Deploying PAWS: Field Optimization of the Protection Assistant for Wildlife Security

Fei Fang[1], Thanh H. Nguyen[1], Rob Pickles[2], Wai Y. Lam[2,3], Gopalasamy R. Clements[2,3,6], Bo An[4], Amandeep Singh[5], Milind Tambe[1], Andrew Lemieux[7]

[1]{feifang,thanhhng,tambe}@usc.edu, University of Southern California, USA
[2]rpickles@panthera.org, Panthera, USA, [3]waiyeee@gmail.com, Rimba, Malaysia
[4]boan@ntu.edu.sg, Nanyang Technological University, Singapore, [5]as4330@columbia.edu, Columbia University, USA
[6]reuben@myrimba.org, Kenyir Research Institute, Universiti Malaysia Terengganu, Malaysia
[7]alemieux@nscr.nl, The Netherlands Institute for the Study of Crime and Law Enforcement (NSCR), Netherlands

Abstract

Poaching is a serious threat to the conservation of key species and whole ecosystems. While conducting foot patrols is the most commonly used approach in many countries to prevent poaching, such patrols often do not make the best use of limited patrolling resources. To remedy this situation, prior work introduced a novel emerging application called PAWS (Protection Assistant for Wildlife Security); PAWS was proposed as a game-theoretic ("security games") decision aid to optimize the use of patrolling resources.

This paper reports on PAWS's significant evolution from a proposed decision aid to a regularly deployed application, reporting on the lessons from the first tests in Africa in Spring 2014, through its continued evolution since then, to current regular use in Southeast Asia and plans for future worldwide deployment. In this process, we have worked closely with two NGOs (Panthera and Rimba) and incorporated extensive feedback from professional patrolling teams. We outline key technical advances that lead to PAWS's regular deployment: (i) incorporating complex topographic features, e.g., ridgelines, in generating patrol routes; (ii) handling uncertainties in species distribution (game theoretic payoffs); (iii) ensuring scalability for patrolling large-scale conservation areas with fine-grained guidance; and (iv) handling complex patrol scheduling constraints.

Introduction

There is an urgent need to protect wildlife from poaching. Indeed, poaching can lead to extinction of species and destruction of ecosystems. For example, poaching is considered a major driver (Chapron et al. 2008) of why tigers are now found in less than 7% of their historical range (Sanderson et al. 2006), with three out of nine tiger subspecies already extinct (IUCN 2015). As a result, efforts have been made by law enforcement agencies in many countries to protect endangered animals; the most commonly used approach is conducting foot patrols. However, given their limited human resources, improving the efficiency of patrols to combat poaching remains a major challenge.

To address this problem, prior work introduced a novel emerging application called PAWS (Protection Assistant for Wildlife Security) (Yang et al. 2014); PAWS is proposed as a game-theoretic decision-aid to optimize the use of human patrol resources to combat poaching. PAWS is an application in the general area of "security games" (Tambe 2011); security-game-based decision support systems have previously been successfully deployed in the real-world in protecting critical infrastructure such as airports, flights, ports, and metro trains. PAWS was inspired by this success, and was the first of a new wave of proposed applications in the subarea now called "green security games" (Fang, Stone, and Tambe 2015; Kar et al. 2015). Specifically, PAWS solves a *repeated* Stackelberg security game, where the patrollers (defenders) conduct randomized patrols against poachers (attackers), while balancing the priorities of different locations with different animal densities. Despite its promise, *the initial PAWS effort did not test the concept in the field.*

This paper reports on PAWS's significant evolution over the last two years from a proposed decision aid to a regularly deployed application. We report on the innovations made in PAWS and lessons learned from the first tests in Uganda in Spring 2014, through PAWS's continued evolution to current regular use in Malaysia (in collaboration with two Non-Governmental Organizations: Panthera and Rimba). Indeed, the first tests revealed key shortcomings in PAWS's initial algorithms and assumptions (we will henceforth refer to the initial version of PAWS as *PAWS-Initial*, and to the version after our enhancement as PAWS). First, a major limitation, the severity of which was completely unanticipated, was that *PAWS-Initial* ignored topographic information. Yet in many conservation areas, high changes in elevation and the existence of large water bodies may result in a big difference in the effort needed for patrollers' movement. These factors also have a direct effect on poachers' movement. Second, *PAWS-Initial* assumed animal density and relevant problem features at different locations to be known. However, in practice, there are uncertainties in the payoffs of different locations, due to uncertainty over animal movement. Not considering such uncertainty may lead to high degradation in patrol quality. Third, *PAWS-Initial* could not scale to provide detailed patrol routes in large conservation areas. Detailed routes require fine-grained discretization, which leads to a large number of feasible patrol routes. Finally, *PAWS-Initial* failed to consider patrol scheduling constraints.

In this paper, we outline novel research advances which remedy the aforementioned limitations, making it possible to deploy PAWS on a regular basis. First, we incorporate elevation information and land features and use a novel hierarchical modeling approach to build a virtual "street map" of the conservation area. This virtual "street map" helps scale-up while providing fine grained guidance, and is an innovation that would be useful in many other domains requiring patrolling of large areas. Essentially, the street map connects the whole conservation area through easy-to-follow route segments such as ridgeline, streams and river banks. The rationale for this comes from the fact that animals, poachers, and patrollers all use these features while moving. To address the second and third limitations, we build on the street map concept with a novel algorithm that uniquely synthesizes two threads of prior work in the security games literature; specifically, the new PAWS algorithm handles payoff uncertainty using the concept of minimax regret (Nguyen et al. 2015), while simultaneously ensuring scalability – using our street maps – via the cutting plane framework (Yang et al. 2013). To address the final limitation, we incorporate in PAWS's algorithm the ability to address constraints such as patrol time limit and starting and ending at the base camp. In the final part of the paper, we provide detailed information about the regular deployment of PAWS.

Background and Related Work

Criminologists have begun to work on the problem of combating poaching, from policy design to illegal trade prevention (Lemieux 2014). Geographic Information Systems (GIS) experts (Hamisi 2008) and wildlife management staff (Wato, Wahungu, and Okello 2006) have carefully studied the identification of poaching hotspots. In recent years, software tools such as SMART (SMART 2013), MIST (Stokes 2010) have been developed to help conservation managers record data and analyze patrols retrospectively. We work on a complementary problem of optimizing the patrol planning of limited security staff in conservation areas.

In optimizing security resource allocation, previous work on *Stackelberg Security Games (SSGs)* has led to many successfully deployed applications for security of airports, ports and flights (Pita et al. 2008; Fang, Jiang, and Tambe 2013). Based on the early work on SSG, recent work has focused on *green security games* (Kar et al. 2015), providing conceptual advances in integrating learning and planning (Fang, Stone, and Tambe 2015) and the first application to wildlife security *PAWS-Initial*. *PAWS-Initial* (Yang et al. 2014) models the interaction between the patroller (defender) and the poacher (attacker) who places snares in the conservation area (see Figure 1) as a basic green security game, i.e., a repeated SSG, where every few months, poaching data is analyzed and a new SSG is setup enabling improved patrolling strategies. The deployed version of PAWS adopts this framework.

We provide a brief review of SSGs, using PAWS as a key example. In SSGs, the defender protects T targets from an adversary by optimally allocating a set of R resources ($R < T$) (Pita et al. 2008). In PAWS, the defender discretizes the conservation area into a grid, where each grid cell is viewed as a target for poachers, to be protected by a

Figure 1: A picture of a snare placed by poachers.

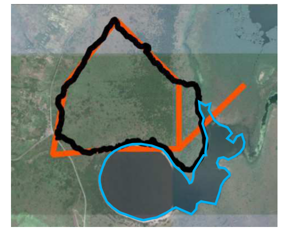


Figure 2: One patrol route during the test in Uganda.

set of patrollers. The defender's pure strategy is an assignment of the resources to targets. The defender can choose a mixed strategy, which is a probability distribution over pure strategies. The defender strategy can be compactly represented as a coverage vector $\mathbf{c} = \langle c_i \rangle$ where c_i is the coverage probability, i.e., the probability that a defender resource is assigned to be at target i (Korzhyk, Conitzer, and Parr 2010). The adversary observes the defender's mixed strategy through surveillance and then attacks a target. An attack could refer to the poacher, a snare, or some other aspect facilitating poaching (e.g., poaching camp). Each target is associated with payoff values which indicate the reward and penalty for the players. If the adversary attacks target i, and i is protected by the defender, the defender gets reward $U^d_{r,i}$ and the adversary receives penalty $U^a_{p,i}$. Conversely, if not protected, the defender gets penalty $U^d_{p,i}$ and the adversary receives reward $U^a_{r,i}$. Given a defender strategy $\mathbf{c}$, the players' expected utilities when target i is attacked are:

$$U^a_i = c_i U^a_{p,i} + (1 - c_i) U^a_{r,i} \tag{1}$$

$$U^d_i = c_i U^d_{r,i} + (1 - c_i) U^d_{p,i} \tag{2}$$

The game in PAWS is zero-sum, $U^d_{r,i} = -U^a_{p,i}$, $U^d_{p,i} = -U^a_{r,i}$. $U^a_{r,i}$ is decided by animal density – higher animal density implies higher payoffs.

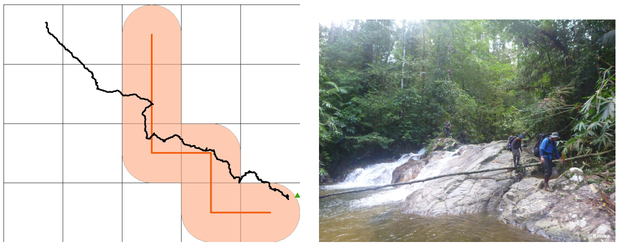
In SSGs, the adversary's behavior model decides his response to the defender's mixed strategy. Past work has often assumed that the adversary is perfectly rational, choosing a single target with the highest expected utility (Pita et al. 2008). PAWS is the first deployed application that relaxes this assumption in favor of a bounded rationality model called SUQR, which models the adversary's stochastic response to defender's strategy (Nguyen et al. 2013). SUQR was shown to perform the best in human subject experiments when compared with other models. Formally, SUQR predicts the adversary's probability of attacking i as follows:

$$q_i = \frac{e^{w_1 c_i + w_2 U^a_{r,i} + w_3 U^a_{p,i}}}{\sum_j e^{w_1 c_j + w_2 U^a_{r,j} + w_3 U^a_{p,j}}} \tag{3}$$

where (w_1, w_2, w_3) are parameters indicating the importance of three key features: the coverage probability and the attacker's reward and penalty. The parameters can be learned from data.

First Tests and Feedback

We first tested *PAWS-Initial* (Yang et al. 2014) at Uganda's Queen Elizabeth National Park (QENP) for 3 days. Sub-



(a) Deployed route (b) Patrollers

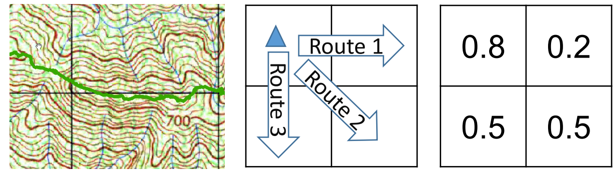
Figure 3: First 4-day patrol in Malaysia. Figure 3(a) shows one suggested route (orange straight lines) and the actual patrol track (black line). Figure 3(b) shows the patrollers walking along the stream during the patrol.

sequently, with the collaboration of Panthera and Rimba, we started working in forests in Malaysia since September 2014[1]. These protected forests are home to endangered animals such as the Malayan Tiger and Asian Elephant, but are threatened by poachers. One key difference of this site compared to QENP is that there is high changes in elevation and the terrain is much more complex. The first 4-day patrol in Malaysia was conducted in November 2014. These initial tests revealed four areas of shortcomings, which restricted *PAWS-Initial* from being used regularly and widely.

The first limitation, which was surprising given that it has received no attention in previous work on security games, is the critical importance of topographic information that was ignored in *PAWS-Initial*. Topography can affect patrollers' speed in key ways. For example, lakes are inaccessible for foot patrols. Not considering such information may lead to the failure of completing the patrol route. Figure 2 shows one patrol route during the test in Uganda. The suggested route (orange straight line) goes across the water body (lower right part of figure), and hence the patrollers decided to walk along the water body (black line). Also, changes in elevation requires extra patrol effort and extreme changes may stop the patrollers from following a route. For example, in Figure 3(a) [Malaysia], *PAWS-Initial* planned a route on a 1km by 1km grid (straight lines), and suggested that the patrollers walk to the north area (Row 1, Column 3) from the south side (Row 2, Column 3). However, such movement was extremely difficult because of the high changes in elevation. So patrollers decided to head towards the northwest area as the elevation change is more gentle. In addition, it is necessary to focus on terrain features such as ridgelines and streams (Figure 3(b)) when planning routes for three reasons: (i) they are important conduits for certain mammal species such as tigers; (ii) hence, poachers use these features for trapping and moving about in general; (iii) patrollers find it easier to move around here than on slopes. Figure 4(a) shows a prominent ridgeline.

The second limitation is that *PAWS-Initial* assumes the payoff values of the targets — e.g., $U^a_{r,i}$ – are known and fixed. In the domain of wildlife protection, there can be uncertainties due to animal movement and seasonal changes. Thus, considering payoff uncertainty is necessary for optimizing patrol strategy.

[1]For security of animals and patrollers, no latitude/longitude information is presented in this paper.



(a) Ridgeline (b) Feasible routes (c) Coverage

Figure 4: Illustrative examples.

The third limitation is that *PAWS-Initial* cannot scale to provide detailed patrol routes in large conservation areas, which is necessary for successful deployment. Detailed routes requires fine-grained discretization, which leads to an exponential number of routes in total.

The fourth limitation is that *PAWS-Initial* considers covering individual grid cells, but not feasible routes. In practice, the total patrolling time is limited and the patrollers can move to nearby areas. A patrol strategy for implementation should be in the form of a distribution over feasible patrol routes satisfying these constraints. Without taking these scheduling (routing) constraints into account, the optimal coverage probabilities calculated by *PAWS-Initial* may not be implementable. Figure 4(b) shows an example area that is discretized into four cells and the base camp is located at the upper left cell. There are three available patrol routes, each protecting two targets. The coverage probabilities shown in Figure 4(c) cannot be achieved by a randomization over the three routes because the coverage of the upper left cell (Target 1) should be no less than the overall coverage of the remaining three cells since all routes start from the base camp.

PAWS Overview and Game Model

Figure 5 provides an overview of the deployed version of PAWS. PAWS first takes the input data and estimates the animal distribution and human activity distribution. Based on this information, an SSG based game model is built and the patrol strategy is calculated. In wildlife protection, there are repeated interaction between patrollers and poachers. When patrollers execute the patrol strategy generated by PAWS over a period (e.g., three months), more information is collected and can become part of the input in the next round.

PAWS provides significant innovations in addressing the aforementioned limitations of *PAWS-Initial*. In building the game model, PAWS uses a novel hierarchical modeling approach to build a virtual street map, while incorporating detailed topographic information. PAWS models the poachers bounded rationality as described by the SUQR model and considers uncertainty in payoff values. In calculating the patrol strategy, PAWS uses ARROW (Nguyen et al. 2015) algorithm to deal with payoff uncertainty and adopts cutting plane approach and column generation to address the scalability issue introduced by scheduling constraints.

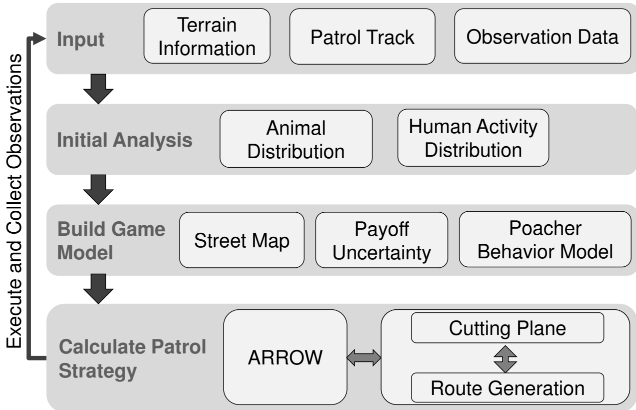


Figure 5: PAWS Overview

Input and Initial Analysis

The input information includes contour lines which describe the elevation, terrain information such as lakes and drainage, base camp locations, previous observations (animals and human activities), as well as previous patrol tracks. However, the point detections of animal and human activity presence are not likely to be spatially representative. As such, it is necessary to predict the animal and human activity distribution over the entire study area. To this end, we used: 1) JAGS (Plummer 2003) to produce a posterior predictive density raster for tigers (as a target species) derived from a spatially explicit capture-recapture analysis conducted in a Bayesian framework; and 2) MaxEnt (Phillips, Anderson, and Schapire 2006) to create a raster of predicted human activity distribution based on meaningful geographical covariates (e.g., distance to water, slope, elevation) in a Maximum Entropy Modelling framework.

Build Game Model

Based on the input information and the estimated distribution, we build a game model abstracting the strategic interaction between the patroller and the poacher as an SSG. Building a game model involves defender action modeling, adversary action modeling, and payoff modeling. We will discuss all three parts but emphasize defender action modeling since this is one of the major challenges to bring PAWS to a regularly deployed application. Given the topographic information, modeling defender actions in PAWS is far more complex than any other previous security game domain.

Defender Action Modeling Based on the feedback from the first tests, we aim to provide detailed guidance to the patrollers. If we use a fine-grained grid and treat every fine-grained grid cell as a target, computing the optimal patrolling strategy is exceptionally computationally challenging due to the large number of targets and the exponential number of patrol routes. Therefore, a key novelty of PAWS is to provide a hierarchical modeling solution, the first such model in security game research. This hierarchical modeling approach allows us to attain a good compromise between scaling up and providing detailed guidance. This approach would be applicable in many other domains for large open area patrolling where security games are applicable, not only other green security games applications, but others including patrolling of large warehouse areas or large open campuses via robots or UAVs.

More specifically, we leverage insights from hierarchical abstraction for heuristic search such as path planning (Botea, Mller, and Schaeffer 2004) and apply two levels of discretization to the conservation area. We first discretize the conservation area into 1km by 1km *Grid Cells* and treat every grid cell as a target. We further discretize the grid cells into 50m by 50m *Raster Pieces* and describe the topographic information such as elevation in 50m scale. The defender actions are patrol routes defined over a virtual "street map" – which is built in the terms of raster pieces while aided by the grid cells in this abstraction as described below. With this hierarchical modeling, the model keeps a small number of targets and reduces the number of patrol routes while allowing for details at the 50m scale.

The street map is a graph consisting of nodes and edges, where the set of nodes is a small subset of the raster pieces and edges are sequences of raster pieces linking the nodes. We denote nodes as Key Access Points (KAPs) and edges as route segments. The street map not only helps scalability but also allows us to focus patrolling on preferred terrain features such as ridgelines. The street map is built in three steps: (i) determine the accessibility type for each raster piece, (ii) define KAPs and (iii) find route segments to link the KAPs.

In the first step, we check the accessibility type of every raster piece. For example, raster pieces in a lake are inaccessible, whereas raster pieces on ridgelines or previous patrol tracks are easily accessible. Ridgelines and valley lines are inferred from the contour lines using existing approaches in hydrology (Tarboton, Bras, and Rodriguez-Iturbe 2007).

The second step is to define a set of KAPs, via which patrols will be routed. We want to build the street map in such a way that each grid cell can be reached. So we first choose raster pieces which can serve as entries and exits for the grid cells as KAPs, i.e., the ones that are on the boundary of grid cells and are easily accessible. In addition, we consider existing base camps and mountain tops as KAPs as they are key points in planning the patroller's route. We choose additional KAPs to ensure KAPs on the boundary of adjacent cells are paired. Figure 6 shows identified KAPs and easily accessible pieces (black and grey raster pieces respectively).

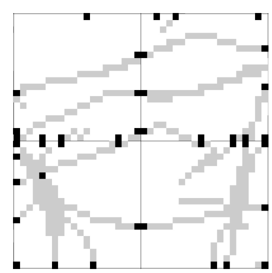


Figure 6: KAPs (black) for 2 by 2 grid cells.

The last step is to find route segments to connect the KAPs. Instead of inefficiently finding route segments to connect each pair of KAPs on the map globally, we find route segments locally for each pair of KAPs within the same grid cell, which is sufficient to connect all the KAPs. When finding the route segment, we design a distance measure which estimates the actual patrol effort and also gives high priority to the preferred terrain features. The effort needed for three-dimensional movement can be interpreted as the equivalent distance on flat terrain. For example, for gentle slopes, equivalent "flat-terrain" distance is obtained by adding 8km for every 1km of elevation ascent according to Naismith's rule (Thompson 2011). In PAWS, we apply Naismith's rule with Langmuir corrections (Langmuir 1995) for gentle slopes ($< 20°$) and apply Tobler's hiking speed function (Tobler 1993) for steep slopes ($\geq 20°$). Very steep slopes ($> 30°$) are not allowed. We penalize not walking on preferred terrain features by adding extra distance. Given the distance measure, the route segment is defined as the shortest distance path linking two KAPs within the grid cell.

The defender's pure strategy is defined as a patrol route on the street map, starting from the base camp, walking along route segments and ending with base camp, with its total distance satisfying the patrol distance limit (all measured as distance on flat terrain). The patroller confiscates the snares along the route and thus protects the grid cells. More specifically, if the patroller walks along a route segment which covers sufficiently large portion (e.g., 50% of animal distribution) of a grid cell, the cell is considered to be protected. The defender's goal is to find an optimal mixed patrol strategy — a probability distribution over patrol routes.

Poacher Action Modeling and Payoff Modeling The poacher's actions are defined over the grid cells to aid scalability. In this game, we assume the poacher can observe the defender's mixed strategy and then chooses one target (a grid cell) and places snares in this target. Following earlier work, the poacher in this game is assumed to be boundedly rational and his actions can be described by the SUQR model.

Each target is associated with payoff values indicating the reward and penalty for the patrollers and the poachers. As mentioned earlier, PAWS models a zero-sum game and the reward for the attacker (and the penalty for the defender) is decided by the animal distribution. However, in this game model, we need to handle uncertainty in the players' payoff values since key domain features such as animal density which contribute to the payoffs are difficult to precisely estimate. In addition, seasonal or dynamic animal migration may lead to payoffs to become uncertain in the next season. We use intervals to represent payoff uncertainty in PAWS; the payoffs are known to lie within a certain interval whereas the exact values are unknown. Interval uncertainty is in fact a well-known concept to capture uncertainty in security games (Nguyen et al. 2014; 2015). We determine the size of the payoff intervals at each grid cell based on patrollers' patrol efforts at that cell. Intuitively, if the patrollers patrol a cell more frequently, there is less uncertainty in the players' payoffs at that target and thus a smaller size of the payoff intervals.

Calculate Patrol Strategy

We build on algorithms from the rich security game literature to optimize the defender strategy. However, we find that no existing algorithm directly fits our needs as we need an algorithm that can scale-up to the size of the domain of

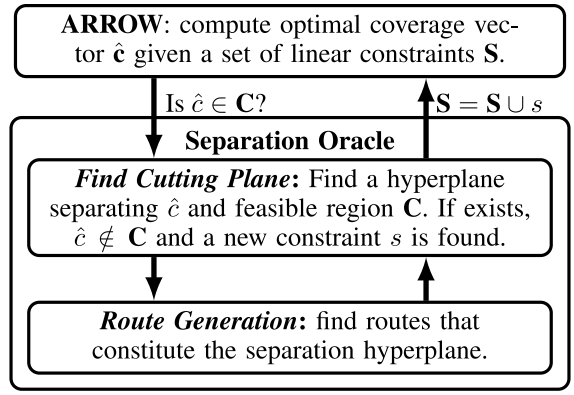


Figure 7: New integrated algorithm

interest, where: (i) we must generate patrol routes over the street map over the entire conservation area region, while (ii) simultaneously addressing payoff uncertainty and (iii) bounded rationality of the adversary. While the ARROW (Nguyen et al. 2015) algorithm allows us to address (ii) and (iii) together, it cannot handle scale-up over the street map. Indeed, while the (virtual) street map is of tremendous value in scaling up as discussed earlier, scaling up given all possible routes ($\approx 10^{12}$ routes) on the street map is still a massive research challenge. We therefore integrate ARROW with another algorithm BLADE (Yang et al. 2013) for addressing the scalability issue, resulting in a novel algorithm that can handle all the three aforementioned challenges. The new algorithm is outlined in Figure 7. In the following, we explain how ARROW and BLADE are adapted and integrated.

ARROW attempts to compute a strategy that is robust to payoff uncertainty given that poachers' responses follow SUQR. The concept of minimizing maximum regret is a well-known concept in AI for decision making under uncertainty (Wang and Boutilier 2003). ARROW uses the solution concept of *behavioral minimax regret* to provide the strategy that minimizes regret or utility loss for the patrollers in the presence of payoff uncertainty and bounded rational attackers. In small-scale domains, ARROW could be provided all the routes (the defender pure strategies), on the basis of which it would calculate the PAWS solution – a distribution over the routes. Unfortunately, in large scale domains like ours, enumerating all the routes is infeasible. We must therefore turn to an approach of incremental solution generation, which is where it interfaces with the BLADE framework.

More specifically, for scalability reasons, ARROW first generates the robust strategy for the patrollers in the form of coverage probabilities over the grid cells without consideration of any routes. Then a separation oracle in BLADE is called to check if the coverage vector is implementable. If it is implementable, the oracle returns a probability distribution over patrol routes that implements the coverage vector, which is the desired PAWS solution. If it is not implementable – see Figure 4(c) for an example of coverage vector that is not implementable – the oracle returns a constraint (cutting plane) that informs ARROW why it is not. For the example in Figure 4(b)-4(c), if ARROW generates a vector as shown in Figure 4(c), the constraint returned could be $c_1 \geq \sum_{i=2}^{4} c_i$ since all implementable coverage vector should satisfy this constraint. This constraint helps ARROW refine its solution. The process repeats until the coverage vector generated by ARROW is implementable.

As described in BLADE (Yang et al. 2013), to avoid enumerating all the routes to check whether the coverage vector is implementable, the separation oracle iteratively generate routes until it has just enough routes (usually after a small number of iterations) to match the coverage vector probabilities or get the constraint (cutting plane). At each iteration of this route generation (shown in the bottom-most box in Figure 7), the new route is optimized to cover targets of high value. However, we cannot directly use any existing algorithm to find the optimal route at each iteration due to the presence of our street map. But we note similarities to the well-studied orienteering problem (Vansteenwegen, Souffriau, and Oudheusden 2011) and exploit the insight of the S-algorithm for orienteering (Tsiligiridis 1984).

In particular, in this bottom-most box of in Figure 7, to ensure each route returned is of high quality, we run a local search over a large number of routes and return the one with the highest total value. In every iteration, we start from the base KAP, and choose which KAP to visit next through a weighted random selection. The next KAP to be visited can be any KAP on the map and we assume the patroller will take the shortest path from the current KAP to the next KAP. The weight of each candidate KAP is proportional to the ratio of the additional target value that can be accrued and distance from current KAP. We set the lower bound of weight to be $\epsilon > 0$ to make sure every feasible route can be chosen with positive probability. The process continues until the patroller has to go back to the base to meet the patrol distance limit constraint. Given a large number of such routes, our algorithm returns a route close to the optimal solution.

Integrating all these algorithms, PAWS calculates the patrol strategy consisting of a set of patrol routes and the corresponding probability for taking them.

Deployment and Evaluation

PAWS patrols are now regularly deployed at a conservation area in Malaysia. This section provides details about the deployment and both subjective and objective evaluations of PAWS patrols.

PAWS patrol aims to conduct daily patrols from base camps. Before the patrol starts, PAWS generates the patrol strategy starting from the base camp selected by patrol team leader. The patrol distance limit considered by PAWS is 10km per day (equivalent flat terrain). As shown in Table 1, this leads to about 9000 raster pieces to be considered. Thus it is impossible to consider each raster piece as a separate target or consider all possible routes over the raster pieces. With the two-level of discretization and the street map, the problem scale is reduced, with $8.57 (= 194.33/22.67)$ KAPs and 80 route segments in each grid cell on average, making the problem manageable. The strategy generated by PAWS is a set of suggested routes associated with probabilities and the average number of suggested routes associated with probability > 0.001 is 12.

Average # of Reachable Raster Pieces	9066.67
Average # of Reachable Grid Cells (Targets)	22.67
Average # of Reachable KAPs	194.33

Table 1: Problem Scale for PAWS Patrols.

Average Trip Length	4.67 Days
Average Number of Patrollers	5
Average Patrol Time Per Day	4.48 hours
Average Patrol Distance Per Day	9.29 km

Table 2: Basic Information of PAWS Patrols.

Each PAWS patrol lasts for 4-5 days, and is executed by a team of 3-7 patrollers. The patrol planner will make plans based on the strategy generated by PAWS. After reaching the base camp, patrollers execute daily patrols, guided by PAWS's patrol routes. Table 2 provides a summary of basic statistics about the patrols. During the patrol, the patrollers are equipped with a printed map, a handheld GPS, and data recording booklet. They detect animal and human activity signs and record them with detailed comments and photos. After the patrol, the data manager will put all the information into a database, including patrol tracks recorded by the handheld GPS, and the observations recorded in the log book.

Figure 8 shows various types of signs found during the patrols. Table 3 summarizes all the observations. These observations show that there is a serious ongoing threat from the poachers. Column 2 shows results for all PAWS patrols. Column 3 shows results for explorative PAWS patrols, the (partial) patrol routes which go across areas where the patrollers have never been before. To better understand the numbers, we show in Column 4 the statistics about early-stage non-PAWS patrols in this conservation area, which were deployed for tiger survey. Although it is not a fair comparison as the objectives of the non-PAWS patrols and PAWS patrols are different, comparing Column 2 and 3 with Column 4 indicates that PAWS patrols are effective in finding human activity signs and animal signs. Finding the human activity signs is important to identify hotspots of poaching activity, and patrollers' presence will deter the poachers. Animals signs are not directly evaluating PAWS patrols but they indicate that PAWS patrols prioritize areas with higher animal density. Finding these signs is aligned with the goal of PAWS – combat poaching to save animals – and thus is a proof for the effectiveness of PAWS. Comparing Column 3 with Column 2, we find the average number of observations made along the *explorative routes* is comparable to and even higher than that of all PAWS patrol routes. The observations on explorative routes are important as they lead to a better understanding of the unexplored area. These results show that PAWS can guide the patrollers towards hotspots of poaching activity and provide valuable suggestions to the patrol planners.

Along the way of PAWS deployment, we have received feedback from patrol planners and patrollers. The patrol planners mentioned that the top routes in PAWS solution (routes with highest probability) come close to an actual

(a) Tiger sign (Nov. 2014)

(b) Human sign (lighter; Jul. 2015)

(c) Human sign (old poacher camp; Aug. 2015)

(d) Human sign (tree marking; Aug 2015)

Figure 8: Various signs recorded during PAWS patrols.

Patrol Type	All PAWS Patrol	Explorative PAWS Patrol	Previous Patrol for Tiger Survey
Total Distance (km)	130.11	20.1	624.75
Average # of Human Activity Signs per km	0.86	1.09	0.57
Average # of Animal Signs per km	0.41	0.44	0.18

Table 3: Summary of observations.

planner's routes, which shows PAWS can suggest feasible routes and potentially reduce the burden of planning effort. As we deploy PAWS in the future at other sites, the cumulative human planners' effort saved by using PAWS will be a considerable amount. In addition, patrollers commented that PAWS is able to guide them towards poaching hotspots. The fact that they found multiple human signs along the explorative PAWS patrol routes makes them believe that PAWS is good at finding good ridgelines that are taken by animals and humans. Patrollers and patrol planners also agree that PAWS generates detailed suggested routes which can guide the actual patrol. Patrollers commented that the suggested routes are mostly along the ridgeline, which are easier to follow, compared with the routes from the first trial by *PAWS-Initial*. Figure 9 shows one suggested route (orange line) and the actual patrol track (black line) during PAWS patrol in Aug. 2015 (shown on 1km by 1km grid). Due to the precision of the contour lines we get, we provide a 50m buffer zone (light orange polygon) around the suggested route (orange line). The patrollers started from the basecamp (green triangle) and headed to the southeast. The patrollers mostly followed PAWS's suggested route, indicating that the route generated by PAWS is easy to follow (contrast with *PAWS-Initial* as shown in Figure 3(a)). Finally, the power of randomization in PAWS solution can be expected in the long-term.

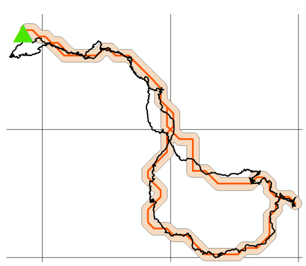


Figure 9: One daily PAWS Patrol route in Aug. 2015.

Lessons Learned

During the development and deployment process, we faced several challenges and here we outline some lessons learned.

First, first-hand immersion in the security environment of concern is critical to understanding the context and accelerating the development process. The authors (from USC and NTU) intentionally went for patrols in the forest with the local patrolling team to familiarize themselves with the area. The first-hand experience confirmed the importance of ridgelines, as several human and animal signs are found along the way, and also confirmed that extreme changes in elevation require considerable extra effort of the patrollers. This gave us the insight for building the street map.

Second, visualizing the solution is important for communication and technology adaptation. When we communicate with domain experts and human planners, we need to effectively convey the game-theoretic strategy generated by PAWS, which is a probability distribution over routes. We first visualize the routes with probability > 0.01 using ArcGIS so that they can be shown on the topographic map and the animal distribution map. Then for each route, we provide detailed information that can assist the human planners' decision making. We not only provide basic statistics such as probability to be taken and total distance, but also estimate the difficulty level for patrol, predict the probability of finding animals and human signs, and provide an elevation chart that shows how the elevation changes along the route. Such information can help planners' understanding the strategy.

Third, minimizing the need for extra equipment/effort would further ease PAWS future deployment, i.e., patrollers would prefer having a single handheld device for collecting patrol data and displaying suggested patrol routes. If PAWS routes could be embedded in the software that is already in use for collecting data in many conservation areas, e.g., SMART, it would reduce the effort required of planners. This is one direction for future development.

Summary

PAWS is a first deployed "green security game" application to optimize human patrol resources to combat poaching. We provided key research advances to enable this deployment; this has provided practical benefit to patrol planners and patrollers. The deployment of PAWS patrols will continue at the site in Malaysia. Panthera has seen the utility of PAWS and we are taking steps to expand PAWS to its other sites. This future expansion and maintenance of PAWS will be taken over by ARMORWAY (ARMORWAY 2015), a “security games” company (starting in Spring 2016); ARMORWAY has significant experience in supporting security-games-based software deployments.

Acknowledgement

This research was supported by MURI Grant W911NF-11-1-0332. And thanks to our partners in the field who made these tests possible.

References

ARMORWAY. 2015. http://armorway.com/.

Botea, A.; Mller, M.; and Schaeffer, J. 2004. Near optimal hierarchical path-finding. *Journal of Game Development* 1:7–28.

Chapron, G.; Miquelle, D. G.; Lambert, A.; Goodrich, J. M.; Legendre, S.; and Clobert, J. 2008. The impact on tigers of poaching versus prey depletion. *Journal of Applied Ecology* 45:16671674.

Fang, F.; Jiang, A. X.; and Tambe, M. 2013. Optimal patrol strategy for protecting moving targets with multiple mobile resources. In *AAMAS*.

Fang, F.; Stone, P.; and Tambe, M. 2015. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Hamisi, M. 2008. *Identification and mapping risk areas for zebra poaching: A case of Tarangire National Park, Tanzania*. Ph.D. Dissertation, Thesis, ITC.

IUCN. 2015. IUCN red list of threatened species. version 2015.2. http://www.iucnredlist.org.

Kar, D.; Fang, F.; Fave, F. D.; Sintov, N.; and Tambe, M. 2015. A Game of Thrones: When human behavior models compete in repeated Stackelberg security games. In *AAMAS 2015*.

Korzhyk, D.; Conitzer, V.; and Parr, R. 2010. Complexity of computing optimal Stackelberg strategies in security resource allocation games. In *AAAI*, 805–810.

Langmuir, E. 1995. *Mountaincraft and Leadership: A Handbook for Mountaineers and Hillwalking Leaders in the British Isles*. Mountain Leader Training Board.

Lemieux, A. M., ed. 2014. *Situational Prevention of Poaching*. Crime Science Series. Routledge.

Nguyen, T. H.; Yang, R.; Azaria, A.; Kraus, S.; and Tambe, M. 2013. Analyzing the effectiveness of adversary modeling in security games. In *AAAI*.

Nguyen, T. H.; Yadav, A.; An, B.; Tambe, M.; and Boutilier, C. 2014. Regret-based optimization and preference elicitation for Stackelberg security games with uncertainty. In *AAAI*.

Nguyen, T. H.; Fave, F. M. D.; Kar, D.; Lakshminarayanan, A. S.; Yadav, A.; Tambe, M.; Agmon, N.; Plumptre, A. J.; Driciru, M.; Wanyama, F.; and Rwetsiba, A. 2015. Making the most of our regrets: Regret-based solutions to handle payoff uncertainty and elicitation in green security games. In *Conference on Decision and Game Theory for Security*.

Phillips, S. J.; Anderson, R. P.; and Schapire, R. E. 2006. Maximum entropy modeling of species geographic distributions. *Ecological Modelling* 190(3-4):231–259.

Pita, J.; Jain, M.; Western, C.; Portway, C.; Tambe, M.; Ordonez, F.; Kraus, S.; and Paruchuri, P. 2008. Deployed ARMOR protection: The application of a game theroetic model for security at the Los Angeles International Airport. In *AAMAS*.

Plummer, M. 2003. JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling.

Sanderson, E.; Forrest, J.; Loucks, C.; Ginsberg, J.; Dinerstein, E.; Seidensticker, J.; Leimgruber, P.; Songer, M.; Heydlauff, A.; OBrien, T.; Bryja, G.; Klenzendorf, S.; and Wikramanayake, E. 2006. Setting priorities for the conservation and recovery of wild tigers: 2005-2015. the technical assessment. Technical report, WCS, WWF, Smithsonian, and NFWF-STF, New York Washington, D.C.

SMART. 2013. The spatial monitoring and reporting tool (SMART). http://www.smartconservationsoftware.org/.

Stokes, E. J. 2010. Improving effectiveness of protection efforts in tiger source sites: developing a framework for law enforcement monitoring using mist. *Integrative Zoology* 5(4):363–377.

Tambe, M. 2011. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press.

Tarboton, D. G.; Bras, R. L.; and Rodriguez-Iturbe, I. 2007. On the extraction of channel networks from digital elevation data. *Hydrologic Processes* 5(1):81–100.

Thompson, S. 2011. *Unjustifiable Risk?: The Story of British Climbing*. Cicerone Press.

Tobler, W. 1993. Three presentations on geographical analysis and modeling. non-isotropic geographic modeling: speculations on the geometry of geography, and global spatial analysis (93-1). Technical report, UC Santa Barbara.

Tsiligiridis, T. 1984. Heuristic methods applied to orienteering. *The Journal of the Operational Research Society* 35(9):pp. 797–809.

Vansteenwegen, P.; Souffriau, W.; and Oudheusden, D. V. 2011. The orienteering problem: A survey. *European Journal of Operational Research* 209(1):1–10.

Wang, T., and Boutilier, C. 2003. Incremental utility elicitation with the minimax regret decision criterion. In *IJCAI*.

Wato, Y. A.; Wahungu, G. M.; and Okello, M. M. 2006. Correlates of wildlife snaring patterns in tsavo west national park, Kenya. *Biological Conservation* 132(4):500–509.

Yang, R.; Jiang, A. X.; Tambe, M.; and Ordonez, F. 2013. Scaling-up security games with boundedly rational adversaries: A cutting-plane approach. In *IJCAI*.

Yang, R.; Ford, B.; Tambe, M.; and Lemieux, A. 2014. Adaptive resource allocation for wildlife protection against illegal poachers. In *AAMAS*.