Complexity of Computing Optimal Stackelberg Strategies in Security Resource Allocation Games

Dmytro Korzhyk, Vincent Conitzer, Ronald Parr

Duke University, Durham, NC 27708, USA

{dima, conitzer, parr}@cs.duke.edu

Abstract

Recently, algorithms for computing game-theoretic solutions have been deployed in real-world security applications, such as the placement of checkpoints and canine units at Los Angeles International Airport. These algorithms assume that the defender (security personnel) can commit to a mixed strategy, a so-called *Stackelberg* model. As pointed out by Kiekintveld et al. (2009), in these applications, generally, multiple resources need to be assigned to multiple targets, resulting in an exponential number of pure strategies for the defender. In this paper, we study how to compute optimal Stackelberg strategies in such games, showing that this can be done in polynomial time in some cases, and is NP-hard in others.

Introduction

In settings with multiple self-interested agents, the optimal action for one agent to take generally depends on what other agents do. *Game theory* provides various solution concepts, which specify what it means to act optimally in such a domain. As a result, there has been much interest in the multiagent systems community in the design of algorithms for computing game-theoretic solutions. Most of this work has focused on computing *Nash equilibria*. A Nash equilibrium consists of a profile of strategies (one for each player) such that no player individually wants to deviate; strategies are allowed to be *mixed*, that is, randomizations over *pure* strategies. This concept has some appealing properties, including that every finite game has at least one Nash equilibrium (Nash 1950). Unfortunately, from a computational perspective, Nash equilibrium is a more cumbersome concept: it is PPAD-complete to find even one Nash equilibrium, even in two-player games (Daskalakis, Goldberg, and Papadimitriou 2006; Chen and Deng 2006). The optimal equilibrium (for any reasonable definition of optimal) is NP-hard to find (or even to approximate), even in two-player games (Gilboa and Zemel 1989; Conitzer and Sandholm 2008).

An alternative solution concept (for two-player games) is the following. Suppose that player one (the *leader*) is able to *commit* to a mixed strategy; then, player two (the *follower*) observes this commitment, and chooses a response. Such a commitment model is known as a *Stackelberg* model, and

	L	R
U	2,1	4,0
D	1,0	3,1

Figure 1: Example game where commitment helps.

we will refer to an optimal mixed strategy for player one to commit to as an (optimal) Stackelberg strategy. For example, consider the game given in normal form in Figure 1. This game has a unique Nash equilibrium, (U, L) (the game is solvable by iterated strict dominance). However, if player one (the row player) can commit, then she is better off committing to playing D, which incentivizes player two to play R, resulting in a utility of 3 for player one. It is even better for player one to commit to the mixed strategy 49% U, 51% D, which still incentivizes player two to play R, so that player one gets an expected utility of 3.49. Of course, it is even better to commit to 49.9% U, 50.1% D—*etc.* At the limit strategy of 50% U, 50% D, player two is indifferent between L and R. In this case, we assume player two breaks ties in player one's favor (plays R), so that we have a well-defined Stackelberg strategy (50% U, 50% D). In two-player *zero-sum* games, Nash equilibrium strategies and Stackelberg strategies both coincide with minimax strategies (and, hence, with each other), due to von Neumann's minimax theorem (von Neumann 1927).

Interestingly, for a two-player normal-form game (not necessarily zero-sum), the optimal Stackelberg strategy can be found in polynomial time, using a set of linear programs (Conitzer and Sandholm 2006).[1] Besides this computational benefit over Nash equilibrium, with Stackelberg strategies there is effectively no *equilibrium selection problem* (the problem that if there are multiple equilibria, it is not clear according to which one to play).

The computation of Stackelberg strategies has recently found some real-world applications in security domains. In

---

[1] The same algorithm appears in a recent paper by von Stengel and Zamir (2009). It is not known whether linear programs are solvable in *strongly* polynomial time, that is, with no dependence on the sizes of the input numbers at all. Consequently, it is not known whether any of the problems in this paper can be solved in strongly polynomial time.

these games, the defender (security personnel) places security resources (*e.g.*, guards) at various potential targets (possibly in a randomized manner), and then the attacker chooses a target to attack. The defender takes the role of the leader. Los Angeles International Airport now uses an algorithm for computing Stackelberg strategies to place checkpoints and canine units randomly (Paruchuri et al. 2008; Pita et al. 2009).

However, as was pointed out by Kiekintveld et al. (2009), the applicability of these techniques to security domains is limited by the fact that the defender generally has exponentially many pure strategies, so that it is not feasible to write out the entire normal form of the game. Specifically, if there are m indistinguishable defensive resources, and n targets to which they can be assigned ($n > m$), then there are $\binom{n}{m}$ pure strategies (allocations) for the defender. Kiekintveld et al. point out that while the LAX application was small enough to enumerate all strategies, this is not the case for new applications, including the problem of assigning Federal Air Marshals to flights (Tsai et al. 2009). They provide a nice framework for representing this type of problem; we follow this framework in this paper (and review it in the following section). However, their paper leaves open the computational complexity of finding the optimal Stackelberg strategy in their framework. In this paper, we resolve the complexity of all the major variants in their framework, in some cases giving polynomial-time algorithms, and in other cases giving NP-hardness results.

Problem Description and Notation

Following Kiekintveld et al. (2009), we consider the following two-player general-sum game. Player one (the "leader" or "defender") commits to a mixed strategy to allocate a set of resources to defend a set of targets.[2] Player two (the "follower" or "attacker") observes the commitment and then picks one of the targets to attack. The utilities of the players depend on which target was attacked and whether that target was defended.

We will consider several variants of this game. Resources of the leader can be either homogeneous, or there can be several types of resources, each with different limitations on what they can defend. It can either be the case that a resource can be assigned to at most one target, or it can be the case that a resource can be assigned to a subset of the targets (such a subset is called a *schedule*). As we will see, the complexity depends on the size of these schedules.

We will use the following notation to describe different variants of the problem.

- **Targets.** Described by a set T ($|T| = n$). A target t is *covered* if there is a resource assigned to t (in the case of no schedules), or if a resource is assigned to a schedule that includes t.

- **Schedules.** Described by a collection of subsets of targets $S \subseteq 2^T$. Here, $s \in S$ is a subset of targets that can

---

be simultaneously covered by some resource. We assume that any subset of a schedule is also a schedule, that is, if $s' \subseteq s$ and $s \in S$, then $s' \in S$. When resources are assigned to individual targets, we have (by a slight abuse of notation) $S = T \cup \{\emptyset\}$, where $\emptyset$ corresponds to not covering any target.

- **Resources.** Described by a set Ω ($|\Omega| = m$). When there are different types of resources, there is a function $A : \Omega \rightarrow 2^S$, where $A(\omega)$ is the set of schedules to which resource ω can be assigned. We assume that if $s' \subseteq s$ and $s \in A(\omega)$, then $s' \in A(\omega)$—that is, if a resource can cover a set of targets simultaneously, then it can also cover any subset of that set of targets simultaneously. If resources are homogeneous, then we assume every resource can cover all schedules, that is, $A(\omega) = S$ for all $\omega \in \Omega$.

- **Utility functions.** If target t is attacked, the defender's utility is $U_d^c(t)$ if t is covered, or $U_d^u(t)$ if t is not covered. The attacker's utility is $U_a^c(t)$ if t is covered, or $U_a^u(t)$ if t is not covered. We assume $U_d^c(t) \geq U_d^u(t)$ and $U_a^c(t) \leq U_a^u(t)$. We note that it makes no difference to the players' utilities whether a target is covered by one resource or by more than one resource.

LP notation. We will use linear programs in all of our positive results (polynomial-time algorithms). We now describe some of the variables used in these linear programs.

- c_t is the probability of target t being covered.

- c_s is the probability of schedule s being covered.

- $c_{\omega,s}$ is the probability of resource ω being assigned to schedule s.

Let $\mathbf{c}$ denote the vector of probabilities $(c_1, \ldots, c_n)$. Then, the utilities of the leader and the follower can be computed as follows, given $\mathbf{c}$ and the target t being attacked:

$$U_d(t, \mathbf{c}) = c_t U_d^c(t) + (1 - c_t) U_d^u(t)$$
$$U_a(t, \mathbf{c}) = c_t U_a^c(t) + (1 - c_t) U_a^u(t)$$

These equalities are implicit in all of our linear programs and, for brevity, are not repeated.

Standard multiple LPs approach. As a benchmark and to illustrate some of the ideas, we first describe the standard algorithm for computing a Stackelberg strategy in two-player normal-form games (Conitzer and Sandholm 2006) in our notation. This approach creates a separate LP for every follower pure strategy—i.e., one for every target t^*. This LP solves for the optimal leader strategy *under the constraint that the follower's best response is t^**. Once we have solved all these n LPs, we compare the n resulting leader strategies and choose the one that is best for the leader; this one must then be optimal overall (without any constraint on which strategy is the best response). The LP for t^* is structured as follows. Create a variable for every leader pure strategy (allocation of resources to schedules) α, representing the probability that the leader puts on that strategy; and a constraint for every follower pure strategy (target) t, representing the best-response constraint that the follower should not

be better off playing t than t^*.

$$\text{maximize } \sum_{\alpha} p_\alpha U_d(\alpha, t^*)$$

subject to

$$\forall t \in T : \sum_{\alpha} p_\alpha U_a(\alpha, t) \leq \sum_{\alpha} p_\alpha U_a(\alpha, t^*)$$

$$\sum_{\alpha} p_\alpha = 1$$

In this paper, we will also follow the approach of solving a separate LP for every t^* and then comparing the resulting solutions, though our individual LPs will be different or handled differently.

Heterogeneous Resources, Singleton Schedules

We first consider the case in which schedules have size 1 or 0 (that is, resources are assigned to individual targets or not at all, so that $S = T \cup \{\emptyset\}$. We show that here, we can find an optimal strategy for the leader in polynomial time. Kiekintveld et al. (2009) gave a mixed-integer program formulation for this problem, and proved that feasible solutions for this program correspond to mixed strategies in the game. However, they did not show how to compute the mixed strategy in polynomial time. Our linear program formulation is similar to their formulation, and we show how to construct the mixed strategy from the solution, using the Birkhoff-von Neumann theorem (Birkhoff 1946).

To solve the problem, we actually solve multiple LPs: for each target t^*, we solve an LP that computes the best mixed strategy to commit to, under the constraint that the attacker is incentivized to attack t^*. We then solve all of these LPs, and take the solution that maximizes the leader's utility. This is similar to the set of linear programs used by Conitzer and Sandholm (2006), except those linear programs require a variable for each pure strategy for the defender, so that these LPs have exponential size in our domain. Instead, we will write a more compact LP to find the probability $c_{\omega,t}$ of assigning resource ω to target t, for each ω and $t \in A(\omega)$. (If $t \notin A(\omega)$, then there is no variable $c_{\omega,t}$.)

$$\text{maximize } U_d(t^*, \mathbf{c})$$

subject to

$$\forall \omega \in \Omega, \forall t \in A(\omega) : 0 \leq c_{\omega,t} \leq 1$$

$$\forall t \in T : c_t = \sum_{\omega \in \Omega : t \in A(\omega)} c_{\omega,t} \leq 1$$

$$\forall \omega \in \Omega : \sum_{t \in A(\omega)} c_{\omega,t} \leq 1$$

$$\forall t \in T : U_a(t, \mathbf{c}) \leq U_a(t^*, \mathbf{c})$$

The advantage of this LP is that it is more compact than the one that considers all pure strategies. The downside is that it is not immediately clear whether we can actually implement the computed probabilities (that is, whether they correspond to a probability distribution over allocations of resources to targets, and how this mixed strategy can be found). Below we show that the obtained probabilities can, in fact, be implemented.
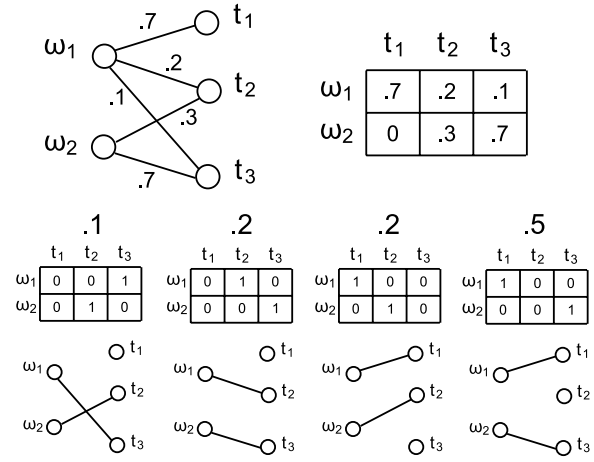

Figure 2: An example of how to apply the BvN theorem. Top Left: Resource ω_1 can cover targets t_1, t_2, t_3; ω_2 can cover t_2, t_3. Top Right: The LP returns the marginal probabilities in the table. We must now obtain these marginal probabilities as a probability mixture over pure strategies, in which every resource is assigned to a separate target. Bottom: The BvN theorem decomposes the top right table into a mixture over pure strategies. It first places probability .1 on the pure strategy on the left, then .2 on the pure strategy to the right of that, and so on. It is easily checked that with the resulting mixture over pure strategies, the marginal probabilities in the top right table are obtained.

Constructing a Strategy that Implements the LP Solution

We will make heavy use of the following theorem (which we state in a somewhat more general form than it is usually stated).

Theorem 1 (Birkhoff-von Neumann (Birkhoff 1946)). *Consider an $m \times n$ matrix M with real numbers $a_{ij} \in [0, 1]$, such that for each $1 \leq i \leq m$, $\sum_{j=1}^{n} a_{ij} \leq 1$, and for each $1 \leq j \leq n$, $\sum_{i=1}^{m} a_{ij} \leq 1$. Then, there exist matrices $M^1, M^2, \ldots, M^q$, and weights $w^1, w^2, \ldots, w^q \in (0, 1]$, such that:*

1. *$\sum_{k=1}^{q} w^k = 1$;*
2. *$\sum_{k=1}^{q} w^k M^k = M$;*
3. *for each $1 \leq k \leq q$, the elements of M^k are $a_{ij}^k \in \{0, 1\}$;*
4. *for each $1 \leq k \leq q$, we have: for each $1 \leq i \leq m$, $\sum_{j=1}^{n} a_{ij}^k \leq 1$, and for each $1 \leq j \leq n$, $\sum_{i=1}^{m} a_{ij}^k \leq 1$.*

Moreover, q is $O((m + n)^2)$, and the M^k and w^k can be found in $O((m + n)^{4.5})$ time using Dulmage-Halperin algorithm (Dulmage and Halperin 1955; Chang, Chen, and Huang 2001).

We can use this theorem to convert the probabilities $c_{\omega,t}$ that we obtain from our linear programming approach into a mixed strategy. This is because the $c_{\omega,t}$ constitute an $m \times n$ matrix that satisfies the conditions of the Birkhoff-von Neumann theorem. Each M^k that we obtain as a result of this
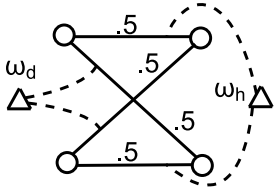
Figure 3: A counterexample that shows that with heterogeneous resources and bipartite schedules, the linear program probabilities are not always implementable. There are 4 targets (shown as circles), 4 schedules (solid edges), and 2 resources. The resource ω_h can be assigned to one of the horizontal edges and the resource ω_d can be assigned to one of the diagonal edges. In the optimal solution to the LP, the probability of a resource being assigned to each edge is 0.5, so that it would seem that the probability of each target being covered is 1. However, it is easy to see that in reality, the two resources can cover at most 3 of the 4 targets simultaneously.
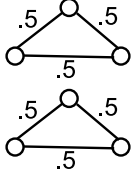

Figure 4: A counterexample that shows that with homogeneous resources and schedules of size two that are not bipartite, the linear program probabilities are not always implementable. The number of resources is $m = 3$. 6 targets are represented by vertices, 6 schedules are represented by edges. In the optimal solution to the LP, the probability of a resource being assigned to each edge is 0.5, so that it would seem that the probability of each target being covered is 1. However, it is easy to see that in reality, the three resources can cover at most 5 of the 6 targets simultaneously.

application of the theorem corresponds to a pure strategy in our domain: M^k consists of entries $c_{\omega,t}^k \in \{0, 1\}$ (by 3), which we can interpret to mean that ω is assigned to t if and only if $c_{\omega,t}^k = 1$, because of the conditions on M^k (in 4). Then, because the weights sum to 1 (by 1), we can think of $\sum_{k=1}^q w^k M^k$ as a mixed strategy in our domain, which gives us the right probabilities (by 2). According to the theorem, we can construct this mixed strategy (represented as an explicit listing of the pure strategies in its support, together with their probabilities) in polynomial time. An example is shown on Figure 2. From this analysis, the following theorem follows:

Theorem 2. *When schedules have size 1 or 0, we can find an optimal Stackelberg strategy in polynomial time, even with heterogeneous resources. This can be done by solving a set of polynomial-sized linear programs and then applying the Birkhoff-von Neumann theorem.*

Heterogeneous Resources, Schedules of Size 2, Bipartite Graph

In this section, we consider schedules of size two. When schedules have size two, they can be represented as a graph, whose vertices correspond to targets and whose edges correspond to schedules. In this section, we consider the special case where this graph is bipartite, and give an NP-hardness proof for it.

One may wonder why this special case is interesting. In fact, it corresponds to the Federal Air Marshals domain studied by Kiekintveld et al. (2009). In this domain, flights are targets. If a Federal Air Marshal is to be scheduled on one outgoing flight from the U.S. (to, say, Europe), and will then return on an incoming flight, this is a schedule that involves two targets; moreover, there cannot be a schedule consisting of two outgoing flights or of two incoming flights, so that we have the requisite bipartite structure.

It may seem that the natural approach is to use a generalization of the linear program from the previous section (or, the mixed integer program from Kiekintveld et al. (2009)) to compute the marginal probabilities $c_{\omega,s}$ that resource ω is assigned to schedule s; and, subsequently, to convert this into a distribution over pure strategies that gives those marginal probabilities. However, it turns out that it is, in some cases,

impossible to find such a distribution over pure strategies. That is, the marginal probabilities from the linear program are not actually implementable. A counterexample is shown in Figure 3. One may wonder if perhaps a different linear program or other efficient algorithm can be given. We next show that this is unlikely, because finding an optimal strategy for the leader in this case is actually NP-hard, even in zero-sum games.

Theorem 3. *When resources are heterogeneous, finding an optimal Stackelberg strategy is NP-hard, even when schedules have size 2 and constitute a bipartite graph, and the game is zero-sum.*

Our reduction is from satisfiability; please see the full version (available online) for the proof.

If the resources are homogeneous, then it turns out that in the bipartite case, we can solve for an optimal Stackelberg strategy in polynomial time, by using the Birkhoff-von Neumann theorem in a slightly different way. We skip the details due to the space constraint; in the next section, we show how a more general case can be solved in polynomial time using a different technique.

Homogeneous Resources, Schedules of Size 2

We now return to the case where resources are homogeneous and schedules have size 2, but now we no longer assume that the graph is bipartite. It turns out that if we use the linear program approach, the resulting marginal probabilities c_s are in general not implementable, that is, there is no mixed strategy that attains these marginal probabilities. A counterexample is shown in Figure 4. This would appear to put us in a position similar to that in the previous section. However, it turns out that here we can actually solve the problem in polynomial time, using a different approach. Our approach here is to use the standard linear programming approach from Conitzer and Sandholm (2006), described at the beginning of the paper. The downside of using such approach is that there are exponentially many variables. In contrast, the dual linear program has only $n+1$ variables, but exponentially many constraints. One approach to solving a linear program with exponentially many constraints is the following: start with only a small subset of the constraints, and solve the resulting reduced linear program. Then, using
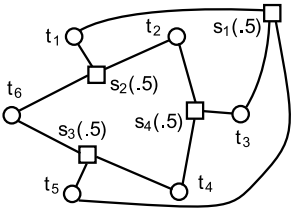
Figure 5: A counterexample that shows that with homogeneous resources and schedules of size three, the linear program probabilities are not always implementable. The number of resources is $m = 2$. 6 targets are represented by round nodes, 6 schedules are represented by square nodes with connections to the targets that they include. In the optimal solution to the LP, the probability of a resource being assigned to each schedule is 0.5, so that it would seem that the probability of each target being covered is 1. However, it is easy to see that in reality, the two resources can cover at most 5 of the 6 targets simultaneously.

some other method, check whether the solution is feasible for the full (original) linear program; and if not, find a violated constraint. If we have a violated constraint, we add it to the set of constraints, and repeat. Otherwise, we have found an optimal solution. This process is known as *constraint generation*. Moreover, if a violated constraint can be found in polynomial time, then the original linear program can be solved in polynomial time using Ellipsoid algorithm. As we will show, in the case of homogeneous resources and schedules of size two, we can efficiently generate constraints in the dual linear program by solving a weighted matching problem. While this solution is less appealing than our earlier solutions based on the Birkhoff-von Neumann theorem, it still results in a polynomial-time algorithm. The dual linear program follows.

minimize y

subject to

$$\forall \alpha : \sum_{t \in T} y_t(U_a(\alpha, t) - U_a(\alpha, t^*)) + y \geq U_d(\alpha, t^*)$$

$$y \in \mathbb{R}$$

Now, we consider the constraint generation problem for the dual LP. Given a (not necessarily feasible) solution y_t, y to the dual, we need to find the most violated constraint, or verify that the solution is in fact feasible. Our goal is to find, given the candidate solution y_t, y,

$$\alpha \in \arg\max_\alpha U_d(\alpha, t^*) - \sum_{t \in T} y_t(U_a(\alpha, t) - U_a(\alpha, t^*)) - y$$

We introduce an indicator function $I_\alpha(t)$ which is equal to 1 if t is covered by α, and 0 otherwise. Then

$$U_a(\alpha, t) = U_a^u(t) + I_\alpha(t)(U_a^c(t) - U_a^u(t))$$
$$U_d(\alpha, t) = U_d^u(t) + I_\alpha(t)(U_d^c(t) - U_d^u(t))$$

Then, we can rearrange the optimization problem as follows.

$$\alpha \in \arg\max_\alpha U_d^u(t^*) + I_\alpha(t^*)(U_d^c(t^*) - U_d^u(t^*))$$

$$- y - \sum_{t \in T} y_t(U_a^u(t) + I_\alpha(t)(U_a^c(t) - U_a^u(t)))$$

$$+ \sum_{t \in T} y_t(U_a^u(t^*) + I_\alpha(t^*)(U_a^c(t^*) - U_a^u(t^*)))$$

$$= U_d^u(t^*) - y - \sum_{t \in T} y_t(U_a^u(t) - U_a^u(t^*))$$

$$+ \sum_{t \in T} I_\alpha(t) y_t(U_a^u(t) - U_c^u(t))$$

$$- I_\alpha(t^*)(U_a^u(t^*) - U_a^c(t^*)) \sum_{t \in T} y_t$$

We define a weight function on the targets as follows:

$$w(t) = y_t(U_a^u(t) - U_a^c(t)) \text{ for } t \neq t^*$$
$$w(t^*) = -(U_a^u(t^*) - U_a^c(t^*)) \sum_{t \in T, t \neq t^*} y_t$$

We then rearrange the optimization problem as follows:

$$\alpha \in \arg\max_\alpha w(\alpha) + U_d^u(t^*) - \sum_{t \in T} y_t(U_a^u(t) - U_a^u(t^*)) - y$$

where $w(\alpha)$ is the total weight of the targets covered by the pure strategy α: $w(\alpha) = \sum_{t \in \cup_{s \in \alpha} s} w(t)$. The only part of the objective that depends on α is $w(\alpha)$, so we can focus on finding an α that maximizes $w(\alpha)$. A pure strategy α is a collection of edges (schedules consisting of pairs of targets). Therefore, the problem of finding an α with maximum weight is a maximum weighted 2-cover problem, which can be solved in polynomial time (for example, using a modification of the algorithm for finding a maximal weighted matching in general graphs (Galil, Micali, and Gabow 1986)). So, we can solve the constraint generation problem, and hence the whole problem, in polynomial time. From this analysis, the following theorem follows:

Theorem 4. *When resources are homogeneous and schedules have size at most 2, we can find an optimal Stackelberg strategy in polynomial time. This can be done by solving the standard Stackelberg linear programs (Conitzer and Sandholm 2006): these programs have exponentially many variables, but the constraint generation problem for the dual can be solved in polynomial time in this case.*

Homogeneous Resources, Schedules of Size 3

We now move on to the case of homogeneous resources with schedules of size 3. Once again, it turns out that if we use the linear program approach, the resulting marginal probabilities c_s are in general not implementable; that is, there is no mixed strategy that attains these marginal probabilities. A counterexample is shown in Figure 5.

We now show that finding an optimal strategy for the leader in this case is actually NP-hard, even in zero-sum games.

Theorem 5. *When schedules have size 3, finding an optimal Stackelberg strategy is NP-hard, even when resources are homogeneous and the game is zero-sum.*

Proof. We reduce an arbitrary 3-cover problem instance—in which we are given a universe $\mathcal{U}$, a family $\mathcal{S}$ of subsets of $\mathcal{U}$, such that each subset contains 3 elements, and we are asked whether we can (exactly) cover all of $\mathcal{U}$ using $|\mathcal{U}|/3$

Hetero-geneous resources	Schedules			
	size 1	size $\leq$ 2, bipartite	size $\leq$ 2	size $\geq$ 3
No	P	P	P	NP-hard
Yes	P	NP-hard	NP-hard	NP-hard

Figure 6: Summary of the computational results. All of the hardness results hold even for zero-sum games.

elements of $\mathcal{S}$—to a game with homogeneous resources and schedules of size 3. We create one target for each element of $\mathcal{U}$, and one schedule for each element of $\mathcal{S}$, which covers the targets in it. We also create $|\mathcal{U}|/3$ homogeneous resources that each can cover any schedule. The utilities are $\forall t : U_d^c(t) = U_a^u(t) = 1, U_d^u(t) = U_a^c(t) = 0$. The defender can obtain a utility of 1 if and only if she can cover every target with probability 1, which is possible if and only if the original 3-cover instance has a solution. □

Conclusion

In this paper, we studied the complexity of solving Stackelberg games in which pure strategies correspond to allocations of resources, resulting in exponentially large strategy spaces. We precisely characterized in which cases the problem is solvable in polynomial time (in some cases by solving linear programs of polynomial size and appealing to the Birkhoff-von Neumann theorem, in another case by solving linear programs of exponential size by using a polynomial-time constraint generation technique), and in which cases it is NP-hard. The results for the case where the attacker has a single resource are given in Figure 6.

Our results are perhaps made more interesting by a recent paper by Yin et al. (2010), which shows that for all of the security games that we studied, an optimal Stackelberg strategy is guaranteed to also be a Nash equilibrium strategy in the version of the game where commitment is not possible. (The converse does not hold, that is, there can be Nash equilibrium strategies that are not Stackelberg strategies.) Thus, our polynomial-time algorithm results also allow us to find a Nash equilibrium strategy for the defender in polynomial time. Conversely, for the cases where we prove a hardness result, finding a Nash equilibrium strategy is also NP-hard, because our hardness results hold even for zero-sum games.

Presumably, the most important direction for future research is to address the NP-hard cases. Can we find algorithms that, although they require exponential time in the worst case, solve typical instances fast? Can we identify additional restrictions on the game so that the problem becomes polynomial-time solvable? Are there good polynomial-time approximation algorithms, or anytime algorithms that find a reasonably good solution fast? Another direction for future research is to consider security games with incomplete information (Bayesian games) or multiple time periods (extensive-form games). In unrestricted games, these aspects can lead to additional complexity (Conitzer and Sandholm 2006; Letchford, Conitzer, and Munagala 2009; Letchford and Conitzer 2010).

References

Birkhoff, G. 1946. Tres observaciones sobre el algebra lineal. *Univ. Nac. Tucumn Rev, Ser. A, no. 5* 147–151.

Chang, C.-S.; Chen, W.-J.; and Huang, H.-Y. 2001. Coherent cooperation among communicating problem solvers. *IEEE Transactions on Communications* 49(7):1145–1147.

Chen, X., and Deng, X. 2006. Settling the complexity of two-player Nash equilibrium. In *FOCS*, 261–272.

Conitzer, V., and Sandholm, T. 2006. Computing the optimal strategy to commit to. In *ACM EC*, 82–90.

Conitzer, V., and Sandholm, T. 2008. New complexity results about Nash equilibria. *GEB* 63(2):621–641.

Daskalakis, C.; Goldberg, P.; and Papadimitriou, C. H. 2006. The complexity of computing a Nash equilibrium. In *STOC*, 71–78.

Dulmage, L., and Halperin, I. 1955. On a theorem of Frobenius-Konig and J. von Neumann's game of hide and seek. *Trans. Roy. Soc. Canada III* 49:23–29.

Galil, Z.; Micali, S.; and Gabow, H. 1986. An $O(EV \log V)$ algorithm for finding a maximal weighted matching in general graphs. *SIAM J. Comput.* 15(1):120–130.

Gilboa, I., and Zemel, E. 1989. Nash and correlated equilibria: Some complexity considerations. *GEB* 1:80–93.

Kiekintveld, C.; Jain, M.; Tsai, J.; Pita, J.; Ordóñez, F.; and Tambe, M. 2009. Computing optimal randomized resource allocations for massive security games. In *AAMAS*, 689–696.

Letchford, J., and Conitzer, V. 2010. Computing optimal strategies to commit to in extensive-form games. In *ACM EC*.

Letchford, J.; Conitzer, V.; and Munagala, K. 2009. Learning and approximating the optimal strategy to commit to. In *SAGT*, 250–262.

Nash, J. 1950. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences* 36:48–49.

Paruchuri, P.; Pearce, J. P.; Marecki, J.; Tambe, M.; Ordóñez, F.; and Kraus, S. 2008. Playing games for security: an efficient exact algorithm for solving Bayesian Stackelberg games. In *AAMAS*, 895–902.

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

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

von Neumann, J. 1927. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen* 100:295–320.

von Stengel, B., and Zamir, S. 2009. Leadership games with convex strategy sets. *GEB*.

Yin, Z.; Korzhyk, D.; Kiekintveld, C.; Conitzer, V.; and Tambe, M. 2010. Stackelberg vs. Nash in security games: Interchangeability, equivalence, and uniqueness. In *AAMAS*.