Interfaces
Vol. 43, No. 5, September–October 2013, pp. 400–420
ISSN 0092-2102 (print) | ISSN 1526-551X (online)

http://dx.doi.org/10.1287/inte.2013.0700
© 2013 INFORMS

A Deployed Quantal Response-Based Patrol Planning System for the U.S. Coast Guard

Bo An
University of Southern California, Los Angeles, California 90089, boa@usc.edu

Fernando Ordóñez
Universidad de Chile, Santiago, RM, Chile; and University of Southern California, Los Angeles, California 90089, fordon@dii.uchile.cl

Milind Tambe, Eric Shieh, Rong Yang
University of Southern California, Los Angeles, California 90089 {tambe@usc.edu, eshieh@usc.edu, yangrong@usc.edu}

Craig Baldwin
United States Coast Guard, New London, Connecticut 06320, craig.w.baldwin@uscg.mil

Joseph DiRenzo III, Kathryn Moretti
United States Coast Guard, Portsmouth, Virginia 23704 {joseph.direnzo@uscg.mil, kathryn.a.moretti@uscg.mil}

Ben Maule
United States Coast Guard, Los Angeles, California 90045, ben.j.maule@uscg.mil

Garrett Meyer
United States Coast Guard, Seattle, Washington 98174, garrett.r.meyer@uscg.mil

In this paper, we describe the model, theory developed, and deployment of PROTECT, a game-theoretic system that the United States Coast Guard (USCG) uses to schedule patrols in the Port of Boston. The USCG evaluated PROTECT's deployment in the Port of Boston as a success and is currently evaluating the system in the Port of New York, with the potential for nationwide deployment. PROTECT is premised on an attacker-defender Stackelberg game model; however, its development and implementation required both theoretical contributions and detailed evaluations. We describe the work required in the deployment, which we group into five key innovations. First, we propose a compact representation of the defender's strategy space by exploiting equivalence and dominance, to make PROTECT efficient enough to solve real-world sized problems. Second, this system does not assume that adversaries are perfectly rational, a typical assumption in previous game-theoretic models for security. Instead, PROTECT relies on a quantal response (QR) model of the adversary's behavior. We believe this is the first real-world deployment of a QR model. Third, we develop specialized solution algorithms that can solve this problem for real-world instances and give theoretical guarantees. Fourth, our experimental results illustrate that PROTECT's QR model handles real-world uncertainties more robustly than a perfect-rationality model. Finally, we present (1) a comparison of human-generated and PROTECT security schedules, and (2) results of an evaluation of PROTECT from an analysis by human mock attackers.

Key words: game theory; security; applications; Stackelberg games.

The United States Coast Guard (USCG) continues to face challenges from potential terrorists within the maritime environment, which includes both the maritime Global Commons and the ports and waterways that make up the U.S. maritime transportation system. The former Director of National Intelligence, Dennis Blair, noted in 2010 a persistent threat "from al-Qaeda and potentially others who share its anti-Western ideology. A major terrorist attack may emanate from either outside or inside the United States" (Blair 2010, p. 8). This threat was reinforced in May 2011 following the raid on Osama Bin Laden's home, where a large trove of material was uncovered, including plans to attack an oil tanker. "There is an indication of intent, with operatives seeking the size and construction of tankers, and concluding it's best to blow them up from the inside because of the strength of their hulls" (Dozier 2011). These oil tankers transit the U.S. maritime transportation system. The USCG plays a key role in the security of this system and the protection of seaports to support the economy, environment, and way of life in the United States (Young and Orchard 2011). These threats, coupled with challenging economic times, force the USCG to operate as effectively as possible, achieving maximum benefit from every hour spent on patrol.

Recent work has successfully deployed game-theory models to help plan patrols for certain real-world security applications. Examples include assistant for randomized monitoring over routes (ARMOR), intelligent randomization in scheduling (IRIS), and game-theoretic unpredictable and randomly deployed security (GUARDS). The Los Angeles International Airport police use ARMOR to determine the location and times of road checkpoints and canine patrols (Pita et al. 2008). The IRIS software helps the U.S. Federal Air Marshal Service to schedule air marshals on international flights (Tsai et al. 2009). GUARDS (Pita et al. 2011) is under evaluation by the U.S. Transportation Security Administration (TSA) to allocate the resources available for airport protection.

This paper presents a new game-theoretic security application, *Port Resilience Operational/Tactical Enforcement to Combat Terrorism* (PROTECT), to aid the USCG. The USCG's mission includes maritime security of the U.S. coasts, ports, and inland waterways, a security domain that faces increased risks in the context of threats such as terrorism and drug trafficking. The USCG conducts patrols, as part of its ports, waterways, and coastal security (PWCS) mission, to protect the critical infrastructure and possible entry locations present at every port, which an adversary may target. Planning PWCS patrols is complicated by limited security resources, implying that USCG patrols cannot provide 24–7 coverage at any, let alone all, of the critical infrastructure; in addition, the adversary has the opportunity to observe the security patrol patterns. To assist the USCG in allocating its patrolling resources, similar to previous applications, PROTECT uses an attacker-defender Stackelberg game framework, with the USCG as the defender against terrorist adversaries that conduct surveillance before potentially launching an attack. PROTECT's solution is to provide a mixed strategy, that is, randomized patrol patterns that take into account the importance of different targets, the adversary's surveillance, and the adversary's anticipated reaction to USCG patrols. Each patrol is a sequence of patrol areas and associated defensive activities at each patrol area, and is constrained by a maximum patrol time. The output of PROTECT is a schedule of patrols that includes the time the patrols will begin, the critical infrastructure each patrol will visit, and the activities each patrol will perform at each critical infrastructure.

Although PROTECT builds on previous work on deployed security games, it extends this work and provides five key contributions that we present in this paper. Three contributions correspond to modeling and algorithmic developments, and the last two have to do with evaluating the model and deployed system. First, PROTECT represents the security game efficiently by using a compact formulation of defender strategies through dominance and equivalence analysis. Experimental results show the significant benefits of this compact representation, which enable the solution of real-world-sized problems (Shieh et al. 2012).

The second and most important contribution is PROTECT's departure from the assumption of perfectly rational human adversaries, as previous applications used. The assumption of perfect rationality is well-recognized as a limitation of classical game theory, and several research directions have been proposed to address this limitation, giving rise to the field of behavioral game theory (Camerer 2003). From this literature, we borrow the quantal response (QR) equilibrium model and adapt it to our security domain. QR models have emerged as a promising approach to model human-bounded rationality (Camerer 2003, McKelvey and Palfrey 1995, Wright and Leyton-Brown 2010), including recent results that illustrate the benefits of the QR model in the context of security games (Yang et al. 2011). We build the PROTECT system using a QR model of a human adversary in a Stackelberg game to plan USCG patrols, and believe that this is the first time that a QR model has been used in a real-world security application.

Third, we developed specialized solution algorithms that are able to solve this problem for real-world instances and give theoretical guarantees. This USCG patrolling problem can be approximated arbitrarily by a mixed-integer linear programming (MILP) formulation. This approach allows us to efficiently compute an arbitrary approximation of the global optimal defender strategy with MILPs, whose size depends on the accuracy of the approximation.

Fourth, this paper presents a detailed simulation analysis of PROTECT's robustness to uncertainty that may arise in the real world. Our results show that PROTECT's QR-based approach leads to significantly improved robustness when compared to an approach that assumes full attacker rationality. PROTECT has been in use at the Port of Boston since April 2011 (see Figure 1) and under evaluation by the USCG during this time. This real-world evaluation provides the final key contribution of this paper: we provide actual real-world data that compare human-generated schedules with those generated via a game-theoretic algorithm. We also provide results from an adversarial perspective team's (APT) analysis and compare patrols before and after the use of the PROTECT system from a viewpoint of an attacker. Again, to the best of our knowledge, this is the first time that results are given on a real-world evaluation of a game-theoretic security system. Given the success of PROTECT in Boston, the USCG is currently testing PROTECT in the Port of New York; based on the outcome, it may extend the system to other ports in the United States.

Partial results of this work have appeared in a number of conference papers, including Yang et al. (2011, 2012), and Shieh et al. (2012). In addition to the archival benefit of an article that presents this work as a whole, the current paper expands on the challenges and lessons learned during the deployment of the PROTECT system and presents more details of the real-world evaluation.

We organized the remainder of this paper as follows: We start with a discussion of related work and a detailed description of the concept of Stackelberg equilibrium. We then discuss how to model the real-world maritime patrolling problem of PWCS patrols as a Stackelberg game and its efficient compact representation. We next discuss the QR-based security game developed for the PROTECT system. In particular, we present its mathematical formulation and an efficient solution algorithm. We describe the details of the USCG's implementation, and present the empirical and real-world evaluation of the PROTECT system, respectively. Finally, we conclude the paper, outline some future research directions, and summarize lessons learned from applying PROTECT in practice.

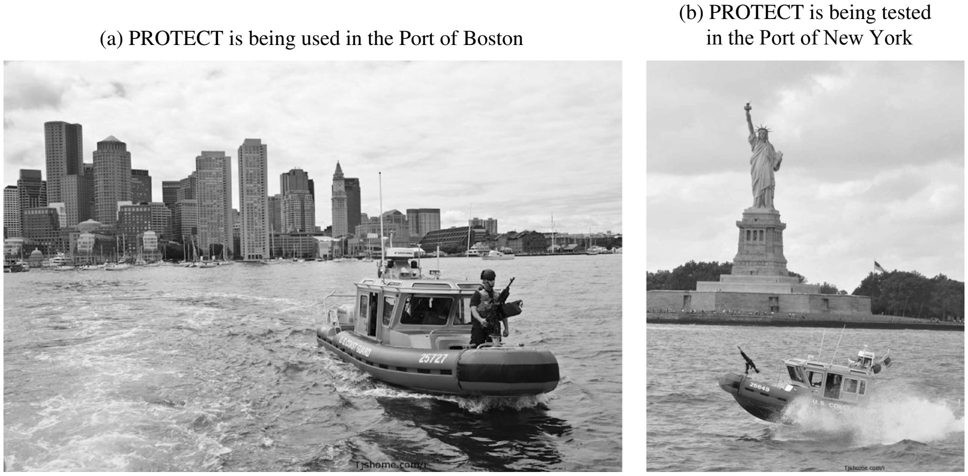
Related Work

This section reviews related work on operations research methods for security. We also introduce the concept of Stackelberg games, which is the foundation of our PROTECT system for the USCG.

OR Models for Security

The related work has four primary lines. The first applies optimization techniques to model the security domain, but does not address the strategic aspects of the problem. These methods provide a randomization strategy for the defender, but they do not consider that the adversaries can observe the defender's actions and then adjust their behavior. Examples of such approaches include those discussed in Paruchuri et al. (2006), which are based on learning, Markov decision processes (MDPs), and partially observable MDPs (POMDPs). Other examples are algorithms for perimeter patrolling in arbitrary topologies (Basilico et al. 2009), maritime patrols in simulations for deterring pirate attacks (Vanek et al. 2011), and research that looks at the impact of uncertainty in adversarial behavior (Agmon et al. 2009). Another example, the hypercube queueing model (Larson 1974), is based on queueing theory and depicts the detailed spatial operation of urban police departments and emergency medical services. Its applications include police beat design and the allocation of patrolling time. Such frameworks can address many of the problems we raise, including different target values and increasing uncertainty by using many possible patrol routes. However, they fail to account for the possibility that an intelligent attacker will observe and exploit patterns in the security policy.

A second set of work uses Stackelberg games to model a variety of security domains. Bier (2007) strongly endorses this type of modeling for security


(a) PROTECT is being used in the Port of Boston

(b) PROTECT is being tested in the Port of New York

Figure 1: USCG boats patrol the ports of Boston and New York.

problems. Game-theoretic models have been applied in a variety of homeland security settings, such as protecting critical infrastructure (Brown et al. 2006, Nie et al. 2007, Pita et al. 2008). Wein (2009) applies Stackelberg games in the context of screening visitors entering the United States. In their work, they model the U.S. Government as the leader who specifies the biometric identification strategy to maximize the detection probability using fingerprint matches, and the follower is the terrorist who can manipulate the image quality of the fingerprint. Stackelberg games have also been used for studying missile defense systems (Brown et al. 2005a) and for studying the development of an adversary's weapon systems (Brown et al. 2005b). A family of Stackelberg games known as inspection games is closely related to the security games in which we are interested and includes models of arms inspections and border patrols (Avenhaus et al. 2002). Another recent work addresses randomized security patrolling using Stackelberg games for generic police-and-robbers scenarios (Basilico et al. 2009) and perimeter patrols (Agmon et al. 2008). Our work differs from this line of work in two main aspects. First, we use a new, more efficient game representation and a MILP for modeling and solving the Stackelberg games to enable systems to scale to complex real-world situations. Second, we model the games with defender actions that incorporate the domain constraints (e.g., scheduling constraints) to more accurately model the specific games in which we are interested.

The third area of related work is the application of game-theoretic techniques that are not based on Stackelberg games to security applications. Security problems are increasingly being studied using game-theoretic analysis; these range from computer network security (Srivastava et al. 2005, wei Lye and Wing 2005) to terrorism (Sandler and Arce 2003). Babu et al. (2006) model passenger security systems at U.S. airports using linear programming approaches; however, their objective is to classify the passengers in various groups and then screen them based on the group to which they belong. Thus, although game theory has been used in security domains in the past, our work focuses on overcoming the challenges that arise from its application in the real world.

These three lines of related work that study the security domain, focus on theoretic analyses of hypothetical scenarios. In contrast, the fourth line of related work focuses on developing tools based on Stackelberg games for use in real-world security operations and addresses many practical aspects of the problem that only arise in fielded applications. (This paper belongs to this line of research.) A Stackelberg security game models an interaction between a defender and an attacker in which the defender first commits to a security policy and the attacker conducts surveillance to learn the defender's policy before launching an attack. Software decision aids based on Stackelberg games have been successfully implemented in several real-world domains. ARMOR (Pita et al. 2008), the first application of this game-theoretic framework, was successfully deployed at the Los Angeles International Airport in 2007, and has been in use since. The second application IRIS, which the U.S. Federal Air Marshal Service has used since 2009, randomizes deployment of air marshals on U.S. air carriers (Tsai et al. 2009). The U.S. TSA is evaluating a third application, GUARDS, for national deployment across more than 400 U.S. airports (Pita et al. 2011). The main differences between PROTECT and these existing applications are as follows: The maritime security domain introduces new modeling changes and new scheduling constraints that make computing the optimal allocation of resources more difficult. For example, the patrolling actions of the USCG are geographically constrained and each patrol must be finished within a specific amount of time. In addition, protecting each target requires multiple actions, each with a different level of effectiveness. Existing deployed systems assume fully rational attackers; PROTECT relaxes this unrealistic assumption and provides the first real-world deployment of a QR model of the adversary's behavior. Additionally, this paper for the first time provides real-world data that provide a (1) comparison of human-generated versus PROTECT security schedules, and (2) results from an APT (i.e., human mock attacker) analysis.

Stackelberg Game Concepts

PROTECT builds on Stackelberg games to reason about the interaction between the USCG and the adversary to provide a randomized security policy. Before introducing how we model the USCG security scenario as a Stackelberg game, we explain basic Stackelberg game concepts.

A generic Stackelberg game has two players: a leader and a follower (Fudenberg and Tirole 1991). A leader commits to a strategy and a follower then optimizes his reward, considering the strategy chosen by the leader (von Stengel and Zamir 2004). The two players in a Stackelberg game need not represent individuals; they could be groups that cooperate to execute a joint strategy, such as a police force or a terrorist organization. Each player has a set of possible specific actions. In the simplest approach, called a pure strategy, a player chooses one available action. Another approach is to assign a probability to each possible action and then select an action by probabilistic sampling, an approach called a mixed strategy. One leader action accompanied by one follower action produces an outcome, whose value is called a payoff. For a mixed strategy, the value of the outcome, as seen before sampling, is calculated by weighting the possible outcome values by their corresponding probabilities, giving what is termed an expected value. The follower can observe the leader's strategy, whether pure or mixed, and then act to optimize his orher own payoffs.

	c	*d*
a	3,1	5,0
b	2,0	4,2

Table 1: This table shows an example of payoffs in a Stackelberg game. The leader chooses either pure strategy *a* or pure strategy *b*; then the follower chooses *c* or *d*. A payoff such as (3, 1) means the leader gains value 3, and the follower gains value 1.

Table 1 shows an example with numbers. Suppose the leader chooses pure strategy (i.e., specific action a) and then the rational follower, seeking the highest follower outcome value given a, chooses pure strategy c; the leader gains a payoff of value 3 and the follower a payoff of value 1, expressed as $(3, 1)$. If instead the leader chooses b, the rational follower then chooses d to maximize the follower's payoff, and the payoffs change to $(4, 2)$, improving upon the $(3, 1)$. Now suppose the leader applies a mixed strategy by, for example, randomly selecting a with probability 0.5 or b with probability 0.5. And suppose the follower responds with a pure strategy (i.e., a planned definite action) such that the follower must decide, before the leader samples, which follower strategy to choose. Noting that c has an expected follower payoff value of 0.5 $(0.5 \times 1 + 0.5 \times 0)$, and d has an expected follower value of 1.0 $(0.5 \times 0 + 0.5 \times 2)$, the rational follower will choose d, in which case the leader's corresponding expected payoff is 4.5 $(0.5 \times 5 + 0.5 \times 4)$.

Stackelberg games are used to model the attacker-defender strategic interaction in security domains; this class of Stackelberg games (with certain restrictions on payoffs) (Yin et al. 2010) is called Stackelberg security games. The security force (defender) is modeled as the leader and the terrorist adversary (attacker) is in the role of the follower. The defender commits to a mixed (randomized) strategy, whereas the attacker conducts surveillance of these mixed strategies and responds with a pure strategy of an attack on a target. Thus, the Stackelberg game framework is a natural approximation of the real-world security scenarios. The objective is to find the optimal mixed strategy for the defender.

The standard solution concept is a strong Stackelberg equilibrium (SSE) (Breton et al. 1988, Leitmann 1978, von Stengel and Zamir 2004). In a SSE, the defender chooses an optimal strategy, accounting for the attacker's best response, under the assumption that the attacker breaks ties in the defender's favor. Strong Stackelberg equilibria exist in all Stackelberg games (Basar and Olsder 1995). A standard argument for the tie-breaking assumption of SSEs suggests that the leader is often able to induce the attacker by selecting a strategy arbitrarily close to the equilibrium that causes the follower to strictly prefer the defender's desired strategy (von Stengel and Zamir 2004). The SSE solution concept is commonly used in the related literature and is also used in all the deployed systems (Pita et al. 2008, 2011; Tsai et al. 2009; An et al. 2011a, b, c, 2012).

A Security Game Model for USCG Patrols

This section discusses the USCG domain and how to practically cast this maritime patrolling problem of PWCS patrols as a Stackelberg game. We also show a compact representation of the game.

Stackelberg games are well-established in the literature, and PROTECT models the PWCS patrol problem as a Stackelberg game with the USCG as the leader (defender) and the terrorist adversary in the role of the follower. In this Stackelberg game framework, the defender commits to a mixed (randomized) strategy of patrols, which is known to the attacker. This is a reasonable approximation of the practice because the attacker conducts surveillance to learn the mixed strategies that the defender carries out, and responds with a pure strategy of an attack on a target. The optimization objective is to find the optimal mixed strategy for the defender.

To model the USCG patrolling domain as a Stackelberg game, we need to define the set of attacker and defender strategies and the payoff function, which center on the targets in a port; ports, such as the Port of Boston, have a significant number of potential targets (i.e., critical infrastructure).

Player Strategies

The attacks an attacker can launch on different possible targets are considered that attacker's pure strategy. However, the definition of defender strategies is not as straightforward. Patrols last for some fixed duration during the day (e.g., four hours), as specified by the USCG. We generate defender strategies by grouping nearby targets into patrol areas; in real-world scenarios, such as the Port of Boston, some targets are very close to each other; thus grouping together targets according to their geographic locations is natural. The presence of patrol areas led the USCG to redefine the set of defensive activities to be performed on patrol areas to provide a more accurate and expressive model of the patrols. Activities that take a longer time to complete provide the defender with a higher payoff compared to activities that take a shorter time. This impacts the final patrol schedule; one patrol may visit fewer areas but conduct defensive activities of longer duration at the areas, and another patrol may visit more areas and activities of shorter duration.

To generate all the permutations of patrol schedules, we create a graph with the patrol areas as vertices and adjacent areas connected via edges. Using the graph of the patrol areas, PROTECT generates all possible patrol schedules; in the graph, each schedule is a closed walk that starts and ends at the patrol area, which is the base patrol area for the USCG. Each patrol schedule is a sequence of patrol areas and the associated defensive activities at each patrol area, and is constrained by a maximum patrol time. (Even when the defender just passes by a patrol area, this is treated as an activity.) The defender may visit a patrol area multiple times in a schedule because (1) of geographic constraints, and (2) each patrol is a closed walk. For example, the defender in each patrol needs to start the patrol from the base and come back to the base to finish the patrol; therefore, the defender should visit the base patrol area at least twice.

The graph, patrol time constraint, and constraint that each patrol must start from and end at the base

Patrol schedule	Target 1	Target 2	Target 3	Target 4
(1:k_1), (2:k_1), (1:k_1)	50, −50	30, −30	15, −15	−20, 20
(1:k_2), (2:k_1), (1:k_1)	100, −100	60, −60	15, −15	−20, 20
(1:k_1), (2:k_1), (1:k_2)	100, −100	60, −60	15, −15	−20, 20
(1:k_1), (3:k_1), (2:k_1), (1:k_1)	50, −50	30, −30	15, −15	10, −10
(1:k_1), (2:k_1), (3:k_1), (1:k_1)	50, −50	30, −30	15, −15	10, −10

Table 2: This table shows part of a simplified example of a game matrix that contains the payoffs for the attacker and defender. The rows denote the defender's strategies and the columns represent the attacker's strategies. (1:k_1) **means that the defender conducts action** k_1 **at patrol area 1.**

patrol area, are used to generate the defender strategies (patrol schedules). Given each patrol schedule, we calculate the total patrol schedule time (this also includes traversal time between areas, but we ignore it for expository purposes); we then verify that the total time is less than or equal to the maximum patrol time. After generating all possible patrol schedules, we form a game in which the set of defender strategies comprises patrol schedules and the set of attacker strategies is the set of targets. The attacker's strategy is based on targets instead of patrol areas because an attacker will choose to attack a single target.

In the example in Table 2, the rows correspond to the defender's strategies and the columns correspond to the attacker's strategies. This example shows two possible defensive activities: k_1 and k_2, where k_2 provides more effective protection (but also takes more time) for the defender than k_1. Suppose the time bound disallows more than two k_2 activities (given the time required for k_2) within a patrol. Patrol area 1 has two targets (target 1 and 2), and patrol areas 2 and 3 each have one target (target 3 and 4, respectively). In the example, a patrol schedule consists of a sequence of patrol areas and a defensive activity in each area. The patrol schedules are ordered so that the first patrol area in the schedule denotes which patrol area the defender must visit first. Patrol area 1 is the base patrol area, and all the patrol schedules begin and end at patrol area 1. For example, the patrol schedule in row 2 first visits patrol area 1 with activity k_2, then travels to patrol area 2 with activity k_1, and returns back to patrol area 1 with activity k_1.

Payoff Matrix

For the payoffs, if a target is the attacker's choice and the attack fails, then the defender gains a reward and the attacker receives a penalty; otherwise, the defender receives a penalty and the attacker gains a reward. Furthermore, if the defender chooses a patrol and the attacker chooses to attack a target, the defender's payoff can be represented as a linear combination of the defender reward and (or) penalty on the target and the effectiveness probability of the defensive activity performed on the target for the patrol. The effectiveness probability of the defensive activity performed on a target depends on the most effective activity on the target in the patrol. The value of the effectiveness probability is zero if the target is not in the patrol.

In the USCG problem, rewards and penalties are based on an analysis completed by a contracted company of risk analysts that looked at the targets in the Port of Boston and assigned corresponding values for each one. The types of factors considered for generating these values include economic damage and injury or loss of life. Meanwhile, the effectiveness probability A_{ij}, for different defensive activities is decided based on the duration of the activities. Longer activities lead to a higher possibility of capturing the attackers.

Although loss of life and property helps in assessing damage in case of a successful attack, assessing payoffs requires that we determine whether the loss is viewed symmetrically by the defender and attacker. Similarly, whether the payoffs are viewed symmetrically for the attacker and defender also holds for the scenario where an attack fails. These questions go to the heart of determining whether security games should be modeled as zero-sum games (Tambe 2011). Past work in security games used nonzero-sum game models (e.g., one assumption made is that the attacker might view publicity of a failed attack as a positive outcome). However, nonzero-sum games require further knowledge acquisition efforts to model the asymmetry in payoffs. For simplicity, as the first step, PROTECT starts with the assumption of a zero-sum game. However, the algorithm PROTECT uses is not restricted to zero-sum games, and the USCG has proposed to relax this assumption in the future. We also note that although Table 2 shows point estimates of payoffs, we recognize that estimates may not be accurate. Therefore, we evaluate the robustness of our approach when payoff noise, observation noise, or execution error exists.

Compact Representation

In our game, the number of defender strategies (i.e., patrol schedules) grows combinatorially, generating a scale-up challenge. To achieve scale-up, PROTECT uses a compact representation of the patrol schedules by combining equivalent patrol schedules and removing dominated schedules.

With respect to equivalence, different permutations of patrol schedules provide identical payoff results. Furthermore, if an area is visited multiple times with different activities in a schedule, we consider only the activity that provides the defender with the highest payoff, not the incremental benefit resulting from additional activities. This decision considers the trade-off between modeling accuracy and efficiency. The additional value of more activities is small. Currently, the patrol time of each schedule is relatively short (e.g., one hour), and the defender may visit a patrol area more than once within the short period and conduct an activity each time. For example, the defender may pass by a patrol area 10 minutes after conducting a more effective activity at the same patrol area. The additional value of the pass-by activity, given the more effective activity, is therefore very small. However, it leads to significant computational benefits, which we describe in this section if we just consider the most effective activity in each patrol.

Therefore, many patrol schedules are equivalent if the set of patrol areas visited and the most effective defensive activities in each patrol area in the schedules are the same, even if their order differs. We combine such equivalent patrol schedules into a single compact defender strategy, represented as a set of patrol areas and defensive activities (and minus any ordering information). The concept of combining equivalent actions is similar to action abstraction for solving large-scale dynamic games (Gilpin 2009). Table 3 presents a compact version of Table 2, which shows how the game matrix is simplified by using equivalence to form compact defender strategies; for example, the patrol schedules in rows 2 and 3 in Table 2 are represented as a compact strategy C_2 in Table 3.

Compact strategy	Target 1	Target 2	Target 3	Target 4
$C_1 = \{(1{:}k_1), (2{:}k_1)\}$	50, −50	30, −30	15, −15	−20, 20
$C_2 = \{(1{:}k_2), (2{:}k_1)\}$	100, −100	60, −60	15, −15	−20, 20
$C_3 = \{(1{:}k_1), (2{:}k_1), (3{:}k_1)\}$	50, −50	30, −30	15, −15	10, −10

Table 3: The table shows an example of compact strategies, which have at most one action for each patrol area, and a game matrix.

Next, we illustrate the concept of dominance using Table 3, and noting the difference between C_1 and C_2 is the defensive activity on patrol area 1. Because activity k_2 gives the defender a higher payoff than k_1, C_1 can be removed from the set of defender strategies because C_2 covers the same patrol areas, while giving a higher payoff for patrol area 1. PROTECT generates compact strategies in the following manner: We start with patrols that visit the most patrol areas with the least-effective activities within the patrol time limit; these activities take a shorter amount of time; but we can cover more areas within the given time limit. Then we gradually consider patrols visiting less patrol areas, but with increasingly effective activities. This process stops when we have considered all patrols in which all patrol areas are covered with the most effective activities and cannot include any additional patrol areas.

A QR Model of Human Adversary

This section presents the mathematical formulation and solution algorithm to solve the Stackelberg security game with a QR model for follower behavior.

Problem Formulation

The QR model is important in behavioral game theory (McKelvey and Palfrey 1995, Rogers et al. 2009). It suggests that instead of strictly maximizing utility, individuals respond stochastically in games: the chance of selecting a nonoptimal strategy increases as the cost of such an error decreases. Recent work (Wright and Leyton-Brown 2010) shows that quantal level-*k* (Stahl and Wilson 1994) is best suited for predicting human behavior in simultaneous-move games. (We applied the QR model instead of quantal level-*k* because in Stackelberg security games, the attacker observes the defender's strategy, so level-*k* reasoning is not applicable.) Next, we present results that show that QR-based Stackelberg security models perform better than perfectly rational models or prospect theory models in empirical results.

The QR model assumes that humans choose better actions at a higher frequency, but with noise added to the decision-making process. In the case of the Stackelberg security game we consider, only the follower has a QR model. This follower selects targets based on the attacker utility of selecting each target. The attacker chooses a target with a higher utility at a higher frequency. The defender aims to maximize the defender's expected utility, given that the adversary attacks a target following the QR model. The problem of computing the optimal defender strategy, given a QR model of the adversary, can be formulated as a nonlinear, nonconvex optimization problem. Appendix A includes the formal formulation of the QR model and the mathematical formulation of the defender's optimization problem P1.

PASAQ Algorithm for Solving the QR Model

We need to solve P1 to compute the optimal defender strategy, which requires optimally solving a nonconvex problem—generally an NP-hard problem (Vavasis 1995). However, P1 has two principle difficulties: it has a fractional objective and the objective is composed of nonlinear functions. We present the *P*iecewise linear *A*pproximation of the optimal *S*trategy *A*gainst *Q*uantal response (PASAQ) algorithm in parts: we first show that a binary search method can be used to handle the fractional objective function by successively solving nonlinear problems; then we present a transformation of these nonlinear optimization problems to a MILP using piecewise linear functions.

Binary Search Method. The key concept of the binary search method is to iteratively bound the optimal value of the fractional objective function of P1 by solving related optimization problems CF-OPT that do not have a fractional objective. The binary search algorithm first initializes the upper bound U_0 and lower bound L_0 of the optimal objective function value. Then, in each iteration, we solve a related optimization problem CF-OPT, and use the result to increase the lower bound or decrease the upper bound. The search continues until the upper and lower bounds are sufficiently close. Appendix B shows the details of the binary search method. However, the objective function in the related optimization problem is still nonconvex; therefore, directly solving it is still a hard problem. We propose the PASAQ algorithm to address this.

PASAQ: Algorithm 1 + Linear Approximation. The PASAQ algorithm computes the approximate optimal defender strategy. The key concept of PASAQ is to use a piecewise linear function to approximate the nonlinear objective function in CF-OPT, and thus convert it into a MILP problem. Such a problem can easily include assignment constraints, giving an approximate solution for a Stackelberg security game (SSG) against a QR adversary with assignment constraints.

PASAQ uniformly divides the range $[0, 1]$ into multiple pieces (segments); Appendix C shows the details of the piecewise linear approximation approach. This piecewise linear approximation leads to a mixed-integer programming problem. Furthermore, given a game instance, the solution quality of PASAQ is bounded linearly by the binary search threshold and the piecewise linear accuracy (see Appendix C, Theorem 1). Therefore, the PASAQ solution can arbitrarily be made close to the optimal solution with sufficiently small ϵ and sufficiently large K.

Implementation of the PROTECT Model Within the USCG

We now describe in detail the implementation of the PROTECT model within the USCG. The end users of PROTECT are security officers, and the system must be simple enough so that they are comfortable using it on a regular basis. In particular, the system is designed to hide as much of the complexity of the game-theoretic models as possible. PROTECT is a standalone desktop Java application. Because of security concerns, it is run on machines that are not connected to any network. The underlying solution methods use CPLEX to solve the necessary mixed-integer programs. The core architecture is divided into three modules, described in detail in the rest of this section: (1) input: various parameters and domain knowledge; (2) back end: inputs are translated into a game model passed to the Stackelberg game solver and then to a final process that generates a specific sample schedule based on the computed probabilities; (3) output: the final schedule is presented to the user.

User Input

We rely on the users and domain experts to provide the knowledge required to specify the game model. The basic inputs that PROTECT requires fall into five categories, as follows: (1) the number of available resources (i.e., boats) and possible actions at each location, and their capabilities (effectiveness of the actions), (2) the set of targets to be protected, (3) payoff values for each target, (4) types of scheduling constraints (e.g., time constraints, geographic constraints), and (5) supplemental data (e.g., geographic information). The application allows users to save and reuse this information across multiple executions. The USCG and risk analysts provide all the basic inputs.

In addition to the previously mentioned basic inputs, for the adversary's QR model we need to decide the value of parameter λ, which adjusts the probability that the attacker will select a particular action (see Appendix A). Clearly, a λ value of 0 (uniform random) and ∞ (fully rational) are not reasonable. Comparing the solutions obtained for the payoff data for Boston, we observe that an attacker's strategy with $\lambda = 4$ starts approaching a fully-rational attacker—the probability of attack focuses on a single target. In addition, an attacker's strategy with $\lambda = 0.5$ is similar to a fully-random strategy that uniformly chooses a target to attack. USCG experts who have expertise in terrorist behavior modeling suggested that we could use a broad range to represent possible λ values used by the attacker. Combining the prior observations, we determined that the attacker's strategy is best modeled with a λ value that is in the range $[0.5, 4]$, rather than a single-point estimate. We used a discrete sampling approach to determine a λ value that gives the highest average defender expected utility across attacker strategies within this range to get $\lambda = 1.5$. Specifically, the defender considers different assumptions of the attacker's λ value and, for each assumption about the λ value, the defender computes his (her) expected utility against the attacker with different λ values within the range $[0.5, 4]$. We find that when the defender assumes the attacker is using the QR model with $\lambda = 1.5$, the defender's strategy leads to the highest defender expected utility when the attacker follows the QR model with a λ value uniformly randomly chosen from the range of $[0.5, 4]$.

Back End: Game Generation and Solving

Based on all of the data provided by domain experts, we generate a Stackelberg game as previously described. However, the definition of defender strategies is not as straightforward. Patrols last for some fixed duration during the day, as specified by the USCG (e.g., four hours). Our first attempt was to model each target as a node in a graph and allow patrol paths to go from each individual target to (almost all) other targets in the port, generating an almost complete graph on the targets. This method yields the most flexible set of patrol routes that can fit within the maximum duration, covering any permutation of targets within a single patrol. This method unfortunately faced significant challenges, as follows: (1) it required determining the travel time for a patrol boat for each pair of targets, a daunting knowledge-acquisition task given the hundreds of pairs of targets; (2) it did not maximize the use of port geography, whereby boat crews could observe multiple targets at once; and (3) it was perceived as micromanaging the activities of USCG boat crews, which was undesirable. We group nearby targets (according to their geographic locations) into patrol areas and generate defender strategies based on patrol areas.

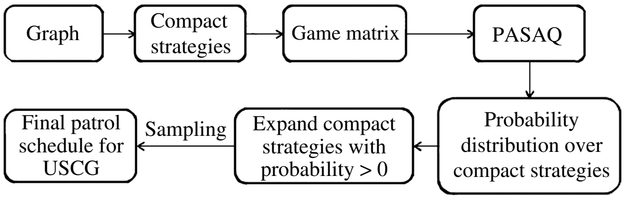

Graph
Compact strategies
Game matrix
PASAQ
Probability distribution over compact strategies
Expand compact strategies with probability > 0
Sampling
Final patrol schedule for USCG

Figure 2: The flowchart shows the seven steps of the PROTECT system.

Once we generate the explicit game model, we pass it as input to the Pasaq algorithm. Note that the Pasaq algorithm only solves the compactly represented games and its output is a probability distribution over compact strategies. Then we generate the probability of full patrol schedules. Figure 2 shows a high-level view of the steps of the algorithm using the compact representation. The compact strategies are used instead of full patrol schedules to generate the game matrix. Once the optimal probability distribution is calculated for the compact strategies, the strategies with a probability greater than zero are expanded to a complete set of patrol schedules.

In this expansion from a compact strategy to a full set of patrol schedules, we must determine the probability of choosing each patrol schedule, because a compact strategy may correspond to multiple patrol schedules. Our focus is to increase the difficulty for the attacker to conduct surveillance by increasing unpredictability, which we achieve by randomizing uniformly over all expansions of the compact defender strategies. (Creating optimal Stackelberg defender strategies that increase the attacker's difficulty of surveillance is an open research issue in the literature; we choose to maximize unpredictability as the first step.) The uniform distribution provides the maximum entropy (greatest unpredictability). Thus, all the patrol schedules generated from a single compact strategy are assigned a probability of v_i/w_i where v_i is the probability of choosing a compact strategy i and w_i is the total number of expanded patrol schedules for compact strategy i. The complete set of patrol schedules and the associated probabilities are then sampled and provided to the USCG, along with the start time of the patrol generated via uniform random sampling.

Output

From the randomized schedule, we generate a sample schedule for USCG. This sample schedule specifies exactly when each boat leaves the base, the sequence of patrol areas to be visited, and the action at each patrol area. The final schedules conform to the domain constraints that the USCG entered. The user can then review the schedule and accept it, or add additional constraints and run the scheduling process again. Table 4 shows one USCG sample schedule for five days. Each row in the table represents one patrol for one day. The second column corresponds to the starting time of each patrol and the third column corresponds to the detailed patrol schedule, which forms a closed walk with different actions at each patrol area. For example, the patrol on Day 1 starts at 3 PM from base patrol area 1, using patrol action A. Then the boat goes to patrol area 5 and conducts patrol action C. After that, the boat goes to patrol area 6 and takes patrol action A. The patrol continues by visiting patrol area 8 (action A), 9 (action B), 8 (action B), and then returns back to the base patrol area 1 via patrol areas 6 and 5, while performing action A.

Day	Hour: 0000–2300	Patrol
Day: 1	Hour: 1500	Patrol: [(1:A), (5:C), (6:A), (8:A), (9:B), (8:B), (6:A), (5:A), (1:A)]
Day: 2	Hour: 0300	Patrol: [(1:A), (5:A), (6:A), (8:A), (9:A), (8:A), (6:A), (5:C), (1:A), (2:A), (1:A)]
Day: 3	Hour: 1700	Patrol: [(1:A), (2:C), (4:B), (2:A), (1:B), (2:B), (1:A)]
Day: 4	Hour: 1600	Patrol: [(1:A), (2:B), (4:B), (2:A), (1:B)]
Day: 5	Hour: 1800	Patrol: [(1:A), (5:A), (6:A), (8:A), (9:B), (8:A), (6:A), (5:B), (1:A)]

Table 4: Sample schedules for five days are sampled from the mixed strategy.

Experimental Results

This section presents our evaluations of the model and algorithm based on planned experiments. These experiments explore the efficiency of a QR-based security game when facing human adversaries, and sensitivity analysis of the results for the USCG patrol planning problem. The first set of experiments presented considers a simple abstract security game that can be explained easily to human participants. The rest of the experiments, including the payoff values and graph (composed of nine patrol areas), are based off the Port of Boston. We ran the solutions for all experiments on a computer with an Intel Dual Core 1.4 GHz Processor and 2 GB of RAM.

Human Subject Experiment

We conducted empirical tests in which human subjects play an online game to evaluate the performance of different defender strategies. For this experiment, we consider a simple security game domain, which is easy to explain to human participants and in which they require little information. In this domain, which corresponds to the problem considered in Pita et al. (2008), a human attacker selects one of eight targets and the defender decides, without additional side constraints, where to place up to three security resources that perfectly detect an attack.

To test defender strategies, human subjects play the role of adversaries. The human subject is able to observe the leader's mixed strategy and the complete payoff matrix. The subject also knows the number of resources the defender can use, and with this information selects a target to attack. Subjects are rewarded based on the reward and (or) penalty shown for each target and the probability that a guard is behind the target (i.e., the exact randomized strategy of the defender). To motivate the subjects, they can earn or lose money based on whether they succeed in attacking a target; a subject who opens a gate not protected by the guards wins; otherwise, the subject loses. Each subject starts with $8 and each point won or lost in a game instance is worth $0.10. On average, subjects earned about $14.10.

We considered seven different payoff matrices: four are representative payoff matrices for a clustering-based classification of randomly generated payoff matrices (Yang et al. 2011). The other three payoffs are originally from Pita et al. (2009). For each payoff structure, we tested the mixed strategies generated by five algorithms, as follows:

1. BRPT: a MILP formulation for the optimal leader strategy against players whose response follows a prospect theory (PT) model (Kahneman and Tversky 1979).
2. RPT: a modified BRPT method that takes into account the uncertainty present in the adversaries' selection, caused (for example) by imprecise computations (Simon 1956).
3. BRQR: the approximate solution of (P1) by solving Pasaq.
4. DOBSS: a Stackelberg security game that assumes perfectly rational adversaries.
5. COBRA: a modified DOBSS that assumes (1) the follower might deviate within $\varepsilon > 0$, and (2) humans exhibit an anchoring bias to protect against limited-observation conditions.

The specific details of each of these algorithms can be found in the conference articles that introduced them. Specifically BRPT, RPT, and BRQR are in Yang et al. (2011), DOBSS is in Paruchuri et al. (2008), and COBRA is in Pita et al. (2010).

For each payoff matrix and the five optimal defender strategies, Figure 3 displays the average and standard deviation of the defender's expected utility with responses coming from 40 adversaries. The performance of the strategies is closer in payoffs 5∼7 than in payoffs 1∼4. The main reason is that strategies are not very different in payoffs 5∼7 in terms of the Kullback-Leibler divergence. We evaluate the statistical significance of our results using the bootstrap-t method (Wilcox 2003), and summarize the comparison as follows:

- BRQR outperforms COBRA in all seven payoff structures. The result is statistically significant in three cases ($p < 0.005$) and borderline ($p = 0.05$) in payoff 3 ($p < 0.06$). BRQR also outperforms DOBSS in all cases, with statistical significance in five of them ($p < 0.02$).

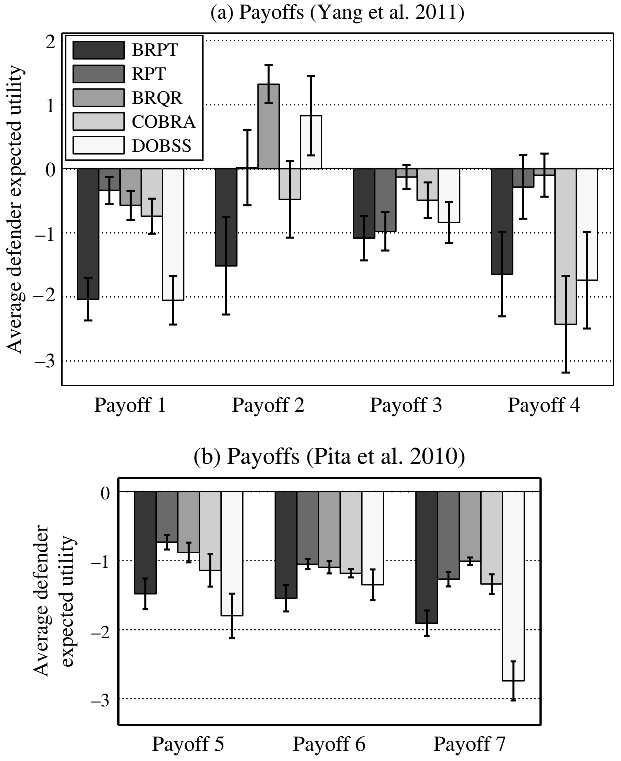


Figure 3: The average expected utility of the defender is based on different payoff structures.

- RPT outperforms COBRA, except in payoff 3. The difference is statistically significant in payoff 4 ($p < 0.005$). In payoff 3, COBRA outperforms RPT ($p > 0.07$). Meanwhile, RPT outperforms DOBSS in five payoff structures, with statistical significance in four of them ($p < 0.05$). In the other two cases, DOBSS has better performance ($p > 0.08$).
- BRQR outperforms RPT in three payoff structures with statistical significance ($p < 0.005$), and shows similar performance in the other four cases.
- BRPT is outperformed by BRQR in all cases with statistical significance ($p < 0.03$). It is also outperformed by RPT in all cases, with statistical significance in five of them ($p < 0.02$), and one borderline situation ($p < 0.06$). BRPT's failure to perform better (and worse than COBRA) is surprising.

In summary, the QR-based BRQR algorithm achieved higher defender utilities than algorithms based on the PT model and perfect rationality. Based on this observation from experiments employing in the security domain, we used the QR-based model as the decision-making function for the adversaries in the PROTECT system.

Utility Analysis

It is useful to understand whether PROTECT using Pasaq with $\lambda = 1.5$ provides an advantage when compared to: (1) a uniform random defender's strategy, (2) a mixed strategy with the assumption that the attacker attacks any target uniformly at random ($\lambda = 0$), or (3) a mixed strategy that assumes a fully rational attacker ($\lambda = \infty$). The previously existing DOBSS algorithm was used for $\lambda = \infty$ (Paruchuri et al. 2008). Additionally, a comparison with the $\lambda = \infty$ approach is important because previous applications extensively use this assumption (for our zero-sum case, DOBSS is equivalent to minimax, but the utility does not change). Typically, we may not be able to estimate the exact value of the attacker's λ value, only a possible range. Therefore, we ideally wish to show that PROTECT (using $\lambda = 1.5$ to compute the optimal defender strategy) provides an advantage over a range of λ values assumed for the attacker (not just over a point estimate) in that attacker's best response, justifying our use of the Pasaq algorithm. That is, we are distinguishing between (1) the actual λ value employed by the attacker in best responding, and (2) the λ assumed by Pasaq in computing the defender's optimal mixed strategy. Our objective is to see how sensitive the choice of (2) is, with respect to prevailing uncertainty about (1).

To achieve this, we compute the average defender utility of the four aforementioned approaches as the λ value of the attacker's strategy changes from $[0, 6]$, which subsumes the range $[0.5, 4]$ of reasonable attacker strategies. The defender's payoff values have a range of $[-10, 5]$, and the attacker's payoff values have a range of $[-5, 10]$. In Figure 4, the y-axis represents the defender's expected utility and the x-axis is the λ value used for the attacker's strategy. Both uniform random strategies perform well when the attacker's strategy is based on $\lambda = 0$. However, as λ increases, both strategies drop quickly to a very low defender-expected utility. In contrast, the Pasaq strategy with $\lambda = 1.5$ provides a higher expected utility than that assuming a fully rational attacker over a range of attacker λ values (and indeed over the range of interest), not just at $\lambda = 1.5$.

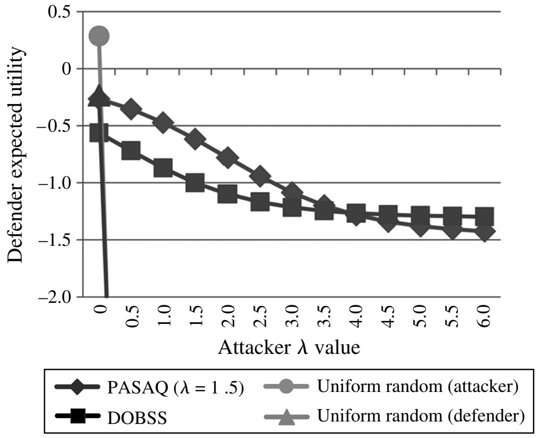


Figure 4: The defender's expected utility changes with the λ value for the attacker's strategy.

Robustness Analysis

In the real world, observation, execution, and payoffs are not always perfect because of the following: noise in the attacker's surveillance of the defender's patrols, the many tasks and responsibilities of the USCG that may cause the crew to be pulled off a patrol, and limited knowledge of the attacker's payoff values. Our hypothesis is that Pasaq with $\lambda = 1.5$ is more robust to such noise than a defender strategy that assumes full rationality of the attacker, such as DOBSS (i.e., Pasaq's expected defender utility will not degrade as much as DOBSS over the range of attacker λ of interest). We illustrate this by comparing both Pasaq and DOBSS against observation, execution, and payoff noise (Kiekintveld et al. 2011, Korzhyk et al. 2011, Yin et al. 2011).

Figure 5 shows the performance of different strategies while considering execution noise. The y-axis represents the defender's expected utility and the x-axis is the attacker's λ value. If the defender covers a target with probability p, this probability now changes to be in $[p - x, p + x]$, where x is the noise. The low execution error corresponds to $x = 0.1$, whereas high execution error corresponds to $x = 0.2$. The key conclusion here is that execution error leads to Pasaq dominating DOBSS over all

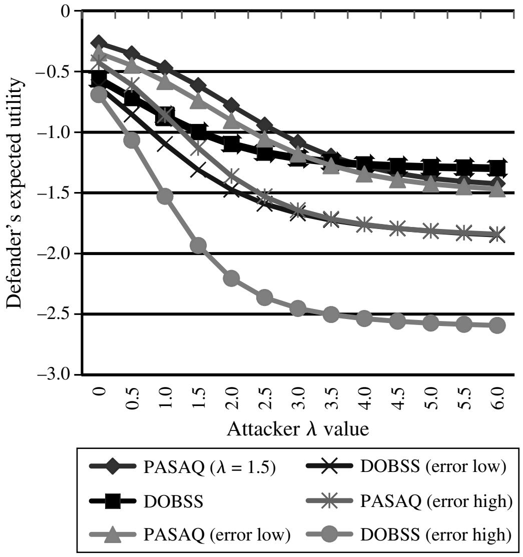


Figure 5: The defender's expected utility changes with the λ value in consideration of execution noise.

tested values of λ. For both algorithms, the defender's expected utility decreases as more execution error is added, because the defender's strategy is impacted by the additional error. When execution error is added, Pasaq dominates DOBSS because the latter seeks to maximize the minimum defender's expected utility; therefore, multiple targets have the same minimum defender utility. For DOBSS, when we add execution error, the probability that one of these targets will have less coverage increases, resulting in a lower-expected utility for the defender. For Pasaq, typically only one target has the minimum defender expected utility. As a result, changes in coverage do not impact it as much as DOBSS. As execution error increases, the advantage in the defender's expected utility of Pasaq over DOBSS increases. This section only shows the execution noise results; Shieh et al. (2012) provide the details of the observation and payoff noise results.

USCG Real-World Evaluation

In addition to the results obtained from our planned experiments, the USCG conducted its own real-world evaluation. With USCG's permission, we present some aspects of the evaluation in this paper.

Real-world scheduling data: Unlike prior publications on real-world applications of game theory for security, a key innovation of this paper is the inclusion of actual data from USCG patrols before and after the deployment of PROTECT at the Port of Boston. Figures 6 and 7 show the frequency of visits by the USCG to different patrol areas over a number of weeks. Figure 6 shows pre-PROTECT patrol visits per day by area; Figure 7 shows post-PROTECT patrol visits per day by area. The x-axis is the day of the week, and the y-axis is the number of times a patrol area is visited for a given day of the week. The y-axis is intentionally blurred for security reasons because it represents real data from the Port of Boston. Figure 6 shows more lines than in Figure 7 because during the implementation of PROTECT, new patrol areas that contained more targets and thus fewer patrol areas in the post-PROTECT figure were formed. Figure 6 depicts a definite pattern in the patrols. Although the patrols executed on Day 5 show a spike, Day 2 shows a dearth of patrols. In addition to this pattern, the

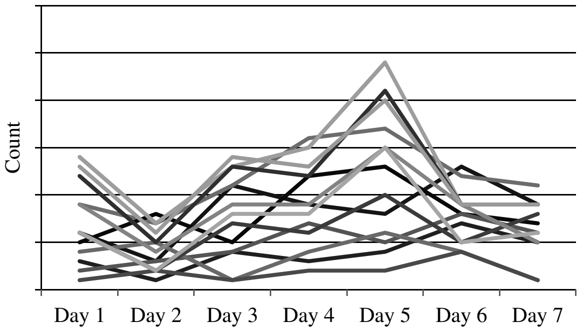


Figure 6: Pre-PROTECT patrol visits per day by area are shown, one line per area.

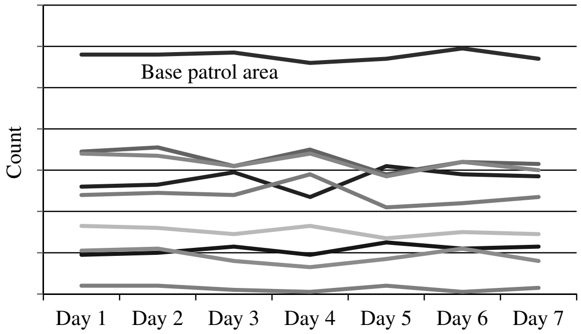


Figure 7: Post-PROTECT patrol visits per day by area are shown, one line per area.

lines in Figure 6 intersect, indicating that a higher-value target was visited more often on some days; on other days, it was visited less often. This means that a consistently high frequency of coverage of higher-value targets did not exist before PROTECT.

Figure 7 shows that the pattern of low patrols on Day 2 (see Figure 6) disappears. Furthermore, lines do not frequently intersect (i.e., higher-valued targets are visited consistently across a week). The top line in Figure 7 is the base patrol area, which is visited at a higher rate than all other patrol areas.

Adversary perspective team (*APT*): To obtain a better understanding of how the adversary views the potential targets in the port, the USCG created the adversarial perspective team (APT), a mock attacker team. The APT provides assessments from the terrorist perspective and, as a secondary function, assesses the effectiveness of the patrol activities before and after PROTECT's deployment. In its evaluation, the APT incorporates the adversary's known intent, capabilities, skills, commitment, resources, and cultural influences. In addition, it screens attack possibilities and assists in identifying the level of deterrence projected at and perceived by the adversary. For the purposes of this research, we define the adversary as an individual(s) with ties to Al-Qaeda or its affiliates.

The APT conducted a pre- and post-PROTECT assessment of the system's impact on an adversary's deterrence at the Port of Boston. This analysis uncovered a positive trend in which the effectiveness of deterrence increased from the pre-PROTECT to post-PROTECT observations.

Additional real-world indicators: PROTECT and APT's improved guidance, which APT gave to boat crews to teach them how to conduct the patrol, jointly provided a noticeable increase in the patrols' quality and effectiveness. Prior to the implementation of PROTECT, no documented reports of illicit activity existed. After its implementation, USCG crews reported some illicit activities within the port (therefore, justifying PROTECT's effectiveness) and provided a noticeable on-the-water presence with industry port partners commenting, that the Coast Guard seems to be everywhere, all the time. With no actual increase in the number of resources applied, and therefore no increase in capital or operating costs, these outcomes support the practical application of game theory in the maritime security environment.

Conclusions and Lessons Learned

This paper reports on PROTECT, a game-theoretic system, that the USCG has deployed in the Port of Boston since April 2011 to schedule its PWCS patrols. The USCG has deemed the deployment of PROTECT in the Port of Boston a success, and efforts are underway to deploy PROTECT in the Port of New York and in other ports in the United States. PROTECT uses an attacker-defender Stackelberg game model, and includes five key innovations.

First, to improve the efficiency of PROTECT, we generate a novel compact representation of the defender's strategy space, exploiting equivalence and dominance. Second, PROTECT moves away from the assumption of perfect adversary rationality seen in previous work, relying instead on a QR model of the adversary's behavior. Although the QR model has been extensively studied in the realm of behavioral game theory, to the best of our knowledge, this is its first real-world deployment. Third, this work presents a new algorithm Pasaq to overcome the difficulties in computing the best defender strategy assuming a QR adversary, including solving a nonlinear and nonconvex optimization problem and handling constraints on assigning security resources in designing defender strategies. Pasaq provides efficient computation of the defender strategy with near-optimal solution quality. Fourth, we provide experimental results illustrating that PROTECT's QR model of the adversary is better able to handle real-world uncertainties than a perfect-rationality model. Finally, for the first time in a security application evaluation, we use real-world data to provide: (1) a comparison of human-generated security schedules versus those generated via a game-theoretic algorithm, and (2) results from an APTs analysis of the impact of the PROTECT system. As a result, PROTECT has advanced the state of the art beyond previous applications of game theory for security. Building on PROTECT's initial success, we hope to deploy it at more and much larger ports. In doing so, we will consider significantly more complex attacker strategies in the future, including potential real-time surveillance and coordinated attacks. We will also consider (1) more complex defender strategies because of larger ports, and waterways and heterogeneous defense resources, (2) better models of human behavior in security games, and (3) better algorithms.

Developing the PROTECT system was a collaborative effort that involved university researchers and USCG personnel who represented decision makers, planners, and operators. Building on the lessons that Pita et al. (2011) report for working with security organizations, we informed the USCG of (1) the assumptions underlying the game-theoretic approaches (e.g., full adversary rationality, and strengths and limitations of different algorithms, rather than preselecting a simple heuristic approach, (2) the need to define and collect correct inputs for model development, and (3) a fundamental understanding of how the inputs affect the results. As a result of this project, we gained three new insights involving real-world applied research, as follows:

(1) Unforeseen positive benefits because security agencies were compelled to reexamine their assumptions. During the project, the USCG was compelled to reassess its operational assumptions as a result of working through the research problem. A positive result of this reexamination prompted the USCG to develop new PWCS mission tactics, techniques, and procedures. Through the iterative development process, the USCG reassessed the reasons why boat crews perform certain activities and whether these activities are sufficient. For example, instead of "covered" or "not covered" as the only two possibilities at a patrol point, each patrol point now has multiple sets of activities.

(2) Requirement to work with multiple teams in a security organization at multiple levels of their hierarchy. Applied research requires the research team to collaborate with planners and operators on the multiple levels of a security organization to ensure that the model considers all aspects of a complex real-world environment. When we initially started working on PROTECT, our focus was on patrolling each individual target. This appeared to micromanage the activities of boat crews; through their input, we grouped individual targets into patrol areas associated with a PWCS patrol. Input from USCG headquarters and the APT also led to other changes in PROTECT (e.g., from a fully-rational model of an adversary to a QR model).

(3) The need to prepare answers to practical end-user questions that are not always directly related to the meaty research problems. One example of the need to explain results involved a user who cited that one patrol area was being repeated and hence, randomization did not seem to occur. After assessing this concern, we determined that the cause for the repeated visits to a patrol area was its high reward—an order of magnitude greater than the rarely visited patrol areas. In another example, the user noted that PROTECT did not assign any patrols to start at 4:00 AM or 4:00 PM over a 60-day test period. The user expected patrols to be scheduled to start at any hour of the day, and asked if the program had a problem. This required us to develop a layman's briefing on probabilities, randomness, and sampling. With 60 patrol schedules, we might not choose a few start hours, given our uniform random sampling of the start time. These practitioner-based issues demonstrate the need for researchers to be not only conversant in the algorithms and math underlying the research, but also be able to explain, from a user's perspective, how solutions are accurate. An inability to address these issues would result in a lack of real-world-user confidence in the model.

Appendix A. The Quantal Response Model Formulation

The QR model assumes that humans will choose better actions at a higher frequency, but with noise added to the decision-making process. The model assumes that a player will select action i with probability q_i given by

$$q_i = \frac{e^{\lambda G_i^a(x_i)}}{\sum_{j=1}^{T} e^{\lambda G_j^a(x_j)}}. \tag{A1}$$

Here, $x_i = \sum_{j=1}^{J} a_j A_{ij}$ is the marginal coverage on target i. The parameter $\lambda \in [0, \infty)$ represents the amount of noise in the attacker's strategy. As λ approaches 0, the attacker's strategy approaches pure random selection. As the value of λ increases, the probabilities q_i become closer to a pure strategy indicating the action with largest value of $G_j^a(x_j)$.

In the case of the Stackelberg security game we consider, only the follower has a QR model. This follower selects targets corresponding to the probability q_i, where $G_j^a(x_j)$ now corresponds to the attacker utility of selecting target j. Letting R_j^a and P_j^a (or R_j^d and P_j^d) correspond to the reward and penalty to the adversary (or defender) of selecting target j, respectively, we have

$$G_j^a(x_j) = x_j P_j^a + (1 - x_j) R_j^a = R_j^a - x_j (R_j^a - P_j^a);$$
$$G_j^d(x_j) = x_j R_j^d + (1 - x_j) P_j^d = P_j^d + x_j (R_j^d - P_j^d).$$

T	Number of targets
J	Total number of compact strategies
R_i^d	Defender reward on covering target i if it is attacked
P_i^d	Defender penalty on not covering target i if it is attacked
R_i^a	Attacker reward on attacking target i if it is not covered
P_i^a	Attacker penalty on attacking target i if it is covered
λ	Noise parameter in QR model
A_{ij}	Effectiveness probability of compact strategy j on target i
a_j	Probability of choosing compact strategy j
x_i	Marginal coverage on target i

Table A.1: This table lists some PASAQ notations.

The defender aims to maximize the defender's expected utility, given that the adversary attacks target i with probability q_i. Given T targets, the coverage vector x, the defender's expected utility against a QR adversary, is:

$$U^d(x) = \sum_{i=1}^{T} q_i(x) U_i^d(x_i) = \sum_{i=1}^{T} q_i(x)(x_i R_i^d + (1-x_i)P_i^d).$$

Therefore, given J compact strategies and effectiveness matrix A_{ij}, the problem of computing the optimal defender strategy, given a QR model of the adversary, can be formulated as the following nonconvex optimization problem P1:

$$\text{P1:} \quad \begin{cases} \max\limits_{x,\,a} \dfrac{\sum_{i=1}^{T} e^{\lambda R_i^a} e^{-\lambda(R_i^a - P_i^a)x_i}((R_i^d - P_i^d)x_i + P_i^d)}{\sum_{i=1}^{T} e^{\lambda R_i^a} e^{-\lambda(R_i^a - P_i^a)x_i}} \\ x_i = \sum\limits_{j=1}^{J} a_j A_{ij}, \quad \forall i \\ \sum\limits_{j=1}^{J} a_j = 1 \\ 0 \le a_j \le 1, \quad \forall j. \end{cases}$$

For ease of reference, we summarize the notation in P1 in Table A.1.

Appendix B. Binary Search Method

For simplicity of the notation, let us define $\theta_i := e^{\lambda R_i^a} > 0$, $\beta_i := \lambda(R_i^a - P_i^a) > 0$, and $\alpha_i := R_i^d - P_i^d > 0$. Using this notation, we can express the objective function of P1 as $N(x)/D(x)$, where

- $N(x) = \sum_{i=1}^{T} \theta_i \alpha_i x_i e^{-\beta_i x_i} + \sum_{i=1}^{T} \theta_i P_i^d e^{-\beta_i x_i}$;
- $D(x) = \sum_{i=1}^{T} \theta_i e^{-\beta_i x_i} > 0$.

Also let $\mathscr{X}_f$ be the feasible region of P1 and p^* the optimal value. Therefore, we can write P1 as

$$p^* = \max_{(x,\,a) \in \mathscr{X}_f} \frac{N(x)}{D(x)}.$$

We can use a binary search method to handle the fractional objective of this type of problem.

The key concept of the binary search method is to iteratively bound the optimal value (p^*) of the fractional objective function of P1 by solving related optimization problems that do not have a fractional objective. Given a real value r, we define the optimization problem:

$$\texttt{CF-OPT:} \quad \delta_r^* = \min_{x \in \mathscr{X}_f} \{ rD(x) - N(x) \}.$$

The following result shows that $r \le p^*$ is equivalent to $\delta_r^* \le 0$.

Lemma 1. *Let $N(x)$, $D(x)$ be continuous functions defined on a closed, bounded set $\mathscr{X}_f$. Let $D(x) > 0 \ \forall x \in \mathscr{X}_f$. If $p^* = \max_{x \in \mathscr{X}_f}(N(x)/D(x))$, $r \in \mathscr{R}$, and δ_r^* as defined in* CF-OPT, *then $r \le p^* \Leftrightarrow \delta_r^* \le 0$.*

Proof. "$\Rightarrow$": Because p^* is the optimal solution value of a continuous objective over a closed, bounded set, then there exists an optimal solution x^* such that $p^* = N(x^*)/D(x^*) \ge r$. By rearranging $p^* = N(x^*)/D(x^*) \ge r$, we can get the result. "$\Leftarrow$": Similarly, there exists $\bar{x}$ such that $\delta_r^* = rD(\bar{x}) - N(\bar{x}) \le 0$, which means that $r \le N(\bar{x})/D(\bar{x}) \le p^*$. □

Therefore, by solving this related optimization problem and checking if $\delta_r^* \le 0$, we can answer whether a given r is larger or smaller than the global maximum. Algorithm 1 presents the binary search method for problem P1 using as inputs the tolerance ϵ, the payoff matrix (P_M), and the total number of security resources (*numRes*).

```
Algorithm 1 (Binary Search)
  Input: $\epsilon$, $P_M$ and $numRes$
  $(U_0, L_0) \leftarrow \texttt{EstimateBounds}(P_M, numRes)$,
     $(U, L) \leftarrow (U_0, L_0)$,
  while $U - L \ge \epsilon$ do
     $r \leftarrow \frac{U+L}{2}$
     Solve CF-OPT, let $x^r$, $\delta_r^*$ be the optimal solution
        and optimal solution value
     if $\delta_r^* \le 0$ then $L \leftarrow r$
     else $U \leftarrow r$
  endwhile
```

Algorithm 1 first initializes the upper bound (U_0) and lower bound (L_0) of the optimal objective function value on Line 2 ($\texttt{EstimateBounds}(P_M, numRes)$). Then, in each iteration, r is set to be the mean of U and L. Line 6 solves CF-OPT to check whether the current $r \le p^*$. If so, the lower bound of the binary search needs to be increased and this process also returns a valid strategy x^r. Otherwise, $p^* < r$ and the upper bound of the binary search should be decreased. The search continues until the upper and lower bounds are sufficiently close (i.e., $U - L < \epsilon$). The number of iterations in Algorithm 1 is bounded by $O(\log((U_0 - L_0)/\epsilon))$. Specifically for SSGs, we can estimate the upper and lower bounds (corresponding to $\texttt{EstimateBounds}(P_M, numRes)$ on line 2) as follows:

Lower bound: Let s_u be any feasible defender strategy. The defender utility based on using s_u against an adversary's QR is a lower bound of the optimal solution of P1. A simple example of s_u is the uniform strategy.

Upper bound: Because $P_i^d \le U_i^d \le R_i^d$, we have $U_i^d \le \max_{i=1}^T R_i^d$. We compute the defender's utility as $\sum_{i=1}^T q_i U_i^d$, where U_i^d is the defender utility on target i, and q_i is the probability that the adversary attacks target i. Thus, the maximum R_i^d serves as an upper bound of U_i^d.

Appendix C. Piecewise Linear Approximation

To demonstrate the piecewise approximation in Pasaq, we rewrite the nonlinear objective function of CF-OPT as

$$\sum_{i=1}^{T} \theta_i (r - P_i^d) e^{-\beta_i x_i} - \sum_{i=1}^{T} \theta_i \alpha_i x_i e^{-\beta_i x_i}.$$

The goal is to approximate the two nonlinear functions, $f_i^{(1)}(x_i) = e^{-\beta_i x_i}$ and $f_i^{(2)}(x_i) = x_i e^{-\beta_i x_i}$, as two piecewise linear functions in the range $x_i \in [0,1]$, for each $1 \le i \le T$.

We first uniformly divide the range $[0,1]$ into K pieces (segments). Simultaneously, we introduce a set of new variables $\{x_{ik}, k = 1\ldots K\}$ to represent the portion of x_i in each of the K pieces, $\{[(k-1)/K, k/K], k = 1,\ldots,K\}$. Therefore, $x_{ik} \in [0, 1/K]$, $\forall k = 1,\ldots,K$ and $x_i = \sum_{k=1}^K x_{ik}$. To ensure that $\{x_{ik}\}$ is a valid partition of x_i, all x_{ik} must satisfy: $x_{ik} > 0$, only if $x_{ik'} = 1/K$, $\forall k' < k$. In other words, x_{ik} can be nonzero only when all the previous pieces are completely filled.

Figures C.1(a) and C.1(b) display two examples of such a partition.

Thus, we can represent the two nonlinear functions as piecewise linear functions using $\{x_{ik}\}$. Let $\{(k/K, f_i^{(1)}(k/K)), k = 0,\ldots,K\}$ be the $K+1$ cut points of the linear segments of function $f_i^{(1)}(x_i)$, and $\{\gamma_{ik}, k = 1,\ldots,K\}$ be the slopes of each of the linear segments. Starting from $f_i^{(1)}(0)$, the piecewise linear approximation of $f_i^{(1)}(x_i)$, denoted as $L_i^{(1)}(x_i)$, is:

$$L_i^{(1)}(x_i) = f_i^{(1)}(0) + \sum_{k=1}^{K} \gamma_{ik} x_{ik} = 1 + \sum_{k=1}^{K} \gamma_{ik} x_{ik}.$$

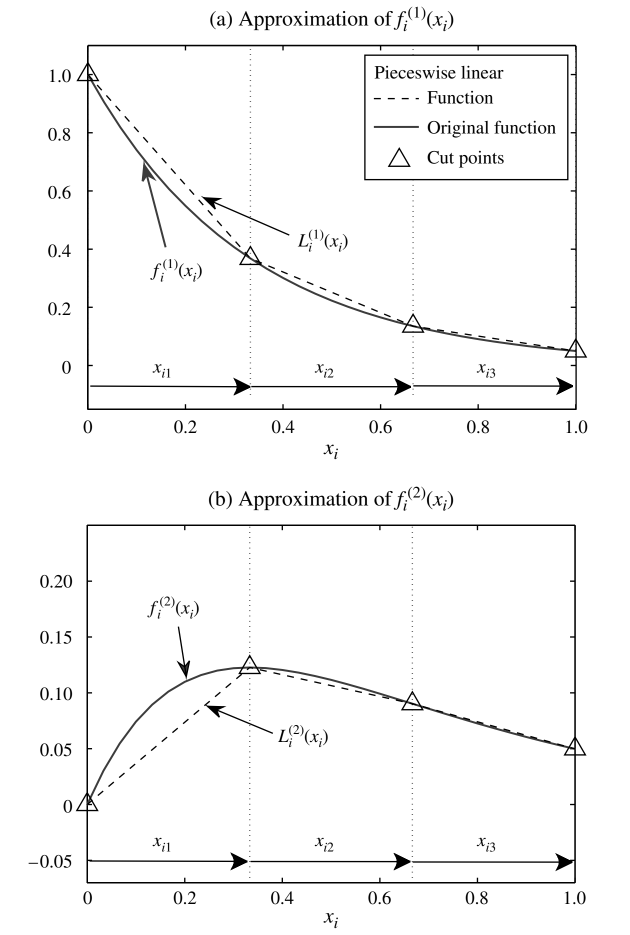
Similarly, we can obtain the piecewise linear approximation of $f_i^{(2)}(x_i)$, denoted as $L_i^{(2)}(x_i)$:

$$L_i^{(2)}(x_i) = f_i^{(2)}(0) + \sum_{k=1}^{K} \mu_{ik} x_{ik} = \sum_{k=1}^{K} \mu_{ik} x_{ik},$$

where $\{\mu_{ik}, k = 1..K\}$ is the slope of each linear segment, and $\{(K/K, f_i^{(2)}(K/K)), k = 0,\ldots,L\}$ is the $L+1$ endpoints of linear segments of function $f_i^{(2)}(x_i)$. To represent the objective function of CF-OPT as a piecewise linear function, we divide each variable x_i into L parts, $\{x_{ik}, k = 1,\ldots,L\}$, with each part related to the lth linear segments of the function. Therefore, we could write the objective function of CF-OPT as

$$\sum_{i}^{T} \theta_i (k - P_i^d) \left(1 + \sum_{k=1}^{K} \gamma_{ik} x_{ik}\right) - \sum_{i}^{T} \theta_i \alpha_i \sum_{k=1}^{K} \mu_{ik} x_{ik},$$

where γ_{ik} represent the slope of the lth line segment of function $e^{-\beta_i x_i}$ and μ_{ik} represent the slope of lth linear segment of function $x_i e^{-\beta_i x_i}$.


(a) Approximation of $f_i^{(1)}(x_i)$

(b) Approximation of $f_i^{(2)}(x_i)$

Figure C.1: Piecewise linear approximation is based on dividing the strategy space in pieces.

Pasaq consists of Algorithm 1, but with CF-OPT rewritten as follows:

$$\min_{x,z,a} \left\{ \sum_{i=1}^{T} \theta_i (r - P_i^d) \left(1 + \sum_{k=1}^{K} \gamma_{ik} x_{ik}\right) - \sum_{i=1}^{T} \theta_i \alpha_i \sum_{k=1}^{K} \mu_{ik} x_{ik} \right\}$$

$$\text{s.t.}\quad 0 \le x_{ik} \le \frac{1}{K}, \quad \forall i,\ k = 1,\ldots,K, \tag{C1}$$

$$z_{ik} \frac{1}{K} \le x_{ik}, \quad \forall i,\ k = 1,\ldots,K-1, \tag{C2}$$

$$x_{i(k+1)} \le z_{ik}, \quad z_{ik} \in \{0,1\}, \quad \forall i,\ k = 1,\ldots,K-1, \tag{C3}$$

$$\sum_{k=1}^{K} x_{ik} = \sum_{A_j \in \mathcal{A}} a_j A_{ij}, \quad \forall i, \tag{C4}$$

$$\sum_{A_j \in \mathcal{A}} a_j = 1, \tag{C5}$$

$$0 \le a_j \le 1, \quad \forall j. \tag{C6}$$

$\underline{\theta} := \min_{i=1}^{T} \theta_i$	$\bar{R}^d := \max_{i=1}^{T} \lvert R_i^d \rvert$	$\bar{\beta} := \max_{i=1}^{T} \beta_i$
$\bar{\theta} := \max_{i=1}^{T} \theta_i$	$\bar{P}^d := \max_{i=1}^{T} \lvert P_i^d \rvert$	$\bar{\alpha} := \max_{i=1}^{T} \alpha_i$

Table C.1: This table shows notations for error-bound proof.

Let us refer to the previous MILP formulation as PASAQ-MILP.

The solution provided by Pasaq is in the feasible region of P1 as shown in Lemma 2.

Lemma 2. *The feasible region for $x = \langle x_i = \sum_{k=1}^{K} x_{ik}, 1 \leq i \leq T \rangle$ of* PASAQ-MILP *is equivalent to that of* P1.

Justification. The auxiliary integer variable z_{ik} indicates whether $x_{ik} = 1/K$. Equation (C2) enforces that $z_{ik} = 0$ only when $x_{ik} < 1/K$. Simultaneously, Equation (C3) enforces that $x_{i(k+1)}$ is positive only if $z_{ik} = 1$. Hence, $\{x_{ik},\ k = 1 \ldots K\}$ is a valid partition of x_i, $x_i = \sum_{k=1}^{K} x_{ik}$, and $x_i \in [0,1]$. Thus, the feasible region of PASAQ-MILP is equivalent to P1. However, Pasaq approximates the minimum value of CF-OPT by using PASAQ-MILP, and furthermore approximately solves P1 using binary search. Hence, we need to show an error bound on the solution quality of Pasaq.

We define two constants that are decided by the game payoffs: $C_1 = (\bar{\theta}/\underline{\theta})e^{\bar{\beta}}\{(\bar{R}^d + \bar{P}^d)\bar{\beta} + \bar{\alpha}\}$ and $C_2 = 1 + (\bar{\theta}/\underline{\theta})e^{\bar{\beta}}$ (the notation is defined in Table C.1). In the following, we are interested in obtaining a bound on the difference between p^* (the optimal result obtained from P1) and $Obj_{P1}(\tilde{x}^*)$, where $\tilde{x}^*$ is the strategy obtained from Pasaq.

Theorem 1. *Let $\tilde{x}^*$ be the defender strategy computed by* Pasaq, *p^* is the optimal defender expected utility*

$$0 \leq p^* - Obj_{P1}(\tilde{x}^*) \leq 2C_1 \frac{1}{K} + (C_2 + 1)\epsilon.$$

The proof of Theorem 1 can be found in Yang et al. (2012). Theorem 1 suggests that, given a game instance, the solution quality of Pasaq is bounded linearly by the binary search threshold ϵ and the piecewise linear accuracy $1/K$. Therefore, the Pasaq solution can be made arbitrarily close to the optimal solution with sufficiently small ϵ and sufficiently large K.

Acknowledgments

We thank the staff members at the USCG offices, and particularly sector Boston, for their exceptional collaboration. The views expressed herein are those of the author(s) and are not to be construed as official or reflecting the views of the Commandant or of the United States Coast Guard. This research was supported by the United States Department of Homeland Security through the National Center for Risk and Economic Analysis of Terrorism Events (CREATE) [Grant 2010-ST-061-RE0001].

References

Agmon N, Kraus S, Kaminka GA, Sadov V (2009) Adversarial uncertainty in multi-robot patrol. *Proc. 21st Internat. Joint Conf. Artificial Intelligence* (*IJCAI*) (International Joint Conferences on Artificial Intelligence, San Francisco), 1811–1817.

Agmon N, Sadov V, Kaminka GA, Kraus S (2008) The impact of adversarial knowledge on adversarial planning in perimeter patrol. *Proc. 7th Internat. Conf. Autonomous Agents Multiagent Systems* (*AAMAS*) (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 55–62.

An B, Jain M, Tambe M, Kiekintveld C (2011a) Mixed-initiative optimization in security games: A preliminary report. *Proc. AAAI Spring Sympos. Help Me Help You: Bridging the Gaps in Human-Agent Collaboration* (Association for the Advancement of Artificial Intelligence, Menlo Park, CA), 8–11.

An B, Tambe M, Ordóñez F, Shieh E, Kiekintveld C (2011b) Refinement of strong Stackelberg equilibria in security games. *Proc. 25th Conf. Artificial Intelligence* (Association for the Advancement of Artificial Intelligence, Menlo Park, CA), 587–593.

An B, Pita J, Shieh E, Tambe M, Kiekintveld C, Marecki J (2011c) GUARDS and PROTECT: Next generation applications of security games. *ACM Special Interest Group on Electronic Commerce* (*SIGECOM*), Vol. 10 (ACM, New York), 31–34.

An B, Kempe D, Kiekintveld C, Shieh E, Singh S, Tambe M, Vorobeychik Y (2012) Security games with limited surveillance. *Proc. 26th Conf. Artificial Intelligence* (Association for the Advancement of Artificial Intelligence, Menlo Park, CA), 1241–1248.

Avenhaus R, von Stengel B, Zamir S (2002) *Inspection Games* (North-Holland, Amsterdam), 1947–1987.

Babu L, Lin L, Batta R (2006) Passenger grouping under constant threat probability in an airport security system. *Eur. J. Oper. Res.* 168(2):633–644.

Basar T, Olsder GJ (1995) *Dynamic Noncooperative Game Theory* (Academic Press, San Diego).

Basilico N, Gatti N, Amigoni F (2009) Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. *Proc. 8th Internat. Conf. Autonomous Agents Multiagent Systems* (*AAMAS*) (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 500–503.

Bier VM (2007) Choosing what to protect. *Risk Anal.* 27(3):607–620.

Blair D (2010) Annual threat assessment of the US intelligence community for the senate select committee on intelligence. Accessed January 9, 2012, http://www.isisnucleariran.org/assets/pdf/2010_NIE.pdf.

Breton M, Alg A, Haurie A (1988) Sequential Stackelberg equilibria in two-person games. *Optim. Theory Appl.* 59(1):71–97.

Brown G, Carlyle M, Diehl D, Kline J, Wood K (2005a) A two-sided optimization for theater ballistic missile defense. *Oper. Res.* 53(5):745–763.

Brown G, Carlyle M, Royset J, Wood K (2005b) *The Next Wave in Computing, Optimization and Decision Technologies* (Springer, New York), 3–17.

Brown G, Carlyle M, Salmerón J, Wood K (2006) Defending critical infrastructure. *Interfaces* 36(6):530–544.

Camerer CF (2003) *Behavioral Game Theory: Experiments in Strategic Interaction* (Princeton University Press, Princeton, NJ).

Dozier K (2011) Bin laden trove of documents sharpen US aim. Accessed January 9, 2012, http://www.msnbc.msn.com/id/43331634/ns/us_news-security/t/bin-laden-trove-documents-sharpen-us-aim/.

Fudenberg D, Tirole J (1991) *Game Theory* (MIT Press, Cambridge, MA).

Gilpin A (2009) Algorithms for abstracting and solving imperfect information games. Doctoral dissertation, Carnegie Mellon University, Pittsburgh, PA.

Kahneman D, Tversky A (1979) Prospect theory: An analysis of decision under risk. *Econometrica* 47(2):263–291.

Kiekintveld C, Marecki J, Tambe M (2011) Approximation methods for infinite bayesian Stackelberg games: Modeling distributional uncertainty. *Proc. 10th Internat. Conf. Autonomous Agents and Multiagent Systems* (*AAMAS*) (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 1005–1012.

Korzhyk D, Conitzer V, Parr R (2011) Solving Stackelberg games with uncertain observability. *Proc. 10th Internat. Conf. Autonomous Agents Multiagent Systems* (*AAMAS*) (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 1013–1020.

Larson RC (1974) A hypercube queueing modeling for facility location and redistricting in urban emergency services. *J. Comput. Oper. Res.* 1(1):67–95.

Leitmann G (1978) On generalized Stackelberg strategies. *Optim. Theory Appl.* 26(4):637–643.

McKelvey RD, Palfrey TR (1995) Quantal response equilibria for normal form games. *Games Econom. Behav.* 10(1):6–38.

Nie X, Batta R, Drury L (2007) Optimal placement of suicide bomber detectors. *Military Oper. Res.* 12(2):65–78.

Paruchuri P, Tambe M, Ordonez F, Kraus S (2006) Security in multiagent systems by policy randomization. *Proc. 5th Internat. Conf. Autonomous Agents and Multiagent Systems* (*AAMAS*) (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 273–280.

Paruchuri P, Pearce JP, Marecki J, Tambe M, Ordóñez F, Kraus S (2008) Playing games with security: An efficient exact algorithm for Bayesian Stackelberg games. *Proc. 7th Internat. Conf. Autonomous Agents and Multiagent Systems* (*AAMAS*) (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 895–902.

Pita J, Jain M, Tambe M, Ordóñez F, Kraus S (2010) Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence* 174(15):1142–1171.

Pita J, Tambe M, Kiekintveld C, Cullen S, Steigerwald E (2011) GUARDS—Game theoretic security allocation on a national scale. *Proc. 10th Internat. Conf. Autonomous Agents Multiagent Systems* (*AAMAS*) (International Foundation for Autonomous Agents and Multiagent Systems Richland, SC), 37–44.

Pita J, Jain M, Ordóñez F, Tambe M, Kraus S, Magori-Cohen R (2009) Effective solutions for real-world Stackelberg games: When agents must deal with human uncertainties. *Proc. 8th Internat. Conf. Autonomous Agents Multiagent Systems* (*AAMAS*) (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 369–376.

Pita J, Jain M, Western C, Portway C, Tambe M, Ordóñez F, Kraus S, Parachuri P (2008) Deployed ARMOR protection: The application of a game-theoretic model for security at the Los Angeles International Airport. *Proc. 7th Internat. Conf. Autonomous Agents Multiagent Systems* (*AAMAS*) (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 125–132.

Rogers BW, Palfrey TR, Camerer CF (2009) Heterogeneous quantal response equilibrium and cognitive hierarchies. *J. Econom. Theory* 144(4):1440–1467.

Sandler T, Arce D (2003) Terrorism and game theory. *Simulation Gaming* 34(3):319–337.

Shieh E, Bo A, 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. *Proc. 11th Internat. Conf. Autonomous Agents Multiagent Systems* (*AAMAS*) (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 13–20.

Simon H (1956) Rational choice and the structure of the environment. *Psych. Rev.* 63(2):129–138.

Srivastava V, Neel J, MacKenzie AB, Menon R, Luiz A, Dasilva LA, Hicks JE, Reed JH, Gilles RP (2005) Using game theory to analyze wireless ad hoc networks. *IEEE Comm. Surveys Tutuorials* 7(4):46–56.

Stahl D, Wilson P (1994) Experimental evidence on players' models of other players. *J. Econom. Behav. Organ.* 25(3):309–327.

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

Tsai J, Rathi S, Kiekintveld C, Ordóñez F, Tambe M (2009) IRIS: A tool for strategic security allocation in transportation networks. *Proc. 8th Internat. Conf. Autonomous Agents Multiagent Systems* (*AAMAS*) (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 37–44.

Vanek O, Jakob M, Hrstka O, Pechoucek M (2011) Using multiagent simulation to improve the security of maritime transit. *Proc. 12th Internat. Workshop Multi-Agent-Based Simulation* (*MABS*) (Springer-Verlag, Berlin, Heidelberg), 44–58.

Vavasis SA (1995) *Handbook of Global Optimization* (Kluwer, Dordrecht, the Netherlands), 27–41.

von Stengel B, Zamir S (2004) Leadership with commitment to mixed strategies. Technical report LSE-CDAM-2004-01, Centre for Discrete and Applicable Mathematics, London School of Economics and Political Science, London.

wei Lye K, Wing JM (2005) Game strategies in network security. *Internat. J. Inform. Security* 4(1–2):71–86.

Wein LM (2009) Homeland security: From mathematical models to policy implementation. *Oper. Res.* 57(4):801–811.

Wilcox RR (2003) *Applying Contemporary Statistical Techniques*, 2nd ed. (Academic Press, Amsterdam).

Wright J, Leyton-Brown K (2010) Beyond equilibrium: Predicting human behavior in normal form games. *Proc. 24th AAAI Conf. Artificial Intelligence* (*AAAI*) (Palo Alto, CA), 901–907.

Yang R, Tambe M, Ordóñez F (2012) Computing optimal strategy against quantal response in security games. *Proc. 11th Internat. Conf. Autonomous Agents Multagent Systems (AAMAS)* (International Founation for Autonomous Agents and Multiagent Systems, Richland, SC), 847–854.

Yang R, Kiekintveld C, Ordóñez F, Tambe M, John R (2011) Improving resource allocation strategy against human adversaries in security games. *Proc. 22nd Internat. Joint Conf. Artificial Intelligence* (*IJCAI*) (Palo Alto, CA), 458–464.

Yin Z, Jain M, Tambe M, Ordóñez F (2011) Risk-averse strategies for security games with execution and observational uncertainty. *Proc. 25th AAAI Conf. Artificial Intelligence* (*AAAI*) (Association for the Advancement of Artificial Intelligence, Menlo Park, CA), 758–763.

Yin Z, Korzhyk D, Kiekintveld C, Conitzer V, Tambe M (2010) Stackelberg vs. Nash in security games: Interchangeability, equivalence, and uniqueness. *Proc. 9th Internat. Conf. Autonomous Agents Multiagent Systems* (*AAMAS*) (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC), 1139–1146.

Young S, Orchard D (2011) Remembering 9/11: Protecting America's ports. Accessed January 9, 2012, http://coastguard.dodlive.mil/2011/09/remembering-911-protecting-americas-ports/.

Bo An is a postdoctoral researcher at the University of Southern California. His research interests include artificial intelligence, multiagent systems, game theory, and automated negotiation. He is the recipient of the 2010 IFAAMAS Victor Lesser Distinguished Dissertation Award, Operational Excellence Award from the Commander, First Coast Guard District, and the AAMAS' 2012 Best Innovative Application Paper. He received a PhD in computer science from the University of Massachusetts, Amherst.

Fernando Ordóñez is an associate professor in the Industrial Engineering Department at the Universidad de Chile. He also has an adjunct research appointment in the Industrial and Systems Engineering Department at the University of Southern California. He received his PhD in operations research from MIT. His research focuses on convex and robust optimization, complexity of algorithms, sensitivity analysis, and applications of optimization to engineering and management science.

Milind Tambe is Helen N. and Emmett H. Jones Professor in engineering at the University of Southern California. He is a fellow of AAAI (Association for Advancement of Artificial Intelligence), recipient of the ACM/SIGART Autonomous Agents Research Award, Christopher Columbus Fellowship Foundation Homeland security award, "influential paper award" from the International Foundation for Agents and Multiagent Systems, the Rist Prize of the Military Operations Research Society, best papers at a number of premier Artificial Intelligence Conferences and workshops, IBM Faculty Award, and Okawa foundation faculty research award. The "security games" framework pioneered by Professor Tambe and his research group is now deployed for real-world use by several agencies including the U.S. Coast Guard, the U.S. Federal Air Marshals service, LAX Police and the LA Sheriff's Department for security scheduling at a variety of U.S. ports, airports, and transportation infrastructure. Recently, he cofounded ARMORWAY, a security resource optimization company.

Eric Shieh is a PhD student in the Department of Computer Science at the University of Southern California. Prior to the doctoral program, he was a software engineer at Northrop Grumman working on mission planning software. He received his master's degree in computer science from the University of Southern California in 2005 and bachelor's degree in computer science from Carnegie Mellon in 2003.

Rong Yang is a PhD candidate in the Department of Computer Science at the University of Southern California. She got her master's degree from the Department of Electrical and Computer Engineer at The Ohio State University in 2009; and her bachelor's degree from the Department of Electronic Engineering, Tsinghua University in 2007.

Craig Baldwin is a research scientist and program manager at the United States Coast Guard Research and Development Center. He manages a portfolio of research projects involving application of game theory to improve port security, analysis and modeling of Coast Guard command, control, communications, computers, intelligence, surveillance and reconnaissance (C4ISR) systems, and major systems acquisition alternatives analyses.

Joseph DiRenzo III is Coast Guard Atlantic Area's Chief of Operations Analysis. He directs a division supporting all analytical, modeling, and research work at the Coast Guard's largest operational command. He is also the command's liaison to the Department of Homeland Security University program and their Centers of Excellence. He is a retired Coast Guard officer, who spent nine years in the Navy, in both the submarine and surface warfare communities. Dr. DiRenzo is one of the most published authors in the history of the Coast Guard. A five-time winner of the service's prestigious JOC Alex Haley award, he has published more than 300 articles on various maritime terrorism, port security topics and Game Theory-related topics. Dr. DiRenzo is a graduate of the United States Naval Academy, holds a Master's in Business Administration from California Coast University, is a graduate of both the Naval War College and Marine Corps Command and Staff College completing his PhD at Northcentral University in Prescott, AZ. He is the Coast Guard Chair at the Joint Forces Staff College, in Norfolk, VA and is a graduate intelligence studies professor for American Military University.

Kathryn Moretti, USCG, is an operations research analyst assigned to Coast Guard Atlantic Area in Portsmouth, VA, and works primarily with risk assessment and mitigation. She holds a bachelor's degree in operations research and computer analysis, from the U.S. Coast Guard Academy and a master's degree in computer science, specializing in computational operations research, from the College of William and Mary.

Ben Maule, USCG, is assigned as the Operations Officer and Chief Pilot at the Coast Guard Air Station in Los Angeles. Prior to this assignment, he was assigned to Coast Guard Atlantic Area in Portsmouth, VA, as an operations research analyst working primarily with risk assessment and mitigation. He holds a bachelor's degree in mechanical engineering, from the U.S. Coast Guard Academy and a master's degree in industrial engineering from Purdue University.

Garrett Meyer, USCG, is currently assigned as the Military Aide to the Thirteenth District Commander in Seattle, Washington. His previous tours of duty include Sector St. Petersburg, Florida, the National Search and Rescue School, and Sector Boston, Massachusetts. He holds a bachelor's degree in management from the U.S. Coast Guard Academy and a master's degree in management from the University of Phoenix.