Security Games with Protection Externalities

Jiarui Gan
The Key Laboratory of Intelligent
Information Processing, ICT, CAS
University of Chinese Academy of Sciences
Beijing 100190, China
ganjr@ics.ict.ac.cn

Bo An
School of Computer Engineering
Nanyang Technological University
Singapore 639798
boan@ntu.edu.sg

Yevgeniy Vorobeychik
Electrical Engineering & Computer Science
Vanderbilt University
Nashville, TN 37235
yevgeniy.vorobeychik@vanderbilt.edu

Abstract

Stackelberg security games have been widely deployed in recent years to schedule security resources. An assumption in most existing security game models is that one security resource assigned to a target only protects that target. However, in many important real-world security scenarios, when a resource is assigned to a target, it exhibits *protection externalities*: that is, it also protects other "neighbouring" targets. We investigate such Security Games with Protection Externalities (SPEs). First, we demonstrate that computing a strong Stackelberg equilibrium for an SPE is NP-hard, in contrast with traditional Stackelberg security games which can be solved in polynomial time. On the positive side, we propose a novel *column generation* based approach—CLASPE—to solve SPEs. CLASPE features the following novelties: 1) a novel mixed-integer linear programming formulation for the slave problem; 2) an extended greedy approach with a constant-factor approximation ratio to speed up the slave problem; and 3) a linear-scale linear programming that efficiently calculates the upper bounds of target-defined subproblems for pruning. Our experimental evaluation demonstrates that CLASPE enable us to scale to realistic-sized SPE problem instances.

Introduction

Stackelberg security games have been successfully used for computing optimal allocation of security resources in adversarial and strategic scenarios (Basilico, Gatti, and Amigoni 2009; Tambe 2011; Conitzer 2012). In a Stackelberg security game, a defender allocates security resources (e.g., police guards, canines, or patrol units) to protect key infrastructures, such as airports, ports, and ferry ships. An attacker first conducts surveillance and then chooses the best target to attack. Algorithms have been designed to calculate the best defender strategy for Stackelberg security games in various settings, and systems applying such models have been developed and deployed in many real-world security domains (e.g., Pita et al. 2008; Tsai et al. 2009; An et al. 2011; Pita et al. 2011; Shieh et al. 2012; An et al. 2013a; An et al. 2013b; Yin, An, and Jain 2014).

A common assumption in existing security game models is that one security resource can be assigned to only one target at a time, and it protects this target only. However, in real-world scenarios security resources often effectively protect multiple targets simultaneously. For instance, surveillance cameras and radars can conduct surveillance over a perceived range (Figure 1(a)). As another example, mobile security units, such as policemen and patrol canines, can keep their eyes on neighbouring targets within certain distance, and react to emergencies therein (Figure 1(b)). We term such scenarios Security Games with Protection Externalities (SPE). Several existing security game applications incorporate a form of protection externalities. Fang, Jiang, and Tambe (2013) and Xu et al. (2014) study patrol path planning for ferry ship protection, where patrollers can protect ferry ships within a protection radius. A major difference is that they assume the targets are located in a one-dimensional space. In the Federal Air Marshal (FAM) allocation domain and other patrolling domains (Tsai et al. 2009; Jain et al. 2010; Shieh et al. 2012), each resource (e.g., an air marshal) can visit multiple targets (e.g., flights) in a schedule and a target is protected only if it is visited by the resource. There are in fact no protection externalities in these domains (e.g., a resource cannot protect more than one flight at a time). Thus, even though a schedule can be viewed as a "neighborhood" in SPEs, past computational approaches have restricted attention to non-overlapping schedules, and made other related structural assumptions (e.g., that a subset of a schedule is also a feasible schedule (Korzhyk, Conitzer, and Parr 2010)). In contrast, SPEs treat neighborhoods as indivisible units, such that targets are automatically protected when resources are assigned to their neighbouring targets.

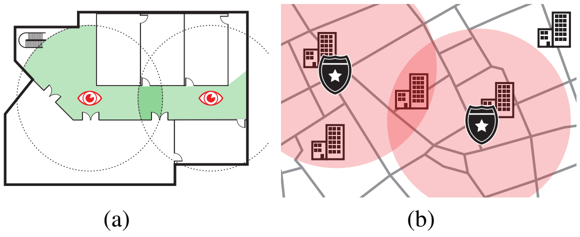

(a)
(b)

Figure 1: Protection externalities in real-world scenarios: (a) visible areas of surveillance cameras in a building; (b) reaction area of mobile security units.

Protection externalities result in a complex marginal coverage space for the defender, and make SPEs quite challenging to solve. The first contribution of this paper is that we show that the problem of finding the optimal defender strategy of an SPE under the *strong Stackelberg equilibrium* (SSE) solution concept is NP-hard. This is in contrast with traditional security games, which can be solved in polynomial time (Kiekintveld et al. 2009; Korzhyk et al. 2011). To address the NP-hardness, our second contribution is CLASPE, a novel algorithm which uses column generation to avoid enumerating the full (exponential in the number N of targets) pure strategy space of the defender. Column generation has been introduced in the security game domain to solve problems facing similar scalability issues (Jain et al. 2010; Shieh et al. 2013). The key to the application of the column generation framework is efficient computation of the optimal column in the slave problem. We formulate the slave problem as a *mixed-integer linear programming* (MILP), and to speed up the process we proposed an extended greedy approach which is in particular able to achieve a constant-factor approximation ratio in the presence of a negative weight. An $O(N)$-scale *linear programming* (LP) is also devised to prune target-defined subproblems by efficiently calculating their upper bounds. Experiments demonstrate that CLASPE is able to solve realistically-sized SPE problem instances, leveraging the high-quality approximations achieved by the greedy algorithm and the pruning LP.

Security Games with Protection Externalities

An SPE is played between a *defender* and an *attacker*. The defender allocates K identical resources to protect a set of targets $[N] = \{1, \dots, N\}$, and the attacker chooses a single target from $[N]$ to attack. The defender has a set $\mathcal{S}$ of pure strategies. Each pure strategy is an allocation of resources over targets $[N]$, and is denoted by a 0/1 vector $\mathbf{s} \in \{0,1\}^N$, such that $s_i = 1$ if a resource is assigned to target i, and $s_i = 0$ otherwise. If a resource is assigned to a target, it protects (covers) this target, as well as its neighbouring targets.[1] The neighbour relationships between targets are specified by an adjacency matrix $\mathbf{A} = \langle a_{ij} \rangle$, such that $a_{ij} = 1$ if assigning a resource to target i also protects target j, and $a_{ij} = 0$ otherwise (notably, targets are neighbouring to themselves, so that $a_{ii} = 1$ for all i). We use a vector $\mathbf{c}$ to represent the protection status of the targets given a pure strategy $\mathbf{s}$, such that $c_i = 1$ if $\sum_j s_j \cdot a_{ji} \geq 1$, i.e., when at least one resource protects i, and $c_i = 0$ if $\sum_j s_j \cdot a_{ji} = 0$.[2] Additionally, we define a function $\Phi : \mathbf{s} \mapsto \mathbf{c}$ that maps each pure strategy to the targets' protection status it leads to. The defender can play a mixed strategy $\mathbf{x}$, with $x_{\mathbf{s}} \geq 0$ being the probability of playing pure strategy $\mathbf{s}$. As other work in the literature (Conitzer and Sandholm 2006), we assume that the attacker only attacks a single target, and thus his strategy space is the set of targets.

The payoffs for each player depend on which target is attacked and the probability that the target is covered. If the attacker attacks target i and target i is covered, the defender receives a reward R_i^d and the attacker receives a penalty P_i^a. Otherwise, if target i is not covered, the payoffs for the defender and attacker are P_i^d and R_i^a. We assume that $R_i^d > P_i^d$ and $R_i^a > P_i^a$ in order to model that the defender would always prefer the attack to fail, while the attacker would prefer it to succeed. Given a strategy profile $\langle \mathbf{x}, i \rangle$, the expected utilities for the defender and the attacker are respectively given by:

$$\sum_{\mathbf{s}\in\mathcal{S}} x_{\mathbf{s}} \cdot U_d\big(\Phi(\mathbf{s}), i\big), \text{ where } U_d(\mathbf{c}, i) = c_i R_i^d + (1 - c_i) P_i^d \quad (1)$$

$$\sum_{\mathbf{s}\in\mathcal{S}} x_{\mathbf{s}} \cdot U_a\big(\Phi(\mathbf{s}), i\big), \text{ where } U_a(\mathbf{c}, i) = c_i P_i^a + (1 - c_i) R_i^a \quad (2)$$

We adopt a Stackelberg game model as most security game researches did. In a Stackelberg game, a leader (i.e., the defender) moves first, and a follower (i.e., the attacker) reacts with an optimal (pure) strategy after observing the defender's strategy. SSE is the standard solution concept for Stackelberg games. In an SSE, the defender chooses an optimal strategy accounting for the attacker's best response to this strategy, under the assumption that the attacker breaks ties in favor of the defender (Von Stengel and Zamir 2004).

CLASPE—A Column Generation Approach

In this section, we first show that solving an SPE is NP-hard in Theorem 1, by reducing an arbitrary *set-cover decision problem*, which is known to be NP-complete (by a straightforward reduction from the *set cover problem*), to an SPE problem. This is in contrast with polynomial-time solvability of security games without protection externalities (e.g., ORIGAMI algorithm by Kiekintveld et al. 2009).

Theorem 1. *Finding an optimal SSE strategy for the defender in an SPE is NP-hard, even if the adjacency matrix is symmetric, i.e., $a_{ij} = a_{ji}$, $\forall i, j \in [N]$.*

Proof. We reduce a *set-cover decision problem* to an SPE with symmetric adjacency matrix. Given a set $\mathcal{U}$ of elements, a set $\mathcal{A}$ of subsets of $\mathcal{U}$ such that $\bigcup_{S\in\mathcal{A}} S = \mathcal{U}$, and an integer K, a set-cover decision problem asks whether there exists a subset $\mathcal{B} \subseteq \mathcal{A}$, such that $|\mathcal{B}| \leq K$ and $\bigcup_{S\in\mathcal{B}} S = \mathcal{U}$. Denote a set-cover decision problem as a tuple $\langle \mathcal{U}, \mathcal{A}, K \rangle$. For arbitrary $\langle \mathcal{U}, \mathcal{A}, K \rangle$, we construct a symmetric SPE as follows.

Step 1: We create a target i for each element $i \in \mathcal{U}$, and set the entries of the adjacent matrix a_{ii}=1, and a_{ij}=0 $\forall j \neq i$.

Step 2: For each $S \in \mathcal{A}$, we create a new target, say target i, and set $a_{ij} = a_{ji} = 1$ for all $j \in S$ and all targets j

[1] We restrict the allocation of resources to targets, while in more general cases resources can also be placed on some non-target places. This can be handled by converting each non-target place to a dummy target which gives the attacker a utility of $-\infty$ no matter whether it is covered or not.

[2] In more general cases, a target can be *partially* protected by a resource such that entries of the adjacency matrix are in the range $[0, 1]$. Our algorithm applies to these general cases with some minor modifications (see Appendix C; Appendix of this paper can be found at http://www.ntu.edu.sg/home/boan/papers/AAAI15_Externality_Appendix.pdf).

created in this step (i.e., $j \notin \mathcal{U}$). The other entries are set to 0. We say that target i and set S *correspond* to each other. Note that now each set in $\mathcal{A}$ corresponds to a target created in Step 2 with each other; and targets created in Step 2 are fully connected, i.e., assigning a resource to any one of them covers all of them.

Step 3: Set the number of security resources to K. Set utilities $P_i^a = P_i^d = 0$ and $R_i^a = R_i^d = 1$, for all target i created in Steps 1 and 2.

For example, for a set cover decision problem with $\mathcal{U} = \{1,2,3,4\}$ and $\mathcal{A} = \{\{1\},\{1,2\},\{2,3,4\}\}$, the above steps create a symmetric SPE with 7 targets $\{1,\ldots,7\}$ and the following adjacency matrix:

$$\left[\begin{array}{cccc:ccc} 1 & & & & 1 & 1 & \\ & 1 & & & & 1 & 1 \\ & & 1 & & & & 1 \\ & & & 1 & & & 1 \\ \hdashline 1 & & & & 1 & 1 & 1 \\ 1 & 1 & & & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right],$$

where the sub-matrix in the upper left is generated in Step 1, while the rest in Step 2.

Obviously, the above steps are done in polynomial time. The defender can obtain a utility of 1 if and only if she can cover every target with probability 1 with no more than K resources, which is possible if and only if the set cover decision problem has a solution $\mathcal{B} \subseteq \mathcal{A}$ that covers $\mathcal{U}$ and contains no more than K elements. This is shown in the follows.

The 'if' direction: If the set cover decision problem has a solution $\mathcal{B}$, the defender can use a pure strategy with probability 1 by which $|\mathcal{B}| \leq K$ resources are assigned to targets corresponding to the sets in $\mathcal{B}$. In such a way, all the targets created in Step 1 must be covered since $\bigcup_{S\in\mathcal{B}} S = \mathcal{U}$; and the other targets (those created in Step 2) are also covered since they are fully connected.

The 'only if' direction: If the defender can cover all targets with probability 1, then she has at least one pure strategy where all the targets are covered with no more than K resources. This indicates that we can construct $\mathcal{B}$ with the sets corresponding to the targets which have been assigned resources, and the union of these sets must cover all elements in $\mathcal{U}$. Note that there is no corresponding set for targets created in Step 1 (e.g., targets $1,\ldots,4$ in the example). Thus if resources are assigned to these targets by the pure strategy, we cannot find a corresponding set when constructing $\mathcal{B}$ in the above way. However, if this happens, i.e., a resource is assigned to such a target i, we can always reassign the resource to another target j created in Step 2 with $a_{ji} = 1$ (such a target j always exists since $\bigcup_{S\in\mathcal{A}} S = \mathcal{U}$). In such a way, the resource on target j covers at least all the targets covered by a resource on i, because assigning a resource on target i only covers itself and some targets created in Step 2, while assigning a resource on target j covers target i (since $a_{ji} = 1$) and *all* targets created in Step 2 (since they are fully connected). This means that we can transfer all resources from targets created in Step 1 to those created in Step 2 while at the same time all the previously covered targets are covered as well. □

The NP-hardness suggests that existing polynomial-time algorithms are not applicable to solving SPEs. The difficulty lies in the combinatorial explosion of defender's pure strategy space, and the complex marginal coverage space that cannot be compactly represented. To address the difficulty, we propose a novel algorithm CLASPE (a CoLumn generation based Algorithm for SPEs). CLASPE formulates an SPE problem as multiple target-defined LPs, and uses column generation to deal with the exponential growth in the size of the LPs as a result of the combinatorial explosion of defender's pure strategy space. To further speed up the process, CLASPE also incorporates a greedy approach to approximate slave problems in the column generation process, as well as an efficient procedure to prune the target-defined LPs by calculating their upper bounds. We present the above components of CLASPE in the rest of this section.

Target-defined LPs

CLASPE formulates the problem of solving an SPE as a set of *target-defined LPs* (*t*-LP). Each *t*-LP corresponds to a target, and computes the defender's optimal strategy under the constraint that the attacker's best response is to attack this target. The *t*-LP with the largest optimal objective value provides an optimal SSE strategy for the defender. Specifically, the *t*-LP defined by target i^+ is structured as follows.

$$\max_{\mathbf{x}} \quad \sum_{\mathbf{s}\in\mathcal{S}} x_{\mathbf{s}} \cdot U_d\big(\Phi(\mathbf{s}), i^+\big) \tag{3}$$

$$\text{s.t.} \quad \sum_{\mathbf{s}\in\mathcal{S}} x_{\mathbf{s}} \cdot U_a\big(\Phi(\mathbf{s}), i^+\big) \geq \sum_{\mathbf{s}\in\mathcal{S}} x_{\mathbf{s}} \cdot U_a\big(\Phi(\mathbf{s}), i\big),\ \forall i \tag{4}$$

$$\sum_{\mathbf{s}\in\mathcal{S}} x_{\mathbf{s}} = 1 \tag{5}$$

$$x_{\mathbf{s}} \geq 0,\ \forall \mathbf{s} \in \mathcal{S} \tag{6}$$

One issue with the above formulation is that the number of variables increases exponentially with the size of defender's pure strategy space, we use *column generation*—a standard technique for solving large scale LPs—to address this issue.

Column Generation for *t*-LPs

The idea of the column generation approach is to begin with a restricted version of a *t*-LP, where only a small subset $\mathcal{S}' \subset \mathcal{S}$ of pure strategies are considered. The solution of this restricted problem might not be optimal since some other pure strategies in $\mathcal{S} \setminus \mathcal{S}'$ might be in the support of the optimal mixed strategy. Therefore, we search in $\mathcal{S} \setminus \mathcal{S}'$ for a pure strategy such that when it is added to the restricted problem, the solution can be improved. The restricted problem is referred to as the *master problem*, while the problem of finding a new pure strategy (i.e., a column) the *slave problem*. The master problem and the slave problem are solved repeatedly, until no pure strategy can be added to improve the solution when the optimum is reached.

To find a new pure strategy, the slave problem uses the concept of *reduced cost*, which captures the potential change in the solution of the master problem when a new column is added. For an LP that maximizes $\mathbf{g}^{\mathsf{T}}\mathbf{x}$ subjecting to $\mathbf{H}\mathbf{x} \leq \mathbf{h}$ and $\mathbf{x} \geq \mathbf{0}$, the reduced costs for all columns as a vector is given by $\mathbf{g} - \mathbf{H}^\mathsf{T}\mathbf{y}$, where $\mathbf{y}$ are the dual variables for constraints $\mathbf{Hx} \leq \mathbf{h}$ (Bertsimas and Tsitsiklis 1994). It follows that in our problem, when $\mathbf{s}$ is added to the master problem, the reduced cost $r(\mathbf{s})$ is

$$U_d\big(\Phi(\mathbf{s}), i^+\big) - \mathbf{y}^\mathsf{T} \begin{bmatrix} U_a\big(\Phi(\mathbf{s}), 1\big) - U_a\big(\Phi(\mathbf{s}), i^+\big) \\ \vdots \\ U_a\big(\Phi(\mathbf{s}), n\big) - U_a\big(\Phi(\mathbf{s}), i^+\big) \end{bmatrix} - y' \quad (7)$$

where $\mathbf{y}$ and y' are the dual variables corresponding to Eqs. (4) and (5), respectively. The optimality is reached by the master problem when no new column can be found with a positive reduced cost. Next, we present a MILP formulation for the slave problem.

A MILP for Slave Problems The reduced cost defined in Eq. (7) is linear to the vector of protection status, i.e., $\mathbf{c} = \Phi(\mathbf{s})$. We can thus write the reduced cost as $r(\mathbf{s}) = w_0 + \sum_i w_i \cdot c_i$ for simplicity, where $w_0, \ldots, w_N$ are constants. Specifically, substituting Eqs. (1)–(2) into Eq. (7), we have

$$\begin{aligned} r(\mathbf{s}) = &\Big(R^d_{i^+} - P^d_{i^+} + \big(P^a_{i^+} - R^a_{i^+}\big) \cdot \sum_{i \neq i^+} y_i\Big) \cdot c_{i^+} \\ &+ \sum_{i \neq i^+} \big(R^a_i - P^a_i\big) \cdot y_i \cdot c_i \\ &+ \Big(P^d_{i^+} + R^a_{i^+} \cdot \sum_{i \neq i^+} y_i - \sum_{i \neq i^+} y_i \cdot R^a_i - y'\Big) \quad (8) \end{aligned}$$

We formulate the slave problem as the following MILP.

$$\max_{\mathbf{c},\mathbf{s}} \quad w_0 + \sum\nolimits_i w_i \cdot c_i \quad (9)$$

$$\text{s.t.} \quad \mathbf{c}, \mathbf{s} \in \{0, 1\}^N \quad (10)$$

$$\sum\nolimits_i s_i \leq K \quad (11)$$

$$c_i \leq \sum\nolimits_j s_j \cdot a_{ji}, \ \forall i \in [N] \quad (12)$$

$$c_i \geq \frac{1}{K} \sum\nolimits_j s_j \cdot a_{ji}, \ \forall i \in [N] \quad (13)$$

Here Eqs. (12) and (13) guarantee that $\mathbf{c} = \Phi(\mathbf{s})$, because Eq. (12) indicates that $c_i = 0$ if $\sum_j s_j \cdot a_{ji} = 0$; and Eq. (13) indicates that $c_i = 1$ if $\sum_j s_j \cdot a_{ji} \geq 1$ as we always have $\sum_j s_j \cdot a_{ji} \leq K$. While this MILP will always compute an optimal solution for the slave, it is relatively slow. Next, we present an approximation algorithm which is much faster, with the idea that we first apply the approximation algorithm, and use the MILP only if no new column is thereby found (which still ensures that the final solution is optimal).

An Extended Greedy Approach for Slave Problems A slave problem can be viewed as a *weighted maximum coverage problem* (WMC). In a WMC we are given a collection of N sets such that the i^{th} set contains target j if $a_{ij} = 1$, and are asked to choose a collection of no more than K sets to cover the targets. If a target is covered, we receive w_i; otherwise, we receive 0. The goal is to maximize the total weight of covered targets. WMC is NP-hard, but when all the weights are non-negative, it admits a $(1 - \frac{1}{e})$-approximation with a polynomial-time greedy algorithm which always assigns a resource to the target with maximum marginal weight (see Algorithm 1) (Nemhauser, Wolsey, and Fisher 1978). However, we cannot apply the greedy algorithm directly, because the weight for target i^+ might be negative (according to Eq. (8), weights for targets $i \neq i^+$ are all non-negative since the dual variables $\mathbf{y}$ are non-negative as they are associated with inequality constraints). This means that the total weight may drop when more targets are covered, and the optimal solution may contain fewer than K sets. One simple solution is to keep track of the total weight in each step of the main loop (Lines 2–4), and choose the maximum record as the final solution. However, such a naïve greedy approach only provides a $\frac{1}{K}$-approximation.[3,4] We therefore introduce an extended greedy approach which preserves the above constant-factor approximation ratio. Specifically, when $w_{i^+} < 0$, we convert the problem into two independent new problems:

Algorithm 1: A greedy algorithm for a WMC with non-negative weights

```
Input:  An adjacency matrix $\mathbf{A} = \langle a_{ij} \rangle$
        A set of weights $\mathbf{w} = \langle w_i \rangle$
Output: A pure strategy $\mathbf{s}$
1 $\mathbf{s} \leftarrow \mathbf{0}, \mathbf{c} \leftarrow \mathbf{0}$;
2 for $k = 1$ to $K$ do
3     $\hat{i} \leftarrow \arg\max_{\{i | s_i = 0\}} \sum_{\{j \mid a_{ij}=1 \text{ and } c_j = 0\}} w_j$;
4     $s_{\hat{i}} \leftarrow 1, \mathbf{c} \leftarrow \Phi(\mathbf{s})$;
```

- WMC^0: by replacing w_{i^+} with 0.
- WMC$^\infty$: by replacing w_{i^+} with $-\infty$.

Then we solve WMC^0 and WMC$^\infty$ separately with Algorithm 1 (note that to apply Algorithm 1 to WMC$^\infty$, we can disable all sets containing target i^+, such that the problem is turned into one with all non-negative weight), and choose the solution that gives larger total weight to the original WMC. The intuition is that WMC^0 properly approximates a WMC when target i^+ is covered by the optimal solution of the WMC, and WMC$^\infty$ properly approximates a WMC when target i^+ is *not* covered. Thus the better solution of WMC^0 and WMC$^\infty$ properly approximates all instances. Indeed, a $(1 - \frac{1}{e})$-approximation of the above approach is demonstrated in Proposition 2.

Proposition 2. *The extended greedy approach provides a* $(1 - \frac{1}{e})$*-approximation to a WMC with a negative weight.*[5]

[3] To evaluate the approximation ability of the greedy algorithm in the presence of the negative weight w_{i^+}, a ratio $\gamma = \frac{GRD + |w_{i^+}|}{OPT + |w_{i^+}|}$ is introduced, where GRD is the objective value of the solution obtained using the greedy approach, and OPT is the objective value of the optimal solution. Clearly, $0 \leq \gamma \leq 1$, and larger γ indicates better approximation to the optimal solution.

[4] Please see Appendix B for details of the approximation bound.

[5] Please see Appendix A for proofs of propositions in this paper.

Upper bounds of *t*-LPs for Pruning

To further improve the efficiency, CLASPE incorporates a pruning procedure, where an efficient *upper bound LP* (*u*-LP) is employed to prune unnecessary *t*-LPs by calculating the upper bounds of their optimal values. Specifically, before we solve a *t*-LP with the column generation approach, we can first evaluate its upper bound with a *u*-LP. If the upper bound is smaller than the maximum obtained so far, the current *t*-LP can be simply skipped.

A *u*-LP redefines a *t*-LP on the *marginal coverage* space, and then bounds the feasible marginal coverage space with $O(N)$ linear constraints. The marginal coverage of a target is the probability that this target is covered. Given a mixed strategy $\mathbf{x}$, the corresponding marginal coverage vector is an N-dimensional vector given by $\bar{\mathbf{c}} = \sum_{\mathbf{s}\in\mathcal{S}} x_\mathbf{s} \cdot \Phi(\mathbf{s})$. Using $\bar{\mathbf{c}}$, we can rewrite the attacker's expected utility as

$$\begin{aligned}\sum_{\mathbf{s}\in\mathcal{S}} x_\mathbf{s}\cdot U_a\big(\Phi(\mathbf{s}), i\big) &= \sum_{\mathbf{s}\in\mathcal{S}} x_\mathbf{s}\Big(\phi_i(\mathbf{s})\cdot P_i^a + \big(1-\phi_i(\mathbf{s})\big)\cdot R_i^a\Big)\\ &= \big(P_i^a - R_i^a\big)\sum_{\mathbf{s}\in\mathcal{S}} x_\mathbf{s}\cdot\phi_i(\mathbf{s}) + R_i^a \cdot \sum_{\mathbf{s}\in\mathcal{S}} x_\mathbf{s}\\ &= \big(P_i^a - R_i^a\big)\cdot \bar{c}_i + R_i^a = U_a(\bar{\mathbf{c}}, i),\end{aligned}$$

where $\phi_i(\mathbf{s})$ is the i^{th} component of $\Phi(\mathbf{s})$. Similarly the defender's expected utility can be rewritten as $U_d(\bar{\mathbf{c}}, i)$. Therefore, the *t*-LP for target i^+ is equivalent to:

$$\max_{\bar{\mathbf{c}}} \quad U_d(\bar{\mathbf{c}}, i^+) \tag{14}$$

$$\text{s.t.} \quad U_a(\bar{\mathbf{c}}, i^+) \geq U_a(\bar{\mathbf{c}}, i),\ \forall i \in [N] \tag{15}$$

$$\bar{\mathbf{c}} \in \left\{ \sum_{\mathbf{s}\in\mathcal{S}} x_\mathbf{s}\cdot\Phi(\mathbf{s}) \,\middle|\, \sum_{\mathbf{s}\in\mathcal{S}} x_\mathbf{s}{=}1,\ x_\mathbf{s}{\geq}0 \right\} \tag{16}$$

The last constraint implicitly defines the feasible marginal coverage space. It is unlikely that this constraint admits any compact representation due to the NP-hardness of an SPE problem. Inspired by the MILP formulation of the slave problem, we bound the complex feasible marginal coverage space with the following linear constraints.

$$\bar{\mathbf{c}}, \bar{\mathbf{s}} \in [0,1]^N \tag{17}$$

$$\sum\nolimits_i \bar{s}_i \leq K \tag{18}$$

$$\bar{c}_i \leq \sum\nolimits_j \bar{s}_j \cdot a_{ji},\ \forall i \in [N] \tag{19}$$

$$\bar{c}_i \geq \frac{1}{K}\sum\nolimits_j \bar{s}_j \cdot a_{ji},\ \forall i \in [N] \tag{20}$$

Proposition 3 shows that Eqs. (17)–(20) defines a superset for the feasible marginal coverage space. Thus by replacing Eq. (16) with Eqs. (17)–(20), we obtain an $O(N)$-scale LP, i.e., the *u*-LP, whose optimal value is an upper bound of the optimal value of *t*-LP. Since the *u*-LP runs significantly faster than the column generation approach, and the upper bounds it returns are usually very close to the accurate optimal objective value of *t*-LPs, a more efficient approach we adopt is to calculate upper bounds of all the *t*-LPs first, and then run column generation for the *t*-LPs in descending order of their upper bounds.

Proposition 3. *Let the polytope defined by Eq.* (16) *be* $\bar{\mathcal{C}}$, *and let the polytope of* $\bar{\mathbf{c}}$ *defined by Eqs.* (17)–(20) *be* $\bar{\mathcal{C}}'$. *Then* $\bar{\mathcal{C}} \subseteq \bar{\mathcal{C}}'$.

Experimental Evaluations

Experimental evaluations are provided in this section to examine the performance of algorithms in this paper. All LPs and MILPs are solved with CPLEX (version 12.4). All results are obtained on a machine with a 3.10GHz quad core CPU and 4.00GB memory. We run our algorithm on game instances randomly generated in the following way. For each target i, R_i^d and R_i^a are randomly chosen between 0 and 100, and P_i^d and P_i^a are randomly chosen between -100 and 0. We set all diagonal entries of the adjacency matrix to 1, and set other entries to 1 with probability ρ and to 0 with $1-\rho$. Different settings of ρ are specified in the experiments. All the experimental results are averaged over 50 samples.

Scalability Analysis

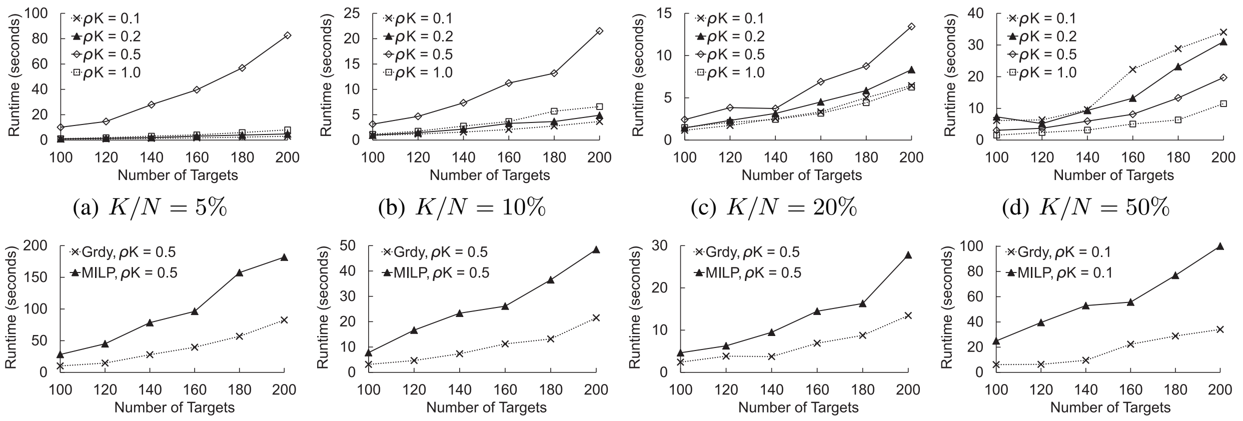
To analyse the scalability of CLASPE we run the algorithm with SPE instances ranging from 100 to 200 targets. We set K to a fixed ratio of N in each group of experiments. We also fix ρK, so that the overall coverage rate offered by the K resources, i.e., $\frac{K\cdot(\rho(N-1)+1)}{N} \approx \rho K + \frac{K}{N}$ is fixed. Various settings of $\frac{K}{N}$ and ρK are tested in the experiments, and the runtime results are depicted in Figure 2:(a)–(d). The results show that our algorithm can solve SPE of real-world problem size very efficiently, with problems of 200 targets being solved in less than 100 seconds in the worst case. One observation from the results is that the computation is most costly when the overall coverage rate is closed to 50%.

Effectiveness of the Greedy Approximation We run another set of experiments, where all slave problems are solved with MILP only. The results are depicted in Figure 2:(e)–(h) (tagged as 'MILP'), in comparison with the results obtained with the greedy approximation (tagged as 'Grdy'). Significant efficiency improvement provided by the greedy algorithm can be seen from the curves. We also record the approximation ratio of the greedy algorithm in Table 1, with the setting $K/N = 5\%$ (results of other settings exhibit similar high ratios). The results suggest high precision and stable performance of the greedy approximation.

Table 1: Approximation ratio of the greedy algorithm

	Number of Targets					
	100	120	140	160	180	200
ρK=0.1	0.999	1.000	1.000	1.000	1.000	1.000
ρK=0.2	0.997	0.996	0.998	0.997	0.997	0.998
ρK=0.5	0.978	0.976	0.982	0.988	0.983	0.986
ρK=1.0	1.000	1.000	1.000	1.000	1.000	1.000

Effectiveness of *u*-LPs We compare solutions of *u*-LPs with the accurate solutions of *t*-LPs computed by column generation. Since a *t*-LP can have negative objective values, we evaluate the approximation ratio of the *u*-LP by



(a) $K/N = 5\%$ (b) $K/N = 10\%$ (c) $K/N = 20\%$ (d) $K/N = 50\%$

(e) $K/N = 5\%$ (f) $K/N = 10\%$ (g) $K/N = 20\%$ (h) $K/N = 50\%$

Figure 2: Scalability of CLASPE: (a)–(d), runtime of CLASPE; (e)–(h), runtime of CLASPE without greedy approximation.

$\frac{U-\min_i P_i^d}{\hat{U}-\min_i P_i^d}$, where U is the accurate optimal objective value of the *t*-LP, and $\hat{U}$ is the upper bound computed by the *u*-LP. The ratio is guaranteed to be in the range $[0, 1]$, and larger values indicate better approximation. The approximation ratios obtained with the setting $K/N = 5\%$ are shown in Table 2 (results of other settings exhibit similar high ratios). We also record the number of *t*-LPs that need to be accurately computed with column generation after pruning (Table 3). Given the high approximation ratios in Table 2, in most cases, only the *t*-LP with the largest upper bound needs to be accurately computed while the others are pruned.

Table 2: Approximation ratio of *u*-LPs

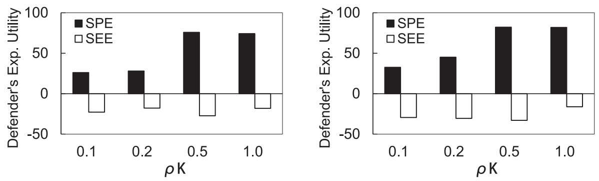
	Number of Targets					
	100	120	140	160	180	200
ρK=0.1	0.999	1.000	1.000	1.000	1.000	1.000
ρK=0.2	0.999	1.000	1.000	1.000	1.000	1.000
ρK=0.5	0.999	0.998	0.999	1.000	1.000	1.000
ρK=1.0	1.000	1.000	1.000	1.000	1.000	1.000

Table 3: Number of *t*-LPs solved by column generation

	Number of Targets					
	100	120	140	160	180	200
ρK=0.1	1.018	1.143	1.000	1.000	1.000	1.000
ρK=0.2	1.009	1.000	1.000	1.000	1.000	1.000
ρK=0.5	1.000	1.000	1.000	1.143	1.000	1.000
ρK=1.0	1.000	1.000	1.000	1.000	1.000	1.000

SPE Outperforms SSE

To show the necessity of considering protection externalities under conditions where they do exist, we evaluate the solution quality of SPEs against SSE solutions. The SSE solutions are the defender's optimal strategies when protection externalities are totally ignored, and are obtained with CLASPE by setting the adjacency matrices to identity matrices. We then compare the defender's expected utilities, in the presence of protection externalities, yielded by the SPE and the SSE solutions. Figure 3 shows the results of small-scale and large-scale settings. Utility losses resulted from ignorance of protection externalities can be easily observed in both figures. The losses are significant even when protection externalities are subtle, i.e., when ρ is small.



(a) $N = 50, K = 5$ (b) $N = 200, K = 20$

Figure 3: Solution quality comparison

Conclusion

This paper provides the following key contributions: 1) We formally define and model SPE. 2) We prove the NP-hardness of solving SPE. 3) To address the NP-hardness, we propose a column generation based algorithm featuring the following key components: i) a MILP formulation for the slave problems; ii) a novel polynomial-time greedy approach providing a constant-factor approximation ratio in the presence of a negative weight, to speed up the computation of slave problems; iii) an $O(N)$-scale *u*-LP that efficiently calculates upper bounds of *t*-LPs for pruning. 4) Experimental evaluation demonstrates that our algorithm can scale up to realistic-sized SPE instances, and shows the importance of considering protection externalities in scenarios where they exist, validating the motivation of this research.

Acknowledgements

This work is supported by NSFC grant No. 61202212 and Singapore MOE AcRF Tier 1 grant MOE RG33/13. This research is also supported in part by Interactive and Digital Media Programme Office, National Research Foundation hosted at Media Development Authority of Singapore (Grant No.: MDA/IDM/2012/8/8-2 VOL 01).

References

An, B.; Pita, J.; Shieh, E.; Tambe, M.; Kiekintveld, C.; and Marecki, J. 2011. GUARDS and PROTECT: Next generation applications of security games. *ACM SIGecom Exchanges* 10(1):31–34.

An, B.; Brown, M.; Vorobeychik, Y.; and Tambe, M. 2013a. Security games with surveillance cost and optimal timing of attack execution. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'13)*, 223–230.

An, B.; Ordóñez, F.; Tambe, M.; Shieh, E.; Yang, R.; Baldwin, C.; DiRenzo III, J.; Moretti, K.; Maule, B.; and Meyer, G. 2013b. A deployed quantal response-based patrol planning system for the us coast guard. *Interfaces* 43(5):400–420.

Basilico, N.; Gatti, N.; and Amigoni, F. 2009. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'9)*, 57–64.

Bertsimas, D., and Tsitsiklis, J. N. 1994. *Introduction to Linear Optimization*. Athena Scientific.

Conitzer, V., and Sandholm, T. 2006. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, 82–90.

Conitzer, V. 2012. Computing game-theoretic solutions and applications to security. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI'12)*, 2106–2112.

Fang, F.; Jiang, A. X.; and Tambe, M. 2013. Optimal patrol strategy for protecting moving targets with multiple mobile resources. In *Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS'13)*, 957–964.

Jain, M.; Kardes, E.; Kiekintveld, C.; Ordónez, F.; and Tambe, M. 2010. Security games with arbitrary schedules: A branch and price approach. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10)*, 792–797.

Kiekintveld, C.; Jain, M.; Tsai, J.; Pita, J.; Ordóñez, F.; and Tambe, M. 2009. Computing optimal randomized resource allocations for massive security games. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'09)*, 689–696.

Korzhyk, D.; Yin, Z.; Kiekintveld, C.; Conitzer, V.; and Tambe, M. 2011. Stackelberg vs. Nash in security games: an extended investigation of interchangeability, equivalence, and uniqueness. *Journal of Artificial Intelligence Research* 41:297–327.

Korzhyk, D.; Conitzer, V.; and Parr, R. 2010. Complexity of computing optimal Stackelberg strategies in security resource allocation games. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10)*, 805–810.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functionsłi. *Mathematical Programming* 14(1):265–294.

Pita, J.; Jain, M.; Marecki, J.; Ordóñez, 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 *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'08)*, 125–132.

Pita, J.; Tambe, M.; Kiekintveld, C.; Cullen, S.; and Steigerwald, E. 2011. GUARDS: game theoretic security allocation on a national scale. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'11)*, 37–44.

Shieh, E. A.; An, B.; Yang, R.; Tambe, M.; Baldwin, C.; DiRenzo, J.; Maule, B.; and Meyer, G. 2012. PROTECT: An application of computational game theory for the security of the ports of the United States. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI'12)*, 2173–2179.

Shieh, E.; Jain, M.; Jiang, A. X.; and Tambe, M. 2013. Efficiently solving joint activity based security games. In *Proceedings of the 23th International Joint Conference on Artificial Intelligence (IJCAI'13)*, 346–352.

Tambe, M. 2011. *Security and game theory: algorithms, deployed systems, lessons learned.* Cambridge University Press.

Tsai, J.; Kiekintveld, C.; Ordonez, F.; Tambe, M.; and Rathi, S. 2009. Iris-a tool for strategic security allocation in transportation networks. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'09)*, 82–90.

Von Stengel, B., and Zamir, S. 2004. Leadership with commitment to mixed strategies.

Xu, H.; Fang, F.; Jiang, A. X.; Conitzer, V.; Dughmi, S.; and Tambe, M. 2014. Solving zero-sum security games in discretized spatio-temporal domains. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14)*.

Yin, Y.; An, B.; and Jain, M. 2014. Game-theoretic resource allocation for protecting large public events. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14)*, 826–834.