Efficient Algorithms to Solve Bayesian Stackelberg Games for Security Applications

Praveen Paruchuri*, Jonathan P. Pearce, Janusz Marecki, Milind Tambe, Fernando Ordonez, Sarit Kraus**
*Intelligent Automation Inc., Rockville, MD, USA, (pparuchuri@i-a-i.com)
University of Southern California, Los Angeles, CA, USA ({jppearce, marecki, tambe, fordon}@usc.edu)
**Bar-Ilan University, Israel (sarit@macs.biu.ac.il)

Abstract

In a class of games known as Stackelberg games, one agent (the leader) must commit to a strategy that can be observed by the other agent (the adversary/follower) before the adversary chooses its own strategy. We consider Bayesian Stackelberg games, in which the leader is uncertain about the type of the adversary it may face. Such games are important in security domains, where, for example, a security agent (leader) must commit to a strategy of patrolling certain areas, and an adversary (follower) can observe this strategy over time before choosing where to attack. We present here two different MIP-formulations, ASAP (providing approximate policies with controlled randomization) and DOBSS (providing optimal policies) for Bayesian Stackelberg games. DOBSS is currently the fastest optimal procedure for Bayesian Stackelberg games and is in use by police at the Los Angeles International Airport(LAX) to schedule their activities.

Introduction

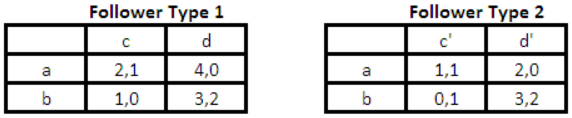
Many multiagent settings are appropriately modeled as Stackelberg games (Fudenberg & Tirole 1991; Paruchuri *et al.* 2007), where a leader commits to a strategy first, and then a follower selfishly optimizes its own reward, considering the action chosen by the leader. Stackelberg games are commonly used to model attacker-defender scenarios in security domains (Brown *et al.* 2006), as well as in patrolling (Paruchuri *et al.* 2007; 2008). For example, security personnel patrolling an infrastructure commit to a patrolling strategy first, before their adversaries act taking this committed strategy into account. Indeed, Stackelberg games are being used at the Los Angeles International Airport to schedule security checkpoints and canine patrols (Murr 2007; Paruchuri *et al.* 2008; Pita *et al.* 2008). They could potentially be used in many other situations such as network routing (Korilis, Lazar, & Orda 1997), pricing in transportation systems (Cardinal *et al.* 2005) and many others.

This paper focuses on determining the optimal strategy for a leader to commit to in a Bayesian Stackelberg game, i.e. a Stackelberg game where the leader may face multiple follower types. Such a Bayesian Stackelberg game may arise in a security domain because for example, when patrolling a region, a security robot may have uncertain knowledge about the different robber types it may face. Unfortunately, this problem of choosing an optimal strategy for the leader to commit to in a Bayesian Stackelberg game is NP-hard (Conitzer & Sandholm 2006). This result explains the computational difficulties encountered in solving such games.

In this paper, we present two of the fastest algorithms to solve Bayesian Stackelberg games published as full papers (Paruchuri *et al.* 2007; 2008). In particular, we present our approximate procedure named ASAP(Agent Security via Approximate Policies) published in AAMAS'07 and our exact method named DOBSS (Decomposed Optimal Bayesian Stackelberg Solver) published in AAMAS'08. ASAP provides policies with controlled randomization and hence are simple and easy to use to in practice but the approach turned out to be numerically unstable. DOBSS provides an efficient, exact solution for the Bayesian Stackelberg games while eliminating the numerical instabilities. Both these methods have three key advantages over earlier existing approaches: (a) Both the methods allow for a Bayesian game to be expressed compactly without requiring conversion to a normal-form game via the Harsanyi transformation described below. (b) Both these methods require only one mixed-integer linear program (MILP) to be solved, rather than a set of linear programs as in the Multiple-LPs method (Conitzer & Sandholm 2006), thus leading to a further performance improvement. (c) They directly search for an optimal leader strategy, rather than a Nash (or Bayes-Nash) equilibrium, thus allowing them to find high-reward non-equilibrium strategies (by exploiting the advantage of being the leader). DOBSS solves the Bayesian Stackelberg game in the ARMOR system deployed at the Los Angeles International Airport as mentioned above (Murr 2007).

Context and Overview

Stackelberg Game: In a Stackelberg game, a leader commits to a strategy first, and then a follower optimizes its reward, *considering the leader's action.* To see the advantage of being a leader in Stackelberg game, consider the game between the leader and follower type 1 as shown in Figure 1 (left). The leader is the row player and the follower types are column players. The only Nash equilibrium for this game is when the leader plays *a* and follower type 1 plays *c* which gives the leader a payoff of 2. However, if the leader commits to a mixed strategy of playing *a* and *b* with equal (0.5)



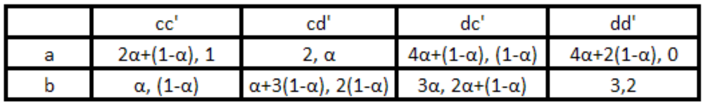
Follower Type 1

	c	d
a	2,1	4,0
b	1,0	3,2

Follower Type 2

	c'	d'
a	1,1	2,0
b	0,1	3,2

Figure 1: Payoff tables for a Bayesian Stackelberg game with 2 follower types.



	cc'	cd'	dc'	dd'
a	2α+(1-α), 1	2, α	4α+(1-α), (1-α)	4α+2(1-α), 0
b	α, (1-α)	α+3(1-α), 2(1-α)	3α, 2α+(1-α)	3,2

Figure 2: Harsanyi Transformed Payoff Table.

probability, then follower type 1 will play d, leading to a higher expected payoff of 3.5 for the leader. Note that in the Stackelberg game, follower knows the mixed strategy of the leader but not the actual action the leader takes in real time.

Bayesian Stackelberg Game: In a Bayesian game of N agents, each agent n must be one of a given set of types. For the two player Stackelberg game, inspired by the security domain of interest in this paper we assume that there is only *one leader type* (e.g. only one police force enforcing security), although there are multiple follower types (e.g. multiple types of adversaries), denoted by $l \in L$. There is an a priori probability p^l that a follower of type l will appear. Figure 1 shows such a game between a leader and two follower types leading to two payoff tables. Note that the leader does not know the follower's type. For each agent type (leader or follower) n, there is a set of strategies σ_n and a utility function $u_n : L \times \sigma_1 \times \sigma_2 \rightarrow \Re$. Our goal is *to find the optimal mixed strategy* for the leader given that the follower knows (has perfectly observed) this leader's strategy and chooses an optimal response to it.

Previous Work: Previous methods to solve a Bayesian Stackelberg game, first need the Bayesian game to be transformed into a normal-form game using Harsanyi transformation (Harsanyi & Selten 1972). Once this is done, techniques like the Multiple-LPs method for finding optimal strategies (Conitzer & Sandholm 2006) or the MIP-Nash technique to find the best Nash equilibrium (Sandholm, Gilpin, & Conitzer 2005), can find a strategy in the transformed game; this strategy from the transformed game can then be used back in the original Bayesian game. However, the compactness in structure of the Bayesian game is lost due to the Harsanyi transformation. In addition, since Nash equilibrium assumes a simultaneous choice of strategies, the advantages of being the leader are not considered. We now explain here the Harsanyi transformation.

Let us assume there are two follower types 1 and 2 as shown in Figure 1. Follower type 1 will be active with probability α, and follower type 2 will be active with probability $1 - \alpha$. Performing the Harsanyi transformation involves introducing a chance node, that determines the follower's type, thus transforming the leader's incomplete information regarding the follower into an imperfect information game. The transformed, normal-form game is shown in Figure 2. In the transformed game, the leader still has two strategies while there is a single follower type with four (2*2) strategies. For example, consider the situation in the transformed game where the leader takes action *a* and follower takes action *cc'*. The leader's payoff in the new game is calculated as a weighted sum of its payoffs from the two tables in Figure 1 i.e., α times payoff of leader when follower type 1 takes action c plus $(1 - \alpha)$ times payoff of leader when follower type 2 takes action *c'*. All the other entries in the new table, both for the leader and the follower, are derived in a similar fashion. In general, for n follower types with k strategies per follower type, the transformation results in a game with k^n strategies for the follower, thus causing an exponential blowup losing compactness.

Relationship to our AAAI'08 submission: Our AAAI'08 submission relaxes the assumption of all algorithms mentioned so far for Stackelberg games, including DOBSS and ASAP, that the follower acts optimally and has perfect observability. We present new algorithms that address uncertainty in follower actions due to their bounded rationality and observational uncertainty. Our AAAI'08 submission focuses on experimental results with human subjects: 800 games with 57 subjects. In contrast, our NECTAR paper describes DOBSS (algorithm in use at LAX), ASAP and the decomposition scheme that provides efficiency.

Exact Solution: DOBSS

We present here DOBSS (Paruchuri *et al.* 2008) first in its more intuitive form as a mixed-integer quadratic program (MIQP) and then show its linearization into an MILP. DOBSS finds the optimal mixed strategy for the leader while considering an *optimal* follower response for this leader strategy. Note that we need to consider only the reward-maximizing pure strategies of the followers, since if a mixed strategy is optimal for the follower, then so are all the pure strategies in the support of that mixed strategy. We denote by x the leader's policy, which consists of a vector of the leader's pure strategies. The value x_i is the proportion of times in which pure strategy i is used in the policy. For a follower type $l \in L$, q^l denotes its vector of strategies, and R^l and C^l the payoff matrices for the leader and the follower respectively, given this follower type l. Furthermore, X and Q denote the index sets of the leader and follower's pure strategies, respectively. Let M be a large positive number. Given *a priori* probabilities p^l, with $l \in L$, of facing each follower type, the leader solves the following problem:

$$\begin{array}{ll} \max_{x,q,a} & \displaystyle\sum_{i \in X} \sum_{l \in L} \sum_{j \in Q} p^l R^l_{ij} x_i q^l_j \\ \text{s.t.} & \sum_{i \in X} x_i = 1 \\ & \sum_{j \in Q} q^l_j = 1 \\ & 0 \leq (a^l - \sum_{i \in X} C^l_{ij} x_i) \leq (1 - q^l_j) M \\ & x_i \in [0 \ldots 1] \\ & q^l_j \in \{0, 1\} \\ & a^l \in \Re \end{array} \tag{1}$$

Where for a set of leader's actions x and actions q^l for each follower type, the objective represents the expected reward for the leader considering the *a-priori* distribution over follower types p^l. Constraints 1 and 4 define the set of feasible solutions x as probability distributions over the action set X. Constraints 2 and 5 limit the vector $q^l$of actions of follower type l to be a pure distribution over the set Q (i.e., each q^l has exactly one coordinate equal to one and the rest equal to zero). The two inequalities in constraint 3 ensure that $q^l_j = 1$ only for a strategy j that is optimal for follower type l. In particular, the leftmost inequality ensures that for all $j \in Q$, $a^l \geq \sum_{i \in X} C^l_{ij} x_i$, which means that given the leader's vector x, a^l is an upper bound on follower type l's reward for any action. The rightmost inequality is inactive for every action where $q^l_j = 0$, since M is a large positive quantity. For the action with $q^l_j = 1$ this inequality states that the follower's payoff for this action must be $\geq a^l$, which combined with previous inequality shows that this action must be optimal for follower type l.

Notice that Problem 1 is a decomposed MIQP in the sense that it does not utilize a full-blown Harsanyi transformation; instead it solves multiple smaller problems using individual adversaries payoffs (indexed by l) rather than a single, large, Harsanyi-transformed payoff. Furthermore, this decomposition does not cause any suboptimality (Paruchuri *et al.* 2008). We can now linearize the quadratic programming problem 1 through the change of variables $z^l_{ij} = x_i q^l_j$, thus obtaining the following equivalent MILP:

$$
\begin{array}{ll}
\max_{q,z,a} & \displaystyle\sum_{i \in X} \sum_{l \in L} \sum_{j \in Q} p^l R^l_{ij} z^l_{ij} \\
\text{s.t.} & \sum_{i \in X} \sum_{j \in Q} z^l_{ij} = 1 \\
& \sum_{j \in Q} z^l_{ij} \leq 1 \\
& q^l_j \leq \sum_{i \in X} z^l_{ij} \leq 1 \\
& \sum_{j \in Q} q^l_j = 1 \\
& 0 \leq (a^l - \sum_{i \in X} C^l_{ij} (\sum_{h \in Q} z^l_{ih})) \leq (1 - q^l_j) M \\
& \sum_{j \in Q} z^l_{ij} = \sum_{j \in Q} z^1_{ij} \\
& z^l_{ij} \in [0 \ldots 1] \\
& q^l_j \in \{0, 1\} \\
& a^l \in \Re
\end{array}
\tag{2}
$$

Proposition 1 *The DOBSS procedure exponentially reduces the problem over the Multiple-LPs approach in the number of adversary types (Paruchuri* et al. *2008).*

Approximate Solution: ASAP

We now present our limited randomization approach (Paruchuri *et al.* 2007), where 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. One advantage of such strategies is that they are compact to represent (as fractions) and simple to understand; therefore they can potentially be efficiently implemented in real patrolling applications. Thus for example, when k = 3, we can have a mixed strategy where strategy 1 is picked twice i.e., probability = 2/3 and strategy 2 is picked once with probability = 1/3. Unfortunately, while ASAP was designed to generate simple policies, our extensive experimental results surprisingly reveal that it suffers from problems of infeasibility. Thus, DOBSS remains the method of choice.

We now present our ASAP algorithm using the mathematical framework developed in the previous section. In particular we start with problem 1 and convert x from continuous to an integer variable that varies between 0 to k; thus obtaining the following problem:

$$
\begin{array}{ll}
\max_{x,q,a} & \displaystyle\sum_{i \in X} \sum_{l \in L} \sum_{j \in Q} \frac{p^l}{k} R^l_{ij} x_i q^l_j \\
\text{s.t.} & \sum_i x_i = k \\
& \sum_{j \in Q} q^l_j = 1 \\
& 0 \leq (a^l - \sum_{i \in X} \frac{1}{k} C^l_{ij} x_i) \leq (1 - q^l_j) M \\
& x_i \in \{0, 1, ..., k\} \\
& q^l_j \in \{0, 1\} \\
& a^l \in \Re
\end{array}
\tag{3}
$$

We then linearize problem (3) through the change of variables $z^l_{ij} = x_i q^l_j$, obtaining the following equivalent MILP:

$$
\begin{array}{ll}
\max_{q,z,a} & \displaystyle\sum_{i \in X} \sum_{l \in L} \sum_{j \in Q} \frac{p^l}{k} R^l_{ij} z^l_{ij} \\
\text{s.t.} & \sum_{i \in X} \sum_{j \in Q} z^l_{ij} = k \\
& \sum_{j \in Q} z^l_{ij} \leq k \\
& k q^l_j \leq \sum_{i \in X} z^l_{ij} \leq k \\
& \sum_{j \in Q} q^l_j = 1 \\
& 0 \leq (a^l - \sum_{i \in X} \frac{1}{k} C^l_{ij} (\sum_{h \in Q} z^l_{ih})) \leq (1 - q^l_j) M \\
& \sum_{j \in Q} z^l_{ij} = \sum_{j \in Q} z^1_{ij} \\
& z^l_{ij} \in \{0, 1, ..., k\} \\
& q^l_j \in \{0, 1\} \\
& a^l \in \Re
\end{array}
\tag{4}
$$

Experimental Results

Our first set of experiments provide scalability results for the four methods namely DOBSS, ASAP, Multiple-LPs (Conitzer & Sandholm 2006) and the MIP-Nash method (Sandholm, Gilpin, & Conitzer 2005). As mentioned earlier the latter two methods require transformation of a Bayesian game using Harsanyi transformation (Harsanyi & Selten 1972). We performed extensive experiments with several patrolling games. We present here results for two such games. The first game has a police patrolling 2 houses resulting in 2 strategies for the police and 2 for each of the adversary types. The second has the police patrolling 3 houses (patrol covers 2 of the 3 houses), resulting in 6 strategies for police and 3 strategies for each of the robber types. Further results on scalability are presented in (Paruchuri *et al.* 2008).

Figure 3 compares the runtime results of the four procedures for two and three houses. Each runtime value in the graph(s) corresponds to an average of twenty randomly generated scenarios. The x-axis shows the number of follower types the leader faces starting from 1 to 14 adversary types and the y-axis of the graph shows the runtime in seconds on logscale ranging from .01 to 10000 seconds. All the experiments that were not concluded in 30 minutes(1800 seconds) were cut off. Note that DOBSS provided the optimal solution while ASAP provided the best possible solution with randomization constraints. ASAP is numerically unstable

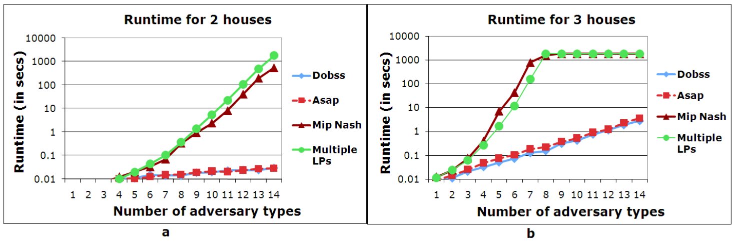


Figure 3: Runtimes for four algorithms on two domains.

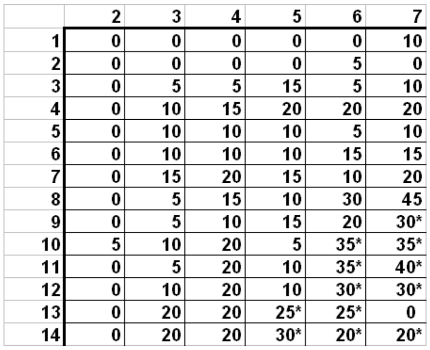
and sometimes incorrectly classifies solutions as infeasible; thus runtime results for ASAP are either time needed to find the solution or to classify the solution as infeasible.

Figure 3(a) shows the trends for all these four methods for the domain with two houses. The runtimes of DOBSS and ASAP are themselves exponential since they show a linear increase on a log-scale graph. Furthermore, they have an exponential speedup over the other two procedures as seen in the graph. Putting the result in numbers, MIP-Nash and Multiple-LPs needed about 1000s for solving the problem with fourteen adversary types while DOBSS and ASAP provided solutions in less than 0.1s. Similar trends are also noticed for the second domain of 3 houses where both MIP-Nash and Multiple-LPs could solve this problem only till seven adversary types within the 1800s cutoff time while DOBSS and ASAP could solve the problem for all fourteen adversary types modeled, under 10s. Between DOBSS and ASAP, DOBSS was found to have a 62% average speedup over ASAP(over all the experiments performed) i.e., ASAP needs 162secs for every 100secs that DOBSS takes.

Our second set of experimental results highlight the infeasibility issue of ASAP. We use the same settings as described above except that the number of houses was varied between two to seven (columns in the table). This means that the number of agent strategies varies between 2 to 42 (n*(n-1) where n is number of houses) while the number of strategies for each adversary type varies between 2 to 7 (n). The number of adversary types was varied between one to fourteen (rows in the table). For each fixed number of houses and follower types, twenty scenarios were randomly generated. Each number in the table represents the percentage of time ASAP classified the problems as infeasible. From the table in Figure 4, the general trend is that as the problem size increases ASAP tends to generate more infeasible solutions. We can calculate from the table that more than 12.5% of the solutions are infeasible for the five house problem when averaged over all the adversary scenarios. This number increases to as high as 18% and 20% on an average for the six and seven house problems, thus making the ASAP approach impractical for bigger problems. The values marked with a star are ones where ASAP ran out of time in many instances, and hence the percentage of infeasible solutions reported is an upper bound on the actual infeasible solutions.

Conclusion and Significance

Given the crucial importance of Bayesian Stackelberg games in many security applications, this paper introduces two of the fastest algorithms: ASAP (an approximate procedure) and DOBSS (an exact procedure). The exponential speedups these algorithms attain over previous algorithms are critically important in real applications. For our application at the Los Angeles Airport the leader has 784 actions and there may be up to 4 adversary types each with 8 actions. While DOBSS could solve the problem for all 4 adversary types within 80s, Multiple-LPs method could not solve for even 3 adversary types within the cutoff time of 20 minutes.



	2	3	4	5	6	7
1	0	0	0	0	0	10
2	0	0	0	0	5	0
3	0	5	5	15	5	10
4	0	10	15	20	20	20
5	0	10	10	10	5	10
6	0	10	10	10	15	15
7	0	15	20	15	10	20
8	0	5	15	10	30	45
9	0	5	10	15	20	30*
10	5	10	20	5	35*	35*
11	0	5	20	10	35*	40*
12	0	10	20	10	30*	30*
13	0	20	20	25*	25*	0
14	0	20	20	30*	20*	20*

Figure 4: % of infeasible solutions for ASAP. Rows represent 1-14 adversary types, columns represent 2-7 houses.

Acknowledgements: 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

Brown, G.; Carlyle, M.; Salmeron, J.; and Wood, K. 2006. Defending critical infrastructure. *Interfaces* 36(6):530–544.

Cardinal, J.; Labbé, M.; Langerman, S.; and Palop, B. 2005. Pricing of geometric transportation networks. In *17th Canadian Conference on Computational Geometry*.

Conitzer, V., and Sandholm, T. 2006. Computing the optimal strategy to commit to. In *EC*.

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

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.

Korilis, Y. A.; Lazar, A. A.; and Orda, A. 1997. Achieving network optima using stackelberg routing strategies. In *IEEE/ACM Transactions on Networking*.

Murr, A. 2007. Random checks. In *Newsweek National News: http://www.newsweek.com/id/43401*.

Paruchuri, P.; Pearce, J. P.; Tambe, M.; Ordonez, F.; and Kraus, S. 2007. An efficient heuristic approach for security against multiple adversaries. In *AAMAS*.

Paruchuri, P.; Pearce, J. P.; Marecki, J.; Tambe, M.; Ordonez, F.; and Kraus, S. 2008. Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. In *AAMAS*.

Pita, J.; Jain, M.; Marecki, J.; Ordonez, F.; Portway, C.; Tambe, M.; Western, C.; Paruchuri, P.; and Kraus, S. 2008. Deployed armor protection: The application of a game theoretic model for security at the los angeles international airport. In *AAMAS Industry Track*.

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