Designing Defender Strategies Against Frequent Adversary Interaction

Fei Fang[1], Peter Stone[2], and Milind Tambe[1]

[1] University of Southern California, Los Angeles, CA 90089, USA,
feifang,tambe@usc.edu,
[2] University of Texas at Austin, Austin, TX 78712, USA,
pstone@cs.utexas.edu

Abstract. Recently, there has been an increase in interest in applying game theoretic approaches to domains involving frequent adversary interactions, such as wildlife and fishery protection. In these domains, the law enforcement agency faces adversaries who repeatedly and frequently carry out illegal activities, and thus, do not have time for extensive surveillance before taking actions. This makes them significantly different from counter-terrorism domains where game-theoretic approaches have been widely deployed. This paper presents a game-theoretic approach to be used by the defender in these Frequent Adversary Interaction (FAI) domains. We provide the following contributions: (i) a novel game model for FAI domains, describing the interaction between the defender and the attackers in a repeated game; (ii) two sets of algorithms that plan for the defender strategies to achieve high average expected utility over all rounds; (iii) a novel framework that incorporates planning and learning in the presence of attack data; and (iv) detailed experimental analysis of the proposed algorithms.

Keywords: Frequent Adversary Interaction, Game Theory, Repeated Game, Defender Strategy Profile

1 Introduction

Whereas game theoretic approaches have been widely deployed in various counter-terrorism settings including protecting airports, ports and trains [16,21,25], recently there has been increasing interest in applying game theory to suppressing environmental crimes such as in protecting fisheries from over-fishing [18,9] and protecting rhinos and tigers from illegal poaching [24][3]. Unfortunately, most previous work in counter-terrorism domains cannot be directly applied to these domains because they have three key differences. Firstly, in counter-terrorism domains, an attacker is assumed to conduct extensive surveillance in order to understand the defender's strategy and then executes a one-shot attack. However, in domains involving environmental crime, the law enforcement agency (the defender) is faced with multiple adversaries (the attackers) who carry out repeated and frequent illegal activities (attacks) and generally do not conduct

[3] A revision of this workshop paper is under review at another conference. If it is seen as a conflict, we would withdraw the workshop paper.

extensive surveillance before performing an attack. Secondly, in carrying out such frequent attacks, the attackers generally spend less time and effort in each attack, and thus it becomes more important to model the attacker's bounded rationality and bounded surveillance. Thirdly, there is more attack data available — at least in comparison with the earlier counter-terrorism domains — that can be collected by the defender in such domains. We propose the term *Frequent Attacker Interaction (FAI) domains* to refer to such domains.

There are some recent efforts that have begun to address FAI domains [24,9]. They model the problem as a repeated game and each round is a Stackelberg security game where the defender commits to a mixed strategy and the attackers respond to it; they do address the bounded rationality of attackers using the SUQR model [15]. While such advances have allowed these works to be tested in the field, whether to protect wildlife or fisheries, there are three key weaknesses in these efforts. First, the Stackelberg assumption in these works – that the defender's mixed strategy is fully observable by the attacker via extensive surveillance before each attack – is unrealistic in the context of FAI 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 work 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 (more related work discussed in Section 8).

In this paper, we offer remedies for these three limitations. First, we introduce a novel model called LEAD (LEad Attackers with Delayed observation). Generalizing the perfect Stackelberg assumption, LEAD assumes 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 several recent rounds. In contrast to previous work which has made firm commitment at one end of the spectrum (i.e., assuming that the attacker always has up to date information), LEAD generalizes the assumption and is more flexibility in planning defender strategies.

Second, we provide two sets of algorithms that plan ahead, providing defender strategies in each round. The generalization of the Stackelberg assumption introduces a need to plan ahead and take into account the effect of defender strategy on future attacker decisions. The first set of algorithms we provide plan a fixed number of steps ahead and try to maximize the overall defender expected utility over these next set of rounds starting from the current round. The second set of algorithms design a short sequence of strategies with high average defender expected utility and require the defender to execute this short sequence repeatedly.

The third contribution of the paper is a novel framework that incorporates learning within our LEAD planning framework. Instead of relying on MLE to provide a single set of parameters, we estimate the probability of different parameters using insights from Bayesian updating.

2 Motivating Example

In FAI domains such as protecting against wildlife poaching, the defender organizes a group of patrollers to protect a large area. The area can be divided into subareas or targets, each of different importance to the defender. A population of attackers perform frequent attacks, each confined to one subarea. The defender is a set of patrollers (or rangers) who use attack data to refine their patrols. Example 1 conceptually shows that the defender can benefit from strategy change, and the focus of this paper is to design the sequence of defender strategies. For the simplicity of the example, we assume perfectly rational attackers while in the rest of the paper, we consider attackers with bounded rationality.

Example 1. Consider a ranger who is in charge of protecting a large area with rhinos. The area is divided into two subareas N_1 and N_2, which are of the same importance (both to the defender and to poachers who place snares to trap animals). The ranger chooses a subarea to patrol every day and she can stop any snaring in the patrolled area. The ranger has been using the uniform random strategy throughout last year and is going to decide the strategy to be used this January. She can choose to continue using the uniform strategy throughout January, confiscating 50% of the snares. However, if she always protects N_1 at the beginning of January, and then switches to always protecting N_2 in mid-January, she can confiscate 75% of the snares as explained below. Presumably, the poachers will have no preference between the two subareas at the beginning of January due to their observation from last year. Thus, 50% of the snares will be placed in N_1 and the ranger can confiscate these snares by only protecting N_1. The poachers may realize the strategy change after a period of time (e.g., two weeks) and will then put all the snares in N_2. The poachers' behavior change is expected by the ranger and the ranger can confiscate 100% of the snares by only protecting N_2 starting from mid-January.

3 Problem Setting and the LEAD Game Model

Definition 1 describes the LEAD game model, which is an abstraction of FAI problems, as we explain in more detail below. Table 1 shows key notations used throughout the paper.

Definition 1. *A LEAD game is a T round repeated game between the defender and L attackers and (i) The defender has K patrollers to protect N targets. (ii) In each round t, the defender chooses a (pure or mixed) strategy c^t from strategy set C, represented by the coverage probability on each target. (iii) All attackers have memory length Γ and respond to a known convex combination of the defender strategy in recent $\Gamma+1$ rounds, i.e., in every round t, attackers respond to $\eta^t = \sum_{\tau=0}^{\Gamma} \alpha_\tau c^{t-\tau}$ where $\sum_{\tau=0}^{\Gamma} \alpha_\tau = 1$ and $c^t = c^0$ if $t \leq 0$. (iv) Each attacker $l \in \mathbf{L}$ has a parameter vector $\omega^l =< \omega_1^l, \omega_2^l, \omega_3^l >$, and responds to η^t in round t according to SUQR model with parameter ω^l for a total of Q times per round.*

Each round of the repeated game corresponds to a period for which the defender (e.g., the ranger team) executes patrols. At the beginning of each round, the defender

Table 1. Summary of key notations.

Notation		Notation	
T	Total number of rounds.	N	Total number of targets.
K	Number of defender resources.	Q	Number of episodes in each round.
L	Total number of attackers.	c^t	Defender strategy in round t.
C	Set of available defender strategies.	η^t	Attackers' belief of the defender's strategy in round t.
Γ	Memory length of attackers, in number of rounds.	α_τ	Coefficient of defender strategy in round $t-\tau$.
ρ	Payoff structure, including P_i^a, R_i^a, P_i^d, R_i^d.	ω^l	Parameter vector of the SUQR model for attacker l. ω_1^l, ω_2^l and ω_3^l are the coefficient on c_i, R_i^a, P_i^a respectively.
q_i	The probability of attacking target i.	E^t	Defender's expected utility in round t.
S	Number of sampled parameter vectors during learning.	p^t	Prior distribution of attackers' parameters in round t.
χ^t	Attack data in round t.		

chooses a strategy from strategy set C. A pure strategy of the defender is a feasible assignment of the patrollers to targets. A mixed strategy for the defender is a randomization over pure strategies, and it can be represented compactly by a coverage vector $c = \langle c_i \rangle$ where c_i is the probability that target i is covered by some patroller [10,11]. We divide each round into Q episodes for the players to take actions. In every episode of round t, the patrollers are assigned to protect targets according to an assignment sampled from c^t and each attacker chooses a target to attack. We assume the choice of each attacker is independent as the area is large and different attackers have different preferences. We approximate the attackers' shared understanding (or attackers' belief) of the defender's strategy in round t as $\eta^t = \sum_{\tau=0}^{\Gamma} \alpha_\tau c^{t-\tau}$, a convex combination of the defender strategy executed in the current round and the last Γ rounds. This is because the attackers may not be capable of knowing the exact strategy used by the defender when they perform attacks. Naturally, they will take into consideration 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. Note that the Stackelberg assumption can be seen as a special case of this approximation with $\alpha_0 = 1$. Before the game starts, the attackers have an initial belief c^0 of the defender strategy based on historical data or previous defender commitment and thus is known to the defender.

Different targets may have different importance to the defender and the attackers due to differences in resource richness, terrain and accessibility. We therefore associate each target $i \in [N]$ with payoff values P_i^a, R_i^a, P_i^d, R_i^d. If an attacker attacks target i and it is protected by a defender resource, the attacker gets utility P_i^a (P stands for penalty) and the defender gets utility R_i^d (R stands for reward). If target i is not protected, the attacker gets utility R_i^a and the defender gets utility P_i^d. Following previous work on security games [26], we require $R_i^d > P_i^d$ and $R_i^a > P_i^a$. If the defender strategy is c and an attacker attacks target i, the expected utility for the defender is $U_i^d(c) = c_i R_i^d + (1 - c_i) P_i^d$.

We adopt the Bayesian SUQR model [24] for multiple attackers. It is well understood that human beings are boundedly rational. The SUQR model has performed the best so far when tested against human subjects, and used to model the bounded rationality in security domains. In this model, an attacker's choice is based on an evaluation of key properties of each target, including the coverage probability, the reward and the penalty, represented by the parameter vector $\omega = (\omega_1, \omega_2, \omega_3)$. If the attackers respond to defender strategy η, the probability that an attacker with parameter ω attacks target i is

$$q_i(\omega, \eta) = \frac{e^{\omega_1 \eta_i + \omega_2 R_i^a + \omega_3 P_i^a}}{\sum_j e^{\omega_1 \eta_j + \omega_2 R_j^a + \omega_3 P_j^a}} \tag{1}$$

The Bayesian SUQR model is based on the SUQR model and captures the heterogeneity of a group of attackers. It assumes different attackers have different parameters. So in our game model, we assume each attacker $l \in [L]$ is associated with a parameter vector $\omega^l = (\omega_1^l, \omega_2^l, \omega_3^l)$. The defender's utility in round t (E^t) is the total expected utility over all attackers. Given that the attackers respond to η^t in round t,

$$E^t = \sum_l E_l^t = \sum\nolimits_l \sum\nolimits_i q_i(\omega^l, \eta^t) U_i^d(c^t) \tag{2}$$

A LEAD defender strategy profile $[c]$ is defined as a strategy sequence of length T, i.e., $[c] = \langle c^1, ..., c^T \rangle$. The utility of a defender strategy profile is defined as the average expected utility for the defender over all rounds, i.e., $E([c]) = \sum_{t=1}^{T} E^t([c])/T$. The objective of the defender is to find the strategy profile with the highest utility $E([c])$.

4 Planning

In a LEAD game, the attackers' belief of the defender strategy is partially based on information from the previous rounds unless $\alpha_0 = 1$. In the rest of the paper, we focus on the case $\alpha_0 < 1$ unless otherwise specified. Therefore, the defender can potentially benefit from changing her strategy from round to round. In this section, we consider the case where parameter vectors of the attackers, i.e., $\omega^l, l = 1..L$, 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., in round t, the attackers respond to the defender strategy in round $t - 1$. We discuss the more general case in Section 6.

When strategy set C is finite, we can find an optimal defender strategy profile in polynomial time using dynamic programming. The idea is to break down the problem of finding a strategy sequence of length T into solving a set of sub-problems by specifying different strategies in the first round. Thus each sub-problem looks for a sequence of strategies with length $T - 1$.

Theorem 1. *In a LEAD game with $\alpha_0 = 0$ and $\Gamma = 1$, there exists an algorithm that runs in $poly(T, |C|)$ time and finds an optimal defender strategy profile when C is a finite set.*

When C contains an infinite number of strategies, although it is possible to discretize the strategy space into meshed grids, the size of the discretized strategy set is exponential in the number of the targets, making it hard to compute the optimal strategy profile. So, we propose two sets of algorithms that can handle this more general situation. We will focus on the case where there are no scheduling constraints, i.e., coverage vectors satisfying $\sum c_i \leq K$ can be realized [11].

To maximize the average defender expected utility $E([c])$, the defender could optimize over all NT variables ($c_i^t \in [0,1]$, $t = 1..T$) in the defender strategy profile simultaneously. However, given the non-convex form of attacking probability in Equation 1, this approach is computationally prohibitive when T is large. Note that in round t, the attackers' strategy can be calculated from c^{t-1} and therefore is known to the defender. A myopic defender would protect the targets that give the highest expected utility in the current round. This myopic choice is optimal if it is the last round of the game. However, it may lead to significant quality degradation as it ignores the impact of c^t in the next round.

Therefore, we provide a set of algorithms named PlanAhead-M (or PA-M) that look ahead a few steps (see Algorithm 1). The insight is to find an optimal strategy for the current round t as if it is the M^{th} last round of the game. For $M = 2$, the defender chooses a strategy c^t assuming she will play a myopic strategy in round $t+1$ and end the game. Note that for $t > T - M + 1$, there are less than M rounds left and the defender only needs to consider $T - t$ future rounds. PA-T corresponds to the global optimal solution and PA-1 corresponds to the myopic strategy.

Algorithm 1 PlanAhead(ω, M)

Output: a defender strategy profile $[c]$

1: **for** t=1 to T **do**
2: $\quad c^t = \text{f-PlanAhead}([c]^{t-1}, \omega, \rho, \min\{T-t+1, M\})$
3: **end for**

$[c]^{t-1}$ represents the defender strategies before round t. Function f-PlanAhead is implemented by solving the following mathematical program (MP).

$$\max_{c^t, c^{t+1}, \ldots c^{t+m-1}} \sum_{\tau=0}^{m-1} E^{t+\tau} \tag{3}$$

$$s.t\ E^\tau = \sum_l \sum_i q_i(\omega^l, \eta^\tau) U_i^d(c^\tau), \tau = t, .., t+m-1 \tag{4}$$

$$\eta^\tau = c^{\tau-1}, \tau = t, .., t+m-1 \tag{5}$$

$$\sum_i c_i^\tau = K, \tau = t, .., t+m-1 \tag{6}$$

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 can *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. Note that the reason PA-2 fails in such cases is because it over-estimates the utility in the next round. To remedy this, we generalize PA-M by introducing a discount factor $0 < \gamma \leq 1$ for future rounds (denoted as PA-M-γ). Instead of using Equation 3, we use Equation 7 as the objective function if the expected utility in future rounds may be over-estimated, i.e., $t + M - 1 < T$.

$$\max_{c^t, c^{t+1}, \ldots c^{t+m-1}} \sum_{\tau=0}^{m-1} \gamma^\tau E^{t+\tau} \tag{7}$$

In addition to PA-M-γ, we provide another set of algorithms called FixedSequence-M (FS-M) which have provable theoretical guarantees (see Algorithm 2). 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 of strategies repeatedly. Take $M = 2$ as an example. Consider a defender strategy that alternates between two mixed strategies a^1 and a^2, i.e., a^1 is used in odd rounds, and a^2 is used in even rounds. Thus, the defender knows that the attacker responds to a^2 in odd rounds and responds to a^1 in even rounds. This makes it possible for the defender to exploit the attackers' delayed response. f-FixedSequence(ω, ρ, M) calculates the optimal fixed sequence of length M through the following MP.

$$\max_{a^1, \ldots, a^M} \sum_{t=1}^{M} E^t \tag{8}$$

$$s.t\ E^t = \sum_l \sum_i q_i(\omega^l, \eta^t) U_i^d(a^t), t = 1, ..., M \tag{9}$$

$$\eta^1 = a^M \tag{10}$$

$$\eta^t = a^{t-1}, t = 2, ..., M \tag{11}$$

$$\sum_i a_i^t \leq K, t = 1, .., M \tag{12}$$

Algorithm 2 Fixed Sequence

Output: defender strategy profile $[c]$

1: $(a^1, ..., a^M) =$ f-FixedSequence(ω, ρ, M).
2: **for** t=1 to T **do**
3: $\quad c^t = a^{(t \bmod M)+1}$
4: **end for**

Although FS-M provides no guarantee on the defender's utility in the first round (as c_0 is not considered in this algorithm), the utility of the strategy profile with sufficiently many rounds is close to the optimal average defender utility of the above MP. We further show that the solution of this MP provides a good approximation of the optimal defenders strategy profile in Theorem 2.

Theorem 2. *In a LEAD game with T rounds, $\alpha_0 = 0$ and $\Gamma = 1$, for any fixed length $1 < M \leq T$, there exists a cyclic defender strategy profile $[s]$ with period M that is a $(1 - \frac{1}{M})\frac{Z-1}{Z+1}$ approximation of the optimal strategy profile in terms of the normalized utility, where $Z = \lceil \frac{T}{M} \rceil$.*

The intuition is to divide the optimal sequence into sections with length $M-1$ and compare the average expected utility of each section with the average expected utility of the optimal fixed sequence. We normalize the value of $E^t([c])$ and $E([c])$ to $[0,1]$ by defining $U^t([c]) = \frac{E^t([c])-minU}{maxU-minU}$ and $U([c]) = \frac{E([c])-minU}{maxU-minU}$ where $minU = \min\{\rho\}$ and $maxU = \max\{\rho\}$. The optimal strategy profile has the highest utility and the highest normalized utility. Denote the optimal normalized utility as U^{opt}.

Definition 2. *A cyclic defender strategy profile for a LEAD game is a profile consisting of a cyclic sequence of strategies, i.e., $\exists \bar{T}$, such that $\forall t > \bar{T}$, $c^t = c^{t-\bar{T}}$, $\bar{T}$ is denoted as the period of the strategy profile.*

Proof of Theorem 2: Use $U(x^1, x^2)$ to denote the defender's normalized expected utility in a round where defender strategy x^2 is used in this round and defender strategy x^1 is used in the previous round. Then $0 \leq U(x^1, x^2) \leq 1$. For the optimal defender strategy profile $[c]$, denote the normalized utility as U^{opt}.

$\langle b^1, ..., b^M \rangle$ is a strategy sequence whose average normalized expected utility for the last $M-1$ rounds, i.e., $U_b = \frac{\sum_{t=2}^{M} U(b^{t-1}, b^t)}{M-1}$, is maximized. $\langle a^1, ..., a^M \rangle$ is a strategy sequence such that the average normalized expected utility of the sequence when it forms a cycle, i.e., $U_a = \frac{U(a^M, a^1) + \sum_{t=2}^{M} U(a^{t-1}, a^t)}{M}$, is maximized. Then

$$\begin{aligned} M * U_a &= U(a^M, a^1) + \sum\nolimits_{t=2}^{M} U(a^{t-1}, a^t) \geq U(b^M, b^1) + \sum\nolimits_{t=2}^{M} U(b^{t-1}, b^t) \\ &\geq \sum\nolimits_{t=2}^{M} U(b^{t-1}, b^t) = (M-1) * U_b \end{aligned}$$

Let $Z = \lceil \frac{T}{M} \rceil$. Construct a cyclic defender strategy profile $[s]$ by repeating the strategy sequence $\langle a^1, ..., a^M \rangle$. Then

$$T * U([s]) = U(c^0, s^1) + \sum\nolimits_{t=2}^{T} U(s^{t-1}, s^t) \tag{13}$$

$$\geq (Z-1) * M * U_a \geq (Z-1) * (M-1) * U_b \tag{14}$$

Strategy profile $[s]$ contains $Z-1$ complete cycles (starting with a^2) with an average normalized utility U_a. The first inequality is derived by ignoring the first round and the last incomplete cycle if any (when $mod(T, M) \neq 1$).

On the other hand, for the optimal defender strategy profile $[c] = [c]^{opt}$, we know that for any consecutive sequence of length M, the average normalized utility of last $M-1$ rounds can be no more than U_b. So we divide the strategy profile into $\lceil \frac{T}{M-1} \rceil$ pieces, each piece with length $M-1$ except the last piece. Then for each piece, the sum of normalized utility is no more than $U_b * (M-1)$. Otherwise, if the sum of normalized utility of the i^{th} piece is higher than $U_b * (M-1)$, then the strategy sequence $< c^{(i-1)*(M-1)}, ..., c^{i*(M-1)} >$ contradicts the optimality of $< b^1, ..., b^M >$. Thus,

$$T * U^{opt} = U(c^0, c^1) + \sum\nolimits_{t=2}^{T} U(c^{t-1}, c^t) \tag{15}$$

$$\leq U_b * (M-1) * \lceil \frac{T}{M-1} \rceil \leq (T + M - 1) * U_b \tag{16}$$

The last inequality is yield by conceptually completing the last piece. Combining 13 - 16, we get

$$\frac{U([s])}{U^{opt}} \geq \frac{(Z-1)*(M-1)}{T+M-1} \geq \frac{(Z-1)*(M-1)}{Z*M+M} = (1-\frac{1}{M})*\frac{Z-1}{Z+1}$$

So $[s]$ is a $(1-\frac{1}{M})\frac{Z-1}{Z+1}$ approximation of the optimal strategy profile in terms of the normalized utility. □

According to Theorem 2, when the game has many rounds ($T \gg M$), the cyclic sequence constructed by repeating $a^1, ...a^M$ is a $1-1/M$ approximation. While in experiments this non-convex MP is solved through local search, with large number of random restarts, we may be able to achieve a near $1-1/M$ approximation.

5 Learning and Planning

In the previous section, we proposed algorithms to find effective defender strategy profiles when the parameters of all the attackers are known. In practice, the defender may not know these parameters at the beginning of the game. In this section, we aim to learn these parameters from attack data and design an effective defender strategy profile.

With abundant attack data, Yang et al. [24] proposed an algorithm to estimate the distribution of the parameters. They treat each data point as an independent attacker, apply maximum likelihood estimation (MLE) to select the most probable parameter vector, and then calculate a normal distribution that fits the learned parameters. The idea of applying MLE to each data point is also used by Haskell et al. [9] to design strategies for protecting fisheries. MLE works well when a large number of data samples are used to estimate one set of parameters [7]. However, some of the assumptions made in previous work in proposing MLE may not always hold true, and therefore estimating the parameter vector from a single data point using MLE can lead to highly biased results; thus sometimes failing to provide reasonable predictions of what happens in reality.

We therefore use the idea of Bayesian Updating instead of MLE and incorporate it with the planning algorithms when the attackers' parameters are unknown. For each data point, we estimate a distribution of parameters using a Bayesian Update instead of selecting the ω vector that maximizes the likelihood of the outcome.

Algorithm 3 describes the learning algorithm for one round of the game when the attackers' parameters can be treated as independent random samples from a prior distribution which is known to the defender. The input of the algorithm includes the number of attacks found on all targets ($\chi = \langle \chi_i \rangle$), the defender strategy that the attackers respond to (η), and the prior distribution $p = \langle p_1, ..., p_S \rangle$ over a discrete set of parameter values $\{\hat{\omega}\} = \langle \hat{\omega}^1, ..., \hat{\omega}^S \rangle$, each of which is 3-element tuple. The parameter vector of an attacker is a random sample from p, so if he attacks target i, we can calculate the posterior distribution of this attacker's parameter by applying Bayes' rule (Line 3). We calculate the average posterior distribution $\bar{p}$ over all attackers (Line 7). $\bar{p}$ can be used to calculate the defender strategy in future rounds using the planning algorithms introduced in the previous section, because calculating the defender's average expected utility over all attackers given each attacker's parameter distribution is equivalent to calculating the defender's expected utility for the average parameter distribution.

Algorithm 3 Learn-BU (η, χ,$\{\hat{\omega}\}$, p)

Output: estimated ω distribution $\bar{p}$.

for i=1 to N **do**
  for s=1 to S **do**
    $\bar{p}_i(s) = \frac{p(s)q_i(\hat{\omega}^s,\eta)}{\sum_r p(r)q_i(\hat{\omega}^r,\eta)}$
  end for
end for
for s=1 to S **do**
  $\bar{p}_i^s = \frac{\sum_i \chi_i \bar{p}_i^s}{\sum_i \chi_i}$
end for

It is straightforward to apply Algorithm 3 to the first round of the LEAD game. When the game has multiple rounds, we use $\bar{p}$ calculated in round t as the prior distribution p in round $t+1$ and apply Algorithm 3 again after collecting attack data in round $t+1$. We assume the attackers' parameters can be treated as random samples from the average posterior distribution of the previous round. This is a simplification of the more rigorous process of using asymptotic analysis for each attacker and it avoids combinatorial enumeration when the attack data is anonymous.

Now we provide a framework that incorporates the learning algorithm with PA-M(-γ) to design the defender profile in a LEAD game. We use p^t to denote the (estimated) distribution of attackers' parameters in the beginning of round t and calculate the defender strategy in round t using Algorithm 1 by substituting Equation 4 with

$$E^t = L \sum\nolimits_s \sum\nolimits_i p^t(s) q_i(\bar{\omega}^s, c^{t-1}) U_i^d(c^t) \tag{17}$$

After attack data in round t is collected (denoted as χ^t), we update the parameter distribution as $p^{t+1} = \text{Learn-BU}(c^t, \chi^t, \{\hat{\omega}\}, p^t)$.

When incorporating the learning algorithm with FS-M, we divide the game into several stages, each containing more than M rounds and only update the parameter distribution at the end of each stage. As FS-M may not achieve a high average expected utility if only a part of the strategy sequence is executed, updating the parameter distribution in every round may lead to a defender strategy profile with low utility.

6 General Case and Extensions

We introduced the planning and learning algorithms in previous sections for simplified cases when $\Gamma = 1$ and $\alpha_0 = 0$. In this section, we generalize our algorithms and theorem to the more general case with $\Gamma > 1$ and $0 \leq \alpha_0 \leq 1$.

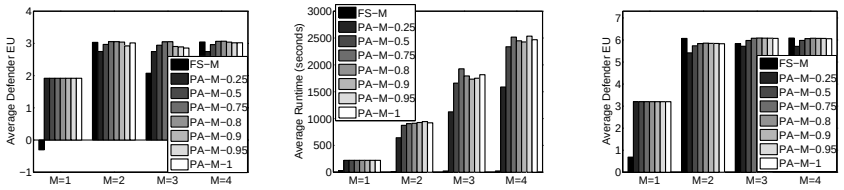
The framework of Algorithms 1 to 3 can be applied to the general case with slight changes in the formulation of η^t. PA-M(-γ) for general case can be calculated by substituting Constraint 5 with $\eta^\tau = \sum_{k=0}^{M} \alpha_k c^{\tau-k}$ and $FS - M$ can be calculated by changing Constraints 10-11 accordingly.

Theorem 3 shows that FS-M can still be a good approximation for games with $\Gamma > 1$ when $\Gamma < M \leq T$ and the proof is similar to that of Theorem 2.

Theorem 3. *In a LEAD game with T rounds, for any fixed length $\Gamma < M \leq T$, there exists a cyclic defender strategy profile $[s]$ with period M that is a $(1 - \frac{\Gamma}{M})\frac{Z-1}{Z+1}$ approximation of the optimal strategy profile in terms of the normalized utility, where $Z = \lceil \frac{T-\Gamma+1}{M} \rceil$.*

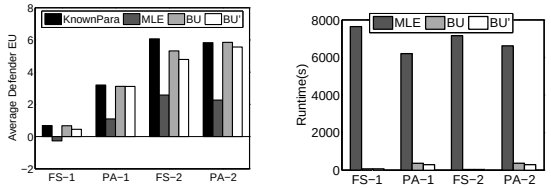
7 Experimental Results

We first test the algorithms on games with $\alpha_0 = 0$ and $\Gamma = 1$ and then provide results for the general case and robustness test at the end of the section. We compare the planning algorithms when the attackers' ω parameters are known. We also test the framework that incorporates planning algorithms and the proposed learning algorithm when the defender is only given a prior distribution of ω. All algorithms are implemented in MATLAB with $fmincon$ function used for local search and tested on 2.4 gigahertz CPUs with 128 GB memory. All key differences noted are statistically significant ($p < 0.05$).

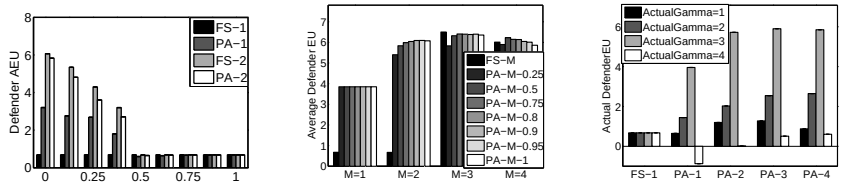


(a) Solution quality with 100 restarts (b) Runtime with 100 restarts (c) Solution quality within 1 hour

μ	$(-17.81, 0.72, 0.47)$
Σ	$\begin{pmatrix} 209.48 & -2.64 & -0.71 \\ -2.64 & 0.42 & 0.24 \\ -0.71 & 0.24 & 0.36 \end{pmatrix}$
μ'	$\|\mu - \mu'\|_\infty \leq 5$
Σ'	$\Sigma'_{ii} - \Sigma_{ii} \leq 5$



(d) Parameter distribution for learning (e) Solution quality for learning (f) Runtime for learning



(g) Solution quality with varying α_0 (h) Solution quality with $\Gamma = 2$. (i) Uncertainty in Γ, estimated $\Gamma = 3$

Fig. 1. Experimental results.

7.1 Known parameters

We first compare the solution quality of proposed planning algorithms given sufficient time for local search. We test 32 game instances of 5 attackers, 3 targets, 1 patroller 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 local search and compare PA-M(-γ) 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 (as in previous work by Yang et. al [24] and Haskell et al. [9]). PA-1 is the myopic greedy strategy for the defender, which tries to optimize the defender expected utility in the current round, without considering future rounds. We assume MAXIMIN strategy is the initial strategy c^0 before the games start.

Figure 1(a) shows that PA-M(-γ) and FS-M with $M > 1$ significantly improve the defender expected utility compared to FS-1 and PA-1 in terms of the average defender expected utility. FS-4 provides equal or slightly better solution quality than FS-2. Interestingly, the solution quality of FS-3 is not as good as FS-2 and FS-4. We give a conceptual explanation through Example 1. If the defender can only choose pure strategies in every round, FS-2 allows the defender to alternate between N_1 and N_2 in every round. However, FS-3 does not allow such a pattern and thus the defender cannot make the most of her advantage in this case. For PA-1, the discount factor does not make any difference as the defender is myopic. When $M > 1$, adding a discount factor may increase the defender's utility for PA-M.

Figure 1(b) shows the runtime for the algorithms. The runtime for FS-M is much lower than that of PA-M when 100 restarts are used for each local search as no matter how many rounds the game has, FS-M only needs to solve one mathematical program while PA-M uses local search to calculate a strategy for each round of the game.

We then compare the solution quality of PA-M(-γ) and FS-M for 32 randomly chosen games with 10 targets, 4 patrollers and 100 attackers (denoted as Game Set 2) in Figure 1(c). We set a 1-hour runtime limit for the algorithms and again, FS-M and PA-M(-γ) with $M > 2$ provide significant improvement in terms of defender's average expected utility compared to FS-1 and PA-1. The improvement from PA-2 to PA-4 is not significant with the best discounting factor given this time limit, indicating that in most cases looking ahead 1 step with a reasonable discount factor can provide a good solution when $\Gamma = 1$.

7.2 Unknown parameters

When the ω parameters of the attackers are unknown to the defender, we compare Algorithm 3 with the baseline learning algorithm that uses MLE (denoted as **MLE**) when incorporated with planning algorithms. We test on Game Set 2 and choose the attackers' parameter vectors ω^l randomly from a three-dimensional normal distribution with mean μ and covariance matrix Σ shown in Figure 1(d). As the learning algorithm is based on attack data, the choice of the attackers may affect the solution quality. We run 16 simulations of attackers' choice for each game and show the average performance.

We set a time limit of 30 minutes for the planning algorithms and show the solution quality and the runtime of the learning algorithms in Figure 1(e) and 1(f). In each figure, the x-axis shows the planning algorithms used and different bars indicate different

learning methods. **BU** represents the case when an accurate prior (a normal distribution with μ and Σ) is given to the defender. **BU**$'$ represents the case when the prior distribution given to the defender is a slightly deviated estimation (a normal distribution with random μ' and Σ' satisfying constraints in Figure 1(d)). We also show the defender's expected utility when the attackers' parameters are known to the defender before the game starts (denote as **KnownPara**).

Figure 1(e) shows that **BU** and **BU**$'$ significantly outperform **MLE** in terms of the defender's average expected utility. Indeed, the solution quality of **BU** and **BU**$'$ is close to that of **KnownPara**, indicating the effectiveness of the proposed learning algorithm. Figure 1(f) shows that learning algorithms based on Bayesian Updating runs much faster than **MLE** which solves a convex optimization problem for each target.

7.3 General case and robustness against uncertainty in α and Γ

We first test algorithms on Game Set 2 with varying α_0 when $\Gamma = 1$ in Figure 1(g). As $\Gamma = 1$, the attackers' decision making in round t depends on the defender strategies in round t and round $t-1$. When $\alpha_0 < 0.5$, i.e., the attackers' understanding of the defender's strategy is closer to c^{t-1}, FS-2 and PA-2 provide significant improvement over FS-1 and PA-1. When $\alpha_0 \geq 0.5$, the attackers' belief is close to the current defender strategy c^t and the improvement from FS-1 to FS-2 is negligible, indicating that there is no need for the strategy change. In the extreme case of $\alpha_0 = 1$, the problem reduces to a repeated Stackelberg game and FS-1 provides the optimal solution.

We then compare proposed algorithms on Game Set 2 with $\Gamma = 2$, $\alpha_1 = \alpha_2 = 0.5$ and $\alpha_0 = 0$ (see Figure 1(h)). As expected, PA-M with $M > 1$ and FS-M with $M > 2$ significantly outperforms FS-1 and PA-1 in terms of the defender's expected utility. The improvement of FS-2 over FS-1 is negligible, as the fixed sequence length is equal to the attackers' memory and thus any sequence can be exploited by the attackers. FS-3 and FS-4 provides significant improvement over FS-1 and interestingly, FS-3 provides better solution quality than FS-4. Combined with the observation that FS-2 outperforms FS-3 when $\Gamma = 1$, this may suggest that $M = \Gamma + 1$ is a reasonable choice for the defender when designing FS-M.

We further provide robustness results against uncertainties in Γ. Figure 1(i) shows the defender's expected utility of PA-M when the defender assumes the attackers' memory length is 3 but the actual Γ varies from 1 to 4. When Γ is slightly over-estimated (actual $\Gamma = 1$ or 2), our proposed algorithms (PA-M with $M > 1$) still significantly outperform the baseline algorithm FS-1 and PA-1. However, when Γ is under-estimated (actual $\Gamma = 4$), results for PA-3 and PA-4 are close to FS-1 as the attackers' memory is longer than the defender's assumption and thus the attackers can exploit the defender's planning. This observation suggests that it is more robust to over-estimate the attackers' memory when the value of Γ is not known to the defender. We defer to future work to learn α_τ and Γ from attack data.

8 Related Work

While related work was discussed throughout the paper, here we discuss additional relevant research. Extensive studies state and model the bounded rationality and bounded

memory of human beings over the years [22,19,6]. Stone et. al [23] studied the optimal strategy to lead a teammate with bounded memory given finite action set. In this paper, we are dealing with players who have conflicting interests and we consider more general cases in which the players can choose from an infinite strategy set. Bounded memory in repeated games has also been studied in some earlier work [20,1,3] and learning against opponents with bounded-memory is considered in some previous work [17,5]. This paper is the first attempt to find a sequence of *mixed* defender strategies to maximize the defender's average expected utility in a repeated game with attackers who have bounded memory and bounded rationality.

Previous work on learning in repeated Stackelberg security games [13,12,4] mainly focus on learning the payoffs of the perfectly rational attackers. Qian et al. [18] model the interaction between protector and extractor in the resource conservation domains as a Partially Observable Markov Decision Process (POMDP) to learn the utility of the targets from the extractor's actions. In our problem, the payoffs are known to both players and the defender aims to design a sequence of *mixed strategies* and learn the parameter vectors of the boundedly rational adversaries. Sequential decision-making in the presence of other players is also studied in the context of computer poker [2,14,8] and most work focuses on zero-sum games with a single action in every round.

9 Conclusion

To conclude, this paper: (i) provided a novel game model, LEAD, for domains involving frequent adversary interaction; (ii) proposed two sets of algorithms PlanAhead-M and FixedSequence-M that design strategies for the defender in a LEAD game; (iii) provided a framework that incorporates a novel learning algorithm and the planning framework; (iv) reported on experimental results that demonstrate the significant improvement of our algorithms over previous best algorithms for FAI domains. This work addresses the limitation of always using a perfect Stackelberg assumption for FAI domains in previous work and provides generalized solutions to deal with scenarios in which there are frequent attacks by attackers with imperfect understanding of the current defender strategy.

References

1. M. Barlo, G. Carmona, and H. Sabourian. Repeated games with one-memory. *Journal of Economic Theory*, 144(1):312 – 336, 2009.
2. D. Billings, N. Burch, A. Davidson, R. Holte, J. Schaeffer, T. Schauenberg, and D. Szafron. Approximating game-theoretic optimal strategies for full-scale poker. In *IJCAI*, pages 661–668, 2003.
3. J. Blocki, N. Christin, A. Datta, and A. Sinha. Adaptive regret minimization in bounded-memory games. *CoRR*, abs/1111.2888, 2011.
4. A. Blum, N. Haghtalab, and A. D. Procaccia. Learning optimal commitment to overcome insecurity. In *NIPS*, 2014.
5. D. Chakraborty, N. Agmon, and P. Stone. Targeted opponent modeling of memory-bounded agents. In *Proceedings of the Adaptive Learning Agents Workshop (ALA)*, 2013.

6. N. Cowan. *Working Memory Capacity*. Essays in cognitive psychology. Psychology Press, 2005.
7. S. Eliason. *Maximum Likelihood Estimation. Logic and Practice.*, volume 96 of *Quantitative Applications in the Social Sciences*. Sage Publications, 1993.
8. A. Gilpin and T. Sandholm. A texas hold'em poker player based on automated abstraction and real-time equilibrium computation. In *AAMAS*, AAMAS '06, pages 1453–1454, New York, NY, USA, 2006. ACM.
9. W. B. Haskell, D. Kar, F. Fang, M. Tambe, S. Cheung, and L. E. Denicola. Robust protection of fsheries with COmPASS. In *IAAI*, 2014.
10. C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordonez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. In *AAMAS*, 2009.
11. D. Korzhyk, V. Conitzer, and R. Parr. Complexity of computing optimal stackelberg strategies in security resource allocation games. In *AAAI*, pages 805–810, 2010.
12. J. Letchford, V. Conitzer, and K. Munagala. Learning and approximating the optimal strategy to commit to. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory*, pages 250–262, 2009.
13. J. Marecki, G. Tesauro, and R. Segal. Playing repeated stackelberg games with unknown opponents. In *AAMAS*, AAMAS '12, pages 821–828, 2012.
14. M. B. Martin Zinkevich, Michael Johanson and C. Piccione. Regret minimization in games with incomplete information. In *NIPS-08*, 2008.
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. J. Pita, M. Jain, C. Western, C. Portway, M. Tambe, F. Ordonez, S. Kraus, and P. Paruchuri. Deployed ARMOR protection: The application of a game theroetic model for security at the los angeles international airport. In *AAMAS*, 2008.
17. R. Powers and Y. Shoham. Learning against opponents with bounded memory. In *IJCAI*, IJCAI'05, pages 817–822, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.
18. Y. Qian, W. B. Haskell, A. X. Jiang, and M. Tambe. Online planning for optimal protector strategies in resource conservation games. In *AAMAS*, 2014.
19. A. Rubinstein. *Modeling Bounded Rationality*, volume 1 of *MIT Press Books*. The MIT Press, December 1997.
20. H. Sabourian. Repeated games with m-period bounded memory (pure strategies). *Journal of Mathematical Economics*, 30(1):1 – 35, 1998.
21. 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.
22. H. Simon. A behavioural model of rational choice. In H. Simon, editor, *Models of man: social and rational; mathematical essays on rational human behavior in a social setting*, pages 241–260. J. Wiley, New York, 1957.
23. P. Stone, G. A. Kaminka, S. Kraus, J. R. Rosenschein, and N. Agmon. Teaching and leading an ad hoc teammate: Collaboration without pre-coordination. *Artificial Intelligence*, 2013.
24. R. Yang, B. Ford, M. Tambe, and A. Lemieux. Adaptive resource allocation for wildlife protection against illegal poachers. In *AAMAS*, 2014.
25. Z. Yin, A. Jiang, M. Johnson, M. Tambe, C. Kiekintveld, K. Leyton-Brown, T. Sandholm, and J. Sullivan. TRUSTS: Scheduling randomized patrols for fare inspection in transit systems. In *IAAI*, 2012.
26. Z. Yin, D. Korzhyk, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. nash in security games: Interchangeability, equivalence, and uniqueness. *In AAMAS*, 2010.