Solving Stackelberg Games with Uncertain Observability

Dmytro Korzhyk, Vincent Conitzer, Ronald Parr
Department of Computer Science, Duke University
Durham, NC 27708 USA
{dima,conitzer,parr}@cs.duke.edu

ABSTRACT

Recent applications of game theory in security domains use algorithms to solve a Stackelberg model, in which one player (the leader) first commits to a mixed strategy and then the other player (the follower) observes that strategy and best-responds to it. However, in real-world applications, it is hard to determine whether the follower is actually able to observe the leader's mixed strategy before acting.

In this paper, we model the uncertainty about whether the follower is able to observe the leader's strategy as part of the game (as proposed in the extended version of Yin et al. [17]). We describe an iterative algorithm for solving these games. This algorithm alternates between calling a Nash equilibrium solver and a Stackelberg solver as subroutines. We prove that the algorithm finds a solution in a finite number of steps and show empirically that it runs fast on games of reasonable size. We also discuss other properties of this methodology based on the experiments.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Algorithms, Economics, Theory

Keywords

game theory, Stackelberg, Nash, observability, strategy generation

1. INTRODUCTION

When multiple self-interested agents interact in the same domain, *game theory* provides a framework for reasoning about how each agent should act. One use of game theory is by an outside party that tries to predict the outcome of a strategic situation. For example, when we design a mechanism (e.g., an auction), we can use game theory to evaluate whether any given design will lead to good outcomes when the agents participating in it are strategic. Another use is by one of the agents in the game that wants to determine how to play. For example, game theory is often used to create poker-playing programs. Recently, algorithms for computing game-theoretic solutions have also started to find applications in security applications, where one of the players, the defender, tries to allocate limited defensive resources in anticipation of an attack by an attacker. Real-world examples include the placement of checkpoints and canine units at Los Angeles International airport [13] and the assignment of Federal Air Marshals to flights [15].

Probably the best-known solution concept in game theory is that of *Nash equilibrium*: a profile of mixed strategies, one for each player, is said to be in Nash equilibrium if no individual player can benefit from deviating. (A mixed strategy is a distribution over pure strategies; a pure strategy is a complete, deterministic plan of action.) Another possibility, especially in the context of an agent who is determining how to play in a game, is to compute a *Stackelberg* mixed strategy for the player. Such a strategy is an optimal solution when the player can commit to the mixed strategy before her opponent moves, so that the opponent will best-respond to the mixed strategy. This latter approach has various desirable properties, including the following. It avoids the *equilibrium selection problem* (if a game has multiple equilibria, which one should we play?). It leads to utilities for the committing player that are at least as high as, and sometimes higher than, what she would get in any Nash equilibrium (in fact, any correlated equilibrium [16]). Finally, in two-player normal-form games, there is a polynomial-time algorithm for computing a Stackelberg mixed strategy [3, 16], whereas computing a Nash equilibrium is PPAD-complete [5, 1, 2], and computing an (even approximately) optimal Nash equilibrium is NP-hard for just about any reasonable definition of optimality [6, 4].

We can illustrate the differences between these concepts using the example game shown in Figure 1. (We will use the same game as an example later in the paper.) This game has no pure-strategy Nash equilibrium. The unique mixed-strategy Nash equilibrium profile of this game is $\langle(0.5, 0.5), (0, 0.5, 0.5, 0)\rangle$.[1] The row player's utility from

[1]The equilibrium is unique because of the following. If the row player plays U with probability > 0.5, then only EL and L can be best responses for the column player, but then U cannot be a best response for the row player. By symmetry, the row player also cannot play D with probability > 0.5. Hence any equilibrium has the row player playing $(0.5, 0.5)$. Only L and R are best responses to this for the column player, and the only way to put probability on these to keep the row player indifferent between U and D is $(0, 0.5, 0.5, 0)$.

Cite as: Solving Stackelberg Games with Uncertain Observability, Dmytro Korzhyk, Vincent Conitzer, Ronald Parr, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1013-1020.

playing this equilibrium is 0.5. In contrast, in the Stackelberg model, the row player can commit to playing U, so that the column player best-responds with EL, which results in a utility of 9 for the row player. The row player can achieve an even higher utility by committing to a mixed strategy. If the row player commits to playing U with probability $8/9 + \epsilon$ and D with probability $1/9 - \epsilon$, the column player's best-response is still EL, and the row player's utility is approximately $9+1/9$. The Stackelberg solution is the limit as $\epsilon \to 0$. (Note that there are symmetric solutions on the other side of the game where the row player puts most of the probability on D and the column player responds with ER.)

	EL	L	R	ER
U	9,10	0,9	1,8	10,0
D	10,0	1,8	0,9	9,10

Figure 1: An example normal-form game.

Of course, playing a Stackelberg strategy seems to make little sense without some argument as to why the player should indeed be able to commit before her opponent moves. In the real-world security applications mentioned above, where Stackelberg strategies are indeed used, the argument is that the attacker (follower) can observe the defender (leader)'s actions over time, and thereby reconstruct the distribution, before attacking. This argument is not entirely uncontroversial: in many contexts, it is not clear that the follower can indeed observe the leader's mixed strategy. A recent study shows that a large class of security games has the property that any Stackelberg strategy is also a Nash equilibrium strategy (and moreover that there is no equilibrium selection problem) [17]. Nevertheless, this is known to not be true for other security games (as well as other non-security games, such as the example game that we just considered).

How should the leader agent play when she is not sure about the follower's ability to observe her mixed strategy, as is often the case in practice? One model that has been proposed in the extended version of Yin et al. [17] for this is to consider an extensive-form game where Nature makes a random move determining whether the leader's mixed strategy is observable or not, and then to find an equilibrium of this larger game. We will discuss this model in detail in Section 2. In this paper, we study properties of this model, present the first algorithm for solving these infinite-size extensive-form games, and evaluate it on random games. Our algorithm calls subroutines for solving Nash and Stackelberg problems; it works on arbitrary games (as long as the Nash and Stackelberg subroutines do).

2. REVIEW: EXTENSIVE-FORM GAME TO MODEL UNCERTAINTY ABOUT OBSERVABILITY

There are two players in the original game (represented in normal form): the leader and the follower. The leader's set of pure actions is A_l. The follower's set of pure actions is A_f. If the outcome of the game is (a_l, a_f), where $a_l \in A_l$ is the leader's action and $a_f \in A_f$ is the follower's action, then the leader's utility is $u_l(a_l, a_f)$, and the follower's utility is $u_f(a_l, a_f)$.

We now present the extensive-form game model introduced by Yin et al. (in the extended version of the paper [17]), which is arguably the most straightforward way to introduce uncertainty about the follower's ability to observe the leader's distribution over A_l. The extensive-form game proceeds as follows. First, Nature decides whether the follower will observe the leader's distribution or not. The probability that the follower observes the leader's distribution is p_{obs}; correspondingly, the probability that the follower does not observe it is $1-p_{\text{obs}}$. Then, the leader, without knowing Nature's choice, chooses a distribution over A_l. Next, the follower chooses a response $a_f \in A_f$, possibly after observing the distribution over A_l chosen by the leader if Nature has decided that the follower is able to observe. Finally, a_l is drawn from the leader's distribution; the leader's utility is $u_l(a_l, a_f)$, and the follower's utility is $u_f(a_l, a_f)$.

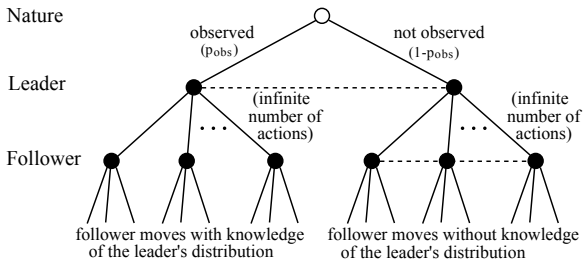

Figure 2: The extensive form of the game.

The extensive form of this game is shown in Figure 2. At the root, Nature makes a choice; at the next level, the leader chooses a distribution over A_l (note that there are infinitely many distributions to choose from—in particular, choosing a distribution is not the same as randomizing over which action to choose here); and at the next level, the follower chooses an action in A_f. Nodes that are in the same information set are connected with dashed lines. The two leader nodes are in the same information set because the leader does not observe Nature's decision. The follower's nodes in the right subtree are in the same information set, because the right subtree corresponds to the case where the follower does not observe the distribution.

It is important to emphasize that a *pure* strategy for the leader in this extensive-form game is a *distribution* over A_l; a mixed strategy for the leader is a distribution over such distributions. (In fact, we will show shortly that a distribution over distributions over A_l cannot be simplified to a distribution over A_l in this context.) A pure strategy for the follower specifies one action in A_f for every follower node on the left-hand side of the tree, plus one additional action for the follower's information set on the right-hand side of the tree. In fact, it is possible to simplify the left-hand side of the tree: we can take the follower's best action at each of his nodes on the left-hand side, and simply propagate the corresponding value up to that node as in backward induction.[2] (If there is a tie for the follower, he will break it in favor of the leader, to stay consistent with the Stackelberg model.) Thus we can eliminate the bottom level of the left-hand side of the tree, so that effectively a follower pure strategy in the extensive form consists of only a single action in A_f, corresponding to his action in the information set on the right-hand side.

[2]Note that we are just doing this at a conceptual level; we never actually write down this (infinite-sized) tree.

Since our goal is to solve an extensive-form game, a natural question is whether off-the-shelf extensive-form game solvers are sufficient for this. As we have pointed out, the leader's strategy space is infinite, preventing the direct application of standard methods. One way to address this is to discretize the leader's strategy space and obtain an approximate solution. Because this strategy space is an $(|A_l| - 1)$-simplex, discretizing it sufficiently finely is likely to lead to scalability issues. Our algorithm, in contrast, generates pure strategies for the leader in an informed way that results in an exact solution. Moreover, as we will see, experimentally our algorithm requires the generation of only very few strategies, so that there can be little doubt that this is preferable to the uninformed discretization approach.

3. EQUILIBRIA MAY REQUIRE RANDOMIZING OVER DISTRIBUTIONS

Because pure strategies for the leader in the extensive-form game are distributions over A_l, it follows that mixed strategies for the leader are distributions over distributions. However, one may be skeptical as to whether it is ever really necessary to randomize over distributions, rather than just simplifying the strategy back down to a single distribution. In this subsection, we show that for some games, randomizing over distributions is in fact necessary, in the sense that there is no equilibrium of the extensive-form game in which the leader plays a pure strategy.

Consider again the example game in Figure 1, whose Nash equilibrium and Stackelberg strategies we have already analyzed. Now consider the extensive-form variant of this game where the leader (row player)'s distribution is observed with probability $p_{\text{obs}} = .99$. Because the leader's distribution is almost always observed, it is suboptimal for the leader to put positive probability on any distribution that has probability strictly between 1/9 and 8/9 on U. This is because, when observed (which happens almost always), such distributions would incentivize the follower to play L or R, whereas any more extreme distributions will incentivize the follower to play EL or ER, leading to much higher utilities for the leader. (We recall that, upon observing the distribution, the follower is assumed to break ties in the leader's favor for technical reasons, though this is not essential for the example.)

It is also suboptimal to put positive probability on any distribution that puts strictly more than 8/9 probability on U. This is because, as long as the probability on U is at least 8/9, any unit of probability mass placed on D results in a utility of 10 rather than 9 in the .99 of cases where the follower observes; this outweighs any benefit that placing this unit of probability elsewhere might have in the .01 of cases where the follower does not observe. Similarly, putting positive probability on any distribution that puts strictly less than 1/9 probability on U is suboptimal. Hence, all of the leader's mass is either on the distribution $(8/9, 1/9)$ or on the distribution $(1/9, 8/9)$.

If the leader places all her mass on the distribution $(8/9, 1/9)$, the follower is incentivized to play EL all the time. However, if this is so, the leader has an incentive to deviate to $(1/9, 8/9)$. This is because this distribution will give her just as high a utility as $(8/9, 1/9)$ if it is observed (the follower will respond with ER); however, if it is not observed, the follower will not know that the leader has deviated and still play EL, and $(1/9, 8/9)$ gives a higher utility against EL than $(8/9, 1/9)$. Hence there is no equilibrium where the leader places all her mass on $(8/9, 1/9)$ (and, by symmetry, there is none where the leader places all her mass on $(1/9, 8/9)$). In fact, by similar reasoning as that used to establish the uniqueness of the Nash equilibrium of the original game, we can conclude that in equilibrium the leader must randomize uniformly between $(8/9, 1/9)$ and $(1/9, 8/9)$; the follower must then respond accordingly with EL or ER when he observes the distribution, and when he does not observe the distribution he must randomize uniformly between L and R (to keep the leader indifferent between her two distributions). Hence, this is the unique equilibrium.

4. THE ALGORITHM

We now present our algorithm for solving for an equilibrium of the extensive-form game (Figure 2). The intuition behind the algorithm is as follows. As we have already pointed out, after applying backward induction to the left-hand side of the extensive-form game, the follower's pure strategy space in the extensive-form game is simply A_f (corresponding to the action he takes on the right-hand side), which is manageable. What is not manageable is the space of all the leader pure strategies in the extensive form: there is one for every distribution over A_l, so there are infinitely many. This prevents us from simply writing down the normal-form game corresponding to the extensive-form game and solving that. (Note that this is *not* the same as the original normal-form game that has no uncertainty about observability.)

To address this, we start with a limited set of leader distributions (for example, the set of all $|A_l|$ degenerate distributions), and solve for a Nash equilibrium of this restricted game. This will give us a mixed strategy for the follower; the next step is to find the best leader pure strategy (distribution over A_l) in response to this follower mixed strategy. As we will see, technically, this corresponds to solving for a Stackelberg solution of an appropriately modified normal-form game. We then add the resulting distribution to the set of leader distributions, solve for a new equilibrium, etc., until convergence.

This type of strategy generation approach has been applied to solve various games where the strategy space is too large to write down [11, 7, 8]. (It has a close relation to the notion of constraint / column generation in linear programming.) Usually, this is because the strategy space is combinatorial—but it is finite, and hence the algorithm is guaranteed to converge eventually. In our case, however, there is a continuum of leader strategies, so we have to prove convergence, which we will do later.

Our algorithm for finding an equilibrium of the extensive-form game is shown in Figure 3. In this algorithm, $\mathcal{G}(D, A_f)$ is a normal-form game, more specifically it is the normal-form game corresponding to the extensive-form game, except that the leader can only choose from the distributions in D.

At any point, D is the set of distributions for the leader that we have generated so far. We find a mixed-strategy Nash equilibrium $\langle \mathbf{p}, \mathbf{q} \rangle$ of a normal-form game $\mathcal{G}$ in which the leader's set of pure strategies is D, the follower's set of pure strategies is A_f, and the players' utilities for the

$D \leftarrow$ *any finite non-empty set of distributions over* A_l
Loop:
  $\mathcal{G} \leftarrow \mathcal{G}(D, A_f)$
  $\langle \mathbf{p}, \mathbf{q} \rangle \leftarrow$ FIND-NE($\mathcal{G}$)
  $\mathbf{p}' \leftarrow$ LEADER-BR($\mathbf{q}$)
  If $u_l^{\mathcal{G}}(\mathbf{p}', \mathbf{q}) \leq u_l^{\mathcal{G}}(\mathbf{p}, \mathbf{q})$ **Then**
    Return $\langle \mathbf{p}, \mathbf{q} \rangle$
  Else
    $D \leftarrow D \cup \{\mathbf{p}'\}$

Figure 3: The algorithm.

outcome $(\mathbf{d}, a_f)$ are defined as follows.

$$u_l^{\mathcal{G}}(\mathbf{d}, a_f) = p_{\text{obs}} \mathbb{E}_{a_l \sim \mathbf{d}}[u_l(a_l, \text{FOLLOWER-BR}_{\text{obs}}(\mathbf{d}))] + (1 - p_{\text{obs}}) \mathbb{E}_{a_l \sim \mathbf{d}}[u_l(a_l, a_f)] \quad (1)$$

$$u_f^{\mathcal{G}}(\mathbf{d}, a_f) = p_{\text{obs}} \mathbb{E}_{a_l \sim \mathbf{d}}[u_f(a_l, \text{FOLLOWER-BR}_{\text{obs}}(\mathbf{d}))] + (1 - p_{\text{obs}}) \mathbb{E}_{a_l \sim \mathbf{d}}[u_f(a_l, a_f)] \quad (2)$$

Here $\mathbf{d} \in D$ is a distribution over A_l; a_l is the leader's action drawn according to $\mathbf{d}$; and $a_f \in A_f$ is the follower's action. u_l and u_f correspond to the utilities in the *original* normal-form game (that did not model uncertain observability). In each of these formulas, the first summand corresponds to the case where the follower observes the leader's chosen distribution over A_l, so that the follower best-responds to that distribution; the second summand corresponds to the case where the follower does not observe the leader's distribution over A_f, so that the follower will follow his strategy a_f for the right-hand side of the extensive-form game. The follower's best-response is computed as follows.

$$\text{FOLLOWER-BR}_{\text{obs}}(\mathbf{d}) \in \arg\max_{a_f \in A_f^*} \mathbb{E}_{a_l \sim \mathbf{d}}[u_l(a_l, a_f)]$$

$$A_f^* = \arg\max_{a_f \in A_f} \mathbb{E}_{a_l \sim \mathbf{d}}[u_f(a_l, a_f)]$$

That is, the follower maximizes his expected utility, breaking the ties in favor of the leader.[3]

We then check whether $\mathbf{p}$ is actually a best-response to $\mathbf{q}$ if the leader considers all possible distributions over A_l (we only know for sure that it is a best response among the restricted set D). To do that, we compute a best-response distribution $\mathbf{p}'$ over A_l that maximizes the leader's expected utility $u_d'(\mathbf{p}', \mathbf{q})$. If it turns out that $u_d'(\mathbf{p}', \mathbf{q})$ is equal to the leader's utility in the computed Nash equilibrium of the game, then it follows that $\mathbf{p}$ is a best response to $\mathbf{q}$, and because $\mathbf{q}$ is also a best response to $\mathbf{p}$, we can return $\langle \mathbf{p}, \mathbf{q} \rangle$ as an equilibrium of the extensive-form game with uncertain observability. Otherwise, we add distribution $\mathbf{p}'$ to D, and the algorithm continues on to the next iteration, in which we construct a new game $\mathcal{G}$, compute its Nash equilibrium, and so on.

In Subsection 4.1, we show how to compute the leader's best response LEADER-BR($\mathbf{q}$) efficiently using a set of linear programs (corresponding to a Stackelberg solve). In Subsection 4.2, we show how the algorithm solves the example game in Figure 1 with $p_{\text{obs}} = .99$. In Subsection 4.3, we show that the algorithm converges in a finite number of iterations.

[3]This is a common assumption in Stackelberg games; without it, it may happen that no solution exists. Specifically, if the original normal-form game is generic, then the follower breaks ties in the leader's favor in every subgame-perfect equilibrium of the regular Stackelberg extensive-form game [16].

4.1 Computing the leader's best response

In this section, we describe an efficient way to compute a distribution $\mathbf{p}'$ over the leader's actions A_l such that the leader's utility of playing $\mathbf{p}'$ is maximized assuming that the follower plays a given strategy $\mathbf{q}$. That is, $\mathbf{p}'$ is the leader's best response to the follower's mixed strategy $\mathbf{q}$, denoted by LEADER-BR($\mathbf{q}$) in the algorithm shown in Figure 3.

Our goal is to formulate LEADER-BR as a linear program. However, the leader's utility is not linear in $\mathbf{p}'$ in the case where the follower observes the leader's mixed strategy, because the leader's utility depends on the follower's best response to this observation, which can be different for different values of $\mathbf{p}'$. Hence, we use a trick that is also used in computing Stackelberg strategies (with certain observability) [3, 16]: we write an LP that maximizes the leader's expected utility under the constraint that the follower's best response in the observed case is a fixed action a_f^*. To find the leader's best response to $\mathbf{q}$ overall, we solve such an LP for each $a_f^* \in A_f$; we obtain a best response for the leader by choosing the optimal solution vector $\mathbf{p}'$ for an LP with the highest objective value (leader utility). Note that some of these LPs may be infeasible.

Specifically, given a_f^*, $\mathbf{q}$, we solve the following LP, whose variables are the p'_{a_l}.

$$\text{Maximize } p_{\text{obs}} \sum_{a_l \in A_l} p'_{a_l} u_l(a_l, a_f^*) + (1 - p_{\text{obs}}) \sum_{a_l \in A_l} \sum_{a_f \in A_f} p'_{a_l} q_{a_f} u_l(a_l, a_f)$$

Subject to

$$\forall a_f \in A_f : \sum_{a_l \in A_l} p'_{a_l} u_f(a_l, a_f^*) \geq \sum_{a_l \in A_l} p'_{a_l} u_f(a_l, a_f)$$

$$\sum_{a_l \in A_l} p'_{a_l} = 1$$

$$\forall a_l \in A_l : p'_{a_l} \geq 0$$

This formulation is almost identical to the standard one for solving for a Stackelberg strategy [3, 16], except the objective is different to account for the fact that the follower may not observe the distribution. In fact, if we modify the leader's utility function to $u_l^{\mathbf{q}}(a_l, a_f^*) = p_{\text{obs}} u_l(a_l, a_f^*) + (1 - p_{\text{obs}}) \sum_{a_f \in A_f} q_{a_f} u_l(a_l, a_f)$, then the objective simplifies to $\sum_{a_l \in A_l} p'_{a_l} u_l^{\mathbf{q}}(a_l, a_f^*)$, and we obtain the standard Stackelberg formulation. Hence, we are just doing a Stackelberg solve on a modified game.

4.2 An example run of the algorithm

In this section, we demonstrate how the algorithm computes an equilibrium of the uncertain-observability extensive-form game for the payoff matrix shown in Figure 1, with probability of observability $p_{\text{obs}} = 0.99$. (We already solved for the equilibrium of this game analytically in Section 3—the purpose here is to show how the algorithm finds this equilibrium.) In this game, there are two actions in A_l, so each leader distribution is represented by a vector of two numbers summing to 1.

Initialization. We initialize the set of leader distributions with the two degenerate distributions over A_l: the distri-

bution $(1,0)$ corresponds to the leader always playing U, and the distribution $(0,1)$ corresponds to the leader always playing D. The normal-form game for the current set of distributions $D = \{(1,0),(0,1)\}$ and the utilities $u_l^{\mathcal{G}}, u_f^{\mathcal{G}}$ computed according to Equations (1), (2) is shown in Figure 4. (Note that the follower strategy has very little effect on the expected payoffs in this game; this is because the follower strategy only concerns the "unobserved" part of the game, which occurs very rarely in this game. The "observed" part has been preprocessed with backward induction.)

	EL	L	R	ER
(1,0)	9,10	8.91, 9.99	8.92, 9.98	9.01, 9.9
(0,1)	9.01, 9.9	8.92, 9.98	8.91, 9.99	9,10

Figure 4: The normal-form game after the initialization.

Iteration 1. We first compute a Nash equilibrium of the normal-form game shown in Figure 4, namely, $\langle(.5,.5),(0,.5,.5,0)\rangle$. Next, we compute the leader's best response to the follower's mixed strategy $(0,.5,.5,0)$. This results in the distribution s_1, in which the leader plays U with probability 8/9 and D with probability 1/9, so that the follower's best response to s_1 is EL.

$$s_1 = (8/9)U + (1/9)D$$

It turns out that the leader's utility from playing s_1 against the follower's mixed strategy $(0,.5,.5,0)$ is higher than the leader's utility in the current NE profile $\langle(.5,.5),(0,.5,.5,0)\rangle$. Thus, we add s_1 to D. The resulting normal-form game is shown in Figure 5.

	EL	L	R	ER
(1,0)	9,10	8.91, 9.99	8.92, 9.98	9.01, 9.9
(0,1)	9.01, 9.9	8.92, 9.98	8.91, 9.99	9,10
s_1	9.11, 8.89	9.02, 8.89	9.03, 8.88	9.12, 8.81

Figure 5: The normal-form game after the first iteration.

Iteration 2. We compute a Nash equilibrium of the game shown in Figure 5, namely, the pure-strategy Nash equilibrium $\langle s_1, \mathrm{L}\rangle$. The leader's best response to the follower's strategy L is s_2, where

$$s_2 = (1/9)U + (8/9)D$$

The leader's utility from playing s_2 against L is higher than the leader's utility from playing s_1 against L. Thus, we add s_2 to the set D. The resulting normal-form game is shown in Figure 6.

	EL	L	R	ER
(1,0)	9,10	8.91, 9.99	8.92, 9.98	9.01, 9.9
(0,1)	9.01, 9.9	8.92, 9.98	8.91, 9.99	9,10
s_1	9.11, 8.89	9.02, 8.89	9.03, 8.88	9.12, 8.81
s_2	9.12, 8.81	9.03, 8.88	9.02, 8.89	9.11, 8.89

Figure 6: The normal-form game after the second iteration.

Iteration 3. We compute a mixed-strategy Nash equilibrium of the normal-form game shown in Figure 6, namely, $\langle(0,0,.5,.5),(0,.5,.5,0)\rangle$. When we compute the leader's best-response to the follower's mixed strategy $(0,.5,.5,0)$, it turns out that there is no distribution that gives the leader a utility higher than the leader's utility in the computed NE profile. Thus we have found an equilibrium of the uncertain-observability extensive-form game, in which the leader plays s_1 with probability .5 and s_2 with probability .5, while the follower plays L with probability .5 and R with probability .5.

4.3 A bound on the number of iterations

In this section, we prove that the algorithm is guaranteed to find an equilibrium of the extensive-form game in a finite number of iterations. For each a_f, the set of leader mixed strategies S_{a_f} to which a_f is a best response is a polytope in $\mathbb{R}^{|A_l|}$. Denote the number of vertices of S_{a_f} by $v(S_{a_f})$. Typical linear program solvers will return a vertex of the feasible region; we will assume that we use such a solver. Then, the number of iterations of our algorithm can be bounded as follows.

THEOREM 1. *The algorithm finds an equilibrium of the extensive-form game modeling uncertain observability in no more than* $1 + \sum_{a_f \in A_f} v(S_{a_f})$ *iterations.*

PROOF. LEADER-BR returns the optimal solution to one of the linear programs in Subsection 4.1. The feasible region of each of these linear programs is one of the regions S_{a_f}. Hence, by the assumption on our LP solver, LEADER-BR always returns a vertex of such a region.

When we generate a vertex corresponding to a distribution that is already in D, we have converged: this vertex cannot be a better response to $\mathbf{q}$ than $\mathbf{p}$, because $\mathbf{p}$ is a best response to $\mathbf{q}$ among distributions in D. Because there are at most $\sum_{a_f \in A_f} v(S_{a_f})$ distinct vertices to generate, the bound on the number of iterations follows. □

5. A STRONGER BOUND ON THE LEADER'S SUPPORT SIZE

Theorem 1 implies that there always exists an equilibrium in which the leader randomizes over at most $1 + \sum_{a_f \in A_f} v(S_{a_f})$ distributions. This is still a rather loose bound. The following theorem establishes a much tighter bound.

THEOREM 2. *In any uncertain-observability extensive-form game, there exists an equilibrium in which the number of distributions on which the leader places positive probability is at most* $|A_l|$.

PROOF. Let d denote a distribution over leader actions, where $d(a_l)$ denotes the probability d places on leader action $a_l \in A_l$. Suppose there is an equilibrium of the whole game with p_d denoting the leader probability on distribution d, and q_{a_f} denoting the follower probability on follower action a_f (conditional on the follower not being able to observe). Let $\pi(a_l) = \sum_d p(d)d(a_l)$ be the marginal probability that the leader plays a_l. Finally, let $u_l^s(d)$ denote the utility that the leader would get for committing to d in a pure Stackelberg version of the game (corresponding to the "observed" side of the game tree). Then, consider the following linear program whose variables are p'_d (one for every distribution d in the support of p_d). (This LP is just for the purpose of

analysis.)

$$\text{Maximize} \sum_d p'_d u_l^s(d)$$

$$\text{Subject to}$$

$$(\forall a_l) \sum_d p'_d d(a_l) = \pi(a_l)$$

$$(\forall d)\ p'_d \geq 0$$

That is, this linear program tries to modify the leader's equilibrium strategy to maximize the leader's overall Stackelberg utility (the utility on the "observed" side of the game tree) under the constraint that the marginal probabilities do not change (so that nothing changes on the "unobserved" side of the tree).

The original equilibrium p_d must be an optimal solution to this LP, because, if we suppose to the contrary that there is a better solution, then the leader would want to switch to that better solution (it would not change her utility on the "unobserved" side and it would improve it on the "observed" side), contradicting the equilibrium assumption. In fact, any optimal solution to this linear program must be an equilibrium when combined with the q_{a_f}, because it will do just as well as p_d for the leader, and the follower will still be best-responding (on the "unobserved" side) because the marginal probabilities on the a_l remain the same. A linear program with $|A_l|$ constraints (not counting the nonnegativity constraints for each variable) must have an optimal solution with at most $|A_l|$ of its variables set to nonzero values (which follows, for example, from the simplex algorithm). It follows that there exists an equilibrium where the leader places positive probability on at most $|A_l|$ distributions. □

6. EXPERIMENTS

The goal of our experiments is to study a number of properties of the proposed algorithm and the solutions it generates. Since the bound on the number of iterations given in Theorem 1 is quite loose, we want to measure the number of iterations and the overall run time of the algorithm for different payoff matrices and values of p_{obs}. Another goal of the experiments is to measure the leader's support size, that is, the number of distributions played with positive probability in the leader's equilibrium strategy, which we showed to be bounded by the number of the leader's actions $|A_l|$ (Theorem 2). We also want to study the dependence of the leader's equilibrium utility on the probability of observability p_{obs}. Finally, we want to find out how often the leader's equilibrium strategy in the extensive-form game is actually different from Nash and Stackelberg strategies in the original normal-form game.

In our experimental results we consider 15×15 payoff matrices and vary p_{obs}. We used two different Nash equilibrium solvers, a MIP solver with different objectives [14], and the Gambit [10] implementation of the Lemke-Howson algorithm [9]. For the MIP solver, we used three different objective functions: no objective, minimizing the size of the leader support, and maximizing the leader utility.

We considered two distributions over games. The first distribution (*uniform*) generated payoff matrices with individual payoffs drawn uniformly at random from $[0, 1]$. The second (*gamut*) generated payoff matrices from the various game types offered in GAMUT [12], with uniform weight given to each type.

Figures 7(a) and 7(b) show the run time of the different algorithms as a function of p_{obs}. One general trend is that the MIP solver that minimizes the leader support is the fastest solver. One interesting difference is that run time generally increases with p_{obs} for the GAMUT distribution, but is fairly flat or decreasing for uniform. The short run time is due to the low number of iterations, which we discuss next.

Figures 7(c) and 7(d) show the number of iterations taken by the algorithm. Each iteration corresponds to a complete pass through the loop in Figure 3, which includes a Nash equilibrium computation in the extensive form game followed by a LEADER-BR solve. The number of iterations generally tracks the run time fairly closely. Two exceptions are GAMBIT and MIP with leader support minimization for the GAMUT distribution. As we can see, the number of iterations is surprisingly low compared to our theoretical bound of Theorem 1. We leave the question of whether a tighter theoretical bound on the number of iterations can be obtained for future research.

The support size (number of distributions over which the leader randomizes in the equilibrium) is shown in Figures 7(e) and 7(f). The small support size is explained in part by the low number of iterations. Since we initialize the algorithm with $|A_l|$ pure strategies for the leader, the leader's support size cannot be larger than $|A_l|$ plus the number of iterations. However, it is significantly lower than that bound.

Figures 7(g) and 7(h) show the leader's expected utility in the equilibrium. As expected, higher values of p_{obs} lead to higher utility for the leader—this is the benefit of commitment. Using the MIP that maximizes leader utility (within a single Nash solve) tends to lead to high leader utilities in the final equilibrium, but intriguingly the MIP with no objective surpasses it for the GAMUT games.

Finally, Figures 7(i) and 7(j) show how often the leader's equilibrium strategy coincided with Stackelberg (full observability) or Nash (no observability) strategies of the game. The Nash subroutine that is used by the algorithm here is the MIP formulation that minimizes the support size. Naturally, the higher the value of p_{obs} is, the more often the equilibrium strategy coincides with Stackelberg and the less often it coincides with Nash. In general, it coincides with Nash very often and with Stackelberg quite often. We can also see that the equilibrium strategy concides with both Nash and Stackelberg at the same time in a high percentage of GAMUT games. This indicates that in certain game families, simply playing a Nash/Stackelberg strategy of the original normal-form game is also an equilibrium strategy in the extensive-form game with uncertain observability across intervals of p_{obs}. However, this is not the case in games with uniformly random payoffs, which suggests the need for an algorithm like the one we present in this paper.

The main lessons that we take away from this set of experiments are as follows. First, our proposed algorithm is quite fast in practice, especially compared to the loose theoretical bound on the number of iterations that we established in Theorem 1. Second, there are games in which the defender's equilibrium strategy is sensitive to the value of p_{obs}, which suggests that it is important to model the uncertainty about the observability. Third, there are families of games in which the equilibrium does not change across wide intervals of p_{obs}—in such cases, playing Nash or Stackelberg strategies of the original normal-form game may be

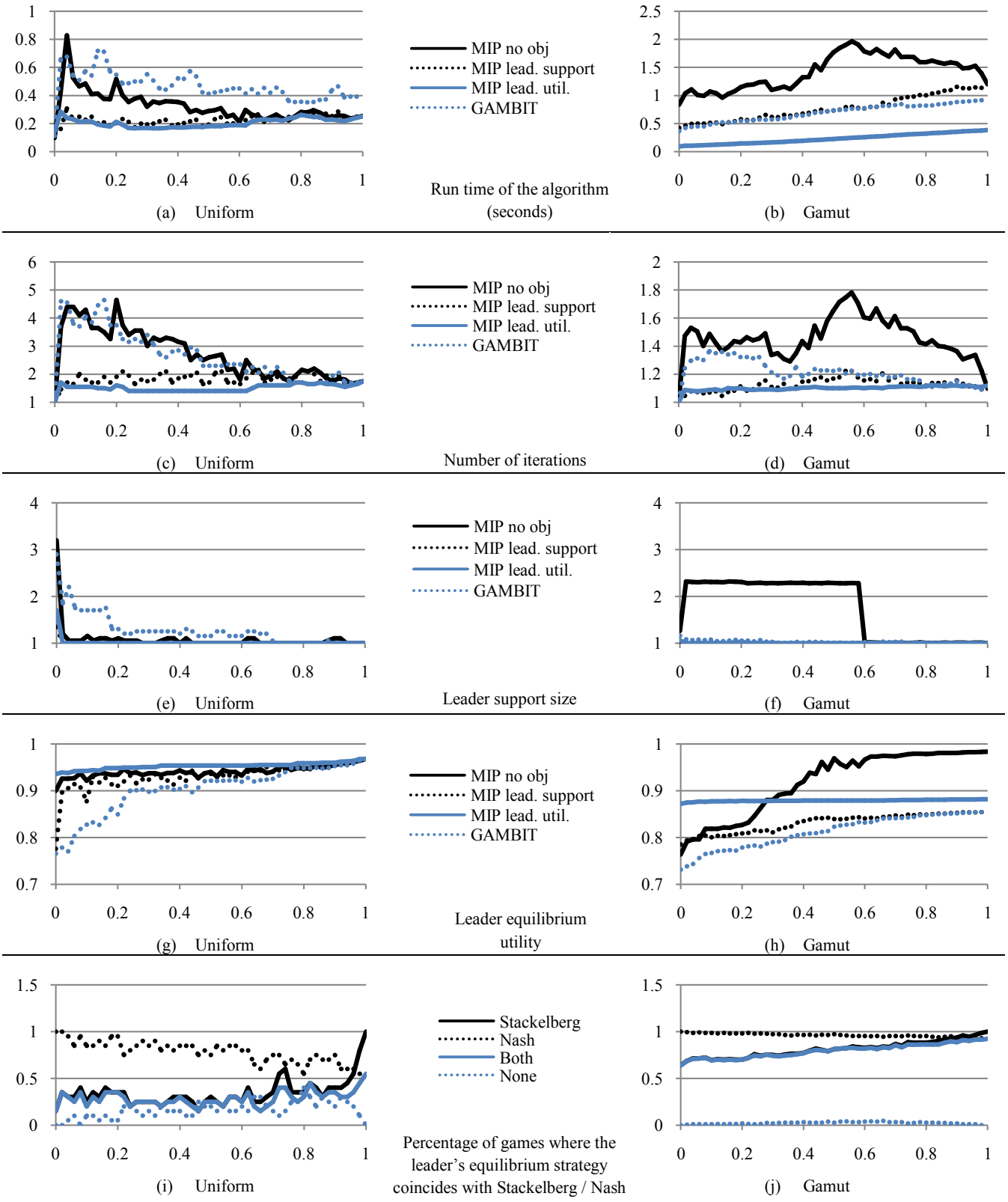

Figure 7: Experimental results

"good enough".

7. CONCLUSION

Several recently deployed applications in security domains use game theory for the strategic allocation of defensive resources. These applications compute a Stackelberg strategy rather than a Nash strategy. For this to make sense, the follower needs to be able to observe the leader's distribution; however, in many applications, there is some uncertainty about whether the follower has this ability. One previously proposed solution to this dilemma is to model model this uncertainty explicitly as a move by Nature in an extensive-form game of infinite size. We pursued this approach in this paper, and proposed an iterative algorithm for computing an equilibrium of the extensive-form game. The algorithm alternately calls subroutines for computing Nash and Stackelberg solutions, and is guaranteed to terminate in finite time. In experiments, the algorithm required very few iterations to compute an equilibrium. While we proved the perhaps unintuitive property that in some of these games, the leader must randomize over distributions in equilibrium, this happened very rarely in the experiments. We also proved an upper bound on the number of distributions in the leader's support, though this bound is still well above what we typically see in the experiments.

We believe that our algorithm constitutes a useful addition to the toolbox of techniques for computing game-theoretic solutions, especially in ambiguous real-world domains. Strengths of the algorithm include that it can be applied to any game (as opposed to, for instance, just security games), and it can also use as subroutines Nash and Stackelberg solvers that are tailored to particular game families. The algorithm is efficient in practice, and is guaranteed to produce a solution with support no larger than the number of actions in the original game despite solving an extensive form game with a potentially infinite branching factor.

A potential drawback to the overall framework, not the algorithm, is that it requires us to determine the number p_{obs}. This may not be an issue insofar as the solution stays the same across a range of values of p_{obs}, yet many open problems remain. As p_{obs} shrinks, we are more likely to encounter equilibrium selection problems—how do we address these? What happens if we have some degree of control over p_{obs}? Are there other ways of addressing the problem of uncertainty about observability that do not involve making the uncertainty explicit in the extensive form?

8. ACKNOWLEDGMENTS

We acknowledge NSF CAREER 0953756 and IIS-0812113, ARO 56698-CI, and an Alfred P. Sloan fellowship for support. We also thank Christopher Kiekintveld, Milind Tambe, and Zhengyu Yin for useful discussions about this topic.

9. REFERENCES

[1] X. Chen and X. Deng. Settling the complexity of two-player Nash equilibrium. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 261–272, 2006.

[2] X. Chen, X. Deng, and S.-H. Teng. Computing Nash equilibria: Approximation and smoothed complexity. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 603–612, 2006.

[3] V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pages 82–90, Ann Arbor, MI, USA, 2006.

[4] V. Conitzer and T. Sandholm. New complexity results about Nash equilibria. *Games and Economic Behavior*, 63(2):621–641, 2008. Earlier versions appeared in IJCAI-03 and as technical report CMU-CS-02-135.

[5] C. Daskalakis, P. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, pages 71–78, 2006.

[6] I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1:80–93, 1989.

[7] E. Halvorson, V. Conitzer, and R. Parr. Multi-step multi-sensor hider-seeker games. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 159–166, Pasadena, CA, USA, 2009.

[8] M. Jain, E. Kardes, C. Kiekintveld, F. Ordóñez, and M. Tambe. Security games with arbitrary schedules: A branch and price approach. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Atlanta, GA, USA, 2010.

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

[10] R. D. McKelvey, A. M. McLennan, and T. L. Turocy. Gambit: Software tools for game theory, version 0.97.1.5, 2004.

[11] H. B. McMahan, G. J. Gordon, and A. Blum. Planning in the presence of cost functions controlled by an adversary. In *International Conference on Machine Learning (ICML)*, pages 536–543, Washington, DC, USA, 2003.

[12] E. Nudelman, J. Wortman, K. Leyton-Brown, and Y. Shoham. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, New York, NY, USA, 2004.

[13] J. Pita, M. Jain, F. Ordóñez, C. Portway, M. Tambe, and C. Western. Using game theory for Los Angeles airport security. *AI Magazine*, 30(1):43–57, 2009.

[14] T. Sandholm, A. Gilpin, and V. Conitzer. Mixed-integer programming methods for finding Nash equilibria. In *AAAI*, pages 495–501, Pittsburgh, PA, USA, 2005.

[15] J. Tsai, S. Rathi, C. Kiekintveld, F. Ordonez, and M. Tambe. IRIS - a tool for strategic security allocation in transportation networks. In *AAMAS — Industry Track*, 2009.

[16] B. von Stengel and S. Zamir. Leadership games with convex strategy sets. *Games and Economic Behavior*, 69:446–457, 2010.

[17] Z. Yin, D. Korzhyk, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. Nash in security games: Interchangeability, equivalence, and uniqueness. In *AAMAS*, pages 1139–1146, Toronto, Canada, 2010.