Lightweight No-Regret Online Learning in Repeated Stackelberg Security Games

Guanda Chen, Yiding Ji, *Member, IEEE*, and Shuo Han, *Member, IEEE*

***Abstract*— This work investigates the problem of learning no regret defender strategies in repeated Stackelberg Security Games (SSGs) against an unknown sequence of attackers. We propose an efficient algorithm that combines the Decomposed Optimal Bayesian Stackelberg Solver (DOBSS) with the online combinatorial optimization algorithm Follow the Perturbed Leader (FPL). Specifically, the optimal strategy is dynamically computed by solving a mixed integer linear programing problem where random perturbation is strategically integrated with historical observations of the attacker sequence. Our algorithm provably achieves no-regret with respect to the optimal hindsight strategy and remains resilient against adversarial attacker sequences, even when an adversary is aware of the defender's algorithmic framework and intentionally selects sequences causing a higher regret. Additionally, we prove that our algorithm achieves exponential reductions in both space and time complexity compared with subprocedure of baseline methods. Comprehensive empirical studies confirm that our algorithm outperforms baseline methods by achieving lower regret bound and reduced computational cost.**

***Index Terms*— Stackelberg security games, mixed integer linear programming, online learning, complexity analysis, game theory.**

I. Introduction

STACKELBERG security games model the strategic interaction between defenders and attackers, where defenders allocate limited resources to protect critical information from being disclosed, meanwhile anticipate and mitigate adversarial behavior [1]. Some successful applications include surveillance patrol scheduling at Los Angeles International Airport [2], optimal resource allocation in urban transportation networks [3], protection of critical infrastructures [4] and wildlife conservation against illegal poaching [5]. Beyond physical security, these models are also extended to address cyber security challenges in various frontiers, such as control systems [6], [7], [8], [9], network resource allocation [10], [11], [12], and signaling scheme design [13], [14].

Recent years have seen an increasing interest in repeated Stackelberg Security Games (SSG) with unknown and where the adversarial attackers are unknown a priori. However, tractability or convergence of strategies often requires restrictive assumptions on the follower's strategy space or best response structure, including the follower's best response region [15], the Lipschitz continuity of utility functions [16], learning rewards [17], [18], [19], or specific parametric distributions of follower responses [20]. Although algorithms with no-regret guarantees have been proposed [21], [22], [23], they typically rely on computationally intensive sub-procedures, such as the enumeration and storage of an exponential number of vertices via membership oracles. This necessitates tracking performance metrics for each expert during the learning process, which significantly hinders scalability and practical applicability in tactical and dynamic environments.

To address the above mentioned limitations, we propose an online learning approach named FPL-DOBSS, which implicitly tracks the vertices of the defender's best-response polytopes without explicit enumeration or storage. The defender's utility function is piecewise linear over mixed strategy space, also remains linear within each polytope yet nonconvex across different polytopes. By exploiting the inherent structure of these polytopes, we reformulate the online learning task as a linear optimization problem with nonconvex constraints, which can be cast as a mixed integer linear program (MILP) based on adaptations of the DOBSS framework [24]. We adapt DOBSS, originally designed for Bayesian SSGs with known attacker distributions, to handle unknown and adversarial attacker sequences by using the empirical distribution of attacker types. By integrating randomized perturbations, our algorithm achieves a sublinear regret bound of $O(\sqrt{T})$ against adversarial attackers, ensuring that the average regret diminishes as $T \to \infty$. In particular, FPL-DOBSS achieves exponential reductions in both time and space complexity compared to vertex-enumeration-based approaches [22], as it avoids explicitly storing and tracking all vertices. Extensive numerical simulations further validate that our algorithm outperforms benchmarks in several attack scenarios.

Received 28 November 2025; revised 5 February 2026; accepted 1 March 2026. Date of publication 9 March 2026; date of current version 7 April 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62303389 and Grant 62373289, in part by Guangdong Basic and Applied Basic Research Funding under Grant 2024A1515012586, in part by Guangdong Scientific Research Platform and Project Scheme under Grant 2024KTSCX039, in part by the Youth Talent Support Program of Guangdong Association for Science and Technology under Grant SKXRC2025463, and in part by the Guangdong Provincial Key Lab of Integrated Communication, Sensing and Computation for Ubiquitous Internet of Things under Grant 2023B1212010007. This article has not been presented at a conference. Recommended by Senior Editor A. P. Aguiar. *(Corresponding authors: Yiding Ji; Shuo Han.)*

Guanda Chen and Yiding Ji are with the Robotics and Autonomous Systems Thrust, Systems Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China (e-mail: gchen553@connect.hkust-gz.edu.cn; jiyiding@hkust-gz.edu.cn).

Shuo Han is with the Department of Electrical and Computer Engineering, University of Illinois Chicago, Chicago, IL 60607 USA (e-mail: hanshuo@uic.edu).

Digital Object Identifier 10.1109/LCSYS.2026.3671784

The remainder of this paper is organized as follows. Section II introduces the repeated SSG model and formulates the key problem of this work. Section III proposes the no-regret online learning algorithm to design optimal defense strategies against unknown attacks. Section IV analyzes the regret bound and computational complexity of the algorithm. Section V presents an empirical study to demonstrate the performance of our approach against various adversaries. Finally, Section VI concludes the work.

II. Preliminaries and Problem Formulation

Repeated Stackelberg Security Games (SSGs) model the sequential interaction between a defender (leader) and a series of attackers (followers). The formal definition is presented as:

Definition 1 (Repeated Stackelberg Security Games): A repeated SSG is a tuple $\mathcal{G} = (T, \mathcal{N}, \mathcal{A}, \mathcal{D}, \mathcal{P})$ where

- T is the time horizon of the game, i.e., rounds of plays;
- $\mathcal{N} = \{1, \ldots, N\}$ is the set of targets;
- $\mathcal{A} = \{\alpha_1, \ldots, \alpha_K\}$ is the set of attacker types, where each type $\alpha_i \in \mathcal{A}$ has a unique payoff structure;
- $\mathcal{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$ is the set of pure strategies, where $\mathbf{d}_i$ indicates the resource coverage solely on target i;
- $\mathbf{p} = (p_1, \ldots, p_N) \in \mathcal{P}$ is a mixed strategy, that is, a probability distribution over the set of pure strategies $\mathcal{D}$ where p_i is the probability of taking $\mathbf{d}_i$. The mixed strategy space $\mathcal{P}$ is the simplex $\mathcal{P} = \left\{\mathbf{p} \in \mathbb{R}^N_{\geq 0} \mid \sum_{i=1}^N p_i = 1\right\}$.
- The defender and the attacker receive their respective payoffs when a target is attacked, specifically,
  - If target $i \in \mathcal{N}$ is *defended*, the *attacker* $\alpha_j \in \mathcal{A}$ *'s payoff* is given as $c^i_j \in [-1, 0]$, while the *defender's payoff* is $c^i_d \in [0, 1]$;
  - If target i is *not defended*, α_j*'s payoff* is $u^i_j \in [0, 1]$, while the *defender's payoff* is $u^i_d \in [-1, 0]$.

The attacker's *best response* is to attack the target with the highest expected utility. Formally, the best-response function is $b : \mathcal{A} \times \mathcal{P} \to \mathcal{N}$ where $b(\alpha_j, \mathbf{p}) = \arg\max_{i \in \mathcal{N}} \left\{c^i_j p_i + u^i_j (1 - p_i)\right\}$, with ties broken by selecting the target that maximizes the defender's utility. With respect to a specific attacker α_j, the defender's *expected utility function* is $f_{\alpha_j}(\mathbf{p}) = c_d^{b(\alpha_j, \mathbf{p})} p_{b(\alpha_j, \mathbf{p})} + u_d^{b(\alpha_j, \mathbf{p})}(1 - p_{b(\alpha_j, \mathbf{p})})$. Next, the *utility vector function* with respect to all attackers is given as $\mathbf{f}(\mathbf{p}) = \big(f_{\alpha_1}(\mathbf{p}), \ldots, f_{\alpha_K}(\mathbf{p})\big)$. In round t and under attacker α_j, the defender's *realized utility* is expressed as the inner product $\langle \mathbf{f}(\mathbf{p}), \mathbf{a}_t \rangle = \mathbf{f}(\mathbf{p})^\top \mathbf{a}_t$, where $\mathbf{a}_t = \mathbf{e}_j \in \{0, 1\}^K$ is a K-dimensional indicator vector.

When a repeated SSG is played, nature selects an *attacker sequence* $\{\mathbf{a}_t\}_{t=1}^N$ to inflict potential damage on the defender, which is potentially adversarial. Nature may behave either stochastically according to a fixed probability distribution or adversarially by choosing strategies to minimize utility of defender. The defender commits to strategy $\mathbf{p}_t$ before observing the type of attacker $\mathbf{a}_t$, which is revealed subsequently on round t. The *best strategy in hindsight* of the defender after T rounds is $\mathbf{p}^*_T = \arg\max_{\mathbf{p} \in \mathcal{P}} \langle \mathbf{f}(\mathbf{p}), \bar{\mathbf{a}}_T \rangle$, where $\bar{\mathbf{a}}_T = \frac{\sum_{t=1}^T \mathbf{a}_t}{T}$.

Under the settings of repeated SSGs in this work, the strategic interactions are framed with complete information, wherein the defender and each attacker have perfect knowledge of each other's utilities. This scenario frames SSG as a dynamic game, where the outcome of each round influences future rounds. The attacker's objective is to devise a strategy that maximizes their expected payoff, while the defender's goal is to optimize their strategy to protect the targets as effectively as possible.

A defender sequentially selects mixed strategies $\mathbf{p}_t \in \mathcal{P}$ against an unknown and potentially adversarial attacker sequence $\{\mathbf{a}_t\}_{t=1}^T$, using historical observations $\{\mathbf{a}_1, \ldots, \mathbf{a}_{t-1}\}$ before the current step. The defender's performance is measured by its *regret* against the best fixed strategy in hindsight:

$$\mathcal{R}_T = \sum_{t=1}^T \langle \mathbf{f}(\mathbf{p}^*_T), \mathbf{a}_t \rangle - \mathbb{E}\left[\sum_{t=1}^T \langle \mathbf{f}(\mathbf{p}_t), \mathbf{a}_t \rangle\right] \tag{1}$$

where expectation is over algorithm's internal randomness.

A representative no-regret approach is introduced in [22], which achieves no-regret convergence by formulating the problem within an expert learning framework. Each expert corresponds to a vertex of a strategy polytope—a convex region where every defender strategy induces the same fixed best response from the attacker. However, a major computational limitation of this approach is its reliance on enumerating an exponentially large number of such polytope vertices prior to the online learning phase. The pre-processing often becomes computationally intractable in practice, resulting in significant inefficiencies in both time and space, which ultimately restricts the algorithm's scalability and real-world applicability. Now we are ready to formulate the key problem of this work.

Problem 1 (Efficient Online Learning in Repeated SSGs): Consider a repeated SSG $\mathcal{G} = (T, \mathcal{N}, \mathcal{A}, \mathcal{D}, \mathcal{P})$ under a sequence of attackers $\{\mathbf{a}_t\}_{t=1}^T$ chosen by an adversary and unknown a priori. Our goal is to design an efficient online algorithm that achieves: (i) an upper bound $O(\sqrt{T})$ of regret $\mathcal{R}_T$; (ii) an exponential reduction in time and space complexity compared with the vertex enumeration procedure in [22].

III. Online Learning Algorithm

In this section, we augment the DOBSS framework with strategic perturbation to present an efficient online learning algorithm with a provable regret guarantee for Problem 1.

A. Efficient Computation via MILP Reformulation

In high-dimensional Stackelberg security games (SSGs), the complexity of computing the optimal strategy becomes prohibitively high due to the exponential growth of potential attacker-target combinations, which may scale as $O(N^K)$. The Decomposed Optimal Bayesian Stackelberg Solver (DOBSS) algorithm [25] provides an efficient solution for Bayesian Stackelberg games by reformulating the problem as a MILP, which assumes that the attacker type is stochastic and drawn from a known prior distribution $\mathbf{q}$. To defend adversarial attacker sequences with unknown prior distributions, we leverage DOBSS for problem reformulation where we replace the unknown prior $\mathbf{q}$ with the empirical distribution $\hat{\mathbf{q}}_t$ observed over the first t rounds of play. Accordingly, the following notions are defined as part of the DOBSS formulation:

- Defender's payoff when attacker α_j attacks target l while defender protects target i: $R^j_{il} = \begin{cases} c^i_d & \text{if } i = l \\ u^l_d & \text{otherwise} \end{cases}$.
- Attacker α_j's payoff when attacking target l while defender protects target i: $C^j_{il} = \begin{cases} c^l_j & \text{if } i = l \\ u^l_j & \text{otherwise} \end{cases}$.

- Attacker frequency at round t: $\hat{\mathbf{q}}_t = (\hat{q}_{1,t}, \ldots, \hat{q}_{K,t})$, where $\hat{q}_{j,t} = \frac{1}{t}\sum_{\tau=1}^{t} \mathbb{I}(\mathbf{a}_\tau = \mathbf{e}_j)$ denotes the observed frequency of attacker type α_j in the first t rounds.

$$\max_{\mathbf{z},\mathbf{b},\mathbf{a}} \quad \sum_{j=1}^{K}\sum_{i=1}^{N}\sum_{l=1}^{N} \hat{q}_{j,t} R_{il}^{j} z_{il}^{j} \tag{2a}$$

$$\text{s.t.} \quad \sum_{i,l} z_{il}^{j} = 1, \quad \sum_{l} z_{il}^{j} \le 1, \tag{2b}$$

$$\sum_{i} z_{il}^{j} = b_l^{j}, \quad \sum_{l} b_l^{j} = 1, \tag{2c}$$

$$a^{j} - \sum_{i} C_{il}^{j}\left(\sum_{h} z_{ih}^{j}\right) \le (1 - b_l^{j})M, \tag{2d}$$

$$\sum_{i} C_{il}^{j}\left(\sum_{h} z_{ih}^{j}\right) - a^{j} \le 0, \tag{2e}$$

$$z_{il}^{j} \in [0,1], \quad b_l^{j} \in \{0,1\}, \quad a^{j} \in \mathbb{R} \tag{2f}$$

Using the variable transformation $z_{il}^{j} = p_i b_l^{j}$ from DOBSS framework [22], Problem 1 can be written as optimization problem (2) above, with $b_l^{j} \in \{0,1\}$ indicating the target attacked by α_j. Specifically, a^j is an upper bound on attacker α_j's reward for any action and M is a sufficiently large constant. By the Big-M method in constraint 2d, the logical condition that the attacker selects the utility-maximizing action (best response) is effectively linearized. This reformulation enables efficient computation of the best hindsight strategy.

B. Online Learning With Stochastic Perturbation

Perturbation for Robustness Deterministic strategies based on empirical frequencies ($\hat{\mathbf{q}}_t$) expose the defenders to exploitation by adaptive adversaries that are capable to infer protection patterns. Thus, adversaries may anticipate protection patterns and attack uncovered targets. To mitigate this, we introduce stochastic perturbation through the follow the perturbed leader (FPL) framework [24], where the empirical frequency vector is augmented with a random perturbation:

$$\tilde{\mathbf{q}}_t = \hat{\mathbf{q}}_{t-1} + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \text{Uniform}\left[0, \frac{1}{\delta\sqrt{t}}\right]^{K} \tag{3}$$

where $\delta > 0$ is a scaling parameter controlling the perturbation magnitude. Crucially, this transformation preserves the linearity of the MILP objective function.

Unified Optimization Framework We substitute $\tilde{\mathbf{q}}_t$ into the DOBSS MILP, which yields the following problem:

$$\max_{\mathbf{z},\mathbf{b},a} \quad \sum_{j=1}^{K}\sum_{i=1}^{N}\sum_{l=1}^{N} \tilde{q}_{j,t} R_{il}^{j} z_{il}^{j}$$
$$\text{s.t.} \quad \text{Constraints from (2)} \tag{4}$$

The MILP structure remains unchanged, ensuring computational tractability. This procedure is formalized in Algorithm 1, which provides the complete computational implementation.

A critical challenge arises from the defender's nonlinear utility structure $f_{\alpha_j}(\mathbf{p}) = c_d^{b(\alpha_j,\mathbf{p})} p_{b(\alpha_j,\mathbf{p})} + u_d^{b(\alpha_j,\mathbf{p})}(1 - p_{b(\alpha_j,\mathbf{p})})$ where $b(\alpha_j, \mathbf{p})$ is the *best-response function* inducing discontinuity (piecewise-linearity) in $f_{\alpha_j}(\mathbf{p})$. This nonlinearity violates the linearity assumption in FPL analysis [24] and prevents the direct use of existing $O(\sqrt{T})$ regret bound results. Notably, we will establish in Section IV that our algorithm still achieves $\mathcal{R}_T = O(\sqrt{T})$ regret despite this complication.

Algorithm 1 FPL-DOBSS Algorithm Against Unknown Attacker Sequences

Input: Time horizon T, perturbation parameter δ, payoff matrices
Output: Defender strategy sequence $\{\mathbf{p}_t\}_{t=1}^{T}$

Initialize $\hat{\mathbf{q}}_1 \leftarrow \mathbf{0} \in \mathbb{R}^K$; **for** $t = 1$ ***to*** T **do**
  if $t > 1$ **then**
    $\hat{\mathbf{q}}_{t-1} \leftarrow \frac{1}{t-1}\sum_{\tau=1}^{t-1} \mathbf{a}_\tau$;
  Adversary selects $\mathbf{a}_t$;
  Sample $\boldsymbol{\epsilon}_t \sim \mathcal{U}\left[0, 1/\delta\sqrt{t}\right]^K$;
  Construct $\tilde{\mathbf{q}}_t \leftarrow \hat{\mathbf{q}}_{t-1} + \boldsymbol{\epsilon}_t$;
  Solve MILP: $\mathbf{z}^* \leftarrow \arg\max_{\mathbf{z},\mathbf{b}} \sum_{j=1}^{K}\sum_{i=1}^{N}\sum_{l=1}^{N} \tilde{q}_{j,t} R_{il}^{j} z_{il}^{j}$ subject to constraints in (2);
  Compute $\mathbf{p}_t \leftarrow \left(K^{-1}\sum_{j=1}^{K}\sum_{l=1}^{N} z_{il}^{j*}\right)_{i=1}^{N}$;
  Defender observes attacker $\mathbf{a}_t$; ▷ Observation after $\mathbf{p}_t$ is committed

IV. Algorithm Analysis

This section analyzes two central properties of Algorithm 1. Through linear transformation of the online optimization problem, we derive a sublinear regret bound for the defender of the algorithm. Then we prove that our algorithm has exponentially lower time and space complexity compared with the subprocedure of vertex enumeration based approach in [22].

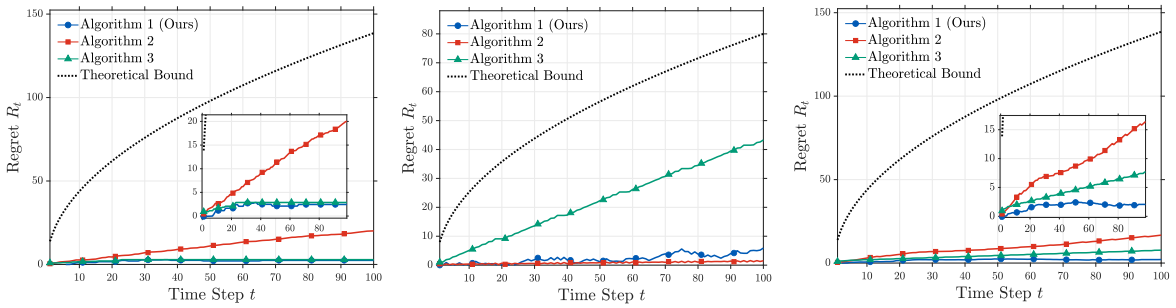
A. Regret Bound

Linear Transformation The online optimization problem at each round t is $\max_{\mathbf{p}} \sum_{\tau=1}^{t} \langle \mathbf{f}(\mathbf{p}), \mathbf{a}_\tau \rangle$. We reformulate this problem by introducing a decision variable $\mathbf{y} = [y_1, \cdots, y_j, \cdots, y_K]$, where each component $y_j = \sum_{i=1}^{N}\sum_{l=1}^{N} R_{il}^{j} z_{il}^{j}$ represents the expected payoff of the defender against the attacker α_j. Then the problem is stated as:

$$\max_{\mathbf{y}} \quad \langle \mathbf{y}, \hat{\mathbf{q}}_t \rangle,$$
$$\text{s.t.} \quad \mathbf{y} \in \mathcal{Y}, \tag{5}$$

where $\mathcal{Y} = \mathbf{f}(\mathcal{P})$ is the image of the original feasible region under the mapping $\mathbf{f}$ defined in Section II. Although the original utility vector function of the defender $\mathbf{f}(\mathbf{p})$ is nonlinear due to the best-response structure, our reformulated objective function becomes linear in terms of the new variable $\mathbf{y}$ and is subject to a non-convex feasible set.

Additionally, the above linear reformulation allows us to leverage the FPL framework to address Problem 1, which ensures a sublinear regret bound for linear optimization problems. Notably, FPL achieves sublinear regret for linear objectives over any decision set including non-convex ones, contingent on the existence of efficient optimization oracles. The DOBSS-based MILP just serves as such an oracle required by FPL [24]. Therefore, it is unnecessary to enumerate all vertices. We now recall a key result from [24] and it is instrumental to derive the regret bound of our approach.



(a) Stochastic Attacker (b) Non-adaptive Adversarial Attacker (c) Adaptive Adversarial Attacker

Fig. 1. Algorithm performance comparison under different scenarios.

Lemma 1: [24] Let $\mathbb{E}[\mathcal{R}_T(\delta)]$ denote the expected regret of **Hanna**(δ) up to time T. Follow Hannan lead and use gradually increasing perturbations on each period t, then (i) choose $\boldsymbol{\epsilon}_t$ uniformly at random from the cube $[0, 1/\delta\sqrt{t}]^K$; (ii) Use strategy $\arg\max_{\mathbf{y}}\langle \mathbf{y}, \tilde{\mathbf{q}}_t\rangle$. Then for any attacker sequence $\mathbf{a}_1, \mathbf{a}_2, \ldots$, after any number of periods $T > 0$, we have that

$$\mathbb{E}[\mathcal{R}_T(\delta)] \leq 2\delta RA\sqrt{T} + \frac{D\sqrt{T}}{\delta} \tag{6}$$

where the constants are $D = \sup_{\mathbf{p},\mathbf{p}'\in\mathcal{P}} \|\mathbf{f}(\mathbf{p}) - \mathbf{f}(\mathbf{p}')\|_1$, $R = \sup_{\mathbf{p},\mathbf{p}'\in\mathcal{P}} \|\mathbf{f}(\mathbf{p}) - \mathbf{f}(\mathbf{p}')\|_\infty$ and $A = \max_t \|\mathbf{a}_t\|_1$.

In the context of Stackelberg security games (SSGs), since $\mathbf{a}_t$ is a unit vector (e.g., $\mathbf{e}_j$), we have $A = 1$. Note that typically in SSGs, $D \geq 2K$ and $R \geq 2$ hold and they stem from the properties of the strategy space and payoff functions. Then Theorem 1 presents the regret bound of our approach and the proof is omitted since it is naturally derived from the lemma.

Theorem 1: The minimal expected regret after T rounds of strategies returned by Algorithm 1 is achieved by choosing $\delta = \sqrt{\frac{D}{2RA}}$ and $A = 1$, that is, $\mathbb{E}[\mathcal{R}_T] \leq 2\sqrt{2DRT}$. Therefore, the regret is $O(\sqrt{T})$ and dependents on constants D and R.

B. Computational Complexity

Proposition 1 (Reduced computational complexity): Algorithm 1 exponentially reduces both computation time complexity and space complexity compared with the vertex enumeration procedure in [22].

Proof: As is known, any extreme point of an N-dimensional convex polytope is the intersection of N linearly independent half-spaces of that polytope. The equality constraint $\sum_{i=1}^{N} p_i = 1$ reduces the number of independent dimensions to $N-1$. For a polytope defined by the attacker's specific best-response sequence, each attacker type requires $N-1$ linear inequalities to enforce that its designated best response yields higher utility than all other $N-1$ target responses. With K distinct attacker types, this results in a total of $K(N-1)$ such constraints. Thus the total number of hyperplanes is $N + K(N-1)$ as it should be non-negative.

In an $(N-1)$-dimensional space, each vertex is determined by the intersection of $N-1$ hyperplanes. The maximum number of vertices of a polyhedron defined by $N + K(N-1)$ half-spaces is of complexity class $O\left((N + K(N-1))^{N-1}\right)$. The time complexity of the reverse search algorithm [26] to enumerate all vertices of such a polytope is $O((N + K(N-1)) \cdot V \cdot N)$, where V denotes the number of vertices. In the worst case, this method yields a complexity of $O\left((N + K(N-1))^N \cdot N\right) = O\left(N^{N+1}K^N\right)$. Since there are N^K such polytopes to consider, the overall time complexity becomes $O\left(K^N N^{K+N+1}\right)$.

The DOBSS algorithm in [22] formulates a MILP with N^2K continuous variables, NK binary variables, and $O(NK)$ constraints. Due to the constraints 2c and 2f, it explores $O(N^K)$ nodes via a branch-and-bound procedure in the worst case, where each solves a linear programming of size $O(N^2K)$ [25]. The complexity for one round of computation via interior-point methods is $O(N^7K^{3.5}\log(1/\sigma))$, where σ is the error tolerance. In the worst case, the time complexity of our algorithm is $O\left(N^K \cdot N^7K^{3.5}\right) = O\left(N^{K+7}K^{3.5}\right)$ and exponentially lower than the vertex enumeration procedure in [22].

Regarding space complexity, the method in [22] stores $O\left((N + K(N-1))^{N-1}\right)$ vertices and tracks their weights throughout repeated SSG rounds. In comparison, our approach only stores $O(N^3K^2)$ entries in the constraint matrix per round. Thus, our algorithm also achieves an exponential reduction in space complexity, and only requires storing a polynomial number of elements per round, rather than maintaining an exponential number of expert vertices over all iterations. ■

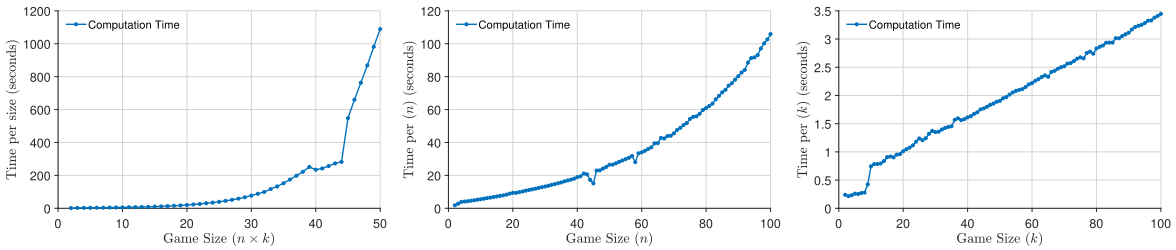
Remark 1: Note that the polytope associated with this problem possesses a specific structure, where each considered hyperplane involves at most two variables. This sparsity and limited variable involvement distinguish the polytopes in our framework from generic ones. Consequently, the worst-case complexity bound derived for general polyhedra may not be attainable in practice. The numerical experiments presented in Section V will demonstrate that our algorithm achieves exponentially lower computational time in application settings.

V. Numerical Experiments

This section provides comprehensive MATLAB simulations to validate the performance of our FPL-DOBSS algorithm. Specifically, Algorithm 1 is evaluated under three scenarios, i.e., stochastic, non-adaptive adversarial and adaptive adversarial attacker sequences, and against two baseline methods:

- **Balcan's algorithm** [22]: a vertex-enumeration-based algorithm proposed in [22] and is referred as Algorithm 2 for the sake of simplicity in the remainder of this paper.
- **Follow-the-Leader**: at round t, employs the best fixed strategy up to $t-1$, and is referred as Algorithm 3.

For Algorithms 1 and 3, we used YALMIP toolbox (version June 2023) [27] for optimization modeling and Gurobi v12.0.2 [28] as a solver. Algorithm 2 was implemented using vertex enumeration and MPT3 toolbox [29]. The perturbation parameter in Algorithm 1 is set to $\delta = \sqrt{K/2}$ and the online learning rate in Algorithm 2 is set to $\eta = 0.1$.



(a) Game Size: n=k (b) Game Size: k=2 (c) Game Size: n=2

Fig. 2. Computational performance of Algorithm 1 under different scenarios.

A. Stochastic Attacker Sequence

In the stochastic setting, i.e., Bayesian Stackelberg game, the attacker sequence is generated by independently sampling attacker types from a fixed but unknown distribution $\mathcal{F}$ over the type space $\mathcal{A}$. Specifically, for each round t, we have that

$$\mathbf{a}_t = \mathbf{e}_j \quad \text{with probability} \quad \mathbb{P}(\alpha_j) \quad \text{for some} \quad \alpha_j \in \mathcal{A},$$

where $\mathcal{F} = \{\mathbb{P}(\alpha_j)\}_{j=1}^{K}$ is a probability distribution satisfying $\sum_{j=1}^{K} \mathbb{P}(\alpha_j) = 1$. The defender has no prior knowledge of $\mathcal{F}$.

B. Non-Adaptive Adversarial Attacker Sequence

Similar to Example 2.2 in [30], the adversary designs a repeated SSG to induce linear regret of the defender. The utility matrix of the defender is given as $M_d = \begin{matrix} \\ T_1 \\ T_2 \end{matrix}\!\!\begin{matrix} \begin{matrix} A_1 & A_2 \end{matrix} \\ \begin{pmatrix} 0 & -1 \\ \gamma - 1 & 0 \end{pmatrix} \end{matrix}$

where T_i and A_i stand for the i-th protected and attacked target, respectively, with $i \in \{1, 2\}$. The attacker's payoff matrices when the target is uncorvered and covered are given as $M_a^u = \begin{matrix} \\ T_1 \\ T_2 \end{matrix}\!\!\begin{matrix} \begin{matrix} \alpha_1 & \alpha_2 \end{matrix} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{matrix}$ and $M_a^c = \begin{matrix} \\ T_1 \\ T_2 \end{matrix}\!\!\begin{matrix} \begin{matrix} \alpha_1 & \alpha_2 \end{matrix} \\ \begin{pmatrix} 0 & 0.5 \\ 0.5 & 0 \end{pmatrix} \end{matrix}$, respectively.

To generate an attacker sequence, we choose$\gamma = \frac{1}{m}$ where $m \in \mathbb{N}_+$ and *mod*stands for the operation for the remainder of the division. Then the attack sequence is set as:

$$\alpha_t = \begin{cases} \alpha_1 & \text{if } (((t-1) \mod (2m-1)) \mod 2) = 0 \\ \alpha_2 & \text{if } (((t-1) \mod (2m-1)) \mod 2) = 1. \end{cases}$$

We set $m = 10$ and the simulation results are in Figure 1. With this designated attacker sequence, Algorithm 3 is significantly inferior since its defense strategies are predictable even when the best fixed strategy in hindsight is implemented over the first $t-1$ rounds. An adversary exploits this by computing the defender's committed strategy and generating a sequence that forces the defender to take suboptimal actions in round t.

C. Adaptive Adversarial Attacker Sequence

This is a more general and adaptive form of the non-adaptive adversarial scenario. The adversary has knowledge of the defender's algorithmic framework and utilizes historical information such as attacker frequency $\hat{\mathbf{q}}_{t-1}$ to infer defender's decisions. According to the predicted defender's optimal strategy $\mathbf{p}_t^{\text{pred}}$, the adversary tries to harm the defender to the maximum extent by selecting the attacker type to cause the lowest expected utility of the defender in the next round.

Although Algorithm 1 is known to the adversary, it fails to determine the realization of the perturbation formula (3) as a result of randomness and nondisclosure to the outside. Instead, the adversary infers the defender's next strategy following a Follow-the-Leader (non-perturbed) algorithm. That is, it predicts the defender's strategy at round t by computing $\mathbf{p}_t^{\text{pred}} = \arg\max_{\mathbf{p} \in \mathcal{P}} \sum_{\tau=1}^{t-1} \langle \mathbf{f}(\mathbf{p}), \mathbf{a}_\tau \rangle$ and subsequently selects the attacker type α_j that minimizes the defender's expected utility in the next round, that is, $\alpha_t = \arg\min_{\alpha \in \mathcal{A}} f_\alpha(\mathbf{p}_t^{\text{pred}})$.

For Algorithm 2, the adversary is assumed to be aware that the defender employs an online algorithm based on vertex enumeration. Leveraging this knowledge, the adversary mimics the defender's update process by computing the loss incurred in each round and updating the weight distribution over the set of vertices (experts) online. Similar to the case of Algorithm 1, the adversary selects the attacker type to minimize the expected utility of the defender in the next round.

D. Simulation Results Analysis

Simulations in stochastic and adaptive adversarial scenarios are conducted in a 6×6 repeated SSG setting with randomly generated utility matrices. In non-adaptive adversarial scenario, we consider a 2×2 repeated SSG with utility matrices M_d, M_a^u and M_a^c defined in Section V-B. As shown in Figures 1a and 1c, our algorithm outperforms Algorithm 2 in high-dimensional games under stochastic and adaptive adversarial scenarios. Algorithm 2 requires online updates over an exponential number of experts, suffering from slower convergence rate and higher regret. In 2×2 non-adaptive adversarial scenario, our algorithm is marginally inferior to Algorithm 2. The regret of our algorithm quantifies the performance gap between our deterministic strategy and the best strategy in hindsight, whereas Algorithm 2's regret takes an expectation over all vertices, which is often infeasible in practice. Thus, the marginal gap of our algorithm is not conclusive.

Additionally, in graph 1b, Algorithm 3 exhibits linear regret when deceived by a pre-designed adversarial sequence, leading it to choose a suboptimal action at round t. In contrast, our algorithm and Algorithm 2 maintain sublinear regret due to their inherent stochasticity, which mitigates such deception.

In all three scenarios, our algorithm consistently achieves lower regret than the theoretical bound of Theorem 1, thereby confirming its sublinear regret property. Consequently, our algorithm qualifies as a no-regret online learning algorithm, even when confronted with an unknown adversarial sequence.

Then we conducted simulations to compare the efficiency of our FPL-DOBSS algorithm with vertex enumeration procedure of Algorithm 2 across repeated SSGs of varying sizes. As shown in Figure 3, Algorithm 1 exhibits marginally higher running times than the vertex enumeration procedure at small scales. However, as the game size increases, the vertex enumeration procedure becomes significantly more costly. In contrast, the running time of Algorithm 1 increases at an

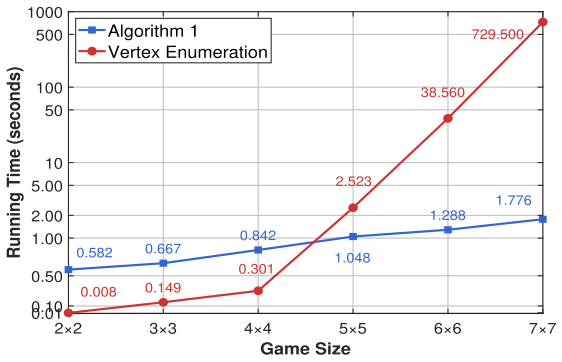


Fig. 3. Running time comparison in different size SSGs.

exponentially slower rate, which is consistent with the bound presented in Proposition 1. Therefore, Algorithm 1 demonstrates superior suitability for general applications, achieving exponential reductions in both computational time growth rate and parameter storage requirements compared to Algorithm 2.

Complementary simulations are provided to demonstrate the scalability of Algorithm 1. Figure 2 shows that the practical upper bound is approximately 45 for the larger case $N = K$, where the runtime is roughly 600 seconds. With a small number of targets (e.g., $N = 2$), our algorithm remains highly scalable even for a massive and heterogeneous pool of attackers ($K > 100$). The computational advantage facilitates the deployment of our approach in large-scale security domains with diverse attacker profiles where vertex enumeration based methods such as [22] becomes prohibitively expensive.

VI. Conclusion

This study investigates no-regret defender strategies in SSGs against unknown and potentially adversarial attacker sequences. We propose FPL-DOBSS algorithm that integrates DOBSS approach with FPL online learning framework. FPL-DOBSS achieves an expected regret bound of $O(\sqrt{T})$ against arbitrary attacker sequence, including adversarial sequences. By reformulating the optimization problem as a MILP, our approach exponentially reduces time and space complexity compared with conventional vertex enumeration based methods, which facilitate the real-time deployment in large-scale security domains. Extensive simulations in diverse scenarios further validate the performance of FPL-DOBSS in high dimensional SSGs against adversarial attacker sequences.

References

[1] A. Sinha, F. Fang, B. An, C. Kiekintveld, and M. Tambe, "Stackelberg security games: Looking beyond a decade of success," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 5494–5501.

[2] J. Pita et al., "Deployed ARMOR protection: The application of a game theoretic model for security at the Los Angeles international airport," in *Proc. 7th Int. Joint Conf. Auto. Agents Multiagent Syst.*, 2008, pp. 125–132.

[3] J. Tsai, S. Rathi, C. Kiekintveld, F. Ordonez, and M. Tambe, "Iris—A tool for strategic security allocation in transportation networks," in *Proc. 8th Intl. Joint Conf. Auto. Agents Multiagent Syst.*, 2009, pp. 37–44.

[4] E. Shieh et al., "PROTECT: A deployed game theoretic system to protect the ports of the United States," in *Proc. Int. Joint Conf. Auto. Agents Multiagent Syst.*, Jun. 2012, pp. 13–20.

[5] R. Yang, B. Ford, M. Tambe, and A. Lemieux, "Adaptive resource allocation for wildlife protection against illegal poachers," in *Proc. Int. Joint Conf. Auto. Agents Multiagent Syst.*, May 2014, pp. 453–460.

[6] P. Shukla, L. An, A. Chakrabortty, and A. Duel-Hallen, "A robust Stackelberg game for cyber-security investment in networked control systems," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 2, pp. 856–871, Mar. 2023.

[7] B. Cui, A. Giua, and X. Yin, "Towards supervisory control theory in tactical environments: A Stackelberg game approach," in *Proc. 62nd IEEE Conf. Decis. Control (CDC)*, Dec. 2023, pp. 7937–7943.

[8] S. Yang, H. Zheng, C.-I. Vasile, G. Pappas, and R. Mangharam, "STLGame: Signal temporal logic games in adversarial multi-agent systems," in *Proc. 7th Annu. Learn. Dyn. Control Conf.*, vol. 283. PMLR, 2025, pp. 1102–1114.

[9] J. Zhao, K. Zhu, M. Feng, S. Li, and X. Yin, "No-regret path planning for temporal logic tasks in partially-known environments," *Int. J. Robot. Res.*, vol. 44, no. 9, pp. 1526–1552, Aug. 2025.

[10] O. Vanek, Z. Yin, M. Jain, B. Bosanský, M. Tambe, and M. Pechoucek, "Game-theoretic resource allocation for malicious packet detection in computer networks," in *Proc. Int. Joint Conf. Auto. Agents Multiagent Syst.*, Jun. 2012, pp. 905–912.

[11] R. Bai, H. Lin, X. Yang, X. Wu, M. Li, and W. Jia, "Stackelberg security games with contagious attacks on a network: Reallocation to the rescue," *J. Artif. Intell. Res.*, vol. 77, pp. 487–515, Jun. 2023.

[12] J. Gan, E. Elkind, S. Kraus, and M. Wooldridge, "Defense coordination in security games: Equilibrium analysis and mechanism design," *Artif. Intell.*, vol. 313, Dec. 2022, Art. no. 103791.

[13] M. Castiglioni, A. Celli, A. Marchesi, and N. Gatti, "Online Bayesian persuasion," in *Proc. Adv. neural Inf. Process. Syst.*, vol. 33, 2020, pp. 16188–16198.

[14] M. Bernasconi, M. Castiglioni, A. Celli, A. Marchesi, F. Trovò, and N. Gatti, "Optimal rates and efficient algorithms for online Bayesian persuasion," in *Proc. Intl. Conf. Mach. Learn.*, 2023, pp. 2164–2183.

[15] B. Peng, W. Shen, P. Tang, and S. Zuo, "Learning optimal strategies to commit to," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 2149–2156.

[16] C. Maheshwari, J. Cheng, S. Sastry, L. Ratliff, and E. Mazumdar, "Follower agnostic learning in Stackelberg games," in *Proc. IEEE 63rd Conf. Decis. Control*, Aug. 2024, pp. 222–228.

[17] N. Lauffer, M. Ghasemi, A. Hashemi, Y. Savas, and U. Topcu, "No-regret learning in dynamic Stackelberg games," *IEEE Trans. Autom. Control*, vol. 69, no. 3, pp. 1418–1431, Mar. 2024.

[18] I. Anagnostides, I. Panageas, G. Farina, and T. Sandholm, "On the convergence of no-regret learning dynamics in time-varying games," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 16367–16405.

[19] M. Zhang, P. Zhao, H. Luo, and Z.-H. Zhou, "No-regret learning in time-varying zero-sum games," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 26772–26808.

[20] J. Wu, W. Shen, C.-W. Lee, H. Luo, and K. Wang, "Inverse game theory for Stackelberg games: The blessing of bounded rationality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 32186–32198.

[21] P. G. Sessa, I. Bogunovic, M. Kamgarpour, and A. Krause, "Learning to play sequential games versus unknown opponents," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 8971–8981.

[22] M.-F. Balcan, A. Blum, N. Haghtalab, and A. D. Procaccia, "Commitment without regrets: Online learning in Stackelberg security games," in *Proc. 16th ACM Conf. Econ. Comput.*, Jun. 2015, pp. 61–78.

[23] M.-F. Balcan, K. Harris, and Z. Wu, "Regret minimization in Stackelberg games with side information," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 12944–12976.

[24] A. Kalai and S. Vempala, "Efficient algorithms for online decision problems," *J. Comput. Syst. Sci.*, vol. 71, no. 3, pp. 291–307, Oct. 2005.

[25] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus, "Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games," in *Proc. Int. Joint Conf. Auto. Agents Multiagent Syst.*, May 2008, pp. 895–902.

[26] D. Avis and K. Fukuda, "A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra," in *Proc. 7th Annu. Symp. Comput. geometry - SCG*, 1991, pp. 98–104.

[27] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2004, pp. 284–289.

[28] *Gurobi Optimizer Reference Manual*, Gurobi Optimization, LLC, Beaverton, OR, USA, 2024.

[29] M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2013, pp. 502–510.

[30] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, Mar. 2012.