Including Human Behavior in Stackelberg Game for Security

Rong Yang[1], Christopher Kiekintveld[2],
Fernando Ordonez[1], Milind Tambe[1], and Richard John[1]

[1] University of Southern California
[2] University of Texas El Paso

Abstract. Recently, Stackelberg games have garnered significant attention given their deployment for real world security. However, a fundamental challenge of applying game-theoretic techniques to real-world security problem is the standard assumption that the adversary is perfectly rational in responding to security force's strategy, which can be unrealistic for human adversaries. Previous work has presented COBRA as a leading contender for accounting for the bounded rationality of human adversaries in security games. This paper presents an advance over this previous work by providing new algorithms based on two human behavior theories: Prospect Theory (PT) and Quantal Response Equilibrium (QRE). The paper's key contributions include: (i) efficient algorithms for computing optimal strategic solutions using PT and QRE; (ii) most comprehensive experiment to date on effectiveness of different models against human subjects; (iii) new techniques for generating representative payoff structures for behavioral experiments in generic classes of games. Our results with human subjects show that our new strategies significantly outperform COBRA.

Keywords: Human Behavior, Stackelberg Games, Decision-making

1 Introduction

Game-theoretic models have recently become important tools for analyzing real-world security resource allocation problems. These models provide a sophisticated approach for generating randomized strategies that mitigate attackers' ability to find weaknesses using surveillance. The ARMOR system at LAX airport [9] and IRIS at the Federal Air Marshals Service [13] are notable real-world deployments of this approach. One of the key sets of assumptions these systems make is about how attackers choose attack strategies based on their knowledge of the security policy. Typically, such systems have applied the standard game-theoretic assumption that attackers are perfectly rational and will strictly maximize their expected utility. This is a reasonable proxy for the worst case of a highly intelligent attacker, but it leaves open the possibility that the defender's strategy is not robust against attackers using different decision procedures, and it fails to exploit known weaknesses in the decision-making of human attackers. Indeed, it is widely accepted that standard game-theoretic assumptions of perfect rationality are not ideal for predicting the behavior of humans in multi-agent decision problems [1]. In the multi-agent systems community there is a growing interest in adopting these models to improve decisions in agents that interact with humans or to provide better decision support in systems that use multi-agent systems techniques to provide advice to human decision-makers [2,3]. Our work in this paper focuses on integrating more realistic models of human behavior into the computational analysis of security problems.

There are several challenges in moving beyond perfect rationality assumptions to integrate more realistic models of human decision-making. First, the literature has introduced a multitude of candidate models, but there is an important empirical question of which model best represents the salient features of human behavior in applied security games. Second, integrating any of the proposed models into a decision-support system (even for the purpose of empirically evaluating the model) requires developing new methods for computing solutions to security games, since the existing algorithms are based on mathematically optimal attackers [6,8]. In this context, COBRA (Combined Observability and Rationality Assumption), developed in most recent work [10] is the leading contender that accounts for human behavior in security games. Thus, the open question is whether there are other approaches that allow for fast solution and yet outperform COBRA in addressing human behaviors.

This paper addresses the challenges and answers the open questions: it develops two new methods for generating defender strategies in security games based on using two well-known models of human behavior to model the attacker's decisions. The first is *Prospect Theory* (PT), which provides a descriptive framework for decision-making under uncertainty that accounts for both risk preferences and variations in how humans interpret probabilities through a weighting function [5]. The second model is Quantal Response Equilibrium (QRE). QRE adapts ideas from the literature on discrete choice problems to a game-theoretic framework with the basic premise that humans will choose better actions more frequently, but with some noise in the decision-making process that leads to stochastic choice probabilities following a logit distribution. We develop new techniques to compute optimal defender strategies in Stackelberg security games under the assumption that the attacker will make choices according to either the PT or QRE model.

To test these new methods we performed experiments with human subjects using an online game called 'The Guard and the Treasure' designed to simulate a security scenario similar to the ARMOR program for the Los Angeles International (LAX) airport. Furthermore, we designed classification techniques to select payoff structures for experiments such that the models are well separated from each other and the payoff structures are representative of the game space. We compare these models against both a perfect rationality baseline (DOBSS) and COBRA. Our data shows that the new approaches yield statistically significantly better strategies against human attackers than previous methods including COBRA in most of the payoff structures, and comparable results in others.

2 Stackelberg Security Game

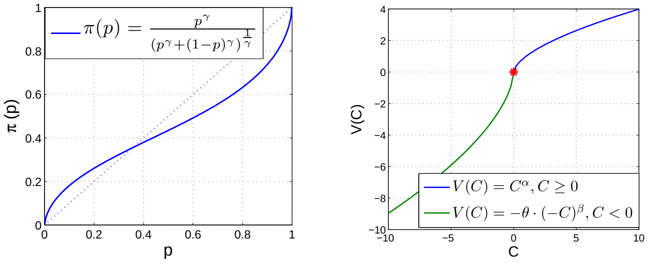
We consider a Stackelberg game with a single leader and at least one follower. The leader commits to a strategy first, taking into account the follower's response to her strategy. The followers decide their actions knowing the leader strategy. Security games refer to a special class of attacker-defender Stackelberg games, used in deployed applications mentioned earlier [9,13], where the defender plays the role of leader and an adversary plays the role of follower. In these non zero-sum games the attacker's utility of attacking a target decreases as the defender allocates more resources to protect it (and vice versa for the defender). The defender (leader) first commits to a mixed strategy, assuming the attacker (follower) decides on a pure strategy after observing the defender's strategy. This models the situation where an attacker conducts surveillance to learn the defender's mixed strategy and then launches an attack on a single target. In this work, we constrain the adversary to select a pure strategy. Given that the defender has limited resources (e.g., she may need to protect 8 targets with 3 guards), she must design her mixed strategy to optimize against the adversary's response to maximize effectiveness.

3 Related Work

Motivated by various applications, there have been many algorithms developed to compute optimal defender strategies in Stackelberg games[6,8]. One leading family of algorithms to compute such mixed strategies are DOBSS (Decomposed Optimal Bayesian Stackelberg Solver) [8] and its successors [6,9], which are used in the deployed ARMOR and IRIS applications. These algorithms formulate the problem as a Mixed Integer Linear Program (MILP), and compute an optimal mixed strategy for the defender assuming that the attacker responds optimally. However, in many real world domains, agents face human adversaries whose behavior may not be optimal under perfect rationality. Recent work [10] developed a new algorithm COBRA, which provided a solution for designing better defender strategies against human adversaries by accounting for their bounded rationality on computing the optimal strategy; and anchoring biases caused by limited observation conditions of the defender's strategy. COBRA outperforms DOBSS with statistical significance in experiments using human subjects, and represents the best available benchmark for how to determine defender strategies in security games against human adversaries.

This paper introduces alternative methods for computing strategies to play against human adversaries, based on two well-known theories from the behavioral literature, Prospect Theory (PT) and Quantal Response Equilibrium (QRE).

Prospect Theory is the subject of a Nobel Prize winning work. It provides a descriptive model of how humans make decisions among alternatives with risk.



(a) weighting function

(b) value function

Fig. 1. PT functions from Hastie et al.

The theory describes such a decision-making process as a process of maximizing the so-called ‘prospect’. Prospect is defined as $\sum_i \pi(p_i) \cdot V(C_i)$, where p_i is the probability of receiving C_i as the outcome. The weighting function $\pi(\cdot)$ describes how probability p_i is perceived by individuals.The key concepts of a weighting function are that individuals overestimate low probability and underestimate high probability [4,5], shown in Fig. 1(a). Also, $\pi(\cdot)$ is not consistent with the definition of probability in the sense that $\pi(p) + \pi(1-p) \leq 1$ in general. The value function $V(C_i)$ reflects the value of the outcome C_i. PT indicates that individuals are risk averse regarding gain but risk seeking regarding loss, implying an S-shaped value function [4,5], as shown in Fig. 1(b). A key component of Prospect Theory is the reference point. Outcomes lower than the reference point are considered as loss and higher as gain.

Quantal Response Equilibrium is an important model in behavior theory [7] and is the baseline model of many studies [12,15]. It suggests that instead of strictly maximizing utility, individuals respond stochastically in games: the chance of selecting a non-optimal strategy increases as the cost of such an error decreases. Recent work [15] shows Quantal Level-k[3] [12] to be best suited for predicting human behavior in simultaneous move games. However, the applicability of QRE and PT to security games and their comparison with COBRA remain open questions.

4 Defender Mixed-Strategy Computation

We now describe efficient computation of the optimal defender mixed strategy assuming a human adversary’s response is based on either PT or QRE.

4.1 Methods for Computing PT

Best Response to Prospect Theory **(BRPT)** is a mixed integer programming formulation for the optimal leader strategy against players whose response fol-

[3] We applied QRE instead of Quantal Level-k because in Stackelberg security games the attacker observes the defender’s strategy, so level-k reasoning is not applicable.

lows a PT model. Only the adversary is modeled using PT in this case, since the defender's actions are recommended by the decision aid. BRPT maximizes d, the defender's expected utility. The defender has a limited number of resources, $\varUpsilon$, to protect the set of targets, $t_i \in T$ for i=1..n. The defender selects a mixed strategy x that describes the probability that each target will be protected by a resource; we denote these individual probabilities by x_i. The attacker chooses a target to attack after observing x. We denote the attacker's choice using the vector of binary variables q_i, where q_i=1 if t_i is attacked and 0 otherwise.

$$
\begin{aligned}
\max_{x,q,a,d,z} \quad & d \\
\text{s.t.} \quad & \sum_{i=1}^{n}\sum_{k=1}^{5} x_{ik} \leq \varUpsilon \qquad (1) \\
& \sum_{k=1}^{5}(x_{ik} + \bar{x}_{ik}) = 1, \forall i \qquad (2) \\
& 0 \leq x_{ik}, \bar{x}_{ik} \leq c_k - c_{k-1}, \forall i, k = 1..5 \qquad (3) \\
& z_{ik} \cdot (c_k - c_{k-1}) \leq x_{ik}, \forall i, k = 1..4 \qquad (4) \\
& \bar{z}_{ik} \cdot (c_k - c_{k-1}) \leq \bar{x}_{ik}, \forall i, k = 1..4 \qquad (5) \\
& x_{i(k+1)} \leq z_{ik}, \forall i, k = 1..4 \qquad (6) \\
& \bar{x}_{i(k+1)} \leq \bar{z}_{ik}, \forall i, k = 1..4 \qquad (7) \\
& z_{ik}, \bar{z}_{ik} \in \{0, 1\}, \forall i, k = 1..4 \qquad (8) \\
& x'_i = \sum_{k=1}^{5} b_k x_{ik},\ \bar{x}'_i = \sum_{k=1}^{5} b_k \bar{x}_{ik}, \forall i \qquad (9) \\
& \sum_{i=1}^{n} q_i = 1,\ q_i \in \{0, 1\}, \forall i \qquad (10) \\
& 0 \leq a - (x'_i (P_i^a)' + \bar{x}'_i (R_i^a)') \leq M(1 - q_i), \forall i \qquad (11) \\
& M(1 - q_i) + \sum_{k=1}^{5}(x_{ik} R_i^d + \bar{x}_{ik} P_i^d) \geq d, \forall i \qquad (12)
\end{aligned}
$$

The defender optimization problem is given in Equations (1)-(12). In security games, the payoffs depend only on whether or not the attack was successful, so given a target t_i, the defender (resp., adversary) receives reward R_i^d (penalty P_i^a) if the adversary attacks the target and it is covered by the defender; otherwise, the defender (adversary) receives penalty P_i^d (reward R_i^a).

PT comes into the algorithm by adjusting the weighting and value functions as described above. The benefit (prospect) perceived by the adversary for attacking target t_i if the defender plays the mixed strategy x is given by $\pi(x_i)V(P_i^a) + \pi(1 - x_i)V(R_i^a)$. Let $(P_i^a)' = V(P_i^a)$ and $(R_i^a)' = V(R_i^a)$ denote the adversary's value of penalty P_i^a and reward R_i^a. We use a piecewise linear function $\tilde{\pi}(\cdot)$ to approximate the non-linear weighting function $\pi(\cdot)$ and empirically set 5 segments[4] for $\tilde{\pi}(\cdot)$. This function is defined by $\{c_k | c_0 = 0, c_5 = 1, c_k < c_{k+1}, k = 0, ..., 5\}$ that represent the endpoints of the linear segments and $\{b_k | k = 1, \ldots, 5\}$ that represent the slope of each linear segment. Thus, each of the defender's $x_i {=} \sum_{k=1}^{5} x_{ik}$; the follower will perceive this x_i as $x'_i {=} \pi(x_i) {=} \sum_{k=1}^{5} b_k {\cdot} x_{ik}$ as discussed below.

In order to represent the piecewise linear function, we break x_i (and $1 - x_i$) into five segments, denoted by variable x_{ik} (and $\bar{x}_{ik}$). We can enforce that such breakup of x_i (and $1 - x_i$) is correct if segment x_{ik} (and $\bar{x}_{ik}$) is positive only if the previous segment is used completely, for which we need the auxiliary integer variable z_{ik} (and $\bar{z}_{ik}$). This is enforced by Equations (3)$\sim$(8). Equation (9) defines x'_i and $\bar{x}'_i$ as the value of the piecewise linear approximation of x_i and $1 - x_i$: $x'_i {=} \tilde{\pi}(x_i)$ and $\bar{x}'_i {=} \tilde{\pi}(1 - x_i)$. Equations (10) and (11) define the optimal adversary's pure strategy. In particular, Equation (11) enforces that $q_i{=}1$ for the action that achieves maximal prospect for the adversary. Equation (12) enforces that d is the defender's expected utility on the target that is attacked by the adversary ($q_i{=}1$).

Robust-PT (RPT) modifies the base BRPT method to account for some uncertainty about the adversaries choice, caused (for example) by imprecise computations [11]. Similar to COBRA, RPT assumes that the adversary may choose any strategy within ϵ of the best choice, defined here by the prospect of each action. It optimizes the worst-case outcome for the defender among the set of strategies that have prospect for the attacker within ϵ of the optimal prospect. We modify the BRPT optimization problem as follows: the first 11 Equations are equivalent to those in BRPT; in Equation (13), the binary variable h_i indicates all the $\epsilon-$optimal strategies for the adversary; Equation (16) enforces that d is the minimum expected utility of the defender against the $\epsilon-$optimal strategies of the adversary.

$$\max_{x,h,q,a,d,z} d$$

$$\text{s.t. Equations (1)}\sim\text{(11)}$$

$$\sum_{i=1}^{n} h_i \geq 1 \tag{13}$$

$$h_i \in \{0,1\},\ \ q_i \leq h_i, \forall i \tag{14}$$

$$\epsilon(1 - h_i) \leq a - (x'_i(P_i^a)' + \bar{x}'_i(R_i^a)') \leq M(1 - h_i), \forall i \tag{15}$$

$$M(1 - h_i) + \sum_{k=1}^{5}(x_{ik}R_i^d + \bar{x}_{ik}P_i^d) \geq d, \forall i \tag{16}$$

Runtime: We choose AMPL (http://www.ampl.com/) to solve the MILP with CPLEX as the solver. Both BRPT and RPT take less than 1 second for up to 10 targets.

[4] This piecewise linear representation of $\pi(\cdot)$ can achieve a small approximation error: $\sup_{z\in[0,1]} \|\pi(z) - \tilde{\pi}(z)\| \leq 0.03$.

4.2 Methods for Computing QRE

In applying the QRE model to our domain, we only add noise to the response function for the adversary, so the defender computes an optimal strategy assuming the attacker response with a noisy best-response. The parameter λ represents the amount of noise in the attacker's response. Given λ and the defender's mixed-strategy x, the adversaries' quantal response q_i (i.e. probability of i) can be written as

$$q_i = \frac{e^{\lambda U_i^a(x)}}{\sum_{j=1}^{n} e^{\lambda U_j^a(x)}} \tag{17}$$

where, $U_i^a(x) = x_i P_i^a + (1-x_i) R_i^a$ is the adversary's expected utility for attacking t_i and x is the defender's strategy.

$$q_i = \frac{e^{\lambda R_i^a} e^{-\lambda(R_i^a - P_i^a)x_i}}{\sum_{j=1}^{n} e^{\lambda R_j^a} e^{-\lambda(R_j^a - P_j^a)x_j}} \tag{18}$$

The goal is to maximize the defender's expected utility given q_i, i.e. $\sum_{i=1}^{n} q_i(x_i R_i^d + (1-x_i)P_i^d)$. Combined with Equation (18), the problem of finding the optimal mixed strategy for the defender can be formulated as

$$\begin{aligned} \max_x \ & \frac{\sum_{i=1}^{n} e^{\lambda R_i^a} e^{-\lambda(R_i^a - P_i^a)x_i}((R_i^d - P_i^d)x_i + P_i^d)}{\sum_{j=1}^{n} e^{\lambda R_j^a} e^{-\lambda(R_j^a - P_j^a)x_j}} \qquad (19) \\ \text{s.t.} \ & \sum_{i=1}^{n} x_i \le \Upsilon \\ & 0 \le x_i \le 1, \quad \forall i, j \end{aligned}$$

Given that the objective function in Equation (19) is non-linear and non-convex in its most general form, finding the global optimum is extremely difficult. Therefore, we focus on methods to find local optima. To compute an approximately optimal QRE strategy efficiently, we develop the Best Response to Quantal Response (**BRQR**) heuristic described in Algorithm 1. We first take the negative of Equation (19), converting the maximization problem to a minimization problem. In each iteration, we find the local minimum[5] using a gradient descent technique from the given starting point. If there are multiple local minima, by randomly setting the starting point in each iteration, the algorithm will reach different local minima with a non-zero probability. By increasing the iteration number, $IterN$, the probability of reaching the global minimum increases.

Parameter Estimation: The parameter λ in the QRE model represents the amount of noise in the best-response function. One extreme case is λ=0, when play becomes uniformly random. The other extreme case is λ=∞, when the quantal response is identical to the best response. λ is sensitive to game payoff structure, so tuning λ is a crucial step in applying the QRE model. We employed Maximum Likelihood Estimation (MLE) to fit λ using data from [10].

[5] We use *fmincon* function in Matlab to find the local minimum.

Algorithm 1 BRQR

```
$opt_g \leftarrow -\infty$;                              ▷ Initialize the global optimum
for $i \leftarrow 1, ..., IterN$ do
    $x_0 \leftarrow$ randomly generate a feasible starting point
    $(opt_l, x^*) \leftarrow$ FindLocalMinimum($x_0$)
    if $opt_g > opt_l$ then
        $opt_g \leftarrow opt_l$, $x_{opt} \leftarrow x^*$
    end if
end for
return $opt_g, x_{opt}$
```

Given the defender's mixed strategy x and N samples of the players' choices, the logit likelihood of λ is

$$\log L(\lambda \mid x) = \sum_{j=1}^{N} \log q_{\tau(j)}(\lambda)$$

where $\tau(j)$ denotes the target attacked by the player in sample j. Let N_i be the number of subjects attacking target i. Then, we have $\log L(\lambda \mid x) = \sum_{i=1}^{n} N_i \log q_i(\lambda)$. Combining with Equation (17),

$$\log L(\lambda \mid x) = \lambda \sum_{i=1}^{n} N_i U_i^a(x) - N \cdot \log(\sum_{i=1}^{n} e^{\lambda U_i^a(x)})$$

$\log L(\lambda \mid x)$ is a concave function[6]. Therefore, $\log L(\lambda \mid x)$ only has one local maximum. The MLE of λ is 0.76 for the data used from [10].

Runtime: We implement BRQR in Matlab. With 10 targets and $IterN$=300, the runtime of BRQR is less than 1 minute. In comparison, with only 4 targets, LINGO12 (http://www.lindo.com/) cannot compute the global optimum of Equation (19) within one hour.

5 Payoff Structure Classification

One important property of payoff structures we want to examine is their influence on model performance. We certainly cannot test over all possible payoff structures, so the challenges are: (i) the payoff structures we select should be representative of the payoff structure space; (ii) the strategies generated from different algorithms should be sufficiently separated. As we will discuss later, the payoff structures used in [10] do not address these challenges.

[6] The second order derivative of $\log L(\lambda \mid x)$ is

$$\frac{d^2 \log L}{d\lambda^2} = \frac{\sum_{i<j} -(U_i^a(x) - U_j^a(x))^2 e^{\lambda(U_i^a(x) + U_j^a(x))}}{(\sum_i e^{\lambda U_i^a(x)})^2} < 0$$

We address the first criterion by randomly sampling 1000 payoff structures, each with 8 targets. R_i^a and R_i^d are integers drawn from $Z^+[1,10]$; P_i^a and P_i^d are integers drawn from $Z^-[-10,-1]$. This scale is similar to the payoff structures used in [10]. We then clustered the 1000 payoff structures into four clusters using k-means clustering based on eight features, which are defined in Table 1. Intuitively, features 1 and 2 describe how 'good' the game is for the adversary, features 3 and 4 describe how 'good' the game is for the defender, and features 5∼8 reflect the level of 'conflict' between the two players in the sense that they measure the ratio of one player's gain over the other player's loss. In Fig. 2, all 1000 payoff structures are projected onto the first two Principal Component Analysis (PCA) dimensions for visualization. We select one payoff structure from each cluster, following the criteria below to obtain sufficiently different strategies for the different candidate algorithms:

Table 1. A-priori defined features

Feature 1	Feature 2	Feature 3	Feature 4
mean($\lvert\frac{R_i^a}{P_i^a}\rvert$)	std($\lvert\frac{R_i^a}{P_i^a}\rvert$)	mean($\lvert\frac{R_i^d}{P_i^d}\rvert$)	std($\lvert\frac{R_i^d}{P_i^d}\rvert$)
Feature 5	Feature 6	Feature 7	Feature 8
mean($\lvert\frac{R_i^a}{P_i^d}\rvert$)	std($\lvert\frac{R_i^a}{P_i^d}\rvert$)	mean($\lvert\frac{R_i^d}{P_i^a}\rvert$)	std($\lvert\frac{R_i^d}{P_i^a}\rvert$)

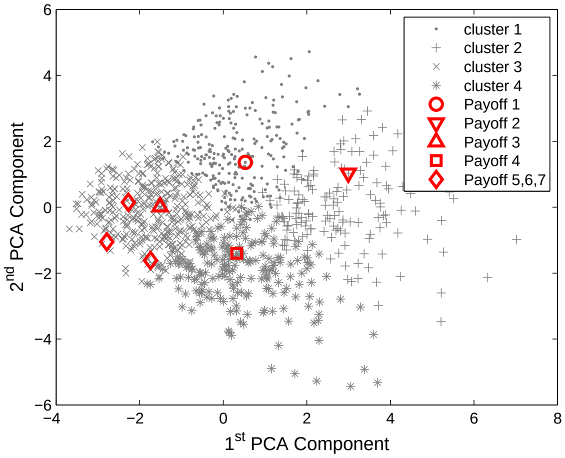


Fig. 2. Payoff Structure Clusters (color)

– We define the distance between two mixed strategies, x^k and x^l, using the Kullback-Leibler divergence:

$$D(x^k, x^l) = D_{KL}(x^k|x^l) + D_{KL}(x^l|x^k)$$

where, $D_{KL}(x^k|x^l) = \sum_{i=1}^{n} x_i^k \log(x_i^k/x_i^l)$.

- For each payoff structure, $D(x^k, x^l)$ is measured for every pair of strategies. With five strategies (discussed later), we have 10 such measurements.
- We remove payoff structures that have a mean or minimum of these 10 quantities below a given threshold. This gives us a subset of about 250 payoff structures in each cluster. We then select one payoff structure closest to the cluster center from the subset of each cluster .

The four payoff structures (payoffs 1-4) we selected from each cluster are marked in Fig. 2, as are the three (payoffs 5-7) used in [10]. Fig. 2 shows that payoffs 5-7 all belong to cluster 3. Furthermore, Table 2 reports the strategy distances in all seven payoff structures. The strategies are not as well separated in payoffs 5-7 as they are in payoffs 1-4. As we discuss in Section. 6.2, the performance of different strategies is quite similar in payoffs 5-7.

Table 2. Strategy Distance

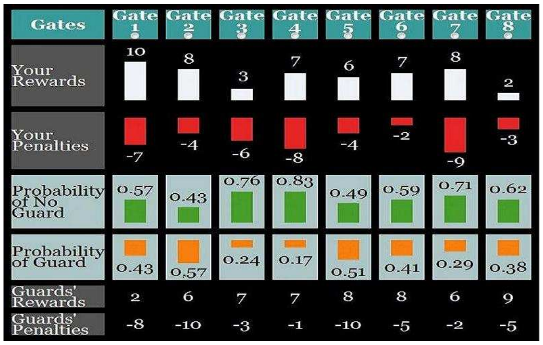
Payoff Structure	1	2	3	4	5	6	7
mean D_{KL}	0.83	1.19	0.64	0.88	0.32	0.15	0.12
min D_{KL}	0.26	0.25	0.21	0.25	0.07	0.02	0.04

6 Experiments

We conducted empirical tests with human subjects playing an online game to evaluate the performances of leader strategies generated by five candidate algorithms. We based our model on the LAX airport, which has eight terminals that can be targeted in an attack [9]. Subjects play the role of followers and are able to observe the leader's mixed strategy (i.e., randomized allocation of security resources).

6.1 Experimental Setup

Fig. 3 shows the interface of the web-based game we developed to present subject with choice problems. Players were introduced to the game through a series of explanatory screens describing how the game is played. In each game instance a subject was asked to choose one of the eight gates to open (attack). They knew that guards were protecting three of the eight gates, but not which ones. Subjects were rewarded based on the reward/penalty shown for each gate and the probability that a guard was behind the gate (i.e., the exact randomized strategy of the defender). To motivate the subjects they would earn or lose money based on whether or not they succeed in attacking a gate; if the subject opened a gate not protected by the guards, they won; otherwise, they lost. Subjects start with an endowment of \$8 and each point won or lost in a game instance was worth \$0.1. On average, subjects earned about \$14.1 in cash.



Gates	Gate 1	Gate 2	Gate 3	Gate 4	Gate 5	Gate 6	Gate 7	Gate 8
Your Rewards	10	8	3	7	6	7	8	2
Your Penalties	-7	-4	-6	-8	-4	-2	-9	-3
Probability of No Guard	0.57	0.43	0.76	0.83	0.49	0.59	0.71	0.62
Probability of Guard	0.43	0.57	0.24	0.17	0.51	0.41	0.29	0.38
Guards' Rewards	2	6	7	7	8	8	6	9
Guards' Penalties	-8	-10	-3	-1	-10	-5	-2	-5

Fig. 3. Game Interface

We tested the seven different payoff structures from Fig. 2 (four new, three from [10]). The seven payoff structures are displayed in Table 4. For each payoff structure we tested the mixed strategies generated by five algorithms: BRPT, RPT, BRQR, COBRA and DOBSS, which are reported in Table 5 and Table 6. There were a total of 35 payoff structure/strategy combinations and each subject played all 35 combinations. In order to mitigate the order effect on subject responses, a total of 35 different orderings of the 35 combinations were generated using Latin Square design. Every ordering contained each of the 35 combinations exactly once, and each combination appeared exactly once in each of the 35 positions across all 35 orderings. The order played by each subject was drawn uniformly randomly from the 35 possible orderings. To further mitigate learning, no feedback on success or failure was given to the subjects until the end of the experiment. A total of 40 human subjects played the game.

We could explore only a limited number of parameters for each algorithm, which were selected following the best available information in the literature. The parameter settings for each algorithm are reported in Table 3. DOBSS has no parameters. The values of PT parameters are typical values reported in the literature [4]. We set ϵ in RPT following two rules: (i) No more than half of targets are in the $\epsilon-$optimal set; (ii) $\epsilon \leq 0.3R^a_{max}$, where R^a_{max} is the maximum potential reward for the adversary. The size of the $\epsilon-$optimal set increases as the value of ϵ increases. When ϵ is sufficiently large, the defender's strategy becomes maximin, since she believes that the adversary may attack any target. The second rule limits the imprecision in the attacker's choice. We empirically set the limit to $0.3R^a_{max}$. For BRQR, we set λ using MLE with data reported

Table 3. Model Parameter

Payoff Structure	1	2	3	4	5	6	7
RPT-ϵ	2.4	3.0	2.1	2.75	1.9	1.5	1.5
COBRA-α	0.15	0.15	0.15	0.15	0.37	0	0.25
COBRA-ϵ	2.5	2.9	2.0	2.75	2.5	2.5	2.5

in [10] (see Section 4.2). For payoffs 1∼4, we set the parameters for COBRA following the advices given by [10] as close as possible. In particular, the values we set for α meet the entropy heuristic discussed in that work. For payoffs 5∼7, we use the same parameter settings as in their work.

6.2 Experiment Result

Quality Comparison: We used the average expected defender's utility to evaluate the performances of the strategies. Let $U_i^d(x) = x_i R_i^d + (1 - x_i) P_i^d$ be the defender's expected utility for target t_i if she plays mixed strategy x and the subject selects target t_i; N_i be the number of subjects that chose target t_i. Then, the average expected defender's utility can is calculated as

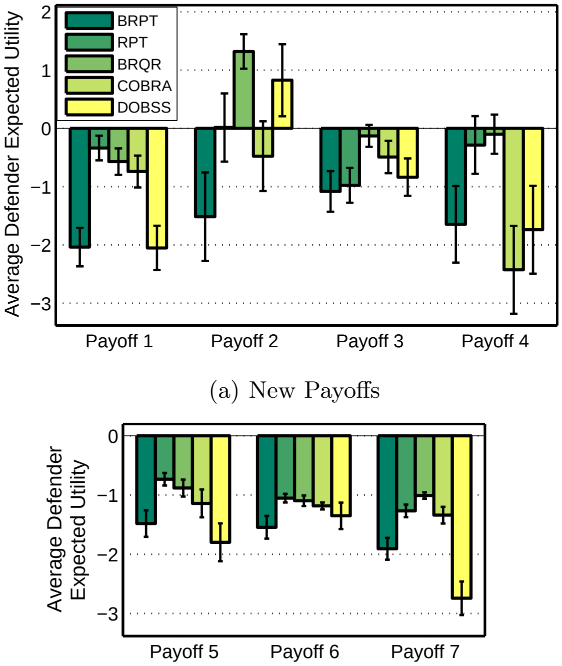
$$\bar{U}_{exp}^d(x) = \frac{1}{N} \sum_{i=1}^{n} N_i U_i^d(x)$$

Fig. 4 displays $\bar{U}_{exp}^d(x)$ for the different strategies in each payoff structure. 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 (see Table 2). We evaluate the statistical significance of our results using the bootstrap-t method [14]. The comparison is summarized below:

- 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 ($p<0.02$).
- 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 ($p<0.005$). They have very 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 ($p<0.06$). BRPT's failure to perform better (and even worse than COBRA) is a surprising outcome.

Overall, BRQR performs best, RPT outperforms COBRA in six of the seven cases, and BRPT and DOBSS perform the worst.

Key Observations: BRPT and DOBSS are not robust against an adversary that deviates from the optimal strategy. BRQR, RPT and COBRA all try to be robust against such deviations. BRQR considers some (possibly very small) probability of adversary attacking any target. In contrast, COBRA and RPT separate the targets into two groups, the ϵ-optimal set and the non-ϵ-optimal set, using a hard threshold. They then try to maximize the worst case for the



(a) New Payoffs

(b) Payoffs from Pita et al.

Fig. 4. Average Expected Utility of Defender

defender assuming the response will be in the ϵ-optimal set, but assign less resources to other targets. When the non-ϵ-optimal targets have high defender penalties, COBRA and RPT become vulnerable, especially in the following two cases:

- 'Unattractive' targets are those with small reward but large penalty for the adversary. COBRA and RPT consider such targets as non-ϵ-optimal and assign significantly less resources than BRQR on them. However, some subjects would still select such targets and caused severe damage to COBRA and RPT (e.g. about 30% subjects selected door 5 in payoff 4 against COBRA strategy, as shown in Fig. 5(d)).
- 'High-risk' targets are those with large reward and large penalty for the adversary. RPT considers such targets as non-ϵ-optimal and assigns far less resources than other algorithms. This is caused by the assumptions made by PT that people care more about loss than gain and that they overestimate small probabilities. However, experiments show RPT gets hurt significantly on such targets (e.g. more than 15% subjects select door 1 in payoff 2, as shown in Fig. 5(b)).

7 Conclusions

There is a significant interest in game-theoretic techniques to solve security problems. However, current algorithms make perfect rationality assumption of the adversaries, which is problematic in many real security domains when agents face human adversaries. New methods need to be developed to compute defender strategy against real human adversaries. This paper successfully integrates two important human behavior theories, PT and QRE, into building more realistic decision-support tool. To that end, the main contributions of this paper are, (i) Developing efficient new algorithms based on PT and QRE models of human behavior; (ii) Conducting the most comprehensive experiments to date with human subjects for security games (40 subjects, 5 strategies, 7 game structures); (iii) Designing techniques for generating representative payoff structures for behavioral experiments in generic classes of games. By providing new algorithms that outperform the leading competitor, this paper has advanced the state-of-the-art.

References

1. C. F. Camerer, T. Ho, and J. Chongn. A congnitive hierarchy model of games. *QJE*, 119(3):861–898, 2004.
2. S. Ficici and A. Pfeffer. Simultaneously modeling humans' preferences and their beliefs about others' preferences. *AAMAS*, 2008.
3. Y. Gal and A. Pfeffer. Modeling reciprocal behavior in human bilateral negotiation. *AAMAS*, 2007.
4. R. Hastie and R. M. Dawes. *Rational Choice in an Uncertain World: the Psychology of Judgement and Decision Making.* Sage Publications, Thounds Oaks, 2001.
5. D. Kahneman and A. Tvesky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292, 1979.
6. C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordonez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. *In AAMAS*, 2009.
7. R. D. McKelvey and T. R. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 2:6–38, 1995.
8. P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus. Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. *In AAMAS*, 2008.
9. J. Pita, M. Jain, F. Ordonez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. Deployed armor protection: The application of a game theoretic model for security at the los angeles international airport. *In AAMAS*, 2008.
10. J. Pita, M. Jain, F. Ordonez, M. Tambe, and S. Kraus. Solving stackelberg games in the real-world: Addressing bounded rationality and limited observations in human preference models. *Artificial Intelligence Journal*, 174(15):1142–1171, 2010.
11. H. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–138, 1956.
12. D. O. Stahl and P. W. Wilson. Experimental evidence on players' models of other players. *JEBO*, 25(3):309–327, 1994.

13. J. Tsai, S. Rathi, C. Kiekintveld, F. Ordonez, and M. Tambe. Iris - a tool for strategic security allocation in transportation networks. *In AAMAS*, 2009.

14. R. R. Wilcox. *Applying contemporary statistical techniques*. Academic Press, 2003.

15. J. R. Wright and K. Leyton-Brown. Beyond equilibrium: Predicting human behavior in normal-form games. *In AAAI*, 2010.

A Payoff Structure Information

The four payoff structures selected from the four clustering groups are displayed in Table 4.

Table 4. Payoff Structure

(a) Payoff Structure 1

Target	1	2	3	4	5	6	7	8
defender reward	2	6	7	7	8	8	6	9
defender penalty	-8	-10	-3	-1	-10	-5	-2	-5
subject reward	10	8	3	7	6	7	8	2
subject penlaty	-7	-4	-6	-8	-4	-2	-9	-3

(b) Payoff Structure 2

Target	1	2	3	4	5	6	7	8
defender reward	3	8	9	9	7	7	4	1
defender penalty	-10	-2	-5	-1	-7	-6	-2	-1
subject reward	9	8	2	9	10	1	10	1
subject penlaty	-10	-1	-10	-8	-4	-10	-5	-3

(c) Payoff Structure 3

Target	1	2	3	4	5	6	7	8
defender reward	5	3	8	3	3	4	3	6
defender penalty	-2	-5	-4	-6	-3	-10	-7	-2
subject reward	8	6	1	3	1	7	3	5
subject penlaty	-6	-9	-3	-7	-7	-2	-5	-2

(d) Payoff Structure 4

Target	1	2	3	4	5	6	7	8
defender reward	5	9	10	2	10	4	8	8
defender penalty	-10	-4	-9	-3	-10	-10	-2	-5
subject reward	3	7	3	9	2	9	7	8
subject penlaty	-4	-8	-5	-8	-9	-4	-1	-6

(e) Payoff Structure 5

Target	1	2	3	4	5	6	7	8
defender reward	1	4	2	3	4	1	5	2
defender penalty	-5	-8	-1	-6	-5	-1	-7	-7
subject reward	1	9	5	6	7	1	10	3
subject penlaty	-2	-4	-3	-3	-3	-2	-4	-3

(f) Payoff Structure 6

Target	1	2	3	4	5	6	7	8
defender reward	4	3	1	5	1	2	5	2
defender penalty	-8	-10	-1	-8	-1	-3	-11	-5
subject reward	8	5	3	10	1	3	9	4
subject penlaty	-3	-2	-3	-2	-3	-3	-2	-3

(g) Payoff Structure 7

Target	1	2	3	4	5	6	7	8
defender reward	4	3	1	5	1	2	5	2
defender penalty	-8	-5	-1	-10	-5	-3	-9	-6
subject reward	8	5	2	10	1	3	9	4
subject penlaty	-3	-3	-3	-3	-3	-3	-3	-3

B Defender Mixed-Strategy

The defender's mixed-strategy from each algorithm in each payoff structures are displayed in Table 5 and Table 6.

Table 5. Defender's Mixed-strategy

(a) Payoff Structure 1

Target	1	2	3	4	5	6	7	8
BRPT	0.39	0.51	0.17	0.26	0.43	0.70	0.26	0.28
RPT	0.43	0.57	0.24	0.17	0.51	0.41	0.29	0.38
BRQR	0.57	0.58	0.18	0.21	0.51	0.47	0.30	0.18
COBRA	0.57	0.62	0.18	0.22	0.51	0.44	0.34	0.11
DOBSS	0.49	0.53	0.15	0.36	0.44	0.59	0.37	0.07

(b) Payoff Structure 2

Target	1	2	3	4	5	6	7	8
BRPT	0.28	0.93	0.07	0.34	0.59	0.05	0.52	0.23
RPT	0.32	0.54	0.10	0.39	0.65	0.07	0.57	0.37
BRQR	0.54	0.52	0.21	0.36	0.64	0.16	0.58	0.00
COBRA	0.48	0.53	0.09	0.43	0.74	0.00	0.70	0.02
DOBSS	0.42	0.78	0.08	0.47	0.64	0.00	0.60	0.00

(c) Payoff Structure 3

Target	1	2	3	4	5	6	7	8
BRPT	0.42	0.24	0.29	0.19	0.09	0.78	0.28	0.72
RPT	0.46	0.27	0.38	0.23	0.12	0.80	0.34	0.40
BRQR	0.36	0.43	0.20	0.36	0.13	0.72	0.43	0.37
COBRA	0.48	0.42	0.16	0.29	0.07	0.81	0.36	0.42
DOBSS	0.53	0.37	0.12	0.25	0.06	0.72	0.31	0.64

(d) Payoff Structure 4

Target	1	2	3	4	5	6	7	8
BRPT	0.28	0.27	0.22	0.33	0.08	0.54	0.90	0.38
RPT	0.37	0.32	0.30	0.37	0.10	0.61	0.49	0.44
BRQR	0.35	0.33	0.30	0.44	0.20	0.62	0.36	0.42
COBRA	0.24	0.42	0.21	0.50	0.04	0.66	0.39	0.53
DOBSS	0.22	0.37	0.19	0.44	0.05	0.58	0.69	0.47

C Histogram of Subjects' Choices

The histograms of subjects' choice in each game instance are displayed in Fig. 5 and Fig. 6.

Table 6. Defender's Mixed-strategy

(a) Payoff Structure 5

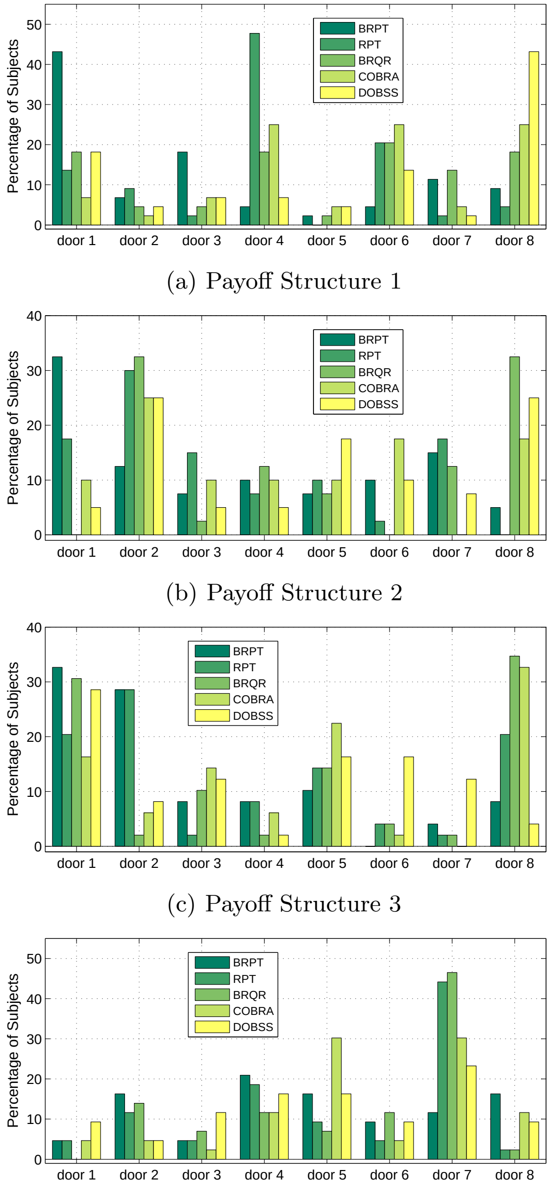
Target	1	2	3	4	5	6	7	8
BRPT	0.16	0.49	0.41	0.46	0.51	0.16	0.52	0.28
RPT	0.23	0.52	0.17	0.50	0.50	0.23	0.54	0.32
BRQR	0.12	0.61	0.16	0.55	0.52	0.00	0.57	0.46
COBRA	0.00	0.64	0.23	0.63	0.52	0.00	0.55	0.40
DOBSS	0.00	0.59	0.45	0.51	0.56	0.00	0.62	0.27

(b) Payoff Structure 6

Target	1	2	3	4	5	6	7	8
BRPT	0.49	0.46	0.21	0.67	0.05	0.21	0.64	0.29
RPT	0.54	0.53	0.07	0.55	0.07	0.27	0.63	0.35
BRQR	0.58	0.59	0.00	0.60	0.00	0.19	0.66	0.38
COBRA	0.58	0.55	0.00	0.53	0.00	0.31	0.62	0.41
DOBSS	0.56	0.45	0.19	0.68	0.00	0.19	0.65	0.30

(c) Payoff Structure 7

Target	1	2	3	4	5	6	7	8
BRPT	0.54	0.41	0.19	0.60	0.09	0.27	0.57	0.35
RPT	0.56	0.43	0.03	0.60	0.12	0.31	0.58	0.38
BRQR	0.59	0.44	0.00	0.63	0.08	0.22	0.60	0.45
COBRA	0.57	0.48	0.00	0.59	0.00	0.33	0.56	0.47
DOBSS	0.59	0.44	0.10	0.65	0.00	0.25	0.62	0.36



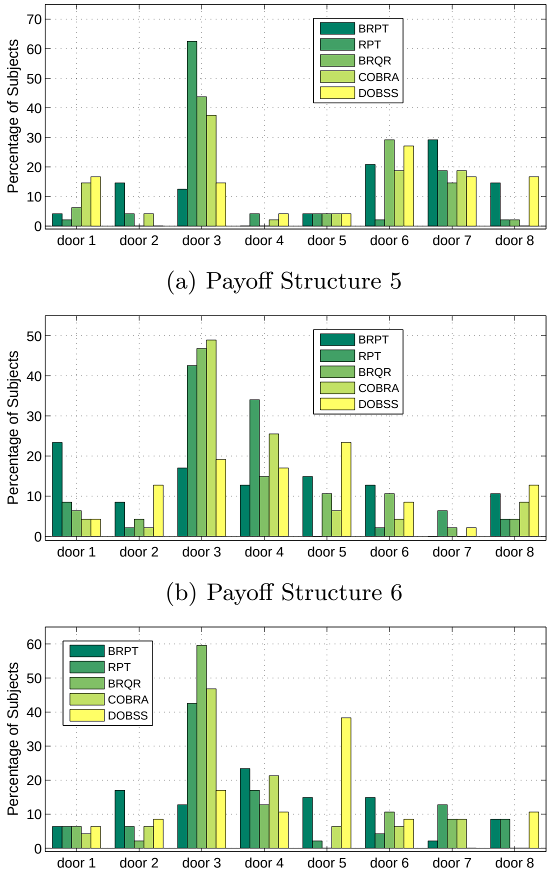
(a) Payoff Structure 1

(b) Payoff Structure 2

(c) Payoff Structure 3

(d) Payoff Structure 4

Fig. 5. Histogram of Subjects' Choices



(a) Payoff Structure 5

(b) Payoff Structure 6

(c) Payoff Structure 7

Fig. 6. Histogram of Subjects' Choices