Proceedings of the Twenty-Second Innovative Applications of Artificial Intelligence Conference (IAAI-10)

Towards Applying Interactive POMDPs to Real-World Adversary Modeling

Brenda Ng and **Carol Meyers** and **Kofi Boakye** and **John Nitao**
Lawrence Livermore National Laboratory
7000 East Avenue
Livermore, CA 94550

Abstract

We examine the suitability of using decision processes to model real-world systems of intelligent adversaries. Decision processes have long been used to study cooperative multi-agent interactions, but their practical applicability to adversarial problems has received minimal study. We address the pros and cons of applying sequential decision-making in this area, using the crime of money laundering as a specific example. Motivated by case studies, we abstract out a model of the money laundering process, using the framework of interactive partially observable Markov decision processes (I-POMDPs). We discuss why this framework is well suited for modeling adversarial interactions. Particle filtering and value iteration are used to solve the model, with the application of different pruning and look-ahead strategies to assess the tradeoffs between solution quality and algorithmic run time. Our results show that there is a large gap in the level of realism that can currently be achieved by such decision models, due to computational demands that limit the size of problems that can be solved. While these results represent solutions to a simplified model of money laundering, they illustrate nonetheless the kinds of agent interactions that cannot be captured by standard approaches such as anomaly detection. This implies that I-POMDP methods may be valuable in the future, when algorithmic capabilities have further evolved.

1 Introduction

Techniques developed in the artificial intelligence community are currently applied to solve a wide range of real problems in crime and counterterrorism, particularly including the methods of anomaly detection (Singh and Silakari 2009) and link analysis (Senator et al. 1995; Chen et al. 2003). These methods typically employ static reasoning in their models of offender behavior, and may or may not explicitly consider the criminals' intent. While extremely valuable, these techniques often neglect the *dynamic* nature of adversarial interactions, whereby the strategy and tactics used by criminals may change in response to actions by law enforcement. In the worst case, such models can lead to an incorrect evaluation of the efficacy of potential countermeasures, by not capturing the phenomenon of adversarial response.

We seek to explore the feasibility of applying *sequential decision-making* methods for the real-world modeling of and response to adversarial behavior. This direction is primarily motivated by our work at a national laboratory, where there is a great interest in the application of novel methods to the study of crime and counterterrorism. In sequential decision-making methods, adversarial decision-making strategies are explicitly modeled, and the interactions between criminals and law enforcement is captured as a multi-agent stochastic decision process. This direction departs from that of a Nash equilibrium, which is designed for the game-theoretic analysis of at-equilibrium environments. Nash equilibria are not always suitable for the stochastic control of agents in a dynamic environment (Russell and Norvig 2003), since such equilibria may not be able to advise which action to select in situations that are off equilibria or with multiple equilibria.

The most extensively studied sequential decision-making framework is that of *decision processes*, which have accumulated a large body of literature since their introduction in the 1970's (Monahan 1982). The simplest of such models is the Markov decision process (MDP), in which a single agent with full knowledge of its environment attempts to optimize a discrete sequence of actions to maximize expected rewards. MDP models have been used to study financial and communications networks (Feinberg and Shwartz 2001). A generalization of the MDP is the Partially Observable Markov Decision Process (POMDP), in which a single agent does not know perfectly the state of the environment, and must infer the state distribution through noisy observations. Solution algorithms for POMDPs have been studied extensively (Kaelbling, Littman, and Cassandra 1998), and POMDPs have been applied to real-world problems including assistance of patients with dementia (Hoey et al. 2007). While POMDPs have also been applied to adversarial problems including simplified poker (Oliehoek 2005), the opposing agent is usually treated as noise, which is not a faithful assumption in settings with dynamic, deliberate adversaries.

Adversarial problems with intentional agents are by nature not one-sided, suggesting the need for a *multi-agent* framework. Research in multi-agent processes is more recent, and different approaches have been proposed to mitigate the computational burden arising in the extension to multiple agents. Within this context, decentralized POMDPs (DEC-POMDPs) (Bernstein, Zilberstein, and Immerman 2000) have been used to model interactions of multi-agent collaborative teams. While algorithms have been developed for solving such problems (Seuken and Zilberstein 2008), DEC-POMDPs are not a suitable framework for adversarial agents because of their assumption of common rewards. Partially observable stochastic games (POSGs) (Hansen, Bernstein, and Zilberstein 2004) avoid this issue by allowing for

different agent rewards, but exact POSG algorithms are limited to very small problems (Guo and Lesser 2006), and approximate POSG algorithms have only been developed for common rewards (Emery-Montemerlo et al. 2004).

After examining the suitability of numerous multi-agent models, we believe that the structure of interactive POMDPs (I-POMDPs) (Doshi and Gmytrasiewicz 2009) is best able to support adversary modeling. An I-POMDP is a multi-agent extension of the POMDP framework, in which each agent maintains beliefs about both the physical states of the world and the decision process models of the other agents. Different agent rewards are allowed, and the incorporation of nested intent into agent beliefs allows the modeling of agents that "game" against each other. Most importantly, there are algorithms designed to solve I-POMDPs approximately, using both particle filtering and value iteration techniques (Doshi and Gmytrasiewicz 2009), that do not impose the restriction of common rewards as in the DEC-POMDP algorithms. This combination of opponent reasoning and established algorithms led to our decision to use the I-POMDP framework.

In what follows, we describe our efforts to apply the I-POMDP model to solve a sample adversarial problem motivated by the money laundering community. We describe our money laundering model in Section 3 , discuss implementation issues in Section 4 , and present empirical results in Section 5 As our work represents one of the first attempts to apply sequential decision-making to problems with dynamic adversaries, we also address the challenges that arose in the implementation of this framework, both in terms of algorithmic improvements and model simplifications. We find that as a whole, the computational demands of such methods can be extreme, but the advantages of being able to model nested intent as part of the agent model and to incorporate this into the computed policies, make I-POMDPs an attractive approach for adversarial modeling and response.

2 Preliminaries

2.1 Partially Observable Markov Decision Processes (POMDPs)

A POMDP describes a sequential decision-making process of a *single agent* with partial observability of its environment. A POMDP is specified by $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{Z}, \mathcal{O} \rangle$, where:

- $\mathcal{S}$ is the finite set of physical states in the environment;
- $\mathcal{A}$ is the finite set of possible actions;
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ is the state transition function, $\mathcal{T}(s,a,s') = P(s'|s,a)$, that defines the probability of ending in state s' if action a is taken in state s;
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, $\mathcal{R}(s,a)$, that defines the reward of taking action a in state s;
- $\mathcal{Z}$ is the finite set of possible observations; and
- $\mathcal{O} : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \rightarrow [0,1]$ is the observation function, $\mathcal{O}(s',a,z) = P(z|a,s')$, that defines the probability of observing z if action a is taken, resulting in state s'.

At each time step, the agent updates its *belief state* $b^t(s')$, which defines the probability of being in each state s' given its history of actions and observations. Formally,

$$b^t(s') = \frac{\mathcal{O}(s', a^{t-1}, z^t) \sum_{s \in \mathcal{S}} \mathcal{T}(s, a^{t-1}, s') b^{t-1}(s)}{P(z^t | b^{t-1}, a^{t-1})},$$

where $P(z|b,a) = \sum_{s' \in \mathcal{S}} \mathcal{O}(s',a,z) \sum_{s \in \mathcal{S}} \mathcal{T}(s,a,s') b(s)$. To solve a POMDP is to find an *optimal policy*, which for every achievable belief state produces an action maximizing the agent's expected reward.

There are two methods for solving POMDPs: value and policy iteration. In value iteration, an optimal solution is derived by iteratively solving for the optimal expected reward for each state, using a form of Bellman's equations. Conversely, in policy iteration, an improving set of policies is generated that iteratively converges to the optimal policy, by evaluating expected rewards at each step and finding an improving policy if one exists. POMDP algorithms that are based on variants of value iteration include Sondik's One-Pass (Sondik 1971) and the Witness algorithm (Littman 1996), while algorithms based on policy iteration include Sondik's policy iteration (Sondik 1971) and Bounded Policy Iteration (Poupart and Boutilier 2004).

2.2 Interactive POMDPs (I-POMDPs)

I-POMDPs are a generalization of POMDPs to multiple agents that can have different (and possibly conflicting) objectives (Doshi and Gmytrasiewicz 2009). In an I-POMDP, each agent augments its current state with the set of models of other agents' behaviors, to form an *interactive* state.

For the case of two intentional agents (i and j), agent i's I-POMDP with nesting level l is specified by the tuple $\langle \mathcal{IS}_{i,l}, \mathcal{A}, \mathcal{T}_i, \mathcal{R}_i, \mathcal{Z}_i, \mathcal{O}_i \rangle$:

- $\mathcal{IS}_{i,l} = \mathcal{S} \times \Theta_{j,l-1}$ is the finite set of i's interactive states, with $\mathcal{IS}_{i,0} = \mathcal{S}$; $\Theta_{j,l-1}$ is the set of intentional models of j, where each intentional model $\theta_{j,l-1} \in \Theta_{j,l-1}$ consists of j's belief $b_{j,l-1}$ and frame $\hat{\theta}_j = \langle \mathcal{A}, \mathcal{T}_j, \mathcal{R}_j, \mathcal{Z}_j, \mathcal{O}_j \rangle$;
- $\mathcal{A} = \mathcal{A}_i \times \mathcal{A}_j$ is the finite set of joint actions;
- $\mathcal{T}_i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ is the state transition function;
- $\mathcal{R}_i : \mathcal{IS}_i \times \mathcal{A} \rightarrow \mathbb{R}$ is i's reward function;
- $\mathcal{Z}_i$ is the finite set of i's observations; and
- $\mathcal{O}_i : \mathcal{S} \times \mathcal{A} \times \mathcal{Z}_i \rightarrow [0,1]$ is i's observation function.

At each time step, agent i maintains a belief state $b_i^t(is^t) = \beta \sum_{is^{t-1}} b_i^{t-1}(is^{t-1}) \sum_{a_j^{t-1}} P(a_j^{t-1} | \theta_{j,l-1}^{t-1}) \mathcal{O}_i(s^t, a^{t-1}, z_i^t) \cdot \mathcal{T}_i(s^{t-1}, a^{t-1}, s^t) \sum_{z_j^t} P(b_{j,l-1}^t | b_{j,l-1}^{t-1}, a_j^{t-1}, z_j^t) \mathcal{O}_j(s^t, a^{t-1}, z_j^t)$, where β is a normalizing factor and $P(a_j^{t-1} | \theta_{j,l-1}^{t-1})$ the probability a_j^{t-1} is Bayes rational for an agent modeled by $\theta_{j,l-1}^{t-1}$.

The belief update procedure is more complicated than in POMDPs, because the physical state transitions depend on both agents' actions. To predict the next physical state, agent i must update its beliefs about agent j's behavior based on its anticipation of what j observes and how j updates its belief. This can lead to a potentially infinite nesting of beliefs, for which it is practical to impose a finite level of nesting l.

Methods used in solving I-POMDPs include equilibrium-based methods (Doshi and Gmytrasiewicz 2006), dynamic influence diagrams (Doshi, Zeng, and Chen 2009), and generalized point-based value iteration (Doshi and Perez 2008). The most thoroughly documented method has been the use of an *interactive particle filter* (I-PF) (Doshi and Gmytrasiewicz 2009), in conjunction with reachability tree sampling, which led us to select this technique for our problem.

2.3 Money Laundering

Money laundering is a crime often involving a complex series of transactions and financial institutions from different jurisdictions. This crime is also very pervasive: estimates suggest that at least 2% of the global gross domestic product is comprised of money laundering activities (Buchanan 2004). Law enforcement has described this laundering as a three-step process (United States Treasury 2002):

Placement: Physical 'dirty' currency enters the financial system. This is the most vulnerable point in the process, as it involves the conversion of large amounts of cash. Common targets of placement include domestic bank accounts, insurance products, and securities.

Layering: Creation of a web of financial transactions to obscure the source of the money. Types of layering include transfers to offshore accounts, shell companies, and trusts. By making numerous transactions in different jurisdictions, the paper trail becomes hard to follow.

Integration: Reintroduction of the laundered money to the mainstream economy. This is done by mingling laundered money with funds from legitimate businesses, including real estate, loans, or casinos. This can be hard to detect unless a paper trail has been formed.

In recent years, significant legislation has been introduced in the US to detect money laundering activity (United States Treasury 2002). The Bank Secrecy Act (BSA) requires banks and other institutions to file Suspicious Activity Reports (SARs) on transactions that they suspect might involve money derived from illegal activity. In addition, any transaction of more than $10,000 automatically generates a Currency Transaction Report (CTR). These reporting requirements have been invaluable in helping trace money laundering activities, though the question still remains of how best to sort through massive amounts of BSA report data.

There has been some previous work on using data mining to find evidence of money laundering activity (Office of Technology Assessment 1995). This work has been primarily in the areas of clustering and link analysis (Zhang, Salerno, and Yu 2003), and such algorithms have been employed in tools used by the US Financial Crimes Enforcement Network (Goldberg and Senator 1998). To our knowledge, sequential decision-making techniques are not currently being used to model or assess money laundering. Thus, this work represents a first step in a novel direction.

3 A Sample Adversarial Problem: an I-POMDP of Money Laundering

In this section, we describe the sample money laundering problem used in our feasibility study. We begin by noting that this problem has had to be downscaled dramatically from our original designs to ensure tractability. Throughout this section and Section 5 , we discuss the modeling tradeoffs that had to be made to ensure solvability of the problem.

3.1 Agents and Joint States

The agents in our model are the Red Team (money launderers) and the Blue Team (law enforcement). The Red Team's goal is to evade capture while moving assets from a 'dirty' pot to a 'clean' pot in a financial network, while the Blue Team's goal is to find and confiscate assets of the Red Team.

Joint physical states are defined as $s = \{redMoneyLoc, blueSensorLoc\}$. The Red Team states represent accounts where money could be held, and the Blue Team states represent potential sensor locations. Specifically, the Red Team has 11 possible states: {dirty pot, bank accounts, insurance, securities, offshore accounts, shell companies, trusts, corporate loans, casino accounts, real estate, clean pot}. The Blue Team has 9 sensor locations: {no sensors, bank accounts, insurance, securities, shell companies, trusts, corporate loans, casino accounts, real estate}. There are thus 99 joint states.

This model presupposes that each team can only occupy one state at a given time. This was a major concession from our early model, in which the Red and Blue Team's assets could be spread among states. Unfortunately, such flexibility caused the state space to increase exponentially, and current I-POMDP algorithms simply could not solve it.

3.2 Actions and Observations

Analogous to real-world money laundering, the Red Team has four primary actions: {Placement, Layering, Integration, Listening}. The Placement action transitions the Red Team from the dirty pot to the 'placement states' of bank accounts, insurance, or securities, with equal probability. If Red's money is in a placement state, then the Layering action can move the assets among the placement states or into the 'layering states', which are offshore accounts, shell companies, and trusts. If Red's money is in a layering state, then the Layering action can move the assets among the layering states, or into the 'integration states' of corporate loans, casino accounts, and real estate. Finally, the Integration action transitions the integration states to the Red Team's clean pot with a fixed probability. The Listening action enables the Red Team to gather noisy intelligence on Blue's location.

Most of the Blue Team's actions relate to sensor placements; in particular, there is a corresponding action for placing sensors at each of the 8 locations listed in the Blue Team's states. The Blue Team also has a special action, Confiscation, in which it attempts to seize the Red Team's assets if it believes that both teams occupy the same physical state.

In terms of observations, the Red Team can obtain evidence of the Blue Team's presence at a given number of locations (banks, shell companies, and casino accounts) by executing the Listening action from that state. The Blue Team observes the Red Team via reports (such as SARs) from most locations (except dirty pot, clean pot, offshore accounts, and shell companies). The Blue Team can also receive observations from its sensors to supplement information gained from the reports.

Currently, the transition and observation probabilities are populated by the 'best guess' ideas of our team. Ideally, these probabilities might be derived by Bayesian reinforcement learning techniques. As part of future work, it might be possible to adopt ideas from Bayes-adaptive POMDPs (Ross, Chaib-draa, and Pineau 2007) to learn these probabilities, but this is beyond the scope of our current study.

3.3 Rewards and Termination Conditions

The reward function formalizes the game progress and termination conditions. Specifically, there is a reward (penalty) of 100 to the Red Team (Blue Team) for evading capture and reaching the clean pot; conversely, there is a penalty (reward) of 100 to the Blue Team (Red Team) for apprehending

the money launderer. In addition, the Listening action by the Red Team incurs a penalty of 20, provided that the Red team is not caught during that turn. The Blue Team experiences a penalty of 25 for a mistaken confiscation (occurring when Red's assets and Blue's sensors are at different locations, and the Confiscation action is taken). All other actions have a penalty of 10, to ensure the game does not stall. We experimented with a range of values for rewards/penalties while maintaining their ordinality (i.e., the stall penalty being less than the misconfiscation penalty, which in turn is less than the misapprehension penalty, etc.) and have found that the results were not sensitive to the specific values chosen.

In a more realistic game, a money launderer might be able to lose only part of their money, and law enforcement might apprehend only part of a criminal ring. These features were also sacrificed to ensure a manageable number of states.

3.4 Comparison with Benchmark Problems

Table 1 compares the size of our problem with three benchmark problems documented and solved using I-POMDPs (see (Doshi and Gmytrasiewicz 2009) for details about the benchmark problems).

Table 1: Size Comparison of Problems Solved by I-POMDPs

Problem	States $\lvert\mathcal{S}\rvert$	Actions $\lvert\mathcal{A}_i\rvert \times \lvert\mathcal{A}_j\rvert$	Observations $\lvert\mathcal{Z}_i\rvert \times \lvert\mathcal{Z}_j\rvert$	Problem size (product)
Tiger	2	3×3	6×6	648
Machine	3	4×4	2×2	192
UAV	36	5×5	3×3	8100
ML (ours)	99	4×9	4×11	156,816

Thus, even after downscaling, the size of our money laundering problem is much larger (by nearly 20 times) than anything previously solved by I-POMDPs. It is our hope that progress on this problem will shed light on bridging the gap between the theory and practice of applying I-POMDPs.

4 I-POMDP Algorithms

4.1 Interactive Particle Filter (I-PF)

Introduced by (Doshi and Gmytrasiewicz 2009), the interactive particle filter extends the standard particle filter to multi-agent state estimation within an I-POMDP. Analogous to the stardard particle filter, I-PF alternates between *importance sampling* and *selection*, inheriting the same convergence properties. In the following discussion, we use k to represent either agent i or j, and $-k$ for the other agent.

The I-PF for agent k starts with a set of N samples or *particles*, $\tilde{b}_{k,l}^{t-1}$, that approximates agent k's belief state. These particles are generated by recursively sampling from the nested beliefs that comprise the belief state.

The algorithm proceeds by propagating each particle forward in time. This operation requires an estimate of agent $-k$'s action, since state transitions are dependent on joint actions. To infer $-k$'s action, its policy must be derived from its belief state. This, in turn, requires a recursive invocation of the particle filter for each of the nesting levels (Figure 1). Recursion terminates at level 1, where agent k uses a POMDP belief update to infer $-k$'s level-0 belief.

The propagation step generates a total of $\lvert\mathcal{Z}_{-k}\rvert N$ particles, consisting of N particles for each possible observation of the other agent. Once this step has been completed, the algorithm assigns an importance weight to each of the $\lvert\mathcal{Z}_{-k}\rvert N$ new particles. This weight is based on the probability of obtaining the associated observations given the interactive state of the particle and the actions of each agent. A set of N particles is then resampled according to these importance weights and returned as the output of the algorithm.

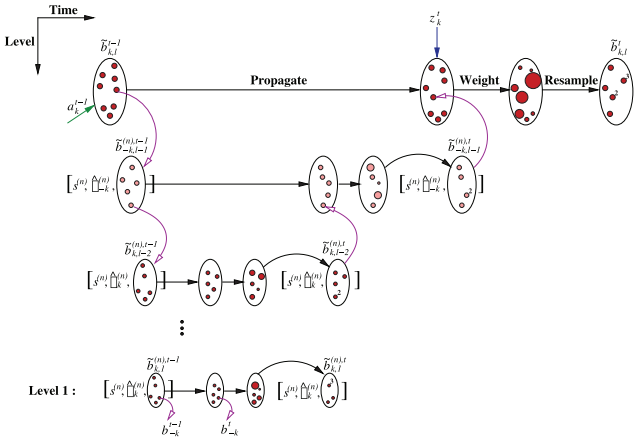

Figure 1: In I-PF, sampling is recursively performed for each level of beliefs. Agent k updates its beliefs based on its anticipation of the other agent's observations, actions, and model updates. This requires updating the other agent's beliefs, leading to particle filtering on its beliefs. Image adopted from (Doshi and Gmytrasiewicz 2009).

4.2 Reachability Tree Sampling

While the I-PF approach addresses the curse of dimensionality due to the complexity of the belief state, the curse of history can also be problematic, because the complexity of policies increases with the horizon length. Here, value iteration requires the construction of a reachability tree as part of its reachability analysis. With increasing horizons, this reachability tree grows exponentially to account for every possible sequence of actions and observations during the course of the decision process. To address this issue, (Doshi and Gmytrasiewicz 2009) proposed reachability tree sampling (RTS) as a way to reduce the branching factor of this tree. In RTS, one samples observations according to $z_i^t \sim P(\mathcal{Z}_i | a_i^{t-1}, \tilde{b}_{k,l}^{t-1})$ and builds a partial reachability tree based on the samples and the complete set of actions.

In our money laundering problem, the curse of history required approximations beyond the standard RTS to address a major computational bottleneck: the construction of the opposing agent's reachability tree. As mentioned previously, in order for agent k to behave optimally, it must anticipate what action $-k$ might take. Thus, in solving for k's optimal policy, it must also construct $-k$'s reachability tree and use it to find $-k$'s optimal action. As the size of this tree grows as $\mathcal{O}((\lvert\mathcal{A}_{-k}\rvert\lvert\mathcal{Z}_{-k}\rvert)^l)$, it becomes large very quickly. In our model (Table 1), this presented a major computational hurdle.

To solve our money laundering model, it was necessary to prune not only the agent's reachability tree, but also the opposing agent's reachability tree. To do so, we introduced two approximations in particular: (1) RTS on $-k$'s reachability tree, thus reducing its tree's breadth; and (2) limited look-ahead for $-k$, thus reducing its tree's depth. These are reasonable approximations under a resource-constrained environment. For the first strategy, our modified RTS algo-

rithm uses $N_{\mathcal{Z}_k}$ observation samples to construct k's reachability tree and $N_{\mathcal{Z}_{-k}}$ observation samples to construct $-k$'s reachability tree. For the second strategy, our modified algorithm truncates the construction of $-k$'s reachability tree at a prespecified look-ahead depth.

5 Empirical Results

Up to this point, we have discussed ways to make the algorithms tractable. However, the feasibility of these algorithms is best reflected in the quality of the policies computed. Simulations provide a natural way to evaluate these policies. In what follows, we present simulation results based on the policies computed by I-PF along with our modified RTS.

Our simulations start with all particles in the ⟨Dirty-pot, No-sensor⟩ state and the level-0 opponent belief initialized with all its probability mass in that same state. Each agent computes a policy using the modified RTS, based on a planning horizon of h for itself and an opponent horizon o for its opinion of the opponent's horizon. The agents execute their computed policies over a number of turns based on their planning horizons, accruing a reward or penalty at each turn. Thereafter, each agent repeats its planning; the updated prior belief for this new round is obtained from the leaf node of the reachability tree corresponding to the actions and observations in the last round. Due to the sampling of observations in RTS, if this leaf node does not have an updated belief, the most recently updated node in the path from the root node to the leaf is used as an alternative. This process repeats until one agent reaches a termination condition. (Thus, the number of turns can vary between different simulations.)

For each agent k, there are five tunable parameters that can affect the quality of the computed policy: (1) the faithfulness of belief representation in terms of the number of particles N; (2) the horizon h in which planning is based (i.e., how many steps of look-ahead k uses in deciding its action); (3) the breadth of each horizon (i.e., how broad is the action/observation space at each depth of k's reachability tree), controlled by $N_{\mathcal{Z}_k}$; (4) the assumed planning horizon o of the opponent (i.e., k's assumption of $-k$'s look-ahead); and (5) the assumed breadth of the opponent (i.e., k's assumption of breadth in $-k$'s reachability tree), controlled by $N_{\mathcal{Z}_{-k}}$. The influence of these parameters on the computed policies has practical relevance to real-world applications, in terms of both the quality of policies produced and the sensitivity of each player to its assumptions about its opponent. As such, we seek to answer the following questions:

1. Given the asymmetry of our problem, which agent is at a more advantageous position under the same parameters?
2. What change in parameters, if any, can alter this position?
3. In a resource-bounded computational environment, which change gives more improvement to the policy: making the belief more accurate, or planning with a longer horizon?

These questions illustrate the merits of using a decision process approach to adversarial problems: the information obtained from this analysis easily allows for model refinement in a manner not possible in anomaly detection, for example.

5.1 Simulation Results

To answer our first question, a baseline experiment was performed. In this experiment, the parameters were the same for both agents and were as follows: $N = 10$, $h = 3$, $o = 2$, $N_{\mathcal{Z}_k} = 4$, $N_{\mathcal{Z}_{-k}} = 4$. In particular, both agents are assumed equal with respect to approximations in their belief states and reachability trees. The disparity in the planning horizons ($h > o$) represents a common assumption that each agent "thinks" it is superior to its adversary in terms of planning. In general, we were not as interested in the exact values of this baseline case as in the effects of deviating from *some* set of baseline parameters; as such, these choices were influenced in part by the need for reasonable execution time.

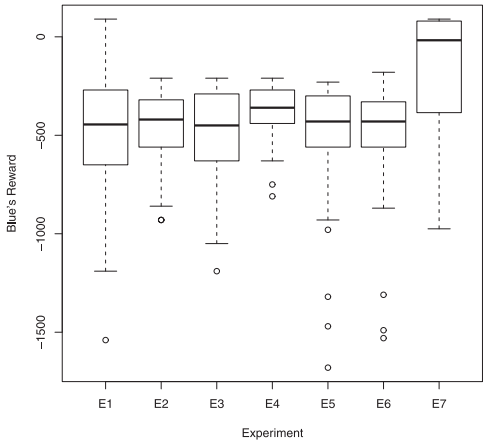

Figure 2: Blue Team Performance in Simulation Experiments

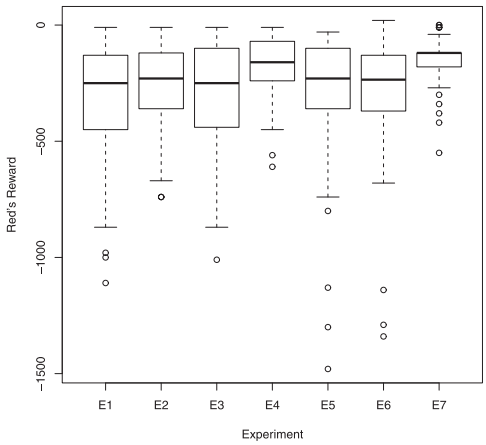

Figure 3: Red Team Performance in Simulation Experiments

For all experiments, results were based on 50 trials. Baseline results appear as the first boxplot (E1) in Figures 2-4. Figure 2 gives the Blue Team reward, Figure 3 the Red Team reward, and Figure 4 the number of turns simulated. Figures 2 and 3 show that on average, Red's reward exceeds Blue's, and thus Red has an advantage for these parameter values.

The next five boxplots (experiments E2-E7) in Figures 2-4 show results addressing the second question. In each experiment, one of the baseline parameters was varied to allocate more resources to the Blue Team. In E2, the Blue Team's number of particles N was increased to 25; in E3, the Red Team's planning horizon h was reduced to 2; in E4, the Red Team's opponent horizon o was reduced to 1; in E5, the Blue Team's $N_{\mathcal{Z}_k}$ was increased to 16; and in E6, the Blue Team's

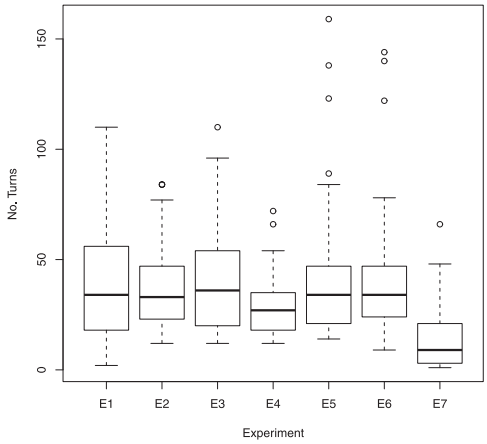

Figure 4: Number of Turns per Simulation

$N_{\mathcal{Z}_{-k}}$ was increased to 16. We observe that a noticeable difference occurs only in E4, and even here the difference is not sufficient to shift the advantage towards the Blue Team.

Such a shift in advantage does occur in the final experiment, E7. Here, the opponent horizon o for both Blue and Red is set to 1. This corresponds to the situation where both players are myopic in the assessment of their opponents, leading to an imbalance in the sophistication levels of the strategies produced. We can account for this anomalous shift by inspecting the particle distributions for the nodes in the reachability tree; our analysis reveals that the entropy of all nodes for the Blue Team's model is at or near zero in all experiments except E7, leading to the policy of repeated identical sensor placement in those cases. In such a situation, allocating resources towards improving the belief accuracy is of little benefit, providing insight to our third question above. By reducing Blue's opponent horizon, a policy involving the Confiscation action emerges, allowing Blue the opportunity to win. In addition, Blue's victories appear to be quick, as evidenced by the significantly reduced number of turns in such simulations (see Figure 4).

5.2 Computational Findings

We note that the computational demands were extreme in solving even our simplified model. Table 2 displays the run times for different numbers of particles and horizon lengths. In all runs, we assume $o = 3$, and $N_{\mathcal{Z}_k} = N_{\mathcal{Z}_{-k}} = 4$. The machine used is a Mac Pro with two 2.93GHz quad-core Intel Xeon processors and 16.0GB of RAM, running OS X.

Table 2: Run Times for Solving the I-POMDP, in Seconds.

	$h = 2$	$h = 3$	$h = 4$	$h = 5$
5 particles	31	930	16602	*
10 particles	64	1774	33968	*
50 particles	428	8998	*	*

Entries of the table indicated with a $*$ did not solve within a specified cutoff time of 24 hours. We observe that even for relatively small opponent horizon lengths and numbers of particles, the model can take an extremely long time to solve. This finding was the primary motivation behind the downscaling of our original money laundering problem.

6 Conclusions

Our work represents a first step towards the use of sequential decision-making methods in real adversarial settings. With the I-POMDP framework, we are able to model adversarial interactions at a level of fidelity not possible in traditional techniques used in crime and counterterrorism. In particular, traditional techniques typically treat the adversary as fixed and/or as noise, employing static methods such as clustering and outlier detection, to merely *identify* adversarial effects. In contrast, we capture the dynamic nature of these interactions as a multi-agent decision process in which strategies and tactics can change over time, and we are able to compute the best response given the evidence presented so far.

While there has been some work addressing dynamic adversaries via methods other than I-POMDPs, none has tackled this problem with as much "dynamicity" as this paper. (Dalvi et al. 2004) frames the problem of a dynamic spammer as a classification problem in which a learner is presented with a tampered distribution of instances where some spam instances are transformed into non-spam instances by word addition or synonym usage. However, this work captures only one-shot interaction between the spammer and the learner, and does not capture adaptivity adopted by the spammer and/or learner upon repeated interactions. (Tan and Cheng 2009) examined games where the environment can be decomposed into constituent POMDP and MDP components, where the POMDP corresponds to the adversary player's model and the MDP is the rest of the perfectly observable world, as in the game of tennis. This work presupposes that the POMDP component is much smaller than the MDP component, hence circumventing some of the computational overhead associated with POMDPs. However, while this type of convenient structure may manifest in the controlled environments of *virtual* games, it is rarely applicable in real-world domains. In sum, the applicability of these ideas are yet to be seen as feasible for modeling of real-world situations of the kind explored in this paper.

While we recognize that our model is far from the actual and complex process of money laundering, we find this exercise to be useful in illustrating the computational challenges that first need to be addressed before I-POMDPs can be deployed in real security-related, adversarial applications. Moreover, we note that our abstract model is sufficiently general to be applicable to other real-world problems beyond the domain of money laundering; it is a representative of a generic class of problems in adversarial search and pursuit, and thus our findings can be extended to this larger domain.

Our findings show that the computational bottleneck in solving I-POMDPs is in the construction of the agents' reachability trees. In this work, we pruned the reachability trees via sampling. Alternatively, we might be able to exploit the problem structure to streamline the construction of reachability trees via distributed processing. However, this would require a more thorough future investigation and its implementation might be problem-dependent. Thus, one direction for making I-POMDPs more widely applicable for adversarial modeling would be to mitigate the computational burden through algorithmic approximations (approach taken in this work) and/or exploitation of problem structure.

Another direction for future research would be to relax the assumption that the model is known a priori. As part of current ongoing work, we are extending the I-POMDP

framework to incorporate Bayesian reinforcement learning, such that the transition and observation models are inferred in conjunction with the state, in a sequential manner analogous to the framework of Bayes-Adaptive POMDPs (Ross, Chaib-draa, and Pineau 2007). This will be an important step in making I-POMDP applications more practical since it is unrealistic to assume complete and exact knowledge of the adversary's parameters, or that these parameters stay fixed and do not change over time. Incorporating reinforcement learning will make the task of I-POMDP modeling more faithful to the actual adversarial process as it happens.

Despite the current limitations of I-POMDP algorithms, we feel that the framework itself is still very promising because an I-POMDP can (1) model agents with both adversarial and cooperative objectives, and (2) incorporate nested intent as part of the agent's model. The aspect of nested intent is especially relevant in modeling human agents, who arguably employ nested strategies in adversarial situations naturally. While the technology of I-POMDPs has not yet evolved to address realistic-sized problems (involving hundreds or thousands of states, actions and/or observations), we are optimistic that, with future advances, I-POMDPs will eventually be applicable for real-world deployment.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory (LLNL) under Contract DE-AC52-07NA27344, and was supported by LLNL funding for the project *Towards Understanding Higher-Adaptive Systems* (09-LW-30). The authors also wish to thank Prashant Doshi for his helpful advice and starter source code.

References

Bernstein, D.; Zilberstein, S.; and Immerman, N. 2000. The complexity of decentralized control of Markov decision processes. In *Proceedings of the 16th UAI Conference*, 32–37. Stanford, CA: Morgan Kaufmann.

Buchanan, B. 2004. Money laundering: a global obstacle. *Research in International Business and Finance* 18:115–127.

Chen, H.; Zeng, D.; Atabakhsh, H.; Wyzga, W.; and Schroeder, J. 2003. Coplink: mining law enforcement data and knowledge. *Communications of the ACM* 46(1):28–34.

Dalvi, N.; Domingos, P.; Mausam; Sanghai, S.; and Verma, D. 2004. Adversarial classification. In *Proceedings of the 10th ACM SIGKDD Conference*, 99–108. Seattle, WA: ACM.

Doshi, P., and Gmytrasiewicz, P. 2006. On the difficulty of achieving equilibrium in interactive POMDPs. In *Proceedings of the 21st AAAI Conference*. Boston, MA: AAAI.

Doshi, P., and Gmytrasiewicz, P. 2009. Monte Carlo sampling methods for approximating Interactive POMDPs. *Journal of Artificial Intelligence Research* 34:297–337.

Doshi, P., and Perez, D. 2008. Generalized point-based value iteration for interactive POMDPs. In *Proceedings of the 23rd AAAI Conference*, 63–68. Chicago, IL: AAAI.

Doshi, P.; Zeng, Y.; and Chen, Q. 2009. Graphical models for interactive POMDPs: representations and solutions. *Autonomous Agents and Multi-Agent Systems* 18(3):376–416.

Emery-Montemerlo, R.; Gordon, G.; Schneider, J.; and Thrun, S. 2004. Approximate solutions for partially observable stochastic games with common payoffs. In *Proceedings of the 3rd AAMAS Conference*, 136–143. New York, NY: IEEE.

Feinberg, E., and Shwartz, A. 2001. *Handbook of Markov Decision Processes: Methods and Applications*. International Series in OR and Management Science. Springer.

Goldberg, H., and Senator, T. 1998. The FinCEN AI system: finding financial crimes in a large database of cash transactions. In *Agent Technology: Foundations, Applications, and Markets*. Springer-Verlag. Chapter 15.

Guo, A., and Lesser, V. 2006. Stochastic planning for weakly coupled distributed agents. In *Proceedings of the 5th AAMAS Conference*, 326–328. Hakodate, Japan: ACM.

Hansen, E.; Bernstein, D.; and Zilberstein, S. 2004. Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th AAAI Conference*, 709–715. San Jose, CA: AAAI.

Hoey, J.; von Bertoldi, A.; Poupart, P.; and Mihailidis, A. 2007. Assisting persons with dementia during handwashing using a partially observable Markov decision process. In *Proceedings of the 5th ICVS Conference*.

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

Littman, M. 1996. *Algorithms for Sequential Decision Making*. PhD Dissertation, Brown University.

Monahan, G. 1982. A survey of partially observable Markov decision processes: Theory, models, and algorithms. *Management Science* 28(1):1–16.

Office of Technology Assessment, U. C. 1995. *Information Technologies for the Control of Money Laundering*. OTA-ITC-630. US Government Printing Office.

Oliehoek, F. 2005. *Game theory and AI: a unified approach to poker games*. Masters Thesis, University of Amsterdam.

Poupart, P., and Boutilier, C. 2004. Bounded finite state controllers. In *Proceedings of the 17th NIPS Conference*, 823–830. Vancouver, BC, Canada: MIT Press.

Ross, S.; Chaib-draa, B.; and Pineau, J. 2007. Bayes-adaptive POMDPs. In *Proceedings of the 21st NIPS Conference*. Vancouver, British Columbia, Canada: MIT Press.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition.

Senator, T.; Goldberg, H.; Wooton, J.; Cottini, M.; Khan, A.; Klinger, C.; Llamas, W.; Marrone, M.; and Wong, R. 1995. The FinCEN artificial intelligence system: identifying potential money laundering from reports of large cash transactions. In *Proceedings of the 7th IAAI Conference*, 156–170. Montreal, Quebec: AAAI.

Seuken, S., and Zilberstein, S. 2008. Formal models and algorithms for decentralized decision making under uncertainty. *Autonomous Agents and Multi-Agent Systems* 17(2):1387–2532.

Singh, S., and Silakari, S. 2009. A survey of cyber attack detection systems. *International Journal of Computer Science and Network Security* 9(5):1–10.

Sondik, E. 1971. *The Optimal Control of Partially Observable Markov Decision Processes*. PhD Dissertation, Stanford University.

Tan, C., and Cheng, H. 2009. IMPLANT: An integrated MDP and POMDP learning agent for adaptive games. In *Proceedings of the 5th AIIDE Conference*, 94–99. Stanford, CA: AAAI.

United States Treasury. 2002. Money laundering: A banker's guide to avoiding problems. Technical report, Office of the Comptroller of the Currency, US Treasury.

Zhang, Z.; Salerno, J.; and Yu, P. 2003. Applying data mining in investigating money laundering crimes. In *Proceedings of the 9th ACM SIGKDD Conference*, 747–752. Washington, DC: ACM.