An Efficient Heuristic for Security Against Multiple Adversaries in Stackelberg Games

Praveen Paruchuri and **Jonathan P. Pearce**
Milind Tambe and **Fernando Ordóñez**
Univ. of Southern California, Los Angeles, CA
paruchur,jppearce,tambe,fordon@usc.edu

Sarit Kraus
Bar-Ilan University, Ramat-Gan 52900, Israel
sarit@cs.biu.il

Abstract

In adversarial multiagent domains, security, commonly defined as the ability to deal with intentional threats from other agents, is a critical issue. This paper focuses on domains where these threats come from unknown adversaries. These domains can be modeled as Bayesian games; much work has been done on finding equilibria for such games. However, it is often the case in multiagent security domains that one agent can commit to a mixed strategy which its adversaries observe before choosing their own strategies. In this case, the agent can maximize reward by finding an optimal strategy, without requiring equilibrium. Previous work has shown this problem of optimal strategy selection to be NP-hard. Therefore, we present a heuristic called ASAP, with three key advantages to address the problem. First, ASAP searches for the highest-reward strategy, rather than a Bayes-Nash equilibrium, allowing it to find feasible strategies that exploit the natural first-mover advantage of the game. Second, it provides strategies which are simple to understand, represent, and implement. Third, it operates directly on the compact, Bayesian game representation, without requiring conversion to normal form. We provide an efficient Mixed Integer Linear Program (MILP) implementation for ASAP, along with experimental results illustrating significant speedups and higher rewards over other approaches.

Introduction

In many multiagent domains, agents must act in order to provide security against attacks by adversaries. A common issue that agents face in such security domains is uncertainty about the adversaries they may be facing. For example, a security robot may need to make a choice about which areas to patrol, and how often (Ruan *et al.* 2005). However, it will not know in advance exactly where a robber will choose to strike. A team of unmanned aerial vehicles (UAVs) (Beard & McLain 2003) monitoring a region undergoing a humanitarian crisis may also need to choose a patrolling policy. They must make this decision without knowing in advance whether terrorists or other adversaries may be waiting to disrupt the mission at a given location. It may indeed be possible to model the motivations of types of adversaries the agent or agent team is likely to face in order to target these adversaries more closely. However, in both cases, the security robot or UAV team will not know exactly which kinds of adversaries may be active on any given day.

A common approach to choose a policy for agents in such scenarios is to model them as Bayesian games. A Bayesian game is a game in which agents may belong to one or more types; the type determines the agents possible actions and payoffs. The distribution of adversary types that an agent will face may be known or inferred from historical data. Usually, these games are analyzed using the solution concept of Bayes-Nash equilibrium, an extension of the Nash equilibrium for Bayesian games. However, in many settings, a Nash or Bayes-Nash equilibrium is not an appropriate solution concept, since it assumes that the agents' strategies are chosen simultaneously (Conitzer & Sandholm 2006).

In some settings, one player can commit to a strategy before the other players choose their strategies, and by doing so, attain a higher reward than if the strategies were chosen simultaneously. These scenarios are known as Stackelberg games (Fudenberg & Tirole 1991). In a Stackelberg game, a leader commits to a strategy first, and then a follower (or group of followers) selfishly optimize their own rewards, *considering the action chosen by the leader.* For example, the security agent (leader) may first commit to a mixed strategy for patrolling various areas in order to be unpredictable to the robbers (followers). The robbers, after observing the pattern of patrols over time, can then choose their own strategy of choosing a location to rob.

To see the advantage of being the leader in a Stackelberg game, consider a simple game with the payoff table as shown in Table 1. The leader is the row player and the follower is the column player. Here, the leader's payoff is listed first.

	1	2	3
1	5,5	0,0	3,10
2	0,0	2,2	5,0

Table 1: Payoff table for example normal form game.

The only Nash equilibrium for this game is when the leader and the follower plays 2 which gives the leader a payoff of 2. However, if the leader commits to a uniform mixed strategy of playing 1 and 2 with equal (0.5) probability, the follower's best response is to play 3 to get an expected pay-

Copyright © 2007, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

off of 5 (10 and 0 with equal probability). The leader's payoff would then be 4 (3 and 5 with equal probability). In this case, the leader now has an incentive to deviate and choose a pure strategy of 2 (to get a payoff of 5). However, this would cause the follower to deviate to strategy 2 as well, resulting in the Nash equilibrium. Thus, by committing to a strategy that is observed by the follower, and by avoiding the temptation to deviate, the leader manages to obtain a reward higher than that of the best Nash equilibrium.

The problem of choosing an optimal strategy for the leader to commit to in a Stackelberg game is analyzed in (Conitzer & Sandholm 2006) and found to be NP-hard in the case of a Bayesian game with multiple types of followers. Thus, efficient heuristic techniques for choosing high-reward strategies in these games is an important open issue. Methods for finding optimal leader strategies for non-Bayesian games (Conitzer & Sandholm 2006) can be applied to this problem by converting the Bayesian game into a normal-form game by the Harsanyi transformation (Harsanyi & Selten 1972). If, on the other hand, we wish to compute the highest-reward Nash equilibrium, new methods using mixed-integer linear programs (MILPs) (Sandholm, Gilpin, & Conitzer 2005) may be used, since the highest-reward Bayes-Nash equilibrium is equivalent to the corresponding Nash equilibrium in the transformed game. However, by transforming the game, the compact structure of the Bayesian game is lost. In addition, since the Nash equilibrium assumes a simultaneous choice of strategies, the advantages of being the leader are not considered.

In this paper we introduce an efficient heuristic method for approximating the optimal leader strategy for security domains, known as ASAP (Agent Security via Approximate Policies). This method has three key advantages. First, it directly searches for an optimal strategy, rather than a Nash (or Bayes-Nash) equilibrium, thus allowing it to find high-reward non-equilibrium strategies like the one in the above example. Second, it generates policies with a support which can be expressed as a uniform distribution over a multiset of fixed size as proposed in (Lipton, Markakis, & Mehta 2003). This allows for policies that are simple to understand and represent (Lipton, Markakis, & Mehta 2003), as well as a parameter (size of the multiset) that controls the simplicity of the policy and can be tuned. Third, the method allows for a Bayes-Nash game to be expressed compactly without needing conversion to a normal-form game, allowing for large speedups over existing Nash methods like (Sandholm, Gilpin, & Conitzer 2005) and (Lemke & Howson 1964).

The rest of the paper is organized as follows. In the next section we fully describe the patrolling domain and its properties. We then introduce the Bayesian game, the Harsanyi transformation, and existing methods for finding an optimal leader's strategy in a Stackelberg game. Next, the ASAP algorithm is presented for normal-form games, and we show how it can be adapted to the structure of Bayesian games with uncertain adversaries. Experimental results showing a higher reward and faster policy computation in comparison to existing Nash methods are shown in the Experimental results section. We then conclude with a discussion of related work.

The Patrolling Domain

In most security patrolling domains, the security agents (which could be UAVs (Beard & McLain 2003) or security robots (Ruan *et al.* 2005)) cannot feasibly patrol all areas all the time. Instead, they choose a policy by which they patrol various routes at different times, accounting for factors like the likelihood of crime in different areas, possible targets for crime, and the security agents' own resources (number of security agents, amount of available time, fuel, etc.). It is usually beneficial for this policy to be nondeterministic so that robbers cannot safely rob certain locations, knowing that they will be safe from the security agents (Paruchuri *et al.* 2006). To demonstrate the utility of our algorithm, we express a simplified version of such a domain as a game.

The most basic version of our game consists of two players: the security agent (the leader) and the robber (the follower) in a world consisting of m houses, $1 \dots m$. The security agent's set of pure strategies consists of possible routes of d houses to patrol (in an order). The security agent can choose a mixed strategy so that the robber will be unsure of exactly where the security agent may patrol, but the robber will know the mixed strategy the security agent has chosen. For example, the robber can observe over time how often the security agent patrols each area. With this knowledge, the robber must choose a single house to rob. We assume that the robber generally takes a long time to rob a house. If the house chosen by the robber is not on the security agent's route, then the robber successfully robs the house. Otherwise, if it is on the security agent's route, then the earlier the house is on the route, the easier it is for the security agent to catch the robber before he finishes robbing it.

We model the payoffs for this game with the following variables:

- $v_{l,x}$: value of the goods in house l to the security agent.
- $v_{l,q}$: value of the goods in house l to the robber.
- c_x: reward to the security agent of catching the robber.
- c_q: cost to the robber of getting caught.
- p_l: probability that the security agent can catch the robber at the lth house in the patrol ($p_l < p_{l'} \iff l' < l$).

The security agent's set of possible pure strategies (patrol routes) is denoted by X and includes all d-tuples $i =< w_1, w_2, ..., w_d >$ with $w_1 \dots w_d = 1 \dots m$. where no two elements are equal (the agent is not allowed to return to the same house). The robber's set of possible pure strategies (houses to rob) is denoted by Q and includes all integers $j = 1 \dots m$. The payoffs (security agent, robber) for pure strategies i, j are:

- $-v_{l,x}, v_{l,q}$, for $j = l \notin i$.
- $p_l c_x + (1 - p_l)(-v_{l,x}), -p_l c_q + (1 - p_l)(v_{l,q})$, for $j = l \in i$.

With this structure it is possible to model many different types of robbers who have differing motivations; for example, one robber may have a lower cost of getting caught than another, or may value the goods in the various houses differently. If the distribution of different robber types is known or inferred from historical data, then the game can be modeled as a Bayesian game (Fudenberg & Tirole 1991).

Bayesian Games

A Bayesian game contains a set of N agents, and each agent n must be one of a given set of types θ_n. For our patrolling domain, we have two agents, the security agent and the robber. θ_1 is the set of security agent types and θ_2 is the set of robber types. Since there is only one type of security agent, θ_1 contains only one element. During the game, the robber knows its type but the security agent does not know the robber's type. For each agent (the security agent or the robber) n, there is a set of strategies σ_n and a utility function $u_n : \theta_1 \times \theta_2 \times \sigma_1 \times \sigma_2 \rightarrow \Re$.

A Bayesian game can be transformed into a normal-form game using the Harsanyi transformation (Harsanyi & Selten 1972). For the transformed game obtained, new, linear-program (LP)-based methods for finding high-reward strategies for normal-form games (Conitzer & Sandholm 2006) can be used to find a strategy; this strategy can then be used for the Bayesian game. While methods exist for finding Bayes-Nash equilibria directly, without the Harsanyi transformation (Koller & Pfeffer 1997), they find only a single equilibrium in the general case, which may not be of high reward. Recent work (Sandholm, Gilpin, & Conitzer 2005) has led to efficient mixed-integer linear program techniques to find the best Nash equilibrium for a given agent. However, these techniques do require a normal-form game, and hence to compare the ASAP policy against the optimal policy, as well as the highest-reward Nash equilibrium, we must apply these techniques to the Harsanyi-transformed matrix. The next two subsections elaborate on how this is done.

Harsanyi Transformation

The first step to solve Bayesian games is to apply the Harsanyi transformation (Harsanyi & Selten 1972) that converts the incomplete information game into a normal form game. Given that the Harsanyi transformation is a standard concept in game theory, we explain it briefly through a simple example without introducing the mathematical formulations. Let us assume there are two robber types a and b in the Bayesian game. Robber a is active with probability α, and robber b, active with probability $1 - \alpha$. The rules described in the domain section allow us to construct simple payoff tables.

Assume that there are two houses in the world (1 and 2) and hence there are two patrol routes (pure strategies) for the agent: {1,2} and {2,1}. The robber can rob either house 1 or house 2 and hence he has two strategies (denoted as 1_l, 2_l for robber type l). Since there are two types assumed (denoted as a and b), we construct two payoff tables (shown in Table 2) corresponding to the security agent playing a separate game with each of the two robber types with probabilities α and $1 - \alpha$. First, consider robber type a. Borrowing the notation from the domain section, we assign the following values to the variables: $v_{1,x} = v_{1,q} = 3/4, v_{2,x} = v_{2,q} = 1/4, c_x = 1/2, c_q = 1, p_1 = 1, p_2 = 1/2$. Using these values we construct a base payoff table as the payoff for the game against robber type a. For example, if the security agent chooses route {1,2} when robber a is active, and robber a chooses house 1, the robber receives a reward of -1 (for being caught) and the agent receives a reward of 0.5 for catching the robber. The payoffs for the game against robber type b are constructed using different values.

Security agent:	{1,2}	{2,1}
Robber a		
1_a	-1, .5	-.375, .125
2_a	-.125, -.125	-1, .5
Robber b		
1_b	-.9, .6	-.275, .225
2_b	-.025, -.025	-.9, .6

Table 2: Payoff tables: Security Agent vs Robbers a and b

Using the Harsanyi technique involves introducing a chance node, that determines the robber's type, thus transforming the security agent's incomplete information regarding the robber into imperfect information (Brynielsson & Arnborg 2004). The Bayesian equilibrium of the game is then precisely the Nash equilibrium of the imperfect information game. The transformed, normal-form game is shown in Table 3. In the transformed game, the security agent is the column player, and the set of all robber types together is the row player. Suppose that robber type a robs house 1 and robber type b robs house 2, while the security agent chooses patrol {1,2}. Then, the security agent and the robber receive an expected payoff corresponding to their payoffs from the agent encountering robber a at house 1 with probability α and robber b at house 2 with probability $1 - \alpha$.

Finding an Optimal Strategy

Although a Nash equilibrium is the standard solution concept for games in which agents choose strategies simultaneously, in our security domain, the security agent (the leader) can gain an advantage by committing to a mixed strategy in advance. Since the followers (the robbers) will know the leader's strategy, the optimal response for the followers will be a pure strategy. Given the common assumption, taken in (Conitzer & Sandholm 2006), in the case where followers are indifferent, they will choose the strategy that benefits the leader, there must exist a guaranteed optimal strategy for the leader (Conitzer & Sandholm 2006).

From the Bayesian game in Table 2, we constructed the Harsanyi transformed bimatrix in Table 3. We denote $X = \sigma_1^{\theta_2} = \sigma_1$ and $Q = \sigma_2^{\theta_2}$ as the index sets of the security agent and robbers' pure strategies, respectively, with R and C as the corresponding payoff matrices. R_{ij} is the reward of the security agent and C_{ij} is the reward of the robbers when the security agent takes pure strategy i and the robbers take pure strategy j. A mixed strategy for the security agent is a probability distribution over its set of pure strategies and will be represented by a vector $x = (p_{x1}, p_{x2}, \ldots, p_{x|X|})$, where $p_{xi} \geq 0$ and $\sum p_{xi} = 1$. Here, p_{xi} is the probability that the security agent will choose its ith pure strategy.

The optimal mixed strategy for the security agent can be found in time polynomial in the number of rows in the normal form game using the following linear program formulation from (Conitzer & Sandholm 2006). For every possible pure strategy j by the follower (the set of all robber types),

	{1,2}	{2,1 }
$\{1_a, 1_b\}$	$-1\alpha - .9(1-\alpha), .5\alpha + .6(1-\alpha)$	$-.375\alpha - .275(1-\alpha), .125\alpha + .225(1-\alpha)$
$\{1_a, 2_b\}$	$-1\alpha - .025(1-\alpha), .5\alpha - .025(1-\alpha)$	$-.375\alpha - .9(1-\alpha), .125\alpha + .6(1-\alpha)$
$\{2_a, 1_b\}$	$-.125\alpha - .9(1-\alpha), -.125\alpha + .6(1-\alpha)$	$-1\alpha - .275(1-\alpha), .5\alpha + .225(1-\alpha)$
$\{2_a, 2_b\}$	$-.125\alpha - .025(1-\alpha), -.125\alpha - .025(1-\alpha)$	$-1\alpha - .9(1-\alpha), .5\alpha + .6(1-\alpha)$

Table 3: Harsanyi Transformed Payoff Table

$$\begin{array}{ll} \max & \sum_{i\in X} p_{xi} R_{ij} \\ \text{s.t.} & \forall j' \in Q, \sum_{i\in\sigma_1} p_{xi} C_{ij} \geq \sum_{i\in\sigma_1} p_{xi} C_{ij'} \\ & \sum_{i\in X} p_{xi} = 1 \\ & \forall_{i\in X}, p_{xi} >= 0 \end{array} \qquad (1)$$

Then, for all feasible follower strategies j, choose the one that maximizes $\sum_{i\in X} p_{xi} R_{ij}$, the reward for the leader. The p_{xi} variables give the optimal strategy for the security agent.

Note that while this method is polynomial in the number of rows in the transformed normal-form game, the number of rows increases exponentially with the number of robber types. Thus, this method needs running $|\sigma_2|^{|\theta_2|}$ separate LPs for a Bayesian game. This is not a surprise, since finding the optimal strategy to commit to for the leader in a Bayesian game is NP-hard (Conitzer & Sandholm 2006).

Heuristic Approaches

Given that finding the optimal strategy for the leader is NP-hard, we provide a heuristic approach. In this heuristic we limit the possible mixed strategies of the leader to select actions with probabilities that are integer multiples of $1/k$ for a predetermined integer k. Previous work (Paruchuri *et al.* 2006) has shown that strategies with high entropy are beneficial for security applications when opponents' utilities are completely unknown. Our method will result in uniform-distribution strategies as solutions. One advantage of such strategies is that they are compact to represent (as fractions) and simple to understand; hence are efficiently implemented by real organizations. We aim to maintain the advantage of simple strategies for our security application problem, incorporating the effect of the robbers' rewards on the agent's rewards. Thus, the ASAP heuristic will produce strategies which are *k-uniform*. A mixed strategy is denoted k-uniform if it is a uniform distribution on a multiset S of pure strategies with $|S| = k$. A multiset is a set whose elements may be repeated multiple times; thus, for example, the mixed strategy corresponding to the multiset $\{1, 1, 2\}$ would take strategy 1 with probability 2/3 and strategy 2 with probability 1/3. ASAP allows the size of the multiset to be chosen in order to balance the complexity of the strategy reached with the goal that the identified strategy will yield a high reward.

Another advantage of the ASAP heuristic is that it operates directly on the compact Bayesian representation, without needing the Harsanyi transformation. This is because the different follower types are independent of each other. Hence, evaluating the leader strategy against a Harsanyi-transformed game is equivalent to evaluating against each of the game matrices for the individual follower types. This independence property is exploited in ASAP to yield a decomposition scheme. Note that the LP method introduced by (Conitzer & Sandholm 2006) to compute optimal Stackelberg policies is unlikely to be decomposable into a small number of games as it was shown to be NP-hard for Bayes-Nash problems. Finally, note that ASAP requires the solution of only one optimization problem, rather than solving a series of problems as in the LP method of problem 1.

For a single follower type, the algorithm works the following way. Given a particular k, for each possible mixed strategy x of the leader that corresponds to multiset of size k, evaluate the leader's payoff from x when the follower plays a reward-maximizing pure strategy. We then take the mixed strategy with the highest payoff. We need to consider only the reward-maximizing pure strategies of the followers, since for a given fixed strategy x of the agent, each robber type faces a problem with fixed linear rewards. If a mixed strategy is optimal for the robber, then so are all the pure strategies in the support of that mixed strategy. Note that because the leader's strategies are limited to discrete values, the assumption from previous section that followers break ties in the leader's favor is not significant, since ties will be unlikely to arise. This is because, in domains where rewards are drawn from any random distribution, the probability of a follower having more than one pure optimal response to a given leader strategy approaches zero, and the leader will have only a finite number of possible mixed strategies.

Our approach to characterize the optimal strategy for the security agent makes use of properties of linear programming. We briefly outline these results here for completeness, for detailed discussion and proofs see one of many references on the topic, such as (Bertsimas & Tsitsiklis 1997). Every linear programming problem, such as:

$$\max c^T x \mid Ax = b, x \geq 0,$$

has an associated dual linear program, in this case:

$$\min b^T y \mid A^T y \geq c.$$

These primal/dual pairs of problems satisfy weak duality: For any x and y primal and dual feasible solutions respectively, $c^T x \leq b^T y$. Thus a pair of feasible solutions is optimal if $c^T x = b^T y$, and the problems are said to satisfy strong duality. In fact if a linear program is feasible and has a bounded optimal solution, then the dual is also feasible and there is a pair x^*, y^* that satisfies $c^T x^* = b^T y^*$. These optimal solutions are characterized with the following optimality conditions:

- primal feasibility: $Ax = b,\ x \geq 0$
- dual feasibility: $A^T y \geq c$

- complementary slackness: $x_i(A^Ty - c)_i = 0$ for all i.

Note that this last condition implies that

$$c^Tx = x^TA^Ty = b^Ty,$$

which proves optimality for primal dual feasible solutions x and y. In the following subsections, we first define the problem in its most intuitive form as a mixed-integer quadratic program, and then show how this problem can be converted into a mixed-integer linear program.

Mixed-Integer Quadratic Program

We begin with the case of a single follower type. Let the leader be the row player and follower, the column player. We denote by x the vector of strategies of the leader and q the vector of strategies of the follower. We also denote X and Q the index sets of the leader and follower's pure strategies, respectively. The payoff matrices R and C correspond to: R_{ij} is the reward of the leader and C_{ij}, the reward of the follower when the leader takes pure strategy i and the follower takes pure strategy j. Let k be the size of the multiset.

We first fix the policy of the leader to some k-uniform policy x. The value x_i is the number of times pure strategy i is used in the k-uniform policy, which is selected with probability x_i/k. We formulate the optimization problem the follower solves to find its optimal response to x as the following linear program:

$$\begin{array}{ll} \max & \displaystyle\sum_{j\in Q}\sum_{i\in X}\frac{1}{k}C_{ij}x_i\ q_j \\ \text{s.t.} & \sum_{j\in Q} q_j = 1 \\ & q \geq 0. \end{array} \tag{2}$$

The objective function maximizes the follower's expected reward given x, while the constraints make feasible any mixed strategy q for the follower. The dual to this linear programming problem is the following:

$$\begin{array}{ll} \min & a \\ \text{s.t.} & a \geq \displaystyle\sum_{i\in X}\frac{1}{k}C_{ij}x_i \quad j \in Q. \end{array} \tag{3}$$

From strong duality and complementary slackness we obtain that the maximum reward value for the follower a is the value of every pure strategy with $q_j > 0$, that is in the support of the optimal mixed strategy. Therefore each of these pure strategies is optimal. Optimal solutions to the follower's problem are characterized by the LP optimality conditions: primal feasibility constraints in (2), dual feasibility constraints in (3), and complementary slackness

$$q_j\left(a - \sum_{i\in X}\frac{1}{k}C_{ij}x_i\right) = 0 \qquad j \in Q.$$

These conditions must be included in the problem solved by the leader in order to consider only best responses by the follower to the k-uniform policy x. The leader seeks the k-uniform solution x that maximizes its own payoff, given that the follower uses an optimal response $q(x)$. Therefore the leader solves the following integer problem:

$$\begin{array}{ll} \max & \displaystyle\sum_{i\in X}\sum_{j\in Q}\frac{1}{k}R_{ij}q(x)_j\ x_i \\ \text{s.t.} & \sum_{i\in X} x_i = k \\ & x_i \in \{0, 1, \ldots, k\}. \end{array} \tag{4}$$

Problem (4) maximizes the leader's reward with the follower's best response (q_j for fixed leader's policy x and hence denoted $q(x)_j$) by selecting a uniform policy from a multiset of constant size k. We complete this problem by including the characterization of $q(x)$ through linear programming optimality conditions. To simplify writing the complementary slackness conditions, we will constrain $q(x)$ to be only optimal pure strategies by just considering integer solutions of $q(x)$. The leader's problem becomes:

$$\begin{array}{ll} \max_{x,q} & \displaystyle\sum_{i\in X}\sum_{j\in Q}\frac{1}{k}R_{ij}x_iq_j \\ \text{s.t.} & \sum_i x_i = k \\ & \sum_{j\in Q} q_j = 1 \\ & 0 \leq (a - \sum_{i\in X}\frac{1}{k}C_{ij}x_i) \leq (1 - q_j)M \\ & x_i \in \{0, 1, ..., k\} \\ & q_j \in \{0, 1\}. \end{array} \tag{5}$$

Here, the constant M is some large number. The first and fourth constraints enforce a k-uniform policy for the leader, and the second and fifth constraints enforce a feasible pure strategy for the follower. The third constraint enforces dual feasibility of the follower's problem (leftmost inequality) and the complementary slackness constraint for an optimal pure strategy q for the follower (rightmost inequality). In fact, since only one pure strategy can be selected by the follower, say $q_h = 1$, this last constraint enforces that $a = \sum_{i\in X}\frac{1}{k}C_{ih}x_i$ imposing no additional constraint for all other pure strategies which have $q_j = 0$.

We conclude this subsection noting that Problem (5) is an integer program with a non-convex quadratic objective in general, as the matrix R need not be positive-semi-definite. Efficient solution methods for non-linear, non-convex integer problems remains a challenging research question. In the next section we show a reformulation of this problem as a linear integer programming problem, for which a number of efficient commercial solvers exist.

Mixed-Integer Linear Program

We linearize the quadratic program of Problem 5 through the change of variables $z_{ij} = x_iq_j$, thus obtaining

$$\begin{array}{ll} \max_{q,z} & \sum_{i\in X}\sum_{j\in Q}\frac{1}{k}R_{ij}z_{ij} \\ \text{s.t.} & \sum_{i\in X}\sum_{j\in Q} z_{ij} = k \\ & \sum_{j\in Q} z_{ij} \leq k \\ & kq_j \leq \sum_{i\in X} z_{ij} \leq k \\ & \sum_{j\in Q} q_j = 1 \\ & 0 \leq (a - \sum_{i\in X}\frac{1}{k}C_{ij}(\sum_{h\in Q} z_{ih})) \leq (1 - q_j)M \\ & z_{ij} \in \{0, 1, ..., k\} \\ & q_j \in \{0, 1\} \end{array} \tag{6}$$

Proposition 1 *Problems (5) and (6) are equivalent.*

Proof: Consider x, q a feasible solution of (5). We will show that q, $z_{ij} = x_i q_j$ is a feasible solution of (6) of same objective function value. The equivalence of the objective functions, and constraints 4, 6 and 7 of (6) are satisfied by construction. The fact that $\sum_{j \in Q} z_{ij} = x_i$ as $\sum_{j \in Q} q_j = 1$ explains constraints 1, 2, and 5 of (6). Constraint 3 of (6) is satisfied because $\sum_{i \in X} z_{ij} = k q_j$.

Let us now consider q, z feasible for (6). We will show that q and $x_i = \sum_{j \in Q} z_{ij}$ are feasible for (5) with the same objective value. In fact all constraints of (5) are readily satisfied by construction. To see that the objectives match, notice that if $q_h = 1$ then the third constraint in (6) implies that $\sum_{i \in X} z_{ih} = k$, which means that $z_{ij} = 0$ for all $i \in X$ and all $j \neq h$. Therefore,

$$x_i q_j = \sum_{l \in Q} z_{il} q_j = z_{ih} q_j = z_{ij}.$$

This last equality is because both are 0 when $j \neq h$. This shows that the transformation preserves the objective function value, completing the proof.

Given this transformation to a mixed-integer linear program (MILP), we now show how we can apply our decomposition technique on the MILP to obtain significant speedups for Bayesian games with multiple follower types.

Decomposition for Multiple Adversaries

The MILP developed in the previous section handles only one follower. Since our security scenario contains multiple follower (robber) types, we change the response function for the follower from a pure strategy into a weighted combination over various pure follower strategies where the weights are probabilities of occurrence of each of the follower types.

Decomposed MIQP

To admit multiple adversaries in our framework, we modify the notation defined in the previous section to reason about multiple follower types. We denote by x the vector of strategies of the leader and q^l the vector of strategies of follower l, with L denoting the index set of follower types. We also denote by X and Q the index sets of leader and follower l's pure strategies, respectively. We also index the payoff matrices on each follower l, considering the matrices R^l and C^l.

Using this modified notation, we characterize the optimal solution of follower l's problem given the leaders k-uniform policy x, with the following optimality conditions:

$$\begin{aligned} \sum_{j \in Q} q_j^l &= 1 \\ a^l - \sum_{i \in X} \frac{1}{k} C_{ij}^l x_i &\geq 0 \\ q_j^l (a^l - \sum_{i \in X} \frac{1}{k} C_{ij}^l x_i) &= 0 \\ q_j^l &\geq 0 \end{aligned}$$

Again, considering only optimal pure strategies for follower l's problem we can linearize the complementarity constraint above. We incorporate these constraints on the leader's problem that selects the optimal k-uniform policy. Therefore, given *a priori* probabilities p^l, with $l \in L$ of facing each follower, the leader solves the following problem:

$$\begin{array}{ll} \max_{x,q} & \displaystyle\sum_{i \in X} \sum_{l \in L} \sum_{j \in Q} \frac{p^l}{k} R_{ij}^l x_i q_j^l \\ \text{s.t.} & \sum_i x_i = k \\ & \sum_{j \in Q} q_j^l = 1 \\ & 0 \leq (a^l - \sum_{i \in X} \frac{1}{k} C_{ij}^l x_i) \leq (1 - q_j^l) M \\ & x_i \in \{0, 1, ..., k\} \\ & q_j^l \in \{0, 1\}. \end{array} \quad (7)$$

Problem (7) for a Bayesian game with multiple follower types is indeed equivalent to Problem (5) on the payoff matrix obtained from the Harsanyi transformation of the game. In fact, every pure strategy j in Problem (5) corresponds to a sequence of pure strategies j_l, one for each follower $l \in L$. This means that $q_j = 1$ if and only if $q_{j_l}^l = 1$ for all $l \in L$. In addition, given the *a priori* probabilities p^l of facing player l, the reward in the Harsanyi transformation payoff table is $R_{ij} = \sum_{l \in L} p^l R_{ij_l}^l$. The same relation holds between C and C^l. These relations between a pure strategy in the equivalent normal form game and pure strategies in the individual games with each followers are key in showing these problems are equivalent.

Decomposed MILP

We can linearize the quadratic programming problem 7 through the change of variables $z_{ij}^l = x_i q_j^l$, obtaining the following problem

$$\begin{array}{ll} \max_{q,z} & \sum_{i \in X} \sum_{l \in L} \sum_{j \in Q} \frac{p^l}{k} R_{ij}^l z_{ij}^l \\ \text{s.t.} & \sum_{i \in X} \sum_{j \in Q} z_{ij}^l = k \\ & \sum_{j \in Q} z_{ij}^l \leq k \\ & k q_j^l \leq \sum_{i \in X} z_{ij}^l \leq k \\ & \sum_{j \in Q} q_j^l = 1 \\ & 0 \leq (a^l - \sum_{i \in X} \frac{1}{k} C_{ij}^l (\sum_{h \in Q} z_{ih}^l)) \leq (1 - q_j^l) M \\ & \sum_{j \in Q} z_{ij}^l = \sum_{j \in Q} z_{ij}^1 \\ & z_{ij}^l \in \{0, 1, ..., k\} \\ & q_j^l \in \{0, 1\} \end{array} \quad (8)$$

Proposition 2 *Problems (7) and (8) are equivalent.*

Proof: Consider x, q^l, a^l with $l \in L$ a feasible solution of (7). We will show that q^l, a^l, $z_{ij}^l = x_i q_j^l$ is a feasible solution of (8) of same objective function value. The equivalence of the objective functions, and constraints 4, 7 and 8 of (8) are satisfied by construction. The fact that $\sum_{j \in Q} z_{ij}^l = x_i$ as $\sum_{j \in Q} q_j^l = 1$ explains constraints 1, 2, 5 and 6 of (8). Constraint 3 of (8) is satisfied because $\sum_{i \in X} z_{ij}^l = k q_j^l$.

Lets now consider q^l, z^l, a^l feasible for (8). We will show that q^l, a^l and $x_i = \sum_{j \in Q} z_{ij}^1$ are feasible for (7) with the same objective value. In fact all constraints of (7) are readily satisfied by construction. To see that the objectives match, notice for each l one q^l_j must equal 1 and the rest equal 0. Let us say that $q^l_{j_l} = 1$, then the third constraint in (8) implies that $\sum_{i\in X} z^l_{ij_l} = k$, which means that $z^l_{ij} = 0$ for all $i \in X$ and all $j \neq j_l$. In particular this implies that

$$x_i = \sum_{j\in Q} z^1_{ij} = z^1_{ij_1} = z^l_{ij_l},$$

this last equality from constraint 6 of (8). Therefore $x_i q^l_j = z^l_{ij_l} q^l_j = z^l_{ij}$. This last equality is because both are 0 when $j \neq j_l$. This shows that the transformation preserves the objective function value, completing the proof.

We can therefore solve this equivalent linear integer program with efficient integer programming packages which can handle problems with thousands of integer variables. We implemented the decomposed MILP and the results are shown in the following section.

Experimental results

The patrolling domain and the payoffs for the associated game are detailed in sections describing the Patrolling domain and the Bayesian games. We performed experiments for this game in worlds of three and four houses with patrols consisting of two houses. The description provided in the domain section earlier is used to generate a base case for both the security agent and robber payoff functions. The payoff tables for additional robber types are constructed and added to the game by adding a random distribution of varying size to the payoffs in the base case. All games are normalized so that, for each robber type, the minimum and maximum payoffs to the security agent and robber are 0 and 1, respectively.

Using the data generated, we performed the experiments using four methods for generating the security agent's strategy:

- uniform randomization
- ASAP
- the multiple linear programs method from (Conitzer & Sandholm 2006) (to find the true optimal strategy)
- the highest reward Bayes-Nash equilibrium, found using the MIP-Nash algorithm (Sandholm, Gilpin, & Conitzer 2005)

The last three methods were applied using CPLEX 8.1. Because the last two methods are designed for normal-form games rather than Bayesian games, the games were first converted using the Harsanyi transformation (Harsanyi & Selten 1972). The uniform randomization method is simply choosing a uniform random policy over all possible patrol routes. We use this method as a simple baseline to measure the performance of our heuristics. We anticipated that the uniform policy would perform reasonably well since maximum-entropy policies have been shown to be effective in multiagent security domains (Paruchuri *et al.* 2006). The highest-reward Bayes-Nash equilibria were used in order to demonstrate the higher reward gained by looking for an optimal policy rather than an equilibria in Stackelberg games such as our security domain.

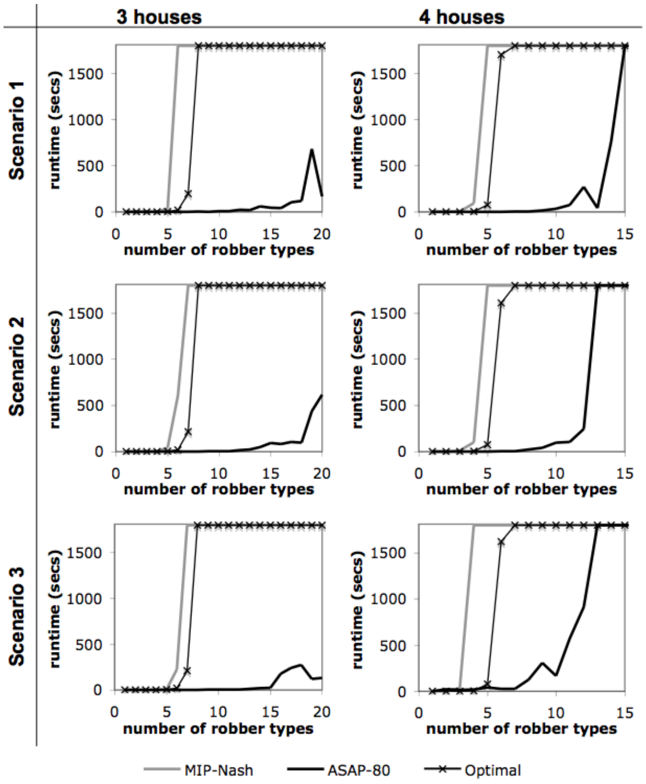

Figure 1: Runtimes for various algorithms on problems of 3 and 4 houses.

Based on our experiments we present three sets of graphs to demonstrate (1) the runtime of ASAP compared to other common methods for finding a strategy, (2) the reward guaranteed by ASAP compared to other methods, and (3) the effect of varying the parameter k, the size of the multiset, on the performance of ASAP. In the first two sets of graphs, ASAP is run using a multiset of 80 elements; in the third set this number is varied.

The first set of graphs, shown in Figure 1 shows the runtime graphs for three-house (left column) and four-house (right column) domains. Each of the three rows of graphs corresponds to a different randomly-generated scenario. The x-axis shows the number of robber types the security agent faces and the y-axis of the graph shows the runtime in seconds. All experiments that were not concluded in 30 minutes (1800 seconds) were cut off. The runtime for the uniform policy is always negligible irrespective of the number of adversaries and hence is not shown.

The ASAP algorithm clearly outperforms the optimal, multiple-LP method as well as the MIP-Nash algorithm for finding the highest-reward Bayes-Nash equilibrium with respect to runtime. For a domain of three houses, the optimal method cannot reach a solution for more than seven robber types, and for four houses it cannot solve for more than six

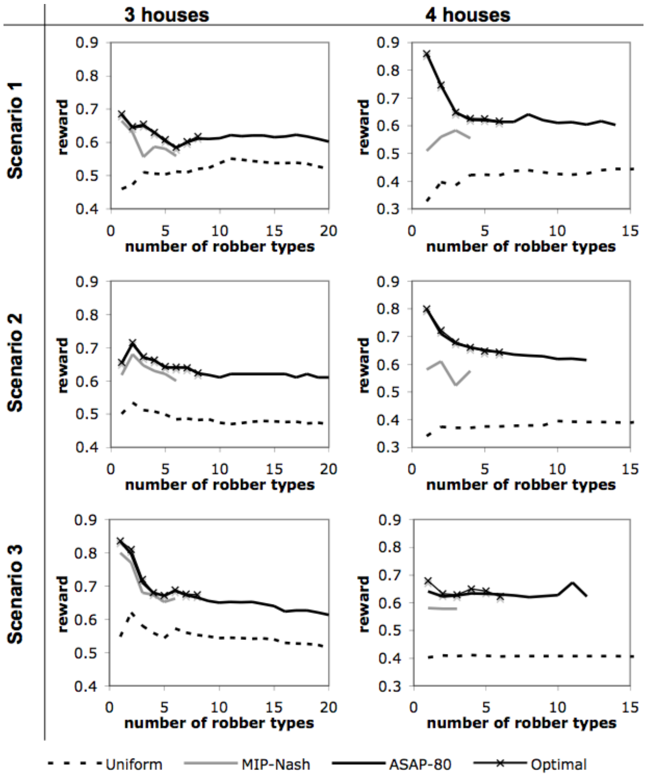

Figure 2: Reward for various algorithms on problems of 3 and 4 houses.

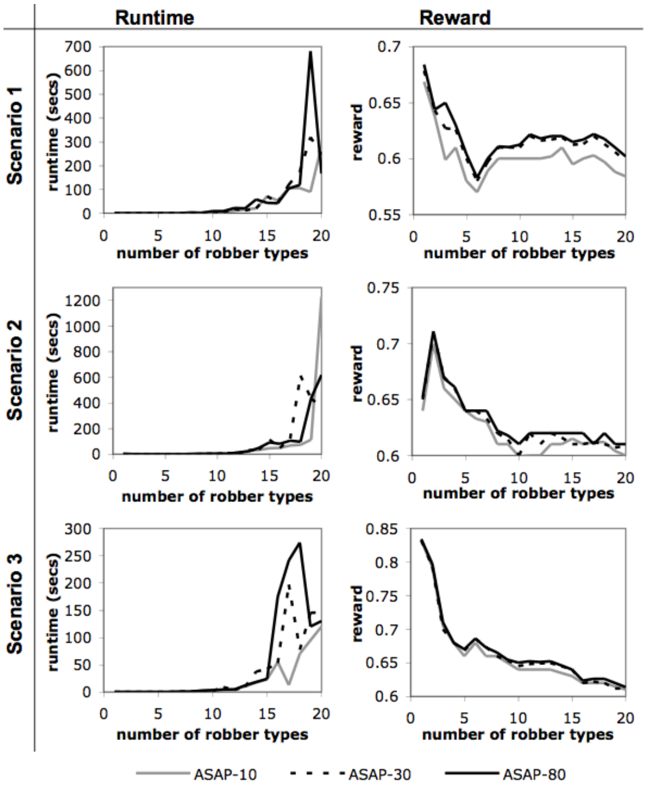

Figure 3: Reward for ASAP using multisets of 10, 30, and 80 elements

types within the cutoff time in any of the three scenarios. MIP-Nash solves for even fewer robber types within the cutoff time. On the other hand, ASAP runs much faster, and is able to solve for at least 20 adversaries for the three-house scenarios and for at least 12 adversaries in the four-house scenarios within the cutoff time. The runtime of ASAP does not increase strictly with the number of robber types for each scenario, but in general, the addition of more types increases the runtime required.

The second set of graphs, Figure 2, shows the reward to the patrol agent given by each method for three scenarios in the three-house (left column) and four-house (right column) domains. This reward is the utility received by the security agent in the patrolling game, and not as a percentage of the optimal reward, since it was not possible to obtain the optimal reward as the number of robber types increased. The uniform policy consistently provides the lowest reward in both domains; while the optimal method of course produces the optimal reward. The ASAP method remains consistently close to the optimal, even as the number of robber types increases. The highest-reward Bayes-Nash equilibria, provided by the MIP-Nash method, produced rewards higher than the uniform method, but lower than ASAP. This difference clearly illustrates the gains in the patrolling domain from committing to a strategy as the leader in a Stackelberg game, rather than playing a standard Bayes-Nash strategy.

The third set of graphs, shown in Figure 3 shows the effect of the multiset size on runtime in seconds (left column) and reward (right column), again expressed as the reward received by the security agent in the patrolling game, and not a percentage of the optimal reward. Results here are for the three-house domain. The trend is that as as the multiset size is increased, the runtime and reward level both increase. Not surprisingly, the reward increases monotonically as the multiset size increases, but what is interesting is that there is relatively little benefit to using a large multiset in this domain. In all cases, the reward given by a multiset of 10 elements was within at least 96% of the reward given by an 80-element multiset. The runtime does not always increase strictly with the multiset size; indeed in one example (scenario 2 with 20 robber types), using a multiset of 10 elements took 1228 seconds, while using 80 elements only took 617 seconds. In general, runtime should increase since a larger multiset means a larger domain for the variables in the MILP, and thus a larger search space. However, an increase in the number of variables can sometimes allow for a policy to be constructed more quickly due to more flexibility in the problem.

Summary and Related Work

This paper focuses on security for agents patrolling in hostile environments. In these environments, intentional threats are caused by adversaries about whom the security patrolling agents have incomplete information. Specifically, we deal with situations where the adversaries' actions and payoffs are known but the exact adversary type is unknown to the security agent. Agents acting in the real world quite frequently have such incomplete information about other agents. Bayesian games have been a popular choice to model such incomplete information games (Brynielsson & Arnborg 2004). The Gala toolkit is one method for defining such games (Koller & Pfeffer 1995) without requiring the game to be represented in normal form via the Harsanyi transformation (Harsanyi & Selten 1972); Gala's guarantees are focused on fully competitive games. Much work has been done on finding optimal Bayes-Nash equilbria for subclasses of Bayesian games, finding single Bayes-Nash equilibria for general Bayesian games (Koller & Pfeffer 1997) or approximate Bayes-Nash equilibria (Singh, Soni, & Wellman 2004). Less attention has been paid to finding the optimal strategy to commit to in a Bayesian game (the Stackelberg scenario (Roughgarden 2001)). However, the complexity of this problem was shown to be NP-hard in the general case (Conitzer & Sandholm 2006), which also provides algorithms for this problem in the non-Bayesian case.

Therefore, we present a heuristic called ASAP, with three key advantages towards addressing this problem. First, ASAP searches for the highest reward strategy, rather than a Bayes-Nash equilibrium, allowing it to find feasible strategies that exploit the natural first-mover advantage of the game. Second, it provides strategies which are simple to understand, represent, and implement. Third, it operates directly on the compact, Bayesian game representation, without requiring conversion to normal form. We provide an efficient Mixed Integer Linear Program (MILP) implementation for ASAP, along with experimental results illustrating significant speedups and higher rewards over other approaches.

As mentioned earlier, our k-uniform strategies are similar to the k-uniform strategies of (Lipton, Markakis, & Mehta 2003). While that work provides epsilon error-bounds based on the k-uniform strategies, their solution concept is still that of a Nash equilibrium, and they do not provide efficient algorithms for obtaining such k-uniform strategies. This contrasts with ASAP, where our emphasis is on a highly efficient heuristic approach that is not focused on equilibrium solutions.

Finally the patrolling problem which motivated our work has recently received growing attention from the multiagent community due to its wide range of applications (Chevaleyre 2004), (Machado *et al.* 2002). However most of this work is focused on either limiting energy consumption involved in patrolling (Gui & Mohapatra 2005) or optimizing on criteria like the length of the path traveled (Chevaleyre 2004), (Machado *et al.* 2002), without reasoning about any explicit model of an adversary(Paruchuri *et al.* 2006).

Acknowledgments : This research is supported by the United States Department of Homeland Security through Center for Risk and Economic Analysis of Terrorism Events (CREATE). Sarit Kraus is also affiliated with UMIACS.

References

Beard, R., and McLain, T. 2003. Multiple uav cooperative search under collision avoidance and limited range communication constraints. In *IEEE CDC*.

Bertsimas, D., and Tsitsiklis, J. 1997. *Introduction to Linear Optimization*. Athena Scientific.

Brynielsson, J., and Arnborg, S. 2004. Bayesian games for threat prediction and situation analysis. In *FUSION*.

Chevaleyre, Y. 2004. Theoretical analysis of multi-agent patrolling problem. In *AAMAS*.

Conitzer, V., and Sandholm, T. 2006. Choosing the best strategy to commit to. In *ACM Conference on Electronic Commerce*.

Fudenberg, D., and Tirole, J. 1991. *Game Theory*. MIT Press.

Gui, C., and Mohapatra, P. 2005. Virtual patrol: A new power conservation design for surveillance using sensor networks. In *IPSN*.

Harsanyi, J. C., and Selten, R. 1972. A generalized Nash solution for two-person bargaining games with incomplete information. *Management Science* 18(5):80–106.

Koller, D., and Pfeffer, A. 1995. Generating and solving imperfect information games. In *IJCAI*, 1185–1193.

Koller, D., and Pfeffer, A. 1997. Representations and solutions for game-theoretic problems. *Artificial Intelligence* 94(1):167–215.

Lemke, C., and Howson, J. 1964. Equilibrium points of bimatrix games. *Journal of the Society for Industrial and Applied Mathematics* 12:413–423.

Lipton, R. J.; Markakis, E.; and Mehta, A. 2003. Playing large games using simple strategies. In *ACM Conference on Electronic Commerce*.

Machado, A.; Ramalho, G.; Zucker, J. D.; and Drougoul, A. 2002. Multi-agent patrolling: an empirical analysis on alternative architectures. In *MABS*.

Paruchuri, P.; Tambe, M.; Ordonez, F.; and Kraus, S. 2006. Security in multiagent systems by policy randomization. In *AAMAS*.

Roughgarden, T. 2001. Stackelberg scheduling strategies. In *ACM Symposium on TOC*.

Ruan, S.; Meirina, C.; Yu, F.; Pattipati, K. R.; and Popp, R. L. 2005. Patrolling in a stochastic environment. In *10th Intl. Command and Control Research Symp.*

Sandholm, T.; Gilpin, A.; and Conitzer, V. 2005. Mixed-integer programming methods for finding nash equilibria. In *AAAI*.

Singh, S.; Soni, V.; and Wellman, M. 2004. Computing approximate Bayes-Nash equilibria with tree-games of incomplete information. In *ACM Conference on Electronic Commerce*.