Auton Agent Multi-Agent Syst (2009) 18:376–416
DOI 10.1007/s10458-008-9064-7

Graphical models for interactive POMDPs: representations and solutions

Prashant Doshi · Yifeng Zeng · Qiongyu Chen

Published online: 25 September 2008

Abstract We develop new graphical representations for the problem of sequential decision making in partially observable *multiagent* environments, as formalized by interactive partially observable Markov decision processes (I-POMDPs). The graphical models called *interactive influence diagrams (I-IDs)* and their dynamic counterparts, *interactive dynamic influence diagrams (I-DIDs)*, seek to explicitly model the structure that is often present in real-world problems by decomposing the situation into chance and decision variables, and the dependencies between the variables. I-DIDs generalize DIDs, which may be viewed as graphical representations of POMDPs, to multiagent settings in the same way that I-POMDPs generalize POMDPs. I-DIDs may be used to compute the policy of an agent given its belief as the agent acts and observes in a setting that is populated by other interacting agents. Using several examples, we show how I-IDs and I-DIDs may be applied and demonstrate their usefulness. We also show how the models may be solved using the standard algorithms that are applicable to DIDs. Solving I-DIDs exactly involves knowing the solutions of possible models of the other agents. The space of models grows exponentially with the number of time steps. We present a method of solving I-DIDs approximately by limiting the number of other agents' candidate models at each time step to a constant. We do this by clustering models that are likely to be behaviorally equivalent and selecting a representative set from the clusters. We discuss the error bound of the approximation technique and demonstrate its empirical performance.

P. Doshi (✉)
Department of Computer Science and Institute for AI, University of Georgia, Athens, GA 30602, USA
e-mail: pdoshi@cs.uga.edu

Y. Zeng
Department of Computer Science, Aalborg University, 9220 Aalborg, Denmark
e-mail: yfzeng@cs.aau.dk

Q. Chen
Department of Computer Science, National University of Singapore, Singapore, Singapore 117543

Keywords Probabilistic graphical models · Interactive POMDPs · Sequential multiagent decision making

1 Introduction

Interactive partially observable Markov decision processes (I-POMDP) [14] provide a framework for sequential decision making in partially observable *multiagent* environments. They generalize POMDPs [19, 34] to multiagent settings by including other agents' computable models in the state space along with the states of the physical environment. The models encompass all information influencing the agents' behaviors, including their preferences, capabilities, and beliefs, and are thus analogous to *types* in Bayesian games as first envisioned by Harsanyi [17]. I-POMDPs adopt a *subjective* approach to understanding strategic behavior, rooted in a decision-theoretic framework that takes a decision-maker's perspective in the interaction.

Enumerative representations of models often obscure important structure that is typically present in many realistic application settings. Graphical models, such as *influence diagrams* (ID) [33, 36] offer a qualitative language that decomposes the state into *chance* (random) variables and dependencies between the variables. Algorithms for solving the models exploit the conditional independence between variables, and often consume less time and space in solving the problem compared to those that operate on traditional enumerative representations. As a case in point, *factored* representations of POMDPs (and MDPs) that utilize IDs often facilitate fast solutions that exploit the structure (see [4, 16] for examples). Graphical models also allow a more explicit qualitative description of the decision-making situation as compared to enumerative forms.

In order to provide a graphical representation for I-POMDPs and make the structure explicit, Polich and Gmytrasiewicz [28] introduced a novel graphical model, called *interactive dynamic influence diagram (I-DID)*. I-DIDs may be viewed as graphical representations of I-POMDPs. They generalize DIDs (dynamic IDs), which are graphical counterparts of POMDPs, to multiagent settings in the same way that I-POMDPs generalize POMDPs.

In this paper, we significantly improve on the previous preliminary representation of I-DIDs by first introducing static *interactive influence diagrams (I-ID)*, relating them to another multiagent graphical model, network of influence diagrams (NID) [13], and then extending I-IDs to their dynamic counterparts, interactive dynamic influence diagrams (I-DIDs). Analogous to DIDs, I-DIDs compactly represent the decision problem by mapping various variables into chance, decision and utility nodes, and denoting the dependencies between variables using directed arcs between the corresponding nodes. However, matters are more complex when we consider multiagent interactions that are extended over time, where predictions about others' future actions must be made using models that change as the agents act and observe. I-DIDs address this gap by allowing the representation of other agents' models as the values of a special *model node*. Both other agents' models and the original agent's beliefs over these models are updated over time using special-purpose implementations. Specifically, the update of the agent's belief over the models of others as the agents act and receive observations is denoted using a special link called the *model update link* that connects the model nodes between time steps.

To facilitate understanding, we explicate the semantics of the model node and the model update link by showing how they can be implemented using the traditional dependency links between the chance nodes that constitute the model nodes. The net result is a representation of I-DID that is transparent and semantically clear in comparison to [28], and capable of

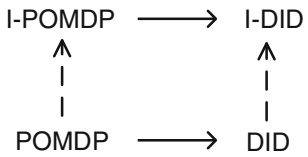

Fig. 1 The relationship between the four representations along two dimensions. The vertical dimension (dashed arrows) specifies the generalization from the single agent to the multiagent setting, while the horizontal dimension (solid arrows) is the mapping from the enumerative to the graphical representation

being implemented using the standard algorithms for solving DIDs. We show how I-DIDs may be used to model an agent's uncertainty over others' models that may themselves be I-DIDs leading to recursive modeling. Solution to the I-DID is a policy that prescribes what the agent should do over time, given its beliefs over the physical state and others' models. Analogous to DIDs, I-DIDs may be used to compute the policy of an agent *online*—given an initial belief of the agent—as the agent acts and observes in a setting that is populated by other interacting agents. We also explain how elements of the I-DID map to the enumerative representation of I-POMDP. Additionally, we illustrate their computational advantages in domains where structure can be exploited.

In Fig. 1, we summarize the relationship along two dimensions between the different formalisms that we mention in this paper. Specifically, I-DIDs generalize DIDs to multiagent settings analogously to the way by which I-POMDPs generalize POMDPs. Additionally, I-DIDs provide a graphical counterpart to the enumerative representation of I-POMDPs similar to how DIDs are graphical counterparts of POMDPs.

As we may expect, I-DIDs acutely suffer from both the curses of dimensionality and history [27]. This is because the state space in I-DIDs includes the models of other agents in addition to the traditional physical states. As the agents act, observe, and update beliefs, I-DIDs must track the evolution of the models over time. Often, the number of candidate models grows exponentially over time. Consequently, I-DIDs not only suffer from the curse of history that afflicts the modeling agent, but also from that exhibited by the modeled agents. This is further complicated by the nested nature of the state space.

In this article, we also present a method of reducing the dimensionality of the interactive state space and mitigate the impact of the curse of history that afflicts the modeled agents. Our method limits and holds constant the number of models, $0 < K \ll M$, where M is the possibly large number of candidate models, of the other agents included in the state space.

Using the insight that beliefs that are spatially close are likely to be behaviorally equivalent [30], our approach is to *cluster* the models of the other agents and select representative models from each cluster. In this regard, we utilize the popular k-means clustering method [22], which gives an iterative way to generate the clusters. Intuitively, the clusters contain models that are likely to be behaviorally equivalent and hence may be replaced by a subset of representative models without a significant loss in the optimality of the decision maker. We select K representative models from the clusters and update them over time.

For the approximation technique, we theoretically bound the worst case error introduced by the approach in the policy of the other agent for two-agent settings and empirically measure its impact on the quality of the policies pursued by the original agent. Our empirical results on two application scenarios—the multiagent tiger and machine maintenance problems—demonstrate the computational savings obtained in solving the I-DIDs and the favorable performances of the approach.

The remainder of this paper is structured as follows. In Sect. 2, we compare and analyze the related work. In Sect. 3, we briefly review the framework of I-POMDPs and influence

diagrams that underlie our work. In Sect. 4, we present the new models of I-IDs and I-DIDs in detail and illustrate them using example applications. In Sect. 5, we present the exact algorithm for solving I-DIDs and discuss example solutions of the illustrative problems. We also demonstrate the computational advantage that I-DIDs offer over the enumerative representations of I-POMDPs. In Sect. 6, we formally propose an approximation technique and discuss the details of its implementation. Furthermore, in Sect. 7, we discuss its computational complexity and provide theoretical error bounds. We then provide, in Sect. 8, experimental results that demonstrate the performance of our approximation technique comparing it with exact solutions with respect to the solution quality and run times. Sect. 9 concludes this paper with a discussion and future lines of work.

2 Related work

Suryadi and Gmytrasiewicz [35] produced an early piece of related work, in which they proposed modeling other agents using IDs. Though IDs (and not DIDs) were used to model other agents, the approach proposed ways to modify the IDs to better reflect the observed behavior. However, unlike I-DIDs, other agents did not model the original agent and the distribution over the models was not updated based on the actions and observations.

I-DIDs contribute to a growing line of work on multiagent decision making that includes multiagent influence diagrams (MAID) [20], and more recently, networks of influence diagrams (NID) [13]. These formalisms seek to explicitly model the structure that is often present in real-world problems by decomposing the situation into chance and decision variables, and the dependencies between the variables. MAIDs provide an alternative to normal and extensive game forms using a graphical formalism to represent games of imperfect information with a decision node for each agent's actions and chance nodes capturing the agent's private information. MAIDs objectively analyze the game, efficiently computing the Nash equilibrium profile by exploiting the independence structure. NIDs extend MAIDs to include agents' uncertainty over the game being played and over models of the other agents. Each model is a MAID and the network of MAIDs is collapsed, bottom up, into a single MAID for computing the equilibrium of the game keeping in mind the different models of each agent.

Graphical formalisms such as MAIDs and NIDs open up a promising area of research that aims to represent multiagent interactions more transparently. However, MAIDs provide an analysis of the game from an external viewpoint and the applicability of both is limited to *static single play* games. The interactions we consider 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 allow the explicit representation of other agents' models as the values of a special *model node*. Other agents' models and the original agent's beliefs over these models are then updated over time.

As we seek a formalism that facilitates planning and problem solving at an agent's own individual level, we extended IDs to the multiagent setting, rather than utilize MAIDs. This is because MAIDs represent multiagent games objectively and facilitate their analysis from an external perspective. They adopt Nash equilibrium as a solution concept. However, equilibrium is not unique—there could be many joint solutions in equilibrium with no clear way for an agent to choose between them—and incomplete—the prescribed policy is not optimal when the policy followed by the other agent is not part of the equilibrium. Specifically, MAIDs do not allow us to define a distribution over non-equilibrium behaviors of other agents. In comparison, I-DIDs provide a way to exploit predicted non-equilibrium behavior.

Thus, MAIDs are not amenable to modeling decision making in multiagent settings from an individual agent's perspective.

In prior work [28], Polich and Gmytrasiewicz introduced I-DIDs as the graphical representations of I-POMDPs. In this article, we significantly improve on their previous preliminary representation of I-DID by using the insight that the static I-ID is a type of NID. Furthermore, we clearly explicate the semantics of the new constructs such as the *model node* and *model update link* by showing how they can be implemented using the traditional chance nodes and dependency links between the chance nodes. Consequently, I-IDs and I-DIDs may be solved using the standard techniques useful in solving IDs and DIDs.

In the context of I-POMDPs, previous solution techniques have focused on their enumerative forms. One such approximation technique [8, 9] reduces the model space complexity by sampling models considered likely by the agent. The models are then propagated over time using a particle filtering technique generalized to multiple agents, called the interactive particle filter. Though applicable in I-DIDs, because the technique does not mitigate the curse of history, it does not provide a way to reduce the exponential growth in the models over time while expanding the I-IDs. As it approximates the belief revision, it finds application only while solving the I-DIDs. However, exponential numbers of models are generated while expanding the I-ID over multiple time steps; thus the technique is less effective in approximating I-DIDs. In addition, because we prune models that are likely to be behaviorally equivalent, our approach results in solutions that are likely to be of similar or better quality given some number of models.

Other principled efforts that generalize decision theory to multiagent systems include Markov games [21], multiagent MDP [3], and decentralized POMDP [24, 32]. All of these assume that the solution, often the equilibrium, is computed centrally and distributed to the agents. Their applicability is limited to fully cooperative settings (teams), in contrast, I-DIDs and I-POMDPs may be used in non-cooperative situations as well.

3 Background

Our work builds on the framework of finitely nested I-POMDPs [14] and generalizes the well-known graphical formalisms of influence diagrams (ID) [18] to multiagent settings. In this section, we briefly review the I-POMDP framework which provides the mathematical foundation for the new graphical models. We then provide a selective overview of IDs referring the reader to [31] for a more introductory description.

3.1 Finitely nested interactive POMDPs

Interactive POMDPs generalize POMDPs to multiagent settings by including other agents' models as part of the state space. Models of other agents include their private information such as beliefs, capabilities, and preferences, and are thus analogous to *types* in Bayesian games [17]. As agents may have beliefs about the models of others, the augmented state space, called the *interactive state space*, is strategically nested—it contains beliefs about other agents' models and their beliefs about others. For the simplicity of presentation let us consider two agents, i and j, which are interacting in a common environment:

Definition 1 (*I-POMDP*$_{i,l}$) A finitely nested I-POMDP of agent i with a strategy level l is

$$\text{I-POMDP}_{i,l} = \langle IS_{i,l}, A, T_i, \Omega_i, O_i, R_i \rangle$$

where:

- $IS_{i,l}$ is a set of interactive states defined as, $IS_{i,l} = S \times M_{j,l-1}$, where $M_{j,l-1} = \{\Theta_{j,l-1} \cup SM_j\}$, for $l \geq 1$, and $IS_{i,0} = S$, where S is the set of states of the physical environment. $\Theta_{j,l-1}$ is the set of computable *intentional models* of agent j. The remaining set of models, SM_j, is the set of *subintentional models* of j;
- $A = A_i \times A_j$, is the set of joint actions of all agents in the environment;
- T_i is a transition function, $T_i : S \times A \times S \rightarrow [0, 1]$. It reflects the possibly uncertain effects of the joint actions on the physical states of the environment;
- Ω_i is the set of observations of agent i;
- O_i is an observation function, $O_i : S \times A \times \Omega_i \rightarrow [0, 1]$. It describes how likely it is for agent i to receive the observations given the physical state and joint actions;
- R_i is a reward function, $R_i : IS_i \times A \rightarrow \mathbb{R}$. It describes agent i's preferences over its interactive states and joint actions, though usually only the physical states and actions matter.

Intentional models ascribe to the other agent beliefs, preferences and rationality in action selection [7] and are analogous to *types* as used in game theory [17]. Each intentional model, $\theta_{j,l-1} = \langle b_{j,l-1}, \hat{\theta}_j \rangle$, where $b_{j,l-1}$ is agent j's belief at level $l-1$, and the *frame*, $\hat{\theta}_j = \langle A, T_j, \Omega_j, O_j, R_j, OC_j \rangle$. Here, j is assumed Bayes rational and OC_j is j's optimality criterion.

A subintentional model is a triple, $sm_j = \langle h_j, O_j, f_j \rangle$, where $f_j : H_j \rightarrow \Delta(A_j)$ is agent j's function, assumed computable, which maps possible histories of j's observations to distributions over its actions. h_j is an element of H_j and O_j gives the probability with which j receives its input. Simple examples of subintentional models include a no-information model [15] and the fictitious play model [11], which is history dependent. Such models would be extended at each time step to incorporate the revised history. Another example of a subintentional model is a finite state automaton.

Notice that because the intentional models include the beliefs as well, the state space is naturally nested. We give a recursive bottom-up construction of the interactive state space below.

$$
\begin{aligned}
&IS_{i,0} = S, && \Theta_{j,0} = \{\langle b_{j,0}, \hat{\theta}_j \rangle \mid b_{j,0} \in \Delta(IS_{j,0})\} \\
&IS_{i,1} = S \times \{\Theta_{j,0} \cup SM_j\}, && \Theta_{j,1} = \{\langle b_{j,1}, \hat{\theta}_j \rangle \mid b_{j,1} \in \Delta(IS_{j,1})\} \\
&\vdots && \vdots \\
&IS_{i,l} = S \times \{\Theta_{j,l-1} \cup SM_j\}, && \Theta_{j,l} = \{\langle b_{j,l}, \hat{\theta}_j \rangle \mid b_{j,l} \in \Delta(IS_{j,l})\}
\end{aligned}
$$

Here, $\Theta_{j,0}$ is the set of POMDPs,[1] and the associated $\hat{\theta}_j$ represents the parameters of the POMDP. Similar formulations of nested state spaces have appeared in the game-theoretic literature (see, for example, [1, 2, 23]).

Solution to a finitely nested I-POMDP (hereafter, referred to as I-POMDP for simplicity) is the agent i's policy which is a mapping of its beliefs on the interactive states to a distribution over its actions, $\Delta(IS_i) \rightarrow \Delta(A_i)$. Analogous to POMDPs, the two steps, namely *belief update* and *policy computation*, are used to solve an I-POMDP.

3.1.1 I-POMDP belief update

Analogous to POMDPs, an agent within the I-POMDP framework updates its belief as it acts and observes. However, there are two differences that complicate the belief update in

[1] Other agent's actions are folded in as noise into the T, O and R functions.

multiagent settings when compared to single agent ones. First, since the state of the physical environment depends on the joint actions of both agents, i's prediction of how the physical state changes has to be made based on its prediction of j's actions obtained from the models. Second, changes in j's models have to be included in i's belief update. Specifically, if j is intentional then an update of j's beliefs due to its action and observation has to be included. In other words, i has to update its belief based on its prediction of what j would observe and how j would update its belief. If j's model is subintentional, then j's probable observations are appended to the observation history contained in the model. Formally, we have:

$$\begin{aligned} Pr(is^{t+1}|a_i^t, b_{i,l}^t) = \beta \sum\nolimits_{IS^t:\widehat{m}_j^t=\widehat{\theta}_j^{t+1}} b_{i,l}^t\left(is^t\right) \sum\nolimits_{a_j^t} Pr\left(a_j^t|\theta_{j,l-1}^t\right) O_i\left(s^{t+1}, a_i^t, a_j^t, o_i^{t+1}\right) \\ \times T_i\left(s^t, a_i^t, a_j^t, s^{t+1}\right) \sum\nolimits_{o_j^{t+1}} O_j\left(s^{t+1}, a_i^t, a_j^t, o_j^{t+1}\right) \\ \times \delta_K\left(SE_{\widehat{\theta}_j^{t+1}}(b_{j,l-1}^t, a_j^t, o_j^{t+1}) - b_{j,l-1}^{t+1}\right) \end{aligned} \quad (1)$$

where β is the normalizing constant, δ_K is the Kronecker delta and is 1 if its argument is 0 otherwise it is 0, $Pr(a_j^t|\theta_{j,l-1}^t)$ is the uniform distribution over actions that are Bayes rational for the agent described by the model, $\theta_{j,l-1}^t$, and $SE(\cdot)$ is an abbreviation for the belief update. If j's models are level 0 POMDPs, then $SE(\cdot)$ represents the standard POMDP belief update, otherwise it represents the update described above. The belief update equation for the case where j's models are subintentional is given in [14].

As we mentioned before, the belief update as formalized by Eq. 1 updates not only agent i's belief over the physical states but also its belief on j's models. Agent i's updated distribution on the physical states is given by the probability of transitioning to the new state, $T_i(s^t, a_i^t, a_j^t, s^{t+1})$, and it is corrected using the likelihood of the observation from the state, $O_i(s^{t+1}, a_i^t, a_j^t, o_i^{t+1})$. However, because the transition and observation depends on the actions of the other agent, the probability of its actions must be predicted. The distribution over j's actions is given by the term, $Pr(a_j^t|\theta_{j,l-1}^t)$. As the other agent acts and observes as well, it's belief must be updated, which is represented by $SE(\cdot)$. Agent i's belief over j's updated belief depends on the probability with which j acts and makes its observations given by the factor, $O_j(s^{t+1}, a_i^t, a_j^t, o_j^{t+1})$.

If agent j is also modeled as an I-POMDP, then i's belief update invokes j's belief update (via the term $SE_{\widehat{\theta}_j^{t+1}}(b_{j,l-1}^t, a_j^t, o_j^{t+1})$), which in turn could invoke i's belief update and so on. This recursion in belief nesting bottoms out at the 0th level. At this level, the belief update of the agent reduces to a POMDP belief update.

3.1.2 Policy computation

Each belief state of agent i in an I-POMDP has an associated value reflecting the maximum payoff the agent can expect in this belief state for the case of a finite horizon, n:

$$\begin{aligned} U^n(\langle b_{i,l}, \widehat{\theta}_i\rangle) = \max_{a_i \in A_i} \Bigg\{ \sum_{is \in IS_{i,l}} ER_i(is, a_i) b_{i,l}(is) + \gamma \sum_{o_i \in \Omega_i} Pr(o_i|a_i, b_{i,l}) \\ \times U^{n-1}(\langle SE_{\widehat{\theta}_i}(b_{i,l}, a_i, o_i), \widehat{\theta}_i\rangle) \Bigg\} \end{aligned} \quad (2)$$

where, $ER_i(is, a_i) = \sum_{a_j} R_i(is, a_i, a_j) Pr(a_j|m_{j,l-1})$ (since $is = (s, m_{j,l-1})$). Eq. 2 is a basis for value iteration in I-POMDPs.

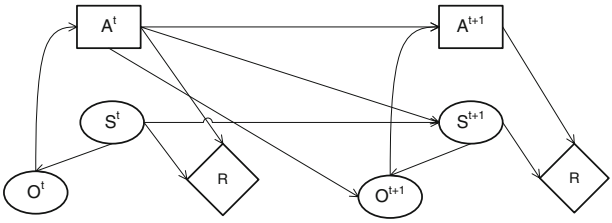

Fig. 2 A two time-slice dynamic ID representing the decision-making problem of an agent. The oval nodes representing the state (S) and the observation (Ω) reflected in the observation function, O, are the chance nodes. The rectangle is the decision node (A) and the diamond is the reward function (R). Influences (links) connect nodes within the same time slice as well as nodes across time slices

For the case of a finite horizon with discount factor γ, agent i's optimal action, a_i^*, is an element of the set of optimal actions for the belief state, $OPT(\theta_i)$, defined in Eq. 3. Thus, the finite horizon policy is a mapping from the agent's belief state to the set of optimal actions, indexed by the horizon.

$$OPT(\langle b_{i,1}, \widehat{\theta_i}\rangle) = \underset{a_i \in A_i}{argmax} \Bigg\{ \sum_{is \in IS_{i,l}} ER_i(is, a_i) b_{i,l}(is) + \gamma \sum_{o_i \in \Omega_i} Pr(o_i | a_i, b_{i,l}) \times U^n(\langle SE_{\widehat{\theta_i}}(b_{i,1}, a_i, o_i), \widehat{\theta_i}\rangle) \Bigg\} \tag{3}$$

3.2 Influence diagrams

A well-known graphical formalism for describing and solving decision-making situations is the *influence diagram* (ID) [18, 33, 36]. Graphical models, such as IDs, offer a formalism that decomposes the state into *chance* (random) variables and dependencies between the variables, *decision* nodes for modeling the action choices, and *utility* nodes for representing the agent's preferences. As we mentioned previously, graphical models are an explicit qualitative description of the decision-making situation. We observe that an ID augments a Bayesian network [26] with *decision* and *utility* nodes.

In an ID, the traditional $|S|^2$-size transition matrices are decomposed into tables of smaller sizes, each of which models the local effect of an action on some variables. IDs also find another purpose: they may be used to deliberate the optimal action of an agent *online* given its initial belief as it acts and observes. On solving an ID unrolled over as many time slices as the horizon, called a *dynamic* ID and shown in Fig. 2, we obtain the value of performing each action in the decision node, with the best action being the one with the largest value.

Dynamic IDs are structured representations of POMDPs [31]. The values of the decision node, A^t, constitute the set of actions, A, in a POMDP. The values of the chance node, S^t,[2] and the observation node, O^t, are the sets of states and observations, respectively, in a POMDP. The conditional probability distribution (CPD), $Pr(S^{t+1}|S^t, A^t)$, of the chance node, S^{t+1}, is the transition function, T in a POMDP. The CPD, $Pr(O^{t+1}|S^{t+1}, A^t)$, of the chance node, O^{t+1}, is the observation function, O, and the utility table of the value node, R, is the reward function, R, in a POMDP.

Dynamic IDs are suitable for describing single agent decision-making situations or multiagent problems where the other agents are modeled as automatons whose actions are guided by a fixed and known probability distribution.

[2] Note that S could be factored into chance nodes and dependency links between them.

4 Graphical models for I-POMDPs

As we mentioned previously, naive extensions of IDs to settings populated by multiple agents are possible by treating other agents as automatons, represented using chance nodes. However, this approach assumes that the agents' actions are controlled using a probability distribution that does not change over time. We introduce graphical formalisms that adopt a more sophisticated approach by generalizing IDs to make them applicable to settings shared with other agents who may act and observe, and update their beliefs.

4.1 Interactive influence diagrams (I-IDs)

We introduce *interactive influence diagrams* (I-ID) that generalize IDs to multiagent settings in this section. In addition to the usual chance, decision, and utility nodes, I-IDs include a new type of node called the *model* node. We show a general level l I-ID in Fig. 3, where the model node, $M_{j,l-1}$, is denoted using a hexagon. In addition to the model node, I-IDs differ from IDs by having a dashed link (called the "policy link" in prior work [28]) between the model node and a chance node, A_j, that represents the distribution over the other agent's actions given its model. In the absence of other agents, the model node and the chance node, A_j, vanish and I-IDs collapse into traditional IDs. 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. The new model nodes are conditioned on the physical state and possibly model nodes of other agents' while the chance nodes are linked to the utility node.

The model node contains as its values the alternative computational models ascribed by i to the other agent from the set, $\Theta_{j,l-1} \cup SM_j$, where $\Theta_{j,l-1}$ and SM_j were defined previously in Sect. 3.1. A model in the model node may itself be an I-ID, ID or a probability distribution over actions, and the recursion terminates when a model is an ID or subintentional. Because the model node contains the alternative models of the other agent as its values, its representation is not trivial. In particular, some of the models within the node are I-IDs that when solved generate the agent's optimal action(s) in their decision nodes. Each decision node is mapped to a corresponding chance node, say A_j^1, in the following way: if OPT is the set of optimal actions obtained by solving the I-ID (or ID), then $Pr(a_j \in A_j^1) = \frac{1}{|OPT|}$ if $a_j \in OPT$, 0 otherwise.

Borrowing insights from previous work [13], 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. 4a. The decision node of each level $l-1$ I-ID is mapped to a chance node, as we mentioned previously, so that the actions with the largest value in the decision node are assigned uniform probabilities in the chance node while the rest are assigned zero probability.

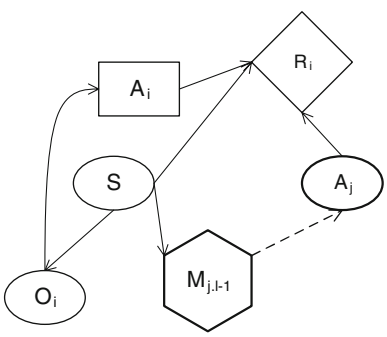

Fig. 3 A generic level l I-ID for agent i situated with one other agent j. The hexagon is the model node ($M_{j,l-1}$) and the dashed arrow is the policy link. Members of the model node could be I-IDs themselves or IDs ($m^1_{j,l-1}, m^2_{j,l-1}$; diagrams not shown here for simplicity) representing intentional models

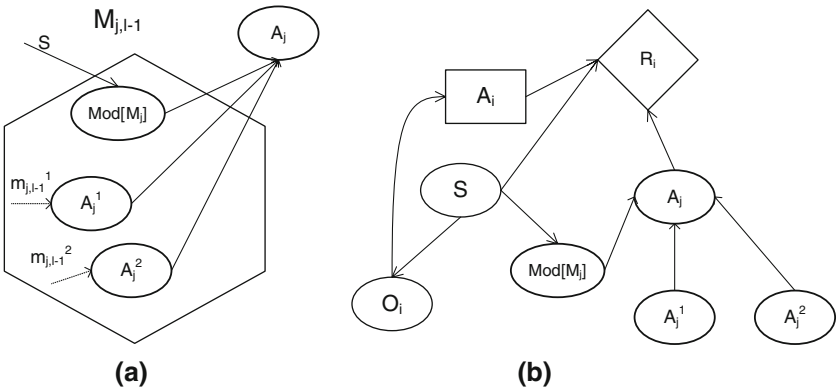

Fig. 4 (**a**) Representing the model node and policy link using chance nodes and dependencies between them. The decision nodes of the lower-level I-IDs or IDs ($m^1_{j,l-1}, m^2_{j,l-1}$) are mapped to the corresponding chance nodes (A^1_j, A^2_j), which is indicated by the dotted arrows. Depending on the value of the node, Mod[M_j], the distribution of each of the chance nodes is assigned to the node A_j in its CPD. (**b**) In order to solve the I-ID, we obtain a flat ID by replacing the model node and the policy link in the I-ID of Fig. 3 with the chance nodes and the relationships between them as shown in (**a**). Distributions for the chance nodes, A^1_j and A^2_j, are obtained by solving the models, $m^1_{j,l-1}$ and $m^2_{j,l-1}$, respectively

The different chance nodes (A^1_j, A^2_j), one for each model, and additionally, the chance node labeled $Mod[M_j]$ form the parents of the chance node, A_j. As each action node is associated with a model, there are as many action nodes in $M_{j,l-1}$ as the number of models in the model node. The CPD of the chance node, A_j, is a *multiplexer* that assumes the distribution of each of the action nodes (A^1_j, A^2_j) depending on the value of $Mod[M_j]$. The values of $Mod[M_j]$ denote the different models of j. In other words, when $Mod[M_j]$ has the value $m^1_{j,l-1}$, the chance node A_j has the distribution over its values that the node A^1_j has, and A_j assumes the distribution of A^2_j when $Mod[M_j]$ has the value $m^2_{j,l-1}$. The distribution over the node, $Mod[M_j]$, is the agent i's top-level belief over the level $l-1$ models of j given a physical state. Notice that Fig. 4a also clarifies the semantics of the policy link, and shows how it can be represented using the traditional dependency links.

In Fig. 4b, we show the flat ID when the model node in Fig. 3 is replaced by the chance nodes and the relationships between them. Distributions for the chance action nodes are obtained by solving the lower level models. There are no special-purpose policy links, rather it is composed of only those types of nodes and dependency relationships between the nodes that are found in traditional IDs. This allows I-IDs to be implemented and solved using conventional application tools that target IDs.

Note that we may view the level l I-ID as a NID [13]. Specifically, each of the level $l-1$ models within the model node are blocks in the NID (see Fig. 5). If the level $l = 1$, each block is a traditional ID, otherwise if $l > 1$, each block within the NID may itself be a NID. Note that within the I-IDs (or IDs) at each level, there is only a single decision node. Thus, our NID does not contain any MAIDs.

4.2 Interactive dynamic influence diagrams (I-DIDs)

Interactive dynamic influence diagrams (I-DIDs) extend the formalism of interactive influence diagrams (I-IDs) to solve dynamic decision problems, just as DIDs extend IDs. We show a general level l I-DID for two time slices in Fig. 6.

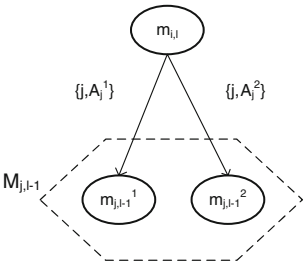

Fig. 5 A level l I-ID represented as a NID. The probabilities assigned to the blocks of the NID are i's beliefs over j's models conditioned on a physical state

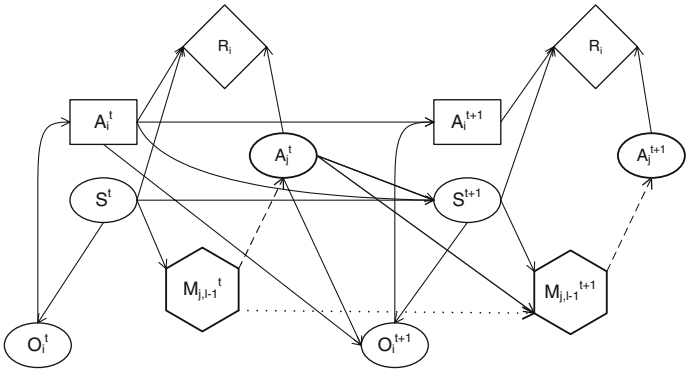

Fig. 6 An I-DID unrolled over two time horizons. The dotted arrow between the model nodes is called the model update link

Note that the CPD, $Pr(S^{t+1}|S^t, A_i^t, A_j^t)$, of the chance node, S^{t+1}, is the transition function, T_i in the I-POMDP$_{i,l}$, the CPD, $Pr(O_i^{t+1}|S^{t+1}, A_i^t, A_j^t)$, of the chance node, O_i^{t+1}, is the observation function, O_i. In addition to the model nodes and the dashed policy link, what differentiates an I-DID from a DID is the *model update link* shown as a dotted arrow in Fig. 6. We explained the semantics of the model node and the policy link in the previous section; we describe the model update next.

The update of the model node over time involves two steps: First, given the candidate models at time t, we identify the updated set of models that reside in the model node at time $t+1$. Recall from Sect. 3.1 that an agent's intentional model includes its belief. Because the agents act and receive observations, their models are updated to reflect their changed beliefs. In some cases, the update may result in a model whose structure may be different from that previously. Since the set of optimal actions for a model could include all the actions, and the agent may receive any one of $|\Omega_j|$ possible observations, the updated set at time step $t+1$ will have at most $|M_{j,l-1}^t||A_j||\Omega_j|$ models. Here, $|M_{j,l-1}^t|$ is the number of models at time step t, $|A_j|$ and $|\Omega_j|$ are the largest spaces of actions and observations respectively, among all the models. 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.

In Fig. 7, we show how the dotted model update link in the I-DID could be implemented. If each of the two level $l-1$ models ascribed to j at time step t results in one action, and j could make one of two possible observations, then the model node at time step $t+1$ contains four updated models ($m_{j,l-1}^{t+1,1}$, $m_{j,l-1}^{t+1,2}$, $m_{j,l-1}^{t+1,3}$, and $m_{j,l-1}^{t+1,4}$). These models differ in their initial beliefs, each of which is the result of j updating its beliefs due to its action and

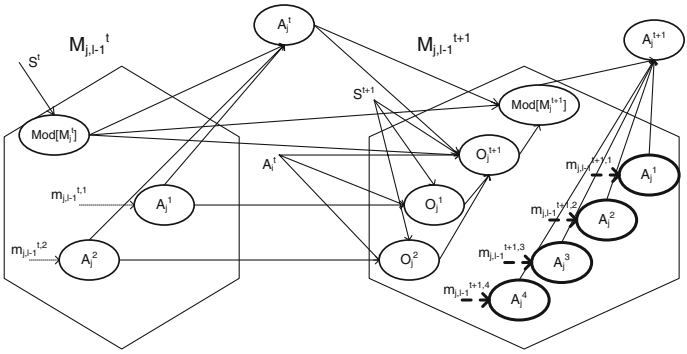

Fig. 7 Representing the model update link between model nodes using chance nodes and dependency links between them. Notice the growth in the number of models in the model node at $t + 1$ (highlighted in bold)

a possible observation. The decision nodes in each of the I-DIDs or DIDs that represent the lower level models are mapped to the corresponding chance nodes, as mentioned previously.

Next, we describe how the distribution over the updated set of models (the distribution over the chance node $Mod[M_j^{t+1}]$ in $M_{j,l-1}^{t+1}$) is computed. The probability that j's updated model is, say $m_{j,l-1}^{t+1,1}$, depends on the probability of j performing the action and receiving the observation that led to this model, and the prior distribution over the models at time step t. Because the chance node A_j^t assumes the distribution of each of the action nodes based on the value of $Mod[M_j^t]$, the probability of the action is given by this chance node. In order to obtain the probability of j's possible observation, we introduce the chance node O_j^{t+1}, which depending on the value of $Mod[M_j^t]$ assumes the distribution of the observation node in the lower level model denoted by $Mod[M_j^t]$. Analogous to A_j^t, the conditional probability table of O_j^{t+1} is also a multiplexer modulated by $Mod[M_j^t]$. Because the probability of j's observations depends on the physical state and the joint actions of both agents, the chance nodes, O_j^1 and O_j^2, are linked with S^{t+1}, A_j^1, and A_j^2 respectively.[3] Finally, the distribution over the prior models at time t is obtained from the chance node, $Mod[M_j^t]$ in $M_{j,l-1}^t$. Consequently, the chance nodes, $Mod[M_j^t]$, A_j^t, and O_j^{t+1}, form the parents of $Mod[M_j^{t+1}]$ in $M_{j,l-1}^{t+1}$. Notice that the model update link may be replaced by the dependency links between the chance nodes that constitute the model nodes in the two time slices.

Expansion of the I-DID over more time steps translates into repeating the two steps of updating the set of models that form the values of the model node and adding the relationships between the chance nodes, as many times as there are model update links. We note that the possible set of models of the other agent j grows exponentially with the number of time steps. For example, after T steps, there may be at most $|M_{j,l-1}^{t=1}|(|A_j||\Omega_j|)^{T-1}$ candidate models residing in the model node.

In Fig. 8 we show 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. Chance nodes and dependency links not in bold are standard, usually found in single agent DIDs.

[3] Note that O_j^1 and O_j^2 represent j's observations at time $t + 1$, and arise from different j's models.

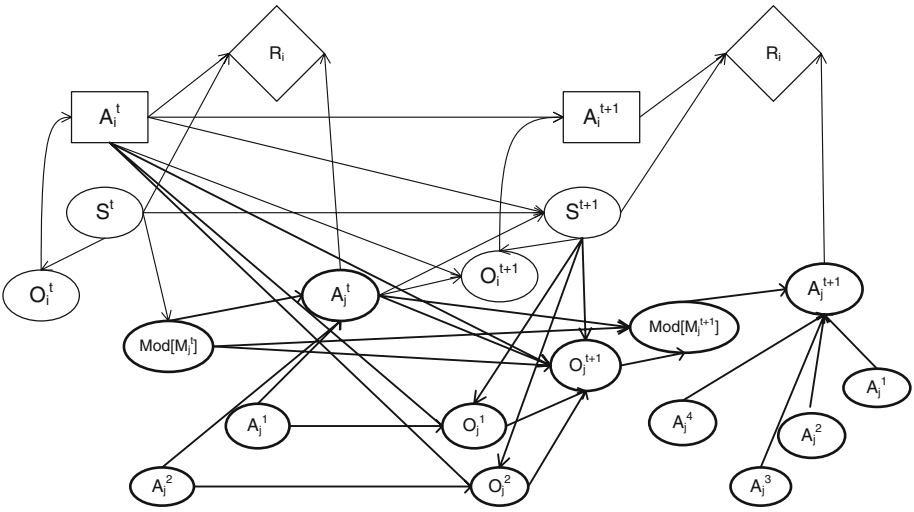

Fig. 8 A flat DID obtained by replacing the model nodes and model update link in the I-DID of Fig. 6 with the chance nodes and the relationships (in bold) as shown in Fig. 7. The lower level models are solved to obtain the distributions for the chance action nodes

4.3 Mapping I-DIDs to I-POMDPs

Analogously to the relation between DIDs and POMDPs, elements of I-DIDs could be mapped to those of I-POMDPs as defined in Sect. 3.1. The values of the decision node, A_i in Fig. 6, is the set of actions of agent i, and similarly for the chance node A_j. Their joint is the set of joint actions of both agents, A, in the definition of I-POMDP$_{i,l}$. The values of the chance node, S, and the observation node, O_i, are the sets of physical states and observations of i, respectively, in the I-POMDP. The CPDs of the chance nodes, S^{t+1} and O_i^{t+1}, are the transition and observation functions, T_i and O_i of agent i in the I-POMDP. The utility table of the value node, R_i, is the reward function, R_i of agent i in the I-POMDP.

The chance, decision, utility nodes and the associated edges in an I-DID constitute the frame of an intentional model as defined in Sect. 3.1. As we mentioned previously, the model node contains as its values the alternative computational models ascribed by i to the other agent from the set, $\Theta_{j,l-1} \cup SM_j$, where $\Theta_{j,l-1}$ and SM_j were defined previously (Sect. 3.1). Thus, the set of pairs, each consisting of a value of node S and a model in node, $M_{j,l-1}$, constitutes the interactive state space, $IS_{i,l}$, in I-POMDP. The joint probability distribution over the chance node, S, and the node $Mod[M_j]$ in the model node represents the *top-level probability distribution* that agent i has over its interactive states, $IS_{i,l}$.

As we may expect, the update of the model node over time closely relates to the belief update process in Eq. 1. The update of agent j's belief given its action and observation (the term, $SE_{\widehat{\theta}_j}(b^t_{j,l-1}, a^t_j, o^{t+1}_j)$ in Eq. 1) results in new models with updated beliefs at time $t+1$ in Fig. 8; one model for each combination of an optimal action and observation of j that results in a unique belief. The distribution over the chance node, A^t_j, conditioned on $Mod[M^t_j]$ is the distribution, $Pr(a^t_j|\theta^t_{j,l-1})$ appearing in Eq. 1, where $\theta^t_{j,l-1}$ is an I-DID or DID in the model node. Finally, the updated distribution over the physical states and models of j is the distribution over S^{t+1} and $Mod[M^{t+1}_j]$ as obtained using the standard inference. The inference propagates through parents of the $Mod[M^{t+1}_j]$ node, which is equivalent to

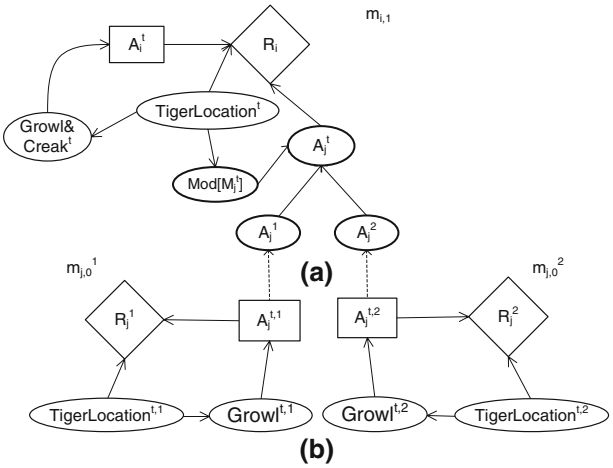

Fig. 9 (**a**) Level 1 I-ID of agent i, (**b**) two level 0 IDs of agent j whose decision nodes are mapped to the chance nodes, A_j^1, A_j^2, in (**a**), indicated by the dotted arrows. The two IDs differ in the distribution over the chance node, *TigerLocation*

summing over a_j^{t-1} and o_j^t in Eq. 1. Note that $Mod[M_j^{t+1}]$ is conditioned on the chance node O_j^{t+1} thereby accounting for j's observation function that appears in Eq. 1.

4.4 Example representations

In order to illustrate the usefulness of I-DIDs, we apply them to three illustrative problems. We describe, in particular, the formulation of the I-DIDs for these examples.

4.4.1 Multiagent tiger problem

We begin our illustrations of using I-IDs and I-DIDs with a slightly modified version of the multiagent tiger problem [14]. The problem has two agents, each of which can open the right door (OR), the left door (OL) or listen (L). In addition to hearing growls (from the left (GL) or from the right (GR)) when they listen, the agents also hear creaks (from the left (CL), from the right (CR), or no creaks (S)), which noisily indicate the other agent's opening one of the doors or listening. When any door is opened, the tiger *persists* in its original location with a probability of 95%. Agent i hears growls with a reliability of 65% and creaks with a reliability of 95%. Agent j, on the other hand, hears growls with a reliability of 95%. Thus, the setting is such that agent i hears agent j opening doors more reliably than the tiger's growls. This suggests that i could use j's actions as an indication of the location of the tiger, as we discuss later. Each agent's preferences are as in the single agent game discussed in the original version [19]. The transition, observation, and reward functions are as shown in Appendix A.

Let us consider a particular setting of the tiger problem in which agent i considers two distinct level 0 models of j. This is represented in the level 1 I-ID shown in Fig. 9. The two IDs could differ, for example, in the probability that j assigns to the tiger being behind the left door as modeled by the node *TigerLocation*.

Given the level 1 I-ID, we may expand it into the I-DID shown in Fig. 10. The model node, $M_{j,0}^t$, contains the different DIDs that are expanded from the level 0 IDs in Fig. 9b.

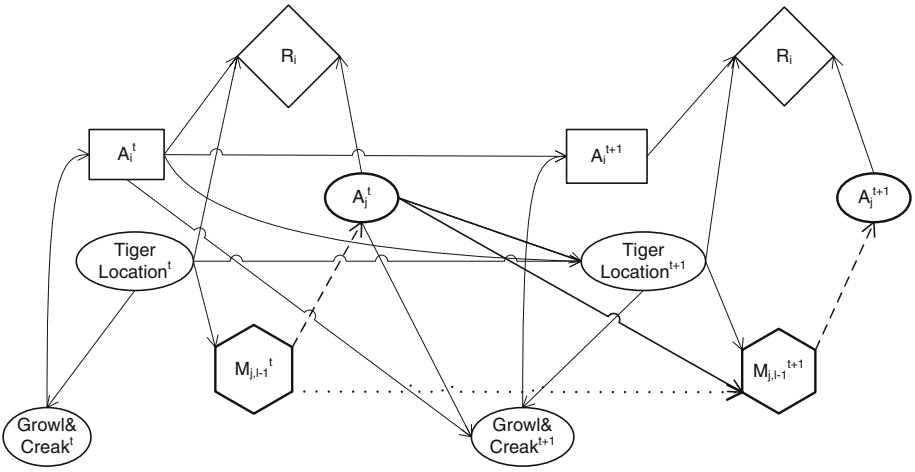

Fig. 10 Level 1 I-DID of agent i for the multiagent tiger problem. The model node contains level 0 DIDs of agent j. At horizon 1, the models of j are IDs

The DIDs may have different probabilities about the tiger location at time step t. We get the probability distribution of j's actions in chance node A_j^t by solving the level 0 DIDs of j. On performing the optimal action(s) at time step t, j may receive observations of the tiger's growls. This is reflected in new beliefs on the tiger's position within j's DIDs at time step $t+1$. Consequently, the model node, $M_{j,0}^{t+1}$, contains more models of j and i's updated belief on j's possible DIDs.

4.4.2 Public good problem

The public good (PG) problem [12], consists of a group of M agents, each of whom must either contribute some resource to a public pot or keep it for themselves. Since resources contributed to the public pot are shared among all the agents, they are less valuable to the agent when in the public pot. However, if all agents choose to contribute their resources, then the payoff to each agent is more than if no one contributes. Since an agent gets its share of the public pot irrespective of whether it has contributed or not, the dominating action is for each agent to not contribute, and instead "free ride" on others' contributions.

For simplicity, we assume that the game is played between $N = 2$ agents, i and j. Let each agent be initially endowed with X_T amount of resources. While the classical PG game formulation permits each agent to contribute any quantity of resources ($\leq X_T$) to the public pot, we simplify the action space by allowing two possible actions. Each agent may choose to either *contribute* (C) a *fixed* amount of the resources, or not contribute. The latter action is denoted as *defect* (D). We assume that the actions are not observable to others. The value of resources in the public pot is discounted by c_i for each agent i, where c_i is the marginal private return. We assume that $c_i < 1$ so that the agent does not benefit enough that it contributes to the public pot for private gain. Simultaneously, $c_i N > 1$, making collective contribution Pareto optimal.

In order to encourage contributions, the contributing agents punish free riders but incur a small cost for administering the punishment. Let P be the punishment meted out to the defecting agent and c_p the non-zero cost of punishing for the contributing agent. For simplicity, we assume that the cost of punishing is same for both the agents. The one-shot PG

Table 1 The one-shot PG game with punishment

i/j	C	D
C	$2c_iX_T, 2c_jX_T$	$c_iX_T - c_p, X_T + c_jX_T - P$
D	$X_T + c_iX_T - P, c_jX_T - c_p$	X_T, X_T

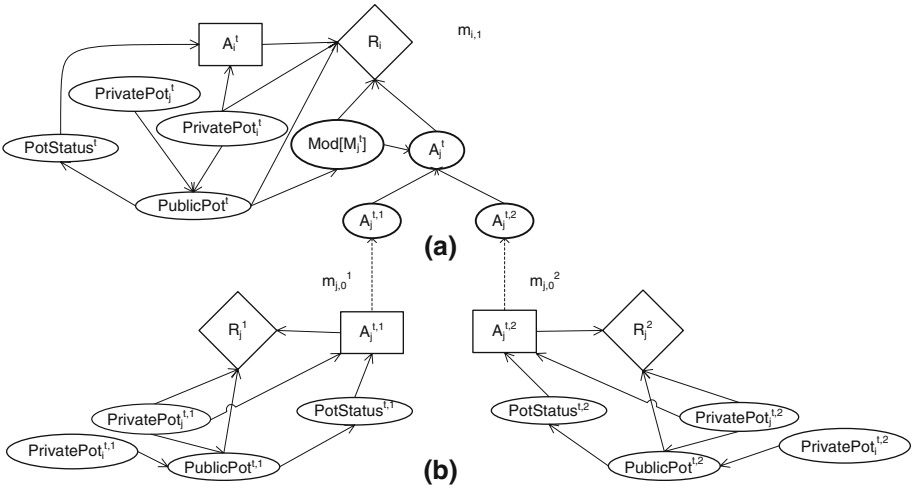

Fig. 11 (**a**) Level 1 I-ID of agent i for the PG problem, (**b**) level 0 IDs of agent j with decision nodes mapped to the chance nodes, A_j^1 and A_j^2, in (**a**)

game with punishment is shown in Table 1. Let $c_i = c_j, c_p > 0$, and if $P > X_T - c_iX_T$, then defection is no longer a dominating action. If $P < X_T - c_iX_T$, then defection is the dominating action for both. If $P = X_T - c_iX_T$, then the game is not dominance-solvable.

Though in the standard repeated PG game, the quantity in the public pot is revealed to all the agents after each round of actions, we assume in our formulation that it is hidden from the agents. Each agent may contribute a fixed amount, x_c, or defect. An agent on performing an action receives an observation of *plenty* (PY) or *meager* (MR) symbolizing the state of the public pot. Notice that the observations are also indirectly indicative of agent j's actions because the state of the public pot is influenced by them. The amount of resources in agent i's private pot, is perfectly observable to i. The payoffs are analogous to Table 1.

We construct level 0 IDs for j that model two distinct types, one whose marginal private return, c_j, is high and does not punish free riders (encoded in the reward function), and the other whose c_j is low. While the former type always contributes, the latter chooses to predominantly defect. We show the level 1 I-ID that represents this problem in Fig. 11. The two level 0 IDs have different reward functions in the utility nodes R_j^1 and R_j^2 respectively.

Expanding the level 1 I-ID of agent i, we show the I-DID in Fig. 12. The two level 0 IDs in Fig. 11b are unrolled into DIDs that are contained in the model node $M_{j,0}^t$. Since level 0 DIDs have different rewards in the utility nodes we get different probability distributions of j's actions in chance node A_j^t. At time step t, j may observe the status of the public pot as indicated in chance node $PotStatus^t$. This results in several more level 0 DIDs at time step $t+1$. Hence the model node, $M_{j,0}^{t+1}$, contains j's DIDs in which j has different beliefs on the status of the public pot depending on its previous observations.

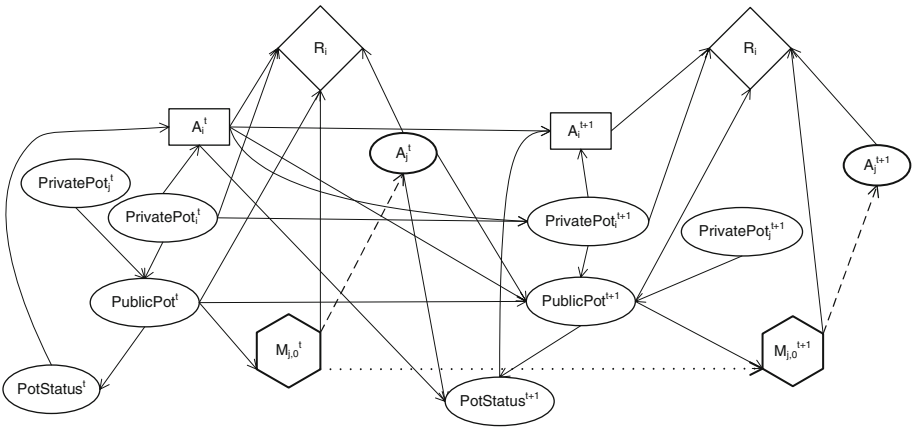

Fig. 12 Level 1 I-DID of agent i. The model node contains level 0 DIDs of agent j, which reduce to IDs at horizon 1

4.4.3 Online shopper's dilemma

Our third example application is in the area of e-commerce and inspired by the behavior of real-world users on online auction portals such as eBay. We consider the scenario where the seller and the buyer will finalize their transaction, after an agreement on the price of some merchandise. The seller will deliver to the buyer the agreed upon item and the buyer will transfer to the seller an amount of money, simultaneously. We consider multiple such transactions occurring sequentially between the buyer and the seller.

As is sometimes the case, the seller may choose to deliver a substandard item to the buyer, while the buyer may elect to transfer a partial amount of the money. Such actions are dependent, in part, on the reputation of the online portal—portals with strict policies against fraud experience less fraudulent behavior—and on the trustworthiness of the seller. The reputations of the portals are often inferred from online reviews which may be good (G) or bad (B).

We model the decision situations of the buyer and the seller using I-DIDs. We suppose that both the buyer, i, and the seller, j, have a valuation for the item. The valuations are denoted by v_i and v_j, respectively, and they represent how much the item is worth to the participants. We assume that the participants have already agreed on the price, c_{ij}, for the item, but that the money has not been transferred. Agent i may transfer the full money, c_{ij}, or a partial amount, γc_{ij} (discount factor: $\gamma \in (0, 1]$), to the seller j. Simultaneously, agent j may deliver items of differing quality levels. For the sake of simplicity, we assume that the delivered item may be of a high or a low quality.

If the delivered item is of low quality, it will be worth αv_i to the buyer, while the item will be worth βv_j to the seller, where $0 < \alpha, \beta \leq 1$. Thus the buyer will stand to make $\alpha v_i - \gamma c_{ij}$, while the seller will gain $\gamma c_{ij} - \beta v_j$. We observe that this game has a pair of dominating strategies, which prescribes the seller to deliver a low quality item and the buyer to transfer a partial amount of the money, given the conditions on the parameters.

Often, online portals implement ways to punish fraudulent users. For example, eBay immediately suspends sellers against whom a large number of complaints have been received. We implement a simple punishment mechanism whereby the gain from a transaction is reduced by p_i if only the buyer cheats by transferring a partial amount of money, p_j if only the seller commits fraud by delivering a low quality item, or a common amount of p_{ij}, if both

Table 2 The one-shot online shopping transaction with punishment

i/j	HighQuality (HQ)	LowQuality (LQ)
FullMoney (FM)	$v_i - c_{ij}, c_{ij} - v_j$	$\alpha v_i - c_{ij}, c_{ij} - \beta v_j - p_j$
PartialMoney (PM)	$v_i - \gamma c_{ij} - p_i, \gamma c_{ij} - v_j$	$\alpha v_i - \gamma c_{ij} - p_{ij}, \gamma c_{ij} - \beta v_j - p_{ij}$

The buyer may choose to transfer the full amount or a partial amount of the money and the seller may elect to deliver the item of high or low quality (possibly defective)

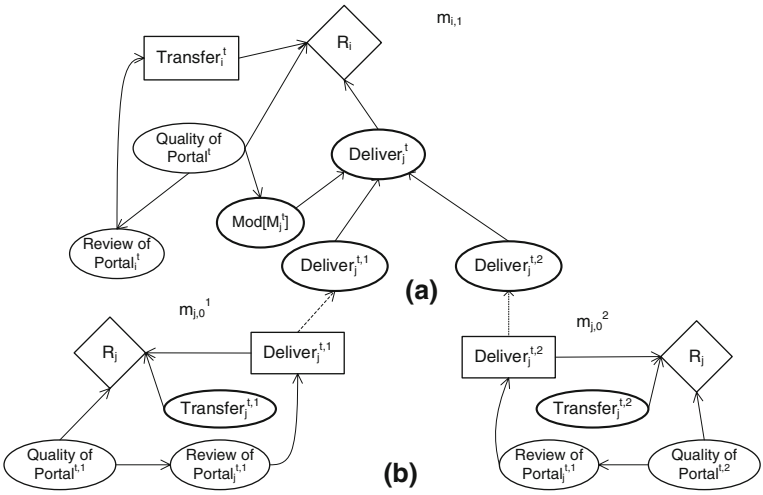

Fig. 13 (**a**) Level 1 I-ID of the buyer, i, (**b**) level 0 IDs of the seller, j, with decision nodes mapped to the chance nodes, $Deliver_j^{t,1}$ and $Deliver_j^{t,2}$, in (**a**)

cheat. The punishments (p_i, p_j, and p_{ij}) depend on the quality of the portal. The one-shot transaction with punishment is shown in Table 2.

Depending on p_{ij}, notice that the buyer's transfer of a partial amount and the seller's delivery of a low quality item are no longer a dominating strategy pair. We show the level 1 I-ID representing the buyer's decision problem in Fig. 13. We show two level 0 models of the seller represented using IDs. The models may represent sellers with different beliefs on the quality of the portal and the buyer's actions. For example, one model could be of a mistrusting seller that initially believes that the portal does not strictly enforce anti-fraud policies and that the buyer is likely to transfer a partial amount of the money. The other model could be of a trusting seller.

We expand the level 1 I-ID into an I-DID and show the I-DID in Fig. 14. The model node, $M_{j,0}^t$, contains the level 0 DIDs, which may have different beliefs on the portal quality or the buyer's behavior. Chance node $Deliver_j^t$ captures the probability of the seller's actions when the level 0 DIDs are solved in the model node.

5 Exact solutions of I-DIDs

The solution to a level l I-DID for agent i expanded over T time steps proceeds in a bottom-up manner and may be carried out recursively. For the purpose of illustration, let $l = 1$ and $T = 2$. The solution method uses the standard look-ahead technique, projecting the agent's action

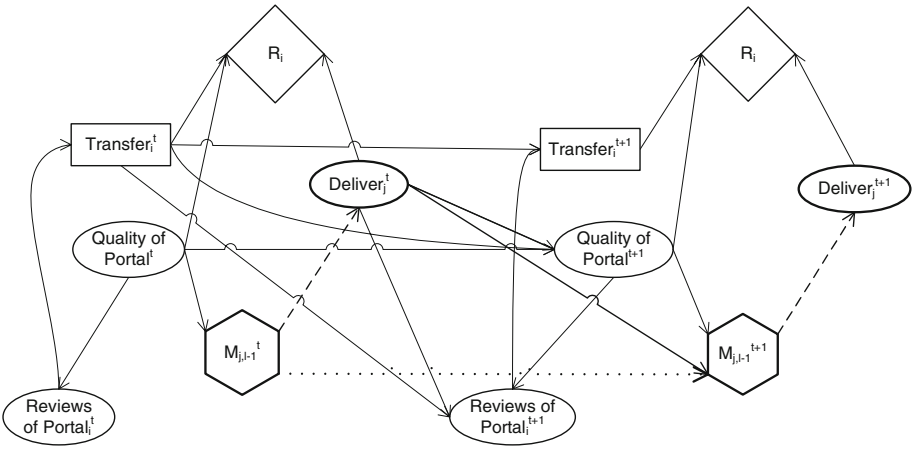

Fig. 14 Level 1 I-DID of the buyer. The model node contains level 0 DIDs of the seller (IDs at horizon 1)

and observation sequences forward from the current belief state [31], and finding the possible beliefs that i could have in the next time step. Because agent i has a belief over j's models as well, the look-ahead includes finding out the possible models that j could have in the future. Consequently, each of j's subintentional or level 0 models (represented using a standard DID) in the first time step must be solved to obtain its optimal set of actions. These actions are combined with the set of possible observations that j could make in that model, resulting in an updated set of candidate models (that include the updated beliefs) that could describe the behavior of j in the next time step. Beliefs over this updated set of candidate models are calculated using the standard inference methods involving the dependency relationships between the model nodes as shown in Fig. 7. We note the recursive nature of this solution: in solving agent i's level 1 I-DID, j's level 0 DIDs must be solved first. If the nesting of models is deeper, all models at all levels starting from 0 are solved in a bottom-up manner.

We briefly outline the recursive algorithm for solving agent i's level l I-DID expanded over T time steps with one other agent j in Fig. 15. We adopt a two-phase approach: Given a two time-slice I-DID of level l with all lower level models also represented as two time-slice I-DIDs or DIDs (if level 0), the first step is to expand the level l I-DID over T time steps adding the dependency links and the conditional probability distributions for each node. We particularly focus on establishing and populating the model nodes (lines 3–11). Note that Range$(\cdot)$ returns the values (lower level models) of the random variable given as input (model node). We consider j's action node A_j^t (line 5) and the observation node O_j^{t+1} (line 7). Both of them, together with the model node $M_{j,l-1}^t$, become parents of the new model node, $M_{j,l-1}^{t+1}$ (line 11). We add the model update link between $M_{j,l-1}^t$ and $M_{j,l-1}^{t+1}$. We build the new I-IDs at time $t+1$ and construct the I-DIDs by connecting relevant chance and decision nodes between time t and $t+1$ (line 12). We specify the CPDs that reflect the transition and observation functions in the dynamic IDs (line 13). In the second phase, if the input is an I-DID, we substitute the policy links, model nodes and the model update links between them in the expanded I-DID with the chance nodes and dependency relationships between them as per Fig. 7, resulting in a flat DID similar to Fig. 8 (lines 14–15). We may use the standard look-ahead technique projecting the action and observation sequences over T time steps in the future, and backing up the expected utility values of the reachable beliefs (see [36], and [31] for a more introductory description). Other more efficient ways of solving DIDs could also be used [6].

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

Solution Phase
14. **If** $l \geq 1$ **then**
15. Represent the model nodes, policy links and the model update links as in Fig. 7 to obtain the DID
16. Apply the standard look-ahead and backup method to solve the expanded DID (other solution approaches may also be used)

Fig. 15 Algorithm for exactly solving a level $l \geq 1$ I-DID or level 0 DID expanded over T time steps

Note that we may optimize the implementation of this algorithm by reusing computations. In particular, j's level $l-1$ models in the model node at time $t+1$ will contain the same beliefs as those encountered in the look-ahead search tree when the $l-1$ I-DID (or DID) is first solved. Because solving the I-DID (or DID) involves computing the solutions at these beliefs as well, we need not recursively invoke the algorithm for solving j's models at subsequent time steps. Instead, we may obtain it from the previously computed (and cached) solutions. In order to exploit this optimization, line 4 of Fig. 15 is performed only if $t = 0$, otherwise the previously computed solutions are utilized at subsequent time steps. These intermediate solutions should be stored for later use while performing line 16.

As we mentioned previously, the 0th level models are the traditional DIDs. Their solutions provide probability distributions over actions of the agent modeled at that level to I-DIDs at level 1. Given probability distributions over other agents' actions the level 1 I-DIDs can themselves be solved analogously to DIDs, and provide probability distributions to yet higher level models. Assume that the number of models considered at each level is bound by a number, M. Solving an I-DID of level l is then equivalent to solving $O(M^l)$ DIDs. Depending on the values of M and l, the level l I-DID may be expensive to solve in practice.

Table 3 Run times for exactly solving both the I-DID and the I-POMDP for PG and online shopper's dilemma problems (Pentium 4, 3.0GHz, 1GB RAM, WinXP)

Problem	Representation	Runtime
Multiagent PG	I-DIDs	0.547 s
	I-POMDPs	12.166 s
Multiagent Shopping	I-DIDs	0.203 s
	I-POMDPs	0.435 s

5.1 Computational advantages of I-DIDs over I-POMDPs

I-DIDs explicitly model variable dependencies that are usually hidden in the enumerative representations of I-POMDPs. The compactness of I-DIDs makes it feasible to handle domains having multiple variables. For example, in the PG game we observe that we need to consider joint states of two variables, *PrivatePot* and *PublicPot*. Since *PrivatePot* and *PublicPot* have 6 and 11 possible values respectively in our context, we need to enumerate 66 states in the I-POMDP definition. Consequently, in the enumerative representation, we need to specify a large transition table (of size $66^2 \times 2 \times 2$ numbers since there are two decision options for each player), which grows exponentially in complexity. In contrast, the I-DID representation decomposes the complex state into the two variables, $PrivatePot_i^t$ and $PublicPot^t$, and models the dependencies over time. As we see in the I-DID in Fig. 12, the status of the private pot at time t does not affect the contents of the public pot at $t+1$. In addition, agent j's actions do not affect the private pot of i. Consequently, we only need to specify two smaller tables of at most $6^2 \times 2$ numbers in the CPD of $PrivatePot_i^{t+1}$ and $11^2 \times 2 \times 2$ numbers in the CPD of $PublicPot^{t+1}$. The outcome of this more compact representation is that the I-DID exhibits some computational advantages over the I-POMDP.

Table 3 shows the run times for solving the I-DID and the I-POMDP for two domains - PG and online shopper's dilemma. We used the I-DIDs shown in Figs. 12 and 14, for the two domains respectively. For comparison, we formulate the I-POMDP definitions of the two domains as in Sect. 3.1. Both the I-DIDs and the I-POMDPs are singly nested with two models of the other initially and each expanded to a horizon of three. We utilized a reduced version of the PG problem as our I-POMDP implementation is unable to solve the version with 66 states and two models of the other agent over a horizon of three.

We observe that the I-DID solves relatively efficiently in comparison to the enumerative representation of the I-POMDP for the PG problem. The I-DID exhibits a significant computational advantage because it adopts a factored representation of the state space and exploits the conditional independence when applying the look-ahead and backup methods during the solution. Further, the computational improvement is obtained because I-DIDs allow models of j to be also represented as DIDs. In comparison, for the online shopper's dilemma, the computational advantage is not significant as the state is simple, represented using a single variable. The reduced runtime is likely due to a more efficient implementation of the I-DID.

5.2 Example solutions

We continue with the illustrations and describe solutions of the example I-DIDs shown in Sect. 4.4. A good indicator of the usefulness of formalisms for decision making such as I-DIDs is the emergence of realistic social behaviors in their prescriptions. Hence, we focus on settings that simulate conditions sufficient for the emergence of such behaviors. We show how changes in the parameters of the problem and the models lead to interesting behaviors.

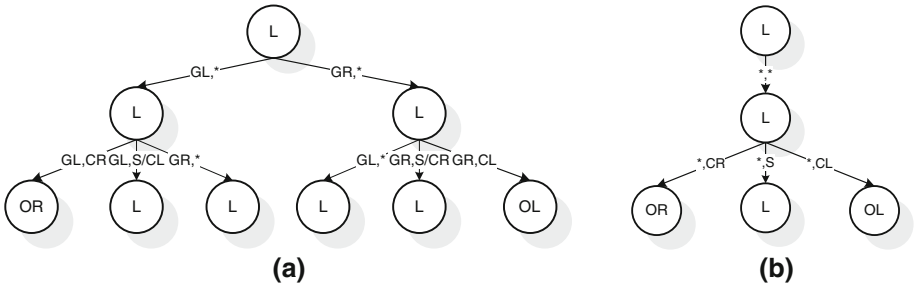

Fig. 16 Emergence of (**a**) conditional followership, and (**b**) blind followership in the third step in the tiger problem. Behaviors of interest are in bold. "*" is a wildcard, and denotes any one of the observations

5.2.1 Followership and leadership in the multiagent tiger problem

We consider a particular setting of the persistent multiagent tiger problem mentioned previously, in which agent i believes that j's preferences are similar to its own—both of them want to get the gold—and j's hearing is more reliable in comparison to itself. As an example, suppose that j, on listening can discern the tiger's location 95% of the times compared to i's 65% accuracy. Agent i does not have any initial information about the tiger's location. In other words, i's single-level nested belief, $b_{i,1}$, assigns 0.5 to each of the two locations of the tiger. In addition, i considers two models of j, which differ in j's flat level 0 initial beliefs. According to one model, j assigns a probability of 0.9 that the tiger is behind the left door, while the other model assigns 0.1 to that location. These extreme initial beliefs of j allow j to possibly open a door in the next time step itself. i is undecided on these two models of j.

If we vary i's hearing ability (by varying the probabilities in the CPD of the observation node, *Growl&Creak*), and solve the corresponding level 1 I-ID, shown in Fig. 9, expanded over three time steps, we obtain the normative behavioral policies shown in Fig. 16 that exhibit followership behavior. If i's probability of correctly hearing the growls is 0.65, then as shown in the policy in Fig. 16a, i begins to *conditionally* follow j's actions: i opens the same door that j opened previously iff i's own assessment of the tiger's location confirms j's pick. If i loses the ability to correctly interpret the growls completely, it *blindly* follows j and opens the same door that j opened previously (Fig. 16b).

We observed that a *single level* of belief nesting—beliefs about the other's models—was sufficient for followership to emerge in the tiger problem. However, the epistemological requirements for the emergence of leadership are more complex. For an agent, say j, to emerge as a leader, followership must first emerge in the other agent i. As we mentioned previously, if i is certain that its preferences are identical to those of j, and believes that j has a better sense of hearing, i will follow j's actions over time. Agent j emerges as a leader if it believes that i will follow it, which implies that j's belief must be nested *two levels* deep to enable it to recognize its leadership role. Realizing that i will follow presents j with an opportunity to influence i's actions in the benefit of the collective good or its self-interest alone.

For example, in the tiger problem, let us consider a setting in which if both i and j open the correct door, then each gets a payoff of 20 that is double the original. If j alone selects the correct door, it gets the payoff of 10. On the other hand, if both agents pick the wrong door, their penalties are cut in half. In this setting, it is in both j's best interest as well as the collective betterment for j to use its expertise in selecting the correct door, and thus be

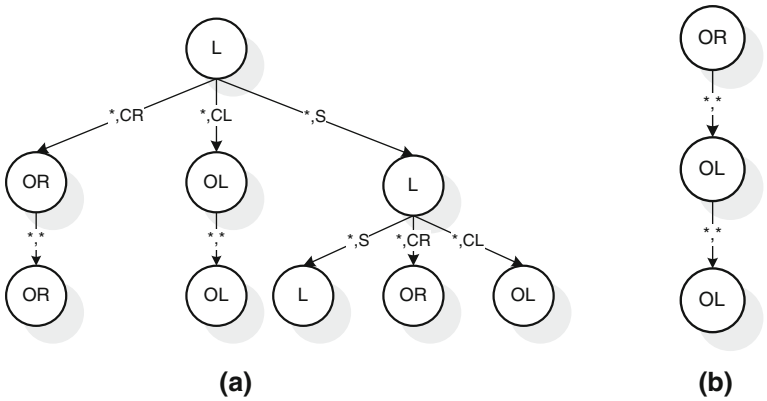

Fig. 17 Emergence of deception between agents in the tiger problem. Behaviors of interest are in bold. '*' denotes as before. (**a**) Agent i's policy demonstrating that it will blindly follow j's actions. (**b**) One of the two optimal policies. Even though j is almost certain that the tiger is on the right, it will start by selecting OR, followed by OL, in order to deceive i. Other optimal policy is to always open the left door, which does not involve deceiving i

a good leader. However, consider a slightly different problem in which j gains from i's loss and is penalized if i gains. Specifically, let i's payoff be subtracted from j's, indicating that j is antagonistic toward i - if j picks the correct door and i the wrong one, then i's loss of 100 becomes j's gain. Here, let the tiger persist in its original location with a probability of 1. Agent j believes that i (incorrectly) thinks that j's preferences are those that promote the collective good and that it starts off by believing with 99% confidence where the tiger is. Because i believes that its preferences are similar to those of j, and that j starts by believing almost surely that one of the two is the correct location (two level 0 models of j), i will start by following j's actions. We build the I-ID (shown in Fig. 9) so that agent j is at the top level and expand it over three time steps. We first show i's normative policy on solving its expanded I-DID in Fig. 17a. The policy demonstrates that i will blindly follow j's actions. Since the tiger persists in its original location with a probability of 1, i will select the same door again.

If j begins the game with a 99% probability that the tiger is on the right, solving j's I-DID nested *two levels* deep, results in two policies one of which is shown in Fig. 17b. Even though j is almost certain that OL is the correct action, it will start by selecting OR, followed by OL. Agent j's intention is to deceive i who, it believes, will follow j's actions, so as to gain \$110 in the second time step, which is more than what j would gain if it were to be honest. Here, j's expected reward in the first time step is: $(0.99 \times -99) + (0.01 \times 11) = -97.9$. Note that i listens in the first time step and incurs a reward of -1, which is subtracted from j's reward. Subsequently, when j does OL, the expected reward is: $(0.99 \times 110) + (0.01 \times -110) = 107.8$. Thus, the total of the first two steps is 9.9. The second optimal policy is the non-deceptive one where agent j always opens the left door. After the first two steps, the expected reward of j is 9.9 as well. Note that both policies open the left door in the last step. Thus, agent j could choose to deceive the other as both deceptive and non-deceptive policies are equally optimal.

5.2.2 Altruism and reciprocity in public good problem

Behaviors of human players in empirical simulations of the PG problem differ from the normative predictions. The experiments reveal that many players initially contribute a large amount to the public pot, and continue to contribute when the PG problem is played repeatedly,

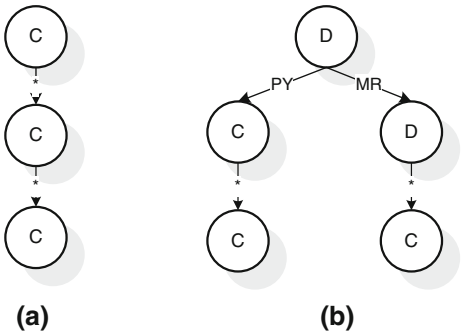

Fig. 18 (**a**) An altruistic level 1 agent always contributes. (**b**) A reciprocal agent i starts off by defecting followed by choosing to contribute or defect based on its observation of plenty (indicating that j is likely altruistic) or meager (j is non-altruistic)

though in decreasing amounts [5]. Many of these experiments [10] report that a small core group of players persistently contributes to the public pot even when all others are defecting. These experiments also reveal that players who persistently contribute have either altruistic or reciprocal preferences matching expected cooperation of others.

We formulate a sequential version of the PG problem with punishment mentioned previously, from the perspective of agent i. Borrowing from the empirical investigations of the PG problem [10], we construct level 0 IDs for j that model altruistic and non-altruistic types (Fig. 11). Specifically, our altruistic agent has a high marginal private return (c_j is close to 1) and does not punish others who defect. On the other hand, the non-altruistic type has a low marginal private return and punishes defectors. Let $x_c = 1$ and the level 0 agent be punished half the times it defects. With one action remaining, both types of agents choose to contribute to avoid being punished. With two actions to go, the altruistic type chooses to contribute, while the other defects. This is because c_j for the altruistic type is close to 1, thus the expected punishment, $0.5P > (1 - c_j)$, which the altruistic type avoids. Because c_j for the non-altruistic type is less, it prefers not to contribute. With three steps to go, the altruistic agent contributes to avoid punishment ($0.5P > 2(1 - c_j)$), and the non-altruistic type defects. For greater than three steps, while the altruistic agent continues to contribute to the public pot depending on how close its marginal private return is to 1, the non-altruistic type prescribes defection.

We analyzed the decisions of an altruistic agent i ($c_i = 0.95$, $P = 0.3$, $c_p = 0$) modeled using a level 1 I-DID expanded over 3 time steps. i ascribes the two level 0 models, mentioned previously, to j (see Fig. 11). If i believes with a probability 1 that j is altruistic, i chooses to contribute for each of the three steps. This behavior persists when i is unaware of whether j is altruistic (Fig. 18a, and when i assigns a high probability to j being the non-altruistic type. However, when i believes with a probability 1 that j is non-altruistic and will thus surely defect, i chooses to defect to avoid the punishment cost and because its marginal private return is less than 1. These results demonstrate that the behavior of our altruistic type resembles that found experimentally. The non-altruistic level 1 agent chooses to defect regardless of how likely it believes the other agent to be altruistic.

We analyzed the behavior of a reciprocal agent type ($c_i = 0.75$, $P = 0.3$, $c_p = 0.03$) that matches expected cooperation or defection. The reciprocal type's marginal private return is similar to that of the non-altruistic type, however, it obtains a greater payoff when its action is similar to that of the other. We consider the case when the reciprocal agent i is unsure of whether j is altruistic and believes that the public pot is likely to be half full. For this prior belief, i chooses to defect. On receiving an observation of plenty, i decides to contribute, while an observation of meager makes it defect (Fig. 18b. This is because an observation of

plenty signals that the pot is likely to be greater than half full, which results from j's action to contribute. Thus, among the two models ascribed to j, its type is likely to be altruistic making it likely that j will contribute again in the next time step. Agent i therefore chooses to contribute to reciprocate j's predicted action. An analogous reasoning leads i to defect when it observes a meager pot. With one action to go, i believing that j contributes, will choose to contribute too to avoid punishment regardless of its observations.

5.2.3 Mistrust in the online shopper's dilemma problem

The buyer's dilemma in the online shopping problem, described in Sect. 4.4, is, in part, due to its uncertainty over the seller's actions, which in turn are predicated on what the seller believes about the buyer's actions. The seller j, mistrusting the buyer i, especially in a portal that does not strictly enforce anti-fraud policies, may believe that the buyer will very likely transfer a partial amount of the agreed price. We represent this situation in a level 0 ID modeling the seller and utilize the following parameters: $v_i = 110$, $v_j = 90$, $c_{ij} = 100$, $\alpha = 0.8$, $\beta = 0.8$, and $\gamma = 0.8$. We assume that the punishments for cheating are larger if the portal is of a high quality ($p_i = p_j = 30$ and $p_{ij} = 15$) as compared to a portal that does not strictly guard against fraud ($p_i = p_j = 25$ and $p_{ij} = 10$).

Solution of the level 0 ID expanded over three time steps generates a policy that prescribes the seller to always deliver a low quality item no matter what the reviews about the portal indicate. This is because the loss expected by the seller in delivering a high quality item but receiving a partial amount is more than that expected from being punished for cheating. On the other hand, for a trusting seller that very likely believes that the buyer will transfer the full amount of money, the level 0 ID will prescribe the seller to deliver items of high quality irrespective of the review of the portal. This is due to the large punishment that the seller expects if it unilaterally decides to cheat by delivering a low quality item while receiving the full price. The expected punishment exceeds the gain expected from delivering a low quality item. Finally, if the seller is uncertain about the behavior of the buyer and the quality of the portal, it initially delivers a high quality item. Subsequently, positive or negative reviews about the portal will then guide the seller's action of delivering a high or low quality item, respectively.

A buyer modeled using the level 1 I-ID, shown in Fig. 13 and expanded to three time steps, which is uncertain whether the seller is trusting (delivers high quality items only) or not, will utilize its observations of the reviews of the portal to guide its actions. We show the corresponding policy tree in Fig. 19a. This behavior persists even when the buyer believes that the seller itself uses the reviews to guide its actions. However, a mistrusting buyer who believes that the seller is more likely to be mistrusting will transfer a partial amount of the money the first two times irrespective of the reviews, but will transfer the full amount if the review of the portal is still good. This is because two good reviews will shift the buyer's opinion of the seller to be trusting and consequently will deliver a high quality item (Fig. 19b).

We continue with the analysis by lifting the I-DIDs to one more level, and administer less punishment on the seller ($p_j = 19.5$ and $p_{ij} = 3$ for a good quality portal while $p_j = 19$ and $p_{ij} = 2$ for a portal of bad quality). We suppose that a seller modeled using an I-DID at level 2 believes that the buyer (at level 1) follows the policy given in Fig. 19a. As we mentioned before, this behavior of the buyer arises because the buyer is uncertain of whether the seller (at level 0) is trusting or not. If we consider a seller who strongly prefers to deliver a low quality item if the buyer transfers a full amount (regardless of the punishment incurred), the seller adopts a policy that it believes will deceive the buyer into likely transferring the full amount while it transfers a low quality item. We show the corresponding policy tree of the

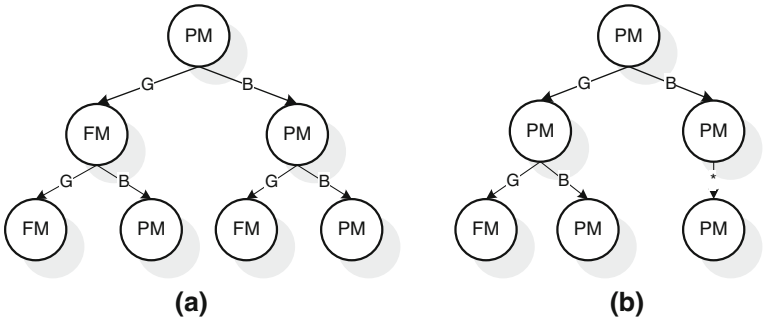

Fig. 19 (**a**) A buyer who is uncertain about whether the (level 0) seller is trusting or not utilizes the portal reviews to condition its actions. (**b**) A mistrusting buyer who believes that the seller is likely mistrusting will transfer a partial amount of the money except for the case where it observes good reviews of the portal twice

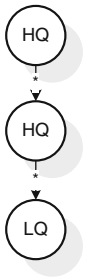

Fig. 20 Deceptive behavior of the seller modeled using a level 2 I-DID in the shopping problem. Though the seller expects the buyer to transfer a partial amount of money, it delivers high quality items. This misleads the buyer into believing that the portal is of a high quality and consequently transfers the full amount of money. At this point, the seller delivers a low quality item incurring the maximum profit

seller in Fig. 20. We first note that the seller gains the most if it delivers a low quality item while the buyer transfers the full amount. Notice that despite the seller standing to lose money immediately, it decides to deliver a high quality item while it expects the buyer to transfer a partial amount of money. This is followed by the delivery of another high quality item. Both these actions are deceptive as they serve to mislead the buyer (through its observations) into thinking that the portal is likely a good one, which causes the buyer to likely transfer the full amount in the final step. The seller expecting this delivers a low quality item in the last step.

6 Approximate solutions of I-DIDs

Because models of the other agent, j, are included as part of the model node in i's I-DID, solution of the I-DID suffers from not only the high dimensionality of the state space due to the possibly large number and complexity of models of j, but also the curse of history responsible for an exponential number of candidate models of j over time. We focus on mitigating the impact of these factors by holding constant the number of candidate models of j in the model node of the I-DID, at each time step.[4] In the following section, we show an approach for maintaining a constant number of models of the other agent over time.

[4] We do not focus on approximating the standard inference and dynamic programming techniques used in solving DIDs. See [25] for such an effort.

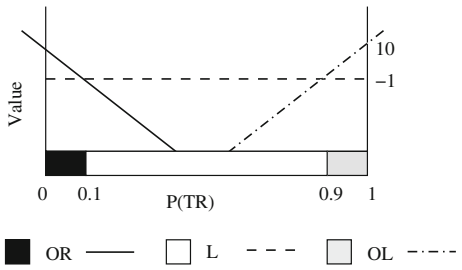

Fig. 21 Horizon 1 solution of j's level 0 model in the tiger problem. Note the belief ranges corresponding to different optimal actions

6.1 Model clustering

We explore an approximation technique based on clustering the agent models and selecting K, where $0 < K \ll M$, representative models from the clusters. In order to initiate clustering, we begin by identifying the initial means around which the models will be clustered. The selection of the initial means is crucial as we wish to select them minimally and avoid discarding models that are behaviorally distinct from the representative ones.

6.1.1 Selecting the initial means

For the sake of illustration, we assume that the models of j are intentional and differ only in their beliefs. Our arguments may be extended to models that differ in their frames and subintentional models as well. In order to selectively pick $0 < K \ll M$ models of j, we begin by identifying the *behaviorally equivalent* regions of j's belief space [30]. These are regions of j's belief simplex in which the beliefs lead to an identical optimal policy. As a simple example, we show in Fig. 21 the behaviorally equivalent regions of j's level 0 belief simplex for the tiger problem mentioned in Sect. 5.2. Here j's hearing is 85% accurate. The agent opens the right door (OR) if its belief that the tiger is behind the right door, *P(TR)*, is less than 0.1. It will listen (L) if $0.1 < P(TR) < 0.9$ and open left door (OL) if $P(TR) > 0.9$. Therefore, each of the optimal policies spans over multiple belief points. For example, OR is the optimal action for all beliefs in the set [0–0.1). Thus, beliefs in [0–0.1) are equivalent to each other in that they *induce the same optimal behavior*. However, notice that at $P(TR) = 0.1$, the agent is indifferent between OR and L.

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

Definition 2 (*SP*) Let $b_{j,l-1}$ be a level $l-1$ belief of agent j and $\text{OPT}(\langle b_{j,l-1}, \hat{\theta}_j \rangle)$ be the set of optimal policies for this belief. Then $b_{j,l-1}$ is a sensitivity point SP), if for any $\epsilon > 0$, there exists a belief, $b'_{j,l-1}$ s.t. $||b_{j,l-1} - b'_{j,l-1}||_1 < \epsilon$ and $\text{OPT}(\langle b_{j,l-1}, \hat{\theta}_j \rangle) \neq \text{OPT}(\langle b'_{j,l-1}, \hat{\theta}_j \rangle)$.

Referring to Fig. 21, *P(TR) = 0.1* is an SP because even infinitesimally small deviations from 0.1 lead to either OR or L as the optimal action, while at 0.1 the agent is indifferent between the two.

In order to compute the SPs, we observe that they are the beliefs at the non-dominated intersection points (or lines) between the value functions of pairs of policy trees. The linear program (LP) in Table 4 provides a straightforward way of computing the SPs. If the intersections are lines, then the LP returns a point on this line. For each pair of possible policies of j, π'_j and π''_j as input, we solve the LP in Table 4.

Table 4 LP for exact computation of SPs

$LP\_SP\ (\pi'_j, \pi''_j, \Pi_j)$	
Objective:	Constraints:
Maximize τ	$\forall \pi_j \in \Pi_j / \{\pi'_j, \pi''_j\}$
Variable:	$b_{j,l-1} \cdot Val_{j,l-1}(\pi'_j) - b_{j,l-1} \cdot Val_{j,l-1}(\pi_j) \geq \tau$
$b_{j,l-1}$	$b_{j,l-1} \cdot Val_{j,l-1}(\pi'_j) - b_{j,l-1} \cdot Val_{j,l-1}(\pi''_j) = 0$
	$b_{j,l-1} \cdot 1 = 1$

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

The LP needs to be solved $\mathcal{O}(|A_j|^{4|\Omega_j|^T})$ times to find the SPs exactly, which is computationally expensive. We approximate this computation by randomly selecting K policy trees from the space of policies and invoking $LP\_SP\ (\pi'_j, \pi''_j, \Pi^K_j)$, where Π^K_j is the reduced space of K policy trees, and $\pi'_j, \pi''_j \in \Pi^K_j$. Computation of the set of new SPs, denoted by SP_K, requires the solution of $\mathcal{O}(K^2)$ reduced LPs allowing computational savings.

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

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

6.1.2 Iterative clustering

The initial clusters group together models of the other agent possibly belonging to multiple behaviorally equivalent regions. Additionally, some of the SP_K may not be candidate models of j as believed by i. In order to promote clusters of behaviorally equivalent models and segregate the non-behaviorally equivalent ones, we update the means using an iterative method often utilized by the k-means clustering approach [22].

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

$$\bar{b}_{j,l-1} = \frac{\sum_{b_{j,l-1} \in \mathcal{B}^n_{j,l-1}} b_{j,l-1}}{|\mathcal{M}^n_{j,l-1}|} \tag{4}$$

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

Next, we recluster the models according to the proximity of their beliefs to the revised means. Specifically, models are grouped with the mean to which their respective beliefs are the closest, and all ties are broken randomly. The steps of recomputing the means (Eq. 4) and reclustering using the revised means are repeated until *convergence* ie. the means no longer

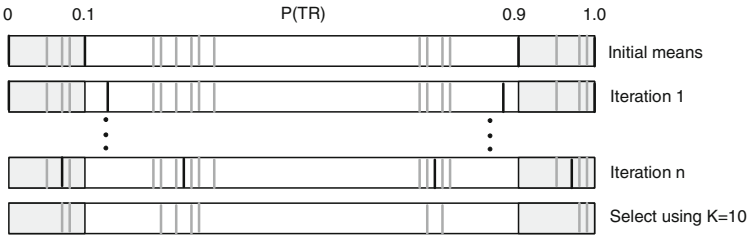

Fig. 22 An illustration of the iterative clustering method. The gray vertical lines are the belief points in the models while the black ones are the means. The SPs and the vertices of the belief simplex form the initial means. Notice the movement of the means over the iterations. Once the means have converged, we select $K = 10$ models

change. Intuitively, this iterative technique converges because over increasing iterations less new models will be added to a cluster, thereby making the means gradually invariant. We illustrate example movements of the means and clusters of beliefs over multiple iterations in Fig. 22.

6.1.3 Model selection

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

We compose the three steps of (i) identifying initial means, (ii) iterative clustering, and (iii) selecting K models in the algorithm *KModelSelection* shown in Fig. 23.

The algorithm for *KModelSelection* takes as input the set of models to be pruned, $M_{j,l-1}$, current horizon H of the I-DID, and the parameter K. We compute the initial means—these are the sensitivity points, SP_K, obtained by solving the reduced LP of Table 1 (line 1; vertices of the belief simplex may also be added). Each model in $M_{j,l-1}$ is assigned to a cluster based on the distance of its belief to a mean (lines 2–9). The algorithm then iteratively recalculates the means of the clusters and reassigns the models to a cluster based on their proximity to the new means of the clusters. These steps (lines 10–16) are carried out until the means of the clusters no longer change. Given the stabilized clusters, we calculate the contribution, k_n, of the nth cluster to the set K of models (line 18), and pick the k_n models from the cluster that are the closest to the mean (lines 19–20).

The models in the model node of i's I-DID, $M^{t+1}_{j,l-1}$, are pruned to include just the K models. These models form the values of the chance node, $Mod[M_j]$ in time step $t + 1$. We show the algorithm for approximately solving I-DIDs in Fig. 24. The algorithm is a slight variation of the one in Fig. 15 that solves I-DIDs exactly. In particular, on generating the candidate models in the model node, $M^{t+1}_{j,l-1}$, during the *expansion* phase (lines 3–9), we cluster and select K models of these using the procedure *KModelSelection*. Notice that models at all levels will be clustered and pruned. We note that our approach is more suited to situations where agent i has some prior knowledge about the possible models of others, thereby facilitating the clustering and selection.

KModelSelection ($M_{j,l-1}$, H, K)

Initial Means

1. Randomly pick K horizon H policy trees and invoke **LP_SP** to obtain the initial means
2. Means$_0 \leftarrow \{SP_K^1, SP_K^2, \ldots, SP_K^n\}$
3. **For** i **from** 1 **to** n **do**
4. $\quad \mathcal{M}^i_{j,l-1} \leftarrow \{SP^i_K\}$ /* *Initialize clusters**/
5. **For each** $m_{j,l-1} = \langle b_{j,l-1}, \hat{\theta}_{j,l-1} \rangle$ **in** $M_{j,l-1}$ **do**
6. $\quad SP^i_K \leftarrow \text{argmin}_{SP_K \in \text{Means}_0} ||SP_K - b_{j,l-1}||_1$
7. $\quad \mathcal{M}^i_{j,l-1} \overset{\cup}{\leftarrow} m_{j,l-1}$
8. **For** i **from** 1 **to** n **do**
9. $\quad \mathcal{M}^i_{j,l-1} \overset{-}{\leftarrow} \{SP^i_K\}$ if SP^i_K is not in $M_{j,l-1}$

Iteration

10. **Repeat**
11. $\quad$ **For** i from 1 to n **do**
12. $\quad\quad$ Recompute the mean of each cluster (Eq. 4)
13. $\quad$ **For each** $m_{j,l-1} = \langle b_{j,l-1}, \hat{\theta}_{j,l-1} \rangle$ **in** $M_{j,l-1}$ **do**
14. $\quad\quad \bar{b}^i_{j,l-1} \leftarrow \text{argmin}_{\bar{b}_{j,l-1}} ||\bar{b}_{j,l-1} - b_{j,l-1}||_1$
15. $\quad\quad \mathcal{M}^i_{j,l-1} \overset{\cup}{\leftarrow} m_{j,l-1}$
16. **Until** no change in the means

Selection

17. **For** i **from** 1 **to** n **do**
18. $\quad k_i \leftarrow \frac{|\mathcal{M}^i_{j,l-1}|}{|M_{j,l-1}|} \times K$
19. $\quad$ Sort the models in cluster i using distance from mean
20. $\quad \mathcal{M}_K \overset{\cup}{\leftarrow}$ top k_i models
21. **Return** $\mathcal{M}_K$

Fig. 23 Algorithm for clustering and selecting K models

6.2 Discussion

Although we need not recursively solve models of agent j at subsequent time steps in the I-DID since we could obtain their solutions from previous computations (see Sect. 5), clustering provides significant improvements. Specifically, it mitigates the impact of the curse of dimensionality affecting agent i and the curse of history afflicting j by reducing the number of models in the model node, at each time step and at every nesting level. The number of models otherwise increases exponentially. Hence, this saves on the size of the interactive state space. Furthermore, because typically, $K \ll M$, it helps reduce the space of models that are considered initially and thereby in subsequent time steps as well. All of this helps speed up the solution of I-DIDs and makes it possible to evaluate I-DIDs for longer horizons.

We selected the initial means as those that lie on the intersections of the behaviorally equivalent regions. This facilitates groups of behaviorally equivalent models to be grouped with a mean, and avoids behaviorally disparate models in the outer regions of a cluster, which may likely get pruned.

Other ways of selecting the means may also seem plausible. For example, the initial means could be the centers of the behaviorally equivalent regions. However, for small regions many

I-DID APPROX(LEVEL $l \geq 1$ I-DID OR LEVEL 0 DID, T, K)

Expansion Phase

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

Solution Phase

15. **If** $l \geq 1$ **then**
16. Represent the model nodes, policy links and the model update links as in Fig. 7 to obtain the DID
17. Apply the standard look-ahead and backup method to solve the expanded DID (other solution approaches may also be used)

Fig. 24 Algorithm for approximately solving a level $l \geq 0$ I-DID using model clustering

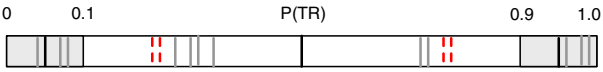

Fig. 25 Initial means are the centers of behaviorally equivalent regions. Belief points shown as red dashed lines are grouped into clusters that span multiple behavioral regions. As they are further away from the means, they will likely be discarded when representative models are selected

models that do not belong to the region and hence are not behaviorally equivalent may also be grouped together, as we illustrate in Fig. 25. As these models are likely to be further away from the means, they are prone to be pruned thereby contributing a larger loss in the optimality of the solution. Another way would be to distribute the initial means uniformly over the belief simplex. However, this approach is also likely to produce clusters with behaviorally disparate models in the outer regions, because the clusters may span over more than one behaviorally equivalent region.

A problem encountered in k-means clustering is that of the clustering converging to a local optimum—the final clusters may not accurately reflect the spatial distribution of the candidate models. This is in part due to the selection of the initial means around which the clustering is initiated. As we mentioned previously, we seek to form clusters of behaviorally equivalent models in order to avoid discarding models that are behaviorally disparate when

we select the representative models from each cluster. In this regard and as demonstrated above, we believe that our choice of the initial means achieves this objective. Furthermore, as we retain K models, we may end up picking models that were incorrectly clustered as we increase K. Thus the effect of increasing K is to reduce the influence of local optima. This is aptly demonstrated by our empirical results which show the quality of the solution approaching optimal as we increase K.

Finally, we note that the idea of behaviorally equivalent models also recently appeared in [29]. However, Pynadath and Marsella do not provide a method that involves clustering the models as we do. Furthermore, our approach is generally applicable to other representations (besides I-DIDs) that model other agents in a multiagent setting.

7 Computational savings, convergence and error bound

The computational complexity of solving I-DIDs is primarily due to the large number of models that must be solved over T time steps. At some time step t, the number of possible models of the other agent j is $M_0(|A_j||\Omega_j|)^t$ where M_0 is the number of models considered initially. The nested modeling further contributes to the complexity since solutions of each model at level $l-1$ requires solving the lower level $l-2$ models, and so on recursively down to level 0. Consider an $N+1$ agent setting in which the number of models is bounded by M at each level. Solving an I-DID at level l requires the solutions of $\mathcal{O}((NM)^l)$ many models. If the models are intentional, exact solutions of the models are at least NP Complete. This complexity precludes practical implementations of I-DIDs beyond simple problems. The approximation technique we consider here reduces the complexity by holding a constant number of K models in the model node. Thus, we only need to solve $\mathcal{O}((KN)^l)$ number of models at the first time step in comparison to $\mathcal{O}((MN)^l)$, where M grows exponentially over time. In general, the setting of $K \ll M$ offers a substantial reduction in the computation.

As the set of K retained models differs at each time step, the approximate value function may not converge asymptotically. We focus on bounding the error introduced by the approximation technique in the value of the optimal t-horizon policy tree for j. Here, we bound the error introduced by the approximation technique given that lower level models are solved exactly. While the usefulness of the bounds is limited, they are applicable to, for example, level 1 I-DIDs when the level 0 DIDs are solved exactly.

We bound the error introduced in j's behavior due to excluding all but K models at the time step t. Note that the K models are assumed to be solved exactly. Recall that for some cluster n, we retain the k_n models closest to the mean. If $K = M$, then we retain all the models and the error is zero. Let $\mathcal{M}_K$ denote the set of K models and $\mathcal{M}_{/K}$ denote the set of the $M-K$ models that are pruned. The error may be bounded by finding the model among the K retained models whose belief is spatially the closest to that of the discarded one. Define d_K as the largest of the distances between a pruned model, $m_{j,l-1}$, and the closest model among the K selected models: $d_K = max_{m_{j,l-1} \in \mathcal{M}_{/K}} min_{m'_{j,l-1} \in \mathcal{M}_K} ||b_{j,l-1} - b'_{j,l-1}||_1$, where $b_{j,l-1}$ and $b'_{j,l-1}$ are the beliefs in $m_{j,l-1}$ and $m'_{j,l-1}$, respectively. Given d_K, the derivation of the error bound for j proceeds in a manner analogous to that for point-based value iteration [27], though over the finite horizon, T, of the I-DID, as we show below.

Let $b_{j,l-1}$ be the discarded nested belief of j where the worst error is made: $b_{j,l-1} = \underset{b_{j,l-1} \in \mathcal{B}_{j,l-1}}{\text{argmax}} \; |b_{j,l-1} \cdot \alpha - b_{j,l-1} \cdot \alpha'|$. Here, $\mathcal{B}_{j,l-1}$ is the space of level $l-1$ beliefs of j, α is the value function associated with the policy tree optimal at $b_{j,l-1}$ and α' is the value function

associated with the policy tree optimal at a belief, $b'_{j,l-1}$, of a retained model that is closest to $b_{j,l-1}$. Then,

$$\begin{aligned}
\epsilon_K &= |b_{j,l-1} \cdot \alpha - b_{j,l-1} \cdot \alpha'| \\
&= |(b_{j,l-1} \cdot \alpha - b_{j,l-1} \cdot \alpha') + (b'_{j,l-1} \cdot \alpha - b'_{j,l-1} \cdot \alpha)| \quad \text{(add zero)} \\
&\leq |(b_{j,l-1} \cdot \alpha - b_{j,l-1} \cdot \alpha') + (b'_{j,l-1} \cdot \alpha' - b'_{j,l-1} \cdot \alpha)| \quad (b'_{j,l-1} \cdot \alpha' \geq b'_{j,l-1} \cdot \alpha) \\
&\leq ||\alpha - \alpha'||_\infty \cdot ||b_{j,l-1} - b'_{j,l-1}||_1 \quad \text{(Hölder inequality)} \\
&\leq (R_j^{max} - R_j^{min}) T \times d_K \qquad (5)
\end{aligned}$$

The error bound in Eq. 5 does not bound the error in agent i's exact policy due to the approximation—this depends on the expected behavior of j and not on the value of j's policy. It measures the worst-case error in j's policy introduced by the approximation technique at some nesting level l. The equation also assumes that the I-DIDs at the lower levels have been solved exactly. However, as we mentioned previously, we may use the approximations recursively at all levels of nesting to approximately solve the I-DIDs. In this case, the bounds shown here may be tighter than desired.

8 Empirical results

We implemented the approximation algorithm in Fig. 24 and demonstrate the empirical performance of the model clustering approach on two problem domains: the multiagent tiger problem (tiger's location resets if a door is opened) and a multiagent version of the machine maintenance problem [34], both of which are described in the Appendix. In particular, we show that the quality of the policies generated using our method approaches that of the exact policy as K increases. As there are infinitely many computable models, we obtain the exact policy by *exactly* solving the I-DID given a finite set of M_0 models of the other agent initially. In addition, we obtain significant computational savings, in comparison with the exact method, from using the approximation techniques as indicated by the low run times.

8.1 Performance profiles

We begin our empirical analysis by reporting the performance of the model clustering based approximate solutions of I-DIDs. In Figs. 26 and 27, we show agent i's average rewards gathered by executing 3 and 4 horizons policies obtained from solving the level 1 I-DIDs approximately. Each data point here is the average of 50 runs where the true model of the other agent, j, is randomly picked according to i's belief distribution over j's models. Each curve within a plot is for a particular M_0, where M_0 denotes the *total* number of candidate models of j at the first time step. Note that this increases exponentially over time.

We observe from the line plots in Figs. 26 and 27 that as we increase the number of models retained, K, the policies improve and converge toward the exact. This remains true for increasing M_0 and for both, the multiagent tiger and machine maintenance problem domains.

8.2 Runtime comparison

We show the run times of the exact and approximate approaches (denoted as MC) in Table 5 which are indicative of the computational savings incurred by pruning the model space to a fixed number of models at each time step in the I-DID. We observe that the approximation

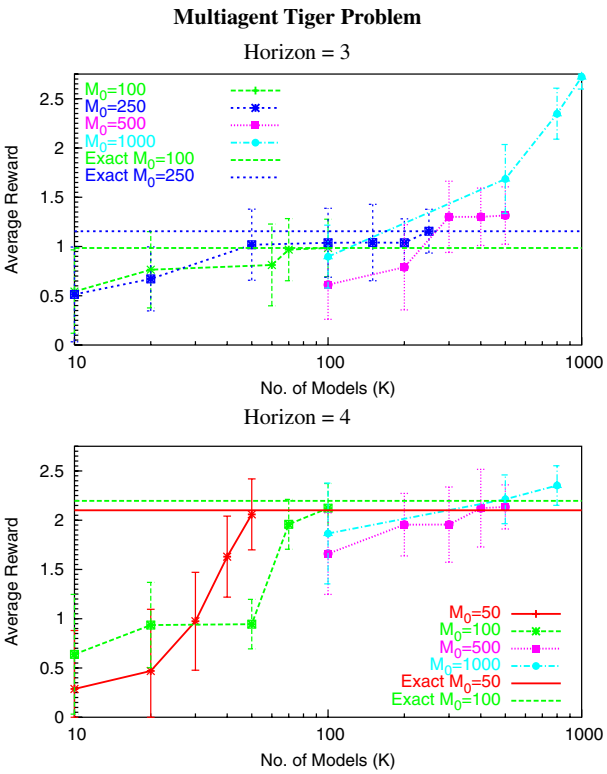

Fig. 26 Performance of the model clustering approach in comparison to the exact solutions on the multiagent tiger problem (standard deviation shown as vertical lines). As we increase K, the approximate solutions converge toward the exact. We do not show the exact solutions for larger values of M_0 as they could not be computed

technique demonstrates significant speedup in comparison to the exact solutions. Note that the speedups increase with the number of horizons. This is because the number of candidate models of the other agent increases exponentially with time for the exact approach but remains fixed in the approximation technique. Using the approximation we were able to solve our I-DIDs up to 8 horizons, while the exact solutions could not be obtained beyond 4 horizons. We expect similar results for deeper levels of strategic nesting of the models.

9 Conclusion

We showed how the traditional DIDs may be extended to I-DIDs that enable sequential decision making in uncertain multiagent settings. Our graphical representation of I-DIDs improves on the previous work significantly by being transparent, semantically clear, and capable of being solved using standard algorithms that target DIDs. I-DIDs extend NIDs to allow sequential decision making over multiple time steps in the presence of other interacting agents. I-DIDs may be seen as concise representations for I-POMDPs providing a graphical language to exploit problem structure and carry out decision making as the agent acts and observes given its prior beliefs.

Because I-DIDs include models of other agents in the representation as well, solving them is computationally complex. We presented the first technique for obtaining approximate solutions to I-DIDs which selects a constant number of representative models at each time step. Our approach was to reformulate the well-known k-means clustering method in the context

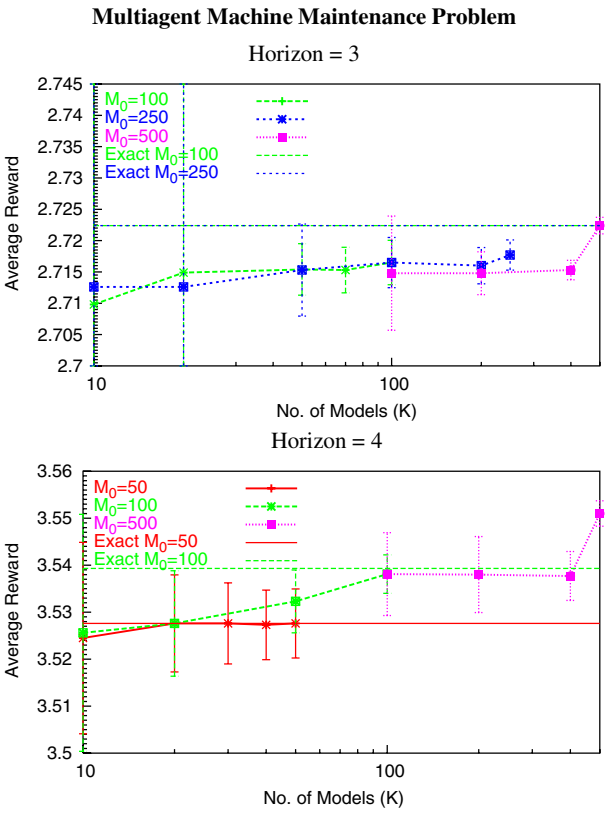

Fig. 27 Performance of the model clustering approach in comparison to the exact solutions on the machine maintenance problem (standard deviations shown as vertical lines). Note that the two horizontal lines for exact solutions in the top figure are too close to be distinguished. As before, the approximate solutions converge toward the exact as we increase K

Table 5 Run times for exactly and approximately solving the I-DID for different steps

Problem	Method	Horizons			
		$t = 2$	$t = 3$	$t = 4$	$t = 5$
Multiagent	Exact	14.079s	33.142s	83.644s	*
Tiger	MC	4.532s	7.110s	10.512s	12.328s
Multiagent	Exact	14.234s	35.847s	99.236s	*
Machine maintenance	MC	8.500s	12.908s	18.688s	33.219s

K and M are equal to 50 and 100 respectively for both approximate and exact approaches (Pentium 4, 3.0GHz, 1GB RAM, WinXP). *Exact solutions ran out of memory

of I-DIDs by strategically initializing the means and obtaining stable clusters of models in an iterative manner. Each cluster consists of models that are likely to be behaviorally equivalent. We select a subset of models from each cluster and update the selected models over time. The technique significantly mitigates the impact of the curse of dimensionality and reduces the space of agents' models in the expansion phase without significantly compromising on the solutions of I-DIDs. We provided empirical performances on the well-known multiagent tiger and a multiagent version of the classical machine maintenance problems. They show that the approach saves on computations over the model space.

As spaces of candidate models are often bounded, the true model of the other agent may not be within the model space. In this context, techniques for identifying models that are

relevant in predicting the true behavior are needed. We are investigating ways of identifying these relevant models using information-theoretic measures of similarity between observed and predicted behaviors.

Acknowledgement Prashant Doshi was supported in part by a grant #FA9550-08-1-0429 from the US Air Force Office of Scientific Research (AFOSR) and in part by a grant from the UGA Research Foundation. The authors thank the anonymous reviewers for their valuable comments.

Appendices

A Multiagent tiger problem

Our multiagent tiger problem is a generalization of the well-known single agent tiger problem [19] to the multiagent setting. It differs from other multiagent versions of the same problem [24] by assuming that the agents hear creaks as well as the growls. Creaks are indicative of which door was opened by the other agent(s). For the sake of simplicity, we restrict ourselves to a two-agent setting, but the problem is extensible to more agents in a straightforward way.

In the two-agent tiger problem, each agent may open doors or listen. To make the interaction more interesting, in addition to the usual observation of growls, we added an observation of door creaks, which depends on the action executed by the other agent. Creak right (CR) is likely due to the other agent having opened the right door, and similarly for creak left (CL). Silence (S) is a good indication that the other agent did not open doors and listened instead. We assume that the accuracy of creaks is 90%, while the accuracy of growls is 85% as in the single agent problem. We consider two settings, one in which the tiger persists in its original location with a probability of 0.95 if any of the agents opened any doors in the current step, the other in which the tiger location is chosen randomly in the next time step if a door is opened. We also assume that the agent's payoffs are analogous to the single agent version. Note that the 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.

We showed the nested I-DID unrolled over two time steps for the multiagent tiger problem in Fig. 10. Agent i at level 1 considers M models of agent j of level 0 which, for example, differ in the distributions over the chance node *Tiger Location*. In agent i's I-DID, we assign the marginal distribution over the tiger's location to the CPD of the chance node $TigerLocation_i^t$. In the next time step, the CPD of the chance node $TigerLocation_i^{t+1}$ conditioned on $TigerLocation_i^t$, A_i^t, and A_j^t is the transition function, shown in Table 6.

We show the CPD of the observation node, $Growl\&Creak_i^{t+1}$, in Table 7. The CPDs of the observation nodes in level 0 DIDs are identical to the observation function in the single agent tiger problem.

The decision node A_i^t includes possible actions of agent i in the scenario such as *listening (L)*, *opening the left door (OL)*, and *opening the right door (OR)*. The utility node R_i in the level 1 I-DID relies on both agent's actions, A_i^t and A_j^t, and the physical states, $TigerLocation_i^t$. We show the utility table in Table 8. The utility tables for level 0 models are identical to the reward function in the single agent tiger problem 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.

Table 6 CPD of the chance node $TigerLocation_i^{t+1}$ in the I-DID of Fig. 10 when the tiger (a) likely persists in its original location on opening doors, and (b) randomly appears behind any door on opening one

$\langle a_i^t, a_j^t \rangle$	TigerLocation_i^t	TL	TR
(a)			
$\langle OL, * \rangle$	*TL*	0.95	0.05
$\langle OL, * \rangle$	*TR*	0.05	0.95
$\langle OR, * \rangle$	*TL*	0.95	0.05
$\langle OR, * \rangle$	*TR*	0.05	0.95
$\langle *, OL \rangle$	*TL*	0.95	0.05
$\langle *, OL \rangle$	*TR*	0.05	0.95
$\langle *, OR \rangle$	*TL*	0.95	0.05
$\langle *, OR \rangle$	*TR*	0.05	0.95
$\langle L, L \rangle$	*TL*	1.0	0
$\langle L, L \rangle$	*TR*	0	1.0
(b)			
$\langle OL, * \rangle$	*	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 7 The CPD of the chance node $Growl\&Creak_i^{t+1}$ in the level 1 I-DID

$\langle a_i^t, a_j^t \rangle$	TgrLoc_i^{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

Table 8 Reward functions of agents i and j for the multiagent tiger problem

$\langle a_i^t, a_j^t \rangle$	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

Finally, the CPD 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.

B Multiagent machine maintenance problem

We extend the traditional single agent based machine maintenance (MM) problem [34] to a two-agent cooperative version. The original MM problem involved a machine containing

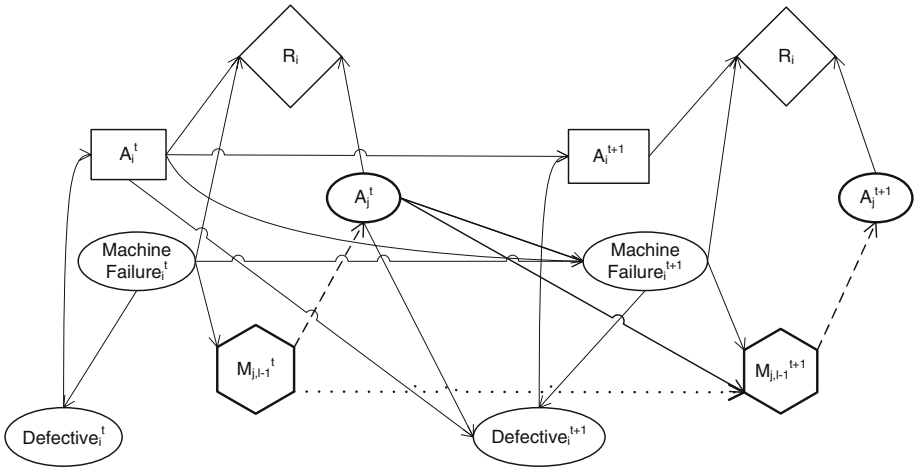

Fig. 28 Level 1 I-DID of agent i for the multiagent MM problem. The hexagonal model node contains M level 0 DIDs (or IDs at horizon 1) of agent j

Table 9 The CPD of the chance node, Machine Failure$_i^{t+1}$, in the level 1 I-DID of agent i

$\langle a_i^t, a_j^t \rangle$	Mch Fail$_i^{t+1}$	0-fail	1-fail	2-fail
⟨M/E,M/E⟩	0-fail	0.81	0.18	0.01
⟨M/E,M/E⟩	1-fail	0.0	0.9	0.1
⟨M/E,M/E⟩	2-fail	0.0	0.0	1.0
⟨M,I/R⟩	0-fail	1.0	0.0	0.0
⟨M,I/R⟩	1-fail	0.95	0.05	0.0
⟨M,I/R⟩	2-fail	0.95	0.0	0.05
⟨E,I/R⟩	0-fail	1.0	0.0	0.0
⟨E,I/R⟩	1-fail	0.95	0.05	0.0
⟨E,I/R⟩	2-fail	0.95	0.0	0.05
⟨I/R,*⟩	0-fail	1.0	0.0	0.0
⟨I/R,*⟩	1-fail	0.95	0.05	0.0
⟨I/R,*⟩	2-fail	0.95	0.0	0.05

two internal components operated by a single agent. Either one or both components of the machine may fail spontaneously after each production cycle (0-fail: no component fails; 1-fail: 1 component fails; 2-fail: 2 components fail). If an internal component has failed, then there is some chance that when operating upon the product, it will cause the product to be defective. An agent may choose to manufacture the product (M) without examining it, examine the product (E), inspect the machine (I), or repair it (R) before the next production cycle. On an examination of the product, the subject may find it to be defective. Of course, if more components have failed, then the probability that the product is defective is greater.

We design a level 1 I-DID for the multiagent MM problem in Fig. 28. We consider M models of agent j at level 0 which differ in the probability that j assigns to the chance node *Machine Failure*$_j$. In the I-DID, the chance node, *Machine Failure*$_i^{t+1}$, has incident arcs from the nodes *Machine Failure*$_i^t$, A_i^t, and A_j^t. The CPD of the chance node is shown in Table 9.

Table 10 The CPD of the chance node, $Defective_i^{t+1}$

$\langle a_i^t, a_j^t \rangle$	Mch fail$_i^{t+1}$	Not-defective	Defective
⟨M,M/E⟩	*	0.5	0.5
⟨M,I/R⟩	*	0.95	0.05
⟨E,M/E⟩	0-fail	0.75	0.25
⟨E,M/E⟩	1-fail	0.5	0.5
⟨E,M/E⟩	2-fail	0.25	0.75
⟨E,I/R⟩	*	0.95	0.05
⟨I/R,*⟩	*	0.95	0.05

Table 11 Reward function for agent i.

$\langle a_i^t, a_j^t \rangle$	0-fail	1-fail	2-fail
⟨M,M⟩	1.805	0.95	0.5
⟨M,E⟩	1.555	0.7	0.25
⟨M,I⟩	0.4025	−1.025	−2.25
⟨M,R⟩	−1.0975	−1.525	−1.75
⟨E,M⟩	1.5555	0.7	0.25
⟨E,E⟩	1.305	0.45	0.0
⟨E,I⟩	0.1525	−1.275	−2.5
⟨E,R⟩	−1.3475	−1.775	−2.0
⟨I,M⟩	0.4025	−1.025	−2.25
⟨I,E⟩	0.1525	−1.275	−2.5
⟨I,I⟩	−1.0	−3.00	−5.00
⟨I,R⟩	−2.5	−3.5	−4.5
⟨R,M⟩	−1.0975	−1.525	−1.75
⟨R,E⟩	−1.3475	−1.775	−2.0
⟨R,I⟩	−2.5	−3.5	−4.5
⟨R,R⟩	−4	−4	−4

The reward function for a level 0 agent is identical to the one in the classical MM problem

For the observation chance node, $Defective_i^{t+1}$, we associate the CPD shown in Table 10. Note that arcs from $Machine\ Failure_i^{t+1}$ and the nodes, A_i^t and A_j^t, in the previous time step are incident to this node. The observation nodes in the level 0 DIDs have CPDs that are identical to the observation function in the original MM problem.

The decision node, A_i, consists of agent i's actions including *manufacture* (M), *examine* (E), *inspect* (I), and *repair* (R). It has one information arc from the observation node $Defective_i^t$ indicating that i knows the examination results before making the choice. The utility node R_i is associated with the utility table in Table 11.

The CPD 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.

References

1. Adam, B., & Dekel, E. (1993). Hierarchies of beliefs and common knowledge. *International Journal of Game Theory, 59*(1), 189–198.
2. Aumann, R. J. (1999). Interactive epistemology i: Knowledge. *International Journal of Game Theory, 28*(3), 263–300.

3. Boutilier, C. (1999). Sequential optimality and coordination in multiagent systems. In *Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 478–485). Stockhom, Sweeden.
4. Boutilier, C., & Poole, D. (1996). Computing optimal policies for partially observable decision processes using compact representations. In *Thirteenth Conference on Artificial Intelligence (AAAI)* (pp. 1168–1175). Portland, Oregon.
5. Camerer, C. (2003). *Behavioral game theory: Experiments in strategic interaction*. Princeton, New Jersey: Princeton University Press.
6. Charnes, J. M., & Shenoy, P. (2004). Multistage monte carlo methods for solving influence diagrams using local computation. *Management Science, 50*(3), 405–418.
7. Dennett, D. (1986). *Intentional systems*. Brainstorms: MIT Press.
8. Doshi, P., & Gmytrasiewicz, P. J. (2005). Approximating state estimation in multiagent settings using particle filters. In *Autonomous Agents and Multi-agent Systems Conference (AAMAS)* (pp. 320–327). Utrecht, Netherlands.
9. Doshi, P., & Gmytrasiewicz, P. J. (2005). A particle filtering based approach to approximating interactive pomdps. In *Twentieth Conference on Artificial Intelligence (AAAI)* (pp. 969–974). Pittsburg, PA.
10. Fehr, E., & Gachter, S. (2000). Cooperation and punishment in public good experiments. *American Economic Review, 90*(4), 980–994.
11. Fudenberg, D., & Levine, D. K. (1998). *The theory of learning in games*. Cambridge, MA: The MIT Press.
12. Fudenberg, D., & Tirole, J. (1991). *Game theory*. MIT Press.
13. 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). Melbourne, Australia.
14. Gmytrasiewicz, P., & Doshi, P. (2005). A framework for sequential planning in multiagent settings. *Journal of Artificial Intelligence Research (JAIR), 24*, 49–79.
15. Gmytrasiewicz, P. J., & Durfee, E. H. (2000). Rational coordination in multi-agent environments. *Journal of Autonomous Agents and Multi-Agent Systems, 3*(4), 319–350.
16. Guestrin, C., Koller, D., & Parr, R. (2001). Solving factored pomdps with linear value functions. In *Workshop on Planning under Uncertainty and Incomplete Information, IJCAI*. Seattle, Washington.
17. Harsanyi, J. C. (1967). Games with incomplete information played by bayesian players. *Management Science, 14*(3), 159–182.
18. Howard, R. A., & Matheson, J. E. (1984). Influence diagrams. In *Readings on the Principles and Applications of Decision Analysis* (pp. 721–762).
19. Kaelbling, L., Littman, M., & Cassandra, A. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence Journal, 101*(1–2), 99–134.
20. 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). Seattle, Washington.
21. Littman, M. (1994). Markov games as a framework for multiagent reinforcement learning. In *International Conference on Machine Learning (ICML)* (pp. 157–163). New Brunswick, New Jersey.
22. MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In L. M. LeCam & J. Neyman (Eds.), *Proceedings of the Fifth Berkeley Symposium on Mathematics, Statistics, and Probablity* (pp. 281–297). Berkeley, CA: UC Press.
23. Mertens, J. F., & Zamir, S. (1985). Formulation of bayesian analysis for games with incomplete information. *International Journal of Game Theory, 14*, 1–29.
24. 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). Acapulco, Mexico.
25. Nilsson, D., & Lauritzen, S. (2000). Evaluating influence diagrams using limids. In *Uncertainty in Artificial Intelligence (UAI)* (pp. 436–445). Stanford, California.
26. Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan-Kaufmann: Los Altos, California.
27. Pineau, J., Gordon, G., & Thrun, S. (2006). Anytime point-based approximations for large pomdps. *Journal of Artificial Intelligence Research (JAIR), 27*, 335–380.
28. Polich, K., & Gmytrasiewicz, P. (2006). Interactive dynamic influence diagrams. In *Game Theory and Decision Theory (GTDT) Workshop, AAMAS*. Hakodate, Japan.
29. Pynadath, D., & Marsella, S. (2007). Minimal mental models. In *Twenty-Second Conference on Artificial Intelligence (AAAI)* (pp. 1038–1044). Canada, Vancouver.
30. Rathnas, B., Doshi, P., & Gmytrasiewicz, P. J. (2006). Exact solutions to interactive pomdps using behavioral equivalence. In *Autonomous Agents and Multi-Agents Systems Conference (AAMAS)* (pp. 1025–1032). Hakodate, Japan.

31. Russell, S., & Norvig, P. (2003). *Artificial intelligence: A modern approach* (2nd edn). Prentice Hall.
32. Seuken, S., & Zilberstein, S. (2008). Formal models and algorithms for decentralized decision making under uncertainty. *Journal of Autonomous Agents and Multi-agent Systems*. doi:10.1007/s10458-007-9026-5.
33. Shachter, R. D. (1986). Evaluating influence diagrams. *Operations Research, 34*(6), 871–882.
34. Smallwood, R., & Sondik, E. (1973). The optimal control of partially observable markov decision processes over a finite horizon. *Operations Research (OR), 21*, 1071–1088.
35. Suryadi, D., & Gmytrasiewicz, P. (1999). Learning models of other agents using influence diagrams. In *International Conference on User Modeling* (pp. 223–232).
36. Tatman, J. A., & Shachter, R. D. (1990). Dynamic programming and influence diagrams. *IEEE Transactions on Systems, Man, and Cybernetics, 20*(2), 365–379.