Optimizing Sensor Allocation Against Attackers With Uncertain Intentions: A Worst-Case Regret Minimization Approach

Haoxiang Ma, *Graduate Student Member, IEEE*, Shuo Han, *Member, IEEE*, Charles A. Kamhoua, *Senior Member, IEEE*, and Jie Fu, *Member, IEEE*

***Abstract*—This letter focuses on the optimal allocation of multi-stage attacks with the uncertainty in attacker's intention. We model the attack planning problem using a Markov decision process and characterize the uncertainty in the attacker's intention using a finite set of reward functions—each reward represents a type of attacker. Based on this modeling, we employ the paradigm of the worst-case absolute regret minimization from robust game theory and develop mixed-integer linear program (MILP) formulations for solving the worst-case regret minimizing sensor allocation strategies for two classes of attack-defend interactions: one where the defender and attacker engage in a zero-sum game and another where they engage in a non-zero-sum game. We demonstrate the effectiveness of our algorithm using a stochastic gridworld example.**

***Index Terms*—Markov process, game theory, optimization.**

I. Introduction

With the increasing severity of cyber- and physical-attacks, developing effective proactive defense aims to enable early detection of attacks by strategically allocating sensors/intrusion detectors. However, this task is complicated by the fact that attackers often have varying objectives and intentions. This letter studies the design of a robust proactive sensor allocation, given the uncertainty in the objective or the intention of the attacker. Our approach is motivated by real-world cyber security incidents, where defenders have limited monitoring resources and must deal with attackers with different objectives, ranging from using a botnet to interrupt services with a DDoS attack, distributing malware to steal sensitive data, or privilege escalation attacks (see [18], a report on recent cyber security incidents).

We formulate the attack planning problem as a Markov decision process (MDP) and enable the defender to allocate intrusion detectors, called sensors in this context, to detect the presence of an attack. The sensor allocation modifies the transition function of the attack MDP. Specifically, when a state is allocated with a sensor, it becomes a sink/absorbing state, as the attack terminates once a sensor state is reached. Therefore, the goal of designing an optimal sensor allocation is to modify the transition function of the attack MDP such that the attacker's value can be minimized given the best response attack strategy. The sensor allocation problem in attack graphs [4] is closely related to Stochastic Stackelberg Game (SSG) [16]. In an SSG, the defender/leader commits to a strategy first to protect a set of targets with limited resources, while the attacker/follower selects the best response attack strategy to the defender's strategy. Related to SSG for sensor allocation, Li et al. [6] developed a mixed-integer linear program (MILP) formulation for solving joint allocation of detectors and stealthy sensors that minimizes the attacker's probability of success. Sengupta et al. [14] modeled the attacker-defender interaction using a normal-form game and proposed a mixed strategy for the defender to randomize intrusion detectors. Besides the security game, other resource allocation problems have been extensively studied. These include distributing preventive resources to contain the spreading process [9] in a network; allocating sensors for maximizing coverage [7]. The main difference is that in game-theoretic resource allocation, the decision maker's objective function is a function of the allocated resource and the best response of the attacker, which can be influenced by how the sensors are allocated.

Traditional Stackelberg security games assume that the defender knows the attacker's payoff function, which is often not the case. To address this, researchers have studied robust defense design from the perspective of robust optimization [1]. In [10], the authors considered robust Stackelberg equilibria in normal form games where the uncertainty comes from multiple ϵ-optimal best responses from the follower. In [17], the authors used an MDP to model the attack planning problem and employed robust optimization to design a moving target

Manuscript received 17 March 2023; revised 22 May 2023; accepted 10 June 2023. Date of publication 28 June 2023; date of current version 14 July 2023. This work was supported in part by the Army Research Office under Grant W911NF-22-1-0034 and Grant W911NF2210166, and in part by NSF under Grant 2144113. Recommended by Senior Editor S. Dey. *(Corresponding author: Haoxiang Ma.)*

Haoxiang Ma and Jie Fu are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: hma2@ufl.edu; fujie@ufl.edu).

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

Charles A. Kamhoua is with the U.S. Army Combat Capabilities Development Command (DEVCOM), U.S. Army Research Laboratory, Adelphi, MD 20783 USA (e-mail: charles.a.kamhoua.civ@mail.mil).

Digital Object Identifier 10.1109/LCSYS.2023.3290489

defense (MTD) policy that is robust to a finite uncertainty set of attack strategies. In [5], the authors introduced a robust Stackelberg equilibrium that maximizes the leader's payoff given the worst-case realization of the follower's payoff in a deterministic sequential game in which each player selects a distribution over action sequences. The uncertainty is assumed to lie within a bounded interval on the follower's payoff.

Similar to [5], we also investigate the problem of robust defense when the defender has incomplete knowledge about the attacker's payoffs, modeled as a finite set of attacker's types. Each type is associated with a unique reward function that describes the attack objective. We consider robust sensor allocation with the solution of the worst-case absolute regret minimization [11].

The proposed Worst-Case Absolute Regret Minimization for Sensor Allocation (WCARM-SA) solution informs a regret-averse defender to choose a strategy that leads to a small regret once he realizes what would have been the best decision if he knew the attacker's type. As shown in operations research, the WCARM-SA solutions are often less conservative solutions compared to those by robust optimization [11], [13].

Our contribution can be summarized as:

- We develop WCARM-SA methods to solve robust sensor allocation problems in zero-sum and non-zero-sum attack-defend interactions with uncertainty in the attack intention, described by a finite set of possible attacker's reward functions. We demonstrate that the WCARM-SA can be formulated as MILP problems for both cases.
- We leverage the zero-sum property to develop efficient solution for the WCARM-SA for the zero-sum case.
- We validate the effectiveness of our proposed approach through experiments with attack motion planning problems in stochastic gridworld environments.

II. Preliminaries and Problem Formulation

Notations: Let $\mathbb{R}$ denote the set of real numbers and $\mathbb{R}^n$ the set of real n-vectors. The vector of all ones is represented as $\mathbf{1}$. The notation z_i refers to the i-th component of a vector $z \in \mathbb{R}^n$ or to the i-th element of a sequence $z_1, z_2, \ldots$, which will be clarified by the context. The set of probability distributions over a finite set Z is denoted as $\mathsf{Dist}(Z)$.

We begin by presenting an attack MDP that captures an attacker's planning problem.

Attack Planning Problem: The attack planning problem is modeled as an attack MDP $M = (S, A, P, \nu, \gamma, R)$, where S is a set of states (nodes in the attack graph) including a special absorbing/sink state s_{sink}, A is a set of attack actions, $P : S \times A \to \mathsf{Dist}(S)$ is a probabilistic transition function such that $P(s'|s, a)$ is the probability of reaching state s' given action a being taken at state s, $\nu \in \mathsf{Dist}(S)$ is the initial state distribution, $\gamma \in (0, 1]$ is a discount factor, and $R : S \times A \to \mathbb{R}$ is the attacker's reward function such that $R(s, a)$ is the reward received by the attacker for taking action a in state s. The attacker's objective is to maximize the total discounted rewards in the attack MDP. For concrete examples of attack graphs generated from network vulnerabilities, readers are directed to [4] and [6].

We consider Markovian policies because it suffices to search in Markovian policies for an optimal policy in the attack MDP [12]. Given a Markovian policy $\pi : S \to \mathsf{Dist}(A)$, the attacker's value function $V_2^\pi : S \to \mathbb{R}$ is defined as

$$V_2^\pi(s) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R(s_k, \pi(s_k)) | s_0 = s\right],$$

where E_π is the expectation and s_k is the k-th state in the Markov chain induced from the MDP M under the policy π, starting from state s. The attacker's value given the initial distribution μ is $V_2^\pi(\mu) = \sum_{s \in S} \mu(s) V_2^\pi(s)$.

Defender's incomplete information: The defender knows the dynamics in the attack MDP. However, the defender does not know the exact reward function of the attacker; rather, the defender is only aware that the attacker can fall into one attack type at any given time. Different attacker types only differ in their reward function in the attack MDP and share the same states, actions, transition function, initial distribution, and discount factor. Specifically, let $\mathcal{T} = \{1, \ldots, N\}$ be the attacker's type space. Let $R_i : S \times A \to \mathbb{R}$ be the reward function for attacker type i.

Defender's countermeasures: To detect an ongoing attack, the defender is capable of allocating sensors to a subset $U \subset S$ of states in the MDP M. The attack will be terminated immediately once the attacker reaches a state monitored by the sensor (assuming the sensor's false negative rate is 0).[1] However, the defender's sensor allocation is constrained. Specifically, we consider a sensor allocation as a Boolean vector $\vec{x} \in \{0, 1\}^{|S|}$. If $\vec{x}(s) = 1$, then the state $s \in U$ is allocated with a sensor. A valid allocation $\vec{x}$ needs to satisfy $\vec{x}(s) = 0$ for any $s \in S \setminus U$ because only states in U can be monitored. In addition, the number of sensors cannot exceed a given integer k, i.e., $\mathbf{1}^\mathsf{T}\vec{x} \leq k$.

We state the problem informally as follows.

Problem 1: In the attack planning modeled as the MDP M with uncertainty in the attacker's type, how to robustly allocate limited sensors with respect to the defense objective?

III. Main Results

First, it is observed that a sensor allocation changes the transition function of the attack MDP as follows.

Definition 1 (Attack MDP Equipped With Sensors): Given a sensor allocation $\vec{x}$ and the original attack MDP $M = (S, A, P, \nu, \gamma, R)$, the attack MDP under $\vec{x}$ is the MDP

$$M(\vec{x}) = \left(S, A, P^{\vec{x}}, \nu, \gamma, R\right),$$

where S, A, ν, γ, R are identical to those in M, and $P^{\vec{x}}$ is defined as

$$P^{\vec{x}}(s'|s, a) = \begin{cases} 1, & \vec{x}(s) = 1,\ s' = s_{\mathsf{sink}}, \\ 0, & \vec{x}(s) = 1,\ s' \neq s_{\mathsf{sink}}, \\ P(s'|s, a), & \vec{x}(s) = 0. \end{cases}$$

To allocate sensors with uncertainty in the attacker's type, we employ a solution of robust game [11], [15], called *worst-case absolute regret minimization*, which optimizes the performance of a decision variable, $\vec{x}$ in our context, with respect to the "worst-case regret" that might be experienced when comparing $\vec{x}$ to the best decision that should have been made given the attacker's type i is known. Next, we discuss

[1] Please see the long version (https://arxiv.org/abs/2304.05962) for how to deal with sensors with nonzero false negative rates.

two approaches to solve worst-case regret minimizing sensor allocations given i) *Zero-sum attack-defend game:* For each attack type i, the defender's value in $M(\vec{x})$ is the negation of the attacker i's value. ii) *Non-zero-sum attack-defend game:* Regardless of the attacker's type, the defender's value in $M(\vec{x})$ is defined by evaluating the attacker's strategy with respect to a defender's cost function $C : S \times A \to \mathbb{R}$. The defender's goal is to minimize the total discounted costs respecting C incurred by the attacker's strategy.

A. Worst-Case Regret Minimization in Zero-Sum Game

In the zero-sum case, it is first noted that the optimal sensor allocation $\vec{x}_i$ for attacker's type i can be obtained by solving the following optimization problem:

$$\vec{x}_i = \arg\min_{\vec{x}\in\mathcal{X}} \max_{\pi} V^{\pi}_{2,i}(\nu; \vec{x}).$$

where $V^{\pi}_{2,i}(\nu; \vec{x})$ is attacker i's value given attack strategy π in the MDP $M(\vec{x})$ and the attacker i's reward R_i. The optimal sensor allocation problem can be formulated as an MILP (see [6]). As a result, the WCARM-SA problem is formulated as:

$$\text{(WCARM-SA)}\ \text{minimize}_{\vec{x}\in\mathcal{X}} \max_{i\in\mathcal{T}} \big(V_{2,i}(\vec{x}) - V_{2,i}(\vec{x}_i)\big)$$

where $V_{2,i}(\vec{z}) = \max_{\pi} V^{\pi}_{2,i}(\nu, \vec{z})$ for $z = \vec{x}, \vec{x}_i$. That is, the attacker always chooses the optimal strategy in MDP $M(\vec{x})$. The difference $V_{2,i}(\vec{x}) - V_{2,i}(\vec{x}_i)$ measures the regret of the defender for choosing $\vec{x}$ instead of $\vec{x}_i$ when the attacker is type i. The regret is always non-negative for any sensor allocation decision $\vec{x} \in \mathcal{X}$ because $\vec{x}_i = \arg\min_{\vec{x}} V_{2,i}(\vec{x})$.

Because $\vec{x}_i$ is pre-computed for each attacker type, the quantity $V_{2,i}(\vec{x}_i)$ is a constant, denoted by v_i for clarity. The optimization problem is then written as:

$$\text{minimize}_{\vec{x}\in\mathcal{X}} \max_{i\in\mathcal{T}} \big(V_{2,i}(\vec{x}) - v_i\big),$$

which is a robust optimization problem. The following lemma shows how the robust optimization problem can be reformulated as an MILP.

Lemma 1: The worst-case absolute regret minimization problem for robust sensor allocation in a zero-sum game is equivalent to the following optimization problem:

$$\min_{y,\vec{x}\in\mathcal{X}} \quad y \tag{1}$$

$$\text{s.t.}\quad y \geq V_{2,i}(\vec{x}) - v_i, \forall i \in \mathcal{T}, \tag{2}$$

$$V_{2,i}(\vec{x}) = \sum_{s\in S} \nu(s) V_{2,i}(s; \vec{x}), \forall i \in \mathcal{T}, \tag{3}$$

$$V_{2,i}(s; \vec{x}) \geq R_i(s, a) + \gamma \sum_{s'} P^{\vec{x}}\big(s'|s, a\big) V_{2,i}\big(s'; \vec{x}\big),$$

$$\forall s \in S, \forall a \in A, \forall i \in \mathcal{T}, \tag{4}$$

$$\mathbf{1}^{\mathsf{T}}\vec{x} \leq k. \tag{5}$$

Proof: For attacker type i and a sensor design $\vec{x}$, the optimal attacker's value vector, denoted $V^{*}_{2,i}(\vec{x}) \in \mathbb{R}^{|S|}$ satisfies the Bellman optimality condition: For all $s \in S$,

$$V^{*}_{2,i}(s, \vec{x}) = \max_{a\in A}\left(R_i(s, a) + \gamma \sum_{s'\in S} P^{\vec{x}}(s'|s, a) V^{*}_{2,i}(s'; \vec{x})\right).$$

Based on the linear program formulation of dynamic programming [2], any vector $V_{2,i}(\vec{x})$ satisfying the set of constraints in (4) is an upper bound on the $V^{*}_{2,i}(\vec{x})$, for all $s \in S$ element-wise. Therefore, $\sum_{s\in S} \nu(s) V_{2,i}(s; \vec{x}) \geq \sum_{s\in S} \nu(s) V^{*}_{2,i}(s; \vec{x})$. Constraints (2), (3), and (4) together enforce $y \geq \max_{i\in\mathcal{T}}(\sum_{s\in S} \nu(s)\ V_{2,i}(s; \vec{x}) - v_i) \geq \max_{i\in\mathcal{T}}(V^{*}_{2,i}(\nu; \vec{x}) - v_i)$.

For an arbitrary $\vec{x}$, let $r(\vec{x}) = \text{argmax}_{i\in\mathcal{T}}(V^{*}_{2,i}(\nu; \vec{x}) - v_i)$, that is, the $r(\vec{x})$ is the attacker type for which the defender's regret of using $\vec{x}$ is the largest among the regret for all attacker's types. Then we have $y \geq \max_{i\in\mathcal{T}}(V^{*}_{2,i}(\nu; \vec{x}) - v_i) = V^{*}_{r(\vec{x})}(\nu; \vec{x}) - v_{r(\vec{x})}$. Because $v_{r(\vec{x})}$ is a constant once $r(\vec{x})$ is determined, minimizing y is equivalent to minimizing the upper bound of $V_{r(\vec{x})}(\nu; \vec{x})$ and thus $y = V^{*}_{r(\vec{x})}(\nu; \vec{x}) - v_{r(\vec{x})}$. The optimization problem is then $\min_{\vec{x}}(V^{*}_{r(\vec{x})}(\nu; \vec{x}) - v_{r(\vec{x})})$, which is equivalent to the WCARM-SA formulation. ■

It is noted that the optimization problem in (1) is nonlinear because the transition function $P^{\vec{x}}$ depends on the decision variable $\vec{x}$ and the constraints in (4) include product terms between the transition probabilities with the other variable $V_{2,i}(s, \vec{x})$. We show how to transform the nonlinear program into an MILP next.

By the definition of $P^{\vec{x}}$ in (2), the term $\sum_{s'} P^{\vec{x}}(s'|s, a) V_{2,i}(s'; \vec{x})$ in (5) satisfies

$$\begin{aligned}
&\sum_{s'} P^{\vec{x}}(s'|s, a) V_{2,i}(s'; \vec{x}) \\
&\quad= \begin{cases} V_{2,i}(s_{\text{sink}}; \vec{x}), & \vec{x}(s) = 1, \\ \sum_{s'} P(s'|s, a) V_{2,i}(s'; \vec{x}), & \vec{x}(s) = 0 \end{cases} \\
&\quad= \sum_{s'} P(s'|s, a) V_{2,i}(s'; \vec{x})(1 - \vec{x}(s)) + V_2(s_{\text{sink}}; \vec{x})\vec{x}(s) \\
&\quad= \sum_{s'} P(s'|s, a) V_{2,i}(s'; \vec{x})(1 - \vec{x}(s)),
\end{aligned}$$

where the last equality is implied by $V_2(s_{\text{sink}}; \vec{x}) = 0$. Define

$$W_{2,i}(s, s') = V_{2,i}(s'; \vec{x})(1 - \vec{x}(s)) = \begin{cases} V_{2,i}(s'; \vec{x}), & \vec{x}(s) = 0, \\ 0, & \vec{x}(s) = 1. \end{cases} \tag{6}$$

Using the big-M method [3], Eq. (6) can be expressed equivalently as affine inequalities (in $\vec{x}$, $V_{2,i}$, and $W_{2,i}$)

$$W_{2,i}(s, s') \leq M(1 - \vec{x}(s)), \tag{7}$$

$$W_{2,i}(s, s') \geq m(1 - \vec{x}(s)), \tag{8}$$

$$W_{2,i}(s, s') - V_{2,i}(s'; \vec{x}) \leq M\vec{x}(s), \tag{9}$$

$$W_{2,i}(s, s') - V_{2,i}(s'; \vec{x}) \geq m\vec{x}(s). \tag{10}$$

with proper choices of constants $M > 0$ and $m < 0$. For example, let M be the upper bound on the total rewards and m be the negation of the upper bound on the total rewards.

B. Worst-Case Regret Minimization in Non-Zero-Sum Game

Next, we consider the scenario where the defender aims to minimize her cost function $C : S \times A \to \mathbb{R}$, which maps a state and an attack action to a cost penalty that measures the loss incurred by the attacker's action. Because the penalty is not necessarily the negation of the attack reward, the attack-defend game is non-zero-sum. For a single type of attacker, we formulate the problem as a Stackelberg game as follows.

Definition 2: For a single type of attacker whose attack MDP is $M = (S, A, P, \mu, \gamma, R)$, and the defender's capability of allocating sensors, an SSG is formulated as a tuple $G = (S, A_1, A_2, \mathcal{P}, \nu, \gamma, R_1, R_2)$, where

- S, ν, γ are the same components in the attack MDP M for states, initial distribution, and discount factor.
- $A_1 = \{0, 1\}$ is the defender/leader's action set. Action 0 for not allocating a sensor, 1 for allocating a sensor.
- $A_2 = A$ is the attacker/follower's actions.
- $\mathcal{P}(s'|s, a_1, a_2)$ is the probability of reaching state s' given action a_1, a_2 being taken by the defender and the attacker at state s. For a state $s \in S$, a defender's action $a_1 \in A_1$ and an attacker's action $a_2 \in A_2$, let

$$\mathcal{P}(s'|s, a_1, a_2) = \begin{cases} 1, & a_1 = 1,\ s' = s_{\text{sink}}, \\ 0, & a_1 = 1,\ s' \neq s_{\text{sink}}, \\ P(s'|s, a_2), & a_1 = 0. \end{cases}$$

- $R_1 : S \times A_1 \times A_2 \to \mathbb{R}$ (resp. $R_2 : S \times A_1 \times A_2 \to \mathbb{R}$) is the leader(resp. follower)'s reward function, defined with the defender's cost C (resp. the attacker's reward R):

$$R_1(s, a_1, a_2) = \begin{cases} 0, & a_1 = 1, \\ -C(s, a_2), & a_1 = 0, \end{cases}$$

$$R_2(s, a_1, a_2) = \begin{cases} 0, & a_1 = 1, \\ R(s, a_2), & a_1 = 0. \end{cases}$$

In this SSG, the defender/leader decides the sensor allocation, which determines the transition function $\mathcal{P}$. The attacker/follower decides on the best response to maximize his reward. The reward function is understood as follows: When the defender allocates a sensor to state s (i.e., $a_1 = 1$), the attack terminates in that state, and no further costs/rewards will be incurred for either player. Otherwise (i.e., $a_1 = 0$), the attacker continues to reach the next state with action a_2, and both the defender and the attacker receive a reward of $-C(s, a_2)$ and $R(s, a_2)$, respectively. In this formulated SSG, both the defender and the attacker aim to maximize their respective total discounted rewards.

Since sensors cannot be moved once allocated, we restrict the defender's strategy to be deterministic and memoryless. Given a fixed sensor allocation, the best response attack strategy can also be deterministic. For a strategy profile (π_1, π_2)–a tuple of the defender's strategy $\pi_1 : S \to A_1$ and $\pi_2 : S \to A_2$, the defender's value function V_1 is defined as

$$V_1(s; \pi_1, \pi_2) = E_{(\pi_1, \pi_2)}\left[\sum_{k=0}^{\infty} \gamma^k R_1(s_k, \pi_1(s_k), \pi_2(s_k))|s_0 = s\right],$$

where the expectation is taken in the Markov chain induced from G given the strategy profile (π_1, π_2).

For the non-zero-sum case, the WCARM-SA problem takes the following form:

$$\text{(WCARM-SA)}\text{minimize}_{\pi_1} \max_{i \in \mathcal{T}} \big(V_{1,i}(\nu, \pi_1^i) - V_{1,i}(\nu, \pi_1)\big),$$

where $V_{1,i}(\nu, \pi_1^i)$ is the defender's value given both the defender and the attacker committing to the Stackelberg equilibrium in the SSG G (Def. 2) where the attacker's reward R_2 is defined based on the reward function of attack type i. The Stackelberg equilibrium can be solved with methods in [19], with a modification that constrains the defender's strategy to be deterministic. The regret $V_{1,i}(\nu, \pi_1^i) - V_{1,i}(\nu, \pi_1)$ measures the defender's regret in using strategy π_1 against attacker i to the defender's best strategy π_1^i that should have been employed when playing against attacker i.

To find π_1^* that minimizes the worst-case regret, we introduce a decision variable y and rewrite the optimization problem as follows:

$$\min_{\pi_1} \quad y \tag{11}$$

$$\text{s.t.} \quad y \geq V_{1,i}(\nu, \pi_1^i) - V_{1,i}(\nu, \pi_1), \forall i \in \mathcal{T}, \tag{12}$$

where $V_{1,i}(\nu, \pi_1^i)$ is a constant and is denoted by $\bar{v}_{1,i}$.

We extend the MILP formulation in [19] and get the following mixed-integer nonlinear program (MINLP) formulation to solve the WCARM-SA problem:

$$\min_{\pi_1, \{V_{1,i}, \pi_2^i, V_{2,i} | i \in \mathcal{T}\}} y \tag{13a}$$

subject to:

$$y \geq \bar{v}_{1,i} - \sum_{s \in S} \nu(s) V_{1,i}(s), \quad \forall i \in \mathcal{T}, \tag{13b}$$

$$\pi_1(a_1 \mid s) \in \{0, 1\}, \ \forall s \in S, a_1 \in \{0, 1\}, \tag{13c}$$

$$\sum_{a_1 \in \{0,1\}} \pi_1(a_1 \mid s) = 1 \ \ \forall s \in S, \tag{13d}$$

$$\pi_2^i(a_2 \mid s) \in \{0, 1\}, \quad \forall i \in \mathcal{T}, s \in S, a_2 \in A, \tag{13e}$$

$$\sum_{a_2 \in A} \pi_2^i(a_2 \mid s) = 1, \quad \forall i \in \mathcal{T}, s \in S, \tag{13f}$$

$$\sum_{s \in S} \pi_1(1|s) \leq k, \tag{13g}$$

The following constraints hold $\forall\ i \in \mathcal{T}, s \in S, a_2 \in A_2$:

$$\big(\pi_2^i(a_2 \mid s) - 1\big)Z \leq V_{1,i}(s) - \tilde{R}_1^i(s, \pi_1, a_2) \leq \big(1 - \pi_2^i(a_2 \mid s)\big)Z, \tag{13h}$$

$$0 \leq V_{2,i}(s) - \tilde{R}_2^i(s, \pi_1, a_2) \leq \big(1 - \pi_2^i(a_2 \mid s)\big)Z, \tag{13i}$$

where $\tilde{R}_1^i(s, \pi_1, a_2) = \sum_{a_1 \in \{0,1\}} \pi_1(a_1|s)(R_1(s, a_1, a_2) + \gamma \sum_{s'} \mathcal{P}(s'|s, a_1, a_2) V_{1,i}(s'))$ represents the defender's expected value from state s given the sensor allocation π_1 and attacker's action a_2 from state s, and the function $\tilde{R}_2^i(s, \pi_1, a_2)$ is defined for the attacker analogously by substituting the defender's reward and value of the next state with the attacker's. Z is a large constant number, which can be the upper bound on the absolute value of total rewards.

The constraints in (13) are explained as follows: Constraint (13b) enforces y to be the worst-case regret. Constraint (13c) and (13d) enforce the defender takes a deterministic strategy, constraint (13e), (13f) enforce attacker takes a deterministic strategy. Constraint (13g) enforces the defender can not allocate more than k sensors. Constraint (13h) enforces that when the attacker i takes action a_2, the defender's value at that state $V_{1,i}(s)$ should equal the expected value $\tilde{R}_1^i(s, \pi_1, a_2)$ of defender against attacker i who takes action a_2 at state s. When the attacker does not take action a_2, the constraint is non-binding. This set of constraints obtain $V_{1,i}(s)$ by evaluating strategy π_1 at the state s against the attacker's best response for π_1. Constraint (13i) enforces that when the attacker i takes action a_2, his value $V_{2,i}(s)$ should be the same as his expected value $\tilde{R}_2^i(s, \pi_1, a_2)$ given that action a_2. When a_2 is not taken at s, the attacker's value $V_{2,i}(s)$ should be greater than the expected value $\tilde{R}_2^i(s, \pi_1, a_2)$ due to the fact that

$V_{2,i}(s) = \max_{a_2} \tilde{R}_2^i(s, \pi_1, a_2)$. By enforcing this constraint, we can ensure $V_{2,i}$ is the attacker's value given the defender's strategy π_1 and the attacker's best response to π_1.

Lemma 2: The worst-case absolute regret minimization solution for robust sensor allocation in the non-zero-sum attack-defender game is equivalent to the solution of (13).

The proof is similar to that of Lemma 1 and can be found in the long version.[2] The key insight is that constraints (13b), (13i) and(13h) together enforce $y \geq \max_{i\in\mathcal{T}}(\bar{v}_{1,i} - \sum_{s\in S} \nu(s) V_{1,i}(s))$ where $V_{1,i}(\cdot)$ is the defender's value given the best response of attacker type i (enforced by (13h)).

The above formulation is nonlinear due to the interaction between the integer variable π_1 and the continuous variable $V_{1,i}$ in $\tilde{R}_1^i$ ($\tilde{R}_2^i$ analogously). But since the integer variable is binary, we can use McCormick Relaxation [8] to reformulate MINLP into MILP. To do so, let's introduce new variables for the defender: for $i \in \mathcal{T}, s \in S, a_1 \in \{0, 1\}, a_2 \in A_2$, define $w_{s,i}^{a_1,a_2} = \pi_1(a_1|s) \sum_{s'} \mathcal{P}(s'|s, a_1, a_2) V_{1,i}(s')$. Analogously let $z_{s,i}^{a_1,a_2}$ be defined for the attacker.

We then replace $\tilde{R}_1^i(s, \pi_1, a_2)$ in (13h) with $\sum_{a_1\in\{0,1\}}(\pi_1(a_1|s) R_1(s, a_1, a_2) + \gamma w_{s,i}^{a_1,a_2})$ and add the following constraints: $\forall i \in \mathcal{T}, s \in S, a_1 \in \{0, 1\}, a_2 \in A$:

$$w_{s,i}^{a_1,a_2} \geq \sum_{s'\in S} \mathcal{P}(s'|s, a_1, a_2) V_{1,i}(s') - Z(1 - \pi_1(a_1|s)),$$

$$w_{s,i}^{a_1,a_2} \leq \sum_{s'\in S} \mathcal{P}(s'|s, a_1, a_2) V_{1,i}(s') + Z(1 - \pi_1(a_1|s)),$$

$$-Z \cdot \pi_1(a_1|s) \leq w_{s,i}^{a_1,a_2} \leq Z \cdot \pi_1(a_1|s).$$

We replace $\tilde{R}_2^i(s, \pi_1, a_2)$ in (13i) and add constraints for $z_{s,i}^{a_1,a_2}$ analogously. For both zero-sum and non-zero-sum cases, the formulated MILP can be solved using the Gurobi Solver.

Remark 1: If different attackers have different transition functions P_i, then their corresponding transition functions are used in Constraints (4) for the zero-sum case and in defining $\tilde{R}_1^i(s, \pi_1, a_2)$ and $\tilde{R}_2^i(s, \pi_1, a_2)$ for the non-zero-sum case.

Complexity analysis: Solving an MILP is NP-complete and its runtime complexity depends on the number of constraints and integer variables. In the non-zero-sum case, the number of integer variables required for the WCARM-SA is $O(|S| \times |A_2| \times |\mathcal{T}|)$. The number of constraints is $O(|S| \times |A_1| \times |A_2| \times |\mathcal{T}|)$. For the zero-sum case, the number of integer variables and constraints is $O(|S|)$ and $O(|S| \times |A_2| \times |\mathcal{T}|)$ respectively. While the WCARM-SA solution for the non-zero-sum case can be applied to the zero-sum case, the zero-sum case formulation in Section III-A is more efficient.

IV. Experiments

We used an 8×8 gridworld environment, depicted in Figure 1, to demonstrate our solutions. A state is denoted by (row, col). There are two types of attackers with the same initial state distribution. Both attackers have a 70% probability of starting from state $(2, 0)$ and a 30% probability of starting from state $(6, 0)$. Each attacker can move in one of four compass directions. When given the action "N", the attacker enters the intended cell with a $1-2\alpha$ probability, and enters the neighboring cells, which are the west and east cells, with probability α. In our experiment, we set $\alpha = 0.1$. If the attacker moves into the buildings or the boundary, he remains in the previous cell. The environment contains three final states, each with a different value for the attackers. The attackers only receive a reward when they reach these final states. Table I lists the rewards for the attackers.

[2]https://arxiv.org/abs/2304.05962

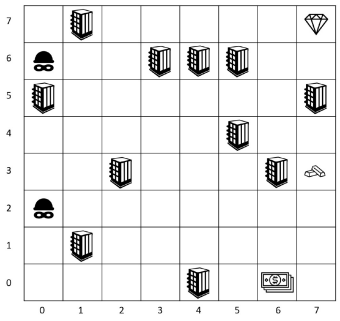

Fig. 1. The 8×8 Gridworld Example.

TABLE I
Reward Functions of Two Attackers and the Cost Function of the Defender (for the Non-Zero-Sum Case)

Reward/Cost	Cash (0, 6)	Gold (3, 7)	Diamond (7, 7)
Attacker 1	15	12	12
Attacker 2	12	15	15
Defender	15	10	10

First, we consider the worst-case regret minimization in the zero-sum case. We change the number of sensors the defender can allocate to evaluate how sensor numbers affect the defender's value. As shown in Figure 2, the attacker's expected value decreases when the sensor number increases.

In the case where only two sensors can be allocated, we compare the robust sensor allocation $\vec{x}^*$ with the optimal allocation strategies $\vec{x}_1$ and $\vec{x}_2$ against attacker types 1 and 2, respectively. When the defender knows the attacker's type, $\vec{x}_1$ yields a value of 5.96 for attacker 1 and $\vec{x}_2$ yields a value of 6.56 for attacker 2. However, when the defender does not know the attacker's type and implements $\vec{x}^*$, the values become 6.12 and 7.04 for attackers 1 and 2, respectively. In this case, the defender's worst-case absolute regret is $\max(6.12 - 5.96, 7.04 - 6.56) = 0.48$.

Using $\vec{x}_1$ against attacker 2 results in a regret of 0.89, while using $\vec{x}_2$ against attacker 1 results in a regret of 1.49. In both cases, the worst-case regret is higher than that of $\vec{x}^*$, indicating the advantage of the WCARM-SA method. The WCARM-SA sensor allocations for different numbers of sensors are listed in Table II as well as the defender's worst-case regret under the robust policy. When 4 sensors are allocated, $\vec{x}_1 = \vec{x}_2 = \vec{x}^* = (1, 5), (2, 5), (5, 5), (7, 5)$ and the defender's worst-case regret $y = 0$. An attacker is prevented from reaching any final state with probability 1.

Moving on to the non-zero-sum case, the defender now receives a penalty when the attacker reaches the goal state, and the cost function for the defender is listed in Table I. The sensor allocation strategies for the non-zero-sum cases, as well as the defender's worst-case regret under the robust policy, can be found in Table III. Noted that the worst-case

TABLE II
SENSOR ALLOCATION IN THE ZERO-SUM CASE. $\vec{x}_i$ IS THE OPTIMAL DEFENDER POLICY AGAINST ATTACKER TYPE $i \in \{1, 2\}$

k	Policy $\vec{x}_1$	Policy $\vec{x}_2$	Robust policy $\vec{x}^*$	y
1	$(1,5)$	$(2,2)$	$(2,3)$	0.22
2	$(1,5),(2,4)$	$(2,2),(6,2)$	$(1,5)(6,2)$	0.48
3	$(1,5),(2,4),(6,2)$	$(2,3)(5,2)(6,2)$	$(1,5),(2,2)(6,2)$	0.77

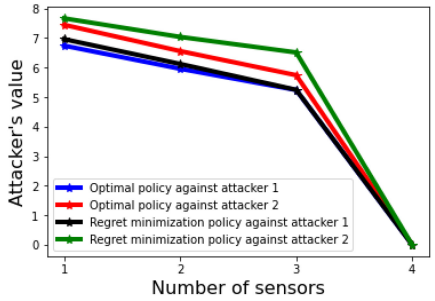

Fig. 2. The attacker's expected value for zero-sum case.

TABLE III
SENSOR ALLOCATION IN THE NON-ZERO-SUM CASE. $\vec{x}_i$ IS THE OPTIMAL DEFENDER POLICY AGAINST ATTACKER TYPE $i \in \{1, 2\}$

k	Policy π_1	Policy π_2	Robust Policy π^*	y
1	$(1,5)$	$(2,2)$	$(2,3)$	0.26
2	$(1,5),(2,4)$	$(2,3),(6,2)$	$(1,5),(7,5)$	1.19
3	$(1,4),(2,5),(6,2)$	$(2,3),(5,2),(6,2)$	$(1,5),(2,2),(6,2)$	0.20

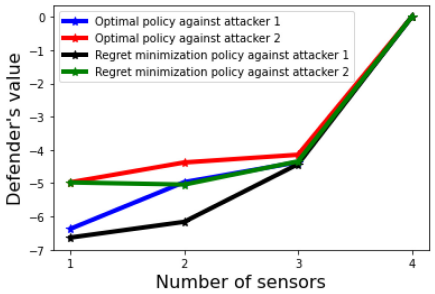

Fig. 3. The defender's expected value.

regret is not indicative of the effectiveness of sensor allocations, that is, a small regret does not necessarily mean a large value for the defender. Thus, Figure 3 demonstrates that the defender's expected value increases as the number of sensors increases. From Fig. 3, it is observed that when two sensors are deployed, the worst-case regret is the largest for both attackers.

Similar to the zero-sum case, when 4 sensors can be allocated, $\vec{x}_1 = \vec{x}_2 = \vec{x}^* = (1,5),(2,5),(5,5),(7,5)$ and the defender's worst-case regret $y = 0$.

The experiments are conducted on a Windows 10 machine with Intel i7-11700k CPU and 32 GB RAM. The computation time for robust sensor allocations in the zero-sum cases is less than 1 second. For non-zero-sum cases, it takes from 77 sec to 3 hours to solve given increasing sensor numbers.

V. CONCLUSION

We develop robust sensor allocation methods in probabilistic attack planning problems using worst-case absolute regret minimization from robust game theory. We demonstrated that both robust zero-sum and non-zero-sum sensor allocation problems can be formulated as MILPs. Our approach is suitable for a wide range of safety-critical scenarios that involve constructing probabilistic attack graphs from known network vulnerabilities. Future work could focus on developing more efficient and approximate solutions for robust sensor allocations in non-zero-sum games. Additionally, the solution concept for robust games can be extended to design moving target defenses that randomize network topologies and the integrated design of sensor allocation and moving target defense.

REFERENCES

[1] M. Aghassi and D. Bertsimas, "Robust game theory," *Math. Program.*, vol. 107, nos. 1–2, pp. 231–273, 2006.
[2] D. P. de Farias and B. Van Roy, "The linear programming approach to approximate dynamic programming," *Oper. Res.*, vol. 51, no. 6, pp. 850–865, Dec. 2003.
[3] I. Griva, S. G. Nash, and A. Sofer, *Linear Nonlinear Optimization*, vol. 108. Philadelphia, PA, USA: SIAM, 2009.
[4] S. Jha, O. Sheyner, and J. Wing, "Two formal analyses of attack graphs," in *Proc. 15th IEEE Comput. Security Found. Workshop*, 2002, pp. 49–63.
[5] C. Kroer, G. Farina, and T. Sandholm, "Robust Stackelberg equilibria in extensive-form games and extension to limited lookahead," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 1–9.
[6] L. Li, H. Ma, S. Han, and J. Fu, "Synthesis of proactive sensor placement in probabilistic attack graphs," in *Proc. Amer. Control Conf. (ACC)*, 2023, pp. 1–8.
[7] J. R. Marden, "The role of information in distributed resource allocation," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 3, pp. 654–664, Sep. 2017.
[8] A. Mitsos, B. Chachuat, and P. I. Barton, "Mccormick-based relaxations of algorithms," *SIAM J. Optim.*, vol. 20, no. 2, pp. 573–601, 2009.
[9] C. Nowzari, V. M. Preciado, and G. J. Pappas, "Optimal resource allocation for control of networked epidemic models," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 2, pp. 159–169, Jun. 2017.
[10] J. Pita, M. Jain, M. Tambe, F. Ordónez, and S. Kraus, "Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition," *Artif. Intell.*, vol. 174, no. 15, pp. 1142–1171, 2010.
[11] M. Poursoltani and E. Delage, "Adjustable robust optimization reformulations of two-stage worst-case regret minimization problems," *Oper. Res.*, vol. 70, no. 5, pp. 2906–2930, 2022.
[12] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: Wiley, 2014.
[13] L. J. Savage, "The theory of statistical decision," *J. Amer. Stat. Assoc.*, vol. 46, no. 253, pp. 55–67, 1951.
[14] S. Sengupta, A. Chowdhary, D. Huang, and S. Kambhampati, "Moving target defense for the placement of intrusion detection systems in the cloud," in *Decision and Game Theory for Security* (Lecture Notes in Computer Science 11199), L. Bushnell, R. Poovendran, and T. Başar, Eds. Cham, Switzerland: Springer Int., 2018, pp. 326–345.
[15] J. Stoye, "Statistical decisions under ambiguity," *Theory Decis.*, vol. 70, no. 2, pp. 129–148, 2011.
[16] M. Tambe, *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. New York, NY, USA: Cambridge Univ. Press, 2011.
[17] J.-L. Tan, C. Lei, H.-Q. Zhang, and Y.-Q. Cheng, "Optimal strategy selection approach to moving target defense based on Markov robust game," *Comput. Security*, vol. 85, pp. 63–76, Aug. 2019.
[18] *Data Breach Investigations Report*, Verizon, New York, NY, USA, 2022.
[19] Y. Vorobeychik and S. Singh, "Computing Stackelberg equilibria in discounted stochastic games," in *Proc. AAAI Conf. Artif. Intell.*, vol. 26, 2012, pp. 1478–1484.