Interactive Dynamic Influence Diagrams

Kyle Polich and **Piotr Gmytrasiewicz**
Department of Computer Science, University of Illinois at Chicago
Chicago, IL, 60607-7053, USA
E-mail: kpolich@cs.uic.edu, piotr@cs.uic.edu

Abstract

This paper extends the framework of dynamic influence diagrams (DIDs) to the multi-agent setting. DIDs are computational representations of the Partially Observable Markov Decision Processes (POMDP), which are frameworks for sequential decision-making in single agent settings. The Interactive Dynamic Influence Diagrams (I-DIDs), presented here, are computational representations of Interactive Partially Observable Markov Decision Processes (I-POMDPS). I-POMDPs generalize POMDPs to multi-agent settings by including the models of other agents in the state space. In I-DIDs agents maintains their beliefs over models of other agents. They then use these models to predict the other agents' likely behavior and compute their own best response given these predications. Models of other agents could themselves be I-DIDs, DIDs, or simply probability distributions over their actions. The possibility that models are I-DIDs leads to recursive nesting of models. To ensure that models are always computable we assume that the nesting is finite. The solution process solves lower level models and incorporates the predicted behavior into the upper level ones thus converting them into classical DIDs. Since the framework is sequential, agents update their beliefs about the world and about the other agents as they receive new information using Bayesian update.

Introduction

Partially Observable Markov Decision Processes (POMDPs) emerged as the primary framework for decision-theoretic planning in single agent settings. Solutions to POMDPs are optimal plans which are conditional on future observations. Dynamic Influence Diagrams (DIDs) are computational representations of POMDPs which compute solutions for finite time horizons in an on-line fashion. Interactive POMDPs (I-POMDPs) (Gmytrasiewicz & Doshi 2005) generalize POMDPs to multi-agent settings by including models of other agents in the state space. Interactive DIDs (I-DIDs), presented in this paper, are computational representations of I-POMDPs, and thus generalizations of DIDs. DIDs are themselves temporal generalizations of influence diagrams (Howard & Matheson 1984).

Solutions of I-POMDPs, computed by I-DIDs, are conditional plans which take into account the continually revised probabilistic prediction of other agents' behavior. The predictions are formed based on the models of the other agents. The models can themselves be I-DIDs, or, in simpler cases, DIDs or probability distributions over others' actions (Gmytrasiewicz & Doshi 2005). The possible nesting of agents' beliefs about each other's models has been studied in recent advances in game theory (Aumann & Heifetz 2002; Battigalli & Bonano 1998; Battigalli & Siniscalchi 1999; Mertens & Zamir 1985). While the nesting of beliefs could be infinite, we assume finite nesting to ensure computability of the belief updates.[1] I-DIDs, analogously to DIDs, use a "forward" solution method, and do not rely on computing the value function over the whole belief simplex. In other related work (Rathnasabapathy, Doshi, & Gmytrasiewicz 2006) we pursue that approach to solving I-POMDPs.

The solutions maximize the agent's expected utility, and thus do not rely on the notion of equilibrium. This approach has been called the decision-theoretic approach to game theory by some authors (Kadane & Larkey 1982; Myerson 1991). Thus, as in POMDPs, the solutions are intended to be computed by each agent individually during their own on-line planning effort. Briefly, the main reasons why equilibria are not suitable for on-line sequential planning performed by the individual agents are that they are incomplete and not unique. Equilibria are incomplete since they do not specify an agent's action if the agent is not certain that it is in an equilibrium. They are not unique since there may be multiple equilibria with no clear criteria based on which the agent could chose one of them.

Several equilibrium based approaches to multi-agent decision making have been put forward. One such framework is Multi-Agent Influence Diagrams (MAIDs) framework (Koller & Milch 2001). MAIDS describe a multi-agent

[1] A question frequently asked in this regard is: What is the appropriate level on which the nesting of models should be terminated? On this issue we would like to make few brief points. First, the question is really about the level of detail included in describing the models of other agents. Second, this is analogous to the issue of choosing the level of detail used to model the environment in classical POMDPs. Third, the choice of information to include in the model is a matter of trading off computational resources and the quality of the solution obtained. Finally, in POMDPs and in I-POMDPs alike, the solutions may depend on the level of detail included.

setting using influence diagrams with multiple decision and reward nodes. They capture the information about the environment, the decisions the agents can make, and the rewards each agent will receive. The solution concept used in MAIDs is an equilibrium.

Influence Diagram Networks (IDNs) are a generalization of MAIDs which relax the assumption of common knowledge of the game (Gal & Pfeffer 2003). In an IDN, the first agent (say i) considers the possibility that the second agent, j, may be misinformed about the game they are playing in various ways. If so it is assumed that j would compute and act according to a "false" equilibrium. In this case, i should not respond with any of the corresponding equilibrium behaviors but instead compute its best response to the probabilistic mixture of the alternative equilibrium behaviors of j. Agent i may also entertain the possibility that j models agent i in a similar manner. This gives rise to a recursive structure which is assumed to always terminate with equilibria solutions at the leaves. Like MAIDs, IDNs are not suitable for sequential planning – they involve no belief update over time and are static recommendations as to the agent's behavior.

Other work includes Multi-Agent Markov Decision Processes (MMDPs) which focus on coordination in teams of agents with a common reward (Boutilier 1999). In MMDPs the state space encompasses the, a priori given, coordination mechanisms agents could use. The agents deliberate about (and learn) the best coordination mechanism to use. Related are decentralized POMDPs (DEC-POMDPs) which computes a joint equilibrium policy for agents with a common reward function by a central controller (Nair *et al.* 2003; Bernstein *et al.* 2002). The resulting policies for each agent are then distributed to them for execution.

In contrast to the teamwork frameworks, (Littman 1994) presents a framework using Markov decision processes and reinforcement learning applied to two player zero sum stochastic games. This approach assumes perfect observability of all agent's actions and perfect observability of the one's own reward from the previous time step. Optimal policies can be found efficiently. The reinforcement learning approach to multi-agent games is further explored in (Hu & Wellman 1998), intended for general-sum stochastic games. There, agents learn and converge on an equilibrium play, but, if there are multiple equilibria, there is no guarantee that the agents arrive at the same one.

Yet another approach is presented in (Emery-Montemerlo *et al.* 2004). It presents an approximation technique for multi-agent games in which decisions at each time step are computed by each agent using the same framework with one step look-ahead. The value of the plans beyond the look-ahead are approximated using a heuristic. The models of the other agent are taken from a finite pre-specified set, assumed to be common knowledge.

(Chang & Kaelbling 2001) presents a hierarchy for categorization of decision making and modeling other agent(s) based on the sizes of the game's history used during decision making. The most general category is H^{∞} x B^{∞}, denoting that an agent considers the entire available history and allows its models of another agent to consider the entire history as well. I-POMDPs are in this category because beliefs

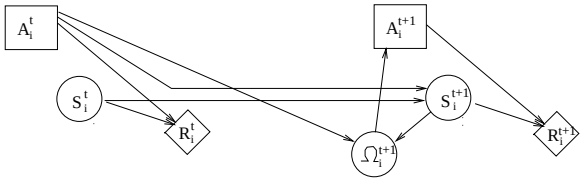

Figure 1: A Two Time Horizon Dynamic Influence Diagram

used in our approach are sufficient statistics for any observation history.

Single Agent Decision Making

A partially observable Markov decision processes (POMDP) (Boutilier, Dean, & Hanks 1999; Hauskrecht 2000; Kaelbling, Littman, & Cassandra 1998; Monahan 1982) of an agent i is defined as

$$\textit{POMDP}_i = \langle S, A_i, T_i, \Omega_i, O_i, R_i \rangle \quad (1)$$

where: S is a set of possible states of the environment, A_i is a set of actions agent i can execute, T_i is a transition function – $T_i : S \times A_i \times S \rightarrow [0, 1]$ which describes results of agent i's actions, Ω_i is the set of observations that the agent i can make, O_i is the agent's observation function – $O_i : S \times A_i \times \Omega_i \rightarrow [0, 1]$ which specifies probabilities of observations if agent executes various actions that result in different states, R_i is the reward function representing the agent i's preferences $R_i : S \times A_i \rightarrow \mathbf{R}$.

The solution of a POMDP is a policy which maps the observable history of the game to actions (Kaelbling, Littman, & Cassandra 1998). A dynamic influence diagram is a computational representation of a POMDP. The nodes in a DID, like the one in Figure 1, are named for the elements of the POMDP which they represent. DIDs perform planning using a forward exploration technique known as reachability analysis. This technique explores the possible states of belief an agent may be in in the future, the likelihood of reaching each state of belief, and the expected utility of each belief state. The agent then adopts the plan which maximizes the expected utility. DIDs provide exact solutions for finite horizon POMDP problems, and finite look-ahead approximations for POMDPs of infinite horizon.

Finitely Nested I-POMDPs

I-POMDPs generalize the POMDP framework to multi-agent environments (Gmytrasiewicz & Doshi 2005). Agent i considers the finitely nested $interactive$ state space $IS_{i,l}$ = $S \times M_j$ where S is the physical state and M_j is the set of models of agent j (for simplicity, we assume only two agents.) l is the level of nesting of i's I-POMDP. These models may include models j could have of i, but are constrained to be nested to the level not greater than $l - 1$. Particular models may be referred to as m_j or, if the models are intentional (see (Gmytrasiewicz & Doshi 2005) for details) θ_j.

(I-POMDP) An *I-POMDP* of agent i, is:

$$\textit{I-POMDP}_{i,l} = \langle IS_{i,l}, A, T_i, \Omega_i, O_i, R_i \rangle \quad (2)$$

An agent may receive evidence about the physical state of the world and/or the action taken by agent j. Agent i will compute the behavior for all the models in M_j and update its belief over the set $IS_{i,l}$ given its observations. Formally, the belief update is defined as (see (Gmytrasiewicz & Doshi 2005) for derivation):

$$\begin{aligned} b_i^t(is^t) &= Pr(is^t|o_i^t, a_i^{t-1}) \\ &= \beta \sum_{IS^{t-1}} \sum_{a_j^{t-1}} Pr(a_j^{t-1}|\theta_j^{t-1}) O_i(is^t, a^{t-1}, o_i^t) \\ &\times \sum_{o_j^t} \tau_{\theta_j^t}(b_j^{t-1}, a_j^{t-1}, o_j^t, b_j^t) O_j(is_j^t, a^{t-1}, o_j^t) \\ &\times T_i(is^{t-1}, a^{t-1}, is^t) \end{aligned} \tag{3}$$

where $\theta_i = \langle b_i, A_i, \Omega_i, T_i, O_i, R_i, OC_i \rangle$, $is = (s, \theta_j)$, $is_j = (s, \theta_i)$, b_j^{t-1} and b_j^t are the belief elements of θ_j^{t-1} and θ_j^t, respectively, β is a normalizing constant, O_j is the observation function in θ_j^t, and $Pr(a_j^{t-1}|\theta_j^{t-1})$ is the probability that a_j^{t-1} is Bayesian rational for agent described by type θ_j^{t-1}. $\tau_{\theta_i}(b_i^{t-1}, a_i^{t-1}, o_i^t, b_i^t)$ stands for $Pr(b_i^t|b_i^{t-1}, a_i^{t-1}, o_i^t)$. OC_i is the optimality criterion used by agent i.

The τ_{θ_j} function performs agent j's belief update. The update equation makes explicit the fact that agent i must perform agent j's belief update as a part of its own.

Analogously to POMDPs, each belief state in I-POMDP has an associated value reflecting the maximum payoff the agent can expect in this belief state:

$$\begin{aligned} U(\theta_i) = \max_{a_i \in A_i} \Bigg\{ &\sum_{is} ER_i(is, a_i) b_i(is) + \\ &\gamma \sum_{o_i \in \Omega_i} Pr(o_i|a_i, b_i) U(\langle SE_{\theta_i}(b_i, a_i, o_i), \widehat{\theta}_i \rangle) \Bigg\} \end{aligned} \tag{4}$$

where, $ER_i(is, a_i) = \sum_{a_j} R_i(is, a_i, a_j) Pr(a_j|m_j)$ (since $is = (s, \theta_j)$) (Gmytrasiewicz & Doshi 2005).

Agent i's optimal action, is an element of the set of optimal actions for the belief state, $OPT(\theta_i)$, defined as:

$$\begin{aligned} OPT(\theta_i) = \underset{a_i \in A_i}{argmax} \Bigg\{ &\sum_{is} ER_i(is, a_i) b_i(is) + \\ &\gamma \sum_{o_i \in \Omega_i} Pr(o_i|a_i, b_i) U(\langle SE_{\theta_i}(b_i, a_i, o_i), \widehat{\theta}_i \rangle) \Bigg\} \end{aligned} \tag{5}$$

Interactive Dynamic Influence Diagrams

Interactive Dynamic Influence Diagrams are a generalization of Dynamic Influence Diagrams to multi-agent settings and computational representations for I-POMDPs. They compute finite look-ahead approximations for I-POMDPs. I-DIDs contain some elements not present in classical dynamic influence diagrams (see Figure 2): hexagonal nodes, M_j, are called modeling nodes, dotted links are the belief update links, and double links, called policy links. Additionally, there is a chance node, A_j, representing the actions of the other agent.[2]

[2]For simplicity we assume only two interacting agents.

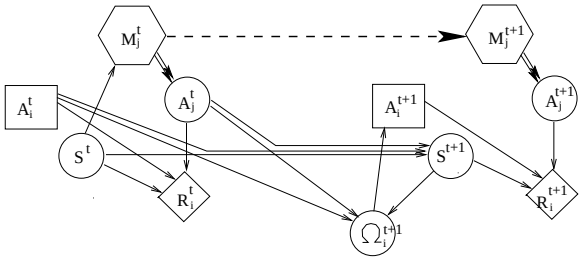

Figure 2: An Interactive Dynamic Influence Diagram

The chance node A_j contains i's belief about j's action. The values of the node M_j are the possible models of j, which themselves are I-DIDs. We assume that the set of possible models is finite.[3] Node S, representing the set of physical states, and M_j together represent the interactive state space, IS_i. The link between S_i and M_j captures the possible dependence between physical states and j's models.

The policy link between M_j and A_j represents the dependence between j's models and j's behavior, i.e., the $Pr(a_j^{t-1}|\theta_j^{t-1})$ in Eq 3. This is not a classical link, however, because the probability distribution over actions are obtained based on solutions (i.e., optimal actions) for each model in M_j.

The belief update link in I-DIDs connects the M_j nodes over time. It implements the τ_{θ_j} function in Eq. 3, representing the change in j's belief after its own observation and action. It should be clear why this is not a classical link – it involves computation nested at the level of j's models.

In (Gmytrasiewicz & Doshi 2005) two simplifying assumptions are made. According to the Model Nonmanipulability assumption agent i's actions cannot directly change agent j's model. Therefore, there is no direct link between A_i and M_j. According to the Model Nonobservability assumption, agent i cannot directly observe agent j's model. Hence, there is no direct link between M_j and Ω_i in Figure 2. However, agent i may be capable of receiving signals which are informative about the actions of agent j and about its models.

Formally, an interactive static influence diagram is a tuple $I - SID_i = (N_i, E_i, P_i, F_i)$. N_i is the set of nodes consisting of five types of nodes: random nodes, decision nodes, utility nodes, modeling nodes, and behavior nodes. There is one modeling node and one behavior node for each of the agents that agent i interacts with. Each node, $n \in N_i$, has a set of possible values, V_n. The values of modeling nodes are models of other agents. These can be either $I - SID$s, SIDs, or probability distributions over the other agents' actions. The values of the behavior nodes are the possible actions of the other agents.

E_i is the set of edges between the nodes. Given the edges, E_i, the function $parents(n)$ which returns the set of parent nodes of node n. For each random node, n, P_i is a conditional probability $Pr(n|parents(n))$. Each behavior

[3]It is up to the implementation to manage the space of models, which is potentially very large, efficiently. We describe our implementation further below.

node may have only one parent, namely the corresponding modeling node. The links between modeling nodes and the corresponding behavior nodes are called policy links. For each behavior node, A_j, its policy link specifies a conditional probability $Pr(A_j|M_j)$ describing likelihoods of j's actions given its model.[4]

F is a set $\{\mu_u : parents(u) \rightarrow R\}$ for each utility node u where R is the set of rational numbers.

An interactive dynamic influence diagram is a tuple $I - DID_i = (I - SID_i, h_i, T_i, P_i^D)$. h_i is the horizon to which $I - SID_i$ will be extended. T_i is the set of temporal edges so that $parents(n)$ include also parents in the previous time slice. For random nodes P_i^D is the dynamic extensio of P_i in $I - SID_i$ by including all of the nodes' parents. The values of the modeling nodes are now also allowed to include I-DIDs and DIDs. The modeling nodes are connected to each other across time via a special temporal link called a belief update link for agent j. It specifies the probability $Pr(M_j^t|M_j^{t+1})$ which is τ_{θ_j} defined above.

Solving an I-DID

Solution of an I-DID proceeds analogously to solution of a DID. This involves building a reachability tree containing the agent's beliefs for various sequences of its actions and observations. In case of I-DIDs the beliefs include ones over the models of the other agent. Using the example in Figure 2, the models of j contained in node M_j^t are solved and impart the probability distribution to node A_j^t, which corresponds to computing the $Pr(a_j^t|\theta_j^t)$ (this involves computing $OPT(\theta_j)$.) The policy link then becomes a conventional link between chance nodes (the node M_j^t values are now policies which are optimal for agent j.)

Now, given i's observation at time $t+1$, the probability distributions residing in nodes A_j^t and M_j^t are corrected. The corrected distribution over A_j^t can now be used to compute the corrected distribution over physical state, S^{t+1}. Also, given the distribution over A_j^t and i's action at time t, the probabilities of various observations available to j in each of its alternative models at time t can be computed.

Next, i simulates j's belief update for each of its models, and its action and observation pair. This implements the τ_{θ_j} function in Eq. 3 and updates the distribution in the M_j^{t+1} node. This completes the update of i's beliefs over one time step (i.e., implementing the τ_{θ_j} function), and constructs one branch of the reachability tree in I-POMDPs.

Given the above, the recursive character of the solution process becomes transparent: Updating i's own beliefs involves computing solutions to i's models of j, and performing j's belief update, τ_{θ_j}. The recursion terminates at level not deeper than l, given the finitely nested *I-POMDP*$_{i,l}$ of agent i.

Our implementation of I-DIDs and their solutions uses the Netica package (Corporation 1996). Each model is specified within a DNE file. The recursion of nested models is implemented by files containing probability distributions over pointers to other DNE files which specify the models of other agents.

[4]This corresponds to $Pr(a_j^{t-1}|\theta_j^{t-1})$ in the definition of agent i's I-POMDP.

I-DID Solution of the Multi-Agent Persistent Tiger Game

The single agent tiger game was first introduced in (Kaelbling, Littman, & Cassandra 1998). In this game, an agent is confronted with two doors. Behind one is a pot of gold (10 point reward for finding it) and behind the other is a tiger that attacks the agent (100 point penalty). During each time step, the agent may open either door or listen for the tiger's growl (listening incurs a 1 point penalty). When listening, the agent hears growls which indicate the location of the tiger with some reliability. In the single agent game, an agent updates its beliefs about the location of the tiger using the growls until it becomes sufficiently certain about tiger's location to risk opening a door.

In the multi-agent persistent tiger game variation, first, there are two agents facing the doors, and second, the tiger's location is reset with the probability of only 0.05 after any door is opened. Agents also receive richer observations. Their observations come from the set $\{GL, GR\}$ x $\{CL, S, CR\}$. The elements of $\{GL, GR\}$ represent tiger's growl from left and right door, respectively. The signals in the second set represent door creak from the left, silence, and door creak coming from the right. The creaks are informative of the action taken by the other agent. Thus, agents can infer the other's action from the creak signal, and use this as an additional source of information about the location of the tiger. Of course, a model of the other agent is needed to do this.

Agents may have a variety of different reward functions. Let us consider two types of reward functions: Friend and Enemy. An agent with the Friend reward receives the sum of the single agent reward structure (e.g. 10 + -1 = 9 if the first agent opens the door with the gold and the second listens). An agent with the Enemy reward receives the difference of the single agent's rewards (e.g. -100 - 10 = -110 if the first agent opens the tiger door and the second agent opens the door with the gold). When either door is opened, the tiger is likely to switch locations with 5% probability. If no door is opened, the tiger will stay put.

In this simple game, I-DID solutions elucidate many interesting interactive phenomena. Below, we examine the optimal behavior of agent i which begins the game with 99% certainly that the tiger is behind the right door, a reward function of Enemy, and 85% reliability of both creaks and growls.

For simplicity, let us assume that agent i has only one model of agent j. According to this model, first, j has no information about the tiger's location. Second, j hears creaks with high reliability (95% accuracy), but its hearing of the tiger's growls is uninformative (uniform likelihood of growls). Third, j has the Friend reward function. And finally, j has two models of i which are both almost certain (99%) of the Tiger's correct location. These models have the Enemy reward function and the same O function as agent i. These models are nesting level 0, so our recursion ends here.

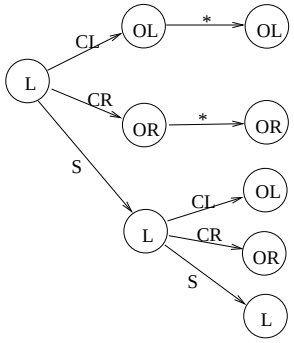

Figure 3: The policy of agent i's model of agent j.

Figure 4: The top-level policy of agent i.

We examine this situation for a game of three time horizons. To solve agent i's I-DID, we must first solve its model of j, which requires us to solve the two models j has of i which are of nesting level 0. The solutions to the I-DIDs of these agents are to open the door they believe hides the gold for all three time horizons. This is because they begin the game so certain of the tiger's location and find it so rewarding to get the gold that the value of acting with more information (gained from listening) does not increase their expected utility because it is "wasting a turn". Even though these model's are enemies of agent j, they do not model j's decision making process. Instead they assume uniform likelihood over agent j's action.

This solution is now incorporated into the model of agent j. The policy of this agent is described in the Figure 3.

Since agent j learns nothing from growls it needs to rely entirely on the actions of agent i to learn about the location of the tiger. We can see this behavior which is appropriately called "Follow the Leader." Agent j listens to the action of agent i, whom agent j believes is going to open the door hiding the gold in each time horizon. After it hears a creak indicating agent i's action, it twice opens the doors hiding the gold.

This policy is now incorporated into the top level I-DID. Agent i's best response to j's policy in Figure 3 is its own policy of $OpenRight$, $OpenLeft$, and $Listen$ no matter its observations. Note that the first action in this sequence opens the door which i believes is likely to hide the tiger! The reason is that, given j's "Follow the Leader" policy, it pays off for i to deceive agent j. The cost of this deception is rewarded in the subsequent time step by agent i's Enemy reward function. The deception in this example is possible because agent j's nesting leaves it to adopt a vulnerable "Follow the Leader" policy. Agent i does not open any door at time 3 due to mounting uncertainties: Agent j has imperfect sensors. If agent j did not hear CR after the first time step, then the agent would not have fallen for the deception. Similarly, with all these door openings, agent i is less certain where the tiger is located. The L action is a cautious (and optimal) action due to these reasons. Even still, if agent j does OR which would be ideal, agent i still benefits because it is rewarded when agent j is attacked by the tiger.

Conclusion

I-DIDs are a powerful tool for deliberative agents acting in partially observable, non-deterministic, stochastic multi-agent environments. I-DIDs act as a computational representation of I-POMDPs are offer all the benefits traditionally associated with influence diagrams including the ability to make statistical queries, compact representation of probability distributions, and expressing the conditional elements of a network in a clear, graphical manner.

For even simple examples, I-POMDPs can be very computationally demanding. Thus, a good approximation technique is very helpful for agents with limited memory or deliberating time. I-DIDs can function as a useful finite look-ahead approximation for I-POMDPs. Depending on available resources, the look-ahead of an I-DID may be extended or contracted, trading off optimality for computational demands.

In our future work we will explore approximate techniques for evaluating I-DIDs, and we will apply them to analyze other interactive decision making problems.

Future Work

Chess has long been a game of interest to the AI community for many reasons. We are particularly interested in a variant of chess known as Kreigspiel (or invisible chess). In this game, players do not see the positions of the other player's pieces. A referee announces the results of each agent's modes (invalid move, capture, check, etc). Much of the existing work in Kreigspiel assumes that one's opponent behaves randomly. Our approach includes modeling the decision making process of the adversary. We will show interactive phenomena as well as sophisticated strategies developed from deeper modeling in Kreigspiel.

There are many social behaviors which arise in I-DIDs. Of particular interest is the phenomenon of deception. The example in this paper, as well as most work on deception demonstrate its existence when the deceived agent fails to assign non-zero probability to a model of the other agent consistent with the true model of the deceiving agent (Jehiel 2006). We will investigate the value of planning to deceive other agents as an interesting facet of an agent's decision making process. In addition, a good methodology for predicting if one has been deceived would give an agent strong motivation to revise its belief over models of the other agent.

We are also exploring several approximation techniques for managing the set of models an agent considers with non-zero probability. Two of the most promising approaches are particle filtering (Doshi & Gmytrasiewicz 2005) and quantal response.

References

Aumann, R. J., and Heifetz, A. 2002. Incomplete information. In Aumann, R., and Hart, S., eds., *Handbook of Game*

Theory with Economic Applications, Volume III, Chapter 43. Elsevier.

Battigalli, P., and Bonano, G. 1998. Recent results on belief, knowledge and the epistemic foundations of game theory. Technical Report 98-14, University of California, Davis - Department of Economics.

Battigalli, P., and Siniscalchi, M. 1999. Hierarchies of conditional beliefs and interactive epistemology in dynamic games. *Journal of Economic Theory* (88):188–230.

Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research* 27(4):819–840.

Boutilier, C.; Dean, T.; and Hanks, S. 1999. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial intelligence Research* 11:1–94.

Boutilier, C. 1999. Sequential optimality and coordination in multiagent systems. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 478–485.

Chang, Y.-H., and Kaelbling, L. P. 2001. Playing is believing: The role of beliefs in multi-agent learning. In *Advances in Neural Information Procsesing Systems*.

Corporation, N. S. 1996. Netica: User's guide. Technical report, Norsys Software Corporation, 2315 DUnbar St., Vancouver, BC, Canada, http://www.norsys.com.

Doshi, P., and Gmytrasiewicz, P. 2005. Approximating state estimation in multiagent settings using particle filters. In *Proceeding of AAMAS 2005*.

Emery-Montemerlo, R.; Gordon, G.; Schneider, J.; and Thrun, S. 2004. Approximate solutions for partially observable stochastic games with common payoffs. In *Proceedings of Autonomous Agents and Multi-Agent Systems*.

Gal, Y., and Pfeffer, A. 2003. A language for modeling agents' decision making processes in games. *Profeedings of the Second International join Conference on Autonomus Agents and Multi-Agent Systems*.

Gmytrasiewicz, P., and Doshi, P. 2005. A framework for sequential planning in multiagent settings. *Journal of Artificial Intelligence Research* 24:49–79. http://jair.org/contents/v24.html.

Hauskrecht, M. 2000. Value-function approximations for partially observable markov decision processes. *Journal of Artificial Intelligence Research* (13):33–94.

Howard, R. A., and Matheson, J. E. 1984. Influence diagrams. In Howard, R. A., and Matheson, J. E., eds., *Readings on Principles and Applications of Decision Analysis*. Strategic Decisions Group, Menlo Park, CA. 721–762.

Hu, J., and Wellman, M. P. 1998. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Fifteenth International Conference on Machine Learning*, 242–250.

Jehiel, D. E. . P. 2006. Towards a theory of deception. Technical report, UCLA Department of Economics.

Kadane, J. B., and Larkey, P. D. 1982. Subjective probability and the theory of games. *Management Science* 28(2):113–120.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(2):99–134.

Koller, D., and Milch, B. 2001. Multi-agent influence diagrams for representing and solving games. In *Seventeenth International Joint Conference on Artificial Intelligence*, 1027–1034.

Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning*.

Mertens, J.-F., and Zamir, S. 1985. Formulation of Bayesian analysis for games with incomplete information. *International Journal of Game Theory* 14:1–29.

Monahan, G. E. 1982. A survey of partially observable markov decision processes: Theory, models, and algorithms. *Management Science* 1–16.

Myerson, R. B. 1991. *Game Theory: Analysis of Conflict*. Harvard University Press.

Nair, R.; Pynadath, D.; Yokoo, M.; Tambe, M.; and Marsella, S. 2003. Taming decentralized pomdps: Towards efficient policy computation for multiagent settings. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*.

Rathnasabapathy, B.; Doshi, P.; and Gmytrasiewicz, P. J. 2006. Exact solutions of interactive pomdps using behavioral equivalence. In *AAMAS 2006*, in press.