Equilibrium Refinement in Security Games with Arbitrary Scheduling Constraints

Kai Wang[1], Qingyu Guo[2], Phebe Vayanos[1], Milind Tambe[1], Bo An[2]
[1]Center for Artificial Intelligence in Society, University of Southern California
[2]School of Computer Science and Engineering, Nanyang Technological University
{wang319,phebe.vayanos,tambe}@usc.edu,{qguo005,boan}@ntu.edu.sg

ABSTRACT

Significant research effort in security games has focused in devising strategies that perform well even when the attacker deviates from optimal (rational) behavior. In most of these frameworks, a price needs to be paid to ensure robustness against this unpredictability. However, equilibrium refinement is an attractive alternative to boost solution robustness at no cost even though it has not received as much attention in security game literature. In this framework, resources are strategically allocated to secure an optimal outcome against a rational adversary while simultaneously protecting other targets to ensure good outcomes against boundedly rational or constrained attackers. Unfortunately, existing approaches for equilibrium refinement in security games cannot effectively address scheduling constraints that arise frequently in real-world applications. In this paper, we aim to fill this gap and make several key contributions. First, we show that existing approaches for equilibrium refinement can fail in the presence of scheduling constraints. Second, we investigate the properties of the best response of the attacker. Third, we leverage these properties to devise novel iterative algorithms to compute the optimally refined equilibrium, with polynomially many calls to an LP oracle for zero-sum games. Finally, we conduct extensive experimental evaluations that showcase *i)* the superior performance of our approach in the face of a boundedly rational attacker and *ii)* the attractive scalability properties of our algorithm that can solve realistic-sized instances.

KEYWORDS

Equilibrium refinement; security games; arbitrary scheduling constraints; strong Stackelberg equilibrium

ACM Reference Format:
Kai Wang[1], Qingyu Guo[2], Phebe Vayanos[1], Milind Tambe[1], Bo An[2]. 2018. Equilibrium Refinement in Security Games with Arbitrary Scheduling Constraints. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION

Stackelberg Security Games (SSG) have been successfuly applied in a variety of domains to optimize the use of limited security resources against a strategic adversary, with examples such as ARMOR for airport security [16], IRIS for security of flights [8], ports [20] and border [3, 11] patrolling, traffic enforcement [17, 18], and transit network [23]. In SSG, the defender (security agencies) protects targets using limited security resources, but allocation of resources to targets must obey many scheduling constraints. For example, some resources may be prohibited from being assigned to certain targets or may be able to cover several targets at the same time. After conducting surveillance of the defender strategy, the strategic attacker (terrorists/criminals) may respond with an optimal attack.

The standard solution concept adopted by SSG is the Strong Stackelberg Equilibrium (SSE) [13, 24]. Significant research in SSG has focused on providing efficient algorithms to compute SSE under various constraints [8, 21]. Recently, significant research efforts have focused on devising strategies that perform well even under uncertainty in the adversary behavior. For example, [15, 25] investigate adversary bounded rationality, [9] considers execution uncertainty, and [26] focuses on observational uncertainty. In most of these frameworks, the defender either pays a price or slightly sacrifices her first priority target to ensure robustness against unpredictability in the adversary's behavior. However, *equilibrium refinement* is an attractive alternative to provide robustness at no cost by choosing, among all SSEs, the one that performs best in all possible events although it has not received as much attention in the security game literature.

In most real-world applications, security resources must be allocated in the presence of scheduling constraints. This is the case for example of the Federal Air Marshal Service [8], cyber security [14, 22], network security [10, 19], and more generally in domains where security resources exhibit protection externalities [5, 7]. Yet, existing algorithms for equilibrium refinement in security games do not apply in the presence of such constraints. The presence of scheduling constraints complicates the problem of equilibrium refinement significantly, since multiple equilibria are the norm for security games with schedules, and even finding an arbitrary SSE [12] is already a challenging task and prevents the adoption of existing techniques [1] in our problem. To the best of our knowledge, the only paper to investigate the problem of equilibrium refinement under scheduling constraints is [6], wherein a heuristic algorithm is proposed to conduct equilibrium refinement in the spatio-temporal domain. While the paper provides a significant step in our research direction, it only addresses a special case of scheduling constraints. In fact, we are not aware of any algorithm that can cater for *arbitrary scheduling constraints* in security games to provide an optimal refined equilibrium.

In this paper, we focus on the equilibrium refinement on Security Problems with ARbitrary Schedules (SPARS) [8] , where we assign each resource to cover one schedule and each schedule can cover multiple targets. We follow the same dominance criteria mentioned in [1] and introduce a counterexample showing that in the presence of scheduling constraints, their method fails to return a

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

non-dominated equilibrium. We propose a new method to analyze the topology of the attacker's best response. This analysis provides us with key insights into the structure of multiple equilibria. Leveraging these insights, we introduce a new iterative (resp. recursive) algorithm that successfully returns the non-dominated solution of zero-sum (resp. general-sum) SPARS. We show that in the worst case, our iterative algorithm only necessitates $O(n^3)$ calls to an LP oracle, where n corresponds to the number of targets and an LP oracle could be either a linear program solver or a column generation method used to approximate the optimal solution. For the general-sum games, our recursive algorithm successfully provides the optimal solution with $O(n^3)$ oracle calls for each subproblem.

Our experimental results demonstrate significant improvement on the robustness of our computed solution over existing approaches which also serves to showcase the benefit of equilibrium refinement on SPARS. Moreover, our computations show the average number of oracle calls is $O(n^2)$ in both zero-sum and general-sum cases, illustrating practical scalability of our approach.

2 SECURITY GAMES WITH ARBITRARY SCHEDULES

In this work, we consider SPARS [8]. This is a two-player Stackelberg game played between an attacker and a defender. The attacker's pure strategy space is the set of targets T that could be attacked, $T = \{t_1, \ldots, t_n\}$. The attacker's corresponding mixed strategy $\mathbf{a} = \langle a_i \rangle_{i=1}^n$ is a vector where a_i represents the probability of attacking t_i. To protect targets, the defender has at her disposal a collection of resources indexed by $r \in R$, where the set R collects all resources. Each resource r can be assigned to a *schedule* $s \subseteq T$ that covers multiple targets. Associated with each resource r is the set of all possible schedules $S_r \subseteq \mathcal{P}(T)$ to which it can be assigned. For notational convenience, we assume that $\emptyset \in S_r$ so that a resource that is assigned to $\emptyset$ is effectively unused.

The defender's pure strategy space J is the set of all joint schedules that assign each resource to exactly one schedule. Thus,

$$J = \{j \subseteq T \ : \ j = \cup_{r \in R} s_r, \ s_r \in S_r\}$$

and target $t \in T$ is covered by the joint schedule $j \in J$ if and only if $t \in j$. For any joint schedule, a target can be covered by more than one schedule, and a target is considered covered (or protected) whenever the total number of resources allocated to a schedule that covers the target equals or exceeds one (1).

Associated with each joint schedule $j \in J$ is a vector $\mathbf{P}_j = \langle P_{jt} \rangle \in \{0,1\}^n$, where P_{jt} indicates whether target t is covered in joint schedule j, i.e., $P_{jt} = \mathbb{I}(t \in j)$. The defender's mixed strategy $\mathbf{x}$ specifies the probabilities of playing each $j \in J$, where $x_j \geq 0, \sum_{j \in J} x_j = 1$. Let $\mathbf{c} = \langle c_t \rangle_{t=1}^n$ be the vector of coverage probabilities corresponding to $\mathbf{x}$, where $c_t = \sum_{j \in J} P_{jt} x_j$, is the marginal probability of covering t and we can write $\mathbf{c} = \mathbf{P}^\top \mathbf{x}$.

The payoffs of players are decided by the target chosen by the attacker and whether the target is protected by the defender. The defender's payoff for an uncovered attack on target t is denoted by $U_d^u(t)$ and for a covered attack $U_d^c(t)$. Similarly, $U_a^u(t)$ and $U_a^c(t)$ are the attacker's payoffs for the uncovered and covered cases, respectively. A widely adopted assumption in security games is that $U_d^c(t) > U_d^u(t)$ and $U_a^u(t) > U_a^c(t)$. In other words, covering an attack is beneficial for the defender, while hurts the attacker. Given a strategy profile $\langle \mathbf{x}, \mathbf{a} \rangle$, $\mathbf{c} = \mathbf{P}^\top \mathbf{x}$, the expected utilities for both players are denoted as follows:

$$U_d(\mathbf{c}, \mathbf{a}) = \sum_{t \in T} a_t U_d(\mathbf{c}, t), \text{where } U_d(\mathbf{c}, t) = c_t U_d^c(t) + (1 - c_t) U_d^u(t)$$

$$U_a(\mathbf{c}, \mathbf{a}) = \sum_{t \in T} a_t U_a(\mathbf{c}, t), \text{where } U_a(\mathbf{c}, t) = c_t U_a^c(t) + (1 - c_t) U_a^u(t)$$

We adopt a Stackelberg model in which the defender acts first and the attacker chooses a strategy after observing the defender's mixed strategy. Stackelberg games are common in security domains where attackers can surveil the defender strategy. The standard solution concept is SSE [13, 24], in which the leader selects an optimal mixed strategy based on the assumption that the follower will choose an optimal response, breaking ties in favor of the leader. There always exists an optimal pure-strategy response for the attacker, so we restrict our attention to this set in this paper.

3 REFINEMENT OF SSE IN SECURITY GAMES

A well-known property of SSE is that all SSEs give the same expected payoff for the leader (defender) [2, 13]. The refinement of SSEs in security games is first discussed in [1]. They indicate that multiple equilibria exist frequently (especially when there are resources, scheduling constraints) and in many of these solutions, a portion of the resources are not efficiently used since they can be abandoned without affecting the expected utility. We follow the same dominance criteria in [1]. The defender assumes there is an infinitesimal probability that the attacker will deviate from his first choice to his second or other preferable targets due to some unexpected events. But, even when the attacker is forced to deviate, he still behaves intelligently by choosing the next-best alternative rather than acting randomly. Therefore, the defender will still need to efficiently arrange the remaining resources to achieve her highest defender utilities, sequentially, on the secondary targets.

Based on this model, our equilibrium concept can be written as following: Given an SSE $\langle \mathbf{x}, \mathbf{a} \rangle$ and its coverage vector $\mathbf{c}$, an ordering over targets is defined such that target $t(1)$ is the target that will be attacked by the unconstrained attacker, and $t(i)$ is the target that will be attacked by the constrained attacker who cannot attack targets $t(1), \ldots, t(i-1)$. Utility vector $\mathbf{v} = \langle v_i \rangle_{i=1}^n$ represents the defender's utilities where v_i is the defender's utility if target $t(i)$ is attacked, i.e., $v_i = c_{t(i)} U_d^c(t(i)) + (1 - c_{t(i)}) U_d^u(t(i))$. They define a dominance relation between SSEs based on the utility vectors (if there is no ambiguity, we will consistently use coverage vector $\mathbf{c}$ to refer the defender's strategy $\mathbf{x}$).

Definition 3.1. Given two SSEs $\langle \mathbf{c}, \mathbf{a} \rangle, \langle \mathbf{c}', \mathbf{a}' \rangle$ and their utility vectors $\mathbf{v}$ and $\mathbf{v}'$. We say that SSE $\langle \mathbf{c}, \mathbf{a} \rangle$ **dominates** SSE $\langle \mathbf{c}', \mathbf{a}' \rangle$ if there exists i such that *i)* $v_i > v_i'$ and *ii)* $v_j = v_j'$ for all j such that $1 \leq j < i$.

There is an iterative algorithm [1] which can find the non-dominated SSE in the security games without scheduling constraints. In those cases, the multiple SSEs only exist when the best response target of the attacker is fully covered. In the security games with scheduling constraints, multiple SSEs are more common which motivates further needs for refinement. Unfortunately, in the presence of scheduling constraints, the method in [1] may return a dominated SSE, as illustrated by the following example.

Example 3.2 (Dominated SSEs in zero-sum SPARS games). Consider a zero-sum game with one resource $R = \{r_1\}$, three targets $T = \{t_1, t_2, t_3\}$, three schedules $S_1 = \{s_1, s_2, s_3\}$:

$$s_1 = \{t_1, t_3\},\ s_2 = \{t_2\},\ s_3 = \{t_3\}$$

and with the following payoffs: $U_d^c(t) = U_d^c(t) = 0\ \forall\ t \in T$

$$U_d^u(t_1) = -3, U_d^u(t_2) = -3, U_d^u(t_3) = -6, \quad U_a^u(t) = -U_d^u(t)\ \forall t \in T$$

There are infinite SSE solutions. One possible SSE could be $\mathbf{x}^1 = \langle \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \rangle$ with the corresponding coverage vector $\mathbf{c}^1 = \langle \frac{1}{3}, \frac{1}{3}, \frac{2}{3} \rangle$. The unsorted defender's utility vector is given by $\mathbf{d}^1 = \langle -2, -2, -2 \rangle$ for targets t_1, t_2, t_3. Accordingly, the sorted utility vector is given by $\mathbf{v}^1 = \langle -2, -2, -2 \rangle$. In this case, $\mathbf{v}^1$ and $\mathbf{d}^1$ are the same because the attacker feels indifferent between all of the targets. Applying the iterative algorithm from [1], given the arbitrary SSE $\mathbf{x}_1$, first we fix the coverage of one target among those with the highest attacker expected utility (in this case $\{t_1, t_2, t_3\}$). We assume the algorithm chooses target t_1 with $c_1 = \frac{1}{3}$ fixed and solves it iteratively, which returns the same strategy $\mathbf{x}^1$. However, the strategy $\mathbf{x}^1$ is dominated by strategy $\mathbf{x}^2 = \langle \frac{2}{3}, \frac{1}{3}, 0 \rangle$ with coverage $\mathbf{c}^2 = \langle \frac{2}{3}, \frac{1}{3}, \frac{2}{3} \rangle$ providing a better defender's utility vector $\mathbf{d}^2 = \langle -1, -2, -2 \rangle$ and $\mathbf{v}^2 = \langle -2, -2, -1 \rangle$ sorted by the attacker's preference. Both $\mathbf{x}^1, \mathbf{x}^2$ provide the highest defender's utility $d^* = -2$.

Example 3.2 shows that a non-dominated solution can perform significantly better than an arbitrary chosen SSE. In this case, if the attacker deviates from his best response (target t_2, t_3) to the third preferable target (target t_1), the defender's utility will be -2 and -1 for strategies $\mathbf{x}^1$ and $\mathbf{x}^2$, respectively, yielding a 50% difference between refined and arbitrary SSE. [1]

4 ZERO-SUM GAMES

In this paper, we start with zero-sum games where the attacker is completely opposite to the defender. We define the idea of minimum attack set, prove its uniqueness, and show the SSE with minimum attack set is better than all the other SSEs. We also show that the minimum attack set can be computed by a polynomial number of calls to an oracle that solves linear programs. Accordingly, we propose an algorithm which iteratively solves the minimum attack set of restricted instance and fixes the coverage on the minimum attack set. We prove that our algorithm requires at most $O(n^3)$ oracle calls and returns a non-dominated SSE.

4.1 Uniqueness of Minimum Attack Set of SSE

Definition 4.1. Given a feasible SSE coverage vector $\mathbf{c}$, the **Attack Set** $\Gamma(\mathbf{c}) := \arg\max_{t \in T} U_a(\mathbf{c}, t)$ is the best response of the attacker.

Definition 4.2. Let $\Psi := \{T' \subseteq T \mid \exists\ \text{SSE}\ \langle \mathbf{c}, \mathbf{a} \rangle : \Gamma(\mathbf{c}) = T'\}$ be the set of all possible attack sets of SSEs.

In zero-sum games, the less optimal choices of the attacker (attack set) imply the less targets that he can achieve his highest utility. Thus, for the defender, the SSE with a smaller attack set is always better than the SSE with a larger attack set.

[1] One intuitive heuristic algorithm of refinement, in the presence of constraints, is to eliminate those inefficient schedules [6]. E.g., in the context of Example 3.2, schedule $s_1 = \{t_1\}$ is dominated by $s_3 = \{t_1, t_3\}$. However, in the experiment part, we will show that the heuristic method provides only a little improvement in the zero-sum games and it does not work in the general-sum games.

Definition 4.3. A **Minimum Attack Set** is a set $M \in \Psi$ such that any proper subset V of M is not an element of Ψ, i.e., $V \notin \Psi$ for all $V \subset M$.

Example 4.4. Consider a zero-sum game with one resource $R = \{r_1\}, T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$, and four schedules $S_1 = \{s_1, s_2, s_3, s_4\}$:

$$s_1 = \{t_1, t_2, t_3\},\ s_2 = \{t_2, t_3, t_4\},\ s_3 = \{t_3, t_4, t_5\},\ s_4 = \{t_6\}$$

with the following payoffs: $U_d^c(t) = 0\ \forall\ t \in T$

$$U_d^u(t_1) = -4, U_d^u(t_2) = -4, U_d^u(t_3) = -12$$
$$U_d^u(t_4) = -4, U_d^u(t_5) = -2, U_d^u(t_6) = -4$$

There are infinite possible SSE solutions. For example, one possible SSE is $\mathbf{x}^1 = \langle \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4} \rangle$ with coverage $\mathbf{c}^1 = \langle \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4} \rangle$ which gives the defender's utility vector $\mathbf{d}^1 = \langle -3, -2, -3, -2, -1.5, -3 \rangle$ with $d^* = -3$. In this case, the best response of the attacker is $\Gamma(\mathbf{c}^1) = \{t_1, t_3, t_6\}$. The following mixed strategies also apply.

$$\mathbf{x}^2 = \langle \frac{1}{2}, \frac{1}{6}, \frac{1}{12}, \frac{1}{4} \rangle, \qquad \mathbf{c}^2 = \langle \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{1}{4}, \frac{1}{12}, \frac{1}{4} \rangle$$
$$\mathbf{d}^2 = \langle -2, -\frac{4}{3}, -3, -3, -\frac{11}{6}, -3 \rangle, \qquad \Gamma(\mathbf{c}^2) = \{t_3, t_4, t_6\}$$
$$\mathbf{x}^3 = \langle \frac{3}{8}, \frac{5}{24}, \frac{1}{6}, \frac{1}{4} \rangle, \qquad \mathbf{c}^3 = \langle \frac{3}{8}, \frac{7}{12}, \frac{3}{4}, \frac{3}{8}, \frac{1}{6}, \frac{1}{4} \rangle$$
$$\mathbf{d}^3 = \langle -2.5, -\frac{5}{3}, -3, -2.5, -\frac{5}{3}, -3 \rangle, \qquad \Gamma(\mathbf{c}^3) = \{t_3, t_6\}$$
$$\mathbf{x}^4 = \langle \frac{1}{4}, 0, \frac{1}{2}, \frac{1}{4} \rangle, \qquad \mathbf{c}^4 = \langle \frac{1}{4}, \frac{1}{4}, \frac{3}{4}, \frac{1}{2}, \frac{1}{2}, \frac{1}{4} \rangle$$
$$\mathbf{d}^4 = \langle -3, -3, -3, -2, -1, -3 \rangle, \qquad \Gamma(\mathbf{c}^4) = \{t_1, t_2, t_3, t_6\}$$

Clearly, all the above strategies are SSE solutions. But the strategy $\mathbf{x}^3$ dominates all the others since the defender's utility on the third-preferable target of the attacker is -2.5, which is higher than all the others' utility -3. Actually, $\mathbf{x}^3$ is the non-dominated strategy.

If we explore all of the possible SSEs in Example 4.4, we will find that the above attack sets are exactly all the possible attack sets:

$$\Psi = \{\{t_3, t_6\}, \{t_1, t_3, t_6\}, \{t_3, t_4, t_6\}, \{t_1, t_2, t_3, t_6\}\}$$

Therefore, the only minimum attack set is $\Gamma(\mathbf{c}^3) = \{t_3, t_6\}$.

Theorem 4.5 (Intersection Property in Zero-sum Games). *For any two attack sets $T^1, T^2 \in \Psi$, we have $T^1 \cap T^2 \neq \emptyset$ and $T^1 \cap T^2 \in \Psi$.*

Proof. Given $T^1 = \Gamma(\mathbf{c}), T^2 = \Gamma(\mathbf{c}')$, there are two cases:

(1) $\Gamma(\mathbf{c}) \cap \Gamma(\mathbf{c}') \neq \emptyset$. Consider another strategy $\mathbf{c}^* = \alpha\mathbf{c} + (1-\alpha)\mathbf{c}'$ with $\alpha \in (0, 1)$. Since $\mathbf{c}^* = \alpha\mathbf{c} + (1-\alpha)\mathbf{c}' = \alpha\mathbf{P}^\top\mathbf{x} + (1-\alpha)\mathbf{P}^\top\mathbf{x}' = \mathbf{P}^\top(\alpha\mathbf{x} + (1-\alpha)\mathbf{x}')$, $\mathbf{c}^*$ is a feasible coverage vector of strategy $\mathbf{x}^*$. It is easy to verify that $\Gamma(\mathbf{c}^*) = \Gamma(\mathbf{c}) \cap \Gamma(\mathbf{c}')$ as follows:

$$U_a(\mathbf{c}, t) \begin{cases} = v & \text{if } t \in \Gamma(\mathbf{c}) \\ < v & \text{otherwise} \end{cases} \qquad U_a(\mathbf{c}', t) \begin{cases} = v & \text{if } t \in \Gamma(\mathbf{c}') \\ < v & \text{otherwise} \end{cases}$$

$$U_a(\mathbf{c}^*, t) = \alpha U_a(\mathbf{c}, t) + (1-\alpha) U_a(\mathbf{c}', t) \begin{cases} = v & \text{if } t \in \Gamma(\mathbf{c}) \cap \Gamma(\mathbf{c}') \\ < v & \text{otherwise} \end{cases}$$

where v is the expected attacker's utility. Thus, we obtain an SSE strategy $\mathbf{c}^*$ with a smaller attack set $\Gamma(\mathbf{c}) \cap \Gamma(\mathbf{c}')$.

(2) $\Gamma(\mathbf{c}) \cap \Gamma(\mathbf{c}') = \emptyset$. Similarly, we consider the feasible strategy $\mathbf{c}^* = \alpha\mathbf{c} + (1-\alpha)\mathbf{c}'$ with $\alpha \in (0, 1)$. It is easy to verify that $U_a(\mathbf{c}^*, t) < v$ for any $t \in T$. In other words, $U_d(\mathbf{c}^*, t) > -v$ where $-v$ is the

highest expected utility of defender in SSE. This contradicts the optimality of SSE. That means this case will never happen. □

Consider the SSE strategy $\mathbf{x}^3$ in Example 4.4. It can be written as the combination of SSE strategies $\mathbf{x}^1, \mathbf{x}^2$ by $\mathbf{x}^3 = \frac{1}{2} \cdot \mathbf{x}^1 + \frac{1}{2} \cdot \mathbf{x}^2$. As Theorem 4.5 states, the attack set $\Gamma(\mathbf{c}^3) = \Gamma(\mathbf{c}^1) \cap \Gamma(\mathbf{c}^2)$.

Theorem 4.6. *The minimum attack set M exists and is unique. Moreover, for each $T' \in \Psi$, $M \subseteq T'$.*

Proof. (1) Existence: Clearly, $M = \bigcap_{T' \in \Psi} T' \neq \emptyset$ is a minimum attack set. (2) Uniqueness: If there are two different minimum attack sets, then by Theorem 4.5, their intersection will be non-empty and is a smaller attack set, which is a contradiction. □

In Example 4.4, the minimum attack set is exactly $\Gamma(\mathbf{c}^3) = \{t_3, t_6\}$ which is the attack set of the non-dominated solution $\mathbf{x}^3$.

4.2 An Iterative Algorithm

In zero-sum games, an SSE with a smaller attack set is better (for the defender) than an SSE with a larger attack set. This motivates us to find the minimum attack set. Moreover, the minimum attack set is included in every attack set, which implies that we can fix the common coverage on the minimum attack set and solve the remaining subproblem. For this aim, we define the *restricted* SPARS.

Definition 4.7. Given a SPARS instance g, we denote by $g^{\mathbf{c},T'}$ the restricted game with respect to coverage vector $\mathbf{c}$ and $T' \subset T$. The **restricted SPARS** instance $g^{\mathbf{c},T'}$ is the same as SPARS instance g except the following rules:

(R1) The attacker is forbidden to attack targets in T'.
(R2) The defender's coverage on $t \in T'$ is fixed to be c_t.
(R3) The defender must cover targets $t \in T \backslash T'$ enough such that the attacker utility on these targets is at most $\min_{t' \in T'} U_a(\mathbf{c}, t')$.

The SSE in a restricted game follows the same definition as in the original SPARS. Rule (R1) guarantees that the attacker will only focus on targets $T \backslash T'$. Rule (R2) guarantees that solving the restricted SPARS will not alter the existing coverage on T' which is already known. Rule (R3) requires the defender to cover targets $t \in T \backslash T'$ enough such that the targets in $T \backslash T'$ are not more preferable for the attacker than those in T'. In addition, we define the **restricted attack set** by $\Gamma(\mathbf{c}', T') = \arg\max_{t \in T \backslash T'} U_a(\mathbf{c}', t)$, where c' is a feasible solution to $g^{\mathbf{c},T'}$. Note that the restricted attack set is the attack set for the restricted instance $g^{\mathbf{c},T'}$, thus they share the same properties of attack sets. Accordingly, we can define the **minimum restricted attack set** for the restricted instance.

Algorithm 1: Iterative Algorithm for Zero-sum Games

```
Parameter: SPARS instance g, T' ← ∅, c ← 0
while |T'| < |T| do
    c ← a restricted SSE strategy in the instance g^{c,T'}
    M ← the minimum restricted attack set of instance g^{c,T'}
    T' ← T' ∪ M
Return: non-dominated SSE strategy c
```

Algorithm 1 depicts the procedure of equilibrium refinement in zero-sum games. We compute an arbitrary SSE strategy, fix the coverage on the minimum restricted attack set (we will discuss how to find the minimum attack set in Section 4.3), and iteratively solve the remaining restricted subproblem. The following theorems guarantee the correctness of Algorithm 1.

Proposition 4.8. *Given a restricted SPARS instance $g^{\mathbf{c},T'}$, its minimum restricted attack set M, and an SSE $\mathbf{c}^*$ of $g^{\mathbf{c},T'}$, we have the following statement: the strategy $\mathbf{c}'$ is a feasible defender coverage of $g^{\mathbf{c}^*,T' \cup M}$ (satisfies Rules (R2), (R3)) if and only if $\mathbf{c}'$ is an SSE of $g^{\mathbf{c}^*,T'}$, which provides a mapping between two different restricted instances.*

Proof. ($\Leftarrow$) Since both $\mathbf{c}'$ and $\mathbf{c}^*$ are SSE strategies and M is the minimum restricted attack set of $g^{\mathbf{c},T'}$, both $\mathbf{c}'$ and $\mathbf{c}^*$ share the same value on $T' \cup M$, which satisfies the Rule (R2) of $g^{\mathbf{c}^*,T' \cup M}$.

Since $\mathbf{c}'$ is an SSE of $g^{\mathbf{c},T'}$, the attacker's utility on $t \in T'$ with SSE strategy $\mathbf{c}'$ must be greater than all the others $t \notin T'$. By the definition of minimum restricted attack set M, the best response of the attacker, it implies that target $t \in M$ must have the highest attacker's utility among $T \backslash T'$. Therefore, the attacker utility on $t \in T' \cup M$ is no less than the others' utilities, which satisfies Rule (R3) of $g^{\mathbf{c}^*,T' \cup M}$.

($\Rightarrow$) Assume that $\mathbf{c}'$ is a solution of $g^{\mathbf{c}^*,T' \cup M}$. By the definition of restricted instance $g^{\mathbf{c}^*,T' \cup M}$, the coverage $\mathbf{c}'$ on targets in $T' \cup M$ has been fixed to be the same as $\mathbf{c}^*$, and Rule (R3) forces all the other targets outside of $T' \cup M$ to have a smaller attacker's utility. It implies that the strategy $\mathbf{c}'$ achieves the highest attacker's utility on minimum attack set M in the restricted instance $g^{\mathbf{c},T'}$, thus an SSE in the restricted instance $g^{\mathbf{c},T'}$. □

Theorem 4.9. *The output of Algorithm1 is a non-dominated SSE.*

Proof. Denote the sequences of minimum restricted attack sets and updated coverage in Algorithm 1 as $M_1, ..., M_k$ and $\mathbf{c}^1, ..., \mathbf{c}^k$, respectively (W.L.O.G let $M_0 = \emptyset, \mathbf{c}^0 = \mathbf{0}$). According to Algorithm 1, $\mathbf{c}^i$ is an SSE, M_i is the minimum attack set of $g^{\mathbf{c}^{i-1}, M_1 \cup \ldots \cup M_{i-1}}$.

By Proposition 4.8, $\forall i \in \{1, 2, ..., k\}$ we have: given a restricted instance $g^{\mathbf{c}^{k-i}, M_1 \cup M_2 \cup \ldots \cup M_{k-i}}$, its minimum restricted attack set M_{k-i+1} and its SSE $\mathbf{c}^{k-i+1}$, the strategy $\mathbf{c}'$ is a feasible coverage of the instance $g^{\mathbf{c}^{k-i+1}, M_1 \cup M_2 \cup \ldots \cup M_{k-i} \cup M_{k-i+1}}$ if and only if $\mathbf{c}'$ is an SSE of the instance $g^{\mathbf{c}^{k-i+1}, M_1 \cup M_2 \cup \ldots \cup M_{k-i}}$.

According to the above argument, $\forall i \in \{1, 2, ..., k\}$ we have: $\mathbf{c}^k$ is the non-dominated solution of $g^{\mathbf{c}^{k-i+1}, M_1 \cup M_2 \cup \ldots \cup M_{k-i+1}} \Leftrightarrow$ $\mathbf{c}^k$ is the non-dominated SSE of $g^{\mathbf{c}^{k-i+1}, M_1 \cup M_2 \cup \ldots \cup M_{k-i}} \Leftrightarrow \mathbf{c}^k$ is the non-dominated SSE of $g^{\mathbf{c}^{k-i}, M_1 \cup M_2 \cup \ldots \cup M_{k-i}}$ (since $\mathbf{c}^{k-i}$ and $\mathbf{c}^{k-i+1}$ share the same coverage on $M_1 \cup M_2 \cup \ldots \cup M_{k-i}$) $\Leftrightarrow$ $\mathbf{c}^k$ is the non-dominated solution of $g^{\mathbf{c}^{k-i}, M_1 \cup M_2 \cup \ldots \cup M_{k-i}}$ (non-dominated solution must be an SSE). When $i = 1$, $\mathbf{c}^k$ is the only solution (thus non-dominated) solution of $g^{\mathbf{c}^k, M_1 \cup M_2 \cup \ldots \cup M_k}$ (since $M_1 \cup M_2 \cup \ldots \cup M_k = T$). By induction, $\mathbf{c}^k$ is the non-dominated solution of $g^{\mathbf{c}^{k-i}, M_1 \cup M_2 \cup \ldots \cup M_{k-i}}$. By letting $i = k$, the statement is exactly our conclusion: $\mathbf{c}^k$ is the non-dominated solution of g. □

4.3 Computing the Minimum Attack Set

In the previous section, we showed that a non-dominated SSE strategy can be obtained by iteratively computing SSE strategies of restricted SPARS instances and their corresponding minimum

restricted attack sets. We now propose a method for finding the unique minimum attack set.

$$\max_{\mathbf{c},\mathbf{x}} \quad U_d(\mathbf{c}, t) \tag{1a}$$

$$\text{s.t.} \quad U_a(\mathbf{c}, t) \geq U_a(\mathbf{c}, t') \quad \forall t' \neq t \tag{1b}$$

$$\mathbf{P}^\top \mathbf{x} = \mathbf{c} \tag{1c}$$

$$\sum_{j \in J} x_j = 1. \tag{1d}$$

First, multiple LPs method [4] is commonly used to compute an SSE to security games. Each LP (1) corresponds to one target t and maximizes the defender's expected utility on this target under the restriction that t is in the best response for the attacker.

Definition 4.10. Given target t, let N_t be the **smallest tight constraint set** with $N_t := \{t' \in T \mid \forall \text{SSE strategies } \mathbf{c} \text{ with } t \in \Gamma(\mathbf{c}), U_a(\mathbf{c}, t) = U_a(\mathbf{c}, t'), U_d(\mathbf{c}, t) = U_d(\mathbf{c}, t')\}$.

Given target t and its LP (1), we are interested in which constraints are necessary and always active for all optimal solutions (SSEs), which is the smallest tight constraint set N_t. Our main idea is to slightly alter the constraint of target t' in LP (1) to

$$U_a(\mathbf{c}, t) \geq U_a(\mathbf{c}, t') + \epsilon$$

where ϵ is a small positive number (e.g., constant times of numerical error). If the modified version of the linear program still provides the same maximum objective value (up to numerical error), then the constraint with respect to t' is not active, which means $t' \notin N_t$. If it provides a smaller objective value or the linear program is infeasible, that means the constraint with respect to t' is always active, which implies $t' \in N_t$.

The procedure of Algorithm 2 is to find out the smallest tight constraint set N_t under the restriction that t is the best response of attacker. Every N_t can be solved by at most n modified linear programs. We will show that the intersection of all the smallest tight constraint sets is exactly the minimum attack set.

Algorithm 2: Algorithm for Finding Minimum Attack Set

```
Parameter: SPARS instance g^{c,T'}
solve an SSE c* using the multiple linear program method and
  record the primal, dual solution of each LP
for t ∈ Γ(c*) do
    given the dual solution d' and primal solution c' of LP (1)
    N_t ← A_t ← {t' | d'_{t'} ≠ 0} ∪ {t}, B_t ← Γ(c')\A_t
    for t' ∈ B_t do
        solve modified LP (1) with one more constraint
          U_a(c, t) ≥ U_a(c, t') + ε
        if the objective value changes then
            N_t ← N_t ∪ {t'}
Return: minimum restricted attack set ⋂_{t∈Γ(c*)} N_t, coverage c*
```

Proposition 4.11. *Given the dual solution $\mathbf{d}'$ of LP (1), the set $\{t' | \mathbf{d}'_{t'} \neq 0\}$ is contained in the smallest tight constraint set N_t.*

Proposition 4.12. *Given a primal solution $\mathbf{c}'$ of LP (1), every target $t \notin \Gamma(\mathbf{c}')$ is not contained in the smallest tight constraint set.*

Proposition 4.11 and 4.12 help eliminate unnecessary enumerations in Algorithm 2. In the average case, there are only a constant number of targets in B_t (in Algorithm 2) needed to be enumerated But in the worst case, we still need to run through at most n targets. The following theorems guarantee correctness of Algorithm 2.

Proposition 4.13. *Given target t, $N_t = \bigcap_{T' \in \Psi: t \in T'} T'$. Moreover, given arbitrary SSE coverage $\mathbf{c}'$, $\bigcap_{t \in \Gamma(\mathbf{c}')} N_t = \bigcap_{T' \in \Psi} T'$, which is the minimum attack set M.*

Proof. First, for each SSE solution $\mathbf{c}$, the targets in $\Gamma(\mathbf{c})\backslash\{t\}$ are exactly the targets which make the constraints (1b) tight. Thus, the smallest tight constraints are the same as the intersection of attack sets containing t as the best response for the attacker, which implies $N_t = \bigcap_{T' \in \Psi: t \in T'} \Gamma(\mathbf{c})$. Second, since $\Gamma(\mathbf{c}')$ contains at least one target t in the minimum attack set, the minimum attack set M must appear in one of the $T' \in \Psi, t \in T'$, which implies $\bigcap_{t \in \Gamma(\mathbf{c}')} N_t = \bigcap_{t \in \Gamma(\mathbf{c}')} \bigcap_{T' \in \Psi: t \in T'} \Gamma(\mathbf{c}) = M = \bigcap_{T' \in \Psi} \Gamma(\mathbf{c})$. □

Theorem 4.14. *Algorithm 2 correctly returns the minimum restricted attack set of $g^{\mathbf{c},T'}$.*

We can employ Algorithm 2 in Algorithm 1 to find the minimum attack set. This provides our iterative algorithm for finding a non-dominated SSE strategy in zero-sum games. In order to find every smallest constraint set N_t, we need to enumerate all target pairs (t, t'). Therefore, the number of oracle calls of each iteration is $O(n^2)$, where oracles are used to solve variants of LP (1). There are at most n iterations, thus the overall runtime is $O(n^3)$ oracle calls.

Theorem 4.15. *Algorithm 1 correctly solves the non-dominated SSE in $O(n^3)$ oracle calls.*

5 GENERAL-SUM GAMES

In this section, we discuss the refinement of SSEs in general-sum games. The method is similar to the zero-sum case, but one of the crucial difficulties is that there is no longer a direct relation between the defender and attacker utilities. Several useful properties of zero-sum games do not hold either. For example, in the general setting the intersection of two attack sets may not be an attack set, leading to non-uniqueness of the minimum attack set. This implies a significant growth of time complexity.

5.1 Non-Uniqueness of Minimum Attack Set

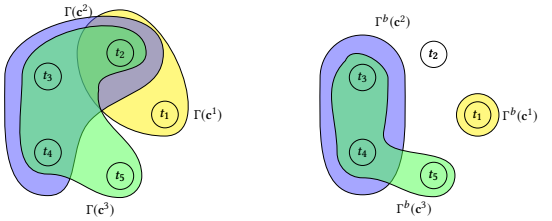
In Section 4, Theorem 4.5 tells us that in zero-sum games the intersection of two attack sets is still an attack set. The following example shows that it is not necessarily true in general-sum games.

Example 5.1. Consider a game with one resource $R = \{r_1\}$, five targets $T = \{t_1, t_2, t_3, t_4, t_5\}$, and three schedules $S_1 = \{s_1, s_2, s_3\}$:

$$s_1 = \{t_1, t_2\}, s_2 = \{t_3, t_4\}, s_3 = \{t_3, t_4, t_5\}$$

We have the following payoffs:

$$t_1 : U_d^c(t_1) = 10, U_d^u(t_1) = -10, U_a^u(t_1) = 10, U_a^c(t_1) = -10$$
$$t_2 : U_d^c(t_2) = 0,\ U_d^u(t_2) = -5,\ U_a^u(t_2) = 5,\ U_a^c(t_2) = -5$$
$$t_3 : U_d^c(t_3) = 6,\ U_d^u(t_3) = -4,\ U_a^u(t_3) = 3,\ U_a^c(t_3) = -7$$
$$t_4 : U_d^c(t_4) = 3,\ U_d^u(t_4) = -2,\ U_a^u(t_4) = 4,\ U_a^c(t_4) = -8.5$$
$$t_5 : U_d^c(t_5) = 4,\ U_d^u(t_5) = -1,\ U_a^u(t_5) = 0,\ U_a^c(t_5) = -5$$



(a) the set of attack sets Ψ **(b) the set of best attack sets Ψ^b**

Figure 1: The attack sets in Example 5.1

Since the schedule s_2 is completely contained in the schedule s_3, the intuition tells us choosing s_3 will always be better than choosing s_2. However, this is wrong in this case. In order to show that, we list some SSE solutions with unsorted attacker utility $\mathbf{f}$, unsorted defender utility $\mathbf{d}$, and defender utility $\mathbf{v}$ sorted in attacking order:

$$\mathbf{x}^1 = \langle 0.5, 0.1, 0.4\rangle, \quad \mathbf{f}^1 = \langle 0, 0, -2, -2.25, -2\rangle$$
$$\mathbf{d}^1 = \langle 0, -2.5, 1, 0.5, 1\rangle, \quad \mathbf{v}^1 = \langle 0, -2.5, 1, 1, 0.5\rangle$$
$$\mathbf{x}^2 = \langle 0.6, 0, 0.4\rangle, \quad \mathbf{f}^2 = \langle -2, -1, -1, -1, -2\rangle$$
$$\mathbf{d}^2 = \langle 2, -2, 0, 0, 1\rangle, \quad \mathbf{v}^2 = \langle 0, 0, -2, 2, 1\rangle$$
$$\mathbf{x}^3 = \langle 0.6, 0.2, 0.2\rangle, \quad \mathbf{f}^3 = \langle -2, -1, -1, -1, -1\rangle$$
$$\mathbf{d}^3 = \langle 2, -2, 0, 0, 0\rangle, \quad \mathbf{v}^3 = \langle 0, 0, 0, -2, 2\rangle$$

$$\Gamma(\mathbf{c}^1) = \{t_1, t_2\}, \Gamma(\mathbf{c}^2) = \{t_2, t_3, t_4\}, \Gamma(\mathbf{c}^3) = \{t_2, t_3, t_4, t_5\}$$

It can be verified that these are all the possible attack sets. We have that $\mathbf{v}^3$ dominates $\mathbf{v}^2$ and $\mathbf{v}^1$, which implies partially using inefficient schedule s_2 will result in better performance. Figure 1(a) illustrates all the attack sets in Example 5.1, which shows that the minimum attack set is not unique in general-sum games.

5.2 Best Attack Set

We introduce the notion of **best attack set**. Similar to Section 4, we iteratively fix the coverage on the minimum best attack set: those targets the attacker will actually attack, up to breaking ties.

Definition 5.2. Given an SSE coverage vector $\mathbf{c}$, the **Best Attack Set** $\Gamma^b(\mathbf{c})$ is the set of targets in the best response of the attacker which also achieves the highest defender utility.

In Example 5.1, as shown in Figure 1(b), the best attack sets are respectively $\Gamma^b(\mathbf{c}^1) = \{t_1\}, \Gamma^b(\mathbf{c}^2) = \{t_3, t_4\}, \Gamma^b(\mathbf{c}^3) = \{t_3, t_4, t_5\}$.

Definition 5.3. Let $\Psi^b = \{T' \subseteq T \mid \exists \text{ SSE } \langle \mathbf{c}, \mathbf{a}\rangle : T' = \Gamma^b(\mathbf{c})\}$ be the set of all possible best attack sets of SSEs.

Theorem 5.4 (Intersection Property in General-sum Games). *For any two attack sets $\Gamma(\mathbf{c}), \Gamma(\mathbf{c}') \in \Psi$ (Definition 4.2), if $\Gamma^b(\mathbf{c}) \cap \Gamma^b(\mathbf{c}') \neq \emptyset$, we have $\Gamma(\mathbf{c}) \cap \Gamma(\mathbf{c}') \in \Psi, \Gamma^b(\mathbf{c}) \cap \Gamma^b(\mathbf{c}') \in \Psi^b$.*

Proof. Given two sets $\Gamma(\mathbf{c}), \Gamma(\mathbf{c}') \in \Psi$, their corresponding SSEs $\langle \mathbf{c}, \mathbf{a}\rangle$ and $\langle \mathbf{c}', \mathbf{a}'\rangle$ with $\Gamma^b(c) \cap \Gamma^b(c') \neq \emptyset$, we follow a similar proof idea as in Theorem 4.5. Consider another strategy $\mathbf{c}^* = \alpha\mathbf{c} + (1-\alpha)\mathbf{c}'$ with $\alpha \in (0, 1)$. $\mathbf{c}^*$ is a feasible coverage vector with strategy $\mathbf{x}^* = \alpha\mathbf{x} + (1-\alpha)\mathbf{x}'$. Moreover, they share some common targets in their best attack sets and thus the same highest attacker's utilities v_a and the highest defender's utility v_d. It is easy to verify that $\Gamma(\mathbf{c}^*) = \Gamma(\mathbf{c}) \cap \Gamma(\mathbf{c}'), \Gamma^b(\mathbf{c}^*) = \Gamma^b(\mathbf{c}) \cap \Gamma^b(\mathbf{c}')$ as follows:

$$U_a(\mathbf{c}, t)\begin{cases} = v_a & \text{if } t \in \Gamma(\mathbf{c}) \\ < v_a & \text{otherwise}\end{cases} \quad U_a(\mathbf{c}', t)\begin{cases} = v_a & \text{if } t \in \Gamma(\mathbf{c}') \\ < v_a & \text{otherwise}\end{cases}$$

$$U_a(\mathbf{c}^*, t) = \alpha U_a(\mathbf{c}, t) + (1-\alpha)U_a(\mathbf{c}', t)\begin{cases} = v_a & \text{if } t \in \Gamma(\mathbf{c}) \cap \Gamma(\mathbf{c}') \\ < v_a & \text{otherwise}\end{cases} \tag{2}$$

$$U_d(\mathbf{c}, t)\begin{cases} = v_d & \text{if } t \in \Gamma^b(\mathbf{c}) \\ < v_d & \text{if } t \in \Gamma(\mathbf{c})\backslash\Gamma^b(\mathbf{c})\end{cases} \quad U_d(\mathbf{c}', t)\begin{cases} = v_d & \text{if } t \in \Gamma^b(\mathbf{c}') \\ < v_d & \text{if } t \in \Gamma(\mathbf{c}')\backslash\Gamma^b(\mathbf{c}')\end{cases}$$

$$U_d(\mathbf{c}^*, t) = \alpha U_d(\mathbf{c}, t) + (1-\alpha)U_d(\mathbf{c}', t)$$

$$\Rightarrow U_d(\mathbf{c}^*, t)\begin{cases} = v_d & \text{if } t \in \Gamma^b(\mathbf{c}) \cap \Gamma^b(\mathbf{c}') \\ < v_d & \text{if } t \in (\Gamma(\mathbf{c}) \cap \Gamma(\mathbf{c}'))\backslash(\Gamma^b(\mathbf{c}) \cap \Gamma^b(\mathbf{c}'))\end{cases} \tag{3}$$

Equation (2) guarantees the attack set of strategy $\mathbf{c}^*$ is $\Gamma(\mathbf{c}) \cap \Gamma(\mathbf{c}')$. Equation (3) guarantees that among the attack set $\Gamma(\mathbf{c}) \cap \Gamma(\mathbf{c}')$, strategy $\mathbf{c}$ achieves the highest defender's utility on target t if and only if the target $t \in \Gamma^b(\mathbf{c}) \cap \Gamma^b(\mathbf{c}')$ which is non-empty. Thus, the best attack set of strategy $\mathbf{c}^*$ is $\Gamma^b(\mathbf{c}^*) = \Gamma^b(\mathbf{c}) \cap \Gamma^b(\mathbf{c}')$. □

Theorem 5.4 implies that the intersection of attack sets is still an attack set if the intersection of their best attack sets is non-empty. But if the intersection of their best attack sets is empty, the combining strategy $\mathbf{c}^*$ is no longer an SSE, and thus the intersection of attack sets may not be an attack set. Based on Theorem 5.4, we can define the minimum best attack set:

Definition 5.5. A **Minimum Best Attack Set** is a best attack set $M \in \Psi^b$ such that any proper subset $V \subset M$ is not an element of Ψ^b, that is $V \notin \Psi^b$ for all $V \subset M \cap V\}$.

Proposition 5.6. *Given any SSE strategy $\mathbf{c}$, its attack set $\Gamma(\mathbf{c})$ must contain one of the minimum best attack sets.*

5.3 A Recursive Algorithm

Based on the above theorems and paralleling Algorithm 1, Algorithm 3 iterates through all minimum best attack sets M and finds the non-dominated SSE in each restricted instances $g^{\mathbf{c}, T' \cup M}$. After enumerating all the possible solutions, it returns the best one. The following results guarantee correctness of Algorithm 3.

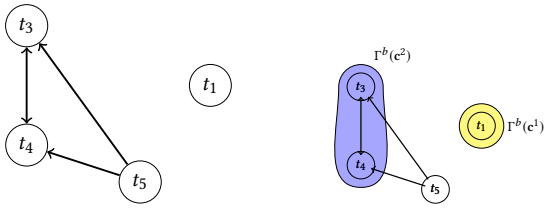
Algorithm 3: ***RefinedSSE***(g)

```
Function RefinedRestrictedSSE(g, c, T')
    Parameter: restricted SPARS instance g^{c,T'}, cList ← []
    for each minimum best attack set M of g^{c,T'} do
        c* ← an SSE of instance g^{c,T'∪M}
        c' ← RefinedRestrictedSSE(g, c*, T' ∪ M)
        add c' into cList
    return non-dominated coverage vector among cList
return RefinedRestrictedSSE(g, c = 0, T' = ∅)
```

Proposition 5.7. *Assume M is a minimum restricted best attack set of $g^{\mathbf{c}, T'}$ and $\mathbf{c}^*$ is an SSE strategy of $g^{\mathbf{c}, T'}$ containing M in the attack set. Then, strategy $\mathbf{c}'$ is an SSE of $g^{\mathbf{c}, T'}$ containing M if and only if $\mathbf{c}'$ is a solution of $g^{\mathbf{c}^*, T' \cup M}$.*

Theorem 5.8. *The output of Algorithm 3 is a non-dominated SSE.*

Proofs are similar to those of Proposition 4.8 and Theorem 4.9.



(a) the corresponding graph **(b) minimum best attack sets**

Figure 2: The minimum best attack sets in the Example 5.1

5.4 Computing Minimum Best Attack Sets

Similar to Section 4, we next propose an efficient method to find all the minimum best attack sets. Following the notations in Section 4.3, it can be shown that in general-sum games, solving the modified LPs (1) with respect to target t will yield the smallest tight constraint set N_t. The following proposition gives an alternative expression of N_t (the proof is similar to that of Theorem 4.13):

Proposition 5.9. $N_t = \bigcap_{T' \in \Psi^b : t \in T'} T'$

The set N_t provides the information between targets: if target t is included in the best attack set T', then all the targets in N_t must be included in the best attack set T' too. We can then focus on those targets which could be in the best attack set.

Definition 5.10. Let Q be the set of targets which achieve the best defender utility in some SSE strategies.

The set Q is equivalent to the set of targets t for which LP (1) provides the highest defender utility and that can be derived while solving the n linear programs. We construct a directed graph $G = (V, E)$ to represent the relations between these targets. Let $V = Q$ be the set of all targets which could achieve the highest defender's utility. Let $E = \bigcup_{t \in Q} \{(t, s) | s \in N_t, s \in Q, s \neq t\}$ where (t, s) is the directed edge from t to s.

Example 5.11 (Continued from Example 5.1). With the help of Figure 1(b), we can visualize the sets $\{N_t | t \in Q\}$ ($Q = \{t_1, t_3, t_4, t_5\}$):

$$N_{t_1} = \{t_1, t_2\}, N_{t_3} = \{t_3, t_4\}, N_{t_4} = \{t_3, t_4\}, N_{t_5} = \{t_3, t_4, t_5\}.$$

We can draw a corresponding graph (Figure 2(a)) according to these sets. Figure 2(b) depicts all of the minimum best attack sets. Notice that the definition of edges implies the inclusion relationship: $e = (t, s) \in E$ if and only if $t \in Q$, and any attack set including target t must also include target s.

Proposition 5.12. *Directed relations are* **transitive** *in graph $G = (V, E)$, i.e., if (t, u) and $(u, v) \in E$, then $(t, v) \in E$.*

The transitive rule has an intuitive meaning: if a best attack set is such that if t is included, so must u; and if u is included, so must v; then if it includes t, it must also include v.

Lemma 5.13. *M is a minimum best attack set if and only if M is a maximal clique without outgoing edge directed from M to any other target in $Q \backslash M$.*

Proof. ($\Rightarrow$) $\forall t \in M$, by Theorem 5.9, $N_t = \bigcap_{T' \in \Psi^b : t \in T'} T'$. Notice that the minimum best attack set M satisfies $M \in \Psi^b, t \in M$, thus $N_t \subseteq M$. Moreover, by Theorem 5.4, N_t is the intersection of best attack sets, which implies that N_t is a best attack set. But we have $N_t \subseteq M$ and M is a minimum best attack set. By the definition of the minimum best attack set, the only possibility is $N_t = M \; \forall \; t \in M$, which implies that M is a maximal clique without outgoing edges. ($\Leftarrow$) Suppose M is a maximal clique without any outgoing edge. Then $N_t = M \; \forall \; t \in M$. Since N_t is the intersection of best attack sets, $M = N_t$ is also a best attack set. Moreover, if a best attack set V includes any vertex $t \in M$, V must include $N_t = M$ (since $N_t = \bigcap_{T' \in \Psi^b : t \in T'} T'$, V satisfies $V \in \Psi^b : t \in V$). Next, we derive a contradiction. Suppose there is a proper subset $V \subset M$ which is also a best attack set. Then, there exists $t \in V \cap M$. By the above argument, we have $M = N_t \subseteq V$, which contradicts that $V \subset M$. We conclude that M is a minimum best attack set. □

Although the maximal clique problem is generically NP-hard, fortunately, the transitive law in Proposition 5.12 reduces the maximal clique problem to a variant of the tournament problem $G = (V, E)$ with time complexity $O(|V| + |E|) = O(n^2)$. In Algorithm 4, we leverage the transitive law to propose a random walk method that successfully discovers all the minimum best attack sets in $O(n^2)$.

Algorithm 4: Find All the Minimum Best Attack Sets

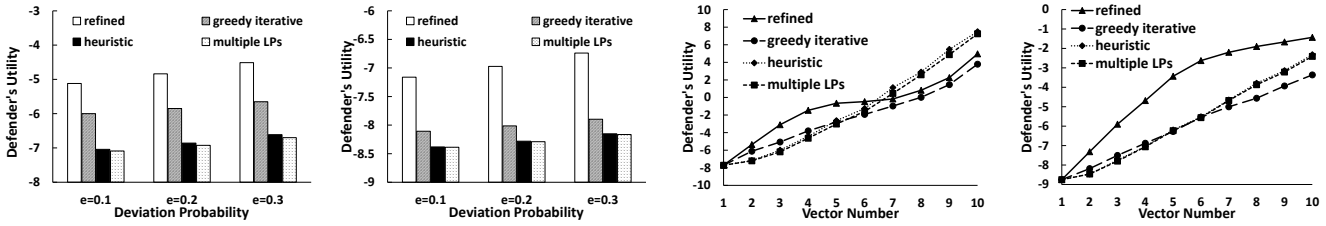
```
Transitive graph G = (V, E), Mlist ← [], V' ← V, E' ← E
while V' ≠ ∅ do
    Start random walk in G' = (V', E') and record all the
      nodes we walked through until we encounter a duplicate
      node or we cannot move anymore. Let v be the last node.
    N_v ← N(v) ∪ {v} where N(v) is the neighborhood of v
    if N_v is a maximal clique without outgoing edges then
        add N_v into Mlist
    V' ← nodes in V' that have not been passed by
    E' ← edges in E with both endpoints ∈ V'
return Mlist
```

Theorem 5.14. *Each subproblem in Algorithm 3 correctly returns the non-dominated solution in $O(n^3)$ oracle calls.*

Both the worst-case runtime of Algorithm 4 and the computation of all the sets $\{N_t | t \in Q\}$ are $O(n^2)$ oracle calls. Therefore, the worst-case runtime of solving each subproblem in the recursive algorithm is still $O(n^3)$ oracle calls, same as the zero-sum cases. The number of subproblems depends on the number of minimum best attack sets. In Example 5.1, there are two minimum best attack sets: $\{t_1\}$ and $\{t_3, t_4\}$, so we need to compute the non-dominated solutions for both cases and choose the best one. The overall runtime depends on the number of subproblems that need to be solved. Fortunately, while iteratively solving the subproblems, rule (R3) enables us to foresee the defender's utilities on the first few targets, thus prune out a large number of subproblems, which reduces the overall runtime significantly relative to the worst-case (reduce from exponential to polynomial many oracle calls in practice).

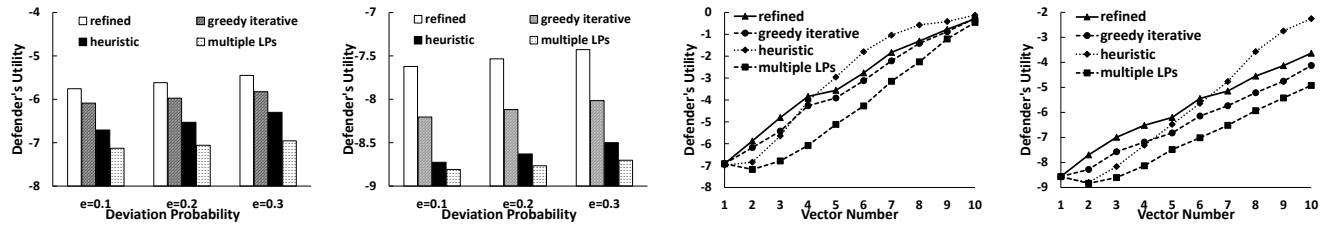
6 EXPERIMENTAL RESULTS

We run experiments to evaluate the solution quality and scalability of the refined SSE on SPARS. All LPs are solved by CPLEX (version 12.7.1) on a machine with an Intel core i5-7200U CPU



(a) Residual utility $n = 10$ (b) Residual utility $n = 20$ (c) Utility vectors $n = 10$ (d) Utility vectors $n = 20$

Figure 3: Defender utility in zero-sum games



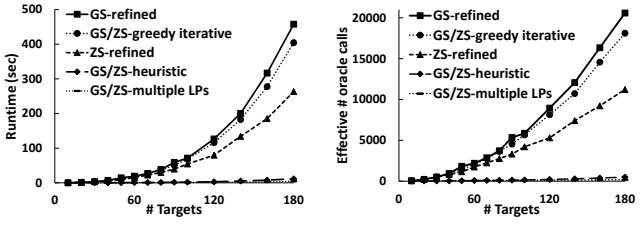
(a) Residual utility $n = 10$ (b) Residual utility $n = 20$ (c) Utility vectors $n = 10$ (d) Utility vectors $n = 20$

Figure 4: Defender utility in general-sum games

and 11.6GB memory. Our experiments use 100 sampled game instances with 2 defender resources, varying the number of targets, and randomly generated payoffs. In zero-sum cases, payoffs $U_a^u(t) = -U_d^u(t), U_d^c(t) = -U_a^c(t)$ are uniformly distributed in the set $\{0, 1, \ldots, 10\}$. In general-sum cases, we are motivated by ARMOR [16] and adopt the following payoff setting: $U_a^u(t) = -U_d^u(t)$ uniformly distributed in the set $\{0, 1, \ldots, 10\}$ (completely opposite on successful attack), $U_d^c(t) = 0$ (zero reward for successful protect), and $U_a^c(t)$ uniformly distributed in $\{0, 1, \ldots, \lfloor U_a^u(t)/2 \rfloor\}$. Each instance also encompasses $O(n)$ randomly generated scheduling constraints with each schedule covering 2 to 5 targets depending on the number of targets. We employ CPLEX as our oracle to obtain exact solutions to linear programs. We compare the solution quality of our refined SSE to the SSEs given by the **multiple LPs** method [4], **heuristic** method [6], and **greedy iterative** method [1][2].

Since the defender utilities on the first preferable target are identical for all SSEs, we display the *residual* expected utility for the remaining targets. Suppose the attacker deviates from his target to the secondary target with probability e. Further assume that the attacker does not attack the first preferable target, then the attacker will attack the second preferable target with probability $1 - e$, third preferable target with $e(1 - e)$ and so on. Given the utility vector $\mathbf{v}$ sorted by the attacking order, the residual value is expressible as $\sum_{2 \le i \le n} (1 - e) e^{i-2} v_i$.

Figures 3(a), 3(b), 4(a), 4(b) illustrate the residual expected utilities in zero and general-sum games with $n = 10$ and 20, respectively. Without spending additional resources, our refined solution outperforms the other SSE solutions, improving the defender utility by $10 - 40\%$. Figures 3(c), 3(d), 4(c), 4(d) depict the defender utilities in attacking order. The figures show that *(i)* the refined SSE and other SSEs provide the same defender utility on the first preferable target; *(ii)* While the heuristic and multiple LPs methods are a lot faster than ours (Figures 5(a), 5(b)), they perform significantly worse since they do not refine the solution; *(iii)* The refined SSE gives a much higher defender utility on the following few targets (second and third preferable) by sacrificing those less preferable targets, which are even more unlikely to be attacked than the first few targets.



(a) Runtime Comparison (b) Effective # oracle calls

Figure 5: Time complexity

Figure 5(a) (resp. 5(b)) compares the runtime (resp. number of oracle calls) of our algorithm relative to other algorithms in zero-sum (ZS) and general-sum (GS) cases. The results show *(i)* the runtime of both zero and general-sum cases is of the same order as the runtime of the greedy iterative algorithm, which requires $O(n^2)$ oracle calls. Thus, the empirical number of oracle calls is significantly lower than our worst-case estimate of $O(n^3)$. This is due to the fact that in random settings, the cardinality of Q (Definition 5.10) is small (usually under 4), resulting in a small number of enumerations of $N_t, t \in Q$; *(ii)* Our algorithm for zero-sum games is almost two times faster than the greedy iterative algorithm because fixing the minimum attack set can significantly reduce the number of iterations, which speeds up our algorithm and also boosts solution quality; *(iii)* Figure 5(a) also shows that the runtime of our optimal algorithm is close to the runtime of the greedy iterative one. Contrary to the greedy iterative approach, our algorithm guarantees optimality and provides a significant improvement in defender utility and robustness, see Figures 3(c), 3(d), 4(c), and 4(d) at low computational cost, which provides a more robust solution with further spending only little more runtime.

Acknowledgment: This research was supported by MURI grant W911NF-17-1-0370.

[2]The **heuristic** method starts from an arbitrary SSE and goes through all of the pure strategies. If there is a strictly better pure strategy than the pure strategy in the current mixed strategy, then move the weight to the better one. The **greedy iterative** method adopts the idea of the iterative algorithm [1] but without finding minimum attack sets. It iteratively fixes the coverage of an arbitrary target in the attack set (best attack set).

REFERENCES

[1] Bo An, Milind Tambe, Fernando Ordóñez, Eric Anyung Shieh, and Christopher Kiekintveld. 2011. Refinement of Strong Stackelberg Equilibria in Security Games. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011.*

[2] Michele Breton, A Alj, and Alain Haurie. 1988. Sequential Stackelberg Equilibria in Two-person Games. *Journal of Optimization Theory and Applications* 59, 1 (1988), 71–97.

[3] Victor Bucarey, Carlos Casorrán, Óscar Figueroa, Karla Rosas, Hugo Navarrete, and Fernando Ordóñez. 2017. Building Real Stackelberg Security Games for Border Patrols. In *International Conference on Decision and Game Theory for Security*. Springer, 193–212.

[4] Vincent Conitzer and Tuomas Sandholm. 2006. Computing the Optimal Strategy to Commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*. ACM, 82–90.

[5] Karel Durkota, Viliam Lisỳ, Christopher Kiekintveld, Karel Horák, Branislav Bošanskỳ, and Tomáš Pevnỳ. 2017. Optimal Strategies for Detecting Data Exfiltration by Internal and External Attackers. In *International Conference on Decision and Game Theory for Security*. Springer, 171–192.

[6] Fei Fang, Albert Xin Jiang, and Milind Tambe. 2013. Optimal Patrol Strategy for Protecting Moving Targets with Multiple Mobile Resources. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 957–964.

[7] Jiarui Gan, Bo An, and Yevgeniy Vorobeychik. 2015. Security Games with Protection Externalities. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*. 914–920.

[8] Manish Jain, Erim Kardes, Christopher Kiekintveld, Fernando Ordóñez, and Milind Tambe. 2010a. Security Games with Arbitrary Schedules: A Branch and Price Approach. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010.* http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1698

[9] Albert Xin Jiang, Zhengyu Yin, Chao Zhang, Milind Tambe, and Sarit Kraus. 2013. Game-theoretic Randomization for Security Patrolling with Dynamic Execution Uncertainty. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 207–214.

[10] Christopher Kiekintveld, Viliam Lisỳ, and Radek Píbil. 2015. Game-theoretic foundations for the strategic use of honeypots in network security. In *Cyber Warfare*. Springer, 81–101.

[11] Richard Klíma, Viliam Lisỳ, and Christopher Kiekintveld. 2015. Combining online learning and equilibrium computation in security games. In *International Conference on Decision and Game Theory for Security*. Springer, 130–149.

[12] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. 2010. Complexity of Computing Optimal Stackelberg Strategies in Security Resource Allocation Games. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010.*

[13] George Leitmann. 1978. On Generalized Stackelberg Strategies. *Journal of Optimization Theory and Applications* 26, 4 (1978), 637–643.

[14] Thanh Nguyen, Michael P Wellman, and Satinder Singh. 2017. A Stackelberg Game Model for Botnet Data Exfiltration. In *International Conference on Decision and Game Theory for Security*. Springer, 151–170.

[15] Thanh Hong Nguyen, Rong Yang, Amos Azaria, Sarit Kraus, and Milind Tambe. 2013. Analyzing the Effectiveness of Adversary Modeling in Security Games. In *AAAI*.

[16] James Pita, Manish Jain, Fernando Ordóñez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. 2009. Using Game Theory for Los Angeles Airport Security. *AI Magazine* 30, 1 (2009), 43–57.

[17] Ariel Rosenfeld and Sarit Kraus. 2017. When security games hit traffic: optimal traffic enforcement under one sided uncertainty. In *Proceedings of the 26th International Conference on Artificial Intelligence, IJCAI*.

[18] Ariel Rosenfeld, Oleg Maksimov, and Sarit Kraus. 2017. Optimizing Traffic Enforcement: From the Lab to the Roads. In *International Conference on Decision and Game Theory for Security*. Springer, 3–20.

[19] Sankardas Roy, Charles Ellis, Sajjan Shiva, Dipankar Dasgupta, Vivek Shandilya, and Qishi Wu. 2010. A survey of game theory as applied to network security. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*. IEEE, 1–10.

[20] Eric Shieh, Bo An, Rong Yang, Milind Tambe, Craig Baldwin, Joseph DiRenzo, Ben Maule, and Garrett Meyer. 2012. PROTECT: A Deployed Game Theoretic System to Protect the Ports of the United States. In *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, Valencia, Spain, June 4-8, 2012 (3 Volumes)*. 13–20.

[21] Eric Shieh, Manish Jain, Albert Xin Jiang, and Milind Tambe. 2013. Efficiently Solving Joint Activity Based Security Games. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 346–352.

[22] Sajjan Shiva, Sankardas Roy, and Dipankar Dasgupta. 2010. Game theory for cyber security. In *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*. ACM, 34.

[23] Pradeep Varakantham, Hoong Chuin Lau, and Zhi Yuan. 2013. Scalable Randomized Patrolling for Securing Rapid Transit Networks.. In *IAAI*.

[24] Bernhard Von Stengel and Shmuel Zamir. 2004. Leadership with Commitment to Mixed Strategies. (2004).

[25] Rong Yang, Christopher Kiekintveld, Fernando OrdóñEz, Milind Tambe, and Richard John. 2013. Improving Resource Allocation Strategies Against Human Adversaries in Security Games: An Extended Study. *Artificial Intelligence* 195 (2013), 440–469.

[26] Zhengyu Yin, Manish Jain, Milind Tambe, and Fernando Ordónez. 2011. Risk-Averse Strategies for Security Games with Execution and Observational Uncertainty. In *AAAI*.