A Survey of Advancements in Green Security Games

Joe McCall

CAP5600

Abstract

The field of Green Security Games (GSG) has proven useful in the protection of wildlife. By modelling attackers and defenders as intelligent agents in a repeated simulation we can employ a winning algorithm to deploy scarce resources in actual green security scenarios. Such an abstraction also serves as the foundation upon which more intricate scenarios can be built, explored, and subsequently adapted to by a learning agent. This paper summarizes the concept of GSGs, surveys the advancements that have been made, provides analysis of the validity of the advancements, and suggests future opportunities for research.

1 Introduction

The domain of green security entails the struggle between poachers/illegal fishers and rangers charged to protect the wildlife. Endangered animals are trapped and killed/trafficked by poachers, and those charged to protect wildlife face a shortage of resources. Typically there are more targets than there are rangers to patrol them, and poachers will continuously attack different targets. Defenders must also protect potentially un-explored/under-explored areas, and operate out of fixed patrol towers to which they must return at the end of a shift. The deployment of limited resources under strict constraints can be facilitated using artificial intelligence. Despite best efforts, species like the African elephant are globally declining due to poaching [1]. Certain strategies to address this treat have been ineffective at best [2]. An approach is needed to capture poachers as they attack, which requires somehow predicting where and when they will attack. These types of predictions fit well within the field of Artificial Intelligence (AI).

Within AI, Stackelberg Security Games (SSG) have been used to predict potential attacker moves in order to assist in defender strategies. The application of this strategy to green security domains is called Green Security Games (GSG, introduced by [3]). The game abstracts the reality of poachers versus rangers patrolling a vast wildlife area into a grid-based simulation between attackers and defenders, both of which are AI agents. The attacker's goal is to place snares for wildlife without being caught, while the defender's goal is to detect both the snare and the attacker. Furthermore, the attacker behaves human-like and does not always behave optimally.

The game is run in episodes [3]. Each round the attacker chooses a target based on the highest estimated utility, and the defender must deploy its resources judiciously to defend those targets. The defender does not have enough resources to defend all targets, so it must employ a strategy to decide which targets should be defended and which targets should be left defenseless. By simulating multiple rounds of this game, an optimized defender strategy is formed. This strategy is applicable to assist real-world rangers in deciding where and how patrols should be deployed.

Advancements in this field introduce additional constraints, challenge assumptions, and propose novel methods for improving defender strategy. Qian et al. [4] show that the environment must be partially-observable to model the defender's decision to explore new areas or patrol areas known to have snares. Wang et al. [5] introduce the idea that both attackers and defenders have access to real-time information that can be used to evade and track. Finally Gholami et al. [6] show that a defender strategy that combines a machine-learning agent with an online learning algorithm that does not rely on prior information can outperform existing models.

The paper is organized in three sections. The Methods/Theory section summarizes the methods for the these papers. The Discussions section summarizes the results, and reflects on the soundness of each paper. The Conclusions section analyzes the research, and provides ideas for further research.

2 Methods/Theory

The methods and theory of each paper are described below. Included are the environment as described by the researcher, the method by which a trained AI agent interacts with the environment, and the output of the simulation. It is then discussed why the research is considered valid.

2.1 Green Security Games (GSG)

The Green Security Game as introduced by Fang et al. [3] is a zero-sum game, meaning that a positive environment for one player is a negative environment for the other. Specifically, for a given reward function, either the attacker is successful, or the defender, but not both. The world is a grid, and the attacker attempts to attack a cell while the defender tries to prevent the attack by guarding the cell.

The game is run in T ($< \infty$) rounds, and each round has multiple episodes. At the end of each episode the defender can change her deployment strategy. The defender has K guards to protect N ($\geq K$) targets, each with a different reward. A guard (defender) can defend one target *i* and an attacker can attack one target. If the attacker attacks an unguarded target, the defender is penalized and the attacker is rewarded. If the attacker attacks a defended target, the attacker is penalized and the defender is rewarded. After each round the defender assigns guards in order to maximize the expected utility. The attacker and defender strategies are described below.

Attacker

The strategy utilized by the attacker depends on the Subject Utility Quantal Response (SUQR) concept, which has proven to accurately model bounded rationality of human attackers [7].

In other words, the attacker uses the his belief in the defender's strategy c_t and his limited memory of previous rounds to decide on the target to attack in that round. This definition is given formally in [3].

Defender

The defender strategy is represented by a coverage vector $c = \langle c_i \rangle$ where $0 \leq c_i \leq 1$ is the probability that target i is covered by a guard. It also satisfies $\sum_{i=1}^{N} c_i \leq K$ [3]. In other words it is impossible to have more total coverage than there are guards.

To compute the utility of the defender we use fig. 1. The expected utility for a defender with a given strategy c is the probability target i is covered times the reward of that target for the defender, plus the probability that target is not covered times the penalty of that target for the defender.

$$U_i^d(c) = c_i R_i^d + (1 - c_i) P_i^d \tag{1}$$

The defender's strategy in round t is denoted by c^t.

The objective is to maximize the defender's average utility over the all the rounds of the game. This is done with a planning algorithm. Each episode the defender uses the configured algorithm to determine which areas to defend. The discussed algorithms include PlanAhead-M (PA-M), FixedSequence-M (FS-M), and an enhanced PA-M that incorporates Bayesian Updating (BU). They are formally defined in [3] and evaluated in the experiment. In particular, BU allows the learning algorithm to build a probability distribution to predict where the attacker will attack, based on historical data.

Output

The experiment evaluates the performance of a learning GSG algorithm with the baseline algorithms typically found in Stackelberg Security Games, and a non-learning myopic strategy which does not consider historical data. The experiment is run under controlled conditions under a strict time limit with the same data. The experiment is sound.

2.2 Exploration/Exploitation Tradeoffs

Qian et al. [4] built upon the work done by Fang et al. [3] by introducing a new constraint that fit the real-world better: the dilemma between exploring new territories and continuing to patrol known hot-spots.

The authors suggest that the original GSGs fail to consider that attacks on unguarded targets can only be discovered if the guards explore that area first. In other words, the environment is only partially observable by the defender. The defender must choose between patrolling targets with known poaching activities and exploring the targets that may or may not have been attacked.

Intuitively, the more the defender patrols an area, the less likely that area will be attacked, given the attacker can observe the defender's actions. Attacks are discretized into intensity. A more intense attack will yield a higher reward for the attacker, but is more observable by the defender, while a less intense attack yields a lower reward but is more covert. The experiment done by Qian et al. [4] models this property using a combination of transition matrices and an observation matrix. The experiment intends to learn the transition matrices, the observation matrix, and a value for the initial belief using an expectation-maximization algorithm (a technique used to handle non-observable variables). Each round, adjustments are made to the attacker transition matrices, a defender observation matrix, and an initial belief value.

Restless Multi-armed Bandit (RMAB) Problem

The dilemma between exploration and exploitation is modelled as a Restless Multi-armed Bandit (RMAB) problem. In such a problem, limited resources (guards) must protect several targets, but they have no insight into targets which they do not protect. In a traditional multi-armed bandit problem the player attempts to maximize his score by pulling one more arms of the slot machine to maximize the reward. The restless variant is introduced here to show that non-activated arms (i.e., unexplored areas) affect the "payout" of the activated arms. This means the environment being simulated is stochastic.

When RMAB is applied to GSG, the defender choosing an area to patrol is akin to the "pulling" of an arm for that round for a reward (the defender catches the attacker). Likewise, the attacker choosing to attack an area is "pulling" that arm for a reward (the attacker successfully ensnares an animal).

Whittle Index

The Whittle index is the heuristic index policy that assists the agent in deciding which arm to activate. Basically, the higher the Whittle index, the more likely the attacker or defender is to attack of defend an area respectively.

Output

A fully-trained agent from this experiment can be considered rational if it predicts where and when actual attacks happen. The simulation more closely matches the real world given the addition of the exploration/exploitation dilemma. This is an improvement over a model that ignores such a dilemma, so the research is valid.

2.3 Imperfect Prior Knowledge

The work done by Fang et al. [3] and Qian et al. [4] serve as a foundation upon which Gholami et al. [6] build. They show that a significant shortcoming is that the collected data on attacked targets (i.e., previously failed attempts prevent poaching) is highly biased toward the normal patrol routes. In other words, since we only know about attacks that we can observe, and the area around guard posts is more observed than other areas, historical data will reward guarding targets closer to the outposts much higher than other, potentially more valuable targets.

A modification of GSG is made to enhance the realism of the simulation. Routes taken by attackers and defenders take into account geographic features that affect the route's feasibility. Furthermore a player can only travel so far in a single shift, before returning to base. This is modelled using a "time-unrolled graph" [6], where each node is a cell reachable from the previous cell. The RMAB strategy is used in this case as well, though it is known as "adversarial bandit" in this paper.

To maximize the utility of the defender in this modified game Gholami et al. [6] introduce MINION (MultI-expert oNline model for constraIned patrOl plaNning [sic]). As its name suggests, it employs the results of two AI agents - an online real-time agent called MINION-sm (sub-module) and a ML agent trained on historical data. The agents are explained in the sections below, and the MINION algorithm is expanded upon.

MINION-sm

MINION-sm as the online-learning algorithm that is used without historical data. It starts with the FPL-UE (follow-the perturbed-leader with uniform exploration) algorithm. It then randomly flips a coin to choose between exploration/exploitation. MINION-sm then adds scheduling constraints to FPL-UE. During the exploration phase it chooses a route using a random approach. During the exploitation phase it chooses an optimized route based on the estimated rewards from previous rounds.

Machine Learning (ML)

The machine learning agent takes in years of historical data as inputs and gives predictions of attacks at certain locations in an observation vector. The agent is trained on this data until the error rate is minimized.

MINION

The MINION algorithm combines the advice given by MINION-sm and ML. It starts with MINION-sm. At the end of each round the historical performance of each expert is assessed, and the results are used to decide which expert will be used in the next round. In other words, MINION learns whether the ML expert based on historical data is reliable or not.

Output

MINION is run in a fixed number of rounds against other defender strategies. A result showing improved performance of MINION over others would yield a potential real-world strategy. Thus the research is valid.

2.4 Real-Time Information

The work done by Fang et al. [3] and Qian et al. [4] describe a partially observable environment in which the attacking agent and the defending agent can observe the actions of the other when they happen. Attackers can observe a defender patrolling an area, and a defender can observe an attacker's attack. This work describes historical information, but in an actual patrol real-time information becomes available, for example when a patrol encounters footprints. Wang et al. [5] addresses this by modelling real-time information to the game. A new learning-based algorithm is introduced (DeDOL) to help the defenders optimize their patrols.

GSG-I

Similar to GSG [3], GSG-I maps the world into a grid, is a zero-sum game between the attacker and the defender, and is episodic. A key difference is when attackers and defenders move they now leave detectable footprints. Both players employ multiple strategies (e.g., follow footsteps or evade) based on a probability distribution.

This is a closer model to reality because both poachers and patrollers leave traces of their presence (footprints, blood, broken branches, etc.), and veterans from each respective group will naturally have an advantage over their less experienced counterparts.

The addition of this real-time information greatly affects the complexity of the game, however. Specifically it now models an extensive-form game, in which an exponentially-growing game tree is used [8]. As such, teaching such an agent is only feasible in small games, but it is shown later that the resulting strategy is applicable to larger games.

DeDOL Algorithm

DeDOL uses a neural network to learn successful defender strategies for fixed movement. The double-oracle method introduced by McMahan et al. [9] is used by providing two heuristic strategies, one for the attacker and one for the defender. Each iteration both neural networks train using the real-time information gained that round.

Outputs

The simulation runs the now more-complex GSG-I game with DeDOL, and compares it with conterfactual regret (CFR) minimization on small games, which is a known solution to extensive-form games. A good result indicates an improvement to existing processes, so the method is sound.

3 Discussions

This section discusses the experimental results of the work outlined in this paper, along with its contributions to the field of GSG.

The introduction of GSG has effectively abstracted the conflict between poachers and the law enforcement agents attempting to stop them. The ability to simulate multiple rounds against a simulated human presents a clear benefit for learning algorithms to assist in the deployment of guards to patrol vast areas. Fang et al. [3] showed that the enhanced PA-M strategy provided a high average expected utility compared to the baseline algorithm used. Assuming the abstraction is valid, this indicates the strategy would be successful if employed by actual law enforcement.

The work done by Qian et al. [4] successfully introduce a more realistic scenario when simulating GSGs. The use of the RMAB is a useful abstraction of the real-world dilemma between exploring new areas and patrolling existing areas. The results of the experiment clearly show an improvement over existing models, especially as learning rounds increase. Care must be taken to address the fact that poachers are often well-funded [2] and have just as much observability of the defenders as the defenders do of the poachers.

Gholami et al. [6] found that a combination of experts in their MINION algorithm significantly out-performed existing algorithms in traditional GSGs. It also follows that the nature of the online expert-evaluation not only rules out faulty historical data, but an over-fitted ML expert. It is remarkable that the combination of two strategies can outperform each individual strategy.

GSG-I was introduced by Wang et al. [5] revealed that real-time information adds a measure of complexity to GSG, but makes the abstraction more accurate. Furthermore, while their simulations could only train on small games, the resulting trained neural network out-performed existing extensive-form games when tested in a larger game. This is a very successful advancement in the field and will likely lead to more sophisticated patrol strategies.

There are many additional areas of research in the field of GSG. Just within the scope of this paper, one must wonder if the assumptions on historical data brought up by Gholami et al. [6] affect the simulation run by Wang et al. Furthermore, could a MINION agent result in improvements when DeDOL is combined with MINION-sm?

Additionally, the abstraction that GSG provides is necessarily loose, but still can be further refined. For example, how have recent ivory laws protecting elephants [1] affected the validity of the historical data? Or more grimly, how can the decline in the numbers of endangered animals affect the availability of more attacks?

More importantly, the GSG papers above assume that the adversaries act with bounded rationality (given their limited observability of defender strategies). However this assumption may only be useful in the short-term. Access to computing resources (and even the PAWS application) is growing more ubiquitous, and it's only a matter of time before poachers have the same access to the AI research as law enforcement. A means to detect whether or not the attacker is utilizing AI assistance would be highly valuable in this case. For example, the defending agent could keep track of how many times a prediction with a high confidence is wrong. Once that counter reaches a threshold it would suggest that the attacker somehow has knowledge of that agent's predictions. This is an area of AI that is very much worth exploring.

4 Conclusions

- Summarize research
- Discuss how PAWS is helping law enforcement currently, and how these algorithms can be used to improve it
- Find a unique idea as a future research goal

5 References

[1] G. Wittemyer, J. M. Northrup, J. Blanc, I. Douglas-Hamilton, P. Omondi, and K. P. Burnham, "Illegal killing for ivory drives global decline in african elephants," *Proceedings of the National Academy of Sciences*, 2014, doi: 10.1073/pnas.1403984111.

[2] J. Rademeyer, "An unwinnable war: Rhino poaching in the kruger," in *Militarised Responses to Transnational Organised Crime: The War on Crime*, 2018, pp. 43–59.

[3] F. Fang, P. Stone, and M. Tambe, "When security games go green: Designing defender strategies to prevent poaching and illegal fishing," in *Proceedings of the 24th international conference on artificial intelligence*, 2015, pp. 2589–2595.

[4] Y. Qian, C. Zhang, B. Krishnamachari, and M. Tambe, "Restless poachers: Handling exploration-exploitation tradeoffs in security domains," in *Proceedings of the 2016 international conference on autonomous agents & multiagent systems*, 2016, pp. 123–131.

[5] Y. Wang *et al.*, "Deep reinforcement learning for green security games with real-time information," in *Proceedings of the thirty-third aaai conference on artificial intelligence*, 2019, pp. 1401–1408.

[6] S. Gholami, A. Yadav, L. Tran-Thanh, B. Dilkina, and M. Tambe, "Don't put all your strategies in one basket: Playing green security games with imperfect prior knowledge," in *Proceedings of the 18th international conference on autonomous agents and multiagent systems*, 2019, pp. 395–403.

[7] T. H. Nguyen, R. Yang, A. Azaria, S. Kraus, and M. Tambe, "Analyzing the effectiveness of adversary modeling in security games," in *Proceedings of the twenty-seventh aaai conference on artificial intelligence*, 2013, pp. 718–724.

[8] S. Russell and P. Norvig, *Artificial intelligence: A modern approach*, 3rd ed. USA: Prentice Hall Press, 2009.

[9] H. McMahan, G. Gordon, and A. Blum, “Planning in the presence of cost functions controlled by an adversary.” 2003, pp. 536–543.