Trends and Applications in Stackelberg Security Games

Debarun Kar[1], Thanh H. Nguyen[1], Fei Fang[1], Matthew Brown[1], Arunesh Sinha[1], Milind Tambe[1], and Albert Xin Jiang[2]

[1] University of Southern California, Los Angeles, USA
[dkar,thanhhng,feifang,mattheab,aruneshs,tambe]@usc.edu
[2] Trinity University, San Antonio, TX, USA
xjiang@trinity.edu

Abstract. Security is a critical concern around the world, whether it is the challenge of protecting ports, airports and other critical infrastructure, interdicting the illegal flow of drugs, weapons and money, protecting endangered wildlife, forests and fisheries, suppressing urban crime or security in cyberspace. Unfortunately, limited security resources prevent full security coverage at all times; instead, we must optimize the use of limited security resources. To that end, a new "security games" framework was developed, which led to building of decision-aids for security agencies around the world. Security games is a novel area of research that is based on computational and behavioral game theory, while also incorporating elements of AI planning under uncertainty and machine learning. Today security-games based decision aids for infrastructure security are deployed in the US and internationally; examples include deployments at ports and ferry traffic with the US coast guard, for security of air traffic with the US Federal Air Marshals, and for security of university campuses, airports and metro trains with police agencies in the US and other countries. Moreover, recent work on "green security games" has led decision aids to be deployed, assisting NGOs in protection of wildlife; and "opportunistic crime security games" have focused on suppressing urban crime. In the cyber-security domain, the interaction between the defender and adversary is quite complicated with a high degree of incomplete information and uncertainty. Recently, applications of game theory to provide quantitative and analytical tools to network administrators through defensive algorithm development and adversary behavior prediction to protect cyber infrastructures has also received significant attention. This chapter provides an overview of use-inspired research in security games including algorithms for scaling up security games to real-world sized problems, handling multiple types of uncertainty, and dealing with bounded rationality and bounded surveillance of human adversaries.

Keywords: Security Games, Scalability, Uncertainty, Bounded Rationality, Bounded Surveillance, Adaptive Adversary, Infrastructure Security, Wildlife Protection.

1 Introduction

Security is a critical concern around the world that manifests in problems such as protecting our ports, airports, public transportation, and other critical national infrastructure from terrorists, in protecting our wildlife and forests from poachers and smugglers, and in curtailing the illegal flow of weapons, drugs, and money across international

borders. In all of these problems, we have limited security resources which prevents security coverage on all the targets at all times; instead, security resources must be deployed intelligently taking into account differences in the importance of targets, the responses of the attackers to the security posture, and potential uncertainty over the types, capabilities, knowledge and priorities of attackers faced.

To address these challenges in adversarial reasoning and security resource allocation, a new "security games" framework has been developed (Tambe (2011)); this framework has led to building of decision-aids for security agencies around the world. Security games are based on computational and behavioral game theory, while also incorporating elements of AI planning under uncertainty and machine learning. Security games algorithms have led to successes and advances over previous human-designed approaches in security scheduling and allocation by addressing the key weakness of predictability in human-designed schedules. These algorithms are now deployed in multiple applications. The first application was ARMOR, which was deployed at the Los Angeles International Airport (LAX) in 2007 to randomize checkpoints on the roadways entering the airport and canine patrol routes within the airport terminals (Jain et al (2010b)). Following that, came several other applications: IRIS, a game-theoretic scheduler for randomized deployment of the US Federal Air Marshals (FAMS), has been in use since 2009 (Jain et al (2010b)); PROTECT, which schedules the US Coast Guard's randomized patrolling of ports, has been deployed in the port of Boston since April 2011 and is in use at the port of New York since February 2012 (Shieh et al (2012)), and has spread to other ports such as Los Angeles/Long Beach, Houston, and others; another application for deploying escort boats to protect ferries has been deployed by the US Coast Guard since April 2013 (Fang et al (2013)); and TRUSTS (Yin et al (2012)) which has been evaluated in field trials by the Los Angeles Sheriffs Department (LASD) in the LA Metro system. Most recently, PAWS— another game-theoretic application was tested by rangers in Uganda for protecting wildlife in Queen Elizabeth National Park in April 2014 (Yang et al (2014)); MIDAS was tested by the US Coast Guard for protecting fisheries (Haskell et al (2014)). These initial successes point the way to major future applications in a wide range of security domains.

Researchers have recently started to explore the use of such security game models in tackling security issues in the cyber world. In (Vanek et al (2012)), the authors study the problem of optimal resource allocation for packet selection and inspection to detect potential threats in large computer networks with multiple computers of differing importance. In their paper, they study the application of security games to deep packet inspection as countermeasure to intrusion detection. In a recent paper (Durkota et al (2015)), the authors study the problem of optimal number of *honeypots* to be placed in a network using a security game framework. Another interesting work, called audit games (Blocki et al (2013, 2015)), enhances the security games model with choice of punishments in order to capture scenarios of security and privacy policy enforcement in large organizations (Blocki et al (2013, 2015)).

Given the many game-theoretic applications for solving real-world security problems, this chapter provides an overview of the models and algorithms, key research challenges and a description of our successful deployments. Overall, the work in security games has produced numerous decision aids that are in daily use by security agen-

cies to optimize their limited security resources. The implementation of these applications required addressing fundamental research challenges. We categorize the research challenges associated with security games into four broad categories: (1) addressing scalability across a number of dimensions of the game, (2) tackling different forms of uncertainty that be present in the game, (3) addressing human adversaries' bounded rationality and bounded surveillance (limited capabilities in surveillance), and (4) evaluation of the framework in the field. Given the success in providing solutions for many security domains involving the protection of critical infrastructure, the topic of security games has evolved and expanded to include new types of security domains, for example, for wildlife and environmental protection.

The rest of the chapter is organized as follows: Section 2 introduces the general security games model, Section 3 discusses three different types of security games, Section 4 describes the approaches used to tackle scalability issues, Section 5 describes the approaches to deal with uncertainty, Section 6 focuses on bounded rationality and bounded surveillance, and Section 7 provides details of field evaluation of the science of security games.

2 Stackelberg Security Games

Stackelberg games were first introduced to model leadership and commitment (von Stackelberg (1934)). A Stackelberg game is a game played sequentially between two players: the first player is the leader who commits to a strategy first, and then the second player, called the follower, observes the strategy of the leader and then commits to his own strategy. The term Stackelberg Security Games (SSG) was first introduced by Kiekintveld et al (2009) to describe specializations of a particular type of Stackelberg game for security as discussed below. This section provides details on this use of Stackelberg games for modeling security domains. We first give a generic description of security domains followed by *security games*, the model by which security domains are formulated in the Stackelberg game framework[3].

2.1 Stackelberg Security Game

In Stackelberg Security Games, a defender must perpetually defend a set of targets T using a limited number of resources, whereas the attacker is able to surveil and learn the defender's strategy and attack after careful planning. An action, or *pure strategy*, for the defender represents deploying a set of resources R on patrols or checkpoints, e.g., scheduling checkpoints at the LAX airport or assigning federal air marshals to protect flight tours. The pure strategy for an attacker represents an attack at a target, e.g., a flight. The *mixed strategy* of the defender is a probability distribution over the pure strategies. Additionally, with each target are also associated a set of payoff values that define the utilities for both the defender and the attacker in case of a successful or a failed attack.

[3] Note that *not* all security games in the literature are Stackelberg security games (see Alpcan and Başar (2010))

A key assumption of Stackelberg Security Games (we will sometimes refer to them as simply security games) is that the payoff of an outcome depends only on the target attacked, and whether or not it is *covered* (protected) by the defender (Kiekintveld et al (2009)). The payoffs do *not* depend on the remaining aspects of the defender allocation. For example, if an adversary succeeds in attacking target t_1, the penalty for the defender is the same whether the defender was guarding target t_2 or not.

	Defender		Attacker	
Target	Covered	Uncovered	Covered	Uncovered
t_1	10	0	-1	1
t_2	0	-10	-1	1

Table 1: Example of a security game with two targets.

This allows us to compactly represent the payoffs of a security game. Specifically, a set of four payoffs is associated with each target. These four payoffs are the rewards and penalties to both the defender and the attacker in case of a successful or an unsuccessful attack, and are sufficient to define the utilities for both players for all possible outcomes in the security domain. More formally, if target t is attacked, the defender's utility is $U_d^c(t)$ if t is covered, or $U_d^u(t)$ if t is not covered. The attacker's utility is $U_a^c(t)$ if t is covered, or $U_a^u(t)$ if t is not covered. Table 1 shows an example security game with two targets, t_1 and t_2. In this example game, if the defender was covering target t_1 and the attacker attacked t_1, the defender would get 10 units of reward whereas the attacker would receive -1 units. We make the assumption that in a security game it is always better for the defender to cover a target as compared to leaving it uncovered, whereas it is always better for the attacker to attack an uncovered target. This assumption is consistent with the payoff trends in the real-world. A special case is *zero-sum games*, in which for each outcome the sum of utilities for the defender and attacker is zero, although general security games are not necessarily zero-sum.

2.2 Solution Concept: Strong Stackelberg Equilibrium

The solution to a security game is a *mixed* strategy[4] for the defender that maximizes the expected utility of the defender, given that the attacker learns the mixed strategy of the defender and chooses a best-response for himself. The defender's mixed strategy is a probability distribution over all pure strategies, where a pure strategy is an assignment of the defender's limited security resources to targets. This solution concept is known as a Stackelberg equilibrium (Leitmann (1978)).

The most commonly adopted version of this concept in related literature is called Strong Stackelberg Equilibrium (SSE) (Breton et al (1988); Conitzer and Sandholm (2006); Paruchuri et al (2008); von Stengel and Zamir (2004)). In security games, the mixed strategy of the defender is equivalent to the probabilities that each target t is

[4] Note that mixed strategy solutions apply beyond Stackelberg games

covered by the defender, denoted by $C = \{c_t\}$ (Korzhyk et al (2010)). Furthermore, it is enough to consider a pure strategy of the rational adversary (Conitzer and Sandholm (2006)), which is to attack a target t. The expected utility for defender for a strategy profile (C, t) is defined as $U_d(t, C) = c_t U_d^c(t) + (1 - c_t)U_d^u(t)$, and a similar form for the adversary. A SSE for the basic security games (non-Bayesian, rational adversary) is defined as follows:

Definition 1. *A pair of strategies (C^*, t^*) form a* Strong Stackelberg Equilibrium *(SSE) if they satisfy the following:*

1. *The defender plays a best-response: $U_d(t^*, C^*) \geq U_d(t(C), C)$ for all defender's strategy C where $t(C)$ is the attacker's response against the defender strategy C.*
2. *The attacker plays a best-response: $U_a(t^*, C^*) \geq U_a(t, C^*)$ for all target t.*
3. *The attacker breaks ties in favor of the defender: $U_d(t^*, C^*) \geq U_d(t', C^*)$ for all target t' such that $t' = \text{argmax}_t\, U_a(t, C^*)$*

The assumption that the follower will always break ties in favor of the leader in cases of indifference is reasonable because in most cases the leader can induce the favorable strong equilibrium by selecting a strategy arbitrarily close to the equilibrium that causes the follower to strictly prefer the desired strategy (von Stengel and Zamir (2004)). Furthermore an SSE exists in all Stackelberg games, which makes it an attractive solution concept compared to versions of Stackelberg equilibrium with other tie-breaking rules. Finally, although initial applications relied on the SSE solution concept, we have since proposed new solution concepts that are more robust against various uncertainties in the model (Yin et al (2011); An et al (2011); Pita et al (2012)) and have used these robust solution concepts in some of the later applications.

For simple examples of security games, such as the one shown above, the Strong Stackelberg Equilibrium can be calculated by hand. However, as the size of the game increases, hand calculation is no longer feasible and an algorithmic approach for generating the SSE becomes necessary. Conitzer and Sandholm (Conitzer and Sandholm (2006)) provided the first complexity results and algorithms for computing optimal commitment strategies in Stackelberg games, including both pure and mixed-strategy commitments. An improved algorithm for solving Stackelberg games, DOBSS (Paruchuri et al (2008)), is central to the fielded application ARMOR that was in use at the Los Angeles International Airport (Jain et al (2010b)).

Decomposed Optimal Bayesian Stackelberg Solver (DOBSS): We now describe the DOBSS[5] algorithm in detail as it provides a starting point for the algorithms we develop in the next section. We first present DOBSS in its most intuitive form as a Mixed-Integer Quadratic Program (MIQP); we then present a linearized equivalent Mixed-Integer Linear Program (MILP). The DOBSS model explicitly represents the actions by the leader and the *optimal* actions for the follower in the problem solved by the leader. Note that we need to consider only the reward-maximizing pure strategies of the follower, since for a given fixed mixed strategy x of the leader, each follower faces a problem with

[5] DOBSS addresses Bayesian Stackelberg games with multiple follower types, but for simplicity we do not introduce Bayesian Stackelberg games here.

fixed linear rewards. If a mixed strategy is optimal for the follower, then so are all the pure strategies in support of that mixed strategy.

Thus, we denote by x the leader's policy, which consists of a probability distribution over the leader's pure strategies $\sigma_i \in \Sigma_\Theta$, wher Σ_Θ is the set of all pure strategies of the leader. Hence, the value x_i is the proportion of times in which pure strategy $\sigma_i \in \Sigma_\Theta$ is used in the policy. Similarly, q_j is the probability of taking strategy $\sigma_j \in \Sigma_\Psi$ for the follower, where Σ_Ψ is the set of all pure strategies for the follower. We denote by X and Q the index sets of the leader and follower pure strategies, respectively. We also index the payoff matrices of the leader and the follower by the matrices R and C where R_{ij} and C_{ij} are the rewards obtained if the leader takes strategy $\sigma_i \in \Sigma_\Theta$ and the follower takes strategy $\sigma_j \in \Sigma_\Psi$. Let M be a large positive number; constraint 3 in the MIQP below requires that the variable a be set to the maximum reward a follower can obtain given the current policy x taken by the leader. The leader then solves the following:

$$\max_{x,q,a} \quad \sum_{i\in X}\sum_{j\in Q} R_{ij}x_i q_j \tag{1}$$

$$\text{s.t.} \quad \textstyle\sum_{i\in X} x_i = 1 \tag{2}$$

$$\textstyle\sum_{j\in Q} q_j = 1 \tag{3}$$

$$0 \le (a - \textstyle\sum_{i\in X} C_{ij}x_i) \le (1-q_j)M \;\; \forall j \in Q \tag{4}$$

$$x_i \in [0\dots 1] \qquad \forall i \in X \tag{5}$$

$$q_j \in \{0,1\} \qquad \forall j \in Q \tag{6}$$

$$a \in \Re \tag{7}$$

Here, for a leader strategy x and a strategy q for the follower, the objective (Line 1) represents the expected reward for the leader. The first (Line 2) and the fourth (Line 5) constraints define the set of feasible solutions $x \in X$ as a probability distribution over the set of strategies $\sigma_i \in \Sigma_\Theta$. The second (Line 3) and third (Line 6) constraints limit the vector of strategies, q, to be a pure strategy over the set Q (that is each q has exactly one coordinate equal to one and the rest equal to zero). The two inequalities in the third constraint (Line 4) ensure that $q_j = 1$ only for a strategy j that is optimal for the follower. Indeed this is a linearized form of the optimality conditions for the linear programming problem solved by each follower. We explain the third constraint (Line 4) as follows: this constraint enforces dual feasibility of the follower's problem (leftmost inequality) and the complementary slackness constraint for an optimal pure strategy q for the follower (rightmost inequality). Note that the leftmost inequality ensures that $\forall j \in Q$, $a \ge \sum_{i\in X} C_{ij}x_i$. This means that given the leader's policy x, a is an upper bound on follower's reward for any strategy. The rightmost inequality is inactive for every strategy where $q_j = 0$, since M is a large positive quantity. In fact, since only one pure strategy can be selected by the follower, say some $q_j = 1$, for the strategy that has $q_j = 1$ the right inequality states $a \le \sum_{i\in X} C_{ij}x_i$, which combined with the left inequality enforces $a = \sum_{i\in X} C_{ij}x_i$, thereby imposing no additional constraint for all other pure strategies which have $q_j = 0$ and showing that this strategy must be optimal for the follower.

We can linearize the quadratic programming problem (Lines 1 to 7) through the change of variables $z_{ij} = x_i q_j$ to obtain the following mixed integer linear programming problem as shown in Paruchuri et al (2008).

$$\max_{q,z,a} \quad \sum_{i \in X} \sum_{j \in Q} pR_{ij} z_{ij} \tag{8}$$

$$\text{s.t.} \quad \sum_{i \in X} \sum_{j \in Q} z_{ij} = 1 \tag{9}$$

$$\sum_{j \in Q} z_{ij} \leq 1 \qquad \forall i \in X \tag{10}$$

$$q_j \leq \sum_{i \in X} z_{ij} \leq 1 \qquad \forall j \in Q \tag{11}$$

$$\sum_{j \in Q} q_j = 1 \tag{12}$$

$$0 \leq (a - \sum_{i \in X} C_{ij} (\sum_{h \in Q} z_{ih})) \leq (1 - q_j) M \quad \forall j \in Q \tag{13}$$

$$\sum_{j \in Q} z_{ij} = \sum_{j \in Q} z^1_{ij} \qquad \forall i \in X \tag{14}$$

$$z_{ij} \in [0 \ldots 1] \qquad \forall i \in X, j \in Q \tag{15}$$

$$q_j \in \{0, 1\} \qquad \forall j \in Q \tag{16}$$

$$a \in \Re \tag{17}$$

DOBSS solves this resulting mixed integer linear program using efficient integer programming packages. The MILP was shown to be equivalent to the MIQP (Lines 1 to 7) and the equivalent Harsanyi transformed Stackelberg game (Paruchuri et al (2008)). For a more in depth explanation of DOBSS please see Paruchuri et al (2008).

3 Categorizing Security Games

With progress in the security games research, and the expanding set of applications, it is valuable to consider categorizing this work into three separate areas. These categories are driven by applications, but they also impact the types of games (e.g., single shot vs repeated games) considered, and the research issues that arise. Specifically, the three categories are: (i) Infrastructure Security Games; (ii) Green Security Games; (iii) Opportunistic Crime Security Games. We discuss each category below.

3.1 Infrastructure Security Games

These types of games and their applications is where the original research on security games was initiated. Key characteristics of these games include the following:

– *Application characteristics:* These games are focused on applications of protecting infrastructure, such as ports, airports, trains, flights and so on; the goal is often assisting agencies engaged in counter-terrorism. Notice that the infrastructure being protected tends to be static, and little changes in a few months, e.g., an airport being protected may have new construction once in 2-3 years. The activities in the infrastructure are regulated by well established schedules of movement of people or goods. Furthermore, the targets being protected often have a discrete structure,

e.g., terminals at an airport, individual flights, individual trains, etc.

– *Overall characteristics of the defender and adversary play:* These games are single shot games. The defender does play her strategy repeatedly, i.e., the defender commits to a mixed strategy in this security game. This mixed strategy may get played for months at a time. However, a single attack by an adversary ends the game. The game could potentially restart after such an attack, but it is not setup as a repeated game as in the game categories described below.

– *Adversary characteristics:* The games assume that the adversaries are highly strategic, who may attack after careful planning and surveillance. These carefully planned attacks have high consequences. Furthermore since these attacks are a result of careful planning with the anticipation of high consequences, attackers commit to these plans of attacks and are not considered to opportunistically move from target to target.

– *Defender characteristics:* The defender does not repeatedly update her strategies. In these domains, there may be just a few attacks that may occur, but these tend to be rare; they are not a very large number of attacks that occur repeatedly. As a result, traditionally, no machine learning is used in this work for the defender to update her strategies over time.

3.2 Green Security Games

These types of games and their applications are focused on trying to protect the environment; and we adopt the term from "green criminology".[6]

– *Application characteristics:* These games are focused on applications of protecting the environment, including forests, fish and wildlife. The goal is thus often to assist security agencies against poachers, illegal fishermen or those illegally cutting trees in national parks in countries around the world. Unlike infrastructure security games, animals or fish being protected may move around in geographical space, introducing new dimensions of complexity. Finally, the targets being protected are spread out over vast open geographical spaces, e.g., large forest regions protect trees from illegal cutting.

– *Overall characteristics of the defender and adversary play:* These games are **not** single shot games. Unfortunately, the adversaries often conducts multiple repeated "attacks", e.g., poaching animals repeated. Thus, a single illegal activity does not

[6] We use the term green security games also to avoid any confusion that may come about given that terms related to the environment and security have been adopted for other uses. For example, the term "environmental security" broadly speaking refers to threats posed to humans due to environmental issues, e.g., climate change or shortage of food. The term "environmental criminology" on the other hand refers to analysis and understanding of how different environments affect crime.

end the game. Instead, usually, after obtaining reports, e.g., over a month, of illegal activities, the defender often replans her security activities. In other words, these are repeated security games where the defender plays a mixed strategy, while the attacker attacks multiple times, and then the defender replans and plays a new mixed strategy and the cycle repeats. Notice also that the illegal activities of concern here may be conducted by multiple individuals, and thus there are multiple adversaries that are active at any one point.

- *Adversary characteristics:* As mentioned earlier, the adversaries are engaged in repeated illegal activities; and the consequences of failure or success are not as severe as in the case of counter-terrorism. As a result, every single attack (illegal action) cannot be carried out with the most detailed surveillance and planning; the adversaries will hence exhibit more of a bounded rationality and bounded surveillance in these domains.
  Nonetheless, these domains are not ones where illegal activities can be conducted opportunistically (as in the opportunistic crime security games discussed below). This is because in these green security games, the adversaries often have to act in extremely dangerous places (e.g., deep in forests, protecting themselves from wild animals), and thus given the risks involved, they cannot take an entirely opportunistic approach.

- *Defender characteristics:* Since this is a repeated game setting, the defender repeatedly updates her strategies. Machine learning can now be used in this work for the defender to update her strategies over time, given that attack data is available over time. The presence of large amounts of such attack data is very unfortunate in that very large numbers of crimes against the environment are recorded in real life, but the silver lining is that the defender can improve her strategy exploiting this data.

3.3 Opportunistic Crime Security Games

These types of games and their applications are focused on trying to combat opportunistic crime. Such opportunistic crime may include criminals engaged in thefts such as snatching of cell phones in metros or stealing student laptops from libraries.

- *Application characteristics:* These games focused on applications involving protecting the public against opportunistic crime. The goal is thus often to assist security agencies in protecting public's property such as cell phones, laptops or other valuables. Here, human crowds may move around based on scheduled activities, e.g., office hours in downtown settings, or class timings on a university campus, and thus the focus of what needs to be protected may shift on a regular schedule. At least in urban settings, these games focus on specific limited geographical areas as opposed to vast open spaces as involved in "green security games".

- *Overall characteristics of the defender and adversary play:* While these games are not explicitly formulated as repeated games, the adversary may conduct or attempt

to conduct multiple "attacks" (thefts) in any one round of the game. Thus, the defender commits to a mixed strategy, but a single attack by a single attacker does not end the game. Instead multiple attackers may be active at a time, conducting multiple thefts while the defender attempts to stop these thefts from taking place.

- *Adversary characteristics:* Once again, the adversaries are engaged in repeated illegal activities; and the consequences of failure or success are not as severe as in the case of counter-terrorism. As a result, once again, given that every single attack (illegal action) cannot be carried out with the most detailed surveillance and planning, the adversaries may thus act even less strategically, and exhibit more of a bounded rationality and bounded surveillance in these domains. Furthermore, the adversaries are not as committed to detailed plans and are flexible in their execution of their plans, as targets of opportunity present themselves.

- *Defender characteristics:* How to update defender strategies in these games from crime data is still an open research challenge.

3.4 Cyber Security Games

These types of games and their applications are focused on trying to combat cyber crimes. Such crimes include attackers compromising network infrastructures to launch a physical attack or stealing digital data, etc.

- *Application characteristics:* These games are focused on applications involving protecting network assets against cyber attacks. The goal is thus often to assist network administrators in protecting computer systems such as data servers, switches, etc, from data theft or damage to hardware, software or information, as well as preventing disruption of services.

- *Overall characteristics of the defender and adversary play:* Depending on the problem at hand, the attacker (or the intruder) may want to gain control over (or to disable) a valuable computer in the network by scanning the network, compromising a more vulnerable system, and/or gaining access to further devices on the computer network. The ultimate goal could be to use the compromised systems to launch further attacks or to steal data, etc. The broader goal of the defender (a human network administrator, or a detection system) could be formulated as preventing the adversary from gaining control over systems in the network by detecting malicious attacks.

- *Adversary characteristics:* The adversary's characteristics vary from one application domain to another. In some application scenarios, the intruder may simply want to gain control over (or to disable) a valuable computer in the network to launch other attacks, by scanning the network and thus compromising a more vulnerable system, and/or gaining access to further devices on the computer network. The actions of the attacker can therefore be seen as sending malicious packets from a

controlled computer (termed source) to a single or multiple vulnerable computers (termed targets). In other scenarios, the attacker may be interested in stealing valuable information from a particular data server and therefore takes necessary actions to compromise the desired system, possibly through a series of disruptions as studied in the Advanced Persistent Threat (APT) literature.

– *Defender characteristics:* Although this is a new and open problem, there has been recent literature that studies the problem of optimal defender resource allocation for packet selection and inspection to detect potential threats in large computer networks with multiple computers of differing importance. Therefore, the objective of the defender in such problems is to prevent the intruder from succeeding by selecting the packets for inspection, identifying the attacker, and subsequently thwarting the attack.

Even though we have categorized the research and applications of security games in these three categories, not everything is very cleanly divided in this fashion. Further research may reveal other categories of need to generate sub-categories of the above three categories.

In the rest of this chapter, we will concentrate only on Infrastructure Security Games and Green Security Games. In the following sections, we first present three key challenges in solving real-world security problems which are summarized in Figure 1: 1) scaling up to real-world sized security problems, 2) handling multiple uncertainties in security games, and 3) dealing with bounded rationality and bounded surveillance of human adversaries. While Figure 1 does not provide an exhaustive overview of all research in SSG, it provides a general overview of the areas of research, and a roadmap to the rest of the book chapter. In each case, we will use a domain example to motivate the specific challenge and then outline the key algorithmic innovation needed to address the challenge.

4 Addressing Scalability in Real-world Problems

The early works in Stackelberg security games such as DOBSS (Paruchuri et al (2008)) required that the full set of pure strategies for both players be considered when modeling and solving Stackelberg security games. However, many real world problems feature billions of pure strategies for either the defender and/or the attacker. Such large problem instances cannot even be represented in modern computers, let alone solved using previous techniques.

In addition to large strategy spaces, there are other scalability challenges presented by different real world security domains. There are domains where, rather than being static, the targets are moving and thus the security resources need to be mobile and move in a continuous space to provide protection. There are also domains where the attacker may not conduct the careful surveillance and planning that is assumed for a Strong Stackelberg Equilibrium and thus it is important to model the bounded rationality and bounded surveillance of the attacker in order to predict their behavior. In

	Challenge	Domain Example	Algorithmic Solution
Scalability	Large defender strategy space	Federal Air Marshals Service	ASPEN: strategy generation approach
	Large defender & attacker strategy spaces	Road Network Security	RUGGED: double oracle approach
	Mobile resources & moving targets	Ferry Protection	CASS: compact representation of strategy
	Multiple boundedly rational attackers	Fishery Protection	MIDAS: cutting plane approach
	Incorporating fine-grained spatial information	Green Security Domains: wildlife/fishery protection	Hierarchical modeling approach
Uncertainty	Unifications of uncertainties	Security in LAX Airport	URAC: multi-dimensional reduction & divide-and-conquer approach
	Dynamic execution uncertainty	Security in Transit System	Markov Decision Processes approach
Attacker Bounded Rationality and Bounded Surveillance	Learning attacker behaviors	Green Security Domains: wildlife/fishery protection	Behavioral models & Human subject experiments

Fig. 1: Summary of Real-world Security Challenges

the former case, both the defender and attacker's strategy spaces are infinite. In the latter case, computing the optimal strategy for the defender given attacker behavioral (bounded rationality and/or bounded surveillance) model is computationally expensive. Furthermore, in certain domains, it is important to incorporate fine-grained topographical information to generate realistic patrol strategies for the defender. However, in doing so, existing techniques lead to a significant challenge in scalability especially when scheduling constraints need to be satisfied. In this section, we thus highlight the critical scalability challenges faced to bring Stackelberg security games to the real world and the research contributions that served to address these challenges.

4.1 Scale Up with Large Defender Strategy Spaces

This section provides an example of a research challenge in security games where the number of defender strategies is too enormous to be enumerated in computer memory.

In this section as in others that will follow, we will first provide a domain example motivating the challenge and then the algorithmic solution for the challenge.

Domain Example – IRIS for US Federal Air Marshals Service. The US Federal Air Marshals Service (FAMS) allocates air marshals to flights departing from and arriving in the United States to dissuade potential aggressors and prevent an attack should one occur. Flights are of different importance based on a variety of factors such as the numbers of passengers, the population of source and destination cities, and international flights from different countries. Security resource allocation in this domain is significantly more challenging than for ARMOR: a limited number of air marshals need to be scheduled to cover thousands of commercial flights each day. Furthermore, these air marshals must be scheduled on tours of flights that obey various constraints (e.g., the time required to board, fly, and disembark). Simply finding schedules for the marshals that meet all of these constraints is a computational challenge. For an example scenario with 1000 flights and 20 marshals, there are over 10^{41} possible schedules that could be considered. Yet there are currently tens of thousands of commercial flights flying each day, and public estimates state that there are thousands of air marshals that are scheduled daily by the FAMS (Keteyian (2010)). Air marshals must be scheduled on tours of flights that obey logistical constraints (e.g., the time required to board, fly, and disembark). An example of a schedule is an air marshal assigned to a round trip from New York to London and back.

Against this background, the IRIS system (Intelligent Randomization In Scheduling) has been developed and deployed by FAMS since 2009 to randomize schedules of air marshals on international flights. In IRIS, the targets are the set of n flights and the attacker could potentially choose to attack one of these flights. The FAMS can assign $m < n$ air marshals that may be assigned to protect these flights.

Since the number of possible schedules exponentially increases with the number of flights and resources, DOBSS is no longer applicable to the FAMS domain. Instead, IRIS uses the much faster ASPEN algorithm (Jain et al (2010a)) to generate the schedule for thousands of commercial flights per day.

Algorithmic Solution – Incremental Strategy Generation (ASPEN). In this section, we describe one particular algorithm ASPEN, that computes strong Stackelberg equilibria (SSE) in domains with a *very large* number of pure strategies (up to billions of actions) for the defender (Jain et al (2010a)). ASPEN builds on the insight that in many real-world security problems, there exist solutions with *small support sizes*, which are mixed strategies in which only a small set of pure strategies are played with positive probability (Lipton et al (2003)). ASPEN exploits this by using a *incremental strategy generation* approach for the defender, in which defender pure strategies are iteratively generated and added to the optimization formulation.

In ASPEN's security game, the attacker can choose any of the flights to attack, and each air marshal can cover one schedule. Each schedule here is a feasible set of targets that can be covered together; for the FAMS, each schedule would represent a flight tour which satisfies all the logistical constraints that an air marshal could fly. For example, $\{t_1, t_2\}$ would be a flight schedule, where t_1 is an outbound flight and t_2 is an inbound

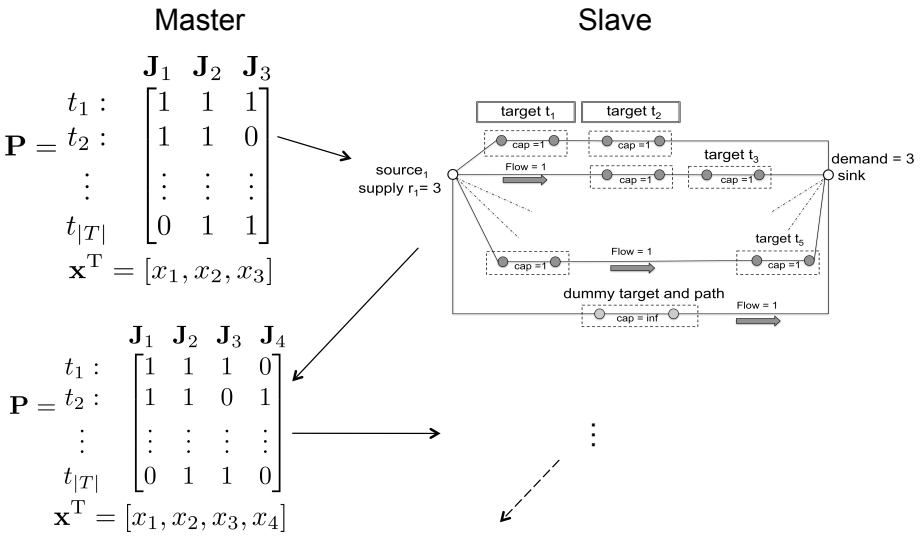


Fig. 2: Strategy generation employed in ASPEN: The schedules for a defender are generated iteratively. The *slave* problem is a novel minimum-cost integer flow formulation that computes the new pure strategy to be added to $\mathbf{P}$; $\mathbf{J}_4$ is computed and added in this example.

flight for one air marshal. A *joint schedule* then would assign every air marshal to a flight tour, and there could be exponentially many joint schedules in the domain. A pure strategy for the defender in this security game is a joint schedule. Thus for example, if there are two air marshals, one possible joint schedule would be $\{\{t_1, t_2\}, \{t_3, t_4\}\}$, where the first air marshal covers flights t_1 and t_2, and the second covers flights t_3 and t_4. As mentioned previously, ASPEN employs incremental strategy generation since all the defender pure strategies cannot be enumerated for such a massive problem. ASPEN decomposes the problem into a *master* problem and a *slave* problem, which are then solved iteratively. Given a number of pure strategies, the master solves for the defender and the attacker optimization constraints, while the slave is used to generate a new pure strategy for the defender in every iteration. *This incremental, iterative strategy generation process allows ASPEN to avoid generation of the entire set of pure strategies*. In other words, by exploiting the small support size mentioned above, only a few pure strategies get generated via the iterative process; and yet we are guaranteed to reach the optimal solution.

The iterative process is graphically depicted in Figure 2. The master operates on the pure strategies (joint schedules) generated thus far, which are represented using the matrix $\mathbf{P}$. Each column of $\mathbf{P}$, $\mathbf{J}_j$, is one pure strategy (or joint schedule). An entry P_{ij} in the matrix $\mathbf{P}$ is 1 if a target t_i is covered by joint-schedule $\mathbf{J}_j$, and 0 otherwise. For example, in Figure 2, the joint schedule J_3 covers target t_1 but not target t_2. The objective of the master problem is to compute $\mathbf{x}$, the optimal mixed strategy of the defender over the pure strategies in $\mathbf{P}$. The objective function for the slave is updated based on

the solution of the master, and the slave is solved to identify the best new column to add to the master problem, using reduced costs (explained later). If no column can improve the solution, the algorithm terminates. Therefore, in terms of our example, the objective of the slave problem is to generate the best joint schedule to add to $\mathbf{P}$. This best joint schedule is identified using the concept of *reduced costs*, which captures the total change in the defender payoff if a candidate column is added to $\mathbf{P}$, i.e., it measures if a pure strategy can potentially increase the defender's expected utility. The candidate column with minimum reduced cost improves the objective value the most. The details of the approach are provided in (Jain et al (2010a)). While a naïve approach would be to iterate over all possible pure strategies to identify the pure strategy with the maximum potential, ASPEN uses a novel minimum-cost integer flow problem to efficiently identify the best pure strategy to add. ASPEN always converges on the optimal mixed strategy for the defender.

Employing incremental strategy generation for large optimization problems is not an "out-of-the-box" approach, the problem has to be formulated in a way that allows for domain properties to be exploited. The novel contribution of ASPEN is to provide a linear formulation for the master and a minimum-cost integer flow formulation for the slave, which enables the application of strategy generation techniques.

4.2 Scale Up with Large Defender & Attacker Strategy Spaces

Whereas the previous section focused on domains where only the defender's strategy was difficult to enumerate, we now turn to domains where both defender and attacker strategies are difficult to enumerate. Once again we provide a domain example and then an algorithmic solution.

Domain Example – Road Network Security One area of great importance is securing urban city networks, transportation networks, computer networks and other network centric security domains. For example, after the terrorist attacks in Mumbai of 2008 (Chandran and Beitchman (29 November 2008)), the Mumbai police started setting up vehicular checkpoints on roads. We can model the problem faced by the Mumbai police as a security game between the Mumbai police and an attacker. In this urban security game, the pure strategies of the defender correspond to allocations of resources to edges in the network—for example, an allocation of police checkpoints to roads in the city. The pure strategies of the attacker correspond to paths from any *source* node to any *target* node—for example, a path from a landing spot on the coast to the airport.

The strategy space of the defender grows exponentially with the number of available resources, whereas the strategy space of the attacker grows exponentially with the size of the network. For example, in a fully connected graph with 20 nodes and 190 edges, the number of defender pure strategies for only 5 defender resources is $\binom{190}{5}$ or almost 2 billion, while the number of attacker pure strategies (i.e., paths without cycles) is on the order of 10^{18}. Real-world networks are significantly larger, e.g., the entire road network of the city of Mumbai has 9,503 nodes (intersections) and 20,416 edges (streets), and the security forces can deploy dozens (but not as many as number of edges) of

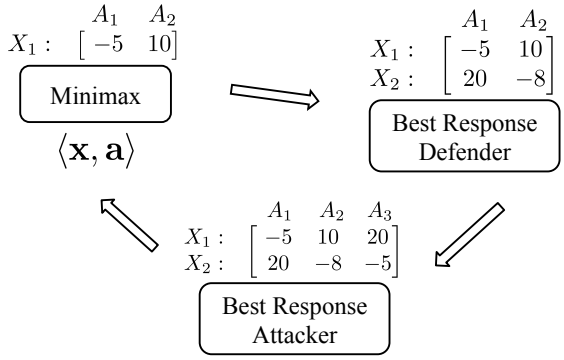


Fig. 3: Strategy Generation employed in RUGGED: The pure strategies for both the defender and the attacker are generated iteratively.

resources. In addressing this computational challenge, novel algorithms based on incremental strategy generation have been able to generate randomized defender strategies that scale up to the entire road network of Mumbai (Jain et al (2013)).

Algorithmic Solution – Double Oracle Incremental Strategy Generation (RUGGED) In domains such as the urban network security setting, the number of pure strategies of both the defender and the attacker are exponentially large. In this section, we describe the RUGGED algorithm (Jain et al (2011)), which generates pure strategies for both the defender and the attacker.

RUGGED models the domain as a zero-sum game, and computes the minimax equilibrium, since the minimax strategy is equivalent to the SSE in zero-sum games. Figure 3 shows the working of RUGGED: at each iteration, the Minimax module generates the optimal mixed strategies $\langle \mathbf{x}, \mathbf{a} \rangle$ for the two players for the current payoff matrix, the Best Response Defender module generates a new strategy for the defender that is a best response against the attacker's current strategy $\mathbf{a}$, and the Best Response Attacker module generates a new strategy for the attacker that is a best response against the defender's current strategy $\mathbf{x}$. The rows X_i in the figure are the pure strategies for the defender, they would correspond to an allocation of checkpoints in the urban road network domain. Similarly, the columns A_j are the pure strategies for the attacker, they represent the attack paths in the urban road network domain. The values in the matrix represent the payoffs to the defender. For example, in Figure 3, the row denoted by X_1 indicates that there was one checkpoint setup, and it provides a defender payoff of -5 against attacker strategy (path) A_1, and a payoff of 10 against attacker strategy (path) A_2.

In Figure 3, we show that RUGGED iterates over two oracles: the defender best response and the attacker best response oracles. In this case, the defender best response oracle has added a strategy X_2, and the attacker best response oracle then adds a strategy A_3. The algorithm stops when neither of the generated best responses improve on the current minimax strategies.

The contribution of RUGGED is to provide the mixed integer formulations for the best response modules which enable the application of such a strategy generation approach. The key once again is that RUGGED is able to converge to the optimal solution without enumerating the entire space of defender and attacker strategies. However, originally RUGGED could only compute the optimal solution for deploying up to 4 resources in real-city network with 250 nodes within a time frame of 10 hours (the complexity of this problem can be estimated by observing that both the best response problems are NP-hard themselves (Jain et al (2011))). More recent work (Jain et al (2013)) builds on RUGGED and proposes SNARES , which allows scale-up to the entire city of Mumbai, with 10–15 checkpoints.

4.3 Scale Up with Mobile Resources & Moving Targets

Whereas the previous two sections focused on incremental strategy generation as an approach for scale-up this section introduces another approach: use of compact marginal probability representations. This alternative approach is shown in use in the context of a new application of protecting ferries.

Domain Example – Ferry Protection for the US Coast Guard The United States Coast Guard is responsible for protecting domestic ferries, including the Staten Island Ferry in New York, from potential terrorist attacks. here are a number of ferries carrying hundreds of passengers in many waterside cities. These ferries are attractive targets for an attacker who can approach the ferries with a small boat packed with explosives at any time; this attacker's boat may only be detected when it comes close to the ferries. Small, fast, and well-armed patrol boats can provide protection to such ferries by detecting the attacker within a certain distance and stop him from attacking with the armed weapons. However, the numbers of patrol boats are often limited, thus the defender cannot protect the ferries at all times and locations. We thus developed a game-theoretic system for scheduling escort boat patrols to protect ferries, and this has been deployed at the Staten Island Ferry since 2013 (Fang et al (2013)).

The key research challenge is the fact that the ferries are continuously moving in a continuous domain, and the attacker could attack at any moment in time. This type of moving targets domain leads to game-theoretic models with continuous strategy spaces, which presents computational challenges. Our theoretical work showed that while it is "safe" to discretize the defender's strategy space (in the sense that the solution quality provided by our work provides a lower bound), discretizing the attacker's strategy space would result in loss of utility (in the sense that this would provide only an upper bound, and thus an unreliable guarantee of true solution quality). We developed a novel algorithm that uses a compact representation for the defender's mixed strategy space while being able to exactly model the attacker's continuous strategy space. The implemented algorithm, running on a laptop, is able to generate daily schedules for escort boats with guaranteed expected utility values.

Algorithmic Solution – Compact Strategy Representation (CASS). In this section, we describe the CASS (Solver for Continuous Attacker Strategy) algorithm (Fang et al

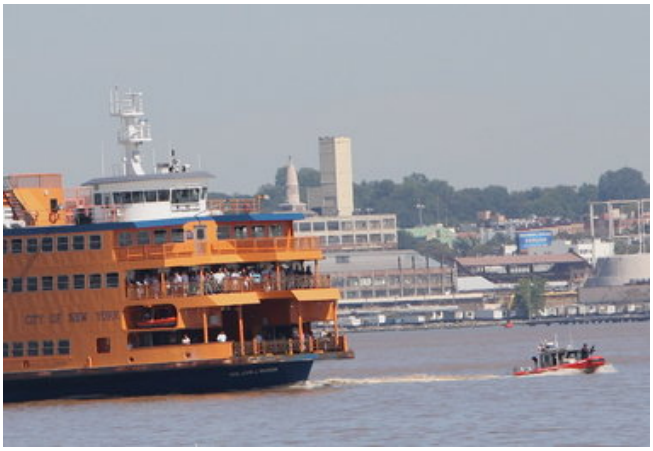


Fig. 4: Escort boats protecting the Staten Island Ferry use strategies generated by our system.

(2013)) for solving security problems where the defender has mobile patrollers to protect a set of mobile targets against the attacker who can attack these moving targets at any time during their movement. In these security problems, the sets of pure strategies for both the defender and attacker are continuous w.r.t the continuous spatial and time components of the problem domain. The CASS algorithm attempts to compute the optimal mixed strategy for the defender without discretizing the attacker's continuous strategy set; it exactly models this set using sub-interval analysis which exploits the piecewise-linear structure of the attacker's expected utility function. The insight of CASS is to compactly represent the defender's mixed strategies as a *marginal* probability distribution, overcoming the short-coming of an exponential number of pure strategies for the defender.

CASS casts problems such as the ferry protection problem mentioned above as a *zero-sum* security game in which targets move along a *one-dimensional* domain, i.e., a straight line segment connecting two terminal points. This *one-dimensional* assumption is valid as in real-world domains such as ferry protection, ferries normally move back-and-forth in a straight line between two terminals (i.e., ports) around the world. Although the targets' locations vary w.r.t time changes, these targets have a fixed daily schedule, meaning that determining the locations of the targets at a certain time is straightforward. The defender has mobile patrollers (i.e., boats) that can move along between two terminals to protect the targets. While the defender is trying to protect the targets, the attacker will decide to attack a certain target at a certain time. The probability that the attacker successfully attacks depends on the positions of the patroller at that time. Specifically, each patroller possesses a protective circle of radius within which she can detect and try to intercept any attack, whereas she is incapable of detecting the attacker prior to that radius.

In CASS, the defender's strategy space is discretized and her mixed strategy is compactly represented using flow distributions. Figure 5 shows an example of a ferry transition graph in which each node of the graph indicates a particular pair of (location,

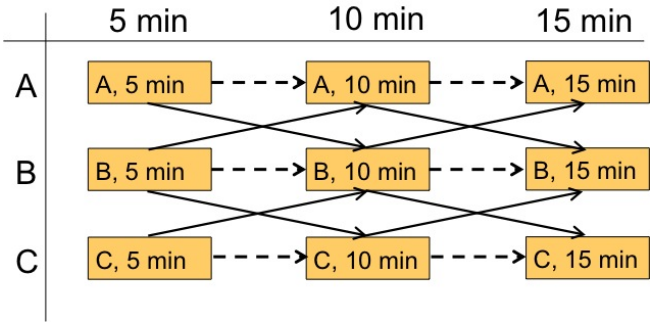


Fig. 5: An example of a ferry transition graph

time step) for the target. Here, there are three location points namely A, B, and C on a straight line where B lies between A and C. Initially, the target is at one of these location points at the 5-minute time step. Then the target moves to the next location point which is determined based on the connectivity between these points at the 10-minute time step and so on. For example, if the target is at the location point A at the 5-minute time step, denoted by (A, 5 min) in the transition graph, it can move to the location point B or stay at location point A at the 10-minute time step. The defender follows this transition graph to protect the target.

A pure strategy for the defender is defined as a trajectory of this graph, e.g., the trajectory including (A, 5 min), (B, 10 min), and (C, 15 min) indicates a pure strategy for the defender. One key challenge of this representation for the defender's pure strategies is that the transition graph consists of an exponential number of trajectories, i.e., $O(N^T)$ where N is the number of location points and T is the number of time steps. To address this challenge, CASS proposes a compact representation of the defender's mixed strategy. Instead of directly computing a probability distribution over pure strategies for the defender, CASS attempts to compute the marginal probability that the defender will follow a certain edge of the transition graph, e.g., the probability of being at the node (A, 5 min) and moving to the node (B, 10 min). We show that given a discretized strategy space for the defender, *any strategy in full representation can be mapped into a compact representation* as well as *compact representation does not lead to any loss in solution quality* compared to the full representation (see Theorem 1 in Fang et al (2013)). This compact representation allows CASS to reformulate the resource-allocation problem as computing the optimal *marginal* coverage of the defender over a number of $O(NT)$ the edges of the transition graph.

4.4 Scale Up with Boundedly Rational Attackers

One key challenge of real-world security problems is that the attacker is boundedly rational; the attacker's target choice is non-optimal. In SSGs, attacker bounded rationality is often modeled via behavior models such as Quantal Response (QR) (McFadden (1972); McKelvey and Palfrey (1995)). In general, QR attempts to predict the probability the attacker will choose each target with the intuition is that the higher the expected

utility at a target is, the more likely that the adversary will attack that target. Another behavioral model that was recently shown to provide higher prediction accuracy in predicting the attacker's behavior than QR is Subjective Utility Quantal Response (SUQR) (Nguyen et al (2013)). SUQR is motivated by the lens model which suggested that evaluation of adversaries over targets is based on a linear combination of multiple observable features (Brunswik (1952)). We provide a detailed discussion on modeling and learning the attacker's behavioral model in Section 6. However, even when the attacker's bounded rationality is modeled and those models are learned efficiently, handling multiple attackers with these behavioral models in the context of a large strategy space for the defender is a computational challenge. Therefore, in this section, we mainly focus on handling the scalability problem given behavioral models of the attacker.

To handle the problem of large defender's strategy space given behavioral models of attackers, we introduce yet another technique of scaling up, which is similar to the incremental strategy generation. Instead, here we use incremental marginal space refinement. We use the compact marginal representation, discussed earlier, but refine that space incrementally if the solution produces violates the necessary constraints.

Domain Example– Fishery Protection for US Coast Guard Fisheries are a vital natural resource from both an ecological and economic standpoint. However, fish stocks around the world are threatened with collapse due to illegal, unreported, and unregulated (IUU) fishing. The United States Coast Guard (USCG) is tasked with the responsibility of protecting and maintaining the nation's fisheries. To this end, the USCG deploys resources (both air and surface assets) to conduct patrols over fishery areas in order to deter and mitigate IUU fishing. Due to the large size of these patrol areas and the limited patrolling resources available, it is impossible to protect an entire fishery from IUU fishing at all times. Thus, an intelligent allocation of patrolling resources is critical for security agencies like the USCG.

Natural resource conservation domains such as fishery protection raise a number of new research challenges. In stark contrast to counter-terrorism settings, there is frequent interaction between the defender and attacker in these resource conservation domains. This distinction is important for three reasons. First, due to the comparatively low stakes of the interactions, rather than a handful of persons or groups, the defender must protect against numerous adversaries (potentially hundreds or even more), each of which may behave differently. Second, frequent interactions make it possible to collect data on the actions of the adversaries actions over time. Third, the adversaries are less strategic given the short planning windows between actions.

Algorithmic Solution – Incremental Constraint Generation (MIDAS). Generating effective strategies for domains such as fishery protection requires an algorithmic approach which is both *scalable* and *robust*. For scalability, the defender is responsible for protecting a large patrol area and therefore must consider a large strategy space. Even if the patrol area is discretized into a grid or graph structure, the defender must still reason over an exponential number of patrol strategies. For robustness, the defender must protect against *multiple* boundedly rational adversaries. Bounded rationality models, such as the quantal response (QR) model (McKelvey and Palfrey (1995))

and the subjective utility quantal response (SUQR) model (Nguyen et al (2013)), introduce stochastic actions, relaxing the strong assumption in classical game theory that all players are perfectly rational and utility maximizing. These models are able to better predict the actions of human adversaries and thus lead the defender to choose strategies that perform better in practice. However, both QR and SUQR are non-linear models resulting in a computationally difficult optimization problem for the defender. Combining these factors, MIDAS models a population of boundedly rational adversaries and utilizes available data to learn the behavior models of the adversaries using the subjective utility quantal response (SUQR) model in order to improve the way the defender allocates its patrolling resources.

Previous work on boundedly rational adversaries has considered the challenges of scalability and robustness separately, by (Yang et al (2012, 2013a)) and (Yang et al (2014); Haskell et al (2014)), respectively. The MIDAS algorithm was introduced to merge these two research threads for the first time by addressing scalability and robustness simultaneously. Figure 6 provides a visual overview of how MIDAS operates as an iterative process. Similar to the ASPEN algorithm described earlier, given the sheer complexity of the game being solved, the problem is decomposed using a master-slave formulation. The master utilizes multiple simplifications to create a relaxed version of the original problem which is more efficient to solve. First, a piecewise linear approximation of the security game is taken to make the optimization problem both linear and convex. Second, the complex spatio-temporal constraints associated with patrols are initially ignored and then incrementally added back using cut generation. In other words, we ignore the spatio-temporal constraint that a patroller cannot simple appear and disappear at different locations instantaneously; and that a patroller must pass through regions connecting two different regions if the patroller is go from one region to another. This significantly simplifies the master problem.

Due to the relaxations, solving the master produces a marginal strategy $\mathbf{x}$ which is a probability distribution over targets. However, the defender ultimately needs a probability distribution over patrols. Additionally, since not all of the spatio-temporal constraints are considered in the master, the relaxed solution $\mathbf{x}$ may not be a feasible solution to the original problem. Therefore, the slave checks if the marginal strategy $\mathbf{x}$ can expressed as a linear combination, i.e., probability distribution, of patrols. Otherwise, the marginal distribution is infeasible for the original problem. However, given the exponential number of patrol strategies, even performing this optimality check is intractable. Thus, column generation is used *within* the slave where only a small set of patrols is considered initially in the optimality check and the set is expanded over time. Much like previous examples of column generation in security games, e.g., (Jain et al (2010a)), new patrols are added by solving a minimum cost network flow problem using reduced cost information from the optimality check. If the optimality check fails, then the slave generates a cut which is returned to refine and constrain the master, incrementally bringing it closer to the original problem. The entire process is repeated until an optimal solution is found. Finally, MIDAS has been successfully deployed and evaluated by the USCG in the Gulf of Mexico.

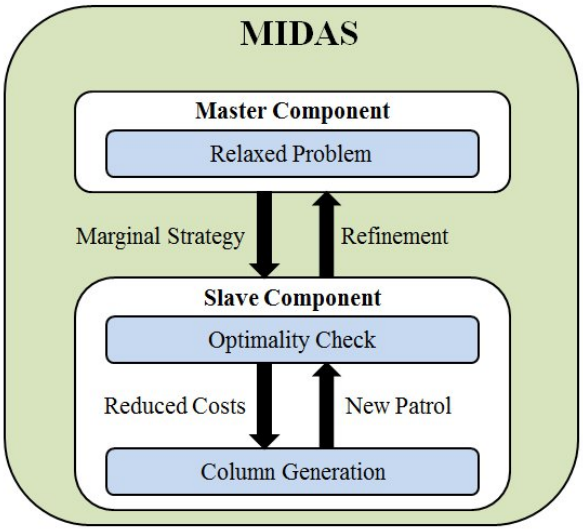


Fig. 6: Overview of the multiple iterative process within the MIDAS algorithm

4.5 Scale Up with Fine-Grained Spatial Information

Discretization is a standard way to convert a continuous problem to a discrete problem. Therefore, a grid map is often used to describe a large area. However, when fine-grained spatial information needs to be considered, each cell in the grid map should be of small size and the total number of cells is large, which leads to a significant challenge in scalability in security games especially when scheduling constraints need to be satisfied. In this section, we introduce a hierarchical modeling approach for problems with fine-grained spatial information, which is used in the context of designing foot patrols in area with complex terrain (Fang et al (2016)).

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

While game-theoretic framework can be used to address this challenge, the complex terrain of the patrolled area introduces additional complexity. In many conservation areas, high changes in elevation and the existence of large water bodies may result in a big difference in the effort needed for patrollers' movement. These factors also have a direct effect on poachers' movement. Therefore, when designing defender strategies, it

(a) A foot-patrol (Malaysia) (b) Illegal camp sign

Fig. 7: Patrols through a forest in Malaysia

is important to incorporate such topographic information. Figure 7(a) shows a sample foot patrol through a forest in Malaysia and the difficulty of conducting these patrols due to topographical constraints. Figure 7(b) shows illegal camping signs observed during those foot patrols. To generate patrol routes that contain detailed information for the patrollers and are compatible with the terrain, a fine-grained discretization of the area is necessary, leading to a large grid map of the area. On the other hand, the number of feasible routes is exponential to the number of discretized cells in the grid map due to the practical scheduling constraints such as patrol time limit and starting and ending at the base camp. Therefore, computing the optimal patrolling strategy is exceptionally computationally challenging.

Algorithmic Solution– Hierarchical Modeling Approach The hierarchical modeling approach allows us to attain a good compromise between scaling up and providing detailed guidance. This approach would be applicable in many other domains for large open area patrolling where security games are applicable, not only other green security games applications, but others including patrolling of large warehouse areas or large open campuses via robots or UAVs.

We leverage insights from hierarchical abstraction for heuristic search such as path planning (Botea et al (2004)) and apply two levels of discretization to the area of interest. We first discretize the area into large-sized *Grid Cells* and treat every grid cell as a target. We further discretize the grid cells into small-sized *Raster Pieces* and describe the spatial information for each raster piece. The defender actions are patrol routes defined over a virtual "street map" – which is built in the terms of raster pieces while aided by the grid cells in this abstraction as described below. With this hierarchical modeling, the model keeps a small number of targets and reduces the number of patrol routes while allowing for details at a fine-grained scale. The street map is a graph consisting of nodes and edges, where the set of nodes is a small subset of the raster pieces and edges are sequences of raster pieces linking the nodes. We denote nodes as Key Access

Points (KAPs) and edges as route segments. While designing foot patrols in areas with complex terrain, the street map not only helps scalability but also allows us to focus patrolling on preferred terrain features such as ridgelines which patrollers find easier to move around and are important conduits for certain mammal species such as tigers.

The street map is built in three steps: (i) determine the accessibility type for each raster piece, (ii) define KAPs and (iii) find route segments to link the KAPs. In the first step, we check the accessibility type of every raster piece. In the example domain, raster pieces in a lake are inaccessible, whereas raster pieces on ridge lines or previous patrol tracks are easily accessible. In other domains, the accessibility of a raster piece can be defined differently. The second step is to define a set of KAPs, via which patrols will be routed. We want to build the street map in such a way that each grid cell can be reached. So we first choose raster pieces which can serve as entries and exits for the grid cells as KAPs, i.e., the ones that are on the boundary of grid cells and are easily accessible. In addition, we consider existing base camps and mountain tops as KAPs as they are key points in planning the patroller's route. We choose additional KAPs to ensure KAPs on the boundary of adjacent cells are paired. Figure 8 shows identified KAPs and easily accessible pieces (black and grey raster pieces respectively). The last step is to find route segments to connect the KAPs. Instead of inefficiently finding route segments to connect each pair of KAPs on the map globally, we find route segments locally for each pair of KAPs within the same grid cell, which is sufficient to connect all the KAPs. When finding the route segment, we design a distance measure which estimates the actual patrol effort according to the accessibility type of the raster pieces. Given the distance measure, the route segment is defined as the shortest distance path linking two KAPs within the grid cell.

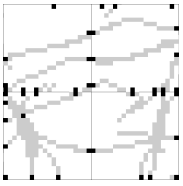


Fig. 8: KAPs (black) for 2 by 2 grid cells.

The defender's pure strategy is defined as a patrol route on the street map, starting from the base node, walking along route segments and ending with the base node, with its total distance satisfying the patrol distance limit. The defender's goal is to find an optimal mixed patrol strategy — a probability distribution over patrol routes. Based on the street map concept, we use a cutting-plane approach (Yang et al (2013b)) that is similar to MIDAS; specifically, in the master component, we use ARROW (Nguyen et al (2015)) algorithm to handle payoff uncertainty using the concept of minimax regret and in the slave component, we also use optimality check and column generation, and in generating new column (new patrol), we use a random selection approach over the street map. This framework is the core of the PAWS (Protection Assistant for Wildlife Security) application. Collaborating with two NGOs (Panthera and Rimba), PAWS has been deployed in Malaysia for tiger conservation.

5 Addressing Uncertainty in Real-world Problems

The standard security game model features a number of strong assumptions including that the defender has perfect information about the game payoff matrix as well as the attacker's behavorial model. Additionally, the defender is assumed to be capable

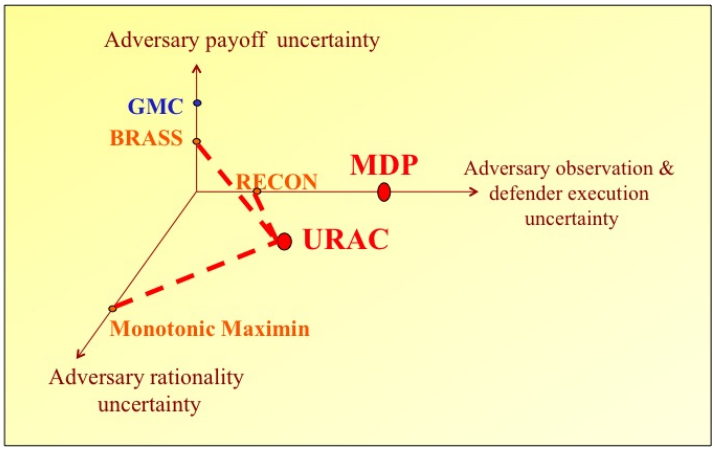


Fig. 9: Uncertainty space and algorithms

of exactly executing the computed patrolling strategy. However, uncertainty is endemic in real-world security domains and thus is may be impossible or impractical for the defender to the accurately estimate various aspects of the game. Also, there are any number of practicalities and unforeseen events that may force the defender to change their patrolling strategy. These types of uncertainty can significantly deteriorate the effectiveness of the defender's strategy and thus addressing uncertainty when generating strategies is a key challenge of solving real-world security problems. This section describes several approaches for dealing with various types of uncertainties in SSGs.

We first summarize the major types of uncertainties in SSGs as a 3-dimensional uncertainty space with the following three dimensions (Figure 9): 1) uncertainty in the adversary's payoffs; 2) uncertainty related to the defender's strategy (including uncertainty in the defender's execution and the attacker's observation); and 3) uncertainty in the adversary's rationality. These dimensions refer to three key attributes which affect both players' utilities. The origin of the uncertainty space corresponds to the case with no uncertainty. Figure 9 also shows existing algorithms for addressing uncertainty in SSGs which follow the two different approaches: 1) applying robust optimization techniques using uncertainty intervals to represent uncertainty in SSGs. For example, BRASS (Pita et al (2009b)) is a robust algorithm that only addresses attacker-payoff uncertainty, RECON (Yin et al (2011)) is another robust algorithm that focuses on addressing defender-strategy uncertainty, and Monotonic Maximin (Jiang et al (2013b)) is to handle the uncertainty in the attacker's bounded rationality. Finally, URAC (Nguyen et al (2014)) is a unified robust algorithm that handles all types of uncertainty; and 2) following Bayesian Stackelberg game model with dynamic execution uncertainty in which the uncertainty is represented using Markov Decision Process (MDP) where the time factor is incorporated.

In the following, we present two algorithmic solutions which are the representatives of these two approaches: URAC – a unified robust algorithm to handle all types of un-

certainty with uncertainty intervals and the MDP-based algorithm to handle execution uncertainty with an MDP representation of uncertainty.

5.1 Security Patrolling with Unified Uncertainty Space

Domain Example – Security in Los Angeles International Airport. Los Angeles International Airport (LAX) is the largest destination airport in the United States and serves 60-70 million passengers per year. The LAX police use diverse measures to protect the airport, which include vehicular checkpoints, police units patrolling the roads to the terminals, patrolling inside the terminals (with canines), and security screening and bag checks for passengers. The application of our game-theoretic approach is focused on two of these measures: (1) placing vehicle checkpoints on inbound roads that service the LAX terminals, including both location and timing, and (2) scheduling patrols for bomb-sniffing canine units at the different LAX terminals. The eight different terminals at LAX have very different characteristics, like physical size, passenger loads, international versus domestic flights, etc. These factors contribute to the differing risk assessments of these eight terminals. Furthermore, the numbers of available vehicle checkpoints and canine units are limited by resource constraints. Thus, it is challenging to optimally allocate these resources to improve their effectiveness while avoiding patterns in the scheduled deployments.

Fig. 10: LAX checkpoints are deployed using ARMOR.

The ARMOR system (Assistant for Randomized Monitoring over Routes) focuses on two of the security measures at LAX (checkpoints and canine patrols) and optimizes security resource allocation using Bayesian Stackelberg games. Take the vehicle checkpoints model as an example. Assuming that there are n roads, the police's strategy is placing $m < n$ checkpoints on these roads where m is the maximum number of checkpoints. ARMOR randomizes allocation of checkpoints to roads. The adversary may conduct surveillance of this mixed strategy and may potentially choose to attack

through one of these roads. ARMOR models different types of attackers with different payoff functions, representing different capabilities and preferences for the attacker. ARMOR has been successfully deployed since August 2007 at LAX (Jain et al (2010b)).

Although standard SSG-based solutions (i.e., DOBSS) have been demonstrated to improve the defender's patrolling effectiveness significantly, there remains potential improvements that can be made to further enhance the quality of such solutions such as taking uncertainties in payoff values, in the attacker's rationality, and in defender's execution into account. Therefore, we propose the unified robust algorithm, URAC, to handle these types of uncertainties by maximizing the defender's utility against the worst-case scenario resulting from these uncertainties.

Algorithmic Solution – Uncertainty Dimension Reduction (URAC). In this section, we present the robust URAC (Unified Robust Algorithmic framework for addressing unCertainties) algorithm for addressing a combination of all uncertainty types (Nguyen et al (2014)). Consider an SSG where there is uncertainty in the attacker's payoff, the defender's strategy (including the defender's execution and the attacker's observation), and the attacker's behavior, URAC represents all these uncertainty types (except for the attacker's behaviors) using uncertainty intervals. Instead of knowing exactly values of these game attributes, the defender only has prior information w.r.t the upper bounds and lower bounds of these attributes. For example, the attacker's reward if successfully attacking a target t is known to lie within the interval $[1, 3]$. Furthermore, URAC assumes the attacker monotonically responds to the defender's strategy. In other words, the higher the expected utility of a target, the more likely that the attacker will attack that target; however, the precise attacking probability is unknown for the defender. This monotonicity assumption is motivated by the Quantal Response model — a well-known human behavorial model for capturing the attacker's decision making (McKelvey and Palfrey (1995)).

Based on these uncertainty assumptions, URAC attempts to compute the optimal strategy for the defender by maximizing her utility against the worst-case scenario of uncertainty. The key challenge of this optimization problem is that it involves several types of uncertainty, resulting in multiple minimization steps for determining the worst-case scenario. Nevertheless, URAC introduces a unified representation of all these uncertainty types as a uncertainty set of attacker's responses. Intuitively, despite of any type of uncertainty mentioned above, what finally affects the defender's utility is the attacker's response, which is unknown to the defender due to uncertainty. As a result, URAC can represent the robust optimization problem as a single maximin problem.

However, the infinite uncertainty set of the attacker's responses depends on the planned mixed strategy for the defender, making this maximin problem difficult to solve if directly applying the traditional method (i.e., taking the dual maximization of the inner minimization of maximin and merging it with the outer maximization — maximin now can be represented a single maximization problem). Therefore, URAC proposes a divide-and-conquer method in which the defender's strategy set is divided into subsets such that the uncertainty set of the attacker's responses is the same for every defender strategy within each subset. This division leads to multiple sub-maximin problems which can be solved by using the traditional method. The optimal solution

of the original maximin problem is now can be computed as a maximum over all the sub-maximin problems.

5.2 Security Patrolling with Dynamic Execution Uncertainty

Domain Example – TRUSTS for Security in Transit Systems. Urban transit systems face multiple security challenges, including deterring fare evasion, suppressing crime and counter-terrorism. In particular, in some urban transit systems, including the Los Angeles Metro Rail system, passengers are legally required to purchase tickets before entering but are not physically forced to do so (Figure 11). Instead, security personnel are dynamically deployed throughout the transit system, randomly inspecting passenger tickets. This proof-of-payment fare collection method is typically chosen as a more cost-effective alternative to direct fare collection, i.e., when the revenue lost to fare evasion is believed to be less than what it would cost to directly preclude it. In the case of Los Angeles Metro, with approximately 300,000 riders daily, this revenue loss can be significant; the annual cost has been estimated at $5.6 million (Hamilton (2007)). The Los Angeles Sheriffs Department (LASD) deploys uniformed patrols on board trains and at stations for fare-checking (and for other purposes such as crime prevention). The LASD's current approach relies on humans for scheduling the patrols, which places a tremendous cognitive burden on the human schedulers who must take into account all of the scheduling complexities (e.g., train timings, switching time between trains, and schedule lengths).

The TRUSTS system (Tactical Randomization for Urban Security in Transit Systems) models the patrolling problem as a leader-follower Stackelberg game (Yin et al (2012)). The leader (LASD) pre-commits to a mixed strategy patrol (a probability distribution over all pure strategies), and riders observe this mixed strategy before deciding whether to buy the ticket or not. Both ticket sales and fines issued for fare evasion translate into revenue for the government. Therefore the utility for the leader is the total revenue (total ticket sales plus penalties). The main computational challenge is the exponentially many possible patrol strategies, each subject to both the spatial and temporal constraints of travel within the transit network under consideration. To overcome this challenge, TRUSTS uses a compact representation of the strategy space which captures the spatiotemporal structure of the domain.

The LASD conducted field tests of this TRUSTS system in the LA Metro in 2012, and one of the feedback comments from the officers was that patrols are often interrupted due to execution uncertainty such as emergencies and arrests.

Algorithmic Solution – Marginal MDP Strategy Representation Utilizing techniques from planning under uncertainty (in particular Markov Decision Processes), we proposed a general approach to dynamic patrolling games in uncertain environments, which provides patrol strategies with contingency plans (Jiang et al (2013a)). This led to schedules now being loaded onto smartphones and given to officers. If interruptions occur, the schedules are then automatically updated on the smartphone app. The LASD has conducted successful field evaluations using the smartphone app, and the TSA is currently evaluating it toward nationwide deployment. We now describe the solution

(a) Los Angeles Metro (b) Barrier-free entrance to transit system

Fig. 11: TRUSTS for transit systems

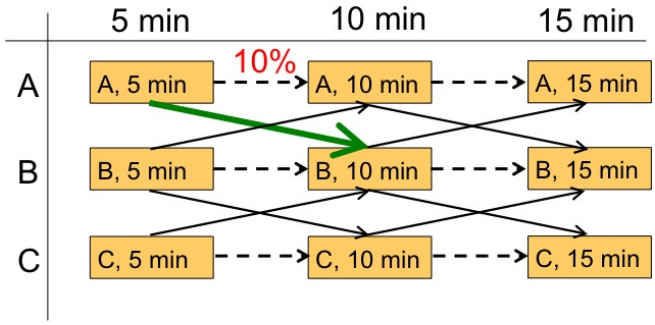


Fig. 12: An example of execution uncertainty in a transition graph

approach in more detail. Note that the targets, e.g., trains normally follow predetermined schedules, thus timing is an important aspect which determines the effectiveness of the defender's patrolling schedules (the defender needs to be at the right location at a specific time in order to protect these moving targets). However, as a result of execution uncertainty (e.g., emergencies or errors), the defender could not carry out her planned patrolling schedule in later time steps. For example, in real-world trials for TRUSTS carried out by Los Angeles Sheriff's Department (LASD), there is interruption (due to writing citations, felony arrests, and handling emergencies) in a significant fraction of the executions, causing the officers to miss the train they are supposed to catch as following the pre-generated patrolling schedule.

In this section, we present the Bayesian Stackelberg game model for security patrolling with dynamic execution uncertainty introduced by (Jiang et al (2013a)) in which the uncertainty is represented using Markov Decision Processes (MDP). The key advantage of this game-theoretic model is that patrol schedules which are computed based on Stackelberg equilibrium have contingency plans to deal with interruptions and are robust against execution uncertainty. Specifically, the security problem with execution

uncertainty is represented as a two-player Bayesian Stackelberg game between the defender and the attacker. The defender has multiple patrol units while there are also multiple types of attackers which are unknown to the defender. A (naive) patrol schedule consists of a set of sequenced commands in the following form: at time t, the patrol unit should be at location l and execute patrol action a. This patrol action a will take the unit to the next location and time if successfully executed. However, due to execution uncertainty, the patrol unit may end up at a different location and time. Figure 12 shows an example of execution uncertainty in a transition graph where if the patrol unit is currently at location A at the 5-minute time step, she is supposed to take the on-train action to move to location B in the next time step. However, unlike CASS for ferry protection in which the defender's action is deterministic, there is a 10% chance that she will still stay at location A due to execution uncertainty. This interactions of the defender with the environment when executing patrol can be represented as an MDP.

In essence, the transition graph as represented above is augmented to indicate the possibility that there are multiple uncertain outcomes possible from a given state. Solving this transition graph results in marginals over MDP policies. When a sample MDP policy is obtained and loaded on to a smart phone, it provides a patroller not only the current action, but contingency actions should the current action fail or succeed. So the MDP policy provides options for the patroller, allowing the system to handle execution uncertainty. A key challenge of computing the SSE for this type of security problem is that the dimension of the space of mixed strategies for the defender is exponential in the number of states in terms of the defender's times and locations. Therefore, instead of directly computing the mixed strategy, the defender attempts to compute the marginal probabilities of each patrolling unit reaching a state $s = (t, l)$, and taking action a which have dimensions polynomial in the sizes of the MDPs (the details of this approach are provided in (Jiang et al (2013a))).

6 Addressing Bounded Rationality and Bounded Surveillance in Real-world Problems

Game theory models the strategic interactions between multiple players who are often assumed to be perfectly rational, i.e., they will always select the optimal strategy available to them. This assumption may be applicable for high-stakes security domains such as infrastructure protection where presumably the adversary will conduct careful surveillance and planning before attacking. However, there are other security domains where the adversary may not be perfectly rational due to short planning windows or because the adversary is less strategic due to lower stakes associated with attacking. Security strategies generated under the assumption of a perfectly rational adversary are not necessarily as effective as would be feasible against a less-than-optimal response.

In addition to bounded rationality, attackers' bounded surveillance (limited capabilities in surveillance) also needs to be considered in real-world domains. In previous sections, a one-shot Stackelberg Security Game model is used, and it is assumed that the adversaries will conduct extensive surveillance to get a perfect understanding of the defender's strategy before an attack. However, this assumption does not apply to real world domains involving frequent and repeated attacks. In carrying out frequent

(a) An illegal trapping tool (b) Illegally cutting trees

Fig. 13: Examples of illegal activities in green security domains

attacks, the attackers generally do not conduct extensive surveillance before performing an attack and therefore the attackers' understanding of the defender strategy may not be up-to-date. As will be shown later in this section, if the bounded surveillance of attackers is known to the defender, the defender can exploit it to improve her average expected utility by carefully planning changes in her strategy. The improvement may depend on the level of bounded surveillance and the defender's correct understanding of the bounded surveillance. Therefore, addressing the human adversaries' bounded rationality and bounded surveillance is a fundamental challenge for applying security games to a wide variety of domains.

Domain Example – Green Security Domains. A number of newer applications are focused on resource conservation, through suppression of environmental crime. One area is protecting forests (Johnson et al (2012)), where we must protect a continuous forest area from extractors by patrols through the forest that seek to deter such extraction activity. With limited resources for performing such patrols, a patrol strategy will seek to distribute the patrols throughout the forest, in space and time, in order to minimize the resulting amount of extraction that occurs or maximize the degree of forest protection. This problem can be formulated as a Stackelberg game and the focus is on computing optimal allocations of patrol density (Johnson et al (2012)).

As mentioned earlier, endangered species poaching is reaching critical levels as the populations of these species plummet to unsustainable numbers. The global tiger population, for example, has dropped over 95% from the start of the 1900s and has resulted in three out of nine species extinctions. Depending on the area and animals poached, motivations for poaching range from profit to sustenance, with the former being more common when profitable species such as tigers, elephants, and rhinos are the targets. To counter poaching efforts and to rebuild the species' populations, countries have set up protected wildlife reserves and conservation agencies tasked with defending these large reserves. Because of the size of the reserves and the common lack of law enforcement resources, conservation agencies are at a significant disadvantage when it comes to deterring and capturing poachers. Agencies use patrolling as a primary method of securing the park. Due to their limited resources, however, patrol managers must carefully create patrols that account for many different variables (e.g., limited patrol

units to send out, multiple locations that poachers can attack at varying distances to the outpost).

6.1 Bounded Rationality Modeling and Learning

Recently, we have conducted some research on applying ideas from behavioral game theory (e.g., prospect theory (Kahneman and Tvesky (1979)) and quantal response (McFadden (1976))) within security game algorithms. One line of approaches is based on the quantal response model to predict the behaviors of the human adversary, and then to compute optimal defender strategies against such behavior of the adversary. These include BRQR (Yang et al (2011)) which follows the logit quantal response (QR) (McFadden (1976)) model, and subsequent work on subjective-utility quantal response (SUQR) models (Nguyen et al (2013)). The parameters of these models are estimated by experimental tuning. Data from a large set of participants on the Amazon Mechanical Turk (AMT) were collected and used to learn the parameters of the behavioral models to predict future attacks. In real-world domains like fisheries protection, or wildlife crime, there are repeated interactions between the defender and the adversary, where the game progresses in "rounds". We call this a Repeated Stackelberg Security Game (RSSG) where in each round the defender would play a particular strategy and the adversary would observe that strategy and act accordingly. In order to simulate this scenario and conduct experiments to identify adversary behavior in such repeated settings, an online RSSG game was developed (shown in Fig. 14) and deployed.

Fig. 14: Interface of the Wildlife Poaching game to simulate an RSSG

Wildlife Poaching Game: In the game, human subjects play the role of poachers looking to place a snare to hunt a hippopotamus in a protected wildlife park. The portion of the park shown in the map is actually a Google Maps view of a portion of the Queen Elizabeth National Park (QENP) in Uganda. The region shown is divided into a 5*5 grid, i.e. 25 distinct cells. Overlaid on the Google Maps view of the park is a heat-map, which represents the rangers' mixed strategy x — a cell i with higher coverage probability x_i is shown more in red, while a cell with lower coverage probability is shown more in green. As the subjects play the game and click on a particular region on the map, they were given detailed information about the poacher's reward, penalty and coverage probability at that region: R_i^a, P_i^a and x_i for each target i. However, the participants are unaware of the exact location of the rangers while playing the game, i.e. they do not know the pure strategy that will be played by the rangers, which is drawn randomly from mixed strategy x shown on the game interface. Thus, we model the real-world situation that poachers have knowledge of past pattern of ranger deployment but not the exact location of ranger patrols when they set out to lay snares. In the game, there were 9 rangers protecting this park, with each ranger protecting one grid cell. Therefore, at any point in time, only 9 out of the 25 distinct regions in the park are protected. A player succeeds if he places a snare in a region which is not protected by a ranger, else he is unsuccessful.

Similar to (Nguyen et al (2013)), here also we recruited human subjects on AMT and asked them to play this game repeatedly for a set of rounds with the defender strategy changing per round based on the behavioral model being used to learn the adversary's behavior. Before we discuss more about the experiments conducted, we first give a brief overview of the bounded rationality models used in our experiments to learn adversary behavior.

Bounded Rationality Models: Subjective Utility Quantal Response (SUQR) (Nguyen et al (2013)) is a behavioral model which builds upon prior work on quantal response (QR) (McFadden (1976)) according to which rather than strictly maximizing utility, an adversary stochastically chooses to attack targets, i.e., the adversary attacks a target with higher expected utility with a higher probability. SUQR proposes a new utility function called Subjective Utility, which is a linear combination of key features that are considered to be the most important in each adversary decision-making step. Nguyen et al (2013) experimented with three features: defender's coverage probability, adversary's reward and penalty (x_i, R_i^a, P_i^a) at each target i. According to this model, the probability that the adversary will attack target $i \in \mathbb{T}$ is given by:

$$q_i(\omega|x) = \frac{e^{SU_i^a(x)}}{\sum_{j \in \mathbb{T}} e^{SU_j^a(x)}} \tag{18}$$

where $SU_i^a(x)$ is the Subjective Utility of an adversary for attacking target i when defender employs strategy x and is given by:

$$SU_i^a(x) = \omega_1 x_i + \omega_2 R_i^a + \omega_3 P_i^a \tag{19}$$

The vector $\omega = (\omega_1, \omega_2, \omega_3)$ encodes information about the adversary's behavior and each component of ω indicates the relative importance the adversary gives to each at-

tribute in the decision making process. The weights are computed by performing Maximum Likelihood Estimation (MLE) on available attack data.

While behavioral models like QR (McFadden (1976)) and SUQR (Nguyen et al (2013)) assume that there is a homogeneous population of adversaries, in the real-world we face heterogeneous populations of adversaries. Therefore Bayesian SUQR was proposed to learn the behavioral model for each attack (Yang et al (2014)). Protection Assistant for Wildlife Security (PAWS) is an application which was originally created using Bayesian SUQR. However, in real-world security domains, we may have very limited data, or may only have some limited information on the biases displayed by adversaries. An alternative approach is based on robust optimization: instead of assuming a particular model of human decision making, try to achieve good defender expected utility against a range of possible models. One instance of this approach is MATCH (Pita et al (2012)), which guarantees a bound for the loss of the defender to be within a constant factor of the adversary loss if the adversary responds non-optimally. Another robust solution concept is monotonic maximin (Jiang et al (2013b)), which tries to optimize defender utility against the worst-case monotonic adversary behavior, where monotonicity is the property that actions with higher expected utility is played with higher probability. Recently, there has been attempts to combine such robust-optimization approaches with available behavior data (Haskell et al (2014)) for RSSGs, resulting in a new human behavior model called Robust SUQR. However, one question of research is how these proposed models and algorithms will fare against human subjects in RSSGs. This has been explored in recent research (Kar et al (2015)) in the 'first-of-its-kind' human subjects experiments in RSSGs over a period of 46 weeks with the 'Wildlife Poaching' game. A brief description of the experimental observations from the RSSG human subjects experiments is presented below.

Results in RSSG Experiments— An Overview: In the human subjects experiments in RSSGs, we observe that: (i) Existing approaches (QR, SUQR, Bayesian SUQR) (Nguyen et al (2013); Yang et al (2014); Haskell et al (2014)) perform poorly in initial rounds, while Bayesian SUQR which is the basis for PAWS (Yang et al (2014)), perform poorly through-out all rounds; (ii) Surprisingly, simpler models like SUQR which were originally proposed for single-shot games performed better than recent advances like Bayesian SUQR and Robust SUQR which are geared specifically towards addressing repeated SSGs. These results are shown in Figures 16(a) – 16(d). Therefore, we proposed a new model called SHARP (Stochastic Human behavior model with AttRactiveness and Probability weighting) (Kar et al (2015)) which is specifically suited for dynamic settings such as RSSGs. SHARP addresses the limitations of the existing models in the following way: (i) Modeling the adversary's adaptive decision making process in repeated SSGs, SHARP reasons based on success or failure of the adversary's past actions on exposed portions of the attack surface, where attack surface is defined as the n-dimensional space of the features used to model adversary behavior; (ii) Addressing limited exposure to significant portions of the attack surface in initial rounds, SHARP reasons about similarity between exposed and unexposed areas of the attack surface, and also incorporates a discounting parameter to mitigate adversary's lack of exposure to enough of the attack surface; (iii) Addressing the limitation that existing models do

not account for the adversary's weighting of probabilities, we incorporate a two parameter probability weighting function. We discuss these three modeling aspects of SHARP.

SHARP— Probability Weighting: SHARP has three key novelties, of which we discuss probability weighting first. The need for probability weighting became apparent when it was observed based on the initial experiments with existing models (Nguyen et al (2013); Yang et al (2014); Haskell et al (2014)) that the weight on coverage probability was positive for experiments. That is, counter-intuitively humans were modeled as being attracted to cells with high coverage probability, even though they were *not* attacking targets with very high coverage but they were going after targets with moderate to very low coverage probability. It is reasonable to hypothesize that this counter-intuitive result of a model with $\omega_1 > 0$ may be because the SUQR model may not be considering people's *actual* weighting of probability. SUQR assumes that people weigh probabilities of events in a linear fashion, while existing work on probability weighting (Kahneman and Tversky (1979); Tversky and Kahneman (1992)) suggest otherwise. To address this issue, the Subjective Utility function is augmented with a two-parameter probability weighting function (Eqn. 20) proposed by Gonzalez and Wu (Gonzalez and Wu (1999)), that can be either inverse S-shaped (concave near probability zero and convex near probability one) or S-shaped.

$$f(p) = \frac{\delta p^{\gamma}}{\delta p^{\gamma} + (1-p)^{\gamma}} \tag{20}$$

The SU of an adversary denoted by 'a' can then be computed as:

$$SU_i^a(x) = \omega_1 f(x_i) + \omega_2 R_i^a + \omega_3 P_i^a \tag{21}$$

where $f(x_i)$ for coverage probability x_i is computed as per Eqn. 20.

One of the key findings is that the curve representing human weights for probability is *S-shaped in nature, and not inverse S-shaped* as prospect theory suggests. The S-shaped curve indicates that people would overweight high probabilities and underweight low to medium probabilities. An example of learned curves on the data over several rounds of the RSSG experiment is shown in Figure 15. Recent studies (Alarie and Dionne (2001); Humphrey and Verschoor (2004); Etchart-Vincent (2009)) have also found S-shaped probability curves which contradict the inverse S-shaped observation of prospect theory. Given S-shaped probability weighting functions, the learned ω_1 was negative as it accurately captured the trend that a significantly higher number of people were attacking targets with low to medium coverage probabilities and *not* attacking high coverage targets.

SHARP— Adaptive Utility Function: A second major innovation in SHARP is the adaptive nature of the adversary and addressing the issue of attack surface exposure where ***attack surface*** α is defined as the n-dimensional space of the features used to model adversary behavior. A ***target profile*** $\beta_k \in \alpha$ is defined as a point on the attack surface α and can be associated with a target. Exposing the adversary to a lot of different target profiles would therefore mean exposing the adversary to more of the attack

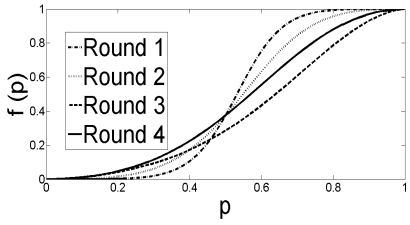


Fig. 15: Probability curves from rounds 1 to 4

surface and gathering valuable information about their behavior. While a particular target location, defined as a distinct cell in the 2-d space, can only be associated with one target profile in a particular round, more than one target may be associated with the same target profile in the same round. β_k^i denotes that target profile β_k is associated with target i in a particular round. Below is an observation from the human subjects data that reveal interesting trends in attacker behavior in RSSGs.

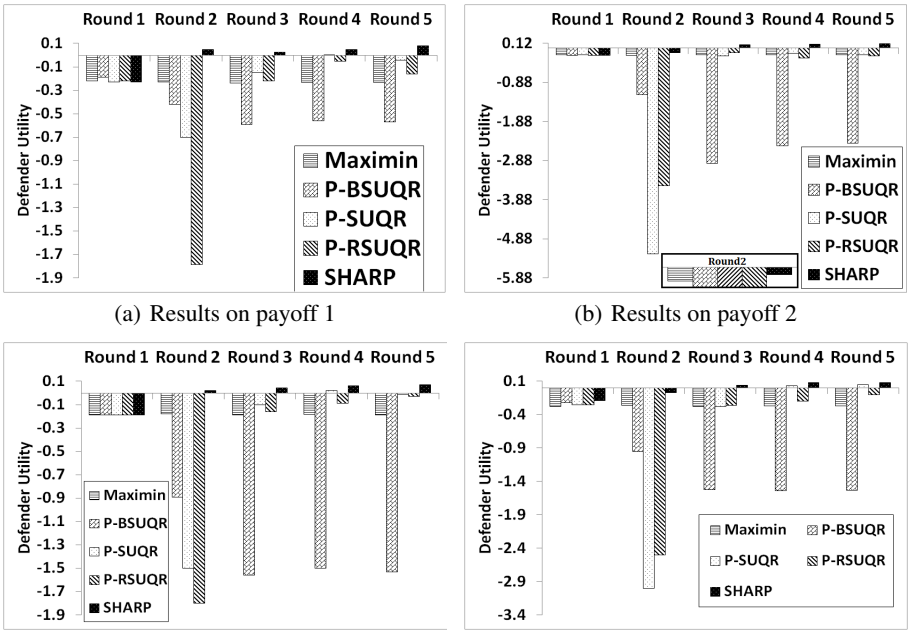
Observation 1 *Consider two sets of adversaries: (i) those who have succeeded in attacking a target associated with a particular target profile in one round; and (ii) those who have failed in attacking a target associated with a particular target profile in the same round. In the subsequent round, the first set of adversaries are significantly more likely than the second set of adversaries to attack a target with a target profile which is 'similar' to the one they attacked in the earlier round.*

Now, existing models only consider the adversary's actions from round $(r-1)$ to predict their actions in round r. However, based on Observation 1, it is clear that the adversary's actions in a particular round are dependent on his past successes and failures. The *adaptive* probability weighted subjective utility function proposed in Eq. 22 captures this adaptive nature of the adversary's behavior in such dynamic settings by capturing the shifting trends in attractiveness of various target profiles over rounds.

$$\begin{aligned} ASU_{\beta_i}^R = (1 - d * A_{\beta_i}^R)\omega_1 f(x_{\beta_i}) + (1 + d * A_{\beta_i}^R)\omega_2 \phi_{\beta_i} \\ +(1 + d * A_{\beta_i}^R)\omega_3 P_{\beta_i}^a + (1 - d * A_{\beta_i}^R)\omega_4 D_{\beta_i} \end{aligned} \tag{22}$$

Here, $A_{\beta_i}^R$ denotes the ***attractiveness*** of a target profile β_i at the end of round R and models the attacker's current affinity towards targets he attacked in the past based on his past successes and failures. The parameter d ($0 \leq d \leq 1$) in Eqn. 22 is a discounting parameter which is based on a measure of the amount of attack surface exposed and mitigates this attack surface exposure problem. Therefore, there are three main parts to SHARP's adaptive utility computation: (i) Adapting the subjective utility based on past successes and failures on exposed parts of the attack surface; (ii) Discounting to handle situations where not enough attack surface has been exposed; and (ii) Reasoning about similarity of unexposed portions of the attack surface based on other exposed parts of the attack surface. See (Kar et al (2015)) for details.

Based on the human subjects experiments with SHARP and other models on four different payoff structures, we observe in Figures 16(a) – 16(d) that SHARP completely outperforms existing approaches consistently over all rounds, most notably in initial



(a) Results on payoff 1

(b) Results on payoff 2

(c) Results on payoff 3

(d) Results on payoff 4

Fig. 16: (a), (b), (c) and (d): Defender utilities for various models on four payoff structures respectively .

rounds (refer to (Kar et al (2015)) for more details about the experimental results and observations).

6.2 Bounded Surveillance Modeling and Planning

We have discussed above some of the bounded rationality models applied to RSSGs. However, sometimes the adversaries may be bounded by their surveillance capabilities. Therefore, to account for adversaries' bounded surveillance, more recent work has generalized the perfect Stackelberg assumption and they assume that the adversaries' 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 (Fang et al (2015)). The RSSG framework, which assume that the attackers always have up-to-date information, can be seen as a special case of this more generalized Green Security Games (GSG) model.

More specifically, a GSG model considers a repeated game between a defender and multiple attackers. Each round 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. In each round, the defender chooses a mixed strategy at the beginning of the round. Different from RSSG, an attacker in GSG is characterized by his memory length and weights on recent rounds in addition to his

SUQR model parameters. The attacker is assumed to respond to a weighted sum of the defender strategies used in recent rounds (within his memory length). The defender aims to maximize her total expected utility over all the rounds.

Due to the bounded surveillance of attackers, the defender can potentially improve her average expected utility by carefully planning changes in her strategy from round to round in a GSG. Based on the GSG model, 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 defender 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.

For clarity of exposition, we first focus on the case where 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. To maximize her average expected utility, the defender could optimize over all rounds simultaneously. However, this approach is computationally expensive when the game has many rounds: it needs to solve a non-convex optimization problem with at least NT variables where N is the number of targets considered and T is the length of the game. 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 current strategy in the future round.

Therefore, we propose an algorithm named PlanAhead-M (or PA-M) in (Fang et al (2015)) that looks ahead a few steps. 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. PA-T corresponds to the optimal solution and PA-1 is the myopic strategy. Choosing $1 < M < T$ can balance the solution quality and the computation complexity.

While PA-M presents an effective way to design sequential defender strategies, we provide another algorithm called FixedSequence-M (FS-M) for GSGs in (Fang et al (2015)). 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.

7 Addressing Field Evaluation in Real-world Problems

Evidence showing the benefits of the algorithms discussed in the previous sections is definitely an important issue that is necessary for us to answer. Unlike conceptual ideas, where one can run thousands of careful simulations under controlled conditions, it is not possible to conduct such experiments in the real world with the deployed applications. Nor is it possible to provide a proof of 100% security – there is no such thing.

Instead, we focus on the specific question of: are the game-theoretic algorithms presented better at security resource optimization or security allocation than how they

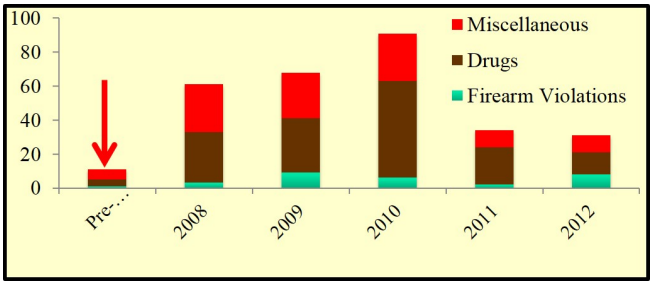


Fig. 17: ARMOR evaluation results.

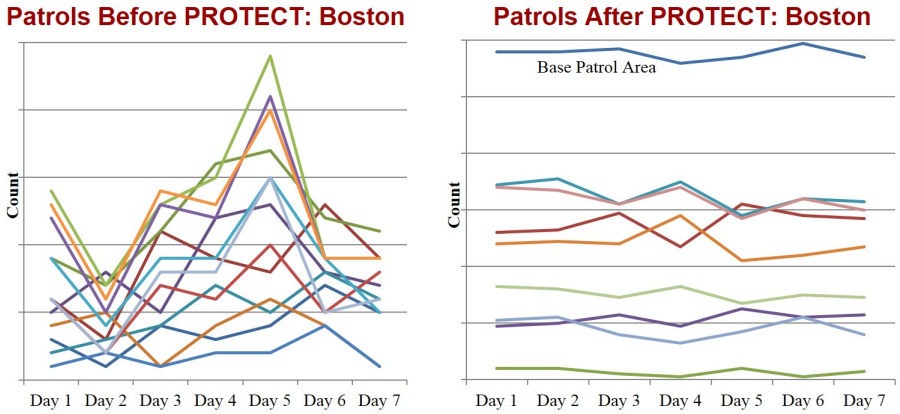


Fig. 18: PROTECT evaluation results: pre-deployment (left) and post deployment patrols (right).

were allocated previously, which was typically relying on human schedulers or a simple dice roll for security scheduling (simple dice roll is often the other "automation" that is used or offered as an alternative to our methods). We have used the following methods to illustrate these ideas. These methods range from simulations to actual field tests.

1. **Simulations (including using a "machine learning" attacker)**: We provide simulations of security schedules, e.g., randomized patrols, assignments, comparing our approach to earlier approaches based on techniques used by human schedulers. We have a machine learning based attacker who learns any patterns and then chooses to attack the facility being protected. Game-theoretic schedulers are seen to perform significantly better in providing higher levels of protections (Pita et al (2008); Jain et al (2010b)). This is also shown in Figure 17.
2. **Human adversaries in the lab**: We have worked with a large number of human subjects and security experts (security officials) to have them get through randomized security schedules, where some are schedules generated by our algorithms, and some are baseline approaches for comparison. Human subjects are paid money based on the reward they collect by successfully intruding through our security

schedules; again our game-theoretic schedulers perform significantly better (Pita et al (2009a)).

3. **Actual security schedules before and after**: For some security applications, we have data on how scheduling was done by humans (before our algorithms were deployed) and how schedules are generated after deployment of our algorithms. For measures of interest to security agencies, e.g., predictability in schedules, it is possible to compare the actual human-generated schedules versus the algorithmic schedules presented in this chapter. Again, game-theoretic schedulers are seen to perform significantly better by avoiding predictability and yet ensuring that more important targets are covered with higher frequency of patrols. Some of this data is published (Shieh et al (2012)) and is also shown in Figure 18.
4. **"Adversary" teams simulate attack**: In some cases, security agencies have deployed adversary perspective teams or "mock attacker teams" that will attempt to conduct surveillance to plan attacks; this is done before and after the game-theoretic algorithms have been deployed to check which security deployments worked better. This was done by the US Coast Guard indicating that the game-theoretic scheduler provided higher levels of deterrence (Shieh et al (2012)).
5. **Real-time comparison: human vs algorithm**: This is a test we ran on the metro trains in Los Angeles. For a day of patrol scheduling, we provided head-to-head comparison of human schedulers trying to schedule 90 officers on patrols vs an automated game-theoretic scheduler. External evaluators then provided an evaluation of these patrols; the evaluators did not know who had generated each of the schedules. The results show that while human schedulers required significant effort even for generating one schedule (almost a day), and the game-theoretic scheduler ran quickly, the external evaluators rated the game theoretic schedulers higher (with statistical significance) (Fave et al (2014a)).
6. **Actual data from deployment**: This is another test run on the metro trains in LA. We had a comparison of game-theoretic scheduler vs an alternative (in this case a uniform random scheduler augmented with real time human intelligence) to check fare evaders. In 21 days of patrols, the game-theoretic scheduler led to significantly higher numbers of fare evaders captured than the alternative (Fave et al (2014a,b)).
7. **Domain expert evaluation (internal and external)**: There have been of course significant numbers of evaluations done by domain experts comparing their own scheduling method with game theoretic schedulers and repeatedly the game theoretic schedulers have come out ahead. The fact that the game-theoretic software is now in use for several years at several different important airports, ports, air-traffic, and so on, is an indicator that the domain experts must consider this software of some value.

8 Conclusions

Security is recognized as a world-wide challenge and game theory is an increasingly important paradigm for reasoning about complex security resource allocation. We have shown in the chapter that the general model of security games is applicable (with appropriate variations) to varied security scenarios. There are applications deployed in

the real world that have led to a measurable improvement in security. We presented approaches to address four significant challenges: scalability, uncertainty, bounded rationality and field evaluation in security games.

In short, we introduced specific techniques to handle each of these challenges. For scalability, we introduced three approaches: (i) incremental strategy generation for addressing the problem of large defender strategy spaces; (ii) double oracle incremental strategy generation w.r.t large defender & attacker strategy spaces; (iii) compact representation of strategies for the case of mobile resources and moving targets (iv) cutting plane (incremental constraint generation) for handling multiple boundedly rational attacker; and (v) a hierarchical approach for incorporating fine-grained spatial information. For handling uncertainty we introduced two approaches: (i) dimensionality reduction in uncertainty space for addressing a unification of uncertainties; and (ii) Markov Decision Process with marginal strategy representation w.r.t dynamic execution uncertainty. In terms of handling attacker bounded rationality and bounded surveillance, we propose different behavioral models to capture the attackers' behaviors and introduce human subject experiments with game simulation to learn such behavioral models. Finally, for addressing field evaluation in real-world problems, we discussed two approaches: (i) data from deployment; and (ii) mock attacker team.

While the deployed game theoretic applications have provided a promising start, significant amount of research remains to be done. These are large-scale interdisciplinary research challenges that call upon multi-agent researchers to work with researchers in other disciplines, be "on the ground" with domain experts and examine real-world constraints and challenges that cannot be abstracted away.

Bibliography

Alarie Y, Dionne G (2001) Lottery decisions and probability weighting function. Journal of Risk and Uncertainty 22(1):21–33

Alpcan T, Başar T (2010) Network security: A decision and game-theoretic approach. Cambridge University Press

An B, Tambe M, Ordonez F, Shieh E, Kiekintveld C (2011) Refinement of Strong Stackelberg Equilibria in Security Games. In: Proc. of the 25th Conference on Artificial Intelligence, pp 587–593

Blocki J, Christin N, Datta A, Procaccia AD, Sinha A (2013) Audit games. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence

Blocki J, Christin N, Datta A, Procaccia AD, Sinha A (2015) Audit games with multiple defender resources. In: AAAI Conference on Artificial Intelligence (AAAI)

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

Breton M, Alg A, Haurie A (1988) Sequential Stackelberg Equilibria in Two-Person Games. Optimization Theory and Applications 59(1):71–97

Brunswik E (1952) The conceptual framework of psychology, vol 1. Univ of Chicago Pr

Chandran R, Beitchman G (29 November 2008) Battle for Mumbai Ends, Death Toll Rises to 195. Times of India `http://articles.timesofindia.indiatimes.com/2008-11-29/india/27930171_1_taj-hotel-three-terrorists-nariman-house`

Chapron G, Miquelle DG, Lambert A, Goodrich JM, Legendre S, Clobert J (2008) The impact on tigers of poaching versus prey depletion. Journal of Applied Ecology 45:16671674

Conitzer V, Sandholm T (2006) Computing the Optimal Strategy to Commit to. In: Proc. of the ACM Conference on Electronic Commerce (ACM-EC), pp 82–90

Durkota K, Lisy V, Kiekintveld C, Bosansky B (2015) Game-theoretic algorithms for optimal network security hardening using attack graphs. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '15

Etchart-Vincent N (2009) Probability weighting and the level and spacing of outcomes: An experimental study over losses. Journal of Risk and Uncertainty 39(1):45–63

Fang F, Jiang AX, Tambe M (2013) Protecting moving targets with multiple mobile resources. Journal of Artificial Intelligence Research, 48:583-634

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

Fang F, Nguyen TH, Pickles R, Lam WY, Clements GR, An B, Singh A, Tambe M, Lemieux A (2016) Deploying paws: Field optimization of the protection assistant for wildlife security. In: Proceedings of the Twenty-Eighth Innovative Applications of Artificial Intelligence Conference (IAAI 2016)

Fave FMD, Brown M, Zhang C, Shieh E, Jiang AX, Rosoff H, Tambe M, Sullivan J (2014a) Security games in the field: an initial study on a transit system(extended abstract). In: International Conference on Autonomous Agents and Multiagent Systems (AAMAS) [Short paper]

Fave FMD, Jiang AX, Yin Z, Zhang C, Tambe M, Kraus S, Sullivan J (2014b) Game-theoretic security patrolling with dynamic execution uncertainty and a case study on a real transit system. Journal of Artificial Intelligence Research, 50:321-367

Gonzalez R, Wu G (1999) On the shape of the probability weighting function. Cognitive psychology - Vol 38 pp 129–166

Hamilton BA (2007) Faregating Analysis. Report Commissioned by the LA Metro, http://boardarchives.metro.net/Items/2007/11_November/20071115EMACItem27.pdf

Haskell WB, Kar D, Fang F, Tambe M, Cheung S, Denicola LE (2014) Robust protection of fisheries with compass. In: Innovative applications of Artificial Intelligence (IAAI)

Humphrey SJ, Verschoor A (2004) The probability weighting function: experimental evidence from Uganda, India and Ethiopia. Economics Letters 84(3):419–425

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

Jain M, Kardes E, Kiekintveld C, Ordonez F, Tambe M (2010a) Security Games with Arbitrary Schedules: A Branch and Price Approach. In: Proc. of The 24th AAAI Conference on Artificial Intelligence, pp 792–797

Jain M, Tsai J, Pita J, Kiekintveld C, Rathi S, Tambe M, Ordonez F (2010b) Software Assistants for Randomized Patrol Planning for the LAX Airport Police and the Federal Air Marshal Service. Interfaces 40:267–290

Jain M, Korzhyk D, Vanek O, Pechoucek M, Conitzer V, Tambe M (2011) A Double Oracle Algorithm for Zero-Sum Security games on Graphs. In: Proc. of The 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)

Jain M, Tambe M, Conitzer V (2013) Security scheduling for real-world networks. In: AAMAS

Jiang A, Yin Z, Kraus S, Zhang C, Tambe M (2013a) Game-theoretic randomization for security patrolling with dynamic execution uncertainty. In: AAMAS

Jiang AX, Nguyen TH, Tambe M, Procaccia AD (2013b) Monotonic maximin: A robust stackelberg solution against boundedly rational followers. In: Conference on Decision and Game Theory for Security (GameSec)

Johnson M, Fang F, Yang R, Tambe M, Albers H (2012) Patrolling to Maximize Pristine Forest Area. In: Proc. of the AAAI Spring Symposium on Game Theory for Security, Sustainability and Health

Kahneman D, Tversky A (1979) Prospect Theory: An Analysis of Decision under Risk. Econometrica 47(2):263–91

Kahneman D, Tvesky A (1979) Prospect Theory: An Analysis of Decision Under Risk. Econometrica 47(2):263–291

Kar D, Fang F, Fave FD, Sintov N, Tambe M (2015) a game of thrones: When human behavior models compete in repeated stackelberg security games. In: International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)

Keteyian A (2010) TSA: Federal Air Marshals http://www.cbsnews.com/stories/2010/02/01/earlyshow/main6162291.shtml, *retrieved* Feb 1, 2011

Kiekintveld C, Jain M, Tsai J, Pita J, Tambe M, Ordonez F (2009) Computing Optimal Randomized Resource Allocations for Massive Security Games. In: Proc. of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp 689–696

Korzhyk D, Conitzer V, Parr R (2010) Complexity of computing optimal stackelberg strategies in security resource allocation games. In: Proc. of The 24th AAAI Conference on Artificial Intelligence, pp 805–810

Leitmann G (1978) On Generalized Stackelberg Strategies. Optimization Theory and Applications 26(4):637–643

Lipton R, Markakis E, Mehta A (2003) Playing large games using simple strategies. In: EC: Proceedings of the ACM Conference on Electronic Commerce, ACM New York, NY, USA, pp 36–41

McFadden D (1972) Conditional logit analysis of qualitative choice behavior. Tech. rep.

McFadden D (1976) Quantal choice analysis: A survey. Annals of Economic and Social Measurement 5(4):363–390

McKelvey RD, Palfrey TR (1995) Quantal Response Equilibria for Normal Form Games. Games and Economic Behavior 10(1):6–38

Nguyen T, Jiang A, Tambe M (2014) Stop the compartmentalization: Unied robust algorithms for handling uncertainties in security games. In: International Conference on Autonomous Agents and Multiagent Systems (AAMAS)

Nguyen TH, Yang R, Azaria A, Kraus S, Tambe M (2013) Analyzing the effectiveness of adversary modeling in security games. In: Conference on Artificial Intelligence (AAAI)

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

Paruchuri P, Pearce JP, Marecki J, Tambe M, Ordonez F, Kraus S (2008) Playing Games with Security: An Efficient Exact Algorithm for Bayesian Stackelberg Games. In: Proc. of The 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp 895–902

Pita J, Jain M, Western C, Portway C, Tambe M, Ordonez F, Kraus S, Parachuri P (2008) Deployed ARMOR protection: The Application of a Game-Theoretic Model for Security at the Los Angeles International Airport. In: Proc. of The 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp 125–132

Pita J, Bellamane H, Jain M, Kiekintveld C, Tsai J, Ordez F, Tambe M (2009a) Security applications: lessons of real-world deployment. ACM SIGecom Exchanges 8(2)

Pita J, Jain M, Ordez F, Tambe M, Kraus S, Magori-Cohen R (2009b) Effective solutions for real-world stackelberg games: When agents must deal with human uncertainties. In: The Eighth International Conference on Autonomous Agents and Multiagent Systems

Pita J, John R, Maheswaran R, Tambe M, Kraus S (2012) A robust approach to addressing human adversaries in security games. In: European Conference on Artificial Intelligence (ECAI)

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

Shieh E, An B, Yang R, Tambe M, Baldwin C, DiRenzo J, Maule B, Meyer G (2012) PROTECT: A Deployed Game Theoretic System to Protect the Ports of the United States. In: Proc. of The 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)

von Stackelberg H (1934) Marktform und Gleichgewicht. Springer, Vienna

von Stengel B, Zamir S (2004) Leadership with Commitment to Mixed Strategies. Tech. Rep. LSE-CDAM-2004-01, CDAM Research Report

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

Tversky A, Kahneman D (1992) Advances in prospect theory: Cumulative representation of uncertainty. Journal of Risk and Uncertainty 5(4):297–323

Vanek O, Yin Z, Jain M, Bosansky B, Tambe M, Pechoucek M (2012) Game-Theoretic Resource Allocation for Malicious Packet Detection in Computer Networks. In: Proc. of The 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)

Yang R, Kiekintveld C, Ordonez F, Tambe M, John R (2011) Improving Resource Allocation Strategy Against Human Adversaries in Security Games. In: IJCAI

Yang R, Ordonez F, Tambe M (2012) Computing optimal strategy against quantal response in security games. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)

Yang R, Jiang AX, Tambe M, Ordóñez F (2013a) Scaling-up security games with boundedly rational adversaries: a cutting-plane approach. In: Proceedings of the Twenty-Third International Joint conference on Artificial Intelligence, AAAI Press, pp 404–410

Yang R, Jiang AX, Tambe M, Ordonez F (2013b) Scaling-up security games with boundedly rational adversaries: A cutting-plane approach. In: IJCAI

Yang R, Ford B, Tambe M, Lemieux A (2014) Adaptive resource allocation for wildlife protection against illegal poachers. In: International Conference on Autonomous Agents and Multiagent Systems (AAMAS)

Yin Z, Jain M, Tambe M, Ordonez F (2011) Risk-Averse Strategies for Security Games with Execution and Observational Uncertainty. In: Proc. of The 25th AAAI Conference on Artificial Intelligence (AAAI), pp 758–763

Yin Z, Jiang A, Johnson M, Tambe M, Kiekintveld C, Leyton-Brown K, Sandholm T, Sullivan J (2012) TRUSTS: Scheduling Randomized Patrols for Fare Inspection in Transit Systems. In: Proc. of The 24th Conference on Innovative Applications of Artificial Intelligence (IAAI)

Index