Security Games on a Plane

Jiarui Gan[1], Bo An[1], Yevgeniy Vorobeychik[2], Brian Gauch[2]
[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798
[2]Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN 37235
[1]{jrgan, boan}@ntu.edu.sg, [2]{yevgeniy.vorobeychik, brian.gauch}@vanderbilt.edu

Abstract

Most existing models of Stackelberg security games ignore the underlying topology of the space in which targets and defence resources are located. As a result, allocation of resources is restricted to a discrete collection of exogenously defined targets. However, in many practical security settings, defense resources can be located on a continuous plane. Better defense solutions could therefore be potentially achieved by placing resources in a space outside of actual targets (e.g., between targets). To address this limitation, we propose a model called Security Game on a Plane (SGP) in which targets are distributed on a 2-dimensional plane, and security resources, to be allocated on the same plane, protect targets within a certain effective distance. We investigate the algorithmic aspects of SGP. We find that computing a strong Stackelberg equilibrium of an SGP is NP-hard even for zero-sum games, and these are inapproximable in general. On the positive side, we find an exact solution technique for general SGPs based on an existing approach, and develop a PTAS (polynomial-time approximation scheme) for zero-sum SGP to more fundamentally overcome the computational obstacle. Our experiments demonstrate the value of considering SGP and effectiveness of our algorithms.

1 Introduction

Security games have attracted much research attention and have been widely adopted to assist with real security tasks in recent years (Tambe 2011). Much of the research has focused on Stackelberg games—games played between a defender and an attacker in which the attacker is assumed to know the defender's strategy when choosing his own (e.g., Conitzer and Sandholm 2006; Kiekintveld et al. 2009; Basilico, Gatti, and Amigoni 2009; An et al. 2011; Vorobeychik et al. 2014). The goal, from the defender's perspective, is to find a strategy (a probability distribution over resource allocations) that maximizes her utility.

A limitation of existing models of Stackelberg security games is that they ignore the underlying topology of the space in which targets and defence resources are located. As a result, allocation of resources is restricted to a discrete collection of exogenously defined targets. However, in many practical settings, resources can be located on a plane. These include security tasks in large open areas, such as protection of ships or natural resources over a sea area, or aerial monitoring by airplanes or drones. Better defense solutions could be potentially achieved by placing resources in a space outside of actual targets (e.g., between targets).

A natural example is a simple game on a plane with one resource capable of covering a circular area of radius 1 and two identical targets in a distance of 1.5 from each other. To protect both targets simultaneously, we need to move the resource away from the targets to a point between them. If we only consider the targets as candidate locations for resource allocation, we lose half of the coverage and as much of the utility (this can be even worse: consider five targets distributed uniformly along a ring of radius 1). With this motivating example in mind, we aim to extend existing work with an assumed topology of the game space. As we note that most security scenarios are fundamentally planar, we develop a novel model called Security Game on a Plane (SGP), in which targets are distributed on a 2-dimensional plane, and resources, to be allocated on the same plane, protect targets within a certain effective distance.

We investigate the algorithmic aspects of SGP. We find that computing a Strong Stackelberg Equilibrium (SSE) of an SGP is NP-hard even for zero-sum games. Despite the negative results, we provide an exact solution which discretizes the continuous space into regions of equivalent effects and, based on an existing approach, uses *column generation* to tackle the scalability issue. To more fundamentally overcome the complexity barrier, we investigate approximation approaches. We develop a *polynomial-time approximation scheme* (PTAS) for zero-sum SGP so that a solution within any given factor of being optimal can be computed in polynomial time. Notably, unlike approximation schemes used in many existing works which are designed only for certain sub-procedures, such as slave problems in the column generation framework or defender/attacker oracles in the double oracle framework (e.g., Jain, Conitzer, and Tambe 2013; Gan, An, and Vorobeychik 2015; Wang, Yin, and An 2016), the PTAS we propose approximates the entire problem. For general-sum SGP, we show that they are generally hard to approximate due to the inherent lack of robustness of the SSE solution concept. To work around this issue, we explore several realistic restrictions of the problem and find that under these considerations, solutions with

guaranteed quality can still be efficiently computed. Finally, through experimental evaluations, we demonstrate the value of considering SGP compared to existing approaches, as well as effectiveness of the proposed algorithms.

2 Problem Formulation

An SGP is played between a *defender* and an *attacker*. The defender places m identical security resources on a 2-dimensional plane to protect a set of targets $[n] = \{1, \ldots, n\}$ located at (u_i, v_i) for each $i \in [n]$. The attacker chooses a target in $[n]$ to attack. A target is said to be *protected* (or *covered*) if it is within a certain distance of at least one resource, and *unprotected* (or *uncovered*) otherwise. We consider Euclidean distance so that the protection area of a resource is a disk.[1] If the attacker attacks some target $i \in [n]$ which is protected, the defender receives a reward R_i^d and the attacker receives a penalty P_i^a. If instead i is unprotected, payoffs for the defender and the attacker are P_i^d and R_i^a, respectively. We assume $R_i^\phi > P_i^\phi \, \forall i \in [n], \phi \in \{d, a\}$; namely, protecting a target is strictly preferred by the defender but disliked by the attacker. Without loss of generality, we normalize all payoff parameters to be in $[0, 1]$, such that after normalization $\max_{i,\phi} R_i^\phi = 1$ and $\min_{i,\phi} P_i^\phi = 0$. When $R_i^d + P_i^a = 1$ and $P_i^d + R_i^a = 1$ for all $i \in [n]$, the game is said to be *zero-sum*.

We denote by $\mathbf{s} = \langle (x_j, y_j) \rangle_{j=1}^m$ an allocation of resources on the plane, with (x_j, y_j) being the coordinate of the j^{th} resource. Equivalently, we also view $\mathbf{s}$ as a set of m points. We denote by a coverage vector $\mathbf{c} = \langle c_i \rangle$ the protection to each target, with $c_i = 1$ (or 0) representing that target i is protected (or unprotected); and we let $\mathbf{c}(\mathbf{s}) = \langle c_i(\mathbf{s}) \rangle$ be a function mapping a pure strategy to the coverage vector it yields, such that

$$c_i(\mathbf{s}) = \begin{cases} 1, & \text{if } \exists\, j \in [m],\ (u_i, v_i) \in \mathcal{D}(x_j, y_j) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $\mathcal{D}(x, y)$ denotes a disk centered at (x, y). Without loss of generality, we focus on disks of diameter 1 throughout the paper. Given the above definitions, utilities for the defender and the attacker are respectively captured by:

$$U^d(\mathbf{c}, i) = c_i \cdot R_i^d + (1 - c_i) \cdot P_i^d, \tag{2a}$$

$$U^a(\mathbf{c}, i) = c_i \cdot P_i^a + (1 - c_i) \cdot R_i^a. \tag{2b}$$

Following the standard model of Stackelberg security game, the defender plays a *mixed* strategy $\mathbf{p} = \langle p_{\mathbf{s}} \rangle$ in which each *pure* strategy, i.e., an allocation $\mathbf{s}$, is chosen with probability $p_{\mathbf{s}}$; whereas the attacker plays a pure strategy as doing so is sufficient for him to achieve optimality under the leader-follower structure of Stackelberg games. We remark that though resources are to be located on a continuous space, resulting in *infinite* pure strategies, we do not need to consider all of them as those covering the same set of targets are equivalent. The defender's pure strategy space can be defined as $\mathcal{S} = \{\zeta(\mathbf{c}(\mathbf{s})) \mid \mathbf{s} \in \mathbb{R}^{m \times 2}\}$ where $\zeta(\cdot)$ can be any function that gives a pure strategy offering the given coverage. $\mathcal{S}$ is finite since $\mathbf{c}(\mathbf{s}) \in \{0, 1\}^n$. Consequently, there always exists an optimal mixed strategy commitment with a finite *support set* (i.e., the set of pure strategies with non-zero probability). To define the players' utilities for mixed strategies $\mathbf{p}$, we first generalize the coverage function as

$$\mathbf{c}(\mathbf{p}) = \textstyle\sum_{\mathbf{s} \in \mathcal{S}} p_{\mathbf{s}} \cdot \mathbf{c}(\mathbf{s}). \tag{3}$$

It follows that Eq. (2), which calculates players' utilities when coverage is a function of pure strategies $\mathbf{c}(\mathbf{s})$, also applies to mixed strategies $\mathbf{c}(\mathbf{p})$ since

$$U^\phi\big(\mathbf{c}(\mathbf{p}), i\big) = \textstyle\sum_{\mathbf{s} \in \mathcal{S}} p_{\mathbf{s}} \cdot U^\phi\big(\mathbf{c}(\mathbf{s}), i\big) \quad \forall \phi \in \{d, a\}.$$

Strong Stackelberg Equilibrium (SSE) Our goal is to compute the SSE of SGP. In an SSE, the defender chooses the 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). Formally, $\langle \mathbf{p}^*, i^* \rangle$ forms an SSE, iff

$$\mathbf{c}(\mathbf{p}^*) = \arg\max_{\mathbf{c} \in \mathcal{C}} U^d\big(\mathbf{c}, f(\mathbf{c})\big), \text{ and } i^* = f(\mathbf{c}(\mathbf{p}^*)),$$

where $\mathcal{C} = \big\{\mathbf{c}(\mathbf{p}) \big|\, \mathbf{p} \geq \mathbf{0} \wedge \mathbf{1}^\mathsf{T}\mathbf{p} = 1\big\}$ is the set of feasible coverage vectors; $f(\mathbf{c}) = \operatorname{argmax}_{i \in \mathcal{F}(\mathbf{c})} U^d(\mathbf{c}, i)$, where $\mathcal{F}(\mathbf{c}) = \operatorname{argmax}_i U^a(\mathbf{c}, i)$, is the attacker's best response to $\mathbf{c}$.

Throughout this paper, we also refer to the problem of computing SSE of an SGP as SGP. Alternatively, we also view SGP as an optimization problem in which the defender's utility is maximized over the mixed strategy space.

A Linear Program (LP) Formulation An LP formulation can be used to compute the SSE, which enumerates the attacker's responses $i^* \in [n]$ with n LPs. For each i^*, we compute with the following LP the optimal defender strategy under the restriction that the attacker's best response is to attack target i^*.

$$\max_{\mathbf{p}} \quad U^d\big(\mathbf{c}(\mathbf{p}), i^*\big) \tag{4a}$$

$$\text{s.t.} \quad U^a\big(\mathbf{c}(\mathbf{p}), i^*\big) \geq U^a\big(\mathbf{c}(\mathbf{p}), i\big) \qquad \forall i \in [n] \tag{4b}$$

$$\textstyle\sum_{\mathbf{s} \in \mathcal{S}} p_{\mathbf{s}} = 1 \tag{4c}$$

$$p_{\mathbf{s}} \geq 0 \qquad \forall \mathbf{s} \in \mathcal{S} \tag{4d}$$

where Eq. (4b) guarantees that the attacker is indeed incentivized to attack i^*. The LP with the highest optimal value yields an SSE to the game. It remains to specify $\mathcal{S}$ to complete the construction of the LP formulation. We present how this can be done a part of a solution algorithm next.

3 Computing the SSE

To specify $\mathcal{S}$, we draw a circle centered at each target, so that a target gets covered if a resource is placed inside the corresponding circle. The borders of all the circles partition the plane into a collection of disjoint regions, and resources in the same region cover exactly the same set of targets. Therefore, we only need to consider one representative location for each of the regions (e.g., we can take the intersections of the circle borders) and obtain a finite $\mathcal{S}$ with $O(2^{|\mathcal{A}|})$ pure strategies, where $\mathcal{A}$ is the collection of all the regions. However, as $|\mathcal{A}| = O(n^2)$ (Gordon and others 1987), the size of $\mathcal{S}$ still grows exponentially.

[1] Our results extend to rectangular protection areas trivially.

To address the scalability issue, we note that after we introduce the representative locations, the SGP actually degenerates to a Security game with Protection Externalites (SPE)—an existing problem where resources are to be allocated to a given finite set of candidate locations (not necessarily on a plane), such that allocating a resource to each of the candidate location covers an exogenously defined subset of the targets (Gan, An, and Vorobeychik 2015). The existing approach for SPE then applies to SGP, where column generation is used to decompose the large-scale LP as a way to tackle the scalability issue.

Computational Obstacles While the above approach provides a feasible way to break down the large-scale formulation, it does *not* essentially overcome the computational obstacle as computing an SSE of SPE is NP-hard (Gan, An, and Vorobeychik 2015). The question remains whether SGP can be solved or approximated in polynomial time, perhaps in a way quite different from the solution approach for SPE. We find that computing SSE of SGP is NP-hard as well, even for zero-sum games (Theorem 1). Worse still, SGP cannot be approximated efficiently within any constant factor unless P = NP (Theorem 2). Given this, we investigate approximation approaches for *zero-sum* SGP in the next section.

Theorem 1. *Computing SSE of* SGP *is NP-hard even for zero-sum games.*

Proof. We note the known NP-complete problem, *disk covering problem* (DCov) (Masuyama, Ibaraki, and Hasegawa 1981),[2] which asks if n given points on a plane can be covered by m identical disks.

We reduce DCov to SGP. For any DCov instance, we construct an SGP with: n targets located at the n given points in the DCov; m resources; and utilities $R_i^d = R_i^a = 1$ and $P_i^d = P_i^a = 0$ for all targets $i \in [n]$. We solve the SGP and check if the defender receives an expected utility of 1 under SSE. This answers the DCov as: there exists a mixed strategy with utility $1 \Leftrightarrow$ there exists at least one pure strategy (allocation of m disks) covering all targets (points). □

Theorem 2. SGP *cannot be approximated within any constant factor in polynomial time unless P = NP.*

Proof. Suppose an algorithm computes an ϵ-approximate solution to SGP in polynomial time. We show that DCov can be solved with this algorithm, which implies P = NP.

For any DCov instance, we can construct an SGP with: m resources; $n+1$ targets with the first n of them located at the n points given in the DCov, and the remaining one at (M, M) where M is sufficiently large such that no resource can cover (M, M) and any one of the first n targets simultaneously; and player utilities shown in the table above. In this game, the defender can obtain an expected utility of at least $\frac{1}{2}$ iff the attacker attacks target $n+1$. Furthermore, the attacker will indeed attack target $n+1$ iff target $n+1$ is uncovered and the other targets are fully covered (i.e., $c_{n+1} = 0$ and $c_i = 1\,\forall i \in [n]$), which can happen iff there exists a pure strategy covering all targets except $n+1$. In any other cases, the defender receives a utility of at most $\frac{\epsilon}{3}$.

	R_i^a	P_i^a	R_i^d	P_i^d
$i \in [n]$	1	$1/2$	$\epsilon/3$	0
$i{=}n{+}1$	$1/2$	0	1	$1/2$

Therefore, there exists a pure strategy (an allocation of m resources) covering all $i \in [n] \Leftrightarrow$ the defender receives a utility of at least $\frac{1}{2}$ by playing the optimal strategy $\Leftrightarrow$ the ϵ-approximation algorithm offers a solution with defender utility no less than $\frac{\epsilon}{2} > \frac{\epsilon}{3}$. By checking the solution of the approximation algorithm in polynomial time, we obtain an answer to DCov, an NP-complete problem. The same result holds for SGP by reduction from DCov. □

[2]The problem is originally called *Euclidian m-center Problem*.

4 Approximating Zero-sum SGP

Having shown a series of negative results about SGP equilibrium computation, we now present a strong positive result: a PTAS for zero-sum SGP, so that for any fixed parameter $\epsilon > 0$ a $(1-\epsilon)$-approximation can be computed in time polynomial in the input size. We remark that zero-sum security games, though having a restricted payoff structure, still capture a wide range of realistic security scenarios, as the attacker, being adversarial to the defender, usually benefits directly from making the defender worse. The PTAS is almost the best approximation result we can obtain as there exists no *fully PTAS* (FPTAS) for zero-sum SGP unless P = NP (Theorem 3). An FPTAS computes a $(1-\epsilon)$-approximation in time polynomial in both the input size and $\frac{1}{\epsilon}$.

Theorem 3. *Zero-sum* SGP *does not admit any FPTAS unless P = NP.*

Proof. This follows from the *strong NP-hardness* of SGP. As we can see from the proof of Theorem 1, even if all the parameters are bounded by a polynomial in the length of the input, the problem remains to be NP-hard. Such problems are called *strongly NP-hard* problems (Gary and Johnson 1979). Any strongly NP-hard optimization problem with a polynomially bounded objective function cannot have an FPTAS unless P = NP (Vazirani 2013). □

4.1 A PTAS for Zero-sum SGP

The PTAS is based on a grid-shifting approach, which has been used to derive approximations for many planar problems (e.g., Li et al. 2015). Our contribution is to extend the approach to SGP, particularly to the setting of mixed strategies and show that the approximation ratio and the polynomial runtime still hold, which is not trivial based on the existing results.

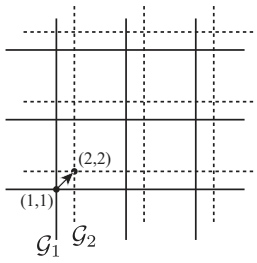

Figure 1

We define a grid $\mathcal{G}_1$ consisting of infinite vertical and horizontal lines distributed uniformly in a distance $l \in \mathbb{Z}_{>0}$, and specifically with two of them intersecting at $(1,1)$, e.g.,

$$\mathcal{G}_1 = \left\{(x,y)\in\mathbb{R}^2 \,|\, x = a\cdot l + 1 \;\vee\; y = a\cdot l + 1,\, a \in \mathbb{Z}\right\}.$$

Shifting $\mathcal{G}_1$ along the vector $(1,1)$ repeatedly, we obtain a series of grids $\mathcal{G}_2,\dots,\mathcal{G}_l$ (Figure 1). A grid divides the plane into infinite disjoint $l\times l$ cells. We assume for simplicity that no targets have an integer coordinate (we can always shift the coordinate system to achieve this), so that all targets are in the interior of the cells. For each grid $\mathcal{G}_k$, we define a filter $\sigma_k(\cdot)$ which, given a pure strategy $\mathbf{s}$, filters out resources in $\mathbf{s}$ whose protection areas overlap $\mathcal{G}_k$, i.e.,

$$\sigma_k(\mathbf{s})=\{(x,y)\in\mathbb{R}^2\,|\,(x,y)\in\mathbf{s}\wedge\mathcal{D}^\circ(x,y)\cap\mathcal{G}_k=\emptyset\},$$

where $\mathcal{D}^\circ(x,y)$ denotes the interior of the disk $\mathcal{D}(x,y)$. Filtering the entire strategy space with $\sigma_k(\cdot)$, we obtain

$$\tilde{\mathcal{S}}_k=\{\sigma_k(\mathbf{s})\,|\,\mathbf{s}\in\mathcal{S}\},$$

with which we define a modified SGP by restricting the strategy space to $\tilde{\mathcal{S}}=\bigcup_{k\in[l]}\tilde{\mathcal{S}}_k$.

Let $\mathbf{p}^*$ and $\tilde{\mathbf{p}}^*$ be the optimal defender strategies in the original and the modified SGPs. Lemmas below complete the construction of the PTAS by showing: (i) $\tilde{\mathbf{p}}^*$ offers a $\big(1-\frac{2}{l}\big)$-approximation to $\mathbf{p}^*$ (Lemma 5); and (ii) a defender strategy at least as good as $\tilde{\mathbf{p}}^*$ can be computed in polynomial time (Lemma 6). We conclude with Theorem 7.

Lemma 4. *For any defender mixed strategy $\mathbf{p}$, there exists a mixed strategy $\tilde{\mathbf{p}}$ with support set $\Delta(\tilde{\mathbf{p}})\subset\tilde{\mathcal{S}}$ such that*

$$\big(1-\tfrac{2}{l}\big)\cdot\mathbf{c}(\mathbf{p})\le\mathbf{c}(\tilde{\mathbf{p}})\le\mathbf{c}(\mathbf{p}).$$

Proof. For any given $\mathbf{p}$, we construct a strategy $\tilde{\mathbf{p}}=\langle\tilde{p}_{\mathbf{s}}\rangle$ such that $\tilde{p}_{\sigma_k(\mathbf{s})}=\frac{1}{l}\cdot p_{\mathbf{s}}$ for all $k\in[l]$ and all $\mathbf{s}\in\Delta(\mathbf{p})$. In this way, we have $\Delta(\tilde{\mathbf{p}})\subset\tilde{\mathcal{S}}$ and

$$\begin{aligned}\mathbf{c}(\tilde{\mathbf{p}})&=\sum_{\mathbf{s}\in\Delta(\mathbf{p})}\sum_{k\in[l]}\tfrac{1}{l}\cdot p_{\mathbf{s}}\cdot\mathbf{c}\big(\sigma_k(\mathbf{s})\big)\\&=\sum_{\mathbf{s}\in\Delta(\mathbf{p})}p_{\mathbf{s}}\cdot\mathbf{c}(\mathbf{s})-\sum_{\mathbf{s}\in\Delta(\mathbf{p})}\sum_{k\in[l]}\tfrac{p_{\mathbf{s}}}{l}\cdot\Big(\mathbf{c}(\mathbf{s})-\mathbf{c}\big(\sigma_k(\mathbf{s})\big)\Big)\\&=\mathbf{c}(\mathbf{p})-\sum_{\mathbf{s}\in\Delta(\mathbf{p})}\tfrac{p_{\mathbf{s}}}{l}\sum_{k\in[l]}\Big(\mathbf{c}(\mathbf{s})-\mathbf{c}\big(\sigma_k(\mathbf{s})\big)\Big).\end{aligned}\tag{5}$$

We have $\mathbf{c}(\mathbf{s})\ge\mathbf{c}\big(\sigma_k(\mathbf{s})\big)$ as $\mathbf{s}$ covers more targets, so that Eq. (5) implies $\mathbf{c}(\tilde{\mathbf{p}})\le\mathbf{c}(\mathbf{p})$. To derive the lower bound, we observe that the interior of the protection area of a resource intersects at most two of $\mathcal{G}_1,\dots,\mathcal{G}_l$. Thus, for arbitrary $\mathbf{s}$,

$$\textstyle\sum_{k\in[l]}\Big(\mathbf{c}(\mathbf{s})-\mathbf{c}\big(\sigma_k(\mathbf{s})\big)\Big)\le 2\cdot\mathbf{c}(\mathbf{s}).$$

Again with Eq. (5), we have

$$\mathbf{c}(\tilde{\mathbf{p}})\ge\mathbf{c}(\mathbf{p})-\textstyle\sum_{\mathbf{s}\in\Delta(\mathbf{p})}\frac{p_{\mathbf{s}}}{l}\cdot 2\cdot\mathbf{c}(\mathbf{s})\ge\big(1-\frac{2}{l}\big)\cdot\mathbf{c}(\mathbf{p}),$$

which completes the proof. □

Lemma 5. $\frac{U(\mathbf{c}(\tilde{\mathbf{p}}^*))}{U(\mathbf{c}(\mathbf{p}^*))}\ge 1-\frac{2}{l}$ *if the* SGP *is zero-sum, where* $U(\mathbf{c})=U^d\big(\mathbf{c},f(\mathbf{c})\big)$.

Proof. According to Lemma 4, there exists a $\tilde{\mathbf{p}}$ such that $\mathbf{c}(\tilde{\mathbf{p}})\ge\big(1-\frac{2}{l}\big)\cdot\mathbf{c}(\mathbf{p}^*)$. A feature of zero-sum security games is that adding more coverage to a solution does not make the solution worse for the defender, i.e., $\mathbf{c}\ge\mathbf{c}'\Rightarrow U(\mathbf{c})\ge U(\mathbf{c}')$ for any coverage vectors $\mathbf{c}$ and $\mathbf{c}'$. Therefore

$$U\big(\mathbf{c}(\tilde{\mathbf{p}}^*)\big)\ge U\big(\mathbf{c}(\tilde{\mathbf{p}})\big)\ge U\left(\big(1-\tfrac{2}{l}\big)\cdot\mathbf{c}(\mathbf{p}^*)\right),$$

$$\begin{aligned}\Rightarrow\frac{U\left(\mathbf{c}(\tilde{\mathbf{p}}^*)\right)}{U\left(\mathbf{c}(\mathbf{p}^*)\right)}&\ge\frac{U\left(\big(1-\frac{2}{l}\big)\cdot\mathbf{c}(\mathbf{p}^*)\right)}{U\left(\mathbf{c}(\mathbf{p}^*)\right)}\\&=\frac{P^d_{i^*}+(R^d_{i^*}-P^d_{i^*})\cdot\big(1-\frac{2}{l}\big)\cdot c_{i^*}(\mathbf{p}^*)}{P^d_{i^*}+(R^d_{i^*}-P^d_{i^*})\cdot c_{i^*}(\mathbf{p}^*)}\\&\ge\frac{(R^d_{i^*}-P^d_{i^*})\cdot\big(1-\frac{2}{l}\big)\cdot c_{i^*}(\mathbf{p}^*)}{(R^d_{i^*}-P^d_{i^*})\cdot c_{i^*}(\mathbf{p}^*)}=1-\frac{2}{l}.\ \square\end{aligned}$$

Lemma 6. *A defender strategy at least as good as $\tilde{\mathbf{p}}^*$ can be computed in time polynomial in n.*

Proof. $\tilde{\mathbf{p}}^*$ can be computed by Eqs. (4) with $\tilde{\mathcal{S}}$ replacing $\mathcal{S}$ as the pure strategy space.[3] To deal with the large variable number, we consider its dual problem structured as follows.

$$\begin{aligned}\min_{\mathbf{q}}\quad&\textstyle\sum_{i\in[n]}(R^a_{i^*}-R^a_i)\cdot q_i&&\text{(6a)}\\\text{s.t.}\quad&\textstyle\sum_{i\in[n]}\Big(c_i(\mathbf{s})\cdot Q^a_i-c_{i^*}(\mathbf{s})\cdot Q^a_{i^*}\Big)\cdot q_i+q_{n+1}\\&\le-c_{i^*}(\mathbf{s})\cdot Q^d_{i^*}&\forall\,\mathbf{s}\in\tilde{\mathcal{S}}\ &\text{(6b)}\\&q_i\ge 0&\forall\,i\in[n]\ &\text{(6c)}\end{aligned}$$

where $Q^\phi_i=R^\phi_i-P^\phi_i\ \forall i\in[n],\phi\in\{d,a\}$. The formulation has only $n+1$ variables but as many as $|\tilde{\mathcal{S}}|+n$ constraints. We use the *ellipsoid method* (Bertsimas and Tsitsiklis 1994) to deal with the constraints, with which the problem reduces to implementing the *separation oracle*, i.e., a procedure that, for any given $\mathbf{q}$, decides whether all constraints are satisfied or not, and, if not, find out one of the violated constraints.

The key is to verify satisfiability to constraints in Eq. (6b). We observe that for a given $\mathbf{q}$, Eq. (6b) can be rewritten as

$$w(\mathbf{s}):=\textstyle\sum_{i\in[n]}w_i\cdot c_i(\mathbf{s})+w_{n+1}\le 0,$$

where $w_i=q_i\cdot Q^a_i\ \forall i\ne i^*$ and $w_{i^*}=Q^d_{i^*}-\sum_{i\ne i^*}q_i\cdot Q^a_i$ are all constants. Since $\tilde{\mathcal{S}}=\bigcup_k\tilde{\mathcal{S}}_k$, to implement the separation oracle is equivalent to check if $\max_{\mathbf{s}\in\tilde{\mathcal{S}}_k}w(\mathbf{s})>0$ for some $k\in[l]$, where $\max_{\mathbf{s}\in\tilde{\mathcal{S}}_k}w(\mathbf{s})$ can be furthermore interpreted as the Maximum Budgeted Coverage problem (MBC): given n weighted elements with weights $w_1,\dots,w_n$ and a collection $\mathcal{A}$ of subsets of the elements, find m subsets such that the sum of weights in the union of the subsets is maximized. Here $\mathcal{A}$ corresponds to the collection of regions defined in Section 3.

As the plane is detached into independent cells by the grids, allocation in one cell does not affect that in the other cells. Given this, we use dynamic programming to tackle the problem. For a grid $\mathcal{G}_k$, let $cell_1,\dots,cell_{\tilde{n}}$ ($\tilde{n}\le n$) be the cells containing at least one target, and let $\mathsf{A}(\eta,j)$ be the maximum weights we can cover with j resources in $cell_1,\dots,cell_\eta$. Our goal is to compute $\mathsf{A}(\tilde{n},m)$, which can be done with the following recursion:

$$\mathsf{A}(\eta,j)=\begin{cases}\max_{j'=0}^{\tilde{m}}\big(\mathsf{A}(\eta-1,j-j')+\mathsf{MBC}(cell_\eta,j')\big),&\text{if }\eta>1\\\mathsf{MBC}(cell_1,j),&\text{if }\eta=1\end{cases}$$

where $\mathsf{MBC}(cell_\eta,j)$ is the maximum weights we can cover with j resources in $cell_\eta$; and $\tilde{m}$ is a cap of the number of

[3] Zero-sum security games admit a more concise single-LP formulation (Xu 2016). This does not change the nature of the proof. To be more general, we prove with the multiple-LP formulation.

resources needed to achieve optimality in a single cell. Here if we allow the protection areas of resources to cross the cell border, we have $\tilde{m} = 2l^2$ as an $l \times l$ square can be fully covered with $2l^2$ disks.[4] We do so as this actually expands the pure strategy space, resulting in a solution at least as good as $\tilde{\mathbf{p}}^*$. Therefore, given the number $O(n^2)$ of candidate locations, we can enumerate all $O(n^{4 \cdot l^2})$ possible allocations to find $\mathsf{MBC}(cell_\eta, j)$, which is polynomial in n. This implies that $\mathsf{A}(\tilde{n}, m)$, or the separation oracle, are polynomial-time computable, which concludes the proof. □

Theorem 7. *For any given $l \in \mathbb{Z}_{>0}$, a $\left(1 - \frac{2}{l}\right)$-approximate solution to* SGP *can be computed in time polynomial in n.*

Proof. This follows readily from Lemmas 5 and 6. □

4.2 Implementation Considerations

The PTAS is highly theoretical in two aspects: first, it relies on the ellipsoid method which, while having a theoretical guarantee of polynomial runtime, is notorious for being inefficient for practical uses; and second, the runtime of the separation oracle has a large exponent on n. These make the PTAS inefficient for practical uses. For better performance, we propose the following practical remedies.

Column Generation (CG) We use CG to replace the ellipsoid method. The slave problem of CG is exactly the same as the separation oracle of the ellipsoid method and is polynomial-time solvable. Note this does not mean that CG has polynomial runtime as there is no guarantee for the number of iterations CG needs. However, CG does exhibit better practical performance, which is similar to the relationship between the ellipsoid method and the simplex method in solving LPs—the latter does not guarantee polynomial runtime but is generally much more efficient in practice.

Greedy Approximation for the Slave Problem Since the ultimate goal of the slave problem is to find a column with a positive reduced cost, there is no need to maximize the reduced cost all the time. Approximations to the slave problem, which find "better" (instead of the best) columns, can be used to simplify the computation. Only when the approximation fails to find a column with a positive reduced cost, we call the exact algorithm for the best column.

We use a cell-wise greedy algorithm. The rough idea is to iteratively place one resource to the location where the most marginal weights are covered until all resources are used. To make use of the cell structure, a within-cell rank of candidate locations is maintained for each cell by the marginal weights of the locations; and so is a between-cell rank of cells by the marginal weights of the best locations in the cells. In each iteration, a best location is chosen from the best cell in the between-cell rank, and then the between-cell rank and only the within-cell rank of the chosen cell are updated. Such a bi-level implementation is faster than a direct implementation on the entire collection of the candidate locations.

[4] While $w_i \geq 0 \, \forall i \neq i^*$, w_{i^*} can be negative. In this case, we may not want to cover i^*. It is easy to see that using a constant number of additional disks, we can leave a small hole around i^* when covering the cell, so that $\tilde{m}$ is still bounded by a constant.

An MILP for MBC To compute the exact solution of $\mathsf{MBC}(cell_\eta, j)$, instead of enumerating all the $O(n^{4 \cdot l^2})$ possible allocations, a more practical approach is to formulate it as a *mixed integer linear program* (MILP). Many mature algorithms or solvers for MILP can then be used to solve the problem. The MILP formulation can be found in the previous work (Gan, An, and Vorobeychik 2015).

5 Extension to General-sum SGP

We have shown that general-sum SGP is hard even to approximate (Theorem 2). Further analysis implies that the inapproximability is due to the inherent lack of robustness of the SSE solution concept: when the game is non-zero-sum, the defender's utility function is non-continuous with respect to the defender strategy. This means that even if we can obtain a coverage vector arbitrarily close to the optimal one (indeed, we can, as Lemmas 4 and 6 also apply to general-sum SGP), the objective value we get does not necessarily converge to the optimum. For example, given one resource and two targets such that $R_1^a = 1$, $P_1^a = \frac{1}{2}$ and $R_2^a = \frac{1}{2}$, $P_2^a = 0$, only when the coverage is exactly $c_1 = 1$ and $c_2 = 0$, the attacker is incentivized to attack target 2. Despite the discouraging result, we point out that such special instances are rarely seen in practice. In most cases if we apply the PTAS for zero-sum SGP to general-sum SGPs we still obtain solutions of good quality. We present several theoretical observations to justify an optimistic outlook.

Quasi-zero-sum Games As we point out previously, in real scenarios, the attacker normally benefits more from attacking targets with higher values to the defender. That is, the payoff structure, if not completely zero-sum, exhibits strong negative correlation between the players. We say that a game is $(1-\epsilon)$-*quasi-zero-sum* if $|R_i^a + P_i^d - 1| \leq \epsilon$ and $|R_i^d + P_i^a - 1| \leq \epsilon$ for all $i \in [n]$. Theorem 9 shows that the PTAS for zero-sum games yields a solution to quasi-zero-sum games with a bounded absolute error.

Lemma 8. *If a security game is $(1-\epsilon)$-quasi-zero-sum, then for any pair of coverage vectors $\mathbf{c}$ and $\mathbf{c}'$ such that $\mathbf{c}' \geq \mathbf{c} - \delta$, $U^d\big(\mathbf{c}', f(\mathbf{c}')\big) \geq U^d\big(\mathbf{c}, f(\mathbf{c})\big) - (2\epsilon + \delta)$.*

Proof. We have, for *any* coverage vector $\mathbf{c}$ and $i \in [n]$

$$\begin{aligned} U^d(\mathbf{c}, i) + U^a(\mathbf{c}, i) &= c_i \cdot \big(R_i^d + P_i^a\big) + (1 - c_i)\big(P_i^d + R_i^a\big) \\ &\geq c_i \cdot (1-\epsilon) + (1 - c_i) \cdot (1 - \epsilon) \\ &= 1 - \epsilon \end{aligned} \tag{7}$$

where the second line follows from the definition of $(1-\epsilon)$-quasi-zero-sum game. Similarly,

$$U^d(\mathbf{c}, i) + U^a(\mathbf{c}, i) \leq 1 + \epsilon \tag{8}$$

$$\begin{aligned} \Rightarrow U^d\big(\mathbf{c}', f(\mathbf{c}')\big) &\geq U^d\big(\mathbf{c} - \delta, f(\mathbf{c}')\big) \\ &\geq U^d\big(\mathbf{c}, f(\mathbf{c}')\big) - \delta && \text{(by Eq. (2a))} \\ &\geq 1 - \epsilon - U^a\big(\mathbf{c}, f(\mathbf{c}')\big) - \delta && \text{(by Eq. (7))} \\ &\geq 1 - \epsilon - U^a\big(\mathbf{c}, f(\mathbf{c})\big) - \delta \\ &\geq U^d\big(\mathbf{c}, f(\mathbf{c})\big) - 2\epsilon - \delta. && \text{(by Eq. (8))} \end{aligned}$$ □

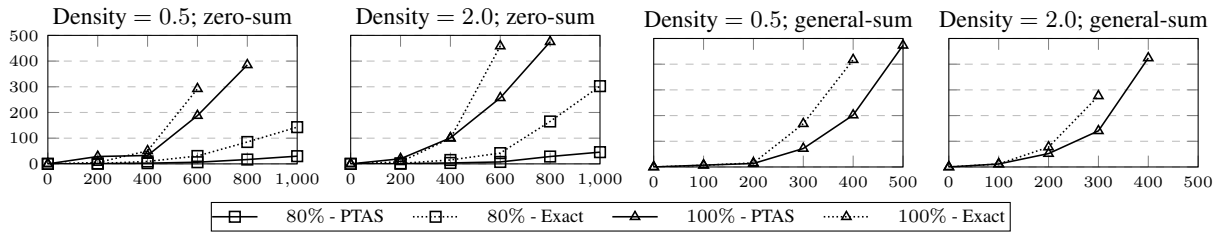

Figure 2: Runtime (y-axis: time in seconds; x-axis: number of targets)

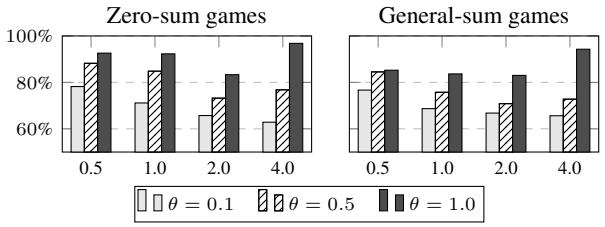

Figure 3: Defender utilities obtained with the existing SPE model as ratios to those obtained with SGP (x-axis: target density)

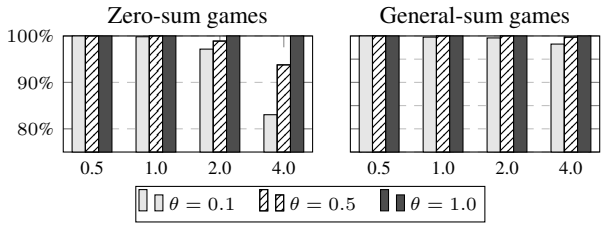

Figure 4: Approximation ratio of solutions obtained with the PTAS (x-axis: target density)

Theorem 9. *Suppose an* SGP *is* $(1-\epsilon)$*-quasi-zero-sum. For any given* $l \in \mathbb{Z}_{>0}$*, a solution* $\tilde{\mathbf{p}}^*$ *can be computed in polynomial time such that* $U\big(\mathbf{c}(\mathbf{p}^*)\big) - U\big(\mathbf{c}(\tilde{\mathbf{p}}^*)\big) \leq 2\epsilon + \frac{2}{l}$*, where* $\mathbf{p}^*$ *is the optimal solution and* $U(\mathbf{c}) = U^d\big(\mathbf{c}, f(\mathbf{c})\big)$.

Proof. By Lemmas 4 and 6, we can obtain in polynomial time a solution $\tilde{\mathbf{p}}^*$ such that $\mathbf{c}(\tilde{\mathbf{p}}^*) \geq \big(1 - \frac{2}{l}\big) \cdot \mathbf{c}(\mathbf{p}^*) \geq \mathbf{c}(\mathbf{p}) - \frac{2}{l}$. By Lemma 8, we obtain the desired result with $\delta = \frac{2}{l}$. □

Robustness Considerations Alternatively, we may consider eliminating unstable solutions on which a slight deviation of the defender may cause the attacker to change his target and result in a large difference to the defender's utility. This is important as in practice the defender can rarely implement her strategy precisely as planned; nor can the attacker observe precisely the mixed strategy the defender implements. We add a buffer ε to Eq. (4b):

$$U^a\big(\mathbf{c}(\mathbf{p}), i^*\big) \geq U^a\big(\mathbf{c}(\mathbf{p}), i\big) + \varepsilon \quad \forall i \in [n].$$

The solution is then robust to errors causing ε loss in the attacker's utility. We call such a solution an *ε-robust solution*. Theorem 10 shows that if we limit our solution to the robust ones, the PTAS can still be used to find approximate solutions with only a slight loss of robustness.

Theorem 10. *For an* SGP *and a given* $l \in \mathbb{Z}_{>0}$*, if there exists an* ε*-robust solution with* $\varepsilon > \frac{2}{l}$*, then an* $\big(\varepsilon - \frac{2}{l}\big)$*-robust solution can be computed in polynomial time, which offers at least* $1 - \frac{2}{l}$ *the utility of the optimal* ε*-robust solution.*

Proof. Let $\mathbf{p}^*$ denote the optimal ε-robust solution. By Lemma 4, there exists a solution $\tilde{\mathbf{p}}^*$ with support set $\Delta(\tilde{\mathbf{p}}) \subset \tilde{S}$ such that $\big(1 - \frac{2}{l}\big) \cdot \mathbf{c}(\mathbf{p}^*) \leq \mathbf{c}(\tilde{\mathbf{p}}^*) \leq \mathbf{c}(\mathbf{p}^*)$. Let i^* be the attacker's best response to $\mathbf{p}^*$, i.e., $i^* = f(\mathbf{c}(\mathbf{p}^*))$. We have

$$\begin{aligned} U^a\big(\mathbf{c}(\tilde{\mathbf{p}}^*), i^*\big) &\geq U^a\left(\big(1 - \tfrac{2}{l}\big) \cdot \mathbf{c}(\mathbf{p}^*),\ i^*\right) \\ &\geq U^a\big(\mathbf{c}(\mathbf{p}^*),\ i^*\big) - \tfrac{2}{l} \\ &\geq U^a\big(\mathbf{c}(\mathbf{p}^*),\ i\big) + \varepsilon - \tfrac{2}{l} \text{ (as } \mathbf{p}^* \text{ is } \varepsilon\text{-robust)} \\ &\geq U^a\big(\mathbf{c}(\tilde{\mathbf{p}}^*), i\big) + \varepsilon - \tfrac{2}{l} \quad \forall i \in [n], \end{aligned}$$

which indicates that $\tilde{\mathbf{p}}^*$ is $\big(\varepsilon - \frac{2}{l}\big)$-robust and the attacker's best response to $\tilde{\mathbf{p}}^*$ is i^*. Therefore, combining Lemma 6, we can obtain in polynomial time an $\big(\varepsilon - \frac{2}{l}\big)$-robust solution which is at least as good as $\tilde{\mathbf{p}}^*$ (by solving the LP formulation with a buffer $\varepsilon - \frac{2}{l}$ added to Eq. (4b)). It follows that (similar to the proof of Lemma 5)

$$\frac{U^d\left(\mathbf{c}(\tilde{\mathbf{p}}^*), i^*\right)}{U^d\left(\mathbf{c}(\mathbf{p}^*), i^*\right)} \geq \frac{U^d\left(\big(1 - \frac{2}{l}\big) \cdot \mathbf{c}(\mathbf{p}^*),\ i^*\right)}{U^d\left(\mathbf{c}(\mathbf{p}^*), i^*\right)} \geq 1 - \frac{2}{l}. \quad \square$$

In addition to the above theoretical observations, our experimental results in Section 6 again corroborate the effectiveness of the PTAS on general SGP.

6 Experimental Evaluations

We experimentally evaluate the proposed model and the algorithms. All results are obtained on a platform with a 3.2 GHz CPU and 16 GB memory. All LPs and MILPs are solved using the existing solver CPLEX (version 12.4).

Performance of Algorithms We compare the runtime of our exact algorithm and the PTAS on both zero-sum and general-sum games. In the experiments, target coordinates are randomly uniformly generated in $[0, \sqrt{n/\rho}]$ for different target densities ρ. Player payoffs are randomly uniformly generated in $[0, 1]$. The comparison is shown in Figure 2, where the result is obtained with $l = 10$ in the PTAS, which guarantees an approximation ratio of 80%. We can

see that while the PTAS runtime is comparable to the exact approach for small instances, it begins to exhibit significant improvements as scale increases, particularly for non-zero-sum games. Moreover, we see that PTAS is significantly faster in reaching its theoretical approximation ratio of 80% than the exact algorithm. This is useful as we can maintain an upper bound of the solution (which can be done with an LP relaxation for MBC (Gan, An, and Vorobeychik 2015)) and terminate the algorithm for faster performance when the solution reaches desired ratio to the upper bound.

Improvement of Solution Quality with SGP We evaluate improvement of solution quality with SGP as compared with the existing SPE model where resource allocation is restricted to targets. The results, as shown in Figure 3, are obtained with instances of 100 targets and 20 resources (the numbers of targets and resources do not have a significant affect on results). The parameter θ defines the variance of the players' payoffs over different targets, with which we first generate the penalty P_i^ϕ in $[0, \theta]$, and then the reward in $[P_i^\phi + 1 - \theta, 1]$ (when $\theta = 1$ the approach is equivalent to the payoff generation model described above; when $\theta = 0$, all targets are identical). This is associated with an interesting observation that when targets have similar payoffs, the gap between solutions of SGP and SPE increases. On the other hand, when payoff variance is large, SPE exhibits good solution quality, potentially being used as a heuristic to SGP.

Solution Quality of the Approximation Approach We evaluate how well PTAS actually approximates solutions in both zero-sum and non-zero-sum settings. As shown in Figure 4, the PTAS yields nearly optimal solution in most runs, and does so even for general-sum games.

7 Conclusions

This paper aims at addressing the limitation of existing models of Stackelberg security games that ignore the underlying topology of the space in which targets and defence resources are or are to be located. A novel model SGP, which incorporates a planar topology, is proposed and studied. Hardness results are established that computing SSE of SGP is NP-hard and is generally hard to approximate. Despite of these results, an PTAS is found and implemented for zero-sum SGPs, which in practice also offers solutions with good quality to general-sum SGPs. Experimental results show the improvement of solution quality with the SGP model, as well as the effectiveness of the proposed algorithms.

Acknowledgement

This research is partially supported by NRF2015NCR-NCR003-004, NAP, NSF (CNS-1238959, CNS-1640624, IIS-1526860), ARO (W911NF-16-1-0069), ONR (N00014-15- 1-2621), and AFRL (FA8750-14-2-0180).

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.

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)*, 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.

Gan, J.; An, B.; and Vorobeychik, Y. 2015. Security games with protection externalities. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, 914–920.

Gary, M. R., and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman and Company, New York.

Gordon, B., et al. 1987. *Challenging Mathematical Problems with Elementary Solutions*, volume 1. Courier Corporation.

Jain, M.; Conitzer, V.; and Tambe, M. 2013. Security scheduling for real-world networks. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 215–222.

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)*, 689–696.

Li, J.; Wang, H.; Zhang, B.; and Zhang, N. 2015. Linear time approximation schemes for geometric maximum coverage. In *Computing and Combinatorics*. Springer. 559–571.

Masuyama, S.; Ibaraki, T.; and Hasegawa, T. 1981. The computational complexity of the m-center problems on the plane. *IEICE Transactions (1976-1990)* 64(2):57–64.

Tambe, M. 2011. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press.

Vazirani, V. V. 2013. *Approximation algorithms*. Springer Science & Business Media.

Von Stengel, B., and Zamir, S. 2004. Leadership with commitment to mixed strategies. *CDAM Research Report LSE-CDAM-2004-01*.

Vorobeychik, Y.; An, B.; Tambe, M.; and Singh, S. P. 2014. Computing solutions in infinite-horizon discounted adversarial patrolling games. In *ICAPS*, 314–322.

Wang, Z.; Yin, Y.; and An, B. 2016. Computing optimal monitoring strategy for detecting terrorist plots. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, 637–643.

Xu, H. 2016. The mysteries of security games: Equilibrium computation becomes combinatorial algorithm design. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, 497–514.