When Security Games Go Green: 
Designing Defender Strategies to Prevent Poaching and Illegal Fishing

Fei Fang\textsuperscript{1}, Peter Stone\textsuperscript{2}, Milind Tambe\textsuperscript{1} \\
\textsuperscript{1}University of Southern California, Los Angeles, United States \\
\textsuperscript{2}University of Texas at Austin, Austin, United States \\
\textsuperscript{1}\{feifang,tambe\}@usc.edu, \textsuperscript{2}pstone@cs.utexas.edu

Abstract

Building on the successful applications of Stackelberg Security Games (SSGs) to protect infrastructure including airports [Pita \textit{et al.}, 2008], ports [Shieh \textit{et al.}, 2012] and trains [Yin \textit{et al.}, 2012], researchers are now applying game theory to green security domains such as protection of endangered animals and fish stocks. Previous efforts in these domains optimize defender strategies based on the standard Stackelberg assumption that the adversaries become fully aware of the defender’s strategy before taking action. Unfortunately, this assumption is inappropriate since adversaries in green security domains often lack the resources to fully track the defender strategy. This paper (i) introduces Green Security Games (GSGs), a novel game model for green security domains with a generalized Stackelberg assumption; (ii) provides algorithms to plan effective sequential defender strategies — such planning was absent in previous work; (iii) proposes a novel approach to learn adversary models that further improves defender performance; and (iv) provides detailed experimental analysis of proposed approaches.

1 Introduction

Poaching and illegal over-fishing are critical international problems leading to destruction of ecosystems. For example, three out of nine tiger species have gone extinct in the past 100 years and others are now endangered due to poaching [Secretariat, 2013]. Law enforcement agencies in many countries are hence challenged with applying their limited resources to protecting endangered animals and fish stocks.

Building upon the success of applying SSGs to protect infrastructure including airports [Pita \textit{et al.}, 2008], ports [Shieh \textit{et al.}, 2012] and trains [Yin \textit{et al.}, 2012], researchers are now applying game theory to green security domains, e.g., protecting fisheries from over-fishing [Brown \textit{et al.}, 2014; Haskell \textit{et al.}, 2014] and protecting wildlife from poaching [Yang \textit{et al.}, 2014]. There are several key features in green security domains that introduce novel research challenges. First, the defender is faced with multiple adversaries who carry out repeated and frequent illegal activities (attacks), yielding a need to go beyond the one-shot SSG model. Second, in carrying out such frequent attacks, the attackers generally do not conduct extensive surveillance before performing an attack and spend less time and effort in each attack, and thus it becomes more important to model the attackers’ bounded rationality and bounded surveillance. Third, there is more attack data available in green security domains than in infrastructure security domains, which makes it possible to learn the attackers’ decision making model from data.

Previous work in green security domains [Yang \textit{et al.}, 2014; Haskell \textit{et al.}, 2014] models the problem as a game with multiple rounds and each round is a SSG [Yin \textit{et al.}, 2010] where the defender commits to a mixed strategy and the attackers respond to it. In addition, they address the bounded rationality of attackers using the SUQR model [Nguyen \textit{et al.}, 2013]. While such advances have allowed these works to be tested in the field, there are three key weaknesses in these efforts. First, the Stackelberg assumption in these works — that the defender’s mixed strategy is fully observed by the attacker via extensive surveillance before each attack — can be unrealistic in green security domains as mentioned above. Indeed, the attacker may experience a delay in observing how the defender strategy may be changing over time, from round to round. Second, since the attacker may lag in observing the defender’s strategy, it may be valuable for the defender to plan ahead; however these previous efforts do not engage in any planning and instead rely only on designing strategies for the current round. Third, while they do exploit the available attack data, they use Maximum Likelihood Estimation (MLE) to learn the parameters of the SUQR model for individual attackers which we show may lead to skewed results.

In this paper, we offer remedies for these limitations. First, we introduce a novel model called Green Security Games (GSGs). Generalizing the perfect Stackelberg assumption, GSGs assume that the attackers’ understanding of the defender strategy may not be up-to-date and can be instead approximated as a convex combination of the defender strategies used in recent rounds. Previous models in green security domains, e.g., such as [Yang \textit{et al.}, 2014; Haskell \textit{et al.}, 2014] can be seen as a special case of GSGs, as they assume that the attackers always have up-to-date information, whereas GSGs allow for more generality and hence planning of defender strategies.

Second, we provide two algorithms that plan ahead — the generalization of the Stackelberg assumption introduces a need to plan ahead and take into account the effect of de-
fender strategy on future attacker decisions. While the first
algorithm plans a fixed number of steps ahead, the second one
designs a short sequence of strategies for repeated execution.

Third, the paper also provides a novel framework that incor-
porates learning of parameters in the attackers’ bounded
rationality model into the planning algorithms where, instead
of using MLE as in past work, we use insights from Bayesian
updating. All proposed algorithms are fully implemented and
we provide detailed empirical results.

2 Motivation and Defining GSGs

Our motivating example assumes a perfectly rational attacker
purely for simplicity of exposition. In the rest of the paper,
we consider attackers with bounded rationality.

Example 1. Consider a
ranger protecting a large area
with rhinos. The area is di-
vided into two subareas \(N_1\) and
\(N_2\) of the same importance.
The ranger chooses a subarea
to guard every day and she can
stop any snaring by poachers in
the guarded area. The ranger has been using a uniform random
strategy throughout last year. So for this January, she
can choose to continue using the uniform strategy throughout
the month, catching 50% of the snares. But now assume that
the poachers change their strategy every two weeks based on
the most recently observed ranger strategy. In this case, the
ranger can catch 75% of the snares by always protecting \(N_1\)
in the first two weeks of January, and then switching to always
protecting \(N_2\): At the beginning of January, the poachers are
indifferent between the two subareas due to their observation
from last year. Thus, 50% of the snares will be placed in \(N_1\)
and the ranger can catch these snares in the first half of Jan-
buary by only protecting \(N_1\). But after observing the change
in ranger strategy, the poachers will switch to only putting
the snares in \(N_2\). The poachers’ behavior change can be ex-
pected by the ranger and the ranger can catch 100% of the
snares by only protecting \(N_2\) starting from mid-January. (Of
course the poachers must then be expected to adapt further).

This example conceptually shows that the defender can
benefit from planning strategy changes in green security do-
 mains. We now define GSG as an abstraction of the problem
between a defender and attackers due to differences
in resource richness and accessibility. We therefore associate
each target \(i \in [N]\) with payoff values. A mixed defender
strategy can be represented compactly by a coverage vector
\(c = (c_i)\) where \(0 \leq c_i \leq 1\) is the probability that target
\(i\) is covered by some guard and it satisfies \(\sum_{i=1}^{N} c_i \leq K\)
[Kiekintveld et al., 2009; Korzhyk et al., 2010]. If an at-
tacker attacks target \(i\), the expected utility for the defender is
\(\Xi_i^d(c) = c_i R_i^d + (1 - c_i) P_i^d\) given defender strategy \(c\).
We denote the mixed defender strategy in round \(t\) as \(c^t\).

Definition 2. A GSG attacker is characterized by his mem-
ory length \(\Gamma\), coefficients \(\alpha_\tau, \ldots, \alpha_0\) and his parameter vector \(\omega\).
In round \(t\), A GSG attacker with memory length \(\Gamma\) re-
sponds to a convex combination of the defender strategy in
recent \(\Gamma + 1\) rounds, i.e., he responds to \(\eta^t = \sum_{\tau=0}^{\Gamma} \alpha_\tau c^{t-\tau}\)
where \(\sum_{\tau=0}^{\Gamma} \alpha_\tau = 1\) and \(c^t\) if \(t \leq 0\). In every episode of
round \(t\), a GSG attacker follows the SUQR model and chooses
a random target to attack based on his parameter vector \(\omega\) in
the SUQR model.

We aim to provide automated decision aid to defenders
in green security domains who defend against human ad-
versaries such as poachers who have no automated tools —
hence we model the poachers as being boundedly rational and
having bounded memory. We approximate a GSG attacker’s
belief of the defender’s strategy in round \(t\) as a convex com-
bination of the defender strategy in the current round and the
2014; Haskell et al., 2014] can be seen as a special case
of this approximation with \(c_0 = 1\). In this paper, we assume
all attackers have the same memory length \(\Gamma\), coefficients \(\alpha_\tau\),
and these values are known to the defender. \(c^t\) is the defender
strategy used before the game starts and is known to players.

To model the bounded rationality of the human attackers
such as poachers, we use the SUQR model, which has per-
formed the best so far against human subjects in security
games [Nguyen et al., 2013]. In this model, an attacker’s
choice is based on key properties of each target, including the
coverage probability, the reward and the penalty, represented

Each round of the repeated game corresponds to a period of
time, which can be a time interval (e.g., a month) after which
the defender (e.g., warden) communicate with local guards
to assign them a new strategy. We divide each round into
multiple episodes for the players to take actions.

Consistent with previous work on green security games
[Yang et al., 2014; Haskell et al., 2014], we divide the pro-
tected area into subareas or grid cells and treat each subarea
or cell as a target. Different targets may have different im-
portance to the defender and the attackers due to differences
in green security domains who defend against human ad-
vaders such as poachers who have no automated tools —
hence we model the poachers as being boundedly rational and
having bounded memory. We approximate a GSG attacker’s
belief of the defender’s strategy in round \(t\) as a convex com-
bination of the defender strategy in the current round and the
last \(\Gamma\) rounds. This is because the attackers may not be capa-
bale of knowing the defender’s exact strategy when attacking;
naturally, they will consider the information they get from the
past. Further, human beings have bounded memory, and the
attackers may tend to rely on recent information instead of
the whole history. The Stackelberg assumption in [Yang et
al., 2014; Haskell et al., 2014] can be seen as a special case
of this approximation with \(c_0 = 1\). In this paper, we assume
all attackers have the same memory length \(\Gamma\), coefficients \(\alpha_\tau\),
and these values are known to the defender. \(c^t\) is the defender
strategy used before the game starts and is known to players.

To model the bounded rationality of the human attackers
such as poachers, we use the SUQR model, which has per-
formed the best so far against human subjects in security
games [Nguyen et al., 2013]. In this model, an attacker’s
choice is based on key properties of each target, including the
coverage probability, the reward and the penalty, represented

![Snare poaching]

Figure 1: Snare poaching

2014; Haskell et al., 2014] can be seen as a special case
of this approximation with \(c_0 = 1\). In this paper, we assume
all attackers have the same memory length \(\Gamma\), coefficients \(\alpha_\tau\),
and these values are known to the defender. \(c^t\) is the defender
strategy used before the game starts and is known to players.

To model the bounded rationality of the human attackers
such as poachers, we use the SUQR model, which has per-
formed the best so far against human subjects in security
games [Nguyen et al., 2013]. In this model, an attacker’s
choice is based on key properties of each target, including the
coverage probability, the reward and the penalty, represented

![Snare poaching]

Figure 1: Snare poaching

2014; Haskell et al., 2014] can be seen as a special case
of this approximation with \(c_0 = 1\). In this paper, we assume
all attackers have the same memory length \(\Gamma\), coefficients \(\alpha_\tau\),
and these values are known to the defender. \(c^t\) is the defender
strategy used before the game starts and is known to players.

To model the bounded rationality of the human attackers
such as poachers, we use the SUQR model, which has per-
formed the best so far against human subjects in security
games [Nguyen et al., 2013]. In this model, an attacker’s
choice is based on key properties of each target, including the
coverage probability, the reward and the penalty, represented

![Snare poaching]

Figure 1: Snare poaching

2014; Haskell et al., 2014] can be seen as a special case
of this approximation with \(c_0 = 1\). In this paper, we assume
all attackers have the same memory length \(\Gamma\), coefficients \(\alpha_\tau\),
and these values are known to the defender. \(c^t\) is the defender
strategy used before the game starts and is known to players.

To model the bounded rationality of the human attackers
such as poachers, we use the SUQR model, which has per-
formed the best so far against human subjects in security
games [Nguyen et al., 2013]. In this model, an attacker’s
choice is based on key properties of each target, including the
coverage probability, the reward and the penalty, represented
Notation

$T$, $N$, $K$: # of rounds, targets and guards, respectively.
$L$, $l$: # of attackers and memory length of attackers.
$c^t$: Defender strategy in round $t$.
$\eta^t$: Attacker's belief of defender strategy in round $t$, which is a convex combination of $c^t$.
$\alpha_{\tau}$: Coefficient of $c^{t-\tau}$ when calculating $\eta^t$.
$\omega^t$: Parameter vector of the SUQR model for attacker $l$. $\omega^t_1$, $\omega^t_2$ and $\omega^t_3$ are the coefficient on $c_t$, $P^t_1$, $P^t_2$ respectively in the SUQR model.
$q_i$: The probability of attacking target $i$.
$E^t$: Defender’s expected utility in round $t$.

\[
q_i(\omega, \eta) = \frac{e^{\omega^T_1 \eta + \omega^T_2 R^t_i + \omega^T_3 P^t_i}}{\sum e^{\omega^T_1 \eta + \omega^T_2 R^t_i + \omega^T_3 P^t_i}}
\]

Table 1: Summary of key notations.

by the parameter vector $\omega = (\omega_1, \omega_2, \omega_3)$. Given $\eta$ as the attacker’s belief (with $\eta^t$, the belief of the coverage probability on target $i$), the probability that an attacker with parameter $\omega$ attacks target $i$ is

Following the work of Yang et al. [2014], in this paper, we assume the group of attackers may have heterogeneous weighting coefficients, i.e., each attacker $l \in [L]$ is associated with a parameter vector $\omega^l = (\omega^l_1, \omega^l_2, \omega^l_3)$. A GSG defender strategy profile $[c]$ is defined as a sequence of defender strategies with length $T$, i.e., $[c] = (c^1, \ldots, c^T)$. The defender’s expected utility in round $t$ is $E^t([c]) = \sum_{i} \sum_c q_i(\omega^c, \eta^t)U^t_i(c^t)$ . The objective of the defender is to find the strategy profile with the highest average expected utility over all rounds, i.e., to maximize $E([c]) = \sum_{t=1}^{T} E^t([c])/T$.

### 3 Planning in GSGs

The defender can potentially improve her average expected utility by carefully planning changes in her strategy from round to round in a GSG. In this section, we consider the case where the attackers’ parameter vectors $\omega^1, \ldots, \omega^L$ are known to the defender. For clarity of exposition, we will first focus on the case where $\alpha_0 = 0$ and $\Gamma = 1$. This is the special case when the attackers have one round memory and have no information about the defender strategy in the current round, i.e., the attackers respond to the defender strategy in the last round. We discuss the more general case in Section 5.

To maximize her average expected utility, the defender could optimize over all rounds simultaneously. However, this approach is computationally expensive when $T$ is large: it needs to solve a non-convex optimization problem with $NT$ variables $(c^t)$ as the defender must consider attacker response, and the attacking probability has a non-convex form (see Equation 1). An alternative is the myopic strategy, i.e., the defender can always protect the targets with the highest expected utility in the current round. However, this myopic choice may lead to significant quality degradation as it ignores the impact of $c^{t+1}$ in the next round.

Therefore, we propose an algorithm named PlanAhead-M (or PA-M) that looks ahead a few steps (see Algorithm 1).

### Algorithm 1 Plan Ahead($\omega$, $M$)

Output: a defender strategy profile $[c]$

1. for $t=1$ to $T$ do
2. $c^t = f$-PlanAhead($c^{t-1}, \omega, \min(T - t + 1, M)$)

PA-M finds an optimal strategy for the current round as if it is the $M^{th}$ last round of the game. If $M = 2$, the defender chooses a strategy assuming she will play a myopic strategy in the next round and end the game. When there are less than $M - 1$ future rounds, the defender only needs to look ahead $T-t$ steps (Line 2). PA-T corresponds to the optimal solution and PA-1 is the myopic strategy. Unless otherwise specified, we choose $1 < M < T$. Function $f$-PlanAhead($c^{t-1}, \omega, m$) solves the following mathematical program (MP).

\[
\max_{c^t, c^{t+1}, \ldots, c^{t+m-1}} \sum_{\tau=0}^{m-1} E^{t+\tau}
\]

s.t. $E^t = \sum_i \sum_c q_i(\omega^c, \eta^t)\mathbb{E}^{t+\tau}_i(c^\tau), \tau = t, \ldots, t + m - 1$

This is a non-convex problem when $m > 0$ and can be solved approximately with local search approaches.

Although we show in the experiment section that PA-2 can provide significant improvement over baseline approaches in most cases, there exist settings where PA-2 cannot perform arbitrarily badly when compared to the optimal solution. The intuition is that the defender might make a suboptimal choice in the current round with an expectation to get a high reward in the next round; however, when she moves to the next round, she plans for two rounds again, and as a result, she never gets a high reward until the last round.

**Example 2.** Consider a guard protecting two subareas with payoff values shown on the right ($X \gg 1$). For simplicity of the example, assume the defender can only choose pure strategies. There is one poacher with a large negative coefficient on coverage probability, i.e., the poacher will always snare in the subarea that is not protected in the last round. The initial defender strategy is protecting $N_1$, meaning the attacker will snare in $N_2$ in round 1. According to PA-2, the defender will protect $N_1$ in round 1 and plan to protect $N_2$ in round 2, expecting a total utility of $3 + X$. However, in round 2, the defender chooses $N_1$ again as she assumes the game ends after round 3. Thus, her average expected utility is $\frac{3(T-1)+X}{T} \approx 3$. On the other hand, if the defender alternates between $N_1$ and $N_2$, she gets a total utility of $X+2$ for two consecutive rounds and her average utility is at least $\frac{X}{2} \gg 3$.

PA-2 fails in such cases because it over-estimates the utility in the future. To remedy this, we generalize PA-M to PA-M-$\gamma$ by introducing a discount factor $0 < \gamma \leq 1$ for future rounds when $T - t < M - 1$, i.e., substituting Equation 2 with

\[
\max_{c^t, c^{t+1}, \ldots, c^{t+m-1}} \sum_{\tau=0}^{m-1} \gamma^{t+\tau} \mathbb{E}^{t+\tau}
\]

While PA-M-$\gamma$ provides an effective way to design sequential defender strategies, we provide another algorithm called FixedSequence-M (FS-M) for GSGs (see Algorithm 2). FS-M not only has provable theoretical guarantees, but may also...
ease the implementation in practice. The idea of FS-M is to find a short sequence of strategies with fixed length $M$ and require the defender to execute this sequence repeatedly. If $M = 2$, the defender will alternate between two strategies and she can exploit the attackers’ delayed response. It can be easier to communicate with local guards to implement FS-M in green security domains as the guards only need to alternate between several types of maneuvers. Function f-FixedSequence($\omega, M$) calculates the best fixed sequence of length $M$ through the following MP.

$$\max_{a_1, \ldots, a_M} \sum_{t=1}^{M} E^t$$

s.t. $E^t = \sum_{i} q_i(\omega^t, \eta^t) U_i^t(a^t)$, $t = 1, \ldots, M$

$$\eta^t = a^M$$

$$\eta^t = a^{t-1}, t = 2, \ldots, M$$

$$\sum_{i} a_i^t \leq K, t = 1, \ldots, M$$

Theorem 1 shows that the solution to this MP provides a good approximation of the optimal defender strategy profile.

**Theorem 1.** In an GSG with $T$ rounds, $a_0 = 0$ and $T = 1$, for any fixed length $1 < M \leq T$, there exists a cyclic defender strategy profile $\{a_i^t\}$ with period $M$ that is a $(1 - \frac{1}{M}) \frac{Z-1}{M+1}$ approximation of the optimal strategy profile in terms of the normalized utility, where $Z = \frac{T}{M^T}$.

We leave the detailed proof to the online appendix\(^1\). According to Theorem 1, when a GSG has many rounds ($T \gg M$), the cyclic sequence constructed by repeating $a_1^t, \ldots, a_M^t$ is a $1 - 1/M$ approximation.

### 4 Learning and Planning in GSGs

In Section 3, we assume that the parameter vectors $\omega^1, \ldots, \omega^N$ in the attackers’ bounded rationality model are known. Since the defender may not know these parameter values precisely at the beginning of the game in practice, we now aim to learn the attackers’ average parameter distribution from attack data. Previous work in green security domains [Yang et al., 2014; Haskell et al., 2014] treats each data point, e.g., each snare or fishnet, as an independent attacker and applies MLE to select the most probable parameter vector. However, some of the assumptions made in previous work in proposing MLE may not always hold as MLE works well when a large number of data samples are used to estimate one set of parameters [Elaision, 1993]. Here we show that estimating $\omega$ from a single data point using MLE can lead to highly biased results.

**Example 3.** Consider a guard protecting two targets in round 1. The payoff structure and initial defender strategy are shown in Table 2 where $X \gg 1$ and $0 < \delta \ll 1$. An attacker with parameter vector $\omega = (-1, 0, 0)$ will choose $N_1$ or $N_2$ with the probability $\approx 0.5$, as $\omega_1 = -1$ means he has

\(^1\)http://ijcai2015.cs.yolasite.com/

---

**Algorithm 2** Fixed Sequence  
Output: defender strategy profile $[\omega]$

1: \( (a_1^t, \ldots, a_M^t) = \text{f-FixedSequence}(\omega, M) \).
2: \text{for} \, t = 1 \, \text{to} \, T \, \text{do}
3: \hspace{5mm} c^t = a_1^{t \mod M} + 1

**Algorithm 3** Learn-BU ($\eta, \chi_i(\hat{\omega}), p$)

Output: $\hat{p}$ - a probability distribution over $\{\hat{\omega}\} = \{\omega^1, \ldots, \omega^N\}$

1: \text{for} \, s = 1 \, \text{to} \, N \, \text{do}
2: \hspace{5mm} \hat{p}_s = \frac{p_s(\hat{\omega}, s)}{\sum_s p_s(\hat{\omega}, s)}
3: \text{for} \, s = 1 \, \text{to} \, S \, \text{do}
4: \hspace{10mm} \hat{p}(s) = \frac{\sum_s \hat{p}_s}{\sum_s \hat{p}_s(s)}$

Table 2: Payoff structure of Example 3.

<table>
<thead>
<tr>
<th>Target</th>
<th>$R_1$</th>
<th>$R_2$</th>
<th>$R_1$</th>
<th>$R_2$</th>
<th>$e_1^0$</th>
<th>$e_2^0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_1$</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>0.5 + $\delta$</td>
<td></td>
</tr>
<tr>
<td>$N_2$</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0.5 - $\delta$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
and tested on 2.4GHz CPU with 128 GB memory. All key
parameters in every round may lead to low utility.
Generally, from Theorem 1 (see online appendix
for details).

Theorem 2. In a GSG with $T$ rounds, for any fixed length
$M < T \leq T$, there exists a cyclic defender strategy profile
$s$ with period $M$ that is a $(1 - \frac{1}{e}) \frac{Z}{Z+1}$ approximation of the optimal strategy profile in terms of the normalized utility,
where $Z = \left\lceil \frac{T-1}{G-1} \right\rceil$.

6 Experimental Results
We test all the proposed algorithms on GSGs motivated by
scenarios in green security domains such as defending against
poaching and illegal fishing. Each round corresponds to 30
days and each poacher/fisherman will choose a target to place
snares/fishnets every day. All algorithms are implemented in
MATLAB with the function used for solving MPs
and tested on 2.4GHz CPU with 128 GB memory. All key
differences noted are statistically significant ($p < 0.05$).

6.1 Planning Algorithms
We compare proposed planning algorithms PA-M(-$\gamma$) and
FS-M with baseline approaches FS-1 and PA-1. FS-1 is
equivalent to calculating the defender strategy with a perfect
Stackelberg assumption, which is used in previous work
[Yang et al., 2014; Haskell et al., 2014], as the defender uses
the same strategy in every round and the attackers’ belief coincides with the defender strategy. PA-1 is the myopic strategy
which tries to maximize the defender’s expected utility in
the current round. We assume $c_0$ is the MAXIMIN strategy.
We first consider the special case ($\alpha_0 = 0$, $\Gamma = 1$) and
test on 32 game instances of 5 attackers, 3 targets, 1 guard
and 100 rounds with random reward and penalty chosen from
[0, 10] and [−10, 0] respectively (denoted as Game Set 1). We
run 100 restarts for each MP. Figure 2(a) shows that PA-M(-$\gamma$) and FS-M significantly outperform FS-1 and PA-1 in terms
of the defender’s average expected utility (AEU). This means
using the perfect Stackelberg assumption would be detrimental
to the defender as the attackers respond to last round’s strategy.
For PA-M, adding a discount factor $\gamma$ may improve the
solution. Figure 2(b) shows FS-M takes much less time than
PA-M overall as FS-M only needs to solve one MP
throughout a game while PA-M solves a MP for each round.
We also test on 32 games with 100 attackers, 10 targets, 4
guards and 100 rounds (denoted as Game Set 2) in the special
case (see Figure 2(c)). We set a 1-hour runtime limit for the
algorithms and again, FS-M and PA-M(-$\gamma$) significantly
outperform FS-1 and PA-1 in solution quality.

We then test general cases on Game Set 2. Figure 2(d)
shows the defender’s AEU with varying $\alpha_0$ when $\Gamma = 1$. In
the extreme case of $\alpha_0 = 1$, i.e., the attackers have perfect
knowledge of the current defender strategy, the problem
reduces to a repeated Stackelberg game and all approaches
provide similar solution quality. However, when $\alpha_0 < 0.5$, FS-2
and PA-2 provide significant improvement over FS-1 and PA-
1, indicating the importance of planning.

We further test the robustness of FS-2 when there is slight
deviation in $\alpha_0$ with $\Gamma = 1$ (see Figure 3). For example,
the value of 5.891 in the 2nd row, 8th column of the table
is the defender’s AEU when the actual $\alpha_0 = 0$ and the
defender assumes (estimates) it to be 0.125 when calculating
her strategies. Cells in the diagonal show the case when the
estimation is accurate. Cells in the last row show results for
baseline algorithm FS-1. FS-1 uses the Stackelberg assumption
and thus the estimated value makes no difference. When
the actual value slightly deviates from the defender’s estimate
(cells adjacent to the diagonal ones in the same column), the
solution quality does not change much if the actual $\alpha_0 > 0.5$.
When the actual $\alpha_0 < 0.5$, FS-2 outperforms FS-1 significantly
even given the slight deviation.

In Figure 2(e), we compare algorithms assuming $\Gamma = 2,
\alpha_1 = \alpha_2 = 0.5$ and $\alpha_0 = 0$. As expected, PA-M with $M > 1$
and FS-M with $M > 2$ significantly outperforms FS-1 and
PA-1. The improvement of FS-2 over FS-1 is negligible, as
any fixed sequence of length 2 can be exploited by the attackers
with memory length $= 2$.

Figure 2(f) shows the solution quality of PA-M when the
defender assumes the attackers’ memory length is 3 but the
actual value of $\Gamma$ varies from 1 to 4. When $\Gamma$ is slightly
overestimated (actual $\Gamma = 1$ or 2), PA-M still significantly
outperforms the baseline algorithm FS-1 and PA-1. However,
when $\Gamma$ is under-estimated (actual $\Gamma = 4$), the attackers have
longer memory than the defender’s estimate and thus the
attackers can exploit the defender’s planning. This observation
suggests that it is more robust to over-estimate the attackers’
memory length when there is uncertainty in $\Gamma$. We defer to
future work to learn $\alpha_0$ and $\Gamma$ from attack data.

6.2 Learning and Planning Framework
When the parameter vectors $\{\omega_i^T\}$ are unknown, we compare
Algorithm 3 with the baseline learning algorithm that uses
Figure 2: Experimental results show improvements over algorithms from previous work.

Figure 3: Robustness against uncertainty in $\alpha_0$ when $\Gamma = 1$

MLE (denoted as MLE) when incorporated into planning algorithms. In each game of Game Set 2, we randomly choose $\{\omega^i\}$ for the 100 attackers from a three-dimensional normal distribution with mean $\mu = (-17.81, 0.72, 0.47)$ and covariance $\Sigma = \begin{pmatrix} 209.48 & -2.64 & -0.71 \\ -2.64 & 0.42 & 0.24 \\ -0.71 & 0.24 & 0.36 \end{pmatrix}$. We use BU to denote the case when an accurate prior ($\mu$ and $\Sigma$) is given to the defender. Recall that the defender plays against 100 attackers throughout a game, and BU aims to learn the parameter distribution of these 100 attackers. BU$'$ represents the case when the prior distribution is a slightly deviated estimation (a normal distribution with random $\mu'$ and $\Sigma'$ satisfying $\|\mu - \mu'\| \leq 5$ and $\|\Sigma - \Sigma'\| \leq 5$). KnownPara represents the case when the exact values of $\{\omega\}$ are known to the defender. We set a time limit of 30 minutes for the planning algorithms. Figure 2(g) – 2(h) show that BU and BU$'$ significantly outperform MLE. Indeed, the solution quality of BU and BU$'$ is close to that of KnownPara, indicating the effectiveness of the proposed learning algorithm. Also, BU and BU$'$ run much faster than MLE as MLE solves a convex optimization problem for each target in every round.

7 Conclusion and Related Work

So far, the field had been lacking an appropriate game-theoretic model for green security domains: this paper provides Green Security Games (GSG) to fill this gap. GSG’s generalization of the Stackelberg assumption which is commonly used in previous work has led it provide two new planning algorithms as well as a new learning framework, providing a significant advance over previous work in green security domains [Yang et al., 2014; Haskell et al., 2014].

Additional related work includes criminological work on poaching and illegal fishing [Lemieux, 2014; Beirne and South, 2007], but a game-theoretic approach is completely missing in this line of research. Planning and learning in repeated games against opponents with bounded memory has been studied [Sabourian, 1998; Powers and Shoham, 2005; Chakraborty et al., 2013; de Cote and Jennings, 2010; Banerjee and Peng, 2005]. However, most of the work considers the case where each player chooses one action from his finite action set in each round of the game, while we focus on the problem motivated by real-world green security challenges where the players can choose a mixed strategy and implement it for multiple episodes in each round; thus previous approaches fail to apply in our domains. We further handle multiple boundedly rational attackers each with a different SUQR model, leading to a need to learn heterogeneous parameters in the SUQR model, which was not addressed in this prior work which assume a single fully rational attacker. Previous work on learning in repeated SSGs [Marecki et al., 2012; Letchford et al., 2009; Blum et al., 2014] has mainly focused on learning the payoffs of attackers assuming perfectly rational attackers. In contrast, we not only generalize the Stackelberg assumption to fit green security domains but also provide algorithms to learn the parameters in the attackers’ bounded rationality model. By embedding models of bounded rationality in GSG, we complement previous work that focus on modeling human bounded rationality and bounded memory [Rubinstein, 1997; Cowan, 2005].
Acknowledgement

This research was supported by MURI Grant W911NF-11-1-0332 and by the United States Department of Homeland Security through the Center for Risk and Economic Analysis of Terrorism Events (CREATE) under grant number 2010-ST-061-RE0001. A portion of this work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (CNS-1330072, CNS-1305287), ONR (21C184-01), AFOSR (FA8750-14-1-0070, FA9550-14-1-0087), and Yujin Robot.

References


