Behavioral Minimax Regret for Security Games and Its Application for UAV Planning

Thanh H. Nguyen[1], Amulya Yadav[1], Francesco Delle Fave[1], Milind Tambe[1], Noa Agmon[2], Manish Jain[1], and Richard Van Deventer[3]

[1] University of Southern California, Los Angeles, USA
[thanhhng,amulyaya,dellefav,tambe,manishja]@usc.edu
[2] Bar-Ilan University, Israel
agmon@cs.biu.ac.il
[3] ShadowView Foundation, The Netherlands
richard@shadowview.com

Abstract. Research on Stackelberg Security Games (SSG) has recently shifted to green security domains, e.g., protecting wildlife from illegal poaching. Previous research on this topic has advocated the use of behavioral (bounded rationality) models of adversaries in SSG. As its first contribution, this paper, for the first time, provides validation of these behavioral models based on real-world data from a wildlife park. The paper's next contribution is the first algorithm to handle payoff uncertainty – an important concern in green security domains – in the presence of such adversary behavioral models. Finally, given the availability of mobile sensors such as Unmanned Aerial Vehicles in green security domains, as our third contribution, we introduce new payoff elicitation strategies to strategically reduce uncertainty over multiple targets at a time.

1 Introduction

Given the successful deployments of Stackelberg Security Games (SSG) for infrastructure protection [20, 1, 10], research on SSG has shifted to green security domains. This research focuses on optimally allocating limited security resources in a vast geographical area against environmental crime, e.g., improving the effectiveness of protection of wildlife or fisheries [23, 4].

These green security domains exhibit at least three unique challenges/opportunities. First, adversaries attack without spending as much time/effort on each attack as in terrorist attacks on infrastructure; it thus becomes more important to model the adversaries' bounded rationality in these domains. Second, there is a significant need to handle uncertainty in both players' payoffs since key domain features, e.g., animal density, that contribute to the payoffs are difficult to precisely estimate. Finally, defenders in these domains have access to mobile sensors such as Unmanned Aerial Vehicles (UAV) to elicit information over multiple targets at once to reduce payoff uncertainty. Unfortunately, previous work has failed to address these challenges and leverage these opportunities. First, although there are a number of behavioral models proposed to handle adversaries' bounded rationality in SSG, none of these have yet been evaluated on real-world data. Second, previous work applied several robust optimization methods

to handle payoff uncertainty; but they have failed to address such uncertainty in the context of aforementioned behavioral models [8]. Third, previous work does not leverage resources that elicit over multiple targets (e.g., UAVs); they only provide simple heuristics for payoff elicitation (PE) at one target at a time [14].

In this paper, as our first contribution, we provide the first results on the usefulness of behavioral models in SSG using real-world data from a wildlife protection domain. Our second contribution is CONQUER (***CO****nstraint ge****N****eration to compute* ***QU****antal Response based minimax r****E****g****R****et*), the first security game algorithm that can solve the *behavioral minimax regret problem*. MiniMax Regret (MMR), to minimize maximum regret from a solution [6], is a robust solution approach to handle payoff uncertainty; a key advantage of using MMR is that it is less conservative than the standard maximin approach [14]. CONQUER is the first algorithm to compute MMR in the presence of a behavioral (bounded rationality) model, rather than assuming a perfectly rational adversary; it is also the first to handle payoff uncertainty in both the adversary and the defenders' payoffs in SSG. However, handling of adversary bounded rationality and uncertainty in both players' payoffs creates the challenge of solving a non-convex optimization problem; CONQUER provides an efficient solution to such problems.

Another significant advantage of MMR is that it is a very effective driver of preference elicitation [3]. We exploit this advantage by presenting two new PE heuristics which select *multiple* targets for reducing payoff uncertainty at a time, leveraging the multi-target-elicitation capability of resources available in green security domains. Lastly, we conduct extensive experiments, including evaluations of CONQUER on a real-world wildlife park.

2 Background & Related Work

Stackelberg Security Games: In SSG, the defender attempts to protect a set of T targets from an attack by an adversary by optimally allocating a set of R resources ($R < T$) [20]. The key assumption here is that the defender commits to a (*mixed*) strategy first and the adversary can observe that strategy and then attack a target. Denote by $\mathbf{x} = \{x_t\}$ the defender's strategy where x_t is the coverage probability at target t, the set of feasible strategies is $\mathbf{X} = \{\mathbf{x} : 0 \leq x_t \leq 1, \sum_t x_t \leq R\}$ [9]. If the adversary attacks t when the defender is not protecting it, he receives a reward R_t^a, otherwise, he gets a penalty P_t^a. Conversely, the defender receives a penalty P_t^d in the former case and a reward R_t^d in the latter case. Let $(\mathbf{R^a}, \mathbf{P^a})$ and $(\mathbf{R^d}, \mathbf{P^d})$ be the payoff vectors. The players' expected utilities at t is computed as:

$$U_t^a(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a}) = x_t P_t^a + (1 - x_t) R_t^a \tag{1}$$

$$U_t^d(\mathbf{x}, \mathbf{R^d}, \mathbf{P^d}) = x_t R_t^d + (1 - x_t) P_t^d \tag{2}$$

Boundedly rational attacker: In SSG, attacker bounded rationality is often modeled via behavior models such as Quantal Response (QR) [11, 12]. The recent SUQR model (Subjective Utility Quantal Response) builds on QR by integrating the subjective utility function (Equation 3) into QR, and it was shown to provide a better prediction accuracy than QR [15]:

$$\hat{U}_t^a(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a}) = w_1 x_t + w_2 R_t^a + w_3 P_t^a \tag{3}$$

where (w_1, w_2, w_3) are parameters indicating the importance of the three target features for the adversary. SUQR predicts the adversary's probability of attacking t, $q_t(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a})$, as:

$$q_t(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a}) = \frac{e^{\hat{U}^a_t(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a})}}{\sum_{t'} e^{\hat{U}^a_{t'}(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a})}} \quad (4)$$

Recall that QR does not use subjective utility in (4), but uses $\lambda U^a_t(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a})$ where the parameter λ governs the adversary's rationality. One key advantage of these behavioral models is that they can be used to predict attack frequency for multiple attacks by the adversary, wherein the attacking probability is a normalization of attacking frequency.

Payoff uncertainty: One key approach to modeling payoff uncertainty is to express the adversary's payoffs as lying within specific intervals [8]: for each t, $R^a_t \in [R^a_{min}(t), R^a_{max}(t)]$ and $P^a_t \in [P^a_{min}(t), P^a_{max}(t)]$. Let $\mathbf{I}$ denote the set of payoff intervals at all targets. An MMR-based solution was introduced in previous work to address payoff uncertainty in SSG while assuming a *perfectly rational* adversary [14]. Furthermore, they only address uncertainty in the adversary's payoff.

Green security domains: These domains include challenges such as protecting wildlife from poaching or protecting fisheries from illegal fishing. We focus on wildlife protection — many species such as rhinos and tigers are in danger of extinction from poaching [13, 17]. To protect wildlife, game-theoretic approaches have been advocated to generate rangers' patrols [23] wherein the forest area is divided into a grid where each cell is a target. These ranger patrols are designed to counter poachers (whose behaviors are modeled using SUQR) that attempt to capture animals by setting snares. A similar system has also been developed for protecting fisheries [4].

Unfortunately, this previous work in green security domains has three weaknesses [23]. First, models like SUQR/QR have not been compared on available real world data, e.g., poaching signs observed by rangers [19]. Second, this work addresses adversary bounded rationality but fails to simultaneously address payoff uncertainty — an important issue because of the difficulty of precisely estimating payoffs in green security domains (e.g., animal densities are hard to estimate within a national park). Finally, previous work does not leverage available mobile sensors such as UAVs to reduce uncertainty in the payoffs.

3 Behavioral Modeling Validation

Our first contribution is to use World Wildlife Fund's (WWF) real-world patrol/poaching data from a wildlife reserve in Indonesia (name of the park is withheld intentionally) to analyze the effectiveness of SUQR/QR in predicting attacks by real-world poachers. Our dataset consisted of information about patrols conducted over a period of 5 months. The wildlife park area is divided into 244 2x2 km grid cells (total area $\sim$1000 km^2). For each patrol, we had information about which grid cells the patrollers covered, along with various features of those cells that they observed (e.g., animal density). Accurate estimates of features like animal density, tree canopy, etc. in each grid cell were obtained by patrollers only after they patrolled these cells. We also obtained information about how many poaching signs (e.g., snares) were observed by the patrollers in each cell.

Dataset preprocessing. We process our patrol data for learning QR and SUQR parameters as follows. First, we find the average frequency of patrolling in each grid cell over the 5 month duration. Similarly, we aggregate key features such as number of poaching signs and animal density. Next, we convert our multiple valued label (number of poaching signs observed in that cell) into a binary label (whether that cell was attacked or not). As long as at least one poaching sign was observed in that cell, we considered that cell to be attacked (label 1), and otherwise, we consider the cell unattacked (label 0). Given this processed dataset, we wish to compare four different models: SUQR-3 (SUQR with 3 features — patrol frequency, animal density (as reward) and a constant penalty term), SUQR-7 (SUQR with 7 features — patrol frequency, animal density, area habitat, slope, canopy, understory, and litter), QR (with animal density as reward and constant penalties on all targets), and a perfectly rational model.

Learning results. We first learn the parameters of the three models: SUQR-3, SUQR-7, and QR for our labelled dataset having 244 data points. We did random subsampling validation to create 1000 random 90:10 training/test splits. For each split, we train our behavioral models to learn their parameters, which are used to get probabilities of attack on each grid cell in the test set. Thus, for each grid cell, we get the actual label (whether the target was attacked or not) along with our predicted probability of attack on the cell. Using these labels and the predicted probabilities, we plotted a Receiver Operating Characteristic (ROC) curve (in Figure 1) to analyze the performance of the various models. The result shows that the perfectly rational model, which deterministically classifies which target gets attacked (unlike SUQR/QR which give probabilities of attack on all targets), achieves a true positive rate and false positive rate of 0 which is extremely poor. Also, QR (Area Under the Curve (AUC)=0.35) performs worse than all other models including Random. Also, both SUQR-3 (AUC=0.87) and SUQR-7 (AUC=0.87) provide high prediction accuracies. Since SUQR-7's extra features do not provide any benefit, we use SUQR-3 as our model of choice in the rest of the paper (as it is generalizable to standard security games). This experiment is the first comparison of any behavioral models on real-world data in the context of SSG, showing SUQR's benefit.

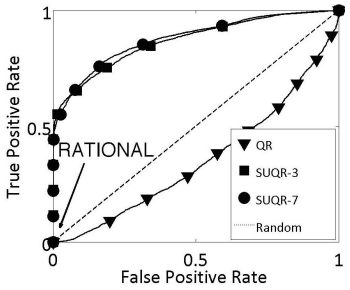


Fig. 1. ROC curve

4 Behavioral Minimax Regret ($\mathrm{MMR_b}$)

While we can learn a behavioral model from real-world data, we still face the challenge of significant payoff uncertainty. For example, since rangers patrol and collect wildlife data within a small portion of a national park, payoffs in other areas of the park may remain uncertain. Also, due to the dynamic changes (e.g., animal migration), players' payoffs may become uncertain in the next season. Hence, this paper introduces our new MMR-based robust algorithm, CONQUER, to handle payoff uncertainty in SSG, taking into account adversarial behavioral models. (Recall that MMR is a robust criteria

less conservative than maximin & an effective driver for preference elicitation.) Here, we primarily focus on zero-sum games as motivated by recent work in green security domains [7, 4], and earlier major SSG applications that use zero-sum games [18, 25]). In addition, we use SUQR as the adversary's behavioral model, given the results in Section 3. However, our methods generalize to non-zero-sum games with a general class of QR (see Online Appendix C).[4]

We now formulate $\mathrm{MMR_b}$ with uncertain payoffs for both players in zero-sum SSG with a boundedly rational attacker.

Definition 1. *Given* $(\mathbf{R^a}, \mathbf{P^a})$*, the defender's* ***behavioral regret*** *is the loss in her utility for playing a strategy* $\mathbf{x}$ *instead of the optimal strategy, which is represented as follows:*

$$\mathrm{R_b}(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a}) = \max_{\mathbf{x'} \in \mathbf{X}} F(\mathbf{x'}, \mathbf{R^a}, \mathbf{P^a}) - F(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a}) \tag{5}$$

$$\text{where } F(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a}) = \sum\nolimits_t q_t(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a}) U_t^d(\mathbf{x}, \mathbf{R^d}, \mathbf{P^d}) \tag{6}$$

Here, $F(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a})$ is the defender's utility (which is non-convex fractional in $\mathbf{x}$) for playing $\mathbf{x}$ where the payoff of the adversary, whose response follows SUQR, is $(\mathbf{R^a}, \mathbf{P^a})$ and $\mathbf{R^d} = -\mathbf{P^a}$ and $\mathbf{P^d} = -\mathbf{R^a}$. In addition, the attacking probability, $q_t(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a})$, is given by Equation 4.

Definition 2. *Given a set of payoff intervals* $\mathbf{I}$*, the* ***behavioral max regret*** *that the defender receives for playing a strategy* $\mathbf{x}$ *is the maximum behavioral regret over all payoff instances:*

$$\mathrm{MR_b}(\mathbf{x}, \mathbf{I}) = \max_{(\mathbf{R^a}, \mathbf{P^a}) \in \mathbf{I}} \mathrm{R_b}(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a}) \tag{7}$$

Definition 3. *Given a set of payoff intervals* $\mathbf{I}$*, the* ***behavioral minimax regret*** *problem attempts to find the optimal strategy for the defender that minimizes the* $\mathrm{MR_b}$ *she receives:*

$$\mathrm{MMR_b}(\mathbf{I}) = \min_{\mathbf{x} \in \mathbf{X}} \mathrm{MR_b}(\mathbf{x}, \mathbf{I}) \tag{8}$$

As our experiments show, if the defender uses MMR for a perfectly rational attacker instead of $\mathrm{MMR_b}$, she may suffer a significant utility loss.

5 CONQUER Algorithm

Algorithm 1 presents the outline of CONQUER to solve the $\mathrm{MMR_b}$ problem in Equation 8. Essentially, CONQUER's two novelties compared to previous work [14] — addressing uncertainty in both players' payoffs and a boundedly rational attacker — lead to two new computational challenges: 1) uncertainty in defender payoffs makes the defender's expected utility at every target t non-convex in $\mathbf{x}$ and $(\mathbf{R^d}, \mathbf{P^d})$ (Equation 2); and 2) SUQR represents the attacker's bounded rationality in the form of a logit function which is non-convex. These two non-convex functions are combined when calculating the defender's utility (Equation 6) — which is then used in computing $\mathrm{MMR_b}$ (Equation 8), making it computationally expensive. Overall, $\mathrm{MMR_b}$ can be reformulated as follows:

$$\min_{\mathbf{x} \in \mathbf{X}, r \in \mathbb{R}} r \tag{9}$$

$$\text{s.t. } r \geq F(\mathbf{x'}, \mathbf{R^a}, \mathbf{P^a}) - F(\mathbf{x}, \mathbf{R^a}, \mathbf{P^a}), \forall (\mathbf{R^a}, \mathbf{P^a}) \in \mathbf{I}, \mathbf{x'} \in \mathbf{X}$$

[4] https://www.dropbox.com/s/vrii1mt32is34d1/Appendix.pdf?dl=0

Algorithm 1: CONQUER Outline

1 Initialize $S = \phi, ub = \infty, lb = 0$;
2 Randomly generate $(\mathbf{x}', \mathbf{R}^{\mathbf{a}}, \mathbf{P}^{\mathbf{a}})$, $S = S \cup \{\mathbf{x}', (\mathbf{R}^{\mathbf{a}}, \mathbf{P}^{\mathbf{a}})\}$;
3 **while** $ub > lb$ **do**
4 Call PALMS to compute relaxed $\mathrm{MMR_b}$ w.r.t S. Let $\mathbf{x}^*$ be its optimal solution with objective value lb;
5 Call REALMS to compute $\mathrm{MR_b}(\mathbf{x}^*, \mathbf{I})$. Let the optimal solution be $(\mathbf{x}'^{,*}, \mathbf{R}^{\mathbf{a},*}, \mathbf{P}^{\mathbf{a},*})$ with objective value ub;
6 $S = S \cup \{\mathbf{x}'^{,*}, \mathbf{R}^{\mathbf{a},*}, \mathbf{P}^{\mathbf{a},*}\}$;
7 return $(lb, \mathbf{x}^*)$;

Unfortunately, since $\mathbf{X}$ and $\mathbf{I}$ are continuous, the set of constraints is infinite. One practical approach to optimization with large constraint sets is *constraint sampling* [5], coupled with *constraint generation* [2]. Following this approach, CONQUER samples a subset of constraints in Problem (9) and gradually expands this set by adding violated constraints to the relaxed problem until convergence to the optimal $\mathrm{MMR_b}$ solution. Specifically, CONQUER begins by sampling pairs $(\mathbf{R}^{\mathbf{a}}, \mathbf{P}^{\mathbf{a}})$ of the adversary payoffs uniformly from $\mathbf{I}$. The corresponding optimal strategies for the defender given these payoff samples, denoted $\mathbf{x}'$, are then computed using the algorithm PASAQ [24] to obtain a finite set S of sampled constraints (Line 2). These sampled constraints are then used to solve the corresponding *relaxed* $\mathrm{MMR_b}$ program (line 4) using the PALMS algorithm (described in Section 5.1) — we call this problem *relaxed* $\mathrm{MMR_b}$ as it only has samples of constraints in (9). We thus obtain the optimal solution $(lb, \mathbf{x}^*)$ which provides a lower bound (lb) on the true $\mathrm{MMR_b}$. Then constraint generation is applied to determine violated constraints (if any). This uses the REALMS algorithm (described in Section 5.2) which computes $\mathrm{MR_b}(\mathbf{x}^*, \mathbf{I})$ — the optimal regret of $\mathbf{x}^*$ which is an upper bound (ub) on the true $\mathrm{MMR_b}$. If $ub > lb$, the optimal solution of REALMS, $\{\mathbf{x}'^{,*}, \mathbf{R}^{\mathbf{a},*}, \mathbf{P}^{\mathbf{a},*}\}$, provides the maximally violated constraint (line 5), which is added to S. Otherwise, $\mathbf{x}^*$ is the minimax optimal strategy and $lb = ub = \mathrm{MMR_b}(\mathbf{I})$.

5.1 PALMS: Compute Relaxed $\mathrm{MMR_b}$

The first step of CONQUER is to solve the relaxed $\mathrm{MMR_b}$ problem using PALMS (***P**iece-wise linear & bin**A**ry search for re**L**axed **M**inimax against **S**UQR adversary*). This relaxed $\mathrm{MMR_b}$ problem is non-convex and fractional. Thus, PALMS presents two key ideas for efficiency: 1) binary search (which iteratively searches the defender's utility space to find the optimal solution) to remove the fractional terms; and 2) it then applies piecewise-linear approximation to linearize the non-convex terms. Overall, the relaxed $\mathrm{MMR_b}$ problem can be represented as follows:

$$\min_{\mathbf{x} \in \mathbf{X}, r \in \mathbb{R}} r \tag{10}$$
$$\text{s.t. } r \geq F(\mathbf{x}'^{,\mathbf{k}}, \mathbf{R}^{\mathbf{a},\mathbf{k}}, \mathbf{P}^{\mathbf{a},\mathbf{k}}) - F(\mathbf{x}, \mathbf{R}^{\mathbf{a},\mathbf{k}}, \mathbf{P}^{\mathbf{a},\mathbf{k}}), \forall k = \overline{1, K}$$

where $(\mathbf{x}'^{,\mathbf{k}}, \mathbf{R}^{\mathbf{a},\mathbf{k}}, \mathbf{P}^{\mathbf{a},\mathbf{k}})$ is the k^{th} sample in S where $k = \overline{1, K}$ and K is the total number of samples in S and r is the defender's max regret for playing $\mathbf{x}$ against sample set S. Finally, $F(\mathbf{x}'^{,\mathbf{k}}, \mathbf{R}^{\mathbf{a},\mathbf{k}}, \mathbf{P}^{\mathbf{a},\mathbf{k}})$ is the defender's optimal utility for every sample

of attacker payoffs $(\mathbf{R}^{\mathbf{a},\mathbf{k}}, \mathbf{P}^{\mathbf{a},\mathbf{k}})$ where $\mathbf{x}'^{,\mathbf{k}}$ is the corresponding defender's optimal strategy (which may be obtained via PASAQ [24]). The term $F(\mathbf{x}, \mathbf{R}^{\mathbf{a},\mathbf{k}}, \mathbf{P}^{\mathbf{a},\mathbf{k}})$, which is included in relaxed $\mathrm{MMR_b}$'s constraints, is non-convex and fractional in $\mathbf{x}$ (Equation 6), making (10) non-convex and fractional. We can use any non-convex solver with multiple starting points, e.g., Fmincon of MATLAB to solve it; however, these solvers are time consuming. We now detail the two key ideas of PALMS.

Binary search. In each binary search step, given a value of r, PALMS tries to solve the following decision problem (**P1**):

$$\textbf{(P1)}: \exists \mathbf{x} \text{ s.t. } r \geq F(\mathbf{x}'^{,\mathbf{k}}, \mathbf{R}^{\mathbf{a},\mathbf{k}}, \mathbf{P}^{\mathbf{a},\mathbf{k}}) - F(\mathbf{x}, \mathbf{R}^{\mathbf{a},\mathbf{k}}, \mathbf{P}^{\mathbf{a},\mathbf{k}}), \forall k = \overline{1, K}?$$

Based on Theorem 1, we can convert (**P1**) into the following *non-fractional* optimization problem (**P2**) of which the optimal solution is used to determine the feasibility of (**P1**):

$$\textbf{(P2):} \min_{\mathbf{x} \in \mathbf{X}, v \in \mathbb{R}} v$$
$$\text{s.t. } v \geq \sum\nolimits_t \left[r_k - U_t^{d,k}(\mathbf{x}) \right] e^{w_1 x_t + w_2 R_t^{a,k} + w_3 P_t^{a,k}}, \forall k = \overline{1, K}$$

where $U_t^{d,k}(\mathbf{x}) = -\left[x_t P_t^{a,k} + (1 - x_t) R_t^{a,k} \right]$ is the defender's utility and $r_k = F(\mathbf{x}'^{,\mathbf{k}}, \mathbf{R}^{\mathbf{a},\mathbf{k}}, \mathbf{P}^{\mathbf{a},\mathbf{k}}) - r$ given the k^{th} sample.

Theorem 1. *Suppose that $(v^*, \mathbf{x}^*)$ is the optimal solution of (**P2**). If $v^* \leq 0$, then $\mathbf{x}^*$ is a feasible solution of the decision problem (**P1**). Otherwise, (**P1**) is infeasible.*[5]

Piecewise linear approximation. Although (**P2**) is non-fractional, its constraints are non-convex. We use a piecewise linear approximation for the RHS of the constraints in (**P2**) which is in the form of $\sum_t f_t^k(x_t)$ where the term $f_t^k(x_t)$ is a non-convex function of x_t. The feasible region of the defender's coverage x_t for all t, $[0, 1]$, is then divided into M equal segments $\left\{ \left[0, \frac{1}{M}\right], \left[\frac{1}{M}, \frac{2}{M}\right], \ldots, \left[\frac{M-1}{M}, 1\right] \right\}$ where M is given. The values of $f_t^k(x_t)$ are then approximated by using the segments connecting pairs of consecutive points $\left(\frac{i-1}{M}, f_t^k\left(\frac{i-1}{M}\right)\right)$ and $\left(\frac{i}{M}, f_t^k\left(\frac{i}{M}\right)\right)$ for $i = \overline{1, M}$ as follows:

$$f_t^k(x_t) \approx f_t^k(0) + \sum\nolimits_{i=1}^{M} \alpha_{t,i}^k x_{t,i} \tag{11}$$

where $\alpha_{t,i}^k$ is the slope of the i^{th} segment. Also, $x_{t,i}$ refers to the portion of the defender's coverage at target t belonging to the i^{th} segment, i.e., $x_t = \sum_i x_{t,i}$. For example, suppose that $M = 5$ and $x_t = 0.3$, as $\frac{1}{5} < x_t < \frac{2}{5}$, we obtain $x_{t,1} = \frac{1}{5}$, $x_{t,2} = 0.1$, and $x_{t,3} = x_{t,4} = x_{t,5} = 0$. By using the approximations of $f_t^k(x_t)$ for all k and t, we

[5] All proofs appear in the Online Appendix.

Algorithm 2: Elicitation process

```
Input: budget: B, regret barrier: δ, uncertainty intervals: I;
Initialize regret r = +∞, cost c = 0 ;
while c < B and r > δ do
    (r, x*, (x'^,*, R^{a,*}, P^{a,*})) = CONQUER(I);
    P = selectPath(x*, (x'^,*, R^{a,*}, P^{a,*}));
    I = executeUAV(P);
    c = updateCost(P);
return (r, x*);
```

can reformulate (**P2**) as the MILP **(P2')** which can be solved by CPLEX:

$$\textbf{(P2')}: \min_{x_{t,i}, z_{t,i}, v} v \tag{12}$$

$$\text{s.t. } v \geq \sum\nolimits_t f_t^k(0) + \sum\nolimits_t \sum\nolimits_i \alpha_{t,i}^k x_{t,i}, \forall k = \overline{1, K} \tag{13}$$

$$\sum\nolimits_{t,i} x_{t,i} \leq R, 0 \leq x_{t,i} \leq \frac{1}{M}, \forall t, i = \overline{1, M} \tag{14}$$

$$z_{t,i} \frac{1}{M} \leq x_{t,i}, \forall t, i = \overline{1, M-1} \tag{15}$$

$$x_{t,i+1} \leq z_{t,i}, \forall t, i = \overline{1, M-1} \tag{16}$$

$$z_{t,i} \in \{0, 1\}, \forall t, i = \overline{1, M-1} \tag{17}$$

where $z_{t,i}$ is an auxiliary integer variable which ensures that the portions of x_t satisfies $x_{t,i} = \frac{1}{M}$ if $x_t \geq \frac{i}{M}$ ($z_{t,i} = 1$) or $x_{t,i+1} = 0$ if $x_t < \frac{i}{M}$ ($z_{t,i} = 0$) (constraints (14 – 17)). Constraints (13) are equivalent to constraints of **(P2)** after the approximation. In addition, constraint (14) guarantees that the resource allocation condition, $\sum_t x_t \leq R$, holds true.

5.2 REALMS: Compute $\mathrm{MR_b}$

Given the optimal solution $\mathbf{x}^*$ returned by PALMS, the second step of CONQUER computes the $\mathrm{MR_b}$ of $\mathbf{x}^*$ using REALMS (*computing max **RE**gret using loc**AL** search with **M**ultiple re**S**tarts*) (line 5 in Algorithm 1). Overall, computing $\mathrm{MR_b}$ can be represented as follows:

$$\max_{\mathbf{x}' \in \mathbf{X}, (\mathbf{R^a}, \mathbf{P^a}) \in \mathbf{I}} F(\mathbf{x}', \mathbf{R^a}, \mathbf{P^a}) - F(\mathbf{x}^*, \mathbf{R^a}, \mathbf{P^a}) \tag{18}$$

This optimization problem is also non-convex. It is difficult to apply binary search and piecewise linear approximation (like PALMS) in REALMS since it is a subtraction of two non-convex fractional functions, $F(\mathbf{x}', \mathbf{R^a}, \mathbf{P^a})$ and $F(\mathbf{x}^*, \mathbf{R^a}, \mathbf{P^a})$. Thus, we use local search with multiple starting points to solve $\mathrm{MR_b}$.

6 UAV Planning for Payoff Elicitation (PE)

Our final contribution is to provide PE heuristics to select the best UAV path to reduce uncertainty in payoffs. While a UAV visits multiple targets to collect data, planning an

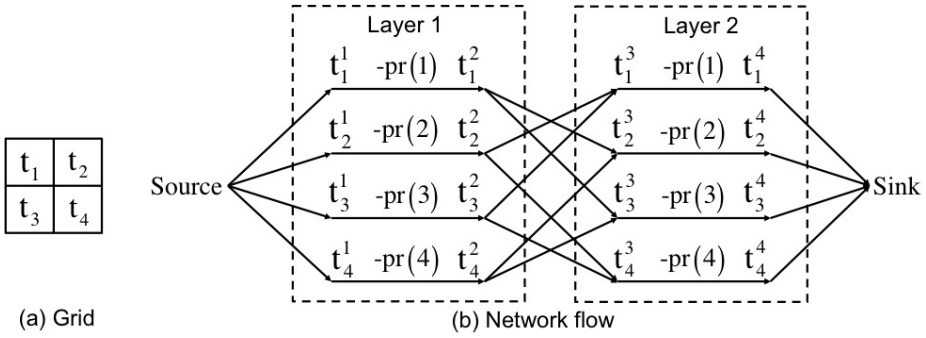


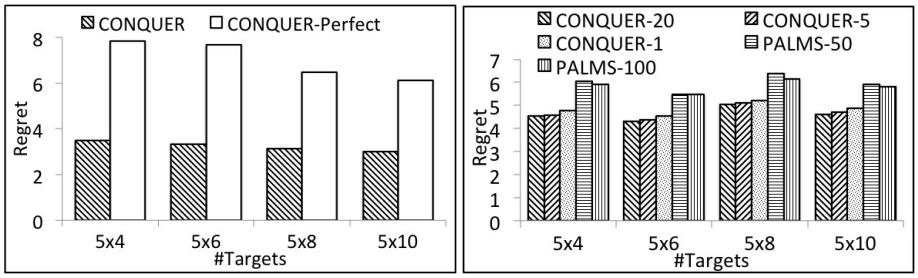
Fig. 2. Min Cost Network Flow

optimal path (which considers all possible outcomes of reducing uncertainty) is computationally expensive. Thus, we introduce the *current solution*-based algorithm which evaluates a UAV path solely based on the $\mathrm{MMR_b}$ solution given current intervals. This current solution idea was introduced in [2] although in a very different uncertainty domain.

We first present a general elicitation process for UAV planning (Algorithm 2). The input includes the defender's initial budget B (e.g., the usage duration of UAVs), the regret barrier δ which indicates how much regret (utility loss) the defender is willing to sacrifice, and the uncertainty intervals $\mathbf{I}$. The elicitation process consists of multiple rounds of flying a UAV and stops when the UAV cost or the defender's regret meets the stopping condition. At each round, CONQUER is applied to compute the optimal $\mathrm{MMR_b}$ solution given current $\mathbf{I}$; CONQUER then outputs the regret r, the optimal strategy $\mathbf{x}^*$, and the corresponding most unfavorable strategy and payoffs $(\mathbf{x}'^{,*}, \mathbf{R}^{\mathbf{a},*}, \mathbf{P}^{\mathbf{a},*})$ which provide the defender's max regret (line 4). Then the best UAV path is selected based on these outputs (line 5). The defender controls the UAV to collect data at targets on that path and obtains new uncertainty intervals (line 6). Finally, line 7 updates the UAV flying cost.

The key aspects of Algorithm 2 are in lines 4 and 5 where the $\mathrm{MMR_b}$ solution is computed by CONQUER and the *current solution* heuristic is used to determine the best UAV path. In this heuristic, the *preference value* of a target t, denoted $pr(t)$, is measured according to the distance in the defender utility between the optimal strategy $\mathbf{x}^*$ and the most unfavorable strategy $\mathbf{x}'^{,*}$ against attacker payoffs $(\mathbf{R}^{\mathbf{a},*}, \mathbf{P}^{\mathbf{a},*})$ at that target, which can be computed as follows: $pr(t) = q_t(\mathbf{x}^*, \mathbf{R}^{\mathbf{a},*}, \mathbf{P}^{\mathbf{a},*}) U_t^d(\mathbf{x}^*, \mathbf{R}^{\mathbf{d}}, \mathbf{P}^{\mathbf{d}}) - q_t(\mathbf{x}'^{,*}, \mathbf{R}^{\mathbf{a},*}, \mathbf{P}^{\mathbf{a},*}) U_t^d(\mathbf{x}'^{,*}, \mathbf{R}^{\mathbf{d}}, \mathbf{P}^{\mathbf{d}})$ where $\mathbf{R}^{\mathbf{d}} = -\mathbf{P}^{\mathbf{a},*}$ and $\mathbf{P}^{\mathbf{d}} = -\mathbf{R}^{\mathbf{a},*}$. Intuitively, targets with higher preference values play a more important role in reducing the defender's regret. We use the sum of preference values of targets to determine the best UAV path based on the two heuristics:

Greedy heuristic: The chosen path consists of targets which are iteratively selected with the maximum pr value and then the best neighboring target of the previously chosen one. **MCNF heuristic:** This path selection problem can be formulated as a special case of the orienteering problem [21]. We can represent the problem as a Min Cost Network Flow (MCNF) where the cost of choosing a target t is $-pr(t)$. For example, there is a grid of four cells (t_1, t_2, t_3, t_4) (Figure 2(a)) where each cell is associated with



(a) CONQUER regret (b) Influence of parameters

Fig. 3. Solution quality of CONQUER

its preference value, namely $(pr(1), pr(2), pr(3), pr(4))$. Suppose that a UAV covers a path of two cells every time it flies and its entry locations (where the UAV takes off or land) can be at any cell. The MCNF for UAV planning is shown in Figure 2(b) which has two layers where each cell t_i has four copies $(t_i^1, t_i^2, t_i^3, t_i^4)$ with edge costs $c(t_i^1, t_i^2) = c(t_i^3, t_i^4) = -pr(i)$. The connectivity between these two layers corresponds to the grid connectivity. There are *Source* and *Sink* nodes which determine the UAV entry locations. The edge costs between the layers and between the *Source* or *Sink* to the layers are set to zero.

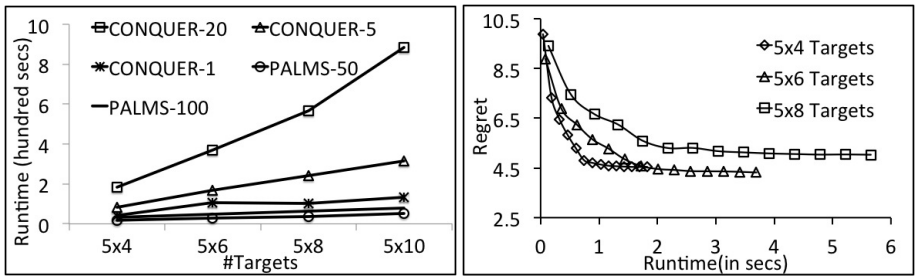
7 Experimental Results

We evaluate solution quality and runtime of CONQUER and PE heuristics in zero-sum games, assuming an SUQR attacker. This section presents key experimental results (more results are in Online Appendix D). We use CPLEX for our algorithms and Fmincon of MATLAB on a 2.3 GHz/4 GB RAM machine. *Key comparison results are statistically significant under bootstrap-t ($\alpha = 0.05$) [22].*

7.1 Synthetic Data

We first conduct experiments using synthetic data to simulate a wildlife protection area. We assume the area is divided into a grid where each cell represents a target, and we create different payoff structures using these grid cells. Each data point in our results is averaged over *40 payoff structures* randomly generated by GAMUT [16]. The attacker reward/defender penalty refers to the animal density while the attacker penalty/defender reward refers to, for example, the amount of snares that are estimated to be confiscated by the defender [23]. Here, the defender's regret indicates the animal loss and thus can be used as a measure for the defender's patrolling effectiveness. Upper and lower bounds for payoff intervals are generated randomly from [-14, -1] for penalties and [1, 14] for rewards with an interval size of 4.0.

Solution Quality of CONQUER. The results are shown in Figure 3 where the x-axis is the grid size (number of targets) and the y-axis is the defender's max regret. First, we demonstrate the importance of handling the attacker's bounded rationality in CONQUER by comparing solution quality of CONQUER with CONQUER-Perfect (an extension of the MMR algorithm for a perfectly rational attacker [14] that addresses un-



(a) Runtime vs #Targets (b) Runtime vs Regret

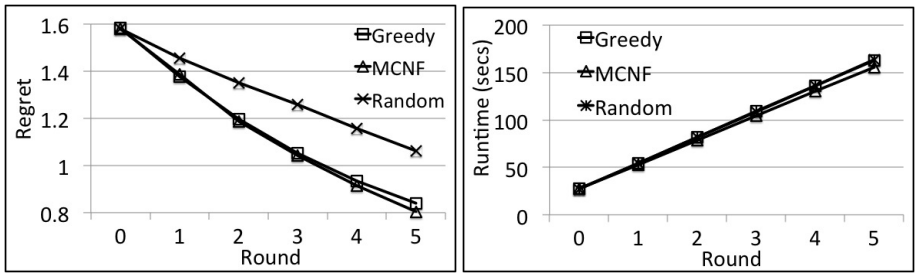
Fig. 4. Runtime performance of CONQUER

certainty in *both* players' payoffs (described in Online Appendix B)). The defender's regret obtained by playing CONQUER and CONQUER-Perfect against the SUQR attacker is shown in Figure 3(a). The defender's regret significantly increases when playing CONQUER-Perfect's strategies, which shows the importance of behavioral minimax regret.

Second, we examine how CONQUER's parameters influence the $\mathrm{MMR_b}$ solution quality; which we show later affects its runtime-solution quality tradeoff. We examine whether the defender's regret significantly increases if (i) the number of starting points in REALMS decreases (i.e., CONQUER with 20 (CONQUER-20), 5 (CONQUER-5) and 1 (CONQUER-1) starting points for REALMS and 40 iterations to iteratively add 40 payoff samples into the set S), or (ii) when CONQUER only uses PALMS (without REALMS) to solve relaxed $\mathrm{MMR_b}$ (i.e., PALMS with 50 (PALMS-50) and 100 (PALMS-100) uniformly random payoff samples). Figure 3(b) shows that the number of starting points in REALMS does not have an impact on solution quality. In particular, CONQUER-1's solution quality is approximately the same as CONQUER-20 after 40 iterations. This result shows that the shortcoming of local search in REALMS (where solution quality depends on the number of starting points) is compensated by a sufficient number (e.g., 40) of iterations in CONQUER; hence the number of REALMS' starting points has low impact. Further, as PALMS-50 and PALMS-100 only solve relaxed $\mathrm{MMR_b}$, they both lead to much higher regret. Thus, it is important to include REALMS in CONQUER.

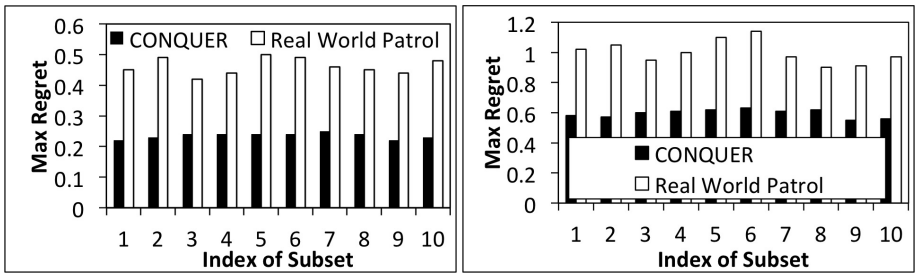
Runtime Performance of CONQUER. Figure 4(a) shows the runtime of CONQUER with different parameter settings. In all settings, CONQUER's runtime linearly increases in the number of targets. Further, Figure 3(a) shows that CONQUER-1 obtains approximately the same solution quality as CONQUER-20 while running significantly faster (Figure 4(a)). This result shows that one starting point of REALMS might be adequate for solving $\mathrm{MMR_b}$ in considering the trade-off between runtime performance and solution quality. Figure 4(b) plots the trade-off between runtime and the defender's regret in 40 iterations of CONQUER-20 for 20-40 targets which shows useful anytime profile.

Payoff Elicitation. We evaluate our PE strategies using synthetic data of 5×5-target (target = 2×2 km cell) games. The length of a UAV path is 3 cells. In addition, the budget for flying a UAV is set to 5 rounds. We assume the uncertainty interval is reduced by half after each round. Our purpose is to examine how the defender's regret is



(a) Solution quality (b) Runtime performance

Fig. 5. UAV planning: uncertainty reduction over rounds



(a) Small interval (b) Larger interval

Fig. 6. Real world max regret comparison

reduced over different rounds. The empirical results are shown in Figure 5 where the x-axis is the number of rounds and the y-axis is the regret obtained after each round (Figure 5(a)) or the accumulated runtime of the elicitation process over rounds (Figure 5(b)). We compare three heuristics: 1) randomly choosing a path (Random) 2) Greedy, and 3) MCNF. Figure 5 shows that the defender's regret is reduced significantly by using Greedy and MCNF in comparison with Random. As mentioned, the difference are statistically significant ($\alpha = 0.05$). Also, both Greedy and MCNF run as quickly as Random. Finally, we have conducted proof-of-concept tests to show that our MCNF strategy was executed by a quadcopter UAV via a field trial in South Africa (video link: *http://vimeo.com/user34515953/videos*).

7.2 Real-world Data

Lastly, we use our WWF dataset (Section 3) to analyze the difference between past patrols conducted by rangers (in the wildlife park) and the patrol strategies generated by CONQUER. Out of 244 grid cells (targets), we pick 25 cells (chosen randomly). Before these wildlife areas were patrolled, there was uncertainty in the features values at those areas. We simulate these conditions faced by real world patrollers by introducing uncertainty intervals in the real-world rewards and penalties on each target in two cases: a small and a larger interval of sizes 0.5 and 1 respectively. For both cases, we compared the max regret achieved by the real world patrols with the max regret of CONQUER's patrols. Figures 6(a) and 6(b) compare the max regret achieved by CONQUER and real

world patrols for 10 different randomly generated subsets of 25 targets when the uncertainty interval size is 0.5 and 1 respectively. The x-axis refers to 10 different subsets and the y-axis is the corresponding max regret. These figures clearly show that CONQUER generates patrols having significantly less regret as compared to real-world patrols.

8 Conclusion

In summary, this paper focuses on solving green security problems while providing the following main contributions: 1) we for the first time compare key behavioral models such as SUQR and QR on real-world wildlife protection data and show SUQR's usefulness for SSG in predicting adversary decisions; 2) we propose a novel algorithm, CONQUER, to solve the behavioral MMR problem which addresses both the attacker's bounded rationality and uncertainty in both players' payoffs; and 3) we introduce new PE strategies for mobile sensors, e.g., UAV to reduce payoff uncertainty.

9 Acknowledgement

This research was supported by MURI Grant W911NF-11-1-0332 and by CREATE Grant 2010-ST-061-RE0001. This research was conducted under the collaboration with World Wildlife Fund, Inc.[6] and The ShadowView Foundation[7].

References

1. N. Basilico, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *AAMAS*, 2009.
2. C. Boutilier, R. Patrascu, P. Poupart, and D. Schuurmans. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artificial Intelligence*, 2006.
3. D. Braziunas and C. Boutilier. Assessing regret-based preference elicitation with the utpref recommendation system. In *EC*, 2010.
4. M. Brown, W. B. Haskell, and M. Tambe. Addressing scalability and robustness in security games with multiple boundedly rational adversaries. In *GameSec*, 2014.
5. D. P. De Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of operations research*, 2004.
6. S. French. *Decision theory: an introduction to the mathematics of rationality*. Halsted Press, 1986.
7. W. B. Haskell, D. Kar, F. Fang, M. Tambe, S. Cheung, and L. E. Denicola. Robust protection of fisheries with compass. In *IAAI*, 2014.
8. C. Kiekintveld, T. Islam, and V. Kreinovich. Security games with interval uncertainty. In *AAMAS*, 2013.
9. D. Korzhyk, V. Conitzer, and R. Parr. Complexity of computing optimal stackelberg strategies in security resource allocation games. In *AAAI*, 2010.
10. J. Letchford and Y. Vorobeychik. Computing randomized security strategies in networked domains. In *AARM*, 2011.

[6] http://www.worldwildlife.org/
[7] http://www.shadowview.org/

11. D. McFadden. Conditional logit analysis of qualitative choice behavior. Technical report, 1972.
12. R. McKelvey and T. Palfrey. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38, 1995.
13. M. Montesh. Rhino poaching: A new form of organised crime1. Technical report, University of South Africa, 2013.
14. T. H. Nguyen, A. Yadav, B. An, M. Tambe, and C. Boutilier. Regret-based optimization and preference elicitation for stackelberg security games with uncertainty. In *AAAI*, 2014.
15. T. H. Nguyen, R. Yang, A. Azaria, S. Kraus, and M. Tambe. Analyzing the effectiveness of adversary modeling in security games. In *AAAI*, 2013.
16. E. Nudelman, J. Wortman, Y. Shoham, and K. Leyton-Brown. Run the gamut: A comprehensive approach to evaluating game-theoretic algorithms. In *AAMAS*, 2004.
17. G. Secretariat. Global tiger recovery program implementation plan: 2013-14. *Report, The World Bank, Washington, DC*, 2013.
18. E. Shieh, B. An, R. Yang, M. Tambe, C. Baldwin, J. DiRenzo, B. Maule, and G. Meyer. Protect: A deployed game theoretic system to protect the ports of the united states. In *AAMAS*, 2012.
19. E. J. Stokes. Improving effectiveness of protection efforts in tiger source sites: developing a framework for law enforcement monitoring using mist. *Integrative Zoology*, 2010.
20. M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.
21. P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden. The orienteering problem: A survey. *EJOR*, 2011.
22. R. Wilcox. *Applying contemporary statistical techniques*. Academic Press, 2002.
23. R. Yang, B. Ford, M. Tambe, and A. Lemieux. Adaptive resource allocation for wildlife protection against illegal poachers. In *AAMAS*, 2014.
24. R. Yang, F. Ordonez, and M. Tambe. Computing optimal strategy against quantal response in security games. AAMAS, 2012.
25. Z. Yin, A. X. Jiang, M. Tambe, C. Kiekintveld, K. Leyton-Brown, T. Sandholm, and J. P. Sullivan. Trusts: Scheduling randomized patrols for fare inspection in transit systems using game theory. *AI Magazine*, 2012.