Improving Resource Allocation Strategy Against Human Adversaries in Security Games

Rong Yang, Christopher Kiekintveld*, Fernando Ordonez, Milind Tambe, Richard John
University of Southern California, Los Angeles, CA
* University of Texas El Paso, El Paso, TX
{yangrong,fordon,tambe,richardj}@usc.edu
*ckiekint@gmail.com

Abstract

Recent real-world deployments of Stackelberg security games make it critical that we address human adversaries' bounded rationality in computing optimal strategies. To that end, this paper provides three key contributions: (i) new efficient algorithms for computing optimal strategic solutions using Prospect Theory and Quantal Response Equilibrium; (ii) the most comprehensive experiment to date studying the effectiveness of different models against human subjects for security games; and (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 techniques outperform the leading contender for modeling human behavior in security games.

1 Introduction

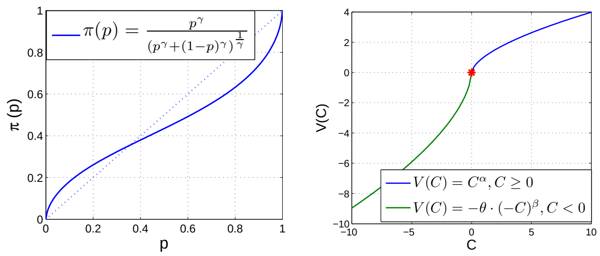
Recent real-world deployments of attacker-defender Stackelberg games, including ARMOR at the LAX airport [Pita *et al.*, 2008] and IRIS at the Federal Air Marshals Service [Tsai *et al.*, 2009], have led to an increasing interest in building decision-support tools for real-world security problems. One of the key sets of assumptions these systems make is about how attackers choose strategies based on their knowledge of the security strategy. Typically, such systems apply the standard game-theoretic assumption that attackers are perfectly rational and strictly maximize their expected utility. This is a reasonable proxy for the worst case of a highly intelligent attacker, but it can lead to a defense strategy that 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 [Camerer *et al.*, 2004].

Thus, integrating more realistic models of human decision-making has become necessary in solving real-world security problems. However, there are several open questions in moving beyond perfect rationality assumptions. 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 contexts. Second, integrating any of the proposed models into a decision-support system (even for the purpose of empirically evaluating the model) requires developing new computational methods, since the existing algorithms for security games are based on mathematically optimal attackers [Pita *et al.*, 2008; Kiekintveld *et al.*, 2009]. The current leading contender that accounts for human behavior in security games is COBRA [Pita *et al.*, 2010], which assumes that adversaries can deviate to $\epsilon-$optimal strategies and that they have an anchoring bias when interpreting a probability distribution. It remains an open question whether other models yield better solutions than COBRA against human adversaries.

We address these open questions by developing three new algorithms to generate defender strategies in security games, based on using two fundamental theories of human behavior to predict an attacker's decisions: Prospect Theory [Kahneman and Tvesky, 1979] and Quantal Response Equilibrium [McKelvey and Palfrey, 1995]. We evaluate our new algorithms using experimental data from human subjects gathered using an online game designed to simulate a security scenario similar to the one analyzed by ARMOR for the LAX airport. Furthermore, we designed classification techniques to select payoff structures for experiments such that the structures are representative of the space of possible games, improving the coverage relative to previous experiments for COBRA. Our results show that our new algorithms outperform both COBRA and a perfect rationality baseline.

2 Background and Related Work

Security games refer to a special class of attacker-defender Stackelberg games, including those used in ARMOR and IRIS [Pita *et al.*, 2008; Tsai *et al.*, 2009]. The defender needs to allocate limited security resources to protect infrastructure from an adversary's attack. In this paper, we will use a more compact representation of defender's strategy: the probability that each target will be protected by a security force, which will be introduced in Section 3.1. In Stackelberg security games, 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



(a) weighting function (b) value function

Figure 1: PT functions [Hastie and Dawes, 2001]

single target. 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). 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 strategy to optimize against the adversary's response to maximize effectiveness.

One leading family of algorithms to compute such mixed strategies are DOBSS and its successors [Pita *et al.*, 2008; Kiekintveld *et al.*, 2009], 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 assuming perfect rationality. COBRA [Pita *et al.*, 2010] represents the best available benchmark for how to determine defender strategies in security games against human adversaries, and it outperforms DOBSS with statistical significance in experiments using human subjects.

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 a nobel-prize-winning theory [Kahneman and Tvesky, 1979], which describes human decision making as a process of maximizing 'prospect'. Prospect is defined as $\sum_i \pi(p_i)V(C_i)$, where p_i is the actual probability of outcome C_i. The weighting function $\pi(p_i)$ describes how probability p_i is perceived. $\pi(\cdot)$ is not consistent with the definition of probability, i.e. $\pi(p) + \pi(1-p) \leq 1$ in general. An empirical form of $\pi(\cdot)$ is shown in Fig. 1(a). The value function $V(C_i)$ reflects the value of outcome C_i. PT indicates that individuals are risk averse regarding gain but risk seeking regarding loss, and care more about loss than gain, as shown in Fig. 1(b) [Hastie and Dawes, 2001].

Quantal Response Equilibrium is an important model in behavioral game theory [McKelvey and Palfrey, 1995]. 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 [Wright and Leyton-Brown, 2010] shows Quantal Level-k[1] [Stahl and Wilson, 1994] 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.

3 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.

3.1 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 follows a PT model. Only the adversary is modeled using PT in this case, since the defender's actions are recommended by the decision aid.

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

$$\text{s.t.} \quad \sum_{i=1}^{n}\sum_{k=1}^{5} x_{ik} \leq \Upsilon \tag{1}$$

$$\sum_{k=1}^{5}(x_{ik} + \bar{x}_{ik}) = 1, \forall i \tag{2}$$

$$0 \leq x_{ik}, \bar{x}_{ik} \leq c_k - c_{k-1}, \forall i, k = 1..5 \tag{3}$$

$$z_{ik} \cdot (c_k - c_{k-1}) \leq x_{ik}, \forall i, k = 1..4 \tag{4}$$

$$\bar{z}_{ik} \cdot (c_k - c_{k-1}) \leq \bar{x}_{ik}, \forall i, k = 1..4 \tag{5}$$

$$x_{i(k+1)} \leq z_{ik}, \forall i, k = 1..4 \tag{6}$$

$$\bar{x}_{i(k+1)} \leq \bar{z}_{ik}, \forall i, k = 1..4 \tag{7}$$

$$z_{ik}, \bar{z}_{ik} \in \{0, 1\}, \forall i, k = 1..4 \tag{8}$$

$$x'_i = \sum_{k=1}^{5} b_k x_{ik}, \; \bar{x}'_i = \sum_{k=1}^{5} b_k \bar{x}_{ik}, \forall i \tag{9}$$

$$\sum_{i=1}^{n} q_i = 1, \; q_i \in \{0, 1\}, \forall i \tag{10}$$

$$0 \leq a - (x'_i(P_i^a)' + \bar{x}'_i(R_i^a)') \leq M(1 - q_i), \forall i \tag{11}$$

$$M(1 - q_i) + \sum_{k=1}^{5}(x_{ik}R_i^d + \bar{x}_{ik}P_i^d) \geq d, \forall i \tag{12}$$

BRPT maximizes, d, the defender's expected utility. The defender has a limited number of resources, Υ, to protect the set of targets, $t_i \in T$ for i=1..n. The defender selects a strategy x that describes the probability that each target will be protected by a resource; we denote these individual probabilities by x_i. Note that $x = \langle x_i \rangle$ is the marginal distribution on each target which is equivalent to a mixed-strategy over all possible assignment of the security forces[2]. The attacker

[1]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.

[2]It is proved in [Korzhyk *et al.*, 2010] that the marginal probability distribution of covering each target is equivalent to a mixed-strategy over all possible resource assignments when there are no assignment restrictions.

chooses a target to attack after observing x. We denote the attacker's choice using the vector of binary variables q_i for $i = 1..n$, where q_i=1 if t_i is attacked and 0 otherwise.

In security games, the payoffs depend only on whether or not the attack was successful. So given a target t_i, the defender receives reward R_i^d if the adversary attacks a target that is covered by the defender; otherwise, the defender receives penalty P_i^d. Respectively, the attacker receives penalty P_i^a in the former case; and reward R_i^a in the latter case.

The defender optimization problem is given in Equations (1)-(12). 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, which are both given input parameters to the MILP. We use a piecewise linear function $\tilde{\pi}(\cdot)$ to approximate the non-linear weighting function $\pi(\cdot)$ and empirically set 5 segments[3] 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. According to PT, the probability x_i is perceived by the attacker as x'_i=$\tilde{\pi}(x_i)$=$\sum_{k=1}^{5} b_k \cdot x_{ik}$, as discussed below.

In order to represent the piecewise linear approximation, i.e. $\tilde{\pi}(x_i)$ (and $\tilde{\pi}(1-x_i)$), 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 [Simon, 1956]. 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; the *epsilon*-optimal assumption is embed in Equation (15), which forces $h_i = 1$ for any target t_i that leads to a prospect that is within ϵ of the optimal prospect, i.e. a; Equation (16) enforces that d is the minimum expected utility of the defender against the $\epsilon-$optimal strategies of the adversary.

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

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

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

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

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

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

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.

3.2 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

$$\max_{x} \frac{\sum_{i=1}^{n} e^{\lambda R_i^a} e^{-\lambda(R_i^a - P_i^a)x_i}((R_i^d - P_i^d)x_i + P_i^d)}{\sum_{j=1}^{n} e^{\lambda R_j^a} e^{-\lambda(R_j^a - P_j^a)x_j}} \tag{19}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_i \leq \Upsilon$$

$$0 \leq x_i \leq 1, \quad \forall i, j$$

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[4] using a gradient

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

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.

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
  $(opt_l, x^*) \leftarrow \texttt{FindLocalMinimum}(x_0)$
  if $opt_g > opt_l$ **then**
    $opt_g \leftarrow opt_l$, $x_{opt} \leftarrow x^*$
  end if
end for
return opt_g, x_{opt}

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 [Pita *et al.*, 2010]. Given the defender's mixed strategy x and N samples of the players' choices, the logarithm 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[5]. Therefore, $\log L(\lambda \mid x)$ only has one local maximum. The MLE of λ is 0.76 for the data used from [Pita *et al.*, 2010].

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.

4 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 [Pita *et al.*, 2010] do not address these challenges.

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($\lvert\frac{R_i^d}{P_i^a}\rvert$)

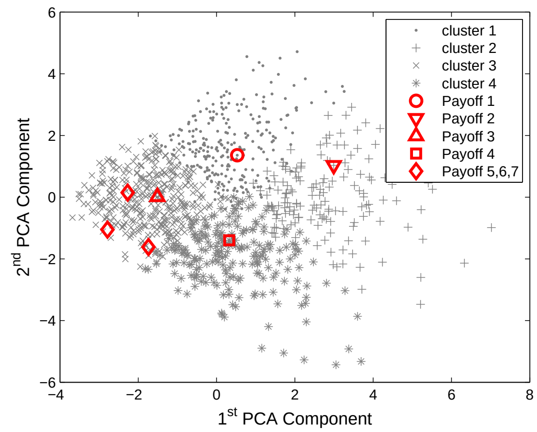


Figure 2: Payoff Structure Clusters (color)

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 [Pita *et al.*, 2010]. 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:

- 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 min-

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

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

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

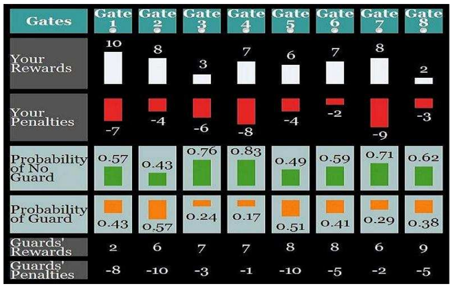


Figure 3: Game Interface

imum 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 [Pita *et al.*, 2010]. 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. 5.2, the performance of different strategies is quite similar in payoffs 5-7.

5 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 [Pita *et al.*, 2008]. Subjects play the role of followers and are able to observe the leader's mixed strategy (i.e., randomized allocation of security resources).

5.1 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

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

We tested the seven different payoff structures[6] from Fig. 2 (four new, three from [Pita *et al.*, 2010]). For each payoff structure we tested the mixed strategies generated by five algorithms: BRPT, RPT, BRQR, COBRA and DOBSS. 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 [Hastie and Dawes, 2001]. 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 in [Pita *et al.*, 2010] (see Section 3.2). For payoffs 1$\sim$4, we set the parameters for COBRA following the advices given by [Pita *et al.*, 2010] as close as possible. In particular, the values we set for α meet the entropy heuristic discussed in that work. For payoffs 5$\sim$7, we use the same parameter settings as in their work.

5.2 Experiment Result

We used defender's expected utility to evaluate the performance of different defender strategies. Given that a subject selects target t_i to attack, the defender's expected utility depends on the strategy she played:

$$U^d_{exp}(x|t_i) = x_i R^d_i + (1 - x_i) P^d_i$$

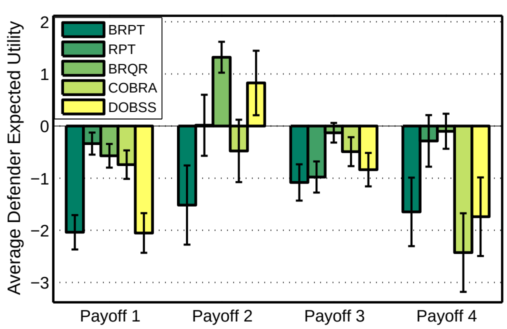
Average Performance: We first evaluate the average defender expected utility, $U^d_{exp}(x)$, of different defender strategies based on all 40 subjects choices:

$$U^d_{exp}(x) = \frac{1}{40} \sum_{i=1}^{n} N_i U^d_{exp}(x|t_i)$$

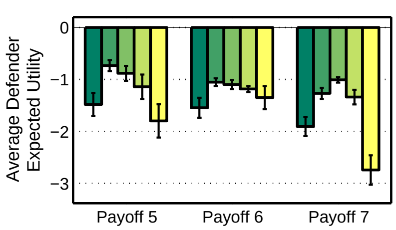
[6]Refer to http://anon-submission.webs.com/ for information of payoff structures, defender's mixed strategy and subjects' choices.

where N_i is the number of subjects that chose target t_i. Fig. 4 displays $U^d_{exp}(x)$ 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 [Wilcox, 2003]. 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.



(a) New Payoffs



(b) Payoffs from Pita et al.

Figure 4: Average Expected Utility of Defender

Robustness: The distribution of defender's expected utility is also analysed to evaluate the robustness of different defender strategies. Figure 5 displays the empirical Cumulative Distributed Function (CDF) of $U^d_{exp}(x|t_i)$ for different defender strategies based the choices of all 40 subjects. The x-axis is the defender expected utility, the y-axis shows the percentage of subjects against whom the defender has gained less than certain amount of expected utility. As the curve moves towards left, the defender expected utility decreases against a certain percentage of the subjects; and vice versa. The left most positive point on the curve indicates the worst defender expected utility of a strategy against different subjects. On the other hand, the range of the curve on the x-axis indicates the reliability of the strategy against various subjects.

As can be seen from Figure 5, defender expected utility has smallest variance when BRQR strategy is played; DOBSS and BRPT strategies lead to large variance in defender expected utility. Furthermore, BRQR achieves highest 'worst' defender expected utility in all payoff structures except in payoff 5, where the CDF of BRQR and RPT strategies are very close.

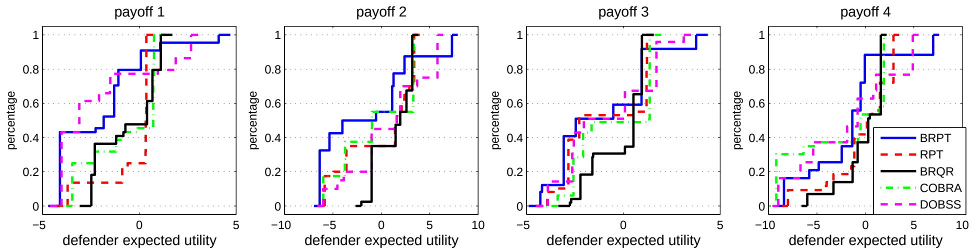
BRPT and DOBSS are not robust against an adversary that deviates from the optimal strategy. BRQR, RPT and COBRA all try to be robust against such deviations. BRQR considers some (possibly very small) probability of adversary attacking any target. In contrast, COBRA and RPT separate the targets into two 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 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[5] selected door 5 in payoff 4 against COBRA).
- '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[5] select door 1 in payoff 2).

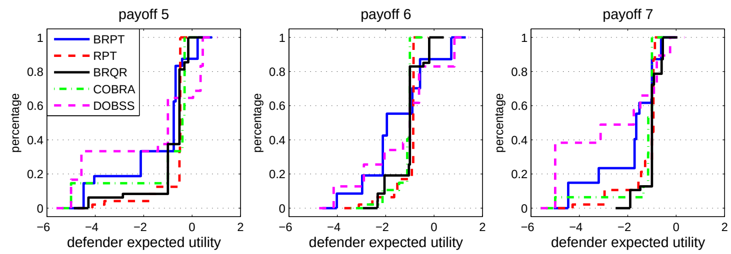
Overall, BRQR performs best, RPT outperforms COBRA in six of the seven cases, and BRPT and DOBSS perform the worst.

6 Conclusions

The unrealistic assumptions of perfect rationality made by existing algorithms applying game-theoretic techniques to real-world security games need to be addressed due to their limitation in facing 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



(a) New Payoffs



(b) Payoffs from Pita et al.

Figure 5: Distribution of Defender's Expected Utility (color)

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.

Acknowledgments

This research was supported by Army Research Office under the grand # W911NF-10-1-0185. We also thank Mohit Goenka and James Pita for their help on developing the web-based game. F. Ordonez would also like to acknowledge the support of Conicyt, through Grant No. ACT87.

References

[Camerer *et al.*, 2004] C. F. Camerer, T. Ho, and J. Chongn. A congnitive hierarchy model of games. *QJE*, 119(3):861–898, 2004.

[Hastie and Dawes, 2001] R. Hastie and R. M. Dawes. *Rational Choice in an Uncertain World: the Psychology of Judgement and Decision Making*. Sage Publications, Thounds Oaks, 2001.

[Kahneman and Tvesky, 1979] D. Kahneman and A. Tvesky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292, 1979.

[Kiekintveld *et al.*, 2009] 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.

[Korzhyk *et al.*, 2010] D. Korzhyk, V. Conitzer, and R. Parr. Complexity of computing optimal stackelberg strategies in security resource allocation games. *In AAAI*, 2010.

[McKelvey and Palfrey, 1995] R. D. McKelvey and T. R. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 2:6–38, 1995.

[Pita *et al.*, 2008] 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.

[Pita *et al.*, 2010] 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.

[Simon, 1956] H. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–138, 1956.

[Stahl and Wilson, 1994] D. O. Stahl and P. W. Wilson. Experimental evidence on players' models of other players. *JEBO*, 25(3):309–327, 1994.

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

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

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