# Sensor Scheduling in Intrusion Detection Games with Uncertain Payoffs

**Jayanth Bhargav**                                                    JBHARGAV@PURDUE.EDU
**Shreyas Sundaram**                                                SUNDARA2@PURDUE.EDU
**Mahsa Ghasemi**                                                        MAHSA@PURDUE.EDU
*Elmore Family School of Electrical and Computer Engineering, Purdue University, IN 47907 USA*

## Abstract

We study the problem of sensor scheduling for an intrusion detection task. We model this as a 2-player zero-sum game over a graph, where the defender (Player 1) seeks to identify the optimal strategy for scheduling sensor orientations to minimize the probability of missed detection at minimal cost, while the intruder (Player 2) aims to identify the optimal path selection strategy to maximize missed detection probability at minimal cost. The defender's strategy space grows exponentially with the number of sensors, making direct computation of the Nash Equilibrium (NE) strategies computationally expensive. To tackle this, we propose a distributed variant of the Weighted Majority algorithm that exploits the structure of the game's payoff matrix, enabling efficient computation of the NE strategies with provable convergence guarantees. Next, we consider a more challenging scenario where the defender lacks knowledge of the true sensor models and, consequently, the game's payoff matrix. For this setting, we develop online learning algorithms that leverage bandit feedback from sensors to estimate the NE strategies. By building on existing results from perturbation theory and online learning in matrix games, we derive high-probability order-optimal regret bounds for our algorithms. Finally, through simulations, we demonstrate the empirical performance of our proposed algorithms in both known and unknown payoff scenarios.

**Keywords:** Online Learning, Matrix Games, Sensor Scheduling, Finite-Sample Convergence

## 1. Introduction

In the modern landscape of security and surveillance, efficient deployment and scheduling of sensors is critical in ensuring comprehensive monitoring and intrusion detection (Guvensan and Yavuz (2011); Murray et al. (2007)). In managing sensing resources, the challenge involves not only their placement (Dhillon and Chakrabarty (2003); Bhargav et al. (2023); Jourdan and Roy (2008)), but also dynamically scheduling their orientations to maximize detection efficiency while minimizing resource usage and operational costs (Osais et al.). Traditionally, sensor scheduling has been approached through various optimization techniques, yet these methods often fall short in adversarial settings where an intelligent intruder (non-oblivious) can actively attempt to evade detection. In such scenarios, game theory offers a robust framework for modeling and solving the intricate interactions between the defending system and the intruding adversary (Pirani et al. (2021); An et al. (2017)). Furthermore, the defending system may not have accurate knowledge of sensor performance due to the inherent complexity of modeling certain sensors, such as cameras, or the lack of prior knowledge about detection probabilities. These uncertainties lead to discrepancies between the simulated and real-world game dynamics, requiring the defending system to learn the true game online, and adaptively refine strategies using feedback from sensors to ensure effective detection and maintain performance despite the model uncertainties.

### 1.1. Related Work

In this section, we review existing works on security games, methods for solving large-scale games, and online learning in games, highlighting how these approaches relate to our contributions.

**Security Games:** Many works in the literature have modeled security resource allocation tasks using the Stackelberg game framework (Sinha et al. (2018); Korzhyk et al. (2010)). In Paruchuri et al. (2008), the authors model a patrolling task to protect airports from attackers as a Stackelberg game, where the defender commits to a mixed strategy for allocating patrolling resources and the attacker best-responds with a pure attack strategy. However, we consider the setting where both the players commit to mixed-strategies, and operate under concurrent decision-making conditions. Since neither of the players can anticipate or respond to the other's exact strategy in real-time, the Stackelberg and Nash equilibria are equivalent in our game setting (Korzhyk et al. (2011)).

**Solving Large Games:** A line of work focuses on developing efficient techniques for approximately solving games with large strategy spaces (Ganzfried and Sandholm (2015); Lipton et al. (2003); Sandholm (2015)). Li and Wellman (2020) introduce an iterative structure-learning approach to search for approximate solutions of many-player games. In Li et al. (2023) and Brown et al. (2019), the authors study team-adversary games with combinatorial action spaces and propose a counterfactual regret minimization framework for learning individual strategies. Along the lines of these works, we propose a distributed variant of the well-known Weighted Majority algorithm (proposed in Littlestone and Warmuth (1994)) which leverages the structure in the game's payoff function, to efficiently and iteratively estimate the NE, and establish convergence guarantees.

**Online Learning in Games:** In Fiez et al. (2021), the authors study online learning in periodic zero-sum games and establish guarantees for convergence to equilibrium strategies. The authors of Bailey and Piliouras (2019) study online learning in $2 \times 2$ zero-sum games and provide convergence guarantees for gradient descent dynamics. However, for a general class of zero-sum games, the dynamics of the equilibrium strategies generated by the learning process can be very intricate and instances may fail to converge (Andrade et al. (2021); Cheung and Piliouras (2019)). Two closest papers to our work on learning in matrix games with bandit feedback are: O'Donoghue et al. (2021) - where the authors propose a UCB style algorithm for no-regret learning in matrix games; and Chen et al. (2024) - where the authors present an adaptation of online mirror descent algorithm with last-iterate convergence rates. The key distinction of our work is the application to the intrusion detection setting, where we exploit the combinatorial structure of the game matrix arising from sensor and path configurations to establish tighter regret bounds and faster convergence guarantees.

### 1.2. Contributions

We begin by modeling the sensor scheduling task as a zero-sum matrix game. First, we propose a distributed variant of the Weighted Majority algorithm that leverages the structure of the game's payoff matrix to efficiently compute NE strategies, even in exponentially large strategy spaces, with provable convergence guarantees. Second, recognizing that the defender may lack accurate knowledge of the true sensor models and hence the game's payoff matrix, we develop online learning algorithms that rely on bandit feedback from the sensors to adaptively estimate the game matrix and refine the defender's sensor scheduling strategies. By building on results from perturbation theory and online learning in matrix games, we derive high-probability order-optimal regret bounds for these algorithms. Finally, we demonstrate the empirical performance of our methods through numerical simulations, for both known and unknown payoff scenarios.

## 2. Problem Formulation

We model the game environment as a graph $G = (N, E)$, where the set of nodes $N$ denotes the regions of an environment, and the set of edges $E$ denotes the possible transitions for the intruder between the regions. Let $\mathcal{S} = \{s_1, \ldots, s_p\}$, with $|\mathcal{S}| = p$, denote the set of sensors deployed on a subset of nodes in the graph. Each sensor has $d$ possible orientations, $\Theta = \{\theta_1, \ldots, \theta_d\}$, with $|\Theta| = d$. Let $c_k^q \in \mathbb{R}_{\geq 0}$ be the cost of orienting the sensor $s_q \in \mathcal{S}$ in the direction $\theta_k \in \Theta$. Each sensor covers a subset of nodes in the graph depending on its orientation, which may not be restricted to its neighboring nodes. Let $I = \{i_1, i_2, \ldots, i_m\}, m = |I|$, denote the set of joint strategies of the defender, i.e., a set of all possible sensor orientations, and let $\{c_1, c_2, \ldots, c_m\}$ be the costs associated with each joint strategy. The cost $c_i$ of a joint strategy $i \in I$ is the sum of the individual costs associated with each sensor orientation in that strategy, i.e., $c_i = \sum_{q=1}^{p} c_{k_{q,i}}^q$, where $k_{q,i}$ is the orientation of sensor $s_q$ in the joint strategy $i$. We note that the number of joint strategies is $m = |\Theta|^{|\mathcal{S}|}$, which is exponential in the number of sensors. We assume that the intruder starts at a source node $S$ and aims to reach a terminal node $T$ in the graph, and has a finite set of possible paths which it can take. We denote this by $J = \{j_1, j_2, \ldots, j_n\}, n = |J|$, which is the set of pure strategies (paths) of the intruder[1], and let $\{r_1, r_2, \ldots, r_n\}, r_j \in \mathbb{R}_{\geq 0}$, be the associated costs of the respective paths. Let $X$ and $Y$ denote the set of mixed-strategies for the defender and intruder, respectively. We have the following: $X = \left\{x \in \mathbb{R}^m; \sum_{i \in I} x_i = 1, x \geq 0\right\}; Y = \left\{y \in \mathbb{R}^n; \sum_{j \in J} y_j = 1, y \geq 0\right\}$. Each sensor $s_q \in \mathcal{S}$ has a probability of detection $p_{detect,q}$. Specifically, if the intruder traverses through a node which is covered by a sensor $s_q$, it will be detected by that sensor with a probability $p_{detect,q}$, and will go undetected by that sensor with probability $(1 - p_{detect,q})$. We now make the following assumption on the sensor models.

**Assumption 1:** The sensors are imperfect with $p_{detect,q} \in [p_{min,q}, p_{max,q}]$, where $0 < p_{min,q} < p_{max,q} < 1, q = 1, \ldots, p$. Furthermore, a detection event from a sensor $s_q$ is independent of detection events from other sensors in $\mathcal{S} \setminus \{s_q\}$.

The goal of the defender is to minimize the probability of not detecting the intruder. If a certain node is covered by multiple sensors and the intruder has traversed that node, then the overall probability of missed detection at that node is the product of the probability of missed detection of all the sensors covering that node. For a scheduling strategy $i \in I$ and a path $j \in J$, let $V_{ij}^q$ be the number of nodes that are covered by the sensor $s_q$ in the orientation strategy $i \in I$ and contained in the intruder's path $j \in J$. The overall probability of missed detection is given by:

$$p_{miss}(i, j) = \prod_{q=1}^{p} (1 - p_{detect,q})^{V_{ij}^q}. \tag{1}$$

We consider the additive log-likelihood representation for the probability of missed detection (as multiplicative operations in (1) can lead to underflow and numerical instability), given by

$$\log(p_{miss}(i, j)) = \sum_{q=1}^{p} V_{ij}^q \log(1 - p_{detect,q}). \tag{2}$$

---

1. If the set of paths is not known a priori, the structure of the underlying graph can be leveraged to generate a diverse set of candidate paths (e.g., see Ghasemi et al. (2021)), which can serve as a proxy for the intruder's strategy space. This enables the defender to reason about and prepare for potential adversarial behaviors. In this setting, the graph becomes a crucial modeling tool for approximating the intruder's possible actions based on the environment's topology.

Let $A, B \in \mathbb{R}^{m \times n}$ denote the payoff matrices for the defender and intruder, respectively. The payoff values for a pure strategy pair $(i, j)$ for the defender and intruder are given by $A_{ij} = \log(p_{miss}(i,j)) + c_i$ and $B_{ij} = -\log(p_{miss}(i,j)) + r_j$, respectively. The defender plays $x \in X$, the intruder plays $y \in Y$, and they receive payoffs $x^\top A y$ and, $x^\top B y$ respectively. The defender aims to find the mixed strategy $x^*$ that minimizes the expected value of its payoff, given the mixed strategy $y$ of the intruder, i.e., the defender aims to solve the following optimization problem:

$$x^* = \underset{x \in X}{\arg\min}\, x^\top A y. \tag{3}$$

Similarly, given the mixed strategy $x$ of the defender, the intruder aims to solve the following:

$$y^* = \underset{y \in Y}{\arg\min}\, x^\top B y. \tag{4}$$

This will form a bimatrix game which we denote by the tuple: $\mathcal{G} = \{A, B\}$. Note that $\mathcal{G}$ is a non-zero sum game, since $A + B \neq 0$ in general. A bimatrix game $\mathcal{G} := \{A, B\}$ is zero-sum if $A + B = 0$. The solution to a zero-sum game, i.e., the Nash equilibrium strategies $(x^*, y^*)$, can be obtained by solving the nested min-max optimization problem below:

$$(x^*, y^*) \in \underset{x \in X}{\arg\min}\, \underset{y \in Y}{\arg\max}\, x^\top A y. \tag{5}$$

**Definition 1 (Nash Equilibrium)** *A pair of mixed-strategies $(x^*, y^*)$ is a Nash Equilibrium of the zero-sum game $A$ if $x^{*\top} A y^* \leq x^\top A y^*$ and $x^{*\top} A y^* \geq x^{*\top} A y$, $\forall x \in X, \forall y \in Y$.*

Consider the following $\mathcal{G}' := \{A', B'\}$, where matrices $A'$ and $B'$ are such that

$$A'_{ij} = A_{ij} - r_j; \quad B'_{ij} = B_{ij} - c_i. \tag{6}$$

Based on the results presented in Section 2.1 of Kannan and Theobald (2010), we have the following result that characterizes the equilibrium property of $\mathcal{G}'$. All detailed proofs supporting our theoretical results can be found in Appendix B of Bhargav et al. (2025).

**Proposition 2** *The game $\mathcal{G}' := \{A', B'\}$ is zero-sum and has the same set of Nash Equilibria as $\mathcal{G}$.*

We consider the setting in which the defender has knowledge of $c_i$'s and $r_j$'s for all $i \in I$ and $j \in J$. However, the defender may not know the sensor detection probabilities $p_{detect}$, consequently the game payoff matrix $A$, accurately. Proposition 2 also implies that an $\epsilon$-NE[2] of $(A', B')$ is also an $\epsilon$-NE for the original game $(A, B)$. With a slight abuse of notation, we will denote the modified game (which is zero-sum) by $\mathcal{G}$, which can be completely specified by the payoff matrix $A$, where $A_{ij} = \log(p_{miss}(i,j)) + c_i - r_j$.

**Remark 3** *We assume that the entries of $A$ are bounded in $[0, 1]$. If this is not the case, we normalize $A$ using its maximum and minimum values, $A_{min} = \min_{i,j} A_{ij}$ and $A_{max} = \max_{i,j} A_{ij}$, respectively. This normalization preserves the Nash equilibrium strategies, as scaling and shifting the payoffs affect only the game's value, not the equilibrium strategies themselves. Additionally, we note that an $\epsilon$-NE of the normalized game is an $\epsilon'$-NE of the un-normalized game, where $\epsilon' = (A_{max} - A_{min})\epsilon$. Therefore, it is sufficient to restrict all analysis to the normalized zero-sum game, as it fully captures the essential properties and results of the original game.*

---

2. A pair of strategies $(x', y')$ is a $\epsilon$-NE of $A$ if $x'^\top A y' \leq x'^\top A y + \epsilon$ and $x'^\top A y' \geq x^\top A y' - \epsilon$, $\forall x \in X, \forall y \in Y$.

## 3. Distributed Weighted Majority Algorithm

In this section, we propose a fast and scalable algorithm which leverages the structure of the game matrix $A$ for efficiently solving for the NE strategies. For a moderate number of strategies, the min-max optimization problem in (5) can be solved to compute the NE strategies using a bi-level Linear Programming (LP) formulation Nehme (2009). However, for the sensor scheduling problem considered in this paper, the defender has an exponentially sized set of scheduling strategies. For example, if $d = |\Theta| = 4$ (i.e., each sensor has 4 possible orientations) and $|\mathcal{S}| = 10$ (i.e., there are 10 sensors), then $|I| = 4^{10} = 1,048,576$. Solving for the exact NE using an LP can become computationally intractable due to a large number of variables and constraints. To this end, we present the Distributed Weighted Majority (DWM) algorithm (Algorithm 1), a variant of the *Weighted Majority (WM)* algorithm originally developed by Littlestone and Warmuth (1994). The DWM algorithm leverages the structure of the game to perform multiplicative weight updates locally for each sensor, which will then be aggregated to estimate the joint scheduling strategy for the defender.

First, we define the set of sub-game payoff matrices $\mathcal{A} = \{A^1, A^2, \ldots, A^p\}$, where $A^q \in \mathbb{R}^{d \times n}$ corresponds to sensor $s_q \in \mathcal{S}$, given by

$$A_{kj}^q = V_{kj}^q \times \log(1 - p_{detect,q}) + c_k^q, \tag{7}$$

where $V_{kj}^q$ contains all the nodes that are covered by the sensor $s_q$ oriented in the direction $\theta_k$ and contained in the intruder path $j$. At each iteration $t$ and for each sensor $s_q \in \mathcal{S}$, we maintain a set of weights $\sigma_t^q \in \mathbb{R}^d$, where $d = |\Theta|$, which are proportional to the likelihood of selecting different orientations for the sensor. The joint sensor scheduling strategy $i \in I$ can be denoted as $i = [k_{1,i}, k_{2,i}, \ldots, k_{p,i}]$, where $k_{q,i} \in \{\theta_1, \ldots, \theta_d\}$ is the orientation of the sensor $s_q \in \mathcal{S}$ in the joint strategy $i \in I$. We update the weights for each joint strategy $i \in I$ as follows:

$$w_t(i) = \prod_{q=1}^{p} \sigma_t^q(k_{q,i}), \tag{8}$$

where $\sigma_t^q(k_{q,i})$ is the weight assigned to the orientation $\theta_k$ for the sensor $s_q$ in the joint strategy $i \in I$. At each iteration $t$, the weights $\sigma_t^q(k_{q,i})$ are updated using the multiplicative rule in (10) using a learning rate $\beta \in (0,1)$ and the loss per sensor $(s_q \in \mathcal{S})$ for each individual strategy $\theta_k \in \Theta$, denoted as $\ell_t^q(\theta_k)$, given by

$$\ell_t^q(\theta_k) = \sum_{j \in J} A_{kj}^q y_t(j) - \frac{1}{p} \sum_{j \in J} r_j y_t(j). \tag{9}$$

Based on the results in Sections 2.4 and 2.5 of Freund and Schapire (1996), we have the following theoretical guarantees for Algorithm 1.

**Theorem 4** *Let $\{x_1, \ldots, x_T\}$ and $\{y_1, \ldots, y_T\}$ be the mixed strategies returned by Algorithm 1 after $T$ iterations with $\beta = \left(1 + \sqrt{\frac{2|\mathcal{S}|ln|\Theta|}{\tilde{L}}}\right)^{-1}$, where $\tilde{L} \geq \min_{i \in I} \sum_{t=1}^{T} \sum_{q=1}^{p} \ell_t^q(\theta_k)$. Let $\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x_t$ and $\bar{y} = \frac{1}{T}\sum_{t=1}^{T} y_t$. Then, $\bar{x}$ and $\bar{y}$ are valid mixed-strategies for the two players, respectively. The pair $(\bar{x}, \bar{y})$ approximates the game's Nash Equilibrium value within $\epsilon_T$, given by*

$$\epsilon_T = \frac{\sqrt{2\tilde{L}|\mathcal{S}|ln|\Theta|}}{T} + \frac{|\mathcal{S}|ln|\Theta|}{T}. \tag{11}$$

5

---

**Algorithm 1** Distributed Weighted Majority (DWM) Algorithm

---

**Data:** Intrusion Detection Game $\mathcal{G}$, Parameter $\beta \in (0, 1)$, Number of Iterations $T$
**Result:** Mixed-strategies: $\{x_1, \ldots, x_T\}, \{y_1, \ldots, y_T\}$
**Initialize:** $\sigma_0^q(\theta_k) = 1/|\Theta|, k = 1, ..., |\Theta|; w_0(i) = 1/|I|, i = 1, ..., m$

**for** $t = 1, \ldots, T$ **do**

Compute defender's mixed-strategy $x_t(i) = \frac{w_t(i)}{\sum_{i' \in I} w_t(i')}, \forall i \in I$

Estimate intruder's mixed-strategy: $y_t = \max_{y \in Y} x_t^\top A y$

Compute per sensor loss $\ell_t^q(\cdot)$ as in Equation (9)

Update weights for each sensor

$$\sigma_{t+1}^q(k_{q,i}) = \sigma_t^q(k_{q,i})\beta^{\ell_t^q(\theta_k)}; \quad k = 1, \ldots, |\Theta|, q = 1, ..., p \tag{10}$$

Update joint strategy weights: $w_{t+1}(i) = \prod_{q=1}^p \sigma_{t+1}^q(k_{q,i}), \forall i \in I$

**end**

---

$\tilde{L}$ in Theorem 4 is an upper bound on the sum of losses over $T$ iterations, which can be naively bounded by $T$. For a desired approximation $\epsilon$, running Algorithm 1 for $T$ iterations (where $T$ can be computed using (11)), with the specified learning rate $\beta$, will yield an $\epsilon$-NE strategy $(\bar{x}, \bar{y})$.

**Remark 5** *The DWM algorithm (Algorithm 1) exploits the additive structure of the game's payoff function, achieving a substantial computational advantage by performing only $\mathcal{O}(|\mathcal{S}||\Theta|)$ multiplicative updates in parallel (i.e., $\mathcal{O}(|\Theta|)$ updates for each sensor $s \in \mathcal{S}$ in parallel), to estimate the NE strategy for the defender. In contrast, directly applying the WM algorithm requires $\mathcal{O}(|\Theta|^{|\mathcal{S}|})$ updates, which cannot be parallelized, making DWM far more efficient for large-scale instances. Furthermore, in Line 2 of Algorithm 1, we estimate the intruder's mixed strategy $y_t$ by solving a standard LP, as the intruder's strategy space is significantly smaller than that of the defender $(n << m)$. However, if the intruder's strategy space is also large, the multiplicative weight update rule can be applied to update $y_t$. This approach, however, results in a significantly slower convergence rate compared to a single player using the multiplicative weight update.*

## 4. Online Learning and Adaptation for Unknown Sensor Models

In this section, we extend our study of the sensor scheduling problem to scenarios where the defender lacks knowledge of the true sensor models and, consequently, the game's payoff matrix. We present online learning algorithms with performance guarantees for both homogenous and heterogeneous sensor settings, for refining the defender's strategies based on bandit feedback.

Since the defender does not have accurate knowledge of the game's payoff, it maintains an estimate of the game matrix, which is updated iteratively using feedback from sensor observations. At each time step $t$, the defender computes a mixed strategy $x_t \in X$ for scheduling sensors, based on its current estimate of the game matrix. The defender and intruder play pure strategies $(i_t, j_t)$, sampled from their mixed-strategy distributions $(x_t, y_t)$ respectively. The defender receives feedback from the sensors monitoring the intruder's path, indicating whether or not the intruder was detected. We will first consider the setting with homogenous sensors, where the game matrix $A$ can be parametrized by a single (unknown) Bernoulli distribution with parameter $p_{detect}$. As a result, the game matrix can now be defined as $A_{ij} = V_{ij} \log (1 - p_{detect}) + c_i - r_j$, where $V_{ij} = \sum_{q=1}^p V_{ij}^q$.

We define $\mathcal{F}_t = ((i_1, j_1), \ldots, (i_t, j_t))$ to be the sequence of observations available to the player at round $t$. Two aspects of this problem distinguish it from other setups considered in the literature (Cheung and Piliouras (2019); Andrade et al. (2021); O'Donoghue et al. (2021)). Firstly, the defender receives the pure strategy played by the intruder ($j_t$) as feedback (as opposed to $y_t$), and secondly, the intruder does not have accurate knowledge of the game payoff function/matrix. Additionally, our setting is different from the one considered in O'Donoghue et al. (2021), where the players receive a noisy feedback of the payoff, and the noise process is assumed to be 1-sub-Gaussian with a zero mean. At each time step $t$, the defender updates its estimate of $\hat{p}^t_{detect}$ of the true parameter $p_{detect}$ and estimates the game matrix $\hat{A}_t$ (see Algorithm 2). Another key distinction from O'Donoghue et al. (2021) is that our problem setting under homogenous sensors does not need complete exploration of the strategy spaces. We introduce a simpler algorithm that eliminates the need for Upper Confidence Bound (UCB) estimates for each entry of the $A$ matrix, as the exploration-driven optimism framework is not required in this scenario. Consequently, we establish tighter regret bounds that do not depend on the strategy spaces, i.e., ($\tilde{\mathcal{O}}(\sqrt{T})$ instead of $\tilde{\mathcal{O}}(\sqrt{|I||J|T})$ ).

---

**Algorithm 2** Online Learning and Adaptation of Strategies for Homogenous Sensors

---

Feedback samples: $\mathcal{H} = \emptyset$

**for** $t = 1, \ldots, T$ **do**

    Compute sample average $\hat{p}^t_{detect}$ using samples in $\mathcal{H}$

    Clipping estimator: $\hat{p}^t_{detect,l} = \min(\max(\hat{p}^t_{detect,l}, p_{min,l}), p_{max,l}); l = 1, \ldots p$

    Solve the estimated game $\hat{A}_t$, where $\hat{A}_t(i, j) = V_{ij} \log(1 - \hat{p}^t_{detect}) + c_i - r_j$:

$$(x^*_t, y^*_t) \in \arg\min_{x \in X} \arg\max_{y \in Y} x^\top \hat{A}_t y \quad \text{(using Algorithm 1)} \tag{12}$$

    Play pure strategy $i_t \sim x^*_t$ and observe intruder's pure strategy $j_t \sim y_t$

    Collect $V_{i_t, j_t}$ samples from sensor feedback into $\mathcal{H}$

**end**

---

In Algorithm 2, we use a clipped estimator for the parameter $\hat{p}^t_{detect}$, which keeps the estimate within a feasible range $[p_{min}, p_{max}]$ ; this ensures stability in small sample sizes without affecting long-term convergence to the true parameter $p_{detect}$. Furthermore, in each iteration of Algorithm 2, the estimated game (as in (12)) can be efficiently solved for NE strategies using Algorithm 1.

We will perform analysis from the perspective of a single player, i.e., the defender, who does not control the actions of the opponent (i.e., the intruder). Let $(x^\dagger, y^\dagger)$ denote a pair of NE strategies of the game $\mathcal{G}$ (with payoff matrix $A$), and $V^*_A = x^{\dagger\top} A y^\dagger$ be the value of the true game. In order to analyze the performance of the online learning algorithm, we consider the individual regret for the defender, i.e., the difference in expected cumulative payoffs relative to the value of the game:

$$\mathcal{R}_T = \sum_{t=1}^{T} \left( x_t^\top A y_t - V^*_A \right). \tag{13}$$

Building on the results from perturbation theory in zero-sum games presented in Lipton et al. (2006) we present the following result, characterizing the bound on the cumulative regret in (13).

**Theorem 6** *Under Assumption 1, for a specified $\alpha \in (0,1)$, with probability at least $1 - \alpha$, Algorithm 2 has the following guarantees for the cumulative regret (as in (13)):*

*1.* $\mathcal{R}_T \leq \dfrac{5\sqrt{2}V_{max}}{(1 - p_{max})}\sqrt{T\log\dfrac{2T}{\alpha}}; y_t = y^\dagger$ *(i.e., $y_t$ is any NE of $\mathcal{G}$);*

*2.* $\mathcal{R}_T \leq \dfrac{2\sqrt{2}V_{max}}{(1 - p_{max})}\sqrt{T\log\dfrac{2T}{\alpha}}; y_t \neq y^\dagger$ *(i.e., $y_t$ is not a NE of $\mathcal{G}$),*

*where $V_{max} = \max\limits_{i,j} V_{ij}$ and $p_{max}$ is the maximum probability of detection of the sensors.*

**Remark 7** *In Cai et al. (2024), the authors propose an online learning algorithm which is an adaptation of* EXP3-IX, *where they assume the existence of mixed-strategy Nash equilibrium with full support. Particularly, at each iteration $t$ of their algorithm, they restrict the space of NE strategies to $\Omega_t = \{x \in X : x_a \geq \frac{1}{|I|t^2}, \forall a \in I\}$. However, many zero-sum games are not guaranteed to have a mixed-strategy Nash equilibrium with full support. In our algorithm, we do not restrict the space of NE strategies for the players. Additionally, our results presented in Theorem 6 provide tighter regret guarantees $\tilde{O}(T^{-\frac{1}{2}})$ compared to $\tilde{O}(T^{-\frac{1}{8}})$ presented in Cai et al. (2024).*

We will now consider the setting with heterogeneous sensors, i.e., each sensor has a different probability of detection. In this setting, the game's payoff matrix $A$ can be parameterized by a set of $|\mathcal{S}|$ Bernoulli distributions, with parameters $\{p_{detect,1}, \ldots, p_{detect,p}\}, p = |\mathcal{S}|$, each corresponding to a sensor $s_l \in \mathcal{S}, l = 1, \ldots, p$. The game's payoff matrix in this setting is given by:

$$A_{ij} = \sum_{l=1}^{p} V_{ij,l}\log(1 - p_{detect,l}) + c_i - r_j, \tag{14}$$

where $V_{ij,l}$ is the number of nodes covered by the sensor $l$ in the joint strategy $i \in I$ and contained in the intruder's path $j \in J$, and $c_i, r_j$ are sensor and path costs as specified before, respectively. Define $\max_{ij} V_{ij,l} = V_{max,l}$. We note that each entry of $A$ may not depend on all $|\mathcal{S}|$ Bernoulli parameters, as $V_{ij,l}$ can be zero for a subset of sensors in $\mathcal{S}$. This is due to the fact that every intruder path $j \in J$ will intersect nodes covered in the graph by only a subset of the sensors. Thus, unlike the homogenous sensors setting, we will need exploration of strategies of both the players to ensure better estimates of these Bernoulli parameters.

We adapt the Upper Confidence Bound (UCB) algorithm for learning in matrix games proposed in O'Donoghue et al. (2021), by constructing the UCB estimate for the game matrix $A$ based on the confidence bounds of the Bernoulli parameters. We assume that $c_i$'s and $r_j$'s are known.

**Lemma 8** *For the game matrix $A$ defined in (14), the following holds with probability at least $1 - \delta$,*

$$A_{ij} \leq \bar{A}_{ij}^t + \sum_{l=1}^{p} \frac{V_{max,l}}{1 - p_{max,l}}\sqrt{\frac{\log(\frac{2}{\delta})}{2(1 \vee n_{ij,l}^t)}} = \tilde{A}_{ij}^t, \forall i, j, \tag{15}$$

*where $\bar{A}_{ij}^t$ is an estimate of $A_{ij}$ evaluated with sample averages $\hat{p}_{detect,l}^t$, $n_{ij,l}^t$ is the number of times the intruder chooses the path $j$ which contains nodes covered by the sensor $l$ in the joint orientation strategy $i$ up to (but not including round $t$), and we use the notation $(1 \vee \cdot) = \max(1, \cdot)$.*

Define $\bar{V}_{max} = \max_l V_{max,l}$ and $\bar{p}_{max} = \max_l p_{max,l}$. We have the following result that characterizes the bound on the regret (Equation (13)) when Algorithm 1 in O'Donoghue et al. (2021) is applied to the instance of the intrusion detection game with the UCB estimate $\tilde{A}_{ij}^t$ as in (15).

**Theorem 9** *Let Assumption 1 hold and let $T \geq |\mathcal{S}||\Theta||J| \geq 2$, $\delta = 1/(2T^2|\mathcal{S}||\Theta||J|)$, $C = \bar{V}_{max}/(1 - \bar{p}_{max})$. When Algorithm 1 in O'Donoghue et al. (2021) is applied to the instance of the game $\mathcal{G}$ with the UCB estimate $\tilde{A}_{ij}^t$ (as in (15)), the regret (as in (13)) is bounded as:*

$$\mathcal{R}_T \leq 1 + C|\mathcal{S}|\sqrt{2|\Theta||J|T\log(4T^2|\mathcal{S}||\Theta||J|)} = \tilde{O}(\sqrt{|\mathcal{S}|^2|\Theta||J|T}). \tag{16}$$

**Remark 10** *Applying Algorithm 1 of O'Donoghue et al. (2021) directly would result in a regret bound that depends exponentially on the number of sensors, i.e., $\tilde{O}(\sqrt{|\Theta|^{|\mathcal{S}|}|J|T})$. We leverage the specific structure of the game to achieve significant improvements in the regret bound in Theorem 9, i.e., $\tilde{O}(\sqrt{|\mathcal{S}|^2|\Theta||J|T})$, which will lead to faster convergence to the NE strategy for the defender. Additionally, the parameter $\delta$, which governs the balance between exploration, regret bounds, and high-probability guarantees depend on $|\Theta||\mathcal{S}||J|$ instead of $|\Theta|^{|\mathcal{S}|}|J|$. We present the adapted UCB algorithm (Algorithm 3) and additional discussions in Appendix A of Bhargav et al. (2025).*

## 5. Experiments

In this section, we evaluate the performance of the proposed algorithms through simulations.

### 5.1. Empirical Evaluation of Distributed Weighted Majority Algorithm

In this section, we present numerical evaluations to show the computational efficiency of DWM algorithm. We consider the nine-room grid world environment as shown in Figure 1(a). This is a $19 \times 19$ cell environment, which can be represented by a graph. Sensors are installed on the cells (resp. nodes in the graph) marked in red (and numbered), and all cells marked in black are obstacle cells. We set $|\Theta| = 4$, i.e., each sensor has 4 possible orientations: { East, North, West, South}. We vary the number of sensors $|\mathcal{S}| \in \{2, 3, 4, 5, 6, 7, 8\}$ to generate multiple instances of the game. The defender's strategy space, given by $4^{|\mathcal{S}|}$, scales exponentially with the number of sensors in $\mathcal{S}$. The intruder's strategy space is set to $|J| = 50$, i.e., there are 50 possible paths for the intruder (e.g., see Figure 1(b) from node $S$ (marked in yellow) to node $T$ (marked in orange) in the grid. All simulations are run on a 2.6 GHz 6-Core Intel Core i7 machine. We set $\epsilon_T = 0.001$, i.e., we wish to compute a $\epsilon$-Nash Equilibrium for the game with $\epsilon = 0.001$. For each instance of the game, we compute the number of iterations $T$ based on $|I|$ and $\epsilon_T$ using Equation (11) by setting $\tilde{L} = T$. We solve for the NE strategies using three approaches: (i) a standard bi-level Linear Program (LP) (using Gurobi Optimization Solver), (ii) WM algorithm (as in Littlestone and Warmuth (1994)) and (iii) DWM algorithm (Algorithm 1) for $T$ iterations, and record the running times in Table 1. We observe that the WM and DWM algorithms solve the game to within a 0.001 approximation of the game value much faster than the LP. We observe that as the strategy space increases, the DWM algorithm outperforms the WM algorithm, with a significant reduction in running time. Due to limitations on the problem sizes in Gurobi Solver (academic license), we cannot solve LPs for problem instances with strategy spaces more than 2000.

### 5.2. Empirical Evaluation of Online Learning Algorithm

We consider the intrusion detection game under homogenous sensors, and evaluate the performance of Algorithm 2. We set the true Bernoulli parameter to $p_{detect} = 0.8$ and generate several instances

| $|\mathcal{S}|$ | Strategy Spaces | Linear Program | Weighted Majority | Distributed Weighted Majority |
|---|---|---|---|---|
| 2 | $16 \times 50$ | 3.22 | 2.73 | 2.68 |
| 3 | $64 \times 50$ | 7.90 | 3.04 | 3.65 |
| 4 | $256 \times 50$ | 24.88 | 6.09 | 4.28 |
| 5 | $1024 \times 50$ | 75.72 | 12.15 | 6.13 |
| 6 | $4096 \times 50$ | - | 35.51 | 7.88 |
| 7 | $16,384 \times 50$ | - | 107.65 | 9.92 |
| 8 | $65,536 \times 50$ | - | 365.43 | 11.83 |

Table 1: Comparison of running times (in seconds) of Distributed Weighted Majority Algorithm (Algorithm 1) with standard Weighted Majority algorithm and Linear Program

($A$ matrices) by uniform random sampling of $V_{ij} \in [10, 20], \forall (i, j) \in I \times J$, with $I = \{1, ..., 10\}$ and $J = \{1, ..., 20\}$. We solve for the NE strategies $(x^\dagger, y^\dagger)$ using Algorithm 1 with $\epsilon_T = 10^{-6}$ and estimate the optimal game value $V_A^* = x^{\dagger \top} A y^\dagger$. We run Algorithm 2 in both settings: (i) $y_t = y^\dagger$; and (ii) $y_t$ being a random point on the probability simplex $Y$ for each $t$. We plot the cumulative regrets $\mathcal{R}_T$ (for $y \neq y^\dagger$) and $\mathcal{R}_T^\dagger$ (for $y = y^\dagger$) by setting $T = 1000$ in Figure 1(c). We observe that the regret grows sublinearly, indicating that the learning algorithm is effectively adapting and improving its performance over iterations. Additionally, we note that the regret is higher when the intruder employs the (optimal) Nash equilibrium strategy of the true game.
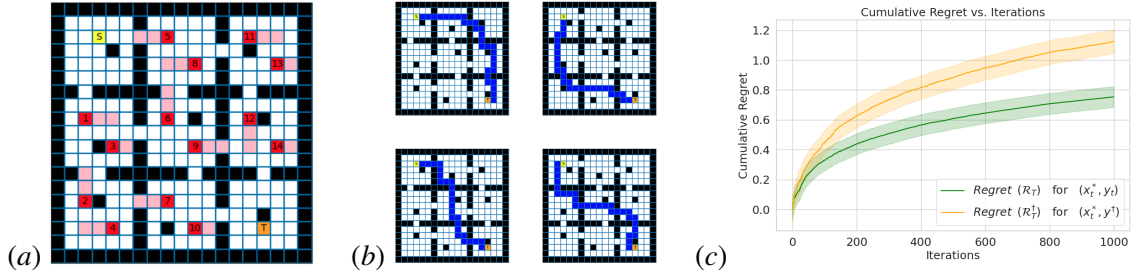


Figure 1: (a) Sample sensor scheduling strategy (pure strategy) of the defender ; (b) Sample paths (pure strategies) of the intruder in the 9-room grid world; (c) Cumulative regret plots for Alg. 2

## 6. Conclusions

In this paper, we address the problem of sensor scheduling for intrusion detection by formulating it as a zero-sum game over a graph. We proposed a distributed algorithm, which leverages the structure of the game's payoff, for efficiently computing equilibrium strategies for instances with exponentially large strategy spaces. We further extend our analysis to scenarios with unknown sensor models, introducing online learning algorithms with high-probability regret guarantees for learning equilibrium strategies under bandit feedback. Our work leverages the structure in the combinatorial action space of the players to achieve significant computational efficiency and scalability. Empirical evaluations demonstrate the practical effectiveness of our approach, showing near-optimal and scalable performance over several instances of the game. In future work, we aim to extend the framework to dynamic game environments, where strategies can adapt to changes in sensor performance, including degradation or reconfiguration, as well as stochastic variations in game payoffs.

## Acknowledgments

## References

Bo An, Milind Tambe, and Arunesh Sinha. Stackelberg security games (ssg) basics and application overview. *Improving Homeland Security Decisions*, 2:485, 2017.

Gabriel P Andrade, Rafael Frongillo, and Georgios Piliouras. Learning in matrix games can be arbitrarily complex. In *Conference on Learning Theory*, pages 159–185. PMLR, 2021.

James Bailey and Georgios Piliouras. Fast and furious learning in zero-sum games: Vanishing regret with non-vanishing step sizes. *Advances in Neural Information Processing Systems*, 32, 2019.

Jayanth Bhargav, Mahsa Ghasemi, and Shreyas Sundaram. On the Complexity and Approximability of Optimal Sensor Selection for Mixed-Observable Markov Decision Processes. In *2023 American Control Conference (ACC)*, pages 3332–3337. IEEE, 2023.

Jayanth Bhargav, Shreyas Sundaram, and Mahsa Ghasemi. Sensor Scheduling in Intrusion Detection Games with Uncertain Payoffs. *arXiv preprint arXiv:2504.14725*, 2025.

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *International conference on machine learning*, pages 793–802. PMLR, 2019.

Yang Cai, Haipeng Luo, Chen-Yu Wei, and Weiqiang Zheng. Uncoupled and convergent learning in two-player zero-sum markov games with bandit feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Zaiwei Chen, Kaiqing Zhang, Eric Mazumdar, Asuman Ozdaglar, and Adam Wierman. Last-iterate convergence of payoff-based independent learning in zero-sum stochastic games. *arXiv preprint arXiv:2409.01447*, 2024.

Yun Kuen Cheung and Georgios Piliouras. Vortices instead of equilibria in minmax optimization: Chaos and butterfly effects of online learning in zero-sum games. In *Conference on Learning Theory*, pages 807–834. PMLR, 2019.

Santpal Singh Dhillon and Krishnendu Chakrabarty. Sensor placement for effective coverage and surveillance in distributed sensor networks. In *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, volume 3, pages 1609–1614. IEEE, 2003.

Tanner Fiez, Ryann Sim, Stratis Skoulakis, Georgios Piliouras, and Lillian Ratliff. Online learning in periodic zero-sum games. *Advances in Neural Information Processing Systems*, 34:10313–10325, 2021.

Yoav Freund and Robert E Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the ninth annual conference on Computational learning theory*, pages 325–332, 1996.

Sam Ganzfried and Tuomas Sandholm. Endgame solving in large imperfect-information games. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Mahsa Ghasemi, Evan Scope Crafts, Bo Zhao, and Ufuk Topcu. Multiple plans are better than one: Diverse stochastic planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 31, pages 140–148, 2021.

M Amac Guvensan and A Gokhan Yavuz. On coverage issues in directional sensor networks: A survey. *Ad Hoc Networks*, 9(7):1238–1255, 2011.

Damien B Jourdan and Nicholas Roy. Optimal sensor placement for agent localization. *ACM Transactions on Sensor Networks (TOSN)*, 4(3):1–40, 2008.

Ravi Kannan and Thorsten Theobald. Games of fixed rank: A hierarchy of bimatrix games. *Economic Theory*, 42:157–173, 2010.

Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. Complexity of computing optimal stackelberg strategies in security resource allocation games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 805–810, 2010.

Dmytro Korzhyk, Zhengyu Yin, Christopher Kiekintveld, Vincent Conitzer, and Milind Tambe. Stackelberg vs. nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *Journal of Artificial Intelligence Research*, 41:297–327, 2011.

Shuxin Li, Youzhi Zhang, Xinrun Wang, Wanqi Xue, and Bo An. Decision making in team-adversary games with combinatorial action space. *CAAI Artificial Intelligence Research*, 2, 2023.

Zun Li and Michael Wellman. Structure learning for approximate solution of many-player games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2119–2127, 2020.

Richard J Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, pages 36–41, 2003.

Richard J Lipton, E Markakis, and Aranyak Mehta. On stability properties of economic solution concepts. *Manuscript*, 2006.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

Alan T Murray, Kamyoung Kim, James W Davis, Raghu Machiraju, and Richard Parent. Coverage optimization to support security monitoring. *Computers, Environment and Urban Systems*, 31 (2):133–147, 2007.

Michael Victor Nehme. *Two-person games for stochastic network interdiction: models, methods, and complexities*. The University of Texas at Austin, 2009.

Yahya Osais, Marc St-Hilaire, and F Richard Yu. On sensor placement for directional wireless sensor networks. In *2009 IEEE International Conference on Communications*, pages 1–5. IEEE.

Brendan O'Donoghue, Tor Lattimore, and Ian Osband. Matrix games with bandit feedback. In *Uncertainty in Artificial Intelligence*, pages 279–289. PMLR, 2021.

Praveen Paruchuri, Jonathan P Pearce, Janusz Marecki, Milind Tambe, Fernando Ordonez, and Sarit Kraus. Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pages 895–902, 2008.

Mohammad Pirani, Ehsan Nekouei, Henrik Sandberg, and Karl Henrik Johansson. A game-theoretic framework for security-aware sensor placement problem in networked control systems. *IEEE Transactions on Automatic Control*, 67(7):3699–3706, 2021.

Tuomas Sandholm. Abstraction for solving large incomplete-information games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Arunesh Sinha, Fei Fang, Bo An, Christopher Kiekintveld, and Milind Tambe. Stackelberg security games: Looking beyond a decade of success. IJCAI, 2018.