Online Agent-Based Aerial Patrol Planning for Wildlife Surveillance

K. Dhoore

Supervisor: dr. O.A. Sharpanskykh
Delft University of Technology, Faculty of Aerospace Engineering, Department of Control & Operations
Kluyverweg 1, 2629HS Delft, The Netherlands

***Abstract*—Wildlife conservation efforts are constrained by a limited amount of resources available for surveillance activities. UAVs are used increasingly to assist rangers in patrol tasks. Effectively patrolling wildlife parks requires detailed knowledge of the environment and its threats, which is not always available. Previous work in Green Security Games (GSGs) that aims to develop defensive strategies to deter adversaries relies on historical poaching data to train machine learning models. Recent advancements in the field have led to the development of an online learning framework that does not require prior data. However, the defensive strategies resulting from this approach are focused on foot patrols by rangers, which do not have the same mobility as UAVs, or do not take into account spatio-temporal constraints associated with patrolling in a real-world situation at all. To address the desire of using UAVs for wildlife surveillance, this paper proposes MEOMAPP, a model that extends on the online learning approach by incorporating a patrol planning algorithm more suitable for aerial patrol. It also includes an evaluative algorithm that considers a human expert next to the online learning expert and balances the application of their strategies based on the observed performance of each expert. By simulating MEOMAPP in a realistic environment, the research demonstrates that the model is suitable to determine aerial surveillance strategies for wildlife conservation.**

***Index Terms*—Green Security Games; Game Theory; Online Learning; Adversarial Bandits; Agent-Based Modelling; Aerial Surveillance; Wildlife Conservation**

I. Introduction

Poaching is still a major problem in large parts of the world. It threatens efforts in wildlife conservation, which negatively impacts biodiversity and possibly results in damaged ecosystems [1]. There is also a large economic cost in the form of reduced income from wildlife tourism and trophy hunters. Currently, the cost of measures to keep animals protected and safe from poachers is not economically viable [2]. Simultaneously, protecting wildlife is not always without risks either. In Africa alone, 349 rangers have died on duty since 2012, although it is thought these figures are substantially higher due to lack of reporting [3].

These high costs are a driver for cost-effective and innovative measures in the wildlife protection domain. Notably, the use of artificial intelligence (AI) has shown potential for detecting animals and poachers with object and image recognition [4, 5], and it can also assist in determining optimal patrol routes based on historical poaching data. Moreover, the deployment of drones is increasingly popular for the conservation of protected areas in general. Their capability to perform surveillance in a relatively low-cost risk-free manner on a high spatio-temporal resolution with a diverse range of sensors makes them a desirable addition to the tools already in place [6, 7]. In the future, it will be possible to develop truly autonomous surveillance systems by coupling current autopilot capabilities of UAVs with AI-driven image recognition tools and surveillance strategies.

A frequently used framework that focuses on developing solutions for the surveillance planning problem is the formulation of a Green Security Game (GSG) [8], a type of Stackelberg Security Game (SSG). In this format, the interactions between patrollers (defenders) and poachers (attackers)are modelled as a repeated single-shot game. The attacker carries out one or multiple attacks, while simultaneously the defender defends according to a specific strategy. The payoff for the defender depends on where the attacker attacked at that round. Multiple repetitions of this single-shot game, which we consider a single round in an infinite game, allow the defender and the attacker to learn and subsequently adapt their strategies. The resulting defender strategy can be used to define where surveillance should take place on the terrain.

A key characteristic of most AI methods is their dependence on large amounts of data to train their internal models. Next to common problems associated with data sets, like imperfections, incompleteness, and data bias [9, 10], an often overlooked fact is that data is not always available in the first place. Especially when developing models to predict optimal patrol routes for wildlife surveillance, there is no guarantee that specific information is available or will be available in the future. This information is about where which animals are at a certain time, where and how many poachers are active on the terrain, and how many attacks have taken place historically. The absence of this knowledge makes it difficult to determine patrol routes for computers and humans alike.

The majority of previously proposed models require historic attack data and/or a complete attacker model with various defining features, such as a specific behaviour model and full knowledge of the attacker's payoff structure [11–18]. However, it is even recognised that usually the attacker's payoffs are unknown to the defender [19, 20]. Moreover, the defender might not even know its own payoffs due to too much

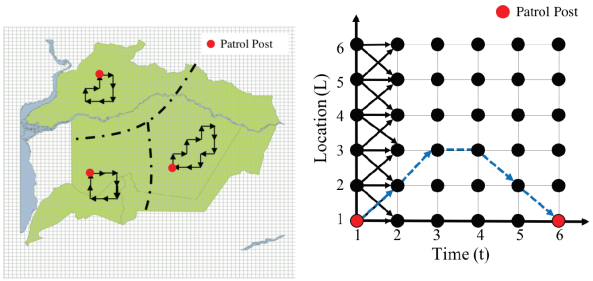

Figure 1. Patrol planning with a time-unrolled graph as by the MINION model [23].

uncertainty in nature. For example, if the payoffs are based on the amount and type of animals at a certain location, they can be random and/or variable, making it difficult to estimate the value of the payoffs.

Recent research by Xu et al. [21] tries to tackle the problems posed by uncertainty by proposing an online learning algorithm to develop a surveillance strategy without prior knowledge. The model, called the Follow the Perturbed Leader with Uniform Exploration (FPL-UE), is an adaptation of the method proposed by Neu and Bartók [22]. It makes no assumptions about adversary behaviour nor defender payoffs while still guaranteeing an efficient theoretical performance. Specifically, it assumes an arbitrary attacker and puts no assumptions on their behaviour or payoff structure. It then chooses between *exploring* (learning which strategy works best) and *exploiting* (maximising utility with gathered knowledge).

the FPL-UE algorithm was further developed by Gholami et al. [23] to take into account spatio-temporal planning constraints proper to patrol rangers, like limited walking time or distance and limited selection of accessible targets. This made the algorithm more applicable for surveillance by foot patrol in the real world. It calculated a feasible patrol route by selecting a starting point at a patrol post and solving an equally distributed time-unrolled graph of adjoining accessible targets on a grid constraint by a specific time horizon (see Figure 1). However, this method is not practical for determining a surveillance flight performed by a drone, since a drone is not bound by flying between adjoining targets. Additionally, Gholami et al. [23] introduces an expert-selection method to evaluate the online learning expert (based on FPL-UE) and a decision tree-based machine learning expert during the game and selecting the best performing expert. However, this approach has its limitations. First of all, since no data is available to train the machine learning expert, a static probability map substitutes as a simulation of the decision tree algorithm's results. Secondly, the expert-selection method proposed by Gholami et al. [23] already starts evaluating both experts at the start of the game. This means that the online learning expert is evaluated before it had the chance to learn.

We propose a novel approach to the patrol planning problem where we take advantage of patrolling with a drone. Continuing on the work by Xu et al. [21], we adopt the same combinatorial adversarial online learning problem formulation to determine a preliminary set of targets for a defender strategy. We formulate the flight path planning as an Orienteering Problem (OP) constrained by the practical limitations of the drone. The solution to the OP results in the final strategy. Inspired by Gholami et al. [23], the model, aptly named the Multi-Expert Online Model for Aerial Patrol Planning (MEOMAPP), also incorporates an expert-selection algorithm that allows evaluating its performance with a second expert. Contrary to Gholami et al. [23], the experts are not evaluated right away, but only after the online expert has had a chance to learn.

In this paper, we present an agent-based model developed for evaluating MEOMAPP's performance. The model represents a simplified wildlife surveillance system, composed of a domain to be surveilled (the *environment*), the drone that performs the surveillance flights (the *defender* agent), and one or multiple poachers (the *attacker* agents). The defender behaves according to the strategies determined by MEOMAPP. We selected two common attacker models to take it up against MEOMAPP: (i) a simple stochastic model with predefined attack probabilities per cell, and (ii) a Quantal Response model [13], which is a state of the art adaptive attacker model with bounded rationality. For a second expert, we assumed a realistic practical scenario where a person familiar with the domain to be surveilled gives every target an attackability score that is used for a probability-based mixed defender strategy.

The question we want to address with this research is whether an FPL-UE algorithm in a multi-expert learning model with a planning method suitable for drones is a viable application for determining wildlife surveillance strategies. To answer this question, we test MEOMAPP using an agent-based model on a real-life wildlife surveillance case. For this case, we develop defender strategies for the Aloegrove Safari Park in Namibia against simulated attackers and with a human expert as a competing strategy method. The test results are inspected for convergence of performance over time and performance variations of MEOMAPP following changes of operational and environmental parameters of the model. The suitability of the model is valid if the convergence behaviour is similar to previous research.

The paper is organised as follows: section II covers additional related work, section III provides a detailed account of the problem formulation, section IV lays out the agent-based model including the specifications of MEOMAPP, section V describes the numerical evaluation by means of the real-world case study. The results are discussed in section VI and we draw a conclusion in section VII. Finally, we present recommendations for future research in section VIII.

II. Related Work

In this section, we address further how our research compares to prior literature regarding GSGs and online learning methods for wildlife surveillance, matheuristics for path planning, and the agent-based modelling and simulation paradigm.

A. *Adversary Modelling in Green Security Games*

It is understood that assuming a perfectly rational, value-maximising adversary is not ideal for addressing human adversaries [24]. Subsequently, two competing approaches have emerged to address human bounded rationality in SSGs and subsequently GSGs. One approach departed from the idea that attackers behave according to specific parametric models of human decision-making of which the parameters could be learned by fitting them to historical data, and subsequently deriving a defender strategy based on the probability where an attacker would attack next. These models include the BRQR algorithm [11] and MATCH [12], based on a Quantal Response (QR) model for adversary behaviour [13]. Based on a new attacker model with an added Subjective Utility function to the QR model (SUQR) [14], algorithms SU-BRQR [14], PAWS [15, 16] and SHARP [17] were developed, followed up by CAPTURE [18]. These models use parameter estimation methods like Maximum Likelihood Estimation (MLE) or Estimation Maximisation (EM) to determine the adversary model's parameters. The underlying data for the estimations comes from real-world experiments or simulations with actual human players.

The other approach is to intentionally avoid adversary modelling and instead focusing directly on reward maximised route optimization based on the individual targets. These methods usually make use of data-driven machine learning techniques. Models like INTERCEPT [25] and others by Gurumurthy et al. [26] and Gholami et al. [27, 28] are based on decision trees that use the target's environmental characteristics and historical attack data to predict the attackability of individual targets. APE, the algorithm by Park et al. [29] uses a variety of classification algorithms in combination with live GPS data of animals and patrol rangers alike to determine real-time dynamic patrol strategies. A black box optimization with neural networks has also been presented by Gurumurthy et al. [26]. These methods all require extensive (historic) data sets, preliminary data manipulation, and extensive knowledge of the terrain.

The recent models proposed by Xu et al. [21] and Gholami et al. [23] also avoid adversary modelling. However, instead of looking at target characteristics for attackability determination, they define a game-theoretic behaviour model for the defender that does not require prior data. The defender does learn an optimal defensive strategy during the game though, regardless of the behaviour the attacker exhibits. This research continues on this specific online approach while avoiding adversary modelling.

B. *Matheuristic Path Planning*

When the question is asked "what is the optimal route for a vehicle given a specific set of constraints?" the resulting problem is always a variant of the Vehicle Routing Problem (VRP) [30]. The case of a single vehicle maximising its reward over a closed route (i.e. returning to the starting point) is known as the Orienteering Problem (OP). The OP, which is NP-hard, is a well-studied problem, and many exact and (meta)heuristic methods have been proposed to solve it [31]. The problem is formulated as an integral problem where a path has to be found on a graph of nodes connected by arcs. All nodes have a certain reward that is collected when the node is visited, and the arcs induce a certain cost when they are part of the route. In the aerial surveillance case, the graph is considered complete, meaning all nodes are interconnected.

Even though the aerial wildlife surveillance problem is presented as a GSG, the planning aspect is considered an OP that has to be solved repeatedly on a graph with rewards that change every round. However, since GSGs assume a defender with limited resources (i.e., it cannot defend all targets in a single round), the possible solution space for the OP is limited. This makes the combination of the defender strategy algorithm and the path planning algorithm in MEOMAPP a method considered a *matheuristic* for solving routing problems [32]. More specifically, it can be classified as a two-phase decomposition approach [33], where the first phase is considered selecting a subset of the nodes in the graph and the second phase is solving the OP on the reduced graph.

This research does not focus on improving solution methods for an OP but it was considered noteworthy that this work is on the intersection between GSGs and operations research.

C. *Agent-Based Modelling and Simulation*

In the majority of referenced literature in this paper, the wildlife surveillance problem is represented as a Multi-Agent System (MAS), wherein GSGs provide a framework to model the agent's interactions. The presence of autonomous actors in the system that interact with each other makes Agent-Based Modelling and Simulation (ABMS) the most appropriate technique to implement and study this model. The ABMS technique enables us to model a natural representation of a system, provide flexibility to modify the system model, and examine emergent outcomes resulting from interactions between autonomous, individual entities with dynamic, adaptive behaviours and heterogeneous characteristics [34]. This is a suitable modelling framework for wildlife conservation in general since models can be specified realistically and dynamically, including changes in environmental conditions and animal movements [35]. This enables them to study externalities related to natural resource management [36].

III. Problem Formulation

In this section, we describe the conceptual formulation of the practical problem of aerial wildlife surveillance. The formulation is similar to the problem formulation by Xu et al. [21] and Gholami et al. [23], as this research aims to extend their proposed solution model.

A. *Game Setup*

The components of the gamified system are the wildlife area that is to be surveilled (the "targets"), the drone that performs the surveillance flights (the "defender"), and the poachers the drone aims to observe (the "attackers"). The entire area is discretized by square grid cells, resulting in set $[N]$ consisting

of N targets. The diagonal of the grid cells is assumed to be the width of the drone camera's Field of View (FoV) in order to always entirely cover a target when choosing it for the route. We assume the drone's height to be constant, ensuring a constant FoV.

The surveillance planning problem is regarded as an infinitely repeated security game between an attacker and a defender. Each round t, m attackers each choose a target to attack. Simultaneously, the defender chooses a surveillance flight to cover k targets specified according to a strategy $v_t \in \{0,1\}^N$. Vector v_t is a binary vector denoting waypoints of the surveillance path by entry $i = 1$ if target i is selected as a waypoint for the flight path. The targets that are observed resulting from this strategy are indicated by binary vector c_t with $|c_t| \geq k$. Targets that are only partially observed due to the nature of the flight path have a chance of being included in c_t equal to the ratio of grid cell area covered by the defender. Similarly to v_t, the attacker strategy is denoted as a_t, where entry $i = 1$ if target i is attacked by the attacker. The path the attacker takes is not taken into account. Also, it is assumed that the attacker remains at the same location for the entire duration of the round. Given that target i is attacked, the defender gets utility U_i^c if target i is covered by the defender, and U_i^u if i is not covered by the defender. It is assumed that covering a target is better than not covering it, which we formalise by stating $U_i^c > U_i^u$.

B. *Information Access and Player Behaviour*

In wildlife surveillance, it is usually unknown to the defender and the attacker what the specific value of a target is as it depends on unknown and/or variable environmental factors and actor-specific preferences. Also, the other players' behaviour is difficult to predict completely, as players can have different knowledge or behave irrationally. Given this information gap, the approach taken for this and previous models is to assume that the defender is unaware of prior information regarding payoffs and attacker behaviour. It can only observe utilities of targets when they are observed. Also, there is no behaviour model of the attacker required for the defender to learn a strategy, since it will adapt to any kind of attacker behaviour. Furthermore, it is required for the attacker that he can only observe the defender when being observed by the defender himself in the current round. That way he is not able to evade the defender during the same round. Finally, we assume a perfect observation from the defender, meaning once the defender covers 100% of an attacked target, the attacker is observed.

C. *Utility and Game Objective*

Given the attacker strategy a_t and the defender observations c_t in round t, the defender's utility at round t is defined as

$$u\left(c_t, a_t\right) = \sum_{i \in N} c_{t,i} a_{t,i} U_i^c + \sum_{i \in N} \left(1 - c_{t,i}\right) a_{t,i} U_i^u \quad (1)$$

where the first term denotes the utility of the covered targets and the second term the utility of the uncovered targets. Both terms are dependent on a_t. The equation can be rewritten as

$$u\left(c_t, a_t\right) = c_t r_t\left(a_t\right) + C\left(a_t\right) \quad (2)$$

with $r_{t,i} = a_{t,i}\left[U_i^c - U_i^u\right]$ and $C\left(a_t\right) = \sum_{i \in [n]} a_{t,i} U_i^u$. This notation helps understanding that the defender's utility at round t is dependent on the attacker's moves during that round. The objective of the security game is to minimise the overall total regret R_T of the defender, defined by

$$\begin{aligned} R_T &= \max_{c \in \mathcal{C}} \sum_{t=1}^{T} u\left(c, a_t\right) - \mathbb{E}\left[\sum_{t=1}^{T} u\left(c_t, a_t\right)\right] \\ &= \max_{c \in \mathcal{C}} \sum_{t=1}^{T} r_t c - \mathbb{E}\left[\sum_{t=1}^{T} r_t \cdot c_t\right] \end{aligned} \quad (3)$$

The first term is the utility of the optimal strategy for round t in hindsight, with $\mathcal{C}$ being the set of all possible observation vectors c. The second term is the expected value of the defender's utility. This notation is consistent with previous online learning theory literature. As noted in Xu et al. [21], the underlying notion of this regret formulation is that it is typically impossible to learn the optimal (adaptive) defender strategy v_t. The reason for this is that the attacker can choose a_t independent from previous actions or even adversarially to the defender. Therefore, the optimal strategy at round t can be independent from history. Without complete knowledge of the attacker behaviour or the environment, there is no way to predict the optimal strategy v_t and it is thus impossible to learn the optimal adaptive strategy.

However, with access to previous observations, it is possible to learn the best strategy in hindsight. The idea behind this problem formulation is that after more and more rounds, the performance of the best strategy in the next round will be affected less and less by a_t, no matter how adversarial (i.e. only caring about minimising the defender's utility) the attacker plays.

IV. Agent-Based Model

The agent-based model forms the framework for the different methods used to solve the formulated problem. These methods and their parameters are represented by the characteristics of the environment and the agents' inputs, internal states, and cognitive models. The representations of the models in Xu et al. [21] and Gholami et al. [23] serve as the baseline of the model formulation. A diagram providing an overview of the agent-based model is given in Figure 2. The following subsections present the properties of the environment and all agents in the model as well as the agents' interactions. Verification of the model is discussed in the last subsection.

A. *Environment Specification*

The environment is defined by a space and time wherein the agents are situated. In this model, one time step equals one round of the GSG, where the attacker attacks one target and the defender performs one surveillance flight.

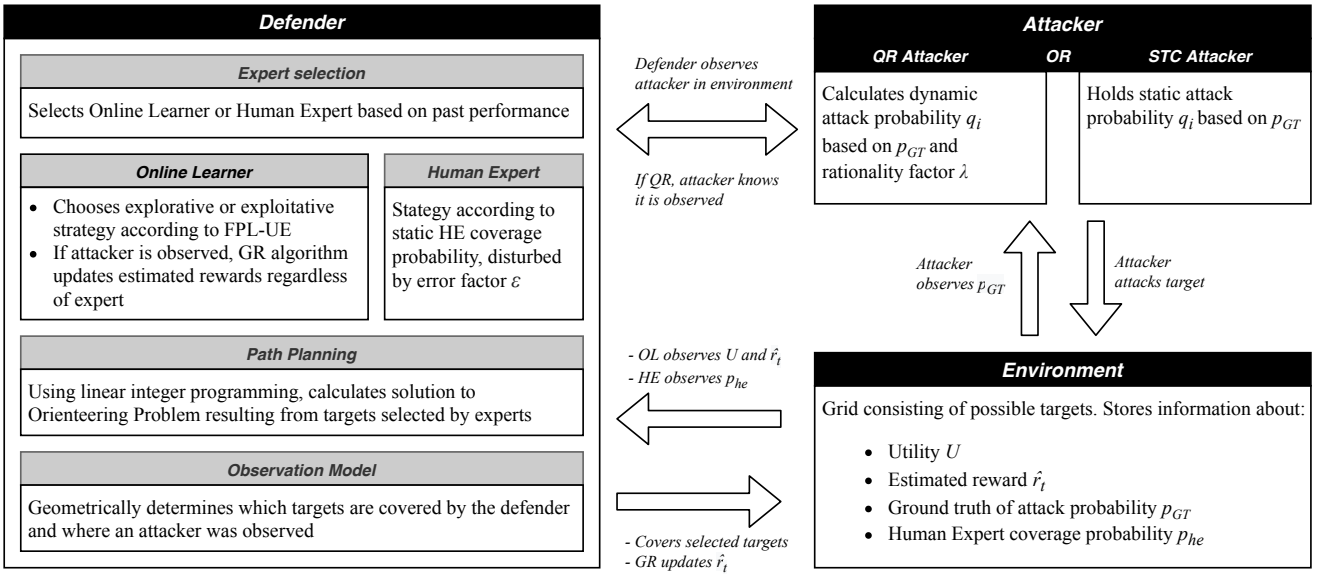

Figure 2. Diagram of the components of the Agent-Based Model and their interactions that occur during one round.

As stated in the previous section, the environment space is discretized by a rectangular grid. The grid cells that result from this representation are all potential targets for an attacker to attack and the defender to defend. The targets i are modelled as patches in the ABM, meaning they are stationary agents. We define $[N]$ to be the set of all targets i. One target is assigned to be the $base$, which means it is the location where the defender will start and end its patrol strategy. When the model is initialised, every target is assigned four numerical characteristics:

1) Utility: Every cell is assigned a utility U_i^c for when it is covered and a utility U_i^u for when it is uncovered. We assume that the utilities are unknown for the defender and that $U_i^c > U_i^u$. For normalisation, we define the values of U_i^c and U_i^u to be within $[-0.5, 0.5]$. This way the maximum regret of the defender per round of the game is at most 1 for each attacker.

2) Estimated reward: The estimated reward $\widehat{r}_{t,i}$ for cell i at round t is initialised as $\widehat{r}_{1,i} = 0$ and indicates the estimated reward of covering that cell in the following step of the game. As explained further down, the estimated reward of a target is incrementally updated by the drone's Online Learner algorithm at every step (see Figure 2), and thus the order of magnitude of the estimations increases gradually throughout the game. It is therefore only compared to the estimated reward of other targets during that same round.

3) Attackability score: The downside of an online learning algorithm without historical data is that it takes time for a model to learn and perform as desired. However, more information about the area to be surveilled is sometimes available in the form of human expertise based on knowledge of the area and the people that live there [26]. We make use of this information in the form of an attackability score. The attackability score is an integer between 0 and 3 assigned to each cell by an expert who is familiar with the environment. It is a positive ratio scale determined with the expert that indicates the likelihood of attacker presence on a specific cell. We convert this score to an attack probability by modifying the standard normalised exponential function (left in Equation 4) to a version that takes into account the variation of discretization (right).

$$\sigma(\mathbf{z})_i = \frac{e^{\beta z_i}}{\sum_{j=1}^{K} e^{\beta z_j}} \Longrightarrow p(\mathbf{a})_{GT,i} = \frac{\sqrt{N}^{a_i}}{\sum_{j\in[N]} \sqrt{N}^{a_j}} \quad (4)$$

R. D. Luce [37] first used the normalised exponential function in decision theory for relative preferences in his Choice Axiom. It has seen multiple applications in psychology [38] and game theory [39, 40] relating to human choices and utility representation [41]. The general idea behind it is that humans have an initial intuition to map numbers onto space logarithmically. We propose setting $\beta = \log\left(\sqrt{N}\right)$, which is equivalent to using N (total number of targets) as base of the exponentiation instead of e. This concentrates the probability distribution more around the positions of the largest input values. Taking a base that varies according to the number of cells also prohibits the higher probabilities to dilute when the grid is discretized further. We chose to select $\sqrt{N}$ instead of N since humans' tendency to logarithmically project numerals is based on a linear distribution. Therefore we linearize the increase of N, which exhibits quadratic growth when we linearly increase the dimensions of the grid. The resulting probability distribution p_{GT} is the ground truth for the behaviour models for attackers, further explained in Section IV-C, and for the defence strategies of the Human

Expert (HE), explained hereafter.

4) Human Expert Coverage Probability: The MEOMAPP algorithm incorporates a self-evaluation procedure for the defender where it compares the performance of an online learning expert with the performance of a human expert (HE). To simulate the performance of a human expert defence strategy, we propose the following. The human expert strategy consists of a set S_t of n targets selected based on the probability distribution p_{he}. This distribution is used in the defender's path planning algorithm to calculate a patrol route. To represent the different levels of inaccuracy of the human expert, we approximate a probability distribution p_{he} with a certain mean absolute error (MAE) as follows:

$$p_{he,i}(\varepsilon) = p_{GT,i} \pm \epsilon \quad , \quad \epsilon \sim \mathcal{N}(\varepsilon, \varepsilon/4) \tag{5}$$

with a resulting MAE of the probability distribution approximately equal to ε. Error factor ϵ is drawn from a normal distribution $\mathcal{N}(\varepsilon, \varepsilon/4)$ with mean ε and variance $\varepsilon/4$. The $+$ or $-$ is as such that $p_{he,i} \in [0, 1]$, with a random choice if both options are viable.

B. Defender Specification

The defender is a reactive although complex agent. It is characterised by the range R or the total distance it can travel in a single round. The defender always departs and returns to the $base$ target, Furthermore, it incorporates four cognitive models: *1)* an online learning game-theoretic expert (OL) that calculates the reaction to the attacker's moves, *2)* a mathematical integer linear program to calculate a spatially optimised patrol route within its planning constraints, *3)* an observation model, and *4)* an expert selection algorithm that chooses between a human expert and the OL expert as the preferred choice for next round's strategy.
A theoretical argument for the defender's parameter values is presented at the end of this subsection.

1) Online Learner: The online learning algorithm presented in Algorithm 1 is capable of generating a defender strategy without any prior knowledge. It is based on the FPL-UE algorithm proposed by Xu et al. [21], wherein a "Follow the Perturbed Leader" and a "Uniform Exploration" element can be distinguished.
The FPL element evaluates the perturbed estimated reward $\widetilde{r}_{t,i}$ for each target i at round t. Let $\widehat{r}_t$ be the vector of the pure estimate rewards at round t, and let $z_t = (z_{t,1}, \ldots, z_{t,n})$ be a random noise vector such that each $z_{t,i} \sim exp(\eta)$ is independently drawn from the exponential distribution $exp(\eta)$. At each round, the algorithm then chooses n targets, from all targets $[N]$, with the highest perturbed estimated reward $\widetilde{r}_{t,i} = \widehat{r}_{t,i} + z_{t,i}$, as formulated by Equation 6. This set is called S_t.

$$S_t = \underset{S \subset [N]}{\arg\max} \left\{ \sum\nolimits_{i \in S} \widetilde{r}_{t,i} \;\middle|\; |S| = n \right\} \tag{6}$$

In this case, the noise vector $z_{t,i}$ represents the uncertainty of the reward estimation and thereof dependent target selection. It also results in unique estimated reward values for every target, which prohibit that the MILP solver selects identical sequences of targets (especially in the first rounds when most targets have $\widehat{r}_{t,i} = 0$). The FPL element is the *exploitative* element of the defender strategy.

The UE element, which is the *explorative* element of the defender strategy, also selects n targets to form S_t but does it randomly and uniformly. For the randomly selected targets in S_t, a noise vector z_t is drawn independently from $exp(\eta)$ as well for the same reasons as for the exploitative strategy. Since these targets were selected regardless of the value of their estimated reward, the perturbed estimated reward is set to equal the noise factor: $\widetilde{r}_{t,i} = z_{t,i}$. In every round, taking a random explorative step happens with probability γ, resulting in a complementary probability $(1-\gamma)$ to pursue an exploitative strategy. The goal of the exploitative element is to maximise the total utility over time, whereas the goal of the explorative element is to learn which strategy is the best against a particular attacker.

Up to now, the online learner follows the FPL-UE algorithm to determine a set of nodes S_t at every round by either following an exploitative or an explorative strategy. The set S_t is used by the MILP to calculate the flight path strategy v_t (as described in section IV-B2), where the targets in S_t are potential waypoints for the flight path. Once flight path strategy v_t is determined and applied at round t, we define set O_t as the targets where the defender observed an attack when executing the strategy during round t.

Knowing this, we can now update the estimated reward values for the next round $\widehat{r}_{t+1,i}$ as follows:

$$\widehat{r}_{t+1,i} = \widehat{r}_{t,i} + \tfrac{r_{t,i}}{p_{t,i}} \mathbb{I}(t,i) \quad \forall i \in O_t \tag{7}$$

where $p_{t,i}$ is the probability that target i was observed by the defender within round t and $\mathbb{I}(t,i)$ is an indicator function indicating if a target was observed by the defender, with $\mathbb{I}(t,i) = 1$ if target i was observed and $\mathbb{I}(t,i) = 0$ otherwise. In online literature, the term $\frac{r_{t,i}}{p_{t,i}}\mathbb{I}(t,i)$ is preferred over directly using $r_{t,i}\mathbb{I}(t,i)$, since it is an unbiased estimator of $r_{t,i}$: $\mathbb{E}\left[\frac{r_{t,i}}{p_{t,i}}\mathbb{I}(t,i)\right] = r_{t,i}$. Note that this corresponds to updating the estimated reward for targets that were attacked *and* defended, and keeping the estimated reward for the other targets the same in the next round.

To efficiently estimate $p_{t,i}$, which is unknown and hard compute exactly, Neu and Bartók [22] proposed a method to calculate the value of $1/p_{t,i}$ called Geometric Re-sampling (GR), presented in Algorithm 2. The method works by simulating defender strategies until the targets that were attacked and defended in round t are defended again by the simulated round. The number of simulations required until target i is defended for the first time follows a geometric distribution with mean $1/p_{t,i}$. The probability of observation $p_{t,i}$ is thus estimated by the number of simulations it requires to defend target i. However, theoretically, the number of

simulations can be infinitely large, so the GR algorithm truncates the number of simulations with a finite quantity M.

Algorithm 1 Online Learner

Parameters: $\gamma \in [0,1], n \in \mathbb{N}, \eta \in \mathbb{R}^+, M \in \mathbb{Z}^+$

1: **for** t = 1, ..., T **do**
2: Sample $\alpha \in 0, 1$ such that $\alpha = 0$ with prob. γ
3: **if** $\alpha = 0$ **then**
4: Let $S_t \subset [N]$ be a set of n randomly selected targets
5: Draw $\widetilde{r}_{t,i} \sim exp(\eta)$ independently for all $i \in S_t$
6: **else**
7: Draw $z_{t,i} \sim exp(\eta)$ independently for all $i \in [N]$
8: Set $\widetilde{r}_{t,i} \leftarrow \widehat{r}_{t,i} + z_{t,i}$
9: Let $S_t \subset [N]$ be the set of n targets with $max(\widetilde{r}_{t,i})$
10: **end if**
11: Let v_t be $\mathcal{P}(S_t)$
12: Adversary picks $r_t \in [0,1]^n$ and defender plays v_t
13: Defender observes O_t
14: Run GR$(\eta, M, \widehat{r}, t)$: estimate $\frac{1}{p_{t,i}}$ as $K(t,i)$
15: Update $\widehat{r}_i \leftarrow \widehat{r}_i + K(t,i) r_{t,i}$;
16: **end for**

Algorithm 2 Geometric Resampling

Input: $\eta \in \mathbb{R}^+, M \in \mathbb{Z}^+, \widehat{r} \in \mathbb{R}^n, t \in \mathbb{N}$
Output: $K(t) := \{K(t,1), \ldots, K(t,n)\} \in \mathbb{Z}^n$

1: Initialize $\forall i \in [N] : K(t,i) = 0; k = 1$
2: **for** $s = 1, 2, \ldots, M$ **do**
3: Repeat lines 2 - 13 in alg. 1 once to produce $\widetilde{O}$ as a simulation of O_t with a_t.
4: **for all** $i \in O_t$ **do**
5: **if** $s < M$ **and** $i \in \widetilde{O}$ **and** $K(t,i) = 0$ **then**
6: Set $K(t,i) = s$;
7: **else if** $s = M$ **and** $K(t,i) = 0$ **then**
8: Set $K(t,i) = M$;
9: **end if**
10: **end for**
11: **if** K(t, i) > 0 for all $i \in O_t$ **then** break
12: **end for**

2) Path Planning: The patrolling strategy v_t is calculated as a solution to a symmetric Orienteering Problem, formulated by the mathematical program $\mathcal{P}(S)$ in Equation 8. It is an integer linear programming problem applied to targets i in S_t, which represent the nodes of a network where the perturbed estimated rewards $\widetilde{r}_{t,i}$ represent the profits collected if a node is visited. The network's edges are defined as the arcs $a_{i,j}$ between targets i and j, with a length of $d_{i,j}$. Equation 8.a limits the total distance travelled by the defender to its range R. To every target the defender goes, it also has to leave from, which is constrained by Equation 8.b. A target can only be visited once, meaning only two arcs can connect to it. This is constrained by Equation 8.c. Constraint 8.d ensures that no subtours (i.e., tours that are not part of the tour that includes the base) are included in the solution. If an arc $a_{i,j}$ is selected, the difference between u_i and u_j is exactly -1 if arc $a_{i,j}$ is selected in the strategy, where u_i and u_j are the orders at which the targets are visited. Subscript $base$ indicates the target where the defender starts and ends its round. that target's inclusion in the path is ensured by constraints 8.e and 8.f. These are not strictly speaking necessary when Equation 8.d is applied (where the $base$ is the only target left that can "close" the loop), but they increase the computational performance.

$$v_t = \arg\max_{v \in \mathcal{V}} \sum_{i,j \in S_t} a_{i,j} \widetilde{r}_i \tag{8}$$

subject to

$$\sum_{i,j \in S_t} a_{i,j} d_{i,j} \leq R \qquad i \neq j \quad \text{(a)}$$

$$\sum_{i \in S_t} a_{i,j} = a_{j,i} \qquad \forall j \in S_t; i \neq j \quad \text{(b)}$$

$$a_{i,j} + a_{j,i} \leq 1 \qquad \forall i,j \in S_t; i \neq j \quad \text{(c)}$$

$$u_i - u_j \leq |S_t| (1 - a_{i,j}) - 1 \qquad \forall i,j \in S_t; i \neq j \neq base \quad \text{(d)}$$

$$\sum_{i \in S_t} a_{i,base} = 1 \qquad i \neq base \quad \text{(e)}$$

$$\sum_{i \in S_t} a_{base,i} = 1 \qquad i \neq base \quad \text{(f)}$$

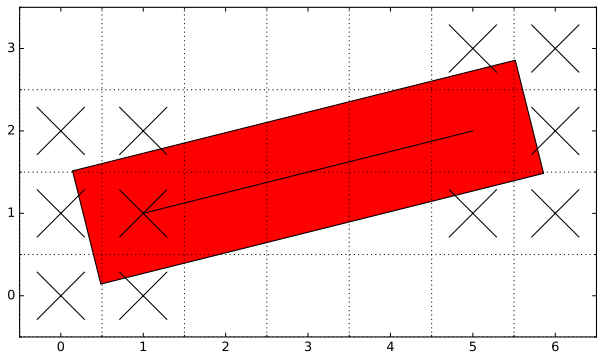

Figure 3. Coverage between target $(1,1)$ and $(5,2)$.

3) Observation Model: After v_t has been calculated by the path planning algorithm, it is necessary to determine which targets are covered and whether an attacker has been observed. Based on the assumption that the defender's view is as wide as the diagonal of a cell, we draw a rectangle of width $w = \sqrt{2l^2}$, where l is the width of a target, and length $d_{i,j} + w$, where d_{ij} is the distance between targets i and j, so that arc $a_{i,j}$ coincides with the longest centerline of the rectangle. Afterwards, for every target that has an overlap with this rectangle, the fraction $frac_i$ of the observed area of the target over the total area of the target is calculated. The exemptions to this are the starting point i and the targets right next to i and j that are not in the line of the path, in order to prevent them to be counted twice. As an example, a schematic of coverage between target $(1,1)$ and $(5,2)$ is shown in Figure 3, where $frac_i > 0$ for targets $(2,0), (2,1), (2,2), (3,1), (3,2), (4,1), (4,2), (4,3)$ and $(5,2)$.

This is done for all arcs $a_{i,j}$ that are part of the solution calculated by the path planning algorithm in Equation 8, resulting in a set C_t consisting of targets i that have been (partially) covered by the defender at round t.
The probability that an attacker is observed by the defender on target i, if an attacker was on that target, is equal to the fraction of the area observed. In other words, target i is considered to be in set O_t (targets where an attack has been observed at round t) following the next equation:

$$P(i \in O_t | i \in C_t \cap A_t) = frac_i \tag{9}$$

where A_t is the set of targets i that are attacked.

4) Expert Selection: The following expert selection algorithm is proposed to enable MEOMAPP to select the best performing expert to decide on a strategy. Formulation wise, it is wrapped around the core algorithms 1 and 2 as presented in algorithm 3. First, a constant θ is defined which is compared to the value of variable γ. Only after the value of γ drops below threshold θ (line 4) we consider the online learning algorithm to have learned enough to compare it to the HE. We chose γ as a threshold measure because it depends on k and m as well, which allows us to maintain the same value for θ throughout different game settings.
To compare both the experts' performance, cumulative reward values r_{ol} and R_{he} are initialised as 0 and updated after every round t. The performance is then evaluated by comparing the average reward over the number of rounds where that expert has been selected, n_{ol} and n_{he}. If the human expert is chosen to be the best expert, S_t is determined by p_{he} as presented in Section IV-A3. If the online learning expert is selected, S_t is determined by Algorithm 1. The patrol strategy v_t is then determined by $\mathcal{P}(S_t)$ resulting in observed attacked targets O_t and the collected rewards are updated accordingly for each expert. Note that until γ reaches the threshold θ the performance of the human expert, which has not been evaluated by then, is synchronised with the performance of the online learning expert (lines 20 - 22).
The idea behind this expert selection algorithm is that given the online learner expert has learned sufficiently, the algorithm will be able to decide whether it is more successful to follow the human expert or the online expert. In this case, it will differentiate between the human expert, whose performance results from the potentially imperfect interpretation of the environment and/or historical data, and the online learner without any prior knowledge.

Defender's parameter values: From the theoretical properties of the FPL-UE algorithm stated by Xu et al. [21], we know that the total regret R_T (i.e., the difference between the performance of FPL-UE and that of the best fixed patrol path in hindsight) is proven to be upper bounded as:

$$R_T \leq \gamma m T + 2Tke^{-M\frac{\gamma}{[N]}} + \frac{k(\log N + 1)}{\eta} + \eta m T \min(m, k) \tag{10}$$

Algorithm 3 The MEOMAPP Algorithm

Parameters: $\gamma \in [0,1], n \in \mathbb{N}, \eta \in \mathbb{R}^+, M \in \mathbb{Z}^+, \theta \in \mathbb{R}^+$
1: Initialise $\widehat{r} = 0, r_{ol} = 0, r_{he} = 0, n_{ol} = 0, n_{he} = 0$
2: Pick a value θ as a threshold for γ for which the Online Learner is considered good enough;
3: **for** $t = 1, \ldots, T$ **do**
4: **if** $\gamma \geq \theta$ **or** $\frac{r_{ol}}{n_{ol}} > \frac{r_{he}}{n_{he}}$ **then**
5: Let S_t be computed by lines 2 - 10 in alg. 1;
6: $n_{ol} \leftarrow n_{ol} + 1$
7: $f = 0$
8: **else**
9: Let S_t be determined by the HE where $\widetilde{r}_{t,i} = p_{he,i}$;
10: $n_{he} \leftarrow n_{he} + 1$
11: $f = 1$
12: **end if**
13: Let v_t be $\mathcal{P}(S_t)$;
14: Adversary picks $r_t \in [0,1]^n$ and defender plays v_t;
15: Defender observes attackers at O_t;
16: **for** $i \in O_t$ **do**
17: $r_{he} \leftarrow r_{he} + f r_i$
18: $r_{ol} \leftarrow r_{ol} + (1-f) r_i$
19: **end for**
20: **if** $\gamma \geq \theta$ **then**
21: $n_{he} \leftarrow n_{he} + 1$
22: $r_{he} \leftarrow r_{ol}$
23: **end if**
24: Run GR$(\eta, M, \widehat{r}, t)$: estimate $\frac{1}{p_{t,i}}$ as $K(t,i)$;
25: Update $\widehat{r}_i \leftarrow \widehat{r}_i + K(t,i) r_{t,i}$;
26: **end for**

where upper bound $\mathcal{O}(\sqrt{kmT \min\{m,k\} \log N})$ can be obtained by taking $\eta = \sqrt{\frac{k(\log N+1)}{mT \min\{m,k\}}}$, $\gamma = \frac{\sqrt{k}}{\sqrt{mT}}$ and $M = N\sqrt{\frac{mT}{k}} \log(Tk)$. This means that the values of η, γ and M depend on the total number of targets (N), the number of attackers (m), the number of intentionally protected targets (k), and the number of rounds T that have passed, the contribution of the latter resulting in a gradual decline of the values of η and γ over time. This can be interpreted as a decline in uncertainty because of the decline in noise z_t and the lower probability of engaging in an explorative strategy respectively.
The number of intentionally protected targets k is the number of targets that are selected as waypoints in the flight path $|v_t|$. This value is not known beforehand, but since the first round is explorative regardless, any number larger than 0 can be chosen as an initial value. After the first round, k is calculated as the average value of $|v_t|$ over time.
In theory, the value of M, the maximum number of simulations in the GR algorithm, is very high relative to T and can result in extremely long running times. For example, for a 500-step simulation with $N = 100, m = 5$ and $k = 15$, the GR algorithm runs up to $M = 11,519$ times in the worst-case scenario. However, as Neu and Bartók [22] theorise a lower expected number of samples in practice, we limit the maximum amount of GR simulations to 100. During the experiments, this number of samples was never reached.

C. Attacker Specification

To simulate attacker behaviour we chose two commonly seen behaviour models: Stochastic behaviour (STC) and Quantal Response behaviour (QR) [14].

The STC attacker model chooses to attack target i based on a stationary attack probability q_i per target i. We assume this probability to be equal to the ground truth attack probability $p_{GT,i}$ of target i presented as the basis for the human expert in Section IV-A3, as this is the best information available.

$$q_i = p_{GT,i} \tag{11}$$

The QR attacker model simulates non-stationary attacker behaviour by observing and responding to the defender strategy. The probability that an attacker will attack target i at round t is given by:

$$q_i = \frac{e^{\lambda U_i^a}}{\sum_{j\in[N]} e^{\lambda U_j^a}} = \frac{e^{\lambda(x_i P_i^a + (1-x_i)R_i^a)}}{\sum_{j\in[N]} e^{\lambda(x_j P_j^a + (1-x_j)R_j^a)}} \tag{12}$$

Parameter $\lambda \in [0, \infty]$ represents the rationality level of the attacker. A lower value for λ indicates lower rationality (resulting in a more uniform q_i) and a higher value indicates higher rationality (resulting in a reward maximising q_i). Parameter $U_i^a = x_i P_i^a + (1 - x_i)\, R_i^a$ is the attacker's expected utility for target i. It depends on the likelihood the target is defended x_i, and on the reward R_i^a and penalty P_i^a for the attacker associated with the target. R_i^a is obtained by normalising the GT probability p_{GT} to a value between $[0, 10]$ in accordance with QR experiment settings in previous literature [11]. We assume penalty $P_i^a = -10$ for all targets since getting caught by the defender is the worst that can happen on any target.

The likelihood x_i that target is defended is not the same as probability p_i estimated by the GR algorithm, but a likelihood calculated by the attacker based on how often it is caught on target i. Previous literature does not explain how x_i is calculated. We propose to initialise $x_i = 0$ and to recalculate it at every round t as:

$$x_i = \frac{\sum_{t=1}^{T} a_{t,i}\mathbb{I}(t,i)}{T} \qquad \forall i \in [N] \tag{13}$$

where we note that this only holds because we make the additional assumptions: (i) The attacker is not initially aware of the fact that the area is under surveillance. (ii) The attacker does not know where or when the defender has been if the attacker was not observed by the defender. (iii) If an attacker was observed in any previous round, all attackers know about this in the next round.

For both models, after the probability q_i of the attacker choosing a target i in round t is determined, one target is picked per attacker based on that likelihood. Furthermore, it is important to note that the attacker is modelled to visit only one target in a single round, meaning no route to and from the target are simulated.

The fact that p_{GT} is used as a foundation to calculate q_i for both attacker models as well as the coverage preference p_{he} of the HE means that the accuracy ε of the HE relates to how good the human expert is in assessing the attacker behaviour.

D. Validation and Verification

As for agent-based models in general, the bottom-up nature of the building process of MEOMAPP led to validation being applied during the model construction itself [42]. To assure the validity of the model as a whole, every model component was validated individually, including its output of information to other components. The goal of this research is to evaluate the application of an online learning defender strategy for GSG in a simulated environment to estimate its performance in a realistic scenario. The representation of the environment, the attackers, and the defender in the agent-based model has been performed with the support of wildlife surveillance experts whose contribution ensured further validity of the model. It is important to note that a higher validity could be obtained for more specific cases depending on the information available. The ultimate test to validate the model would be to perform experiments in the real environment.

Verification of the model was performed at different levels. At the code level, compiler errors were resolved within Spyder, the integrated development environment (IDE) chosen for this research. At the unit level, error-oriented testing [43] has been performed by plausibility tests [44] on parameter values and on results of intermediate computation mechanisms. Attention has also been paid to avoid issues arising from floating-point arithmetic within the computations. At the system level, conceptual verification was performed by observing whether the results matched expectations regarding convergence.

V. Numerical Evaluation

This section describes how we implemented the proposed model, the real-world case we simulated, the simulation setup, the derivation of the model's parameter values for this specific case, and the results from the simulations.

A. Algorithm Implementation

The simulation model was written in Python using Mesa, a Python-specific agent-based modelling framework [45]. The code is written on compliance with the PEP 8 style guide for Python code [46]. It is available in the Delft University of Technology Gitlab repository. For the ILP component, the open-source module PuLP [47] was used to generate the problem file. To solve the ILP problem, the Gurobi™[48] solver was called using an academic licence. It is however interchangeable with open-source solvers readily available in the PuLP toolkit. All simulations that are discussed later have been run using a machine with a 1.2 GHz Intel® Core™ i7-3610QM CPU with 7 available cores and 12GB RAM. The runtime for the presented experiments can be found in Table III.

B. Case Study

To examine the performance of MEOMAPP, we selected a case together with industry experts to simulate wildlife surveillance in a real-world setting. The domain to surveil is the Aloegrove Safari Park in Namibia and is approximately 10 by 10 km in size (see Figure 4). The attackability values

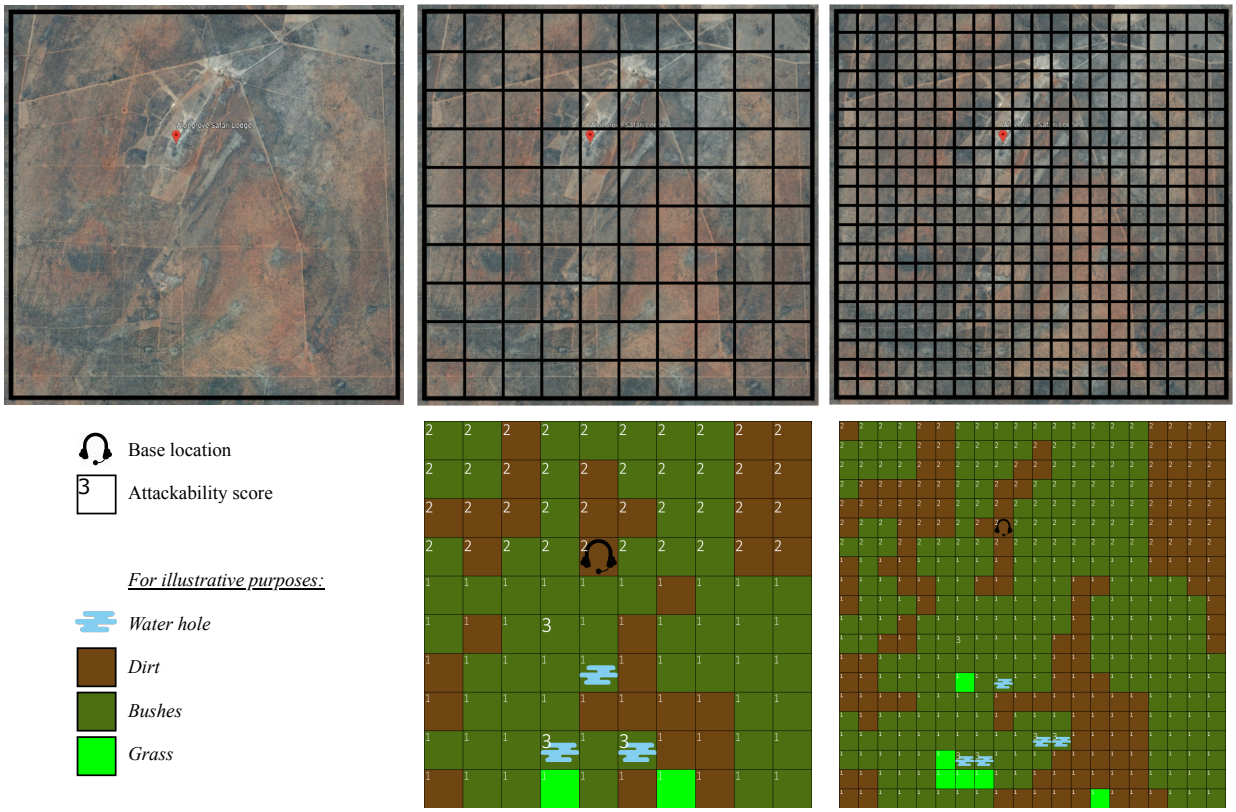

Figure 4. Schematic of Aloegrove Safari Park and its implementation into MEOMAPP *(image from Google Earth - CNES/Airbus Maxar Technologies).*

have been assigned by an expert familiar with the domain. The expert based the attackability values on his interpretation of the accessibility of the terrain, location of fences or barrier, and the location of animal enclosures or water holes. The colour scheme on the map is purely for illustrative purposes. Contrary to simulations in previous research where the values for utilities U_i^c and U_i^u are randomly selected, we chose to define $U_i^c = 0.5$ and $U_i^u = -0.5$. The reasoning behind this is that we consider the utility to be a result primarily of whether a poacher has been observed or not. This is regardless of the damage the poacher has or could have done. Also, defining the utility values specifically eliminates a random factor from the regret equation, which makes the results less prone to random variations.

The drone range is 25 km and we chose to have the algorithm select 20 possible waypoints for a surveillance flight path at every round ($n = 20$). It is not important to determine when or how long a surveillance flight takes place, as long as we assume that the attacker(s) remain(s) at the attacked location(s) for the duration of the surveillance flight. For a grid discretization of 10 by 10 cells we assume a field of view of the drone of approximately 1.4 km wide. For a grid discretization of 20 by 20 cells, the field of view is considered 0.7 km wide, which in practice means that the drone flies lower with a lower resolution camera and therefore has a reduced field of view.

C. *Key Performance Indicators*

To evaluate MEOMAPP's performance, we look at the following three indicators. These are the average regret over time, the average distribution of employed strategies, and the average distribution of observed attacks.

As in previous research, the overall performance is measured by the average regret over time R_T/T. Since we chose to limit the values of U_i^c and U_i^u to 0.5 and -0.5 respectively, the regret value directly relates to the number of targets attacked. Since every attacker attacks one target every round, the regret can be at most m. and at least 0. Experiment results expressed as regret from previous research [21, 23] cannot be used to directly compare MEOMAPP's regret values for two reasons. First of all, information about their specific payoff structure and attacker behaviour is incomplete. Without that information, it is not possible to reproduce their experiments. Secondly, we chose a specific utility structure with extreme values, which will produce relatively higher regret values for any experiment setup. Additionally, the grid discretization used by Gholami et al. [23] was only 5x5 cells, which we deem insufficient for real-world approximations.

Furthermore, we look at the distribution of employed strategies: explorative, exploitative, or defined by the human expert. This indicates which expert is superior in which situation, and gives an insight into whether the expert selection algorithm works.

Finally, in order to evaluate the accuracy of MEOMAPP we look at the distribution of observed attacks that took place on intentionally visited targets (the waypoints of routes selected by the HE and the exploitative OL) and on coincidentally visited targets (explorative routes and the targets between waypoints).

D. Simulation Setup

The parameter space can be varied in seven dimensions:

1) Grid size/discretization
2) ε of the HE
3) Attacker type (STC or QR with different λ)
4) Number of attackers m
5) Number of possible waypoints n
6) Expert selection parameter θ
7) Drone range R

Since it is impractical to evaluate all possible combinations, we select a base setting with parameters that remain constant and manipulate the remaining parameters to deduce their impact on MEOMAPP's performance.

The parameters that remain constant throughout all experiments are n and θ. The number of possible waypoints $n = 20$ is an arbitrary choice. The logic behind it is that it should not be too large for 1) computational reasons and 2) a twisty trajectory which is impractical for the drone. It should also not be too small in order to evaluate enough points of interest for the surveillance strategy. After some iterations, we chose $n = 20$ as it satisfied both accounts. Variations in n and the planning algorithm in general, are included as recommendations for future research.

For the baseline model we propose the following: The attacker types are an input that would not be required in a real-world setting. Therefore we decide to test the performance with an STC attacker as an example of stationary behaviour, and with one QR attacker with a specific λ as an example of adaptive non-stationary behaviour. To choose a value for λ, ten games with one QR attacker's λ value ranging from 0.1 to 1.0 with 0.1 increments were simulated without the HE. The 0.1 to 1.0 range was chosen to include λ values found and used in previous research by Nguyen et al. [14] and Gholami et al. [23]. Additionally, one game with an STC attacker was simulated as well. Each simulation lasted 500 steps and the results are presented in Table I. It can be observed that the final average regret is inversely related to the value of λ. This means that the performance of MEOMAPP is better the more rational the attacker is.

Table I
AVERAGE R_T FOR SIMULATIONS WITH $N = 100$, $m = 1$, $T = 500$ FOR VARYING λ OF QR ADVERSARY AGAINST OL EXPERT.

λ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	STC
R_T	0.70	0.66	0.60	0.52	0.48	0.43	0.39	0.35	0.35	0.30	0.50

Also, the OL's performance against the QR attacker with $\lambda = 0.4$ is similar to the performance against the STC attacker. To confirm this similarity, four additional simulations were performed for both attacker types. The results of the additional simulations, shown in Table II, indicate that the performance is comparable. We therefore decided to use a QR attacker with $\lambda = 0.4$ as the basis for the remaining experiments for to reasons. If any changes in the total average regret occur for those experiments, the difference in attacker type has less influence on the result. Furthermore, the resulting average regret value of 0.52 allows observing changes in the average regret value in both directions when the other parameters of the model are evaluated.

Table II
AVERAGE R_T FOR SIMULATIONS WITH $N = 100$, $m = 1$, $T = 500$ FOR QR ATTACKER WITH $\lambda = 0.4$ AND STC ATTACKER AGAINST OL EXPERT.

Attacker	1	2	3	4	5	Average R_T
QR, $\lambda = 0.4$	0,52	0,51	0,54	0,50	0,54	0,52
STC	0,50	0,54	0,53	0,51	0,50	0,52

The expert selection threshold θ is determined by inspecting preliminary results from testing the OL model alone against QR modelled attackers. In a simulation setup with $m = 1$ and $n = 20$, the number of waypoints for a flight path is on average 16, meaning $\gamma = 0.4$ at $t = 100$. The 100 step mark was chosen because the value to which the average regret converges was reached at step 100 already.

We want to examine the effect of changing the remaining five parameters through five experiments. **Experiment 1** establishes the results for the baseline settings that serve as a reference for the remaining experiments. The differences in resulting defender behaviour against the two different attacker types are of interest as well, so the baseline is established for the QR and the STC attacker. **Experiment 2** evaluates the effect of range variations by setting $R = 15$ and $R = 35$. The range is an important feature when choosing a suitable drone in a real-world situation. **Experiment 3** evaluates the difference in performance of having a perfect HE (i.e. with $\varepsilon = 0$). It aims to uncover the performance of the OL against an expert with more precise knowledge of the attacker behaviour. **Experiment 4** evaluates the impact of having more attackers on the domain by simulating the game with 3 and 5 attackers. **Experiment 5** evaluates the effect of discretizing the terrain with a grid that is twice as fine, i.e. 20x20 cells versus 10x10 cells. Given the relationship between the defender's FoV and the grid size, it is important to analyse the effect of a different grid discretization.

Based on the preliminary experiments that have been performed for establishing the base model, we can visually determine that a simulation duration of 500 steps is sufficient for the average regret value to stabilise.

E. Results

In this subsection, the results for every simulation setup discussed above are presented. All simulation settings and the resulting values of the KPI's at $T = 500$ are summarised in

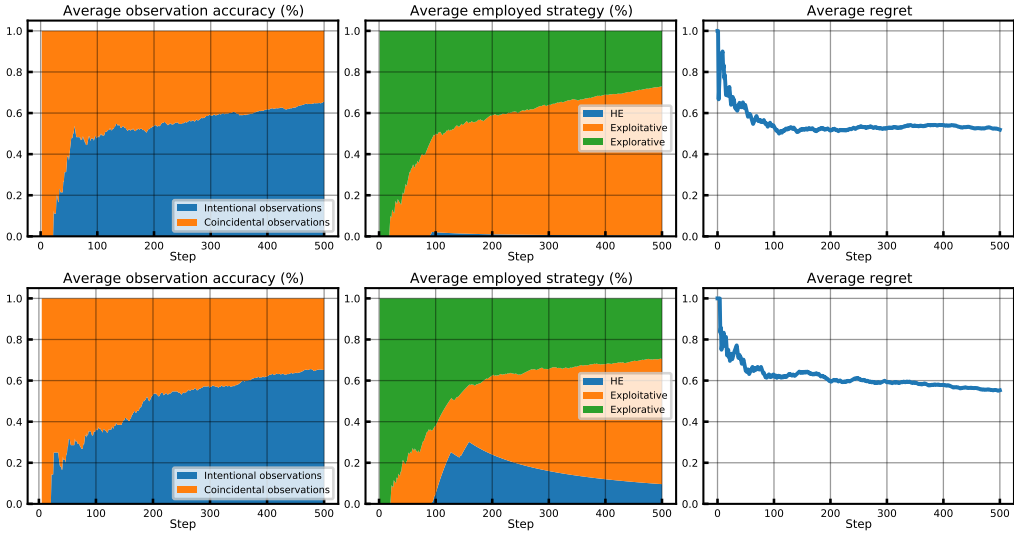

Figure 5. Simulation results from the baseline model with QR adversary (top) and STC adversary (bottom).

Table III
SUMMARY OF THE SIMULATION RESULTS WITH $T = 500$.

Att. type	R	ε	m	N	***Final av.*** R_T	***Final av.*** R_T/m	***Final accuracy***	***Final strategy***	***Runtime (min.)***
	25	**0,3**	**1**	**100**	**0,52**	**0,52**	0,65	OL	189
	15	0,3	1	100	0,63	0,63	0,75	OL	67
	35	0,3	1	100	0,44	0,44	0,48	OL	110
QR	25	**0**	1	100	0,46	0,46	0,71	HE	148
	25	0,3	**3**	100	1,77	0,59	0,58	OL	192
	25	0,3	**5**	100	3,17	0,63	0,54	OL	193
	25	0,3	1	**400**	0,76	0,76	0,52	OL	561
	25	**0,3**	**1**	**100**	**0,55**	**0,55**	**0,65**	OL	120
	15	0,3	1	100	0,72	0,72	0,74	OL	79
	35	0,3	1	100	0,43	0,43	0,63	OL	95
STC	25	**0**	1	100	0,54	0,54	0,69	OL	193
	25	0,3	**3**	100	1,58	0,53	0,71	OL	187
	25	0,3	**5**	100	2,63	0,53	0,69	OL	247
	25	0,3	1	**400**	0,80	0,80	0,23	HE	544

Table III. For every experiment, the values that are discussed are also highlighted in a figure. All other plots can be found in the report accompanying this paper [49].
We emphasise that, even though all these figures are the convergence plots for one randomly generated game instance, the general convergence trends of the KPI's is almost the same across the simulated instances for every setup. However, the initial rounds in the figures may vary among different instances.

Experiment 1 - baseline establishment: The baseline model shows a similar trend in observation accuracy after $t = 200$ for the QR and STC attacker in Figure 5. Before $t = 200$ however, the observation accuracy against the STC attacker was significantly worse. This also translates to a higher average regret in the first stage of the game. As a result, the convergence of the average regret is slower against the STC attacker. For this particular simulation, it can be observed that the HE was briefly superior in the average distribution of defender strategies, but after 59 steps (i.e. 59 steps after step 100) the OL expert was trained enough to outperform it.

Experiment 2 - range variation: Comparing the results in Figure 6 with the results for the QR attacker in Figure 5, we can deduce the following. When the drone range is reduced from 25 to 15 km, we observe a slower convergence of the regret value. Furthermore, from Table III it is clear that the final regret value is higher when the defender has a lower range, meaning decreasing the range results in a decrease in performance. However, we can also observe that the accuracy of the attack observations increases, and from the average coverage per target in Figure 7, we can deduce that the primary reason behind this is the fact that the defender primarily surveilled the three hot spots. The proportion of coincidentally observed attacks is therefore also less, as most of the attacks take place at the hot spots. Secondly, the increase in accuracy could also result from the faster decrease of the value of γ. With lower γ values, the defender performs less explorative strategies in total.
Note that with a range of 15 km, the drone could not reach all cells starting from the base at cell $(4, 6)$, and could barely cover the hot spots. If the hot spots would have been out of its reach, the results could have been worse.

Not surprisingly, when the drone range is increased from 25 to 35 km, we observe a better performance. The average final regret decreases from 0.52 to 0.44. Interestingly, similar to the decrease in range, we can observe an inverse effect on the observation accuracy. Increasing the range results in a decrease in observation accuracy. The reasoning for this behaviour is similar. With a larger range, the drone now covers proportionally more cells, which results in more

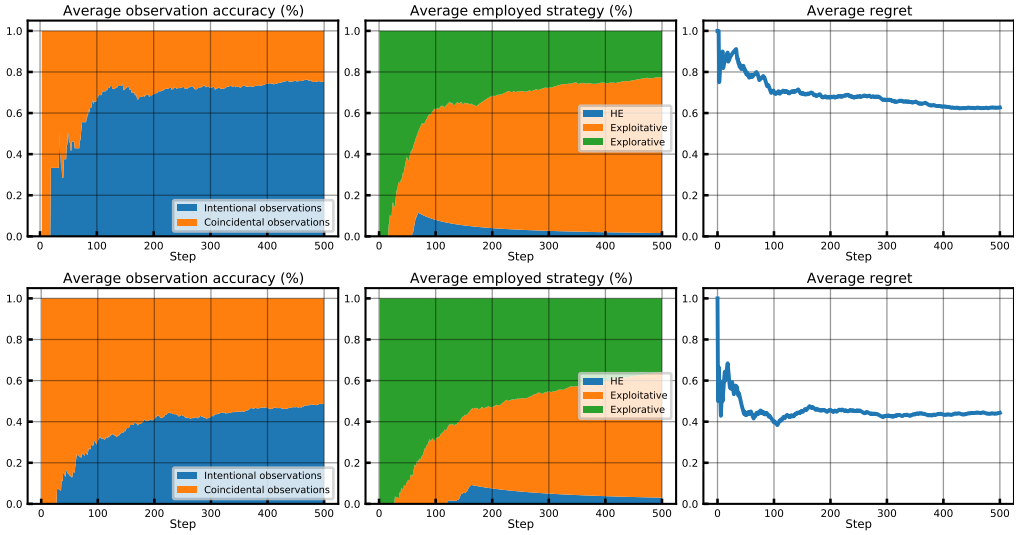

Figure 6. Comparison of the observation accuracy, the employed strategies and the regret for a defender with range $R = 15$ (top) and $R = 35$ (bottom) against a QR adversary.

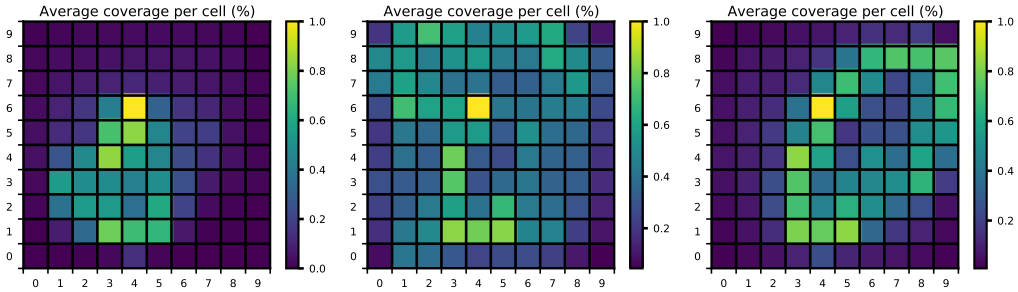

Figure 7. Comparison of the average coverage per cell for a defender with a range of 15, 35, and 25 km LTR.

coincidentally observed attacks. Of course, the value of gamma that is affected by the higher value of k results in more explorative routes, further increasing the coincidentally observed attacks.

Experiment 3 - Human Expert ε variation: Looking at the average employed strategies in this experiment we distinguish the following. As shown in Figure 8, the expert selection algorithm chose the perfect HE with $\varepsilon = 0$ against the QR attacker. This resulted in a lower final average regret than for the baseline model where the HE's error margin is higher ($R_t = 0.46$ versus $R_t = 0.52$ respectfully).

Unexpectedly, the expert selection algorithm chose the OL over the HE when facing an STC attacker. One would think that because of the similar probability distributions q_i and p_{he} of the STC attacker and the HE respectively, the HE would be better suited against the STC attacker. By chance, the HE might have performed poorly in the rounds after the expert selection algorithm was activated and never got a chance to redeem itself, or it is possible that the OL performed exceptionally well. The overall performance is comparable to the base model situation, which makes the latter situation less probable. This indicates that either the proposed expert selection might not function as good as intended, or that the perfect HE's performance against the STC attacker is not as good as expected.

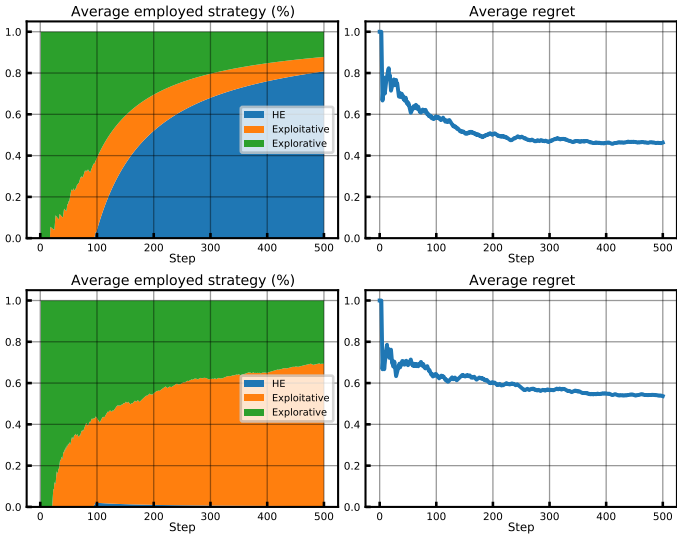

Figure 8. Simulation results from the base model with the HE's $\varepsilon = 0$ against a QR attacker (top) and an STC attacker (bottom).

Experiment 4 - Multiple attackers: The results from simulating the game with $m = 3$ and $m = 5$ in Figure 9 yield the following. When adding more attackers to the model, the γ variable reaches the threshold value $\theta = 0.4$ faster than the case with only a single attacker. Therefore we can see that the expert selection algorithm is activated earlier as well. We can also observe that the HE's estimations are better in the early stages of the game. The same holds for the strategy selection against the STC attacker (not represented in a figure), but in that case, the OL takes over earlier than against the QR attacker.

Even though the proportion of explorative strategies is also reduced due to the increased presence of attackers, the overall increase in attacks and observations results nonetheless in a fast learning process for the OL. This manifests itself in the definitive switch from HE to OL, and the drop in average regret that can be observed after that switch.

Note that the average regret is higher due to the higher number of attackers, but normalised by the number of attackers m we can observe that the performance compared with the base models is worse in the case of QR attackers, and better in the case of STC attackers. The reason for this might be that the QR attackers are adaptive and are modelled in as such that the individual attacker has the knowledge of the collective of attackers. This means that, just as the defender, the attackers in this model learn quicker when there are more attackers. This could also explain the difference in observation accuracy between the simulations against the QR attackers and the STC attackers.

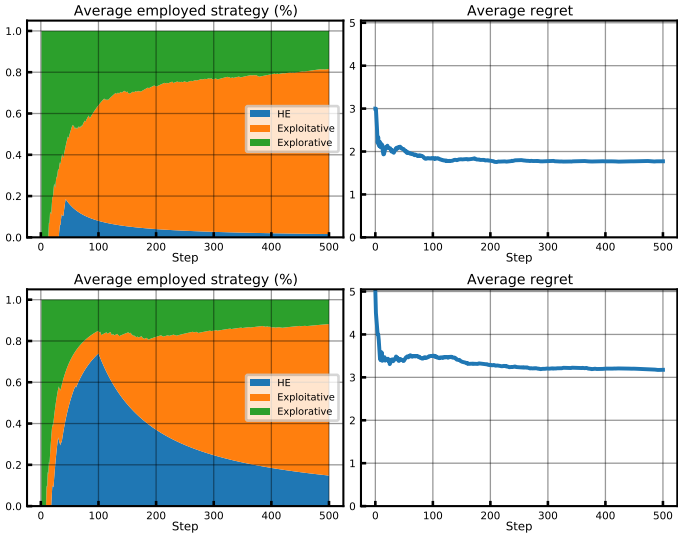

Figure 9. Comparison of the employed strategies and regret for simulations with 3 QR attackers (top) and 5 QR attackers (bottom).

Experiment 5 - Finer grid discretization: The finer discretization of the surveillance area results in the following. First of all, as can be observed in Figure 10, the spatial distribution of the attacks on a 20x20 grid is similar to the spatial distribution on a 10x10 grid. This reveals that the

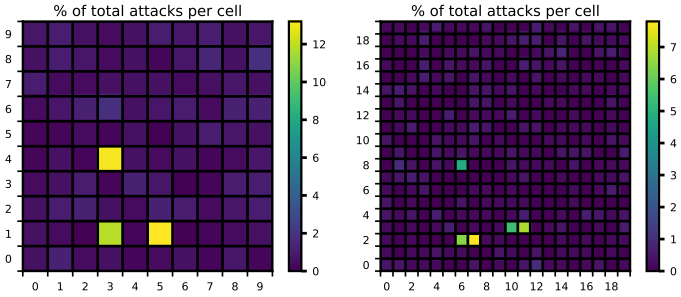

Figure 10. Comparison of attack distribution of a QR attacker with $\lambda = 0.4$ on a 10x10 grid (left) and a 20x20 grid (right).

model's calculation of q_i based on ground truth p_{GT} is not affected by a finer discretization on the same area. In turn, this means that the normalisation of the attackability score $\mathbf{a}$ to p_{GT} is scalable. Secondly, the overall performance of MEOMAPP is worse, with $R_t = 0.76$ and $R_t = 0.80$ against the QR and STC attacker respectively. This is not unexpected as the overall area that is covered by the defender every round is smaller due to the halving of the drone's FoV. Furthermore, comparing the two simulations on a 20x20 grid with each other in Figure 11, it is important to note that the initial performance of both simulations is very different and can be a reason for the difference in performance during the rest of the game. Playing against the STC adversary, MEOMAPP did not find the attacker until round 37. This significantly delayed the start of the learning process and possibly results in the choice to use the HE strategy for the rest of the game. Against the STC attacker, the observation accuracy is relatively low and the average coverage per target is more evenly distributed compared with the other coverage distributions obtained so far. Against the QR attacker, the coverage distribution of the defender is visually traceable to the distribution of estimated rewards per cell. That distribution is not directly relatable to the distribution of attacks per cell, which is more straightforward for the simulation results against the STC attacker. The observation accuracy playing against the QR attacker is significantly higher compared to the STC attacker. The increase in observation accuracy as of step 327 seems to result from the switch to the OL strategies.

VI. Discussion

In this section, we discuss the main findings of this research and the implications of assumptions on its results.

A. Reflection on overall performance

In general, the findings resulting from this study show that MEOMAPP exhibits adaptive behaviour when faced with different attacker types and different game settings. Performance-wise, MEOMAPP's regret converges within 500 steps for all presented game settings. Also, when presented with two experts, each proposing different defender strategies, MEOMAPP can choose the expert that performs best on average at any given time. This is consistent with the work from Xu et al. [21] and Gholami et al. [23], even though we chose to use a different but more realistic and reproducible

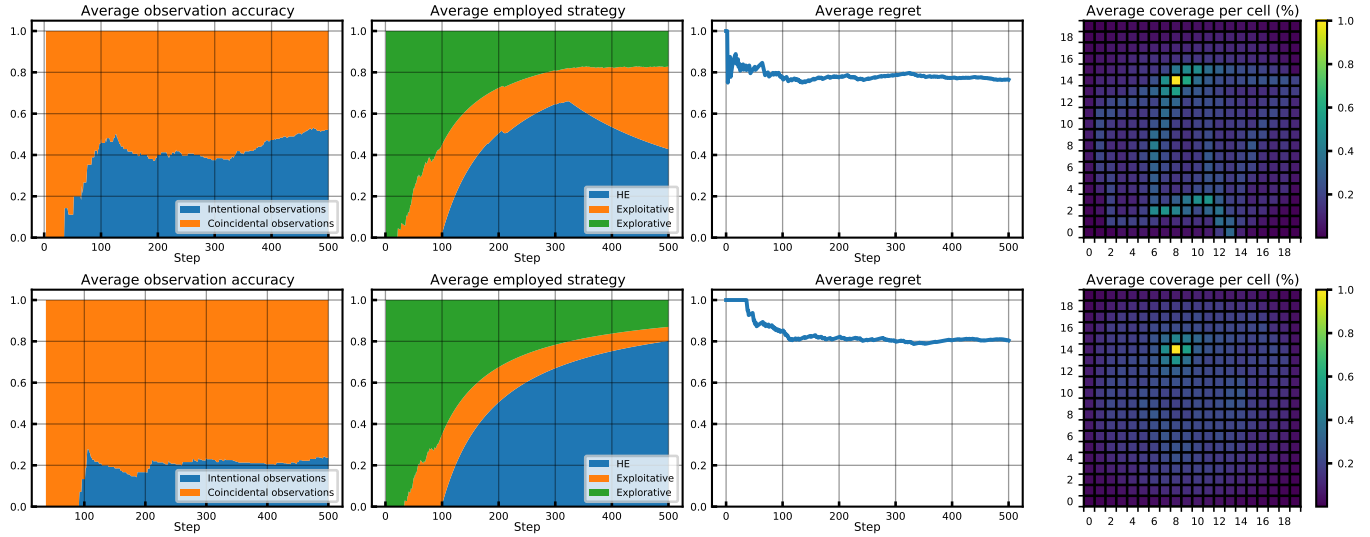

Figure 11. Simulation results from the baseline model on a 20x20 cell grid against a QR attacker (top) and an STC attacker (bottom).

experimental setup. Unfortunately, because of the different setup, it is not possible to directly compare the results, and thus we cannot state with certainty whether MEOMAPP performs better or worse than the stand-alone FPL-UE model [21] or the MINION model [23]. However, the similar trends compared with previous results show that the FPL-UE algorithm can be combined with a route planning algorithm more suitable for UAVs. Therefore, it is expected that MEOMAPP is a more comprehensive solution for aerial surveillance. Additionally, the simulation setup and results from this research can be used as a benchmark for future studies.

B. Reflection on limitations

It is important to put the findings in the perspective of specific limitations resulting from assumptions made for the model and from the specific simulations settings. Even though MEOMAPP is designed for and evaluated by a real-world situation, the simulated attacker behaviour and human expert defender strategies only approximate what a realistic setting would be. For a definitive verdict on MEOMAPP's performance and applicability in real-world wildlife surveillance settings, a field test validating the agent-based model and its simulations is highly recommended. It is important to keep in mind that for a real-world test, the notion of regret cannot be used to evaluate the model. The reason for this is that the regret is calculated using complete information of defender utility, which is not always available. For example: if no attacks are observed by the drones, it is not always possible to know if there was an attack and the drone missed it, or if there was no attack at all.

In this subsection, we elaborate on some important assumptions made for this agent-based model simulation and the limitations they represent.

The definition of the exploration/exploitation variable γ in FPL-UE assumes any adversary behaviour, but only as long as it is constant. The effect of changing adversary behaviour, or the introduction of more or new attackers, in a later stage of the game has not been investigated. In reality, however, this is not an unimaginable scenario.

The model assumes that attackers remain stationary at the attacked target for the whole duration of the defender's surveillance flight, or at least until they are observed. This assumed temporal relation between attacker and defender might heavily influence the real-world performance of MEOMAPP. Also, realistically attackers are present at other locations in the environment before and after attacking a certain target. This is also not reflected in the model. Furthermore, attackers are modelled as independent agents, meaning that attacker cooperation for an attack is not taken into account. Note that QR-based attackers are modelled as having collective knowledge after they were observed, but not as cooperative attackers before an attack.

The observational capabilities of the defender are assumed to be perfect within its modelled observational range. This means that the practical consequences of observing while flying are not taken into account, like bank angles when turning, speed variations, altitude variations, and influences from the weather.

The area that is surveilled is modelled as a two-dimensional, static environment. Changes to the environment and therefore possible variations in attacker and defender payoffs are not taken into account in this model.

It is important to note that even though MEOMAPP was evaluated using a specific real-world scenario, the online learner model does not make any specific assumptions about the park wherein it was simulated. MEOMAPP can be used in any wildlife park whatsoever.

VII. Conclusion

This research investigated if it is possible to apply an FPL-UE algorithm in a multi-expert learning model with a planning method suitable for drones to determine wildlife

surveillance strategies. We proposed MEOMAPP, a Multi-Expert Online Model for Aerial Patrol Planning that compares the performances of a defender strategy by the online learner FPL-UE and a defender strategy by a human expert, and uses the defender strategy to plan a flight path for a surveillance drone. We evaluated MEOMAPP using the agent-based modelling and simulation paradigm, using the real-world case of Aloegrove Safari Park in Namibia as an experimental setup for simulations. We demonstrated that MEOMAPP achieves convergence against two typical attacker models in a variety of simulation settings concerning the environment, the attackers, and the defender. The main contributions of this paper are:

1) The integration of a path planning algorithm for aerial vehicles and a game-theoretic defender strategy algorithm
2) An updated expert selection algorithm that allows the OL to mature before being evaluated
3) The evaluation of the ensemble of algorithms in a reproducible realistic simulation

Despite the agent-based model being a simplification of a real system, MEOMAPP is deemed a suitable algorithm for determining aerial surveillance strategies for wildlife surveillance.

VIII. Recommendations for Future Research

The proposed model and its simulation setup can serve as a foundation for future research in the field of Green Security Games and path planning for wildlife surveillance. The initial assumptions about the wildlife surveillance system made constrained components of the model resulting from this research. These components can be investigated in future research:

A. Attacker model

The following characteristics of the attacker are interesting for future research.

Even though a group of QR attackers could benefit from the collective knowledge, further research into **attacker coordination** could be done.

During the round, attackers are considered stationary. Including **attacker routes** would be a more realistic representation of the system.

If attacker data would be available, it is possible to evaluate MEOMAPP to a more **realistic attacker**. This data can be used to include a more realistic game-theoretic model of attacker behaviour, like SUQR [14], CAPTURE [18] or SHARP [17].

This does not mean that MEOMAPP will require prior knowledge, only that it could be evaluated against realistic players that are modelled using real attacker data.

More information about attackers can also be used for a **different temporal model** than the defender.

B. Defender model

Currently, the defender is a single drone. However, in reality, wildlife surveillance is not done by a drone alone. **Coordination with rangers** and other defensive agents is an interesting research field to elaborate on in the future.

C. Path planning

The path planning algorithm now takes into account the most simple drone model for its constraints. Including constraints related to **flight dynamics** can give insights in the actual flight path.

Even though the online learner's GR algorithm can learn from the coincidental observations, a path planning algorithm that includes estimated rewards of the arcs, as well as rewards of the nodes, could produce more optimised flight paths.

If more defenders enter the game, Multi-Agent Path Finding algorithms could be studied in the context of GSGs.

References

[1] D. Mouillot, D. R. Bellwood, C. Baraloto, J. Chave, R. Galzin, M. Harmelin-Vivien, M. Kulbicki, S. Lavergne, S. Lavorel, N. Mouquet, C. E. T. Paine, J. Renaud, and W. Thuiller, "Rare species support vulnerable functions in high-diversity ecosystems," *PLOS Biology*, vol. 11, no. 5, pp. 1–11, May 2013.

[2] L. O. Smith and C. Gerstetter, "The Costs of Illegal Wildlife Trade: Elephant and Rhino. A study in the framework of the EFFACE research project," Ecologic Institute, Berlin, Tech. Rep. 1, 2015. [Online]. Available: www.efface.eu

[3] Unknown. (2020, July) Recognizing and supporting rangers working against all odds. UNESCO. [Online]. Available: https://whc.unesco.org/en/news/2139

[4] E. Bondi, A. Kapoor, D. Dey, J. Piavis, S. Shah, R. Hannaford, A. Iyer, L. Joppa, and M. Tambe, "Near real-time detection of poachers from drones in airsim," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, July 2018, pp. 5814–5816.

[5] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Clune, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 25, pp. E5716–E5725, 2018.

[6] J. Jiménez López and M. Mulero-Pázmány, "Drones for conservation in protected areas: Present and future," *Drones*, vol. 3, no. 1, 2019.

[7] B. Ivoševi, Y.-g. Han, Y. Cho, and O. Kwon, "The use of conservation drones in ecology and wildlife research," *Journal of Ecology and Environment*, no. February, 2015.

[8] F. Fang, P. Stone, and M. Tambe, "When security games go green: Designing defender strategies to prevent poaching and illegal fishing," in *IJCAI International Joint Conference on Artificial Intelligence*, vol. January, 2015, pp. 2589–2595.

[9] R. Longadge and S. Dongre, "Class imbalance problem in data mining review," *CoRR*, vol. abs/1305.1707, 2013.

[10] T. D. Pigott, "A review of methods for missing data," *Educational Research and Evaluation*, vol. 7, no. 4, pp. 353–383, 2001.

[11] R. Yang, C. Kiekintveld, F. Ordonez, M. Tambe, and R. John, "Improving resource allocation strategy against human adversaries in security games," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One*, ser. IJCAI'11. AAAI Press, 2011, p. 458–464.

[12] J. Pita, R. John, R. Maheswaran, M. Tambe, R. Yang, and S. Kraus, "A robust approach to addressing human adversaries in security games," vol. 242, June 2012, pp. 1297–1298.

[13] M. R. D. and T. Palfrey, "Quantal response equilibria for normal form games," *Games and Economic Behavior*, vol. 10, no. 1, pp. 6–38, 1995.

[14] 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*, ser. AAAI'13. AAAI Press, 2013, p. 718–724.

[15] R. Yang, B. Ford, M. Tambe, and A. Lemieux, "Adaptive resource allocation for wildlife protection against illegal poachers," in *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2014, pp. 453–460.

[16] F. Fang, T. H. Nguyen, R. Pickles, W. Y. Lam, G. R. Clements, B. An, A. Singh, B. C. Schwedock, M. Tambe, and A. Lemieux, "Paws — a deployed game-theoretic application to combat poaching," *AI Magazine*, 2017.

[17] D. Kar, F. Fang, F. M. D. Fave, N. Sintov, and M. Tambe, "A Game of Thrones: When Human Behavior Models Compete in Repeated Stackelberg Security Games," in *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, 2015, pp. 1381–1390.

[18] T. H. Nguyen, A. Sinha, S. Gholami, A. J. Plumptre, L. N. Joppa, M. Tambe, M. Driciru, F. Wanyama, A. Rwetsiba, R. Critchlow, and C. M. Beale, "CAPTURE: A New Predictive Anti-Poaching Tool for Wildlife Protection," *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 767–775, 2016.

[19] C. Kiekintveld, J. Marecki, and M. Tambe, "Approximation methods for infinite bayesian stackelberg games: Modeling distributional payoff uncertainty," in *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, Tumer, Yolum, Sonenberg, and Stone, Eds., Taipei, 2011, pp. 2–6.

[20] A. Blum, N. Haghtalab, and A. D. Procaccia, "Learning optimal commitment to overcome insecurity," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 1826–1834.

[21] H. Xu, L. Tran-Thanh, and N. R. Jennings, "Proceedings of the 15th international conference on au- tonomous agents and multiagent systems," 2016.

[22] G. Neu and G. Bartók, "An efficient algorithm for learning with semi-bandit feedback," *CoRR*, vol. abs/1305.2732, 2013.

[23] 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 *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems*, vol. 9, 2019.

[24] C. Camerer and R. S. Foundation, *Behavioral Game Theory: Experiments in Strategic Interaction*, ser. The Roundtable Series in Behavioral Economics. Princeton University Press, 2003.

[25] D. Kar, B. Ford, S. Gholami, F. Fang, A. Plumptre, M. Tambe, M. Driciru, F. Wanyama, A. Rwetsiba, S. California, and L. Angeles, "Cloudy with a chance of poaching : Adversary behavior modeling and forecasting with real-world poaching data," in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, no. May, 2017, pp. 159–167.

[26] S. Gurumurthy, L. Yu, C. Zhang, Y. Jin, W. Li, H. Zhang, and F. Fang, "Exploiting Data and Human Knowledge for Predicting Wildlife Poaching," in *ACM SIGCAS Conference on Computing and Sustainable Societies 2018*, 2018.

[27] S. Gholami, B. Ford, F. Fang, A. Plumptre, M. Tambe, M. Driciru, F. Wanyama, A. Rwetsiba, M. Nsubaga, and J. Mabonga, "Taking It for a Test Drive: A Hybrid Spatio-Temporal Model for Wildlife Poaching Prediction Evaluated Through a Controlled Field Test," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10536 LNAI, 2017, pp. 292–304.

[28] S. Gholami, S. Mc Carthy, B. Dilkina, A. Plumptre, M. Tambe, M. Driciru, F. Wanyama, A. Rwetsiba, M. Nsubaga, J. Mabonga, T. Okello, and E. Enyel, "Adversary models account for imperfect crime data: Forecasting and planning against real-world poachers," 2018, pp. 823–831.

[29] N. Park, E. Serra, T. Snitch, and V. S. Subrahmanian, "APE: A Data-Driven, Behavioral Model-Based Anti-Poaching Engine," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 2, pp. 15–37, 2015.

[30] P. Toth and D. Vigo, *The vehicle routing problem*. SIAM, 2002.

[31] A. Gunawan, H. C. Lau, and P. Vansteenwegen, "Orienteering problem: A survey of recent variants, solution approaches and applications," *European Journal of Operational Research*, vol. 255, no. 2, pp. 315 – 332, 2016.

[32] R. Martí, P. M. Pardalos, and M. G. C. Resende, Eds., *Handbook of Heuristics*. Springer, 2018.

[33] M. Speranza and C. Archetti, "A survey on matheuristics for routing problems," *EURO Journal on Computational Optimization*, vol. 2, November 2014.

[34] A. Bazghandi, "Techniques, advantages and problems of agent based modeling for traffic simulation," 2012.
[35] A. McLane, C. Semeniuk, G. Mcdermid, and D. Marceau, "The role of agent-based models in wildlife ecology and management," *Ecological Modelling*, vol. 222, pp. 1544–1556, April 2011.
[36] F. Bousquet, R. Lifran, M. Tidball, S. Thoyer, and M. Antona, "Agent-based modelling, game theory and natural resource management issues," *The Journal of Artificial Societies and Social Simulation*, vol. 4, March 2001.
[37] R. Luce, *Individual choice behavior: a theoretical analysis*. Wiley, 1959.
[38] S. Anderson, J. Goeree, and C. Holt, "The logit equilibrium: A perspective on intuitive behavioral anomalies," *Southern Economic Journal*, vol. 69, pp. 21–47, July 2002.
[39] D. O. Stahl and P. W. Wilson, "Experimental evidence on players' models of other players," *Journal of Economic Behavior & Organization*, vol. 25, no. 3, pp. 309 – 327, 1994.
[40] B. Gao and L. Pavel, "On the properties of the softmax function with application in game theory and reinforcement learning," 2017.
[41] D. McFadden, "Conditional logit analysis of qualitative choice behaviour," in *Frontiers in Econometrics*, P. Zarembka, Ed. New York, NY, USA: Academic Press New York, 1973, pp. 105–142.
[42] F. Klügl and A. L. C. Bazzan, "Agent-based modeling and simulation," *AI Magazine*, vol. 33, no. 3, p. 29, September 2012. [Online]. Available: https://www.aaai.org/ojs/index.php/aimagazine/article/view/2425
[43] L. Morell, "Unit testing and analysis," April 1989.
[44] D. Helbing and S. Balietti, "How to do agent-based simulations in the future: From modeling social mechanisms to emergent phenomena and interactive systems design," *Technical Report 11-06-024*, July 2015.
[45] D. Masad and J. Kazil, "Mesa: An agent-based modeling framework," January 2015, pp. 51–58.
[46] G. van Rossum, B. Warsaw, and N. Coghlan, "Style guide for Python code," PEP 8, 2001. [Online]. Available: https://www.python.org/dev/peps/pep-0008/
[47] S. Mitchell, S. M. Consulting, and I. Dunning, "Pulp: A linear programming toolkit for python," 2011.
[48] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2020. [Online]. Available: http://www.gurobi.com
[49] K. Dhoore, "Online agent-based aerial patrol planning for wildlife surveillance," Master's thesis, Delft University of Technology, Delft, October 2020.