	0.02
$\langle Nothing, TV \rangle$	B	0.07	0.63	0.03	0.27
$\langle Nothing, Radio \rangle$	I	0.63	0.27	0.03	0.07
$\langle Nothing, Radio \rangle$	B	0.04	0.06	0.36	0.54
$\langle Nothing, Nothing \rangle$	I	0.81	0.09	0.09	0.01
$\langle Nothing, Nothing \rangle$	B	0.01	0.09	0.09	0.81

Table 8 Utility table for node, R_i, in the I-DID

$\langle a_i, a_j \rangle$	I	B
$\langle TV, TV \rangle$	4	2
$\langle TV, Radio \rangle$	2	−5
$\langle TV, Nothing \rangle$	2	0
$\langle Radio, TV \rangle$	2	−5
$\langle Radio, Radio \rangle$	8	−2.5
$\langle Radio, Nothing \rangle$	3	−5
$\langle Nothing, TV \rangle$	2	0
$\langle Nothing, Radio \rangle$	3	−5
$\langle Nothing, Nothing \rangle$	6	0

Note that the utility table is symmetric over the joint actions

probability distributions assigned to the relative position of the fugitive decomposed into the chance nodes, $FugRelPosX^t$ and $FugRelPosY^t$, which represent the relative location of the fugitive along the row and column, respectively. Its CPTs assume that each action (except *listen*) moves the UAV in the intended direction with a probability of 0.67, while the remaining probability is equally divided among the other neighboring positions. Action *listen* keeps the UAV in the same position.

The observation node, $SenFug$, represents the UAV's sensing of the relative position of the fugitive in the grid. Its CPT assumes that the UAV has good sensing capability (likelihood

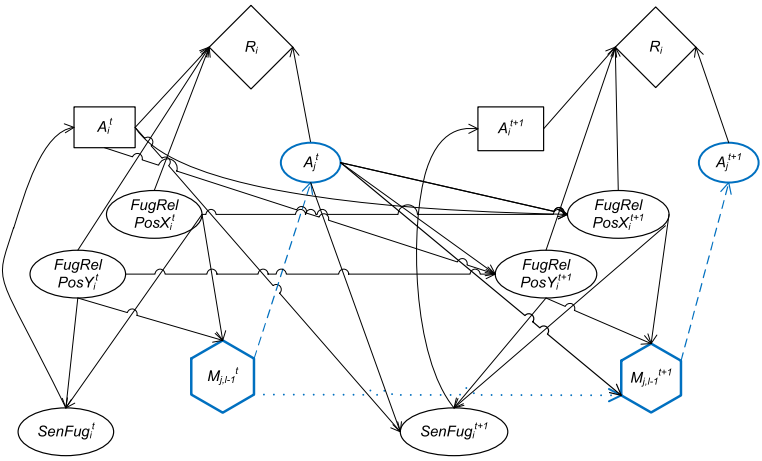


Fig. 27 Level l I-DID of agent i for our UAV reconnaissance problem

of 0.8 for the correct relative location of the fugitive) if the action is *listen*, otherwise the UAV receives random observations during other actions.

The decision node, A_i, contains five actions of the UAV, which includes moving in the four cardinal directions and listening. The edge incident into the node indicates that the UAV ascertains the observation on the relative position of the fugitive before it takes an action.

The utility node, R_i, is the reward assigned to the UAV for its actions given the fugitive's relative position and its actions. The UAV gets rewarded 50 if it captures the fugitive; otherwise, it costs -5 for performing any other action.

Because the actual CPT tables are very large, we do not show them here. All problem domain files are available upon request.

References

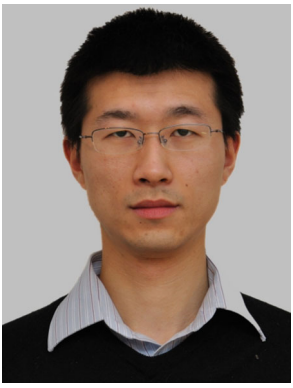
1. Adam B, Dekel E (1993) Hierarchies of beliefs and common knowledge. Int J Game Theory 59(1):189–198
2. Andersen S, Jensen F (1989) Hugin: a shell for building belief universes for expert systems. In: International joint conference on artificial intelligence (IJCAI), pp 332–337
3. Aumann RJ (1999) Interactive epistemology i: Knowledge. Int J Game Theory 28(3):263–300
4. Bernstein DS, Givan R, Immerman N, Zilberstein S (2002) The complexity of decentralized control of Markov decision processes. Math. Oper. Res. 27(4):819–840
5. Boyen X, Koller D (1998) Tractable inference for complex stochastic processes. In: The 14th conference on uncertainty in artificial intelligence (UAI), pp 33–42
6. Chandrasekaran M, Doshi P, Zeng Y (2010) Approximate solutions of interactive dynamic influence diagrams using ϵ-behavioral equivalence. In: International symposium on artificial intelligence and mathematics (ISAIM)
7. Chandrasekaran M, Doshi P, Zeng Y, Chen Y (2014) Team behavior in interactive dynamic influence diagrams with applications to ad hoc teams. In: Proceedings of the seventh international conference on autonomous systems and multiagent systems (AAMAS), pp 1559–1560
8. Chen Y, Doshi P, Zeng Y (2015) Iterative online planning in multiagent settings with limited model spaces and pac guarantees. In: Proceedings of the seventh international conference on autonomous systems and multiagent systems (AAMAS), pp 1161–1169
9. Chen Y, Hong J, Liu W, Godo L, Sierra C, Loughlin M (2013) Incorporating pgms into a bdi architecture. In: 16th international conference on principles and practice of multi-agent systems (PRIMA), pp 54–69

10. Conroy R, Zeng Y, Cavazza M, Chen Y (2015) Learning behaviors in agents systems with interactive dynamic influence diagrams. In: Proceedings of international joint conference on artificial intelligence (IJCAI), pp 39–45
11. Cover T, Thomas J (1991) Elements of information theory. Wiley, New York
12. Daskalakis C, Papadimitriou C (2007) Computing equilibria in anonymous games. In: 48th annual ieee symposium on foundations of computer science (FOCS), pp 83–93
13. Dekel E, Fudenberg D, Morris S (2006) Topologies on types. Theor Econ 1:275–309
14. Doshi P, Chandrasekaran M, Zeng Y (2010) Epsilon-subjective equivalence of models for interactive dynamic influence diagrams. In: WIC/ACM/IEEE conference on web intelligence and intelligent agent technology (WI-IAT), pp 165–172
15. Doshi P, Sonu E (2010) GaTAC: a scalable and realistic testbed for multiagent decision making. In: Fifth workshop on multiagent sequential decision making in uncertain domains (MSDM). AAMAS, pp 62–66
16. Doshi P, Zeng Y (2009) Improved approximation of interactive dynamic influence diagrams using discriminative model updates. In: International conference on autonomous agents and multi-agent systems (AAMAS). pp 907–914
17. Doshi P, Zeng Y, Chen Q (2009) Graphical models for interactive POMDPs: representations and solutions. J Auton Agents Multi-Agent Syst JAAMAS 18(3):376–416
18. Gal K, Pfeffer A (2008) Networks of influence diagrams: a formalism for representing agents' beliefs and decision-making processes. J Artif Intell Res 33:109–147
19. Gal Y, Pfeffer A (2003) A language for modeling agent's decision-making processes in games. In: Autonomous agents and multi-agents systems conference (AAMAS), pp 265–272
20. Gmytrasiewicz P, Doshi P (2005) A framework for sequential planning in multiagent settings. J Artif Intell Res JAIR 24:49–79
21. Howard RA, Matheson JE (1984) Influence diagrams. In: Howard RA, Matheson JE (eds) Readings on the principles and applications of decision analysis, vol 2. Strategic Decisions Group, Menlo Park, pp 719–762
22. Kaelbling L, Littman M, Cassandra A (1998) Planning and acting in partially observable stochastic domains. Artif Intell J 101:99–134
23. Koller D, Milch B (2001) Multi-agent influence diagrams for representing and solving games. In: International joint conference on artificial intelligence (IJCAI), pp 1027–1034
24. Koller D, Milch B (2011) Multi-agent influence diagrams for representing and solving games. Games Econ Behav 45(1):181–221
25. Lauritzen SL, Nilsson D (2001) Representing and solving decision problems with limited information. Manag Sci 47:1235–1251
26. Lipstein B (1965) A mathematical model of consumer behavior. J Mark 2:259–265
27. Luo J, Yin H, Li B, Wu C (2011) Path planning for automated guided vehicles system via interactive dynamic influence diagrams with communication. In: 9th IEEE international conference on control and automation (ICCA), pp 755–759
28. Nair R, Tambe M, Yokoo M, Pynadath D, Marsella S (2003) Taming decentralized POMDPs: towards efficient policy computation for multiagent settings. In: International joint conference on artificial intelligence (IJCAI), pp 705–711
29. Ng B, Meyers C, Boakye K, Nitao J (2010) Towards applying interactive POMDPs to real-world adversary modeling. In: Innovative applications in artificial intelligence (IAAI), pp 1814–1820
30. Oliehoek FA, Whiteson S, Spaan MT (2013) Approximate solutions for factored dec-pomdps with many agents. In: Proceedings of the 2013 international conference on autonomous agents and multi-agent systems (AAMAS). pp. 563–570
31. Oliehoek FA, Witwicki SJ, Kaelbling LP (2012) Influence-based abstraction for multiagent systems. In: Twenty-sixth AAAI conference on artificial intelligence (AAAI), pp 1422–1428
32. Oliehoek F, Spaan M, Whiteson S, Vlassis N (2008) Exploiting locality of interaction in factored Dec-POMDPs. In: Seventh international conference on autonomous agents and multiagent systems (AAMAS), pp 517–524
33. Pajarinen J, Peltonen J (2011) Efficient planning for factored infinite-horizon DEC-POMDPs. In: International joint conference on artificial intelligence (IJCAI), pp 325–331
34. Perry AR (2004) The flightgear flight simulator. In: UseLinux. http://www.flightgear.org
35. Pineau J, Gordon G, Thrun S (2006) Anytime point-based value iteration for large POMDPs. J Artif Intell Res 27:335–380
36. Pynadath D, Marsella S (2007) Minimal mental models. In: Twenty-second conference on artificial intelligence (AAAI). Vancouver, Canada, pp 1038–1044

37. Rathnasabapathy B, Doshi P, Gmytrasiewicz PJ (2006) Exact solutions to interactive POMDPs using behavioral equivalence. In: Autonomous agents and multi-agents systems conference (AAMAS), pp 1025–1032
38. Russell S, Norvig P (2010) Artificial intelligence: a modern approach, 3rd edn. Prentice Hall, Englewood Cliffs
39. Seuken S, Zilberstein S (2008) Formal models and algorithms for decentralized decision making under uncertainty. Auton Agents Multi-Agent Syst 17(2):190–250
40. Seuken S, Zilberstein S (2008) Formal models and algorithms for decentralized decision making under uncertainty. J Auton Agents Multi-agent Syst
41. Shachter RD (1986) Evaluating influence diagrams. Oper Res 34(6):871–882
42. Smallwood R, Sondik E (1973) The optimal control of partially observable Markov decision processes over a finite horizon. Oper Res OR 21:1071–1088
43. Sonu E, Doshi P (2012) GaTAC: A scalable and realistic testbed for multiagent decision making (demonstration). In: Eleventh international conference on autonomous agents and multiagent systems (AAMAS), DEMO track, pp 1507–1508
44. Tatman JA, Shachter RD (1990) Dynamic programming and influence diagrams. IEEE Trans Syst Man Cybern 20(2):365–379
45. Witwicki SJ, Durfee EH (2010) Influence-based policy abstraction for weakly-coupled dec-pomdps. In: International conference on automated planning and scheduling (ICAPS), pp 185–192
46. Woodberry O, Mascaro S (2012) Programming Bayesian network solutions with netica. Bayesian Intelligence, Brookvale
47. Zeng Y, Chen Y, Doshi P (2011) Approximating behavioral equivalence of models using top-k policy paths (extended abstract). In: International conference on autonomous agents and multi-agent systems (AAMAS), pp 1229–1230
48. Zeng Y, Doshi P (2009) Speeding up exact solutions of interactive influence diagrams using action equivalence. In: International joint conference on artificial intelligence (IJCAI)
49. Zeng Y, Doshi P (2012) Exploiting model equivalences for solving interactive dynamic influence diagrams. J Artif Intell Res JAIR 43:211–255
50. Zeng Y, Doshi P, Chen Q (2007) Approximate solutions of interactive dynamic influence diagrams using model clustering. In: Twenty second conference on artificial intelligence (AAAI). Vancouver, Canada, pp 782–787
51. Zeng Y, Doshi P, Pan Y, Mao H, Chandrasekaran M, Luo J (2011) Utilizing partial policies for identifying equivalence of behavioral models. In: Twenty-fifth AAAI conference on artificial intelligence, pp 1083–1088
52. Zeng Y, Pan Y, Mao H, Luo J (2012) Improved use of partial policies for identifying behavioral equivalences. In: Eleventh international conference on autonomous agents and multiagent systems (AAMAS), pp 1015–1022

Yifeng Zeng is a Reader at School of Computing in Teesside University, UK. He received his PhD in 2006 from National University of Singapore, Singapore. Before he moved to Teesside University, Dr. Zeng was an assistant professor and an associate professor during 2006–2012 in Aalborg University, Denmark. He is also an affiliate professor in Xiamen University, China. His current research interests include intelligent agents, decision making, social networks, and computer games. Most of his publications appear in the most prestigious international academic journals and conferences including Journal of Artificial Intelligence Research (JAIR), Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS), IEEE International Conference on Data Mining (ICDM), AAMAS, International Joint Conference on Artificial Intelligence (IJCAI), and Association for the Advancement of Artificial Intelligence (AAAI).

Prashant Doshi is an Associate Professor of Computer Science at The University of Georgia, USA, and founding director of the Faculty of Robotics at UGA (an OVPR initiative). His research interests lie in artificial intelligence and robotics, and specifically in decision making under uncertainty in multiagent settings, game theory, and robot learning. He was a visiting professor at the University of Waterloo in 2015. He has published over 80 papers in journals, conferences, and other forums in the fields of agents and AI. His research has led to publications in the Journal of AI Research, AAMAS, AAAI, and IJCAI conferences among others. Prof. Doshi has taught introductory courses on AI to undergraduate and graduate students for more than 10 years, and a course and conference tutorial on decision making under uncertainty for several semesters, all of which were well received by the students.



Yingke Chen is a Lecture at College of Computer Science in Sichuan University, China. He received his PhD in 2013 from Aalborg University, Denmark. Dr. Chen was a post-doctoral research associate in Queen's University Belfast, UK, and Georgia University, USA, respectively, before joining Sichuan University. His current research interests include intelligent agents, decision making, and their applications in autonomous systems.

Yinghui Pan is a Lecture at Department of Information Management in Jiangxi University of Finance and Economics, China. She received her PhD in 2012 from Xiamen University, China. Her current research interests include intelligent agents and decision making. Most of her publications appear in Journal of Approximate Reasoning, Agent and Data Mining Interaction, AAMAS, IEEE/WIC/ACM International Conference on Intelligent Agent Technology.

Hua Mao is a Lecture at College of Computer Science in Sichuan University, China. She received her PhD in 2013 from Aalborg University, Denmark. Dr. Mao was a research fellow in Queen's University Belfast, UK, before joining Sichuan University. Her current research interests include machine learning and its applications in the context of Big Data.

Muthukumaran Chandrasekaran is currently pursuing his PhD in Computer Science at the University of Georgia, USA. He is a Research Assistant under the supervision of Dr. Prashant Doshi at the THINC Lab. He received his Master degree in Artificial Intelligence, also from the University of Georgia, in 2010. Prior to that, he completed his Bachelor degree in Mechatronics Engineering from SASTRA University, India, in 2007. His current research interests include multiagent planning and decision making, open-agent systems, ad hoc teamwork, graphical models, and game theory. He has been serving as a peer-reviewer for top international academic conferences including AAMAS, IJCAI, and AAAI since 2009. He loves teaching and is passionate about interdisciplinary research and entrepreneurship.