Evaluation of a Pure-Strategy Stackelberg Game for Wildlife Security in a Geospatial Framework

Lisa-Ann Kirkland[1], Alta de Waal[1,2], and Johan Pieter de Villiers[1]

[1] University of Pretoria, Pretoria, South Africa
lisakirkland25@gmail.com, alta.dewaal@up.ac.za,
pieter.devilliers@up.ac.za
[2] Centre for Artificial Intelligence (CAIR), Pretoria, South Africa

Abstract Current research on wildlife security games has minimal focus on performance evaluation. The performance of the rangers is evaluated by assessing their game utility, sometimes in comparison with their maximin utility, and other times in comparison with their real-world utility when the game is implemented in a wildlife park. Currently no evaluation framework exists, and this paper proposes an evaluation suite to address this. The movements of the wildlife, the rangers, and the poachers are simulated over a grid of cells corresponding to the wildlife park, where cells containing geographical obstacles are excluded. Poaching and arrest frequency are the primary evaluation measures used. Firstly, we develop a null game to act as a baseline. Typically, one would expect random behaviour of all agents in the null game. However, we simulate random movement for the rangers but more intelligent movement for the poachers. The motivation for this design is to assess whether executing the Stackelberg game yields significantly better ranger performance than random movement, while keeping the poachers' behaviour consistent. The intelligent poachers move by taking their geographical preferences into account and learn from poaching and arrest events. Secondly, we propose that the rangers act as the Stackelberg follower instead of the leader. We formulate a simple pure-strategy Stackelberg game and implement four variations of the game within the framework. The results of the simulations show that the rangers perform better than random when using the Stackelberg game and perform best when acting as the follower.

Keywords: Evaluation · Wildlife security · Game theory · Stackelberg.

1 Introduction

Rhino poaching continues to be a major problem in South Africa. Although rhino deaths started decreasing in 2015, the number of poaching activities inside and adjacent to the Kruger National Park (KNP) has only decreased from 2 466 in 2015 to 2 014 in 2019 [3,4]. Anti-poaching units offer an attempt to combat rhino poaching [15] and to facilitate such efforts, this paper focuses on wildlife security games [23]. These games make use of the Stackelberg Security Game (SSG) [5], the current game-theoretic approach in security domains. In the domain of wildlife

Kirkland L., de Waal A., de Villiers J.P. (2020) Evaluation of a Pure-Strategy Stackelberg Game for Wildlife Security in a Geospatial Framework. In: Gerber A. (eds) Artificial Intelligence Research. SACAIR 2021. Communications in Computer and Information Science, vol 1342. Springer, Cham. https://doi.org/10.1007/978-3-030-66151-9_7

security, the attackers are poachers, the defenders are rangers and the targets to protect are moving animals. SSGs are used to optimally allocate limited ranger resources in a wildlife park where attacks on the animals occur frequently. To our knowledge, there is currently no evaluation framework for wildlife security games and there are inconsistencies in the evaluation. Evaluating the games based on expected utility is difficult because it is based on the location of the wildlife animals, who are constantly moving. This paper introduces a framework which simulates the movements of the poachers, rangers, and wildlife to address this. The simulation studies are intended to provide the rangers with an estimate of the average behaviour in one month. We propose clear evaluation metrics for the rangers to assess their performance, which allows them to decide on the best strategy for real-world implementation to combat the poachers. Furthermore, we propose acting as the Stackelberg follower instead of the leader in this domain. A simple pure-strategy Stackelberg game is designed and implemented within the framework to test this idea.

The SSG is an extensive form game wherein the defenders act first and the attackers follow. The game can be represented by a game tree where the branches are the actions of the agents and their payoffs for each combination of actions are given at the terminal nodes. The SSG assumes that attackers conduct surveillance on the defenders to obtain complete knowledge of their mixed-strategy and then respond with a pure-strategy by attacking a single target [21]. A pure-strategy for each agent consists of the cross product of the set of actions available to them at each of their information states. A pure-strategy equilibrium provides the optimal action to take at each of their nodes in the tree. A mixed-strategy is a probability distribution over the set of pure-strategies. A Green Security Game (GSG), which includes protection of wildlife, fisheries, and forests, has frequent attacks on targets. Attackers can therefore not afford much time to conduct extensive surveillance to learn the mixed-strategy of the defenders [10]. Furthermore, there could be many attackers present at any given time [17] so assuming a pure-strategy response for the attackers is not viable. Attackers in this domain often return to sites of past success [13] and since attacks occur frequently, the defenders can gather enough observational data to learn the mixed-strategy of the attackers. Thus, we propose that the defenders could perform better when acting as the Stackelberg follower. There is not always an advantage in terms of payoffs to being the Stackelberg leader [16], so it is reasonable to reverse the roles of the defender and the attacker in the domain of GSGs. Although the follower in SSGs acts with a pure-strategy, this is only necessary to ease the computation of the leader's optimisation problem [18]. However, we do not need to find an optimal strategy for the attackers since we learn this from the data. We only need to solve the follower's problem, which is computationally much simpler, to find the defenders' best response to the mixed-strategy of the attacker.

A sensible question is whether we should use the Stackelberg model at all. The Stackelberg duopoly game originates from economics where two firms share the market for a certain product [20]. The price of the product depends on the quantities produced by both firms and each firm's profit depends on this price and the cost per unit of production. The problem is to determine what quantity of the product each firm must produce to maximise their profit. The Cournot duopoly game solves this problem when both firms make their decision at the same time and the Nash equilibrium (NE) is for them to produce the same quantity [7]. In the Stackelberg duopoly, the leader has an advantage and can choose to produce more than the Cournot quantity which will increase their profits. The follower's best response is to produce less than the Cournot quantity which decreases their profit. However, in security games the payoffs are structured differently: defenders do not wish to share the targets with the attackers but instead want to prevent the attackers from making any attacks on the targets. It has been shown that the Stackelberg equilibrium and the NE for the defenders are interchangeable [24]. The SSG model therefore yields the NE mixed-strategy for the defenders and a pure-strategy for the attackers, which results in a defender payoff that may be higher than the NE payoff. However, the attackers' best response to this mixed-strategy has the same maximal value (their NE payoff) for any pure-strategy in the support of their optimal mixed-strategy [18] so they have no reason not to deviate and choose a pure-strategy which may result in a defender payoff that is lower than the NE payoff. When computing optimal strategies to commit to, it is often assumed that the follower will break ties in the leader's favour (to maximise the leader's payoff) [6] but this is not a reasonable assumption in security domains. The Stackelberg model thus seems to have a follower's advantage in GSGs. By acting as the follower, the defenders can learn the mixed-strategy of the attackers from past observations. They can then choose a mixed-strategy best response that secures their own NE payoff but yields an attacker payoff that is lower than the NE payoff.

In Section 2 we provide a brief overview of current research about wildlife security and how the performance of games is evaluated. Section 3 describes the null game, which serves as a baseline model. The formulation of a simple wildlife security game, where both the rangers and the poachers execute a pure-strategy, is described in Section 4. Variations of the simple game are implemented within the framework in Section 5 and compared with the null game and each other. We conclude the paper in Section 6 with some remarks and suggestions for future research, and thereafter provide a summary of input parameters in an Appendix.

2 Related Work

The Bayesian SSG has become the standard approach for security games and handles uncertainty around the attackers' payoffs by assuming different types of attackers with a Bayesian *a priori* distribution assumption [18]. Uncertainties due to the attackers' bounded rationality and limited observations are addressed by using robust algorithms [19]. The algorithms are evaluated by assessing the defenders' reward against two baselines: the uniform strategy, which assigns equal probability of taking each strategy; and the maximin strategy, which maximises the minimum reward of the defenders irrespective of the attackers' actions.

The first application of the Bayesian SSG for wildlife security is the PAWS algorithm [23]. Since the targets to protect are moving animals whose location is not always known, the wildlife park is divided into a grid of cells which become the new targets. Available poaching data is utilised to learn a behavioural model for the poachers to account for their bounded rationality. Evaluation of the SSG algorithm is achieved by comparing the cumulative expected utility of the rangers over 30 simulated rounds of the game against the maximin strategy on a grid of 64 cells. Different behavioural models are compared in Yadav et al. [22] where the models are learned using data from a wildlife park in Indonesia and their predictive performance is tested using ROC curves. The SSG algorithm is evaluated on 25 randomly selected grid cells, where the rangers' maximum regret of the optimal strategy is compared to that of the real world.

The work in Kar et al. [13] follows a similar approach to that of Yadav et al. [22] by comparing different behavioural models. However, they develop a computer game with a 5x5 grid over a Google Maps view of the wildlife park. A probability heat map of the rangers' mixed-strategy is overlayed onto the grid and the wildlife is arranged in four different payoff structures. On average 38 human subjects played as the attackers in 5 rounds of each game to learn the behavioural models and the defenders' utility against these subjects is compared for evaluation. Although there is some similarity to this work, no movements of the wildlife, the rangers or the poachers are considered.

In our earlier work [14], a null game is designed where the rangers and the poachers both act randomly. Thus, the uniform strategy [19] is similar to this null game. Yet none of the research on wildlife security use the uniform strategy for comparison and the maximin strategy is the only baseline model considered. While evaluating models by simulating repeated instances of the game or by applying the game to real-world data is valid, the data used only provides a snapshot of the situation. Since the wildlife are constantly foraging for food, the poachers foraging for wildlife and the rangers foraging for poachers, it becomes important to know more about their spatial and temporal movements [15]. Measuring the expected utility is useful for comparison but since it is calculated based on the locations of the wildlife, who are constantly moving, the actual number of wildlife poached is more valuable. The evaluation framework presented in this paper allows for including real-world data and is supposed to offer an alternative to implementation in a wildlife park.

3 The Null Game

3.1 Geographical Features

Our previous null game [14] focuses on a grid of cells and follows the routes of a herd of wildlife, one group of rangers and one group of poachers until the poachers are arrested or leave the park. It is assumed that 10 games occur within a month, and for each month we record the number of poaching events and the number of times the poachers leave or are arrested. The monthly cycle is simulated for 1 000 Monte Carlo repetitions and the averages over the simulations provide the measures of performance. Some simulations on ways to move are compared and movement towards a destination provides the smoothest and most realistic movement for the rangers and the poachers. The start and destination cells for the rangers and the poachers are chosen completely randomly. Although the grid cells are supposed to correspond to a map of the wildlife park, the game does not take any geographical features into account. We modify that game to incorporate geographical information into the framework. Figure 1 shows a map of the KNP and a subarea used to demonstrate how the algorithm works. Currently we use the public shapefiles in the SANParks data repository [2] for the KNP. The `geopandas` Python library [1] is utilised so that geographical data can be used, and distances can be calculated within the actual wildlife park. Two classes are created, a `Park` class and an `Agent` class. Within the `Park` class, all the geographical information is collected. A grid is also calculated, based on how large the cells should be, for either the whole park or a subarea of the park.

Two attributes are created for the `Agent` class to exclude cells in the grid, corresponding to areas where the agent cannot go. The first restricts agents from entering areas which contain geographical obstacles such as steep mountains, dense vegetation, rivers, and dams. The second restricts an agent to stay within a specific area. For example: you might want to restrict the wildlife to stay within an area defined by a census or home range analysis; ranger patrols might need to stay within a certain distance from their patrol huts; or poachers might need to stay close enough to their homes. Figure 2 shows a grid of 566 cells that are 1.5x1.5 km in size, for the subarea in Figure 1, where the cells excluded are white.

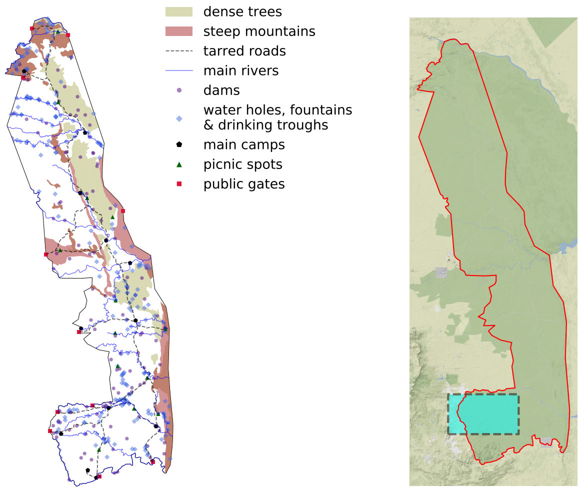

Figure 1. Map and subarea of the Kruger National Park. The map on the left illustrates the geographical information used within the framework, while the subarea shown on the right will be the focus of this study.

3.2 Poacher Movement

The null game serves as a baseline model, so that when compared with other games, the performance of the rangers' success in protecting wildlife and arresting poachers can be assessed. We would like to know whether executing the GSG helps the rangers to perform better than when they execute random motion. However, in our previous attempt to construct a realistic baseline model, both the rangers and the poachers moved from a random starting cell towards a random destination cell (uniform game). This makes it difficult to assess whether the rangers' performance improves when both the rangers and the poachers act according to the GSG since the poachers' strategy improves at the same time as the rangers' strategy. Thus, in the baseline model, the random rangers need to compete with more intelligent poachers, those who learn, to truly evaluate any performance increase of the rangers when they act according to the GSG.

In a game theory algorithm, we would use information about an agent's preferences to try and quantify their payoffs. Similarly, we can use this information to determine how an intelligent agent might move through the wildlife park. Another two attributes were created for this: one for features they dislike and how far they would like to stay away from them; and one for features they like and how near they would like to stay to them. For example: the poachers would probably like to avoid any entrance gates to the park since rangers often conduct searches there; they would likely stay away from main roads, camps and picnic spots to avoid being identified by the public; they might prefer to stay near to the park border to make escape easier; and they would possibly like to stay near to water sources since it is likely that they might find wildlife there. These preferences are implemented by increasing or decreasing selection weights for each cell. For each feature, each cell starts with a weight of $w = 0.5$. If it is a feature that the poachers would like to stay d km away from, then the weight starts decreasing for cells that are within d km away and continues to decrease as the cells get nearer to the feature: if a cell c_i has minimum distance d_i km from the feature, then its weight will be $w = w \times [1-(d-d_i)/d]$. Similarly, if it is a feature that the poacher would like to stay d km near to, then the weight increases more for cells c_i which are nearer to that feature: $w = w \times [1 + (d - d_i)/d]$. The weights are increased or decreased in this manner for each feature that the poacher has preferences for. Figure 2 shows the cell selection weights for a poacher who dislikes being 2 km from camps, 3 km from roads and 5 km from gates, and who likes being within 15 km of dams and water and within 30 km of the border. The weights are depicted by a colour scale, where darker colours indicate higher values.

Along with moving towards their preferences, the poachers also learn from events that occur. The poachers begin in a random cell, either on the border of the park or at the edge of the grid and proceed towards a random destination cell. If they reach the destination cell with no event, then they head off on a new trajectory back towards the start cell and we record that they left before poaching. If they leave safely without being arrested, then they continue to use that point of entry because they assume there is low risk in being arrested there. The poachers can either encounter wildlife for poaching on their trajectory towards

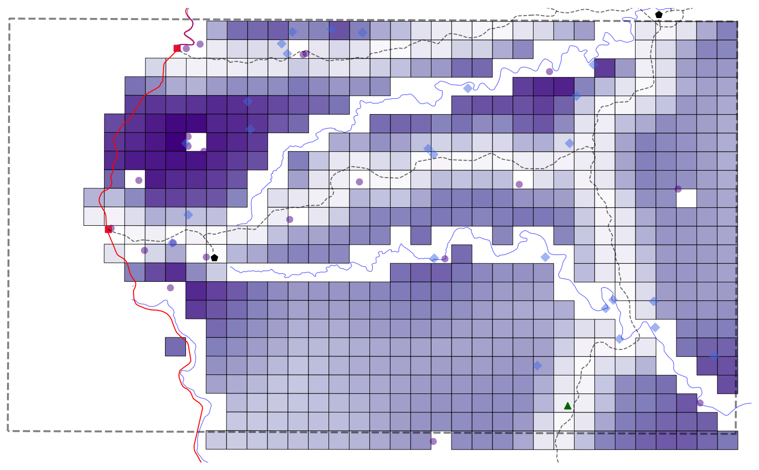

Figure 2. Poacher allowed cells and cell selection weights. The cells excluded for the poachers are shown in white and the selection weights are depicted by a colour scale where darker colours represent higher values.

the destination cell or on their way back to the start. We assume that once the wildlife has been poached, that they would like to exit promptly before being arrested. Thus, after a poaching occurs while going towards the destination, they change direction and head back towards the start cell. Furthermore, since they know where the wildlife are likely to be, they want to return to that area, so the poaching cell becomes their new destination cell. If the poachers encounter the wildlife while going towards the start, then they continue on the same trajectory towards the start after the poaching event. They would also adopt the poaching cell as their new destination cell when re-entering the park in this event. We do not allow for a second poaching event once on the trajectory towards the start cell after a poaching, they can only poach again after re-entry to the park. We record that the poachers left after poaching if they leave the park safely in the past two events described. However, if they are arrested before reaching the start cell then we record that they were arrested after poaching. Of course, the poachers could also be arrested without having poached any wildlife and in this case, we record that they were arrested before poaching. After being arrested, the poachers must re-enter the park, but they will choose a new random entry point since they did not have success going towards the current start cell. Figure 3 demonstrates the different scenarios that occur during one game.

The two attributes describing the poachers' preferences can be helpful in making the wildlife movement more realistic as well. Some examples include wildlife wanting to stay near water; liking specific types of vegetation for grazing; enjoying mud baths or shady areas; and wanting to avoid camp areas where there are lots of people. The null game is thus simulated with better movement for the wildlife as well, where the destination cell is chosen as a random cell near water and movement is based on the preferences specified.

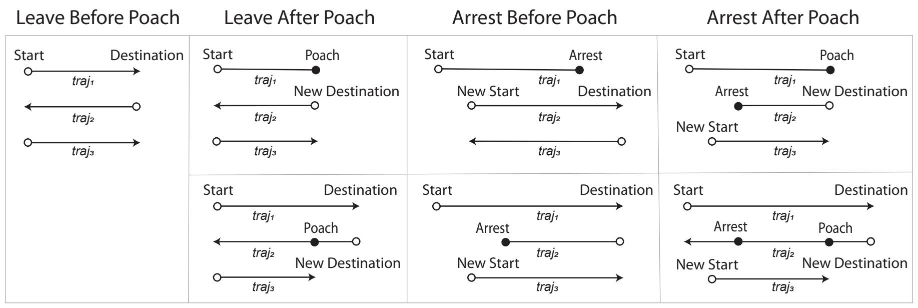

Figure 3. Scenarios that can occur and how the poachers learn from events. Trajectories continue back and forth between the start and destination cell until the game ends after a specified number of moves. An open circle represents the start of each trajectory and an arrow represents the end of that trajectory. A solid circle represents a poach or arrest event which can occur along any trajectory.

The game continues with the wildlife, rangers and poachers moving back and forth between their start and destination cells and ends after a specified number of moves. For example, considering a grid with 1 km^2 cells, a person could walk 100 km in 20 hours at a speed of 5 km/h. Allowing short stops to rest or eat, 100 moves would consume an entire day. Thirty such games could be played per month and the games are simulated for a specified number of months to determine their average monthly behaviour. The number of poaching events and arrest events per month are recorded and are used to calculate the poach frequency per day and arrest frequency per day, so that games of different lengths can be compared. When these measures are similar, we consider two secondary measures: the average number of moves for each arrest and the average distance between the poachers and rangers for games with no arrests. For further understanding and analysis, we also record how many times the agents reach their start cell or destination cell to keep track of their trajectories. The `movingpandas` Python library [12] is utilised to store trajectories which can be easily analysed and plotted after the simulation.

4 Simple Security Game

As a start, we would like to implement the simplest security game within the framework to demonstrate whether there are any improvements in the performance of the rangers. We consider the traditional Stackelberg Game, where the follower observes the leader's pure-strategy and reacts with his best response pure-strategy. As an example, to explain how the game works, we consider the wildlife park being divided into only two grid cells. The idea can easily be extrapolated to use all the cells provided in the grid. Before we continue to develop the game, we first make the following assumptions:

– there is only one group of rangers and one group of poachers, where each group acts together and they cannot split up;
– the park is divided into two grid cells;
– the rangers act as the leader and commit to protecting a single grid cell;
– the poachers observe which cell the rangers protect and react by attacking a single grid cell; and
– the rangers and the poachers act to maximise their own expected utility.

With these assumptions set clearly, we can identify the components of the game. We know that the agents are the rangers and the poachers, and that the actions are the coverage of the two grid cells by these agents. We do not yet know what the payoffs are, but we know that the outcome will include the number of rhino saved and/or poached and whether the poacher is arrested. For this example, we have two rhinos in grid cell 1 which is 500 m from the border and one rhino in grid cell 2 which is on the border. The four possible events are shown in Figure 4. For the outcomes of each event, the solid rhinos represent the number of rhinos saved and the dotted rhinos represent the number of rhinos poached. If the rangers and the poachers are in the same grid cell, then the poachers are arrested and no rhinos are poached.

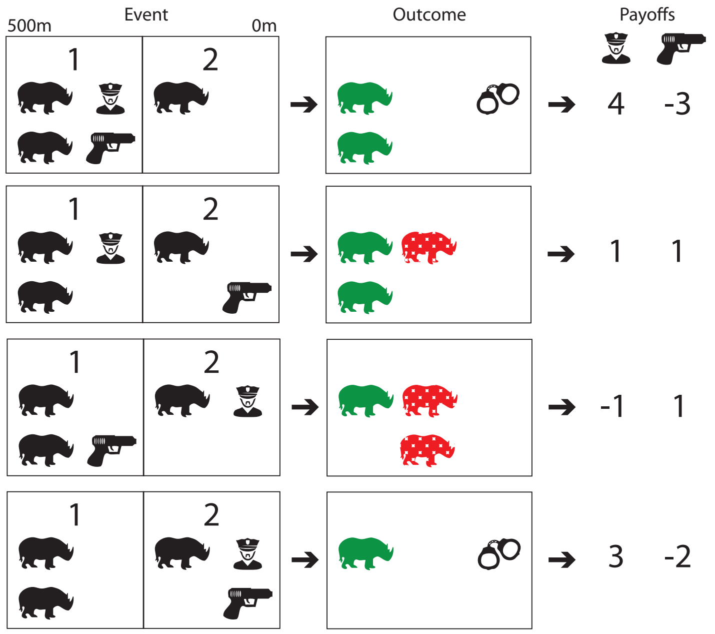

Figure 4. Game events and outcomes. For events, the policeman represents the rangers, the gun represents the poachers and the solid (black) rhino depicts the animals in each grid cell. For outcomes, the solid (green) rhino depicts the saved rhinos, the dotted (red) rhino depicts the poached rhinos and an arrest is represented by handcuffs.

In order to solve the game, we need to define the pure-strategies for each agent and quantify the outcomes as payoffs. Since the rangers act first, they have one information state and their pure-strategies are just their actions, cell 1 and cell 2, given by the set $S^R = \{1, 2\}$. Because the poachers observe the rangers' action, they have two information states: $P.1$ when the rangers go to cell 1 and $P.2$ when the rangers go to cell 2. The pure-strategies of the poachers thus need to specify what action to take at each information state, and are given by the set $S^P = \{(i, j) : \text{for } i, j = 1, 2\}$, which means cell i at $P.1$ and cell j at $P.2$. To calculate the payoffs, let the value of a rhino be 1 and the value of an arrest be 2. A poached rhino will count as negative for the rangers and an arrest will count as negative for the poachers. The payoffs can also include an agent's preferences, so let the penalty for the poachers be -1 for every cell that is 500 m away from the border. Let $u^R\left(S_i^R, S_{j.i}^P\right)$ be the payoff for the rangers and $u^P\left(S_i^R, S_{j.i}^P\right)$ be the payoff for the poachers when the rangers are are in cell i and the poachers are in cell $j.i$, where S_i^R is the ith element in S^R and $S_{j.i}^P$ is the jth element of S^P with the action at information state $P.i$. Then the payoffs are calculated as:

$$u^R\left(S_i^R, S_{j.i}^P\right) = \begin{cases} r_i + a_i + 2 & \text{if } i = j.i \\ r_i + a_i - a_{j.i} & \text{if } i \neq j.i \end{cases} \tag{1}$$

and

$$u^P\left(S_i^R, S_{j.i}^P\right) = \begin{cases} p_{j.i} - 2 & \text{if } i = j.i \\ p_{j.i} + a_{j.i} & \text{if } i \neq j.i \end{cases}, \tag{2}$$

where a_i is the number of animals in cell i, r_i is the geographic utility of the rangers in cell i (0 in this example), $p_{j.i}$ is the geographic utility of the poachers in cell $j.i$, and 2 is for an arrest. With the payoffs known, we can define the game mathematically as $G = \langle A, S, U \rangle$, where

- $A = \{R, P\}$ is the set of agents with R denoting the rangers and P denoting the poachers;
- $S = \left\{S^R, S^P\right\}$, where $S^R = \{1, 2\}$ is the set of pure-strategies for the rangers and $S^P = \{(1, 1), (1, 2), (2, 1), (2.2)\}$ is the set of pure-strategies for the poachers; and
- $U = \left\{u^R, u^P\right\}$ is the set of utility functions, where $u^R, u^P : S^R \times S^P \to \mathbb{R}$ are defined in equations 1 and 2.

We construct the game tree in Figure 5 to describe the game. The solution to a pure-strategy Stackelberg Game is given by the Subgame Perfect Nash equilibrium (SPNE) and is found using backward induction [16]. Figure 5 shows the three subgames in this game. The SPNE requires that the solution has a Nash equilibrium (NE) in each subgame, even if it is never reached. The backward induction process for finding the SPNE is presented visually in Figure 5 with thick lines representing the optimal strategies for each agent. Let $\hat{S}^R$ denote the optimal strategy for the rangers and $\hat{S}^P$ the optimal strategy for the poachers. The SPNE is $\hat{S}^R = 1, \hat{S}^P = (2, 1)$, thus the rangers go to cell 1 and the poachers to cell 2, with payoffs of 1 for the rangers and 1 for the poachers.

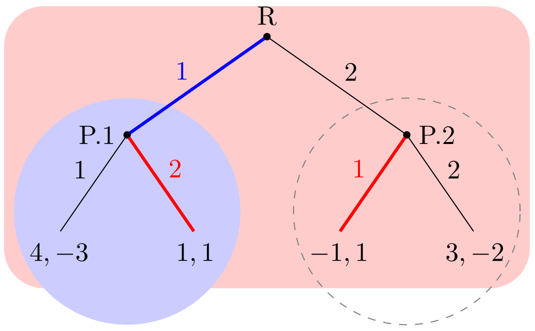

Figure 5. Stackelberg game tree and subgames. The rangers have one information state and the poachers have two information states ($P.1$ and $P.2$). The payoffs are shown at the terminal nodes with the rangers' payoffs first. There are 3 subgames and the thick lines indicate the backward induction process for finding the SPNE.

The game can also be written as the normal form representation in Table 1 and the SPNE is a subset of the NE for the normal form game. Finding all NE for this normal form game includes the SPNE above, as well as an equilibrium where at information state $P.2$ the poachers execute a mixed-strategy by choosing each cell with probability 0.5. If the roles were reversed and the rangers were to act as the follower then the SPNE is $\hat{S}^R = (1, 2), \hat{S}^P = 2$ so the rangers and the poachers both go to cell 2 and the payoffs are 3 for the rangers and -2 for the poachers. This example thus shows a follower's advantage since both agents receive a higher payoff when they follow than when they lead. The use of the SPNE is just for demonstration in this article. In practice, it would be preferred to solve for an optimal mixed-strategy for the rangers, which is a probability distribution over their set of pure-strategies. This mixed-strategy could then be utilised, for example, over the duration of one month in which a random strategy is selected from this distribution every day.

Table 1. Normal form of the Stackelberg game. The rangers are the row player and have two pure-strategies. The poachers are the column player and have 4 pure-strategies, where (i, j) means i at $P.1$ and j at $P.2$. The body of the table shows the payoffs at each combination of their strategies, where the rangers' payoffs are given first.

	P			
R	(1, 1)	(1, 2)	(2, 1)	(2, 2)
1	4, -3	4, -3	1, 1	1, 1
2	-1, 1	3, -2	-1, 1	3, -2

5 Experiments

We perform simulations for the subarea in Figure 1, using the grid shown in Figure 2. The poachers' entry is at the border cells and their preferences are as described in Section 3.2. The wildlife is set to dislike being 1 km near to camps and prefer being within 10 km of dams and water. There are 566 allowed cells for each agent, after excluding geographical obstacles. Simulations are run with 200 moves per game, 10 games per month, and for 500 months. GAME1 is the uniform game, with random movement for all agents. GAME2 is the null game, with random movement for the rangers, intelligent movement for the poachers and improved animal movement. The poach frequency per day and arrest frequency per day are the primary measures for assessing the rangers' performance. We also consider the average number of moves to make an arrest and the average distance between the rangers and the poachers when there are no arrests. Since the distributions of these measures are skewed, we report on the median and calculate the bootstrap standard error (SE) of the median using 1 000 bootstrap samples. Furthermore, we do pairwise comparisons of the games for each measure and test the general hypothesis of identical populations using Mood's median test [11]. Table 2 shows the median for each measure, with bootstrap SE of the median in brackets, and the superscripts indicate where Mood's median test is non-significant. As can be expected, the random rangers have poorer performance against the intelligent poachers than against the random poachers since the poach frequency per day is significantly higher in GAME2 than in GAME1. Figure 6 shows the trajectories for a single round of the null game.

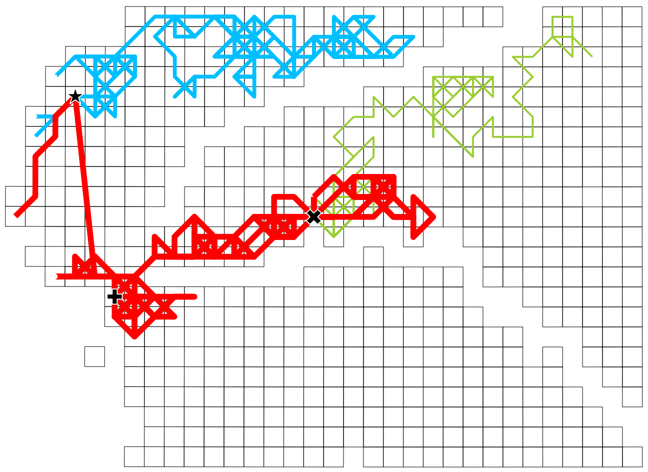

Figure 6. Trajectories for a single round of the null game. The lines represent the movements of the wildlife (green, narrow line), the rangers (blue, medium line) and the poachers (red, thick line). A black * represents a capture event, a black X represents a poaching event and a black + represents a leaving event.

Next, we implement the pure-strategy Stackelberg Game, as described in Section 4. A `Game` class was created with methods to calculate the strategies, payoffs, and game solution. The rangers are set to like being within 10 km of dams and water since that is likely where the wildlife can be found. The poachers' preferences are as described in Section 3.2. For each agent, based on the agent's geographical preferences, each allowed cell has a selection weight which aids in determining the movement into the next cell and a utility which aids in selecting the destination cell. The wildlife is set to dislike being 1 km near to camps and like being within 10 km of dams and water. Figure 7 shows a colour scale of the wildlife preferences as well as a simulated collection of wildlife sightings that serves as an animal density estimate. The payoffs for the rangers and the poachers are calculated using the animal densities (with a weight of 5), their utility based on geographical preferences (with a weight of 2) and arrests when they are in the same cell (with a weight of 40). We perform the following simulations of the pure-strategy Stackelberg game:

- GAME3 the rangers as the Stackelberg leader against the intelligent poachers;
- GAME4 the rangers as the Stackelberg leader against the poachers as the Stackelberg follower;
- GAME5 the rangers as the Stackelberg follower against the intelligent poachers; and
- GAME6 the rangers as the Stackelberg follower against the poachers as the Stackelberg leader.

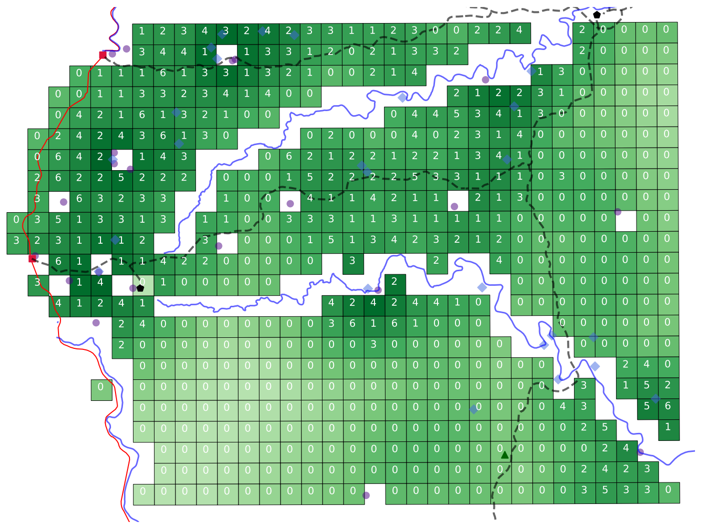

Figure 7. Wildlife density estimates and cell selection weights. The numbers indicate how many wildlife have been sighted in each cell and the selection weights are depicted by a colour scale where darker colours represent higher values.

Table 2. Median and bootstrap SE in brackets for evaluation measures. A superscript i means the null hypothesis for Mood's median test that the medians are the same is not rejected, at a significance level of $\alpha = 0.05$, when compared pairwise with GAMEi.

	Poach Freq per Day	Arrest Freq per Day	Ave Moves for Arrests	Ave Distance (km) for Non-arrests
GAME1	0.067 (0.006) ^{4,6}	0.067 (0.001) ^{2,3}	78.0 (2.1)	10.4 (0.1)
GAME2	0.083 (0.016) ^{3,5}	0.067 (0.001) ^{1}	67.6 (2.7) ^{3,4,5}	11.8 (0.1) ^{3,6}
GAME3	0.067 (0.016) ^{2,5}	0.067 (0.013) ^{1}	66.8 (2.4) ^{2,4,5}	11.4 (0.2) ^{2,4,6}
GAME4	0.033 (0.015) ^{1,6}	0.167 (0.008) ^{5}	62.8 (2.3) ^{2,3,5}	11.1 (0.2) ^{3,6}
GAME5	0.067 (0.005) ^{2,3}	0.133 (0.016) ^{4}	67.4 (1.2) ^{2,3,4}	9.5 (0.1)
GAME6	0.067 (0.005) ^{1,4}	0.267 (0.016)	49.8 (1.5)	11.6 (0.2) ^{2,3,4}

The results for the Stackelberg games are shown in Table 2. When comparing GAME3 and GAME5 with GAME2 (null game), we have a direct comparison of the rangers' performance since the only difference between the games is the movement of the rangers. The poach frequency per day is lower in GAME3 and GAME5 than in GAME2, and the arrest frequency per day is significantly higher in GAME5 than in GAME2. Thus, the rangers perform better than random when playing the Stackelberg game as the leader or the follower against the intelligent poachers. When comparing GAME3 with GAME4, where the rangers act as the Stackelberg leader, they perform better in GAME4 since the poach frequency per day is significantly lower and the arrest frequency per day is significantly higher. Thus, comparison of the rangers' performance against the intelligent poachers (GAME3) represents a worse case for the rangers than against the poachers as the Stackelberg follower (GAME4). This is reasonable when trying to select the better game since we would not want to have an optimistic estimate of their performance. Similarly, when comparing GAME5 and GAME6, where the rangers acts as the Stackelberg follower, GAME5 against the intelligent poacher represents a worse case for the rangers than GAME6 against the poachers as the Stackelberg leader since the arrest frequency per day is much higher and the average moves for an arrest is much lower in GAME6. Comparing GAME4 and GAME6, there is no significant difference in poach frequency per day but GAME6 has a much higher arrest frequency per day and a much lower average number of moves for arrests. Thus, when both agents act according to the Stackelberg game, the rangers perform better when acting as the follower than as the leader.

6 Conclusions

The null game presented in this paper provides a realistic baseline model for assessing any improvement in the rangers' performance. Improved ranger performance is defined as having fewer wildlife poached and more poachers arrested. The primary performance measures of poach frequency per day and arrest frequency per day thus directly address the objectives of the rangers. As expected, the rangers have poorer performance against the intelligent poachers than against the random poachers in the null game. Implementing the simple Stackelberg security game shows that even this simple game-theoretic algorithm results in a significant improvement for the rangers. An Appendix is provided in Section 7 containing a summary of the classes, attributes and simulation parameters required for the null game and the pure-strategy Stackelberg games.

Utilising better geographic data is expected to alleviate the problem with back and forth movement around cells that are excluded due to geographical obstacles. The next step would be to incorporate time into the simulation. For example, the poachers' re-entry into the park after an arrest could be delayed; the visibility of the agents could be increased during dry seasons or nights when it is full moon; rivers might be easily crossed during dry seasons; and where there is dense vegetation or steep mountains the speed of the agents could be decreased within that region instead of excluding those cells.

Further improvements can be implemented to make the null game more realistic. Including multiple groups of rangers, multiple groups of poachers, and multiple herds of wildlife would be a valuable improvement. Utilising an animal movement model for different types of wildlife instead of simulating the movements of the wildlife could also improve the framework considerably. Alternatively, we could design routes for each of the wildlife, the poachers, and the rangers using imaging software to identify sand trails [15], using routes uncovered by poacher tracking [8], or using road segments to define routes [9]. The routes can then be used for their movement within the framework and as their set of strategies in the GSG algorithm.

Since the framework is designed to compare and evaluate different wildlife security games, another task would be to include game-theoretic algorithms discussed in current research within the framework. We would like to test the idea of the rangers acting as the Stackelberg follower against the current algorithms. Observed data can be utilised in a Bayesian network to learn the poachers' mixed-strategy and the rangers' mixed-strategy best response can be calculated.

7 Appendix

The evaluation framework is developed in Python 3.8. It utilises the `geopandas` [1] library to handle geographical information and the `movingpandas` [12] library to store, plot and analyse the trajectories in each simulation. Table 3 provides a summary of the classes, attributes and simulation parameters used in the framework. For the uniform game, the `move_type` is set to `"random"` for all agents but for the null game it is set to `"intelligent"` for the poachers. The `Park` class has methods to calculate the grid, the cells on the edge of the grid, and the cells on the park border. The `Agent` class has methods to find the agent's allowed cells and calculate their geographical selection weights and utilities. Furthermore, it contains methods to find their start cell, destination cell and calculate the next cell to move into. The `Game` class is more useful for the security games as it contains methods to calculate strategies, payoffs, and the game solution. For the null game, it just collects the agents and whether the poachers' entry point should be a cell on the edge of the grid or on the border of the park. The `Sim` class has methods to simulate a single game and to simulate games for a number of months. Additionally, to evaluate the simulations, there are methods for calculating the median of the performance measures, bootstrap standard errors of the median and p-values for Mood's median test.

Table 3. Summary of classes, attributes and simulation parameters.

Class	Attribute	Description
`Park`	`boundary`	`GeoDataFrame` of park boundary (polygon)
	`trees`	`GeoDataFrame` of areas with dense vegetation (polygons)
	`mountains`	`GeoDataFrame` of areas with steep mountains (polygons)
	`roads`	`GeoDataFrame` of main roads (lines)
	`rivers`	`GeoDataFrame` of main rivers (lines)
	`dams`	`GeoDataFrame` of dams (points)
	`water`	`GeoDataFrame` of water holes, fountains and drinking troughs (points)
	`camps`	`GeoDataFrame` of main rest camps (points)
	`picnic`	`GeoDataFrame` of picnic spots (points)
	`gates`	`GeoDataFrame` of public gates (points)
	`custom`	dictionary of the form `{"name": GeoDataFrame}`
	`subarea`	latitude and longitude bounds of park subarea
	`cell_x_length`	horizontal length of grid cells (meters)
	`cell_y_length`	vertical length of grid cells (meters)
`Agent`	`name`	name of the agent
	`park`	`park` object
	`grid_type`	`"full" / "bounded"`
	`move_type`	`"random" / "intelligent" / "game"`
	`area_out`	list of feature names for areas to stay out of
	`area_within`	list of feature names for areas to stay within
	`dislikes`	dictionary of feature names and distances (meters) to stay away, eg. `{"name": distance}`
	`likes`	dictionary of feature names and distances (meters) to stay near, eg. `{"name": distance}`
`Game`	`name`	name of the game
	`wildlife`	`agent` object for wildlife herd
	`ranger`	`agent` object for rangers
	`poacher`	`agent` object for poachers
	`poacher_entry`	`"edge" / "border"`
	`game_type`	`"null" / "stackel_lead" / "stackel_follow"`
	`sightings`	`GeoDataFrame` of wildlife sightings (points)
`Sim`	`game`	`game` object to simulate
	`seed`	sets the random seed
	`end_moves`	number of moves until the game ends
	`games_pm`	number of games to simulate per month
	`months_total`	number of months to simulate

References

1. GeoPandas 0.7.0, https://geopandas.org/
2. SANParks Data Repository, http://dataknp.sanparks.org/sanparks/
3. Minister Molewa highlights progress on Integrated Strategic Management of Rhinoceros (2017), https://www.environment.gov.za/mediarelease/molewa_progressonintegrated_strategicmanagement_ofrhinoceros
4. Department of Environment, Forestry and Fisheries report back on rhino poaching in South Africa in 2019 (2020), https://www.environment.gov.za/mediarelease/reportbackon2019_rhinopoachingstatistics
5. An, B., Tambe, M.: Stackelberg security games (SSG) basics and application overview. In: Abbas, A.E., Tambe, M., von Winterfeldt, D. (eds.) Improving Homeland Security Decisions, pp. 485–507. Cambridge University Press, Cambridge (2017), https://doi.org/10.1017/9781316676714.021
6. Conitzer, V., Sandholm, T.: Computing the Optimal Strategy to Commit to. In: Proceedings of the 7th ACM conference on Electronic commerce. pp. 82–90 (2006), https://dl.acm.org/doi/10.1145/1134707.1134717
7. Cournot, A.A.: Researches Into the Mathematical Principles of the Theory of Wealth. Macmillan, New York (1897), https://www3.nd.edu/~tgresik/IO/Cournot.pdf
8. De Oude, P., Pavlin, G., De Villiers, J.P.: High-Level Tracking Using Bayesian Context Fusion. In: FUSION 2018, 21st International Conference on Information Fusion. pp. 1415–1422. IEEE (2018), https://doi.org/10.23919/ICIF.2018.8455342
9. Fang, F., Nguyen, T.H., Pickles, R., Lam, W.Y., Clements, G.R., An, B., Singh, A., Tambe, M., Lemieux, A.: Deploying PAWS: field optimization of the protection assistant for wildlife security. In: AAAI 2016, Proceedings of the 30th AAAI Conference on Artificial Intelligence. pp. 3966–3973. AAAI Press (2016), https://dl.acm.org/doi/10.5555/3016387.3016464
10. Fang, F., Stone, P., Tambe, M.: When security games go green: designing defender strategies to prevent poaching and illegal fishing. In: IJCAI 2015, Proceedings of the 24th International Joint Conference on Artificial Intelligence. pp. 2589–2595. AAAI Press (2015), https://dl.acm.org/doi/10.5555/2832581.2832611
11. Gibbons, J.D., Chakraborti, S.: Nonparametric Statistical Inference. CRC Press, Boca Raton, 4 edn. (2003), https://doi.org/10.4324/9780203911563
12. Graser, A.: MovingPandas: Efficient Structures for Movement Data in Python. Journal of Geographic Information Science **7**(1), 54–68 (2019), https://doi.org/10.1553/giscience2019_01_s54
13. Kar, D., Fang, F., Fave, F.D., Sintov, N., Tambe, M.: "A game of thrones": When human behavior models compete in repeated stackelberg security games. In: AAMAS 2015, Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems. pp. 1381–1390. IFAAMAS (2015), https://dl.acm.org/doi/10.5555/2772879.2773329
14. Kirkland, L., de Waal, A., de Villiers, J.P.: Simulating Null Games for Uncertainty Evaluation in Green Security Games. In: FUSION 2019, 22nd International Conference on Information Fusion. pp. 1–8. IEEE (2019), https://ieeexplore.ieee.org/document/9011280
15. Lemieux, A.M. (ed.): Situational Prevention of Poaching. Routledge, London (2014), https://doi.org/10.4324/9780203094525
16. Myerson, R.B.: Game theory: analysis of conflict. Harvard University Press, Cambridge (1991), https://doi.org/10.2307/j.ctvjsf522

17. Nguyen, T.H., Sinha, A., Gholami, S., Plumptre, A., Joppa, L., Tambe, M., Driciru, M., Wanyama, F., Rwetsiba, A., Critchlow, R., Beale, C.M.: CAPTURE: A new predictive anti-poaching tool for wildlife protection. In: AAMAS 2016, Proceedings of the 15th International Conference on Autonomous Agents & Multiagent Systems. pp. 767–775. IFAAMAS (2016), https://dl.acm.org/doi/abs/10.5555/2936924.2937037
18. Paruchuri, P., Pearce, J.P., Marecki, J., Tambe, M., Ordonez, F., Kraus, S.: Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games. In: AAMAS 2008, Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems. pp. 895–902. IFAAMAS (2008), https://dl.acm.org/doi/10.5555/1402298.1402348
19. Pita, J., Jain, M., Tambe, M., Ordóñez, F., Kraus, S.: Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition. Artificial Intelligence **174**(15), 1142–1171 (2010), https://doi.org/10.1016/j.artint.2010.07.002
20. von Stackelberg, H.: Market Structure and Equilibrium. Springer, Berlin (2011), https://doi.org/10.1007/978-3-642-12586-7
21. Tambe, M.: Security and game theory: Algorithms, deployed systems, lessons learned. Cambridge University Press, Cambridge (2011), https://doi.org/10.1017/CBO9780511973031
22. Yadav, A., Nguyen, T.H., Fave, F.D., Tambe, M., Agmon, N.: Handling Payoff Uncertainty with Adversary Bounded Rationality in Green Security Domains. In: BECIS 2015, Workshop on Behavioral, Economic and Computational Intelligence for Security. pp. 1–16 (2015), https://teamcore.seas.harvard.edu/publications/handling-payoff-uncertainty-green-security-domains-adversary-bounded-0
23. Yang, R., Ford, B., Tambe, M., Lemieux, A.: Adaptive resource allocation for wildlife protection against illegal poachers. In: AAMAS 2014, Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems. pp. 453–460. IFAAMAS (2014), https://dl.acm.org/doi/10.5555/2615731.2615805
24. Yin, Z., Korzhyk, D., Kiekintveld, C., Conitzer, V., Tambe, M.: Stackelberg vs. Nash in security games: Interchangeability, equivalence, and uniqueness. AAMAS2010, Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems **2**, 1139–1146 (2010), https://dl.acm.org/doi/10.5555/1838206.1838360