Robustness in Green Security: Minimax Regret Optimality with Reinforcement Learning

Lily Xu
Harvard University
lily_xu@g.harvard.edu

Andrew Perrault
Harvard University
aperrault@seas.harvard.edu

Fei Fang
Carnegie Mellon University
feif@cs.cmu.edu

Haipeng Chen
Harvard University
hpchen@seas.harvard.edu

Milind Tambe
Harvard University
milind_tambe@harvard.edu

ABSTRACT

Green security domains feature defenders who plan patrols in the face of uncertainty about the adversarial behavior of poachers, illegal loggers, and illegal fishers. Importantly, the deterrence effect of patrols on adversaries' future behavior makes patrol planning a sequential decision-making problem. Therefore, we focus on robust sequential patrol planning for green security following the minimax regret criterion, which has not been considered in the literature. We formulate the problem as a game between the defender and nature who controls the parameter values of the adversarial behavior and design an algorithm MIRROR to find a robust policy. MIRROR uses two reinforcement learning–based oracles and solves a restricted game considering limited defender strategies and parameter values. We evaluate MIRROR on real-world poaching data.

KEYWORDS

Green security, Reinforcement learning, Minimax regret

ACM Reference Format:
Lily Xu, Andrew Perrault, Fei Fang, Haipeng Chen, and Milind Tambe. 2021. Robustness in Green Security: Minimax Regret Optimality with Reinforcement Learning. In *Proc.of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), London, UK, May 3–7, 2021*, IFAAMAS, 8 pages.

1 INTRODUCTION

Defenders in green security domains aim to protect wildlife, forests, and fisheries and are tasked to strategically allocate limited resources in a partially unknown environment [8]. For example, to prevent poaching, rangers will patrol a protected area to locate and remove snares (Figure 1). Over the past few years, predictive models of poacher behavior have been developed and deployed to parks around the world, creating both opportunity and urgency for effective patrol planning strategies [12, 15, 44].

While patrol planning for security has been studied under game-theoretic frameworks [2, 17, 24], green security domains have two crucial challenges: the *uncertainty* in adversaries' behavior model and the *deterrence* effect of patrols — how current patrols reduce the likelihood that adversaries attack in the *future*. Data is often scarce in these domains and it is hard to learn an accurate adversarial behavior model [8, 37, 43]. Patrols planned without considering the imperfection of the behavior model would have limited effectiveness in practice. Deterrence is hypothesized to be a primary mechanism that makes patrols effective in reducing illegal activity [20], especially in domains such as wildlife protection, as rangers rarely apprehend poachers and only remove an estimated 10% of snares [27]. These characteristics make apparent the need for robust sequential patrol planning for green security, which is the focus of this paper. We confirm the deterrence effect in green security domains for the first time through analyzing real poaching data, providing real-world footing for this research.

Figure 1: Rangers remove a snare in Srepok Wildlife Sanctuary in Cambodia, where the government plans to reintroduce tigers in 2022.

In this paper, we consider the *minimax regret* criterion for robustness [36, 40]: minimize the maximum regret, which is defined as the maximum difference under any uncertainty instantiation between the expected reward of the chosen strategy against the expected reward of an optimal strategy. Compared to maximin reward, minimax regret is more psychologically grounded according to phenomena such as risk aversion [23] and is less conservative and sensitive to worst-case outcomes [18]. However, optimizing for regret is challenging [30], especially for complex sequential decision making problems as evidenced by lack of past work on minimax regret in deep reinforcement learning (RL), despite the success and popularity of deep RL in recent years [22, 26]. The main obstacle is that when the environment parameters change, the reward of a strategy changes and the optimal strategy also changes, which makes it hard to quickly estimate the maximum regret of a strategy.

We overcome this obstacle by developing a new method[1] named MIRROR that enables minimax regret planning under environment uncertainty using RL. We model the robust planning problem as a two-player, zero-sum game between an agent, who looks for minimax regret–optimal policies, and nature, who looks for regret-maximizing instantiations of the uncertain environment parameters (referred to as max-regret game). This model enables us to use the double oracle method [25] and the policy-space response oracle (PSRO) framework [19] to incrementally generate strategies and

Proc.of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, London, UK.

[1] Code will be released with camera-ready version of the paper.

environment parameters to be considered. More specifically, MIRROR includes two RL-based oracles. The defender oracle solves a typical sequential decision-making problem and returns a defender strategy. The nature oracle finds out the environment parameters and the corresponding optimal defender strategy that lead to the highest regret. We use policy-gradient approach for both oracles. In the nature oracle, we treat the environment parameters as input to the policy network and update the environment parameters and the network parameters with a wake–sleep procedure. We further enhance the algorithm with parameter perturbation in both oracles.

Our contributions are summarized as follows. (1) We provide a realistic adversary model learned from real-world poaching data from Queen Elizabeth National Park (QENP) in Uganda, which demonstrates deterrence and opens the door to further RL research in service of protecting the environment. (2) We propose MIRROR, a framework to calculate minimax regret–optimal policies using RL for the first time, and apply this approach to green security domains. (3) We prove that MIRROR converges to an ε–optimal strategy in a finite number of iterations in our green security setting. (4) We empirically evaluate MIRROR on real-world poaching data from QENP.

2 RELATED WORK

Robust planning with minimax regret Minimax regret has been considered for preference elicitation of additive utilities [5] and rewards [33], as well as robotics planning in uncertain Markov decision processes with a model-based approach [34]. Double oracle [25] has been used to optimize for minimax regret in security games and in robust optimization [10, 30]. Double oracle has also been used without minimax regret for solving large zero-sum games [3, 14].

Robust planning in RL Robustness in RL has been heavily studied, both in the context of robust adversarial RL [31, 32, 46] and nonstationarity in multi-agent RL settings [21, 47]. For example, PSRO extends double oracle from state-independent pure strategies to policy-space strategies to be used for multiplayer competitive games [19]. Zhang et al. [46] consider robustness against adversarial perturbations on state observations. The line of work whose setting is most similar to our problem is robust RL with model uncertainty, specifically in the transition and reward functions [39, 47]. However, these approaches all consider robustness subject to maximin reward, whereas we optimize for minimax regret robustness. The two objectives are incompatible; we cannot simply substitute minimax regret into the reward function and solve using minimax reward, as computing the maximum regret incurs the challenge of knowing the optimal reward.

Green security games (GSGs) Literature on GSGs model the problem in green security domains as a game between a defender and boundedly rational attackers, with the assumption that attacker models can be learned from data [7, 29, 42, 45]. Most of this work does not consider uncertainty in the learned attacker model and solve the patrol planning problem using mathematical programming, which is not scalable for planning sequential patrols over time horizons going beyond 2 to 3 timesteps. RL has been used for planning in GSGs with real-time information to model defenders responding to footprints during a patrol [41]. However, uncertainty and robustness have not been explicitly considered together in GSG literature and much existing work on green security do not have access to real-world data and realistic models of deterrence.

3 PROBLEM STATEMENT

In green security settings, we have a *defender* (e.g., ranger) who conducts patrols in a protected area to prevent resource extraction by an *attacker* (e.g., poacher or illegal logger). Let N be the number of targets, such as 1×1 km regions in a protected area, that we are trying to protect. We have timesteps $t = 1, 2, \ldots, T$ up to some finite time horizon T where each timestep can represent, for example, a one-month period. The defender needs to choose a patrol strategy (also called policy) $\pi \in \Pi$, which sequentially allocates patrol effort. We denote patrol effort as $\mathbf{a}^{(t)}$, where $a_i^{(t)} \in [0, 1]$ represents how much effort the patrollers allocate to target i at time t, subject to a budget B such that $\sum_i a_i^{(t)} \leq B$ for all t.

Consider the poaching scenario specifically. Let $\mathbf{w}^{(t)} \in \mathbb{R}^N_{\geq 0}$ describe the distribution of wildlife in a protected area at timestep t, with $w_i^{(t)}$ denoting wildlife density in target i. What the rangers care about the most is the total wildlife density by the end of the planning horizon, i.e., $\sum_i w_i^{(T)}$. Poachers come into the park and place snares to trap animals. Their behavior is governed by a number of factors including the current patrol strategy, the past patrol strategy due to the deterrence effect, the geographic features including distance from the park boundary, elevation, and land cover, and others. Lacking complete and high-quality data about past poaching patterns, we are not able to build an accurate model of the poacher behavior.

Therefore, we consider a parameterized model for attacker's behavior and assume that the values of some of the parameters, denoted by $\mathbf{z}$, are uncertain. We assume that $\mathbf{z}$ is in a given uncertainty region Z that specifies a range $z_j \in [\underline{z_j}, \overline{z_j}]$ for each uncertain parameter j. We have no a priori knowledge about distribution over Z. We want to plan a patrol strategy π for the defender that is robust to parameter uncertainty following the minimax regret criterion. Let $r(\pi, \mathbf{z})$ be the defender's expected reward for taking policy π under environment parameters $\mathbf{z}$, e.g., the expected total wildlife density at the end of the planning horizon. Then the regret incurred by the agent for playing strategy π when the parameter values are $\mathbf{z}$ is $regret(\pi, \mathbf{z}) = r(\pi^*(\mathbf{z}), \mathbf{z}) - r(\pi, \mathbf{z})$, where $\pi^*(\mathbf{z})$ is the optimal policy that maximizes reward under parameters $\mathbf{z}$.

Our objective is then to find a strategy π for the defender that minimizes maximum possible regret under any parameter values $\mathbf{z}$ that falls within the uncertainty region Z. Formally, we want to solve the following optimization problem

$$\min_{\pi} \max_{\mathbf{z}} \left(r(\pi^*(\mathbf{z}), \mathbf{z}) - r(\pi, \mathbf{z}) \right) . \tag{1}$$

We can formulate this robust planning problem as a two-player game between an *agent* who wants to learn an optimal defender strategy (or policy) π against *nature* who selects worst-case parameter values $\mathbf{z}$. Then the agent's payoff is $-regret(\pi, \mathbf{z})$ and nature's payoff is $regret(\pi, \mathbf{z})$.

Definition 1 (Max-regret game)**.** We define the *max-regret game* as a zero-sum game between agent and nature where the agent's

payoff is

$$payoff(\pi, \mathbf{z}) = -regret(\pi, \mathbf{z}) = r(\pi, \mathbf{z}) - r\left(\pi^*(\mathbf{z}), \mathbf{z}\right) \ . \quad (2)$$

The agent can also choose a mixed strategy (or randomized policy) $\tilde{\pi}$, which is a probability distribution over Π. We denote by $\Delta(\Pi)$ the set of defender's mixed strategies. Likewise, we have mixed strategy $\tilde{\mathbf{z}} \in \Delta(Z)$ for nature.

Generalizability. Our approach applies not just to green security domains, but is in fact applicable to any setting in which we must learn a sequential policy π with uncertainty in some environment parameters $\mathbf{z}$ where our evaluation is based on minimax regret. Our framework is also not restricted to hyper-rectangular shaped uncertainty regions; any form of uncertainty with a compact set on which we do not have a prior belief would work.

3.1 Real-World Deterrence Model

No previous work in artificial intelligence or conservation biology has provided evidence of deterrent effect of ranger patrols on poaching, a topic critically important to planning real-world ranger patrols. Thus in our work on planning for green security domains, we began by exploring an open question about how poachers respond to ranger patrols.

Past work has investigated deterrence to inconclusive results [6, 9]. Using real poaching data from Queen Elizabeth National Park (QENP) in Uganda, we study the effect of patrol effort on poacher response. We find clear evidence of deterrence in that higher levels of past patrols reduce the likelihood of poaching. We are the first to do so. We also find that more past patrols on neighboring targets increase the likelihood of poaching, suggesting *displacement*.

Figure 2: Snares.

For each target, we calculate the total ranger patrol effort (in kilometers patrolled) and count the number of instances of illegal activity detected per month. We construct the patrol effort from 138,000 GPS waypoints across seven years of QENP poaching data. Observations of illegal activity are predominantly snares, but also include bullet cartridges, traditional weapons, and encounters with poachers.

Let z_i be the attractiveness of target i. To understand the effect of patrol effort on poaching activity, we learn the probability of detecting illegal activity in target i as a linear combination of $z_i + \gamma \cdot a_i^{(t)} + \beta \cdot a_i^{(t-1)}$, which is then squashed through the logistic function. The parameter β is the coefficient on past patrol effort $a_i^{(t-1)}$, measuring the deterrence effect we are trying to isolate, and γ is the coefficient on current patrol effort $a_i^{(t)}$, measuring the difficulty of detecting snares.

See Table 1 for the learned values of the average attractiveness of each target $\overline{z_i}$, the coefficient on current effort γ, and the coefficient on past effort β. Each row studies this effect for a different time interval. For example, 1 year, 3 months looks at the impact of a year of previous patrol effort on illegal activity in the subsequent three months. The parameter values are normalized. The learned value of β is negative across all datasets and settings. *Thus, increased past patrol effort does have a measurable effect of deterring poaching.*

Table 1: Learned coefficients, revealing deterrence

	$\overline{z_i}$	γ	β
1 month, 1 month	−9.285	1.074	−0.165
3 month, 3 month	−10.624	0.685	−0.077
1 year, 1 month	−9.287	1.061	−0.217
1 year, 3 month	−10.629	0.676	−0.042
1 year, 1 year	−8.559	2.159	−0.306

Table 2: Learned coefficients, with neighbors included, revealing displacement

	$\overline{z_i}$	γ	β	η
3×3	−10.633	0.687	−0.098	0.696
5×5	−10.636	0.688	−0.097	0.392
7×7	−10.632	0.688	−0.097	0.518

Ideally, when poachers are deterred by ranger patrols, they would leave the park completely and desist their hunt of wildlife. Alternatively, they may move to other areas of the park. We show that the latter appears to be true. To do so, we study the spatial relationship between neighboring targets, using three spatial resolutions: 3×3, 5×5, and 7×7. We learn

$$z_i + \gamma \cdot a_i^{(t)} + \beta \cdot a_i^{(t-1)} + \eta \cdot \sum_{j \in \text{neighbors}(i)} a_j^{(t-1)} \quad (3)$$

where η is the coefficient on past patrol effort on neighboring cells. As shown in Table 2, all learned values of η are positive, indicating that increased patrols on neighboring areas increases the likelihood of poaching on a target in the next timestep. This result is consistent across the three spatial resolutions, and strongest for the narrowest window of 3×3. Observe as well that the values for $\overline{z_i}$, γ, and β are remarkably consistent, demonstrating the robustness of our findings.

3.2 Green Security Model

In green security settings, the environment dynamics, including attacker behavior, can be described by an uncertain Markov decision process (UMDP) defined by the tuple $\langle \mathcal{S}, \mathbf{s}_0, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$. The *state* $\mathbf{s}$ is a tuple $(\mathbf{a}^{(t-1)}, \mathbf{w}^{(t-1)})$ of past patrol effort and past wildlife with initial state $\mathbf{s}_0 = (\mathbf{0}, \mathbf{w}^{(0)})$. The *action* $\mathbf{a}^{(t)}$ is an effort vector describing time spent in each target, subject to a budget B. Note that the model can be generalized to consider a sequence of past effort and wildlife to model an attacker with a longer memory length.

The environment dynamics are governed by the *transitions*, a compact set $\mathcal{T}$ containing the possible mappings $\mathcal{T}_\mathbf{z} : \mathcal{S} \mapsto \mathcal{S}$ where the transition $\mathcal{T}_\mathbf{z} \in \mathcal{T}$ depends on environment parameters $\mathbf{z}$. A mixed strategy $\tilde{\mathbf{z}}$ would produce a distribution over $\mathcal{T}$. These transitions are what makes our Markov decision process uncertain, as we do not know which mapping is the true transition. We model the adversary behavior with a simple logistic model, based on learned deterrence effect. The probability that the poacher will attack a

target i is given by the function

$$p_i^{(t)} = \text{logistic}\left(z_i - \beta \cdot a_i^{(t-1)} + \eta \cdot \sum_{j \in \text{neighbors}(i)} a_j^{(t-1)}\right) \quad (4)$$

where parameters $\beta > 0$ and $\eta > 0$ govern the deterrence and displacement effects, as described in Section 3.1. At each time step, the poacher takes some action $k_i \in \{0, 1\}$ where they either place a snare $k_i = 1$ or not $k_i = 0$. The realized adversary attack k_i is drawn from Binomial distribution $k_i \sim B(p_i)$.

The actions of the poacher and ranger then affect the wildlife population of the park. We use a regression model as in

$$w_i^{(t)} = \max\{0, (w_i^{(t-1)})^{\psi} - \alpha \cdot k_i^{(t-1)} \cdot (1 - a_i^{(t)})\} \quad (5)$$

where $\alpha > 0$ is the strength of poachers eliminating wildlife, and $\psi > 1$ is the wildlife natural growth rate.

Our objective is to maximize the number of wildlife. The *reward* R is the sum of wildlife at the time horizon, so $R(\mathbf{s}^{(t)}) = \sum_{i=1}^{N} w_i^{(T)}$ if $t = T$ and $R(\mathbf{s}^{(t)}) = 0$ otherwise. To understand the relationship between defender reward R in the game and the expected reward r of the agent oracle from our objective in Equation 1, we have

$$r(\pi, \mathbf{z}) = \mathbb{E}\left[R(\mathbf{s}^{(T)})\right] \quad (6)$$

taking the expectation over $\mathbf{s}^{(t+1)} \sim \mathcal{T}_{\mathbf{z}}(\mathbf{s}^{(t)}, \pi(\mathbf{s}^{(t)}), \mathbf{s}^{(t+1)})$ with initial state $\mathbf{s}^{(0)} = (\mathbf{w}^{(0)}, \mathbf{0})$.

4 ROBUST PLANNING

We propose MIRROR, which stands for MInimax Regret Robust ORacle. MIRROR is an algorithm for computing minimax regret–optimal policies in green security settings to plan patrols for a defender subject to uncertainty about the attackers' behavior. MIRROR also applies in generic RL contexts with a compact uncertainty set over transitions and rewards.

To learn a minimax regret–optimal policy for the defender, we take an approach based on double oracle [25]. Given our sequential problem setting of green security, we build on policy-space response oracle (PSRO) [19]. As discussed in Section 3, we pose the minimax regret optimization as a zero-sum game in the max regret space, between an agent (representing park rangers) who seeks to minimize max regret and nature (uncertainty over the adversary behavior parameters) which seeks to maximize regret. Our objective can be expressed as an optimization problem, as defined in Equation 1.

The full MIRROR procedure for minimax regret optimization using RL is given in Algorithm 1 and visualized in Figure 3. The three necessary components are:

(1) **Agent oracle**: An RL algorithm that, given mixed strategy $\tilde{\mathbf{z}}_e$ as a distribution over Z_e, learns an optimal policy π_e for the defender to maximize reward in the known environment described by $\tilde{\mathbf{z}}_e$.
(2) **Nature oracle**: An RL algorithm to compute an alternative policy $\hat{\pi}_e$ and new environment parameters $\mathbf{z}_e$ given the current agent mixed strategy $\tilde{\pi}_e$ over all policies Π_e. The nature oracle's objective is to maximize regret: the difference between expected value of alternative policy $\hat{\pi}_e$ and the agent strategy $\tilde{\pi}_e$.

Algorithm 1 MIRROR: MInimax Regret Robust ORacle

Input: Environment simulator and parameter uncertainty set Z
Params: Convergence threshold ε, num perturbations O
Output: Minimax regret–optimal agent mixed strategy $\tilde{\pi}^*$

1: Select an initial parameter setting $\mathbf{z}_0 \in Z$ at random
2: Compute baseline and heuristic strategies $\pi_{B_1}, \pi_{B_2}, \ldots$
3: $Z_0 = \{\mathbf{z}_0\}$
4: $\Pi_0 = \{\pi_{B_1}, \pi_{B_2}, \ldots\}$
5: **for** epoch $e = 1, 2, \ldots$ **do**
6: $(\tilde{\pi}_e, \tilde{\mathbf{z}}_e) = \textsc{ComputeMixedNash}(\Pi_e, Z_e)$
7: $\pi_e = \textsc{AgentOracle}(\tilde{\mathbf{z}}_{e-1})$
8: $(\mathbf{z}_e, \hat{\pi}_e) = \textsc{NatureOracle}(\tilde{\pi}_{e-1})$
9: **if** $regret(\tilde{\pi}_{e-1}, \mathbf{z}_e) - regret(\tilde{\pi}_{e-1}, \tilde{\mathbf{z}}_{e-1}) \leq \varepsilon$ and $r(\pi_e, \tilde{\mathbf{z}}_{e-1}) - r(\tilde{\pi}_{e-1}, \tilde{\mathbf{z}}_{e-1}) \leq \varepsilon$ **then**
10: **return** $\tilde{\pi}_e$
11: **for** perturbation $o = 1, \ldots, O$ **do**
12: perturb $\mathbf{z}_e$ as $\mathbf{z}_e^o$
13: $\pi_e^o = \textsc{AgentOracle}(\mathbf{z}_e^o)$
14: Compute expected rewards as $r(\pi_e, \mathbf{z})$ for all $\mathbf{z} \in Z_{e-1}$ and $r(\pi, \mathbf{z}_e)$ for all $\pi \in \Pi_{e-1}$
15: Compute max-regret game payoffs as Equation 2
16: $Z_e = Z_{e-1} \cup \{\mathbf{z}_e, \mathbf{z}_e^1, \ldots, \mathbf{z}_e^O\}$
17: $\Pi_e = \Pi_{e-1} \cup \{\pi_e, \hat{\pi}_e, \pi_e^1, \ldots, \pi_e^O\}$

Ideally, the alternative policy would be the optimal policy given environment parameters $\mathbf{z}_e$, that is, $\hat{\pi}_e = \pi^*(\mathbf{z}_e)$. However, given that these RL approaches do not guarantee perfect policies, we must account for the imperfection in these oracles, which we discuss in Section 4.4.

(3) **Mixed Nash equilibrium solver**: A solver to compute a mixed Nash equilibrium for each player as a distribution over Π_e for the agent and over Z_e for nature in the max-regret game defined in Definition 1.

The MIRROR procedure would unfold as follows. We begin with arbitrary initial parameter values $\mathbf{z}_0$ and baseline strategies. The agent then learns a best-response defender policy π_1 against these initial parameter values. Nature responds with $\mathbf{z}_1$. We update the payoff matrix in the max-regret game, add the best response strategies π_e and $\mathbf{z}_e$ to the strategy sets Π_e and Z_e for the agent and nature respectively, and continue until convergence. Upon convergence (line 10), we reach an ε-equilibrium in which neither player improves their payoff by more than ε.

In many double oracle settings, the process of computing a best response is typically fast, as the problem is reduced to single-player optimization. However, the nature oracle is particularly challenging to implement due to our objective of minimax regret. Additionally, the imperfect nature of our oracles implies we are not guaranteed to find exact best strategies. We discuss our approaches below.

4.1 The Agent Oracle

We want to find the best policy in a given environment setting. In our specific setting of poaching prevention, we consider deep deterministic policy gradient (DDPG) [22]. Policy gradient methods allow us to differentiate directly through a parameterized policy, making them well-suited to continuous state and action spaces,

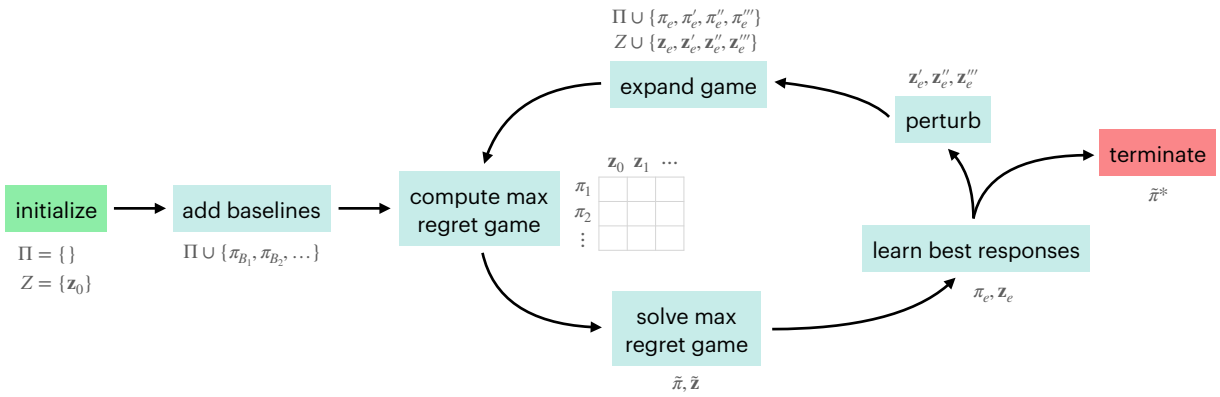

Figure 3: Our MIRROR algorithm, with figure design inspired by the double oracle figure from Bošanskỳ et al. [4].

Algorithm 2 Nature Oracle

Input: Agent mixed strategy $\tilde{\pi} \in \Delta(\Pi)$
Parameters: Wake–sleep frequency κ, num episodes J
Output: Nature best response environment parameters $\mathbf{z}$ and alternative policy $\hat{\pi}$

1: Randomly initialize $\mathbf{z}$ and $\hat{\pi}$
2: **for** episode $j = 1, 2, \ldots, J$ **do**
3: Sample agent policy $\pi \sim \tilde{\pi}$
4: **for** timestep $t = 1, \ldots, T$ **do**
5: **if** j mod $2\kappa = 0$ **then** Unfreeze $\hat{\pi}$ and $\mathbf{z}$
6: **else if** j mod $\kappa = 0$ **then** Freeze $\hat{\pi}$ parameters
7: **else** Freeze $\mathbf{z}$ parameters
8: Update $\hat{\pi}$ and $\mathbf{z}$ using gradient ascent to maximize regret: $r(\hat{\pi}, \mathbf{z}) - r(\pi, \mathbf{z})$
9: **return** $\mathbf{z}$, $\hat{\pi}$

which we have. Note again that MIRROR is agnostic to the specific algorithm used. DDPG specifically is not necessary; technically, the approach need not be RL-based as long as it enables efficient computation of a best response strategy.

We initialize the agent's strategy set Π with the baseline algorithms, described in Section 6. Other heuristic strategies, based on expert knowledge from the rangers, could be added as part of the initialization.

4.2 The Nature Oracle

Learning the nature oracle is one of the key challenges. Our insight is that the nature oracle's task is to perform the same task as the agent oracle, combined with the (non inconsequential) task of learning the optimal environment parameters, made difficult by the minimax regret criterion. The nature oracle may use a similar RL setup as the agent oracle, but we now face the challenging task of updating both the alternative policy $\hat{\pi}$ as well as the environment parameters $\mathbf{z}$ — and the setting of $\mathbf{z}$ changes both the rewards of the policies π and $\hat{\pi}$.

An initial approach might be to use two separate optimizers, one to train $\hat{\pi}$ and another to learn $\mathbf{z}$. However, as the environment parameters $\mathbf{z}$ and the alternative policy $\hat{\pi}$ are strongly correlated, optimizing them separately would lead to sub-optimal solutions. Therefore, we integrate $\mathbf{z}$ and $\hat{\pi}$ in the same actor and critic networks in DDPG and optimize the two together.

Our approach for the nature oracle is given in Algorithm 2. Similar to the agent oracle, we use policy gradient to learn the alternative policy $\hat{\pi}$, which enables us to take the derivative directly through the parameters of $\hat{\pi}$ and $\mathbf{z}$ to perform gradient descent. Note that the input to the DDPG policy learner is not just the state $\mathbf{s}^{(t)} = (\mathbf{a}^{(t-1)}, \mathbf{w}^{(t-1)})$ but also attractiveness: $(\mathbf{z}, \mathbf{a}^{(t-1)}, \mathbf{w}^{(t-1)})$. Ideally, we would incrementally change the parameters $\mathbf{z}$, then optimally learn each time. But that would be very slow in practice, requiring full convergence of DDPG to train $\hat{\pi}$ at every step. We compromise by adopting a wake–sleep procedure [13] where we alternately update only $\hat{\pi}$, only $\mathbf{z}$, or both $\hat{\pi}$ and $\mathbf{z}$ together. We describe the procedure in lines 5–7 of Alg. 2, were κ is a parameter controlling the frequency of updates between $\mathbf{z}$ and $\hat{\pi}$.

4.3 Mixed Nash Equilibrium Solver

We solve for the mixed Nash equilibrium in the max-regret game with the support enumeration algorithm [35], a solution approach based on linear programming, using the Nashpy implementation [16]. There may be multiple mixed Nash equilibria, but given that the game is zero-sum, we may take any one of them as we discuss in Section 5.

4.4 Parameter Perturbation

Ideally, the learned alternative policy would be the optimal policy given environment parameters $\mathbf{z}$, that is, $\hat{\pi} = \pi^*(\mathbf{z})$. However, the RL approaches do not guarantee perfect policies. With RL oracles, we must consider the question: what to do when the oracles (inevitably) fail to find the optimal policy? Empirically, we observe that for a given environment parameter setting $\mathbf{z}$, the policy π learned by DDPG occasionally yields a reward $r(\pi, \mathbf{z})$ that is surpassed by another policy π' trained on a different parameter setting $\mathbf{z}'$, with $r(\pi, \mathbf{z}) < r(\pi', \mathbf{z})$. So clearly the defender oracle is not guaranteed to produce a best response for a given nature strategy.

Inspired by this observation, we make parameter perturbation a key feature of our approach (Algorithm 1 lines 11–13), inspired by reward randomization which has been successful in RL [38, 39]. In doing so, we leverage the property that, in theory, any valid policy can be added to the set of agent strategies Π_e. So we include all of

the best responses to perturbed strategies by the nature oracle (see Figure 3 for an illustration), which enables us to be more thorough in looking for an optimal policy π^* for each parameter setting as well as find the defender best response. In that way, the double oracle serves a role similar to an ensemble in practice.

Parameter perturbation is grounded in three key insights. First, the oracles may be imprecise, but evaluation is highly accurate (relative to the nature parameters). Second, we only have to evaluate reward once, then max regret can be computed with simple subtraction. So the step does not add much computational overhead. Third, adding more strategies to the strategy set comes at relatively low cost, as computing a mixed Nash equilibrium is relatively fast and scalable. Specifically, the problem of finding an equilibrium in a zero-sum game can be solved with linear programming, which has polynomial complexity in the size of the game tree. Thus, even if the oracles add many bad strategies, growing the payoff matrix, the computational penalty is low, and the solution quality penalty is zero as it never takes us further from a solution.

5 CONVERGENCE AND CORRECTNESS

We prove that Algorithm 1 converges to an ε–minimax regret optimal strategy for the agent in a finite number of epochs if the uncertain Markov decision process (UMDP) satisfies a technical condition. The key idea of the proof is to exploit the equivalence of the value of the max-regret game and the minimax regret–optimal payoff in the UMDP. For these quantities to be equivalent, the max-regret game induced by the UMDP must satisfy a variant of the minimax theorem. Two broad classes of games that satisfy this condition are games with finite strategy spaces and continuous games; we show that the green security model of Section 3.2 induces a continuous max-regret game.

We begin by observing that the lower value of the max-regret game is equal to the payoff of the minimax regret–optimal policy of the UMDP. Using Definition 1, we can write the *lower value* of the max-regret game as:

$$\underline{v} = \max_{\tilde{\pi}} \min_{\tilde{\mathbf{z}}} \left(r(\tilde{\pi}, \tilde{\mathbf{z}}) - r(\tilde{\pi}^*(\tilde{\mathbf{z}}), \tilde{\mathbf{z}}) \right) \tag{7}$$

which is algebraically equivalent to Equation 1 by the definition of $\tilde{\pi}^*$ and rearrangement.

The connection between the lower value and the payoff received by the row player is well known in games with finite strategy spaces as a consequence of the seminal minimax theorem [28]. However, no such result holds in general for games with infinite strategy spaces, where a mixed Nash equilibrium may fail to exist. For so-called continuous games, Glicksberg [11] shows that a mixed Nash equilibrium exists and the analogy to the minimax theorem holds.

Definition 2. A game is *continuous* if the strategy space for each player is non-empty and compact and the utility function is continuous in strategy space.

We formalize the required connection in Condition 1, which holds for both finite and continuous games.

Condition 1. *Let $(\tilde{\pi}, \tilde{\mathbf{z}})$ be any ε–mixed Nash equilibrium of the max-regret game and $\underline{v}$ be the lower value of the max-regret game. Then, $|\underline{v} - (r(\tilde{\pi}, \tilde{\mathbf{z}}) - r(\tilde{\pi}^*(\tilde{\mathbf{z}}), \tilde{\mathbf{z}}))| \leq \varepsilon$.*

We show that our green security UMDP induces a continuous max-regret game.

Proposition 1. *The max-regret game induced by the model of Section 3.2 is continuous.*

Proof. The defender's strategy space consists of an action in $[0, 1]^N$ responding to each state. Because each action is compact, the defender's strategy space is compact. Nature has a compact uncertainty space. Both are non-empty.

The defender's expected reward in the max regret game (Definition 1 and Equation 6) can be written as a composition of continuous functions: addition, multiplication, the max (required to compute max regret), the logistic function (required for Equation 4), and exponentiation (Equation 5). The composition of these functions is also continuous. □

We now prove the main technical lemma: that the defender oracle and the nature oracle calculate best responses in the max-regret game. Doing so implies that the mixed Nash equilibrium returned by Algorithm 1 in the final subgame over finite strategy sets (Π_e, Z_e) is an ε–mixed Nash equilibrium of the entire max-regret game. This result allows us to apply Condition 1, showing equivalence of the lower value of the max-regret game and the minimax regret–optimal payoff.

Lemma 1. *At epoch e, policy π_e and environment parameters $\mathbf{z}_e$ are best responses in the max-regret game to mixed strategies $\tilde{\mathbf{z}}_e$ and $\tilde{\pi}_e$, respectively.*

Proof. For the nature oracle, this is immediate because the reward of the nature oracle is exactly the payoff nature would receive in the max-regret game when playing against $\tilde{\pi}_{e-1}$. For the agent oracle, the expected payoff of a strategy π against $\tilde{\mathbf{z}}_{e-1}$ in the max-regret game is $\mathbb{E}_{\mathbf{z}\sim\tilde{\mathbf{z}}_{e-1}}[r(\pi, \mathbf{z}) - r(\pi^*(\mathbf{z}), \mathbf{z})]$. Because $r(\pi^*(\mathbf{z}), \mathbf{z})$ does not depend on π, the policy that maximizes $\mathbb{E}_{\mathbf{z}\sim\tilde{\mathbf{z}}_{e-1}}[r(\pi, \mathbf{z})]$ maximizes the agent's utility in the max-regret game. This quantity is exactly the reward for the agent oracle. □

Theorem 2. *If Condition 1 holds and Algorithm 1 converges, the agent mixed strategy returned by Algorithm 1 achieves a minimax regret that is at most ε less than the minimax regret–optimal policy. If the max-regret game is either continuous with $\varepsilon > 0$ or finite, Algorithm 1 converges in a finite number of epochs.*

Proof. Because the convergence condition for Algorithm 1 is satisfied, $(\tilde{\pi}_e, \tilde{\mathbf{z}}_e)$ is an ε–mixed Nash equilibrium in the max-regret game by Lemma 1. Applying Condition 1 yields the result that the payoff of $\tilde{\pi}_e$ is within ε of the minimax regret–optimal policy of the original UMDP.

If the max-regret game is finite, there are only finite number of strategies to add for each player and each strategy may be added only once—thus, Algorithm 1 converges in finitely many epochs. If the max-regret game is continuous, Theorem 3.1 of [1] guarantees convergence in finite epochs due to Lemma 1. □

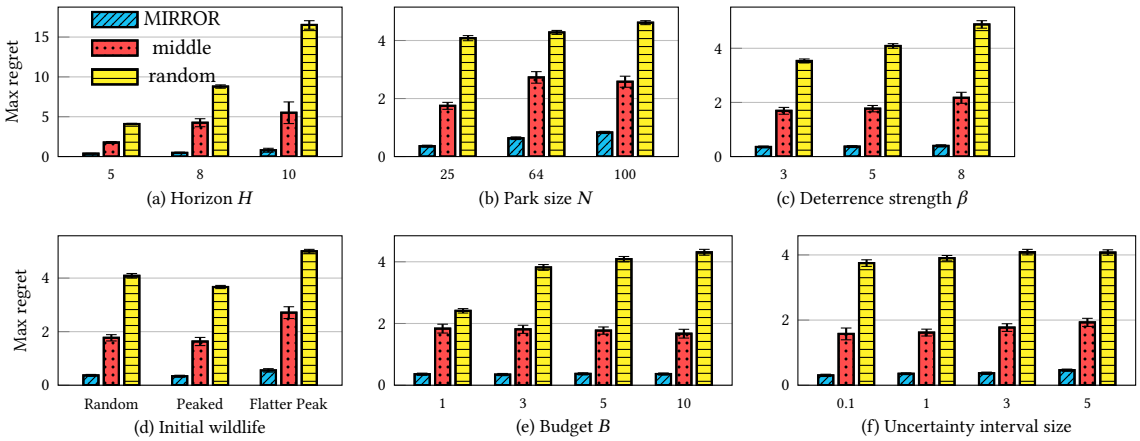

Figure 4: Comparing performance across varied settings, our MIRROR algorithm leads to the lowest max regret in all settings. We evaluate max regret by calculating the average reward difference between the selected policy and the optimal policy, with reward averaged over 100 episodes. We use as the default setting is $H = 5$, $N = 25$, $\beta = 5$, random wildlife initialization, $B = 5$, and uncertainty interval 3. Standard error shown averaged over 30 trials.

6 EXPERIMENTS

We conduct experiments using a simulator built from real poaching data from Queen Elizabeth National Park in Uganda, based on our analysis in Section 3.1. We consider robust patrol planning in the park with $N = 25$ to 100 targets representing reasonably the area accessible from a patrol post. Each target is a 1×1 km region.

We compare against the following baselines. *Middle* computes an optimal defender strategy assuming the true value of each parameter is the middle of the uncertainty interval. *Random* takes a random strategy regardless of state. We apply the same parameter perturbations to the baselines as we do to the others and report the top-performing baseline variant. We evaluate performance of all algorithms in terms of maximum regret, computed using the augmented payoff matrix (with baselines and perturbed strategies) described in Section 4. The max regret is calculated by determining, for each parameter value, the defender strategy with the highest reward. In every experiment setting, we use the same strategy sets to compute max regret for all of the approaches shown. Note that we would not expect any algorithm that optimizes for maximin reward to perform significantly better in terms of max regret than the middle strategy due to the regret criterion.

Figure 4 shows the performance of our MIRROR algorithm compared to the baselines. Across variations of episode horizon, park size, deterrence strength, wildlife initial distributions, budget, and uncertainty interval size, MIRROR significantly reduces max regret. Deterrence strength changes the value of β in Equation 4 to reveal the potential effectiveness of our actions. The wildlife initializations options are a uniform random distribution, a peaked Gaussian kernel (representing a core animal sanctuary in the park center), and a flatter Gaussian kernel (representing animals distributed more throughout the park, although more concentrated in the center). The uncertainty interval size restricts the maximum uncertainty range $\overline{z_i} - \underline{z_i}$ for any target i.

One of the most notable strengths for MIRROR is shown in Figure 4(a). As the episode horizon increases, thus the defender is tasked with planning longer-term sequences of decisions, MIRROR suffers only mildly more regret while the regret of the baseline strategies increases significantly. The scalability of MIRROR is further evidenced in Figure 4(b) as our relative performance holds when we consider larger-sized parks.

Our strong empirical performance offers promise for effective real-world deployment for MIRROR. Uncertainty in the exact environment parameters is one of the most prominent challenges of sequential planning in the complex real-world setting of green security.

7 CONCLUSION

Our work is the first, across artificial intelligence and conservation biology literature, to show ranger patrols do deter poachers on real-world poaching data. Following this finding, we identify the problem of sequential planning for green security that is robust to parameter uncertainty following the minimax regret criterion, a problem that has not been studied in the literature. We address this challenge with our novel RL-based framework, MIRROR, which enables us to learn policies evaluated on minimax regret. We show the strength of MIRROR both theoretically, as it converges to an ε–max regret optimal optimal strategy in finite iterations, and empirically, as it leads to low-regret policies. We hope that our results inspire more work in green security based on our realistic adversary model and that our MIRROR framework is useful for future work on learning RL-policies that are optimal under minimax regret.

REFERENCES

[1] Lukáš Adam, Rostislav Horčík, Tomáš Kasl, and Tomáš Kroupa. 2021. Double Oracle Algorithm for Computing Equilibria in Continuous Games. In *Proc. of AAAI-21*.
[2] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. 2012. Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. *Artificial Intelligence* 184 (2012), 78–123.
[3] Branislav Bosansky, Christopher Kiekintveld, Viliam Lisy, and Michal Pechoucek. 2014. An exact double-oracle algorithm for zero-sum extensive-form games

with imperfect information. *Journal of Artificial Intelligence Research* 51 (2014), 829–866.

[4] Branislav Bošanskỳ, Viliam Lisỳ, Marc Lanctot, Jiří Čermák, and Mark HM Winands. 2016. Algorithms for computing strategies in two-player simultaneous move games. *Artificial Intelligence* 237 (2016), 1–40.

[5] Darius Braziunas and Craig Boutilier. 2007. Minimax regret based elicitation of generalized additive utilities. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI-2007)*.

[6] Anthony Dancer. 2019. *On the evaluation, monitoring and management of law enforcement patrols in protected areas.* Ph.D. Dissertation. University College London.

[7] Fei Fang, Thanh H Nguyen, Rob Pickles, Wai Y Lam, Gopalasamy R Clements, Bo An, Amandeep Singh, Milind Tambe, and Andrew Lemieux. 2016. Deploying PAWS: Field Optimization of the Protection Assistant for Wildlife Security. In *Proc. of IAAI-16*.

[8] Fei Fang, Peter Stone, and Milind Tambe. 2015. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *Proc. of IJCAI-15*.

[9] Benjamin John Ford. 2017. *Real-world evaluation and deployment of wildlife crime prediction models.* Ph.D. Dissertation. University of Southern California.

[10] Hugo Gilbert and Olivier Spanjaard. 2017. A double oracle approach to minmax regret optimization problems with interval data. *European Journal of Operational Research* 262, 3 (2017), 929–943.

[11] I. L. Glicksberg. 1952. A further generalization of the Kakutani fixed point theorem, with application to Nash equilibrium points. In *Proceedings of the American Mathematical Society*. 170–174.

[12] Swaminathan Gurumurthy, Lantao Yu, Chenyan Zhang, Yongchao Jin, Weiping Li, Xiaodong Zhang, and Fei Fang. 2018. Exploiting data and human knowledge for predicting wildlife poaching. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS)*. 1–8.

[13] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. 1995. The "wake-sleep" algorithm for unsupervised neural networks. *Science* 268, 5214 (1995), 1158–1161.

[14] Manish Jain, Dmytro Korzhyk, Ondřej Vaněk, Vincent Conitzer, Michal Pěchouček, and Milind Tambe. 2011. A double oracle algorithm for zero-sum security games on graphs. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. 327–334.

[15] Debarun Kar, Benjamin Ford, Shahrzad Gholami, Fei Fang, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Mustapha Nsubaga, et al. 2017. Cloudy with a chance of poaching: Adversary behavior modeling and forecasting with real-world poaching data. In *Proc. of AAMAS-16*.

[16] Vincent Knight and James Campbell. 2018. Nashpy: A Python library for the computation of Nash equilibria. *Journal of Open Source Software* 3, 30 (2018), 904.

[17] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. 2010. Complexity of computing optimal stackelberg strategies in security resource allocation games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 24. Issue 1.

[18] Panos Kouvelis and Gang Yu. 2013. *Robust discrete optimization and its applications*. Vol. 14. Springer Science & Business Media.

[19] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. 2017. A unified game-theoretic approach to multiagent reinforcement learning. *Advances in neural information processing systems* 30 (2017), 4190–4203.

[20] Steven D Levitt. 1998. Why do increased arrest rates appear to reduce crime: deterrence, incapacitation, or measurement error? *Economic inquiry* 36, 3 (1998), 353–372.

[21] Shihui Li, Yi Wu, Xinyue Cui, Honghua Dong, Fei Fang, and Stuart Russell. 2019. Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4213–4220.

[22] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *Proc. of ICLR-16*.

[23] Graham Loomes and Robert Sugden. 1982. Regret theory: An alternative theory of rational choice under uncertainty. *The economic journal* 92, 368 (1982), 805–824.

[24] Janusz Marecki, Gerry Tesauro, and Richard Segal. 2012. Playing repeated stackelberg games with unknown opponents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. 821–828.

[25] H Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. 2003. Planning in the presence of cost functions controlled by an adversary. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 536–543.

[26] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[27] Jennifer F Moore, Felix Mulindahabi, Michel K Masozera, James D Nichols, James E Hines, Ezechiel Turikunkiko, and Madan K Oli. 2018. Are ranger patrols effective in reducing poaching-related threats within protected areas? *Journal of Applied Ecology* 55, 1 (2018), 99–107.

[28] J v Neumann. 1928. Zur theorie der gesellschaftsspiele. *Mathematische annalen* 100, 1 (1928), 295–320.

[29] Thanh H Nguyen, Arunesh Sinha, Shahrzad Gholami, Andrew Plumptre, Lucas Joppa, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Rob Critchlow, et al. 2016. CAPTURE: A New Predictive Anti-Poaching Tool for Wildlife Protection. In *Proc. of AAMAS-16*. 767–775.

[30] Thanh Hong Nguyen, Amulya Yadav, Bo An, Milind Tambe, and Craig Boutilier. 2014. Regret-Based Optimization and Preference Elicitation for Stackelberg Security Games with Uncertainty.. In *AAAI*. 756–762.

[31] Xinlei Pan, Daniel Seita, Yang Gao, and John Canny. 2019. Risk averse robust adversarial reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 8522–8528.

[32] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. 2017. Robust adversarial reinforcement learning. In *Proc. of ICML-17*.

[33] Kevin Regan and Craig Boutilier. 2009. Regret-based reward elicitation for Markov decision processes. In *Proc. of UAI-09*.

[34] Marc Rigter, Bruno Lacerda, and Nick Hawes. 2021. Minimax Regret Optimisation for Robust Planning in Uncertain Markov Decision Processes. In *Proc. of AAAI-21*.

[35] Tim Roughgarden. 2010. Algorithmic game theory. *Commun. ACM* 53, 7 (2010), 78–86.

[36] Leonard J Savage. 1951. The theory of statistical decision. *Journal of the American Statistical association* 46, 253 (1951), 55–67.

[37] Pier Giuseppe Sessa, Ilija Bogunovic, Maryam Kamgarpour, and Andreas Krause. 2020. Learning to Play Sequential Games versus Unknown Opponents. In *Proc. of NeurIPS-20*.

[38] Zhenggang Tang, Chao Yu, Boyuan Chen, Huazhe Xu, Xiaolong Wang, Fei Fang, Simon Shaolei Du, Yu Wang, and Yi Wu. 2021. Discovering Diverse Multi-Agent Strategic Behavior via Reward Randomization. In *Proc. of ICLR-21*.

[39] Jingkang Wang, Yang Liu, and Bo Li. 2020. Reinforcement Learning with Perturbed Rewards. In *Proc. of AAAI-20*.

[40] Tianhan Wang and Craig Boutilier. 2003. Incremental utility elicitation with the minimax regret decision criterion. In *Proc. of IJCAI-03*, Vol. 3. 309–316.

[41] Yufei Wang, Zheyuan Ryan Shi, Lantao Yu, Yi Wu, Rohit Singh, Lucas Joppa, and Fei Fang. 2019. Deep reinforcement learning for green security games with real-time information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1401–1408.

[42] Haifeng Xu, Benjamin Ford, Fei Fang, Bistra Dilkina, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Mustapha Nsubaga, et al. 2017. Optimal patrol planning for green security games with black-box attackers. In *Proc. of GameSec-17*. Springer, 458–477.

[43] Haifeng Xu, Long Tran-Thanh, and Nicholas R Jennings. 2016. Playing repeated security games with no prior knowledge. In *Proc. of AAMAS-16*. 104–112.

[44] Lily Xu, Shahrzad Gholami, Sara Mc Carthy, Bistra Dilkina, Andrew Plumptre, Milind Tambe, Rohit Singh, Mustapha Nsubuga, Joshua Mabonga, Margaret Driciru, et al. 2020. Stay Ahead of Poachers: Illegal Wildlife Poaching Prediction and Patrol Planning Under Uncertainty with Field Test Evaluations. In *Proc. of ICDE-20*.

[45] Rong Yang, Benjamin Ford, Milind Tambe, and Andrew Lemieux. 2014. Adaptive resource allocation for wildlife protection against illegal poachers. In *Proc. of AAMAS-14*. Citeseer, 453–460.

[46] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Duane Boning, and Cho-Jui Hsieh. 2020. Robust Deep Reinforcement Learning against Adversarial Perturbations on Observations. In *Advances in Neural Information Processing Systems*.

[47] Kaiqing Zhang, Tao Sun, Yunzhe Tao, Sahika Genc, Sunil Mallya, and Tamer Basar. 2020. Robust Multi-Agent Reinforcement Learning with Model Uncertainty. In *Advances in Neural Information Processing Systems*.