Artificial Intelligence 195 (2013) 440–469

Contents lists available at SciVerse ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Improving resource allocation strategies against human adversaries in security games: An extended study

Rong Yang [a,*], Christopher Kiekintveld [b], Fernando Ordóñez [a,c], Milind Tambe [a], Richard John [a]

[a] University of Southern California, Los Angeles, CA, USA
[b] University of Texas El Paso, El Paso, TX, USA
[c] University of Chile, Santiago, Chile

ARTICLE INFO

Article history:
Received 16 December 2011
Received in revised form 13 November 2012
Accepted 15 November 2012
Available online 20 November 2012

Keywords:
Bounded rationality
Stackelberg games
Decision-making

ABSTRACT

Stackelberg games have garnered significant attention in recent years given their deployment for real world security. Most of these systems, such as ARMOR, IRIS and GUARDS have adopted the standard game-theoretical assumption that adversaries are perfectly rational, which is standard in the game theory literature. This assumption may not hold in real-world security problems due to the bounded rationality of human adversaries, which could potentially reduce the effectiveness of these systems.

In this paper, we focus on relaxing the unrealistic assumption of perfectly rational adversary in Stackelberg security games. In particular, we present new mathematical models of human adversaries' behavior, based on using two fundamental theory/method in human decision making: Prospect Theory (PT) and stochastic discrete choice model. We also provide methods for tuning the parameters of these new models. Additionally, we propose a modification of the standard quantal response based model inspired by rank-dependent expected utility theory. We then develop efficient algorithms to compute the best response of the security forces when playing against the different models of adversaries. In order to evaluate the effectiveness of the new models, we conduct comprehensive experiments with human subjects using a web-based game, comparing them with models previously proposed in the literature to address the perfect rationality assumption on part of the adversary.

Our experimental results show that the subjects' responses follow the assumptions of our new models more closely than the previous perfect rationality assumption. We also show that the defender strategy produced by our new stochastic discrete choice model outperform the previous leading contender for relaxing the assumption of perfect rationality. Furthermore, in a separate set of experiments, we show the benefits of our modified stochastic model (QRRU) over the standard model (QR).[1]

* Corresponding author.
E-mail address: yangrong@usc.edu (R. Yang).

[1] This paper significantly extends our previous conference paper (Yang et al., 2011) [1] by providing (i) new methods for setting parameters of the Prospect Theory model; (ii) an additional variant of Quantal Response model and a new algorithm to compute defender strategies against the new model; (iii) a more comprehensive set of experiments which includes multiple new algorithms and updated settings for the algorithms; (iv) new analysis of the robustness of different defender strategies and the predictive accuracy of different models; (v) additional discussion of related work.

0004-3702/$ – see front matter
http://dx.doi.org/10.1016/j.artint.2012.11.004

1. Introduction

Stackelberg game models have recently become important tools for analyzing real-world security resource allocation problems, such as critical infrastructure protection [2] and robot patrolling strategies [3,4]. These models provide a sophisticated approach for generating unpredictable, randomized strategies that mitigate the ability of attackers to find weaknesses using surveillance. The ARMOR [5], IRIS [6], GUARDS [7] and PROTECT [8] systems are notable examples where this approach has been used to develop decision-support systems for real-world security problems. One of the key sets of assumptions that these systems make are about how attackers will choose attack strategies based on their preferences and observations 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 starting point for the first generation of deployed systems. Unfortunately, this standard game-theoretic assumption 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.

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 [9,10]. A large variety of alternative models have been studied in behavioral game theory and cognitive psychology that capture some of the deviations of human decisions from perfect rationality. 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 advice to human decision-makers in multi-agent decision-support systems [11,12]. Our work in this paper focuses on integrating these more realistic models of human behavior into the computational analysis of Stackelberg game models in security settings, which are often referred to as Stackelberg security games [13–15]. We also provide a case study in this general paradigm of introducing more realistic models of human behavior into game theoretic analysis. While there are quite a few studies looking at the problem of predicting human behavior, there are very few examples where this is actually included in a real decision-making system. Our work here is one of the first examples showing that this is possible, and actually improves performance in an important class of games.

In order to move beyond perfect rationality assumptions to integrate more realistic models of human decision-making in real-world security systems, we address several key challenges. First, the literature has introduced a multitude of potential models on human decision making [16,9,17,10], but each of these models has its own set of assumptions and there is little consensus on which model is best for different types of domains. Therefore, there is an important empirical question of which model best represents the salient features of human behavior in the important class of 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 algorithms for computing solutions to Stackelberg security games, since most existing algorithms are based on mathematically optimal attackers [18,19]. One notable exception is COBRA developed by Pita et al. [20]. COBRA is one example of modeling bounded rationality of human adversaries by taking into account

(i) the anchoring bias of humans while interpreting the probabilities of several events [21,22];
(ii) the limited computational ability of humans which may lead to deviation from their best response.

To the best of our knowledge, COBRA is the best performing strategy for Stackelberg security games in experiments with human subjects. Thus, the open question is whether there are other approaches that allow for fast solutions and outperform COBRA in addressing human behavior in security games.

In this paper, we significantly expand the previous work on modeling human behavior in Stackelberg security games by implementing and evaluating strategies based on two very important methods in literature of modeling human decision-making. The first relates to *Prospect Theory* (PT), which provides a descriptive framework for decision-making under uncertainty that accounts for both risk preferences (e.g. loss aversion) and variations in how humans interpret probabilities through a weighting function [16]. The other method adapts the ideas in the literature on discrete choice problems [23–26] 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 first propose two mathematical models of the adversary's decision-making based on Prospect Theory: one of them assumes the adversary maximizes 'prospect' in their decision making process and the other assumes the adversary makes bounded error in computing such 'prospect' so he may deviate to a sub-optimal solution within a bound. We then propose two mathematical models of how an adversary makes decisions based on using a logit discrete choice models. One model (QR) couples the quantal response of the adversary with the expected utility for attacking each target; the other model (QRRU) modifies the expected utility by adding extra weight to the target covered with minimum resources, inspired by rank-dependent expected utility theory [27].

Based on the above models of adversary decision making, computing the defender's corresponding best response is also challenging since it involves solving non-convex and non-linear optimization problems. We develop new techniques to address these problems. In particular, we develop a Mixed Integer Linear Program to compute the defender optimal strategy against the PT based models by representing the non-linear functions from Prospect Theory with piecewise approximations. Furthermore, we present a local search method with random restarts to compute the defender optimal strategy against the stochastic models of the adversary.

Table 1
Notations used in this paper.

T	Set of targets; t_i in T denotes the ith target
x_i	Probability that target t_i is covered by a resource
q_i	Probability that target t_i is attacked by the adversary
R_i^d	Defender reward when covering t_i if it's attacked
P_i^d	Defender penalty when not covering t_i if it's attack
R_i^a	Attacker reward for attacking t_i if it's not covered
P_i^a	Attacker penalty on attacking t_i if it's covered
M	Total number of resources

In order to compare the performance of different adversary models, we conduct an extensive empirical evaluation using the crowd-source platform Amazon Mechanical Turk[2] (AMT). First, we design an online game called "The Guard and the Treasure" to simulate a security scenario similar to the ARMOR program for the Los Angeles International (LAX) airport [5]. We then develop 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 our new methods against a robust baseline algorithm MAXIMIN, a perfect rationality baseline (DOBSS) and the previous leading contender (COBRA) in the experiments. Our experimental results show that: (i) our new models more accurately represent the adversaries' behavior in security games than previous methods; (ii) strategies based on our new models lead to statistically (and practically) significant higher defender expected utility than the previous leading contender (COBRA). Moreover, we identify situations where the QRRU model of adversary leads to significantly better strategies than the QR model.

The rest of the paper is organized as follows. Section 2 provides necessary background information of Stackelberg security games and defines the notation that will be used in the paper. Section 3 presents the new models of adversary decision-making based on Prospect Theory and Quantal Response Equilibrium. Following that, Section 4 describes the algorithms we developed to compute optimal defender strategy against these new adversary models. In Section 5, we explain the methods we used to decide the parameters of different models. Section 6 presents our experimental setup and results. We then discuss additional related work in Section 7 and summarize the paper in Section 8.

2. Stackelberg security games

In this section, we first define Stackelberg security games as well as the notation used in this paper. We then introduce an online game designed as a testbed to collect data and evaluate performance of the different algorithms introduced in this paper for solving Stackelberg security games.

2.1. Definition and notation

We consider a Stackelberg Security Game (SSG) [1,28] with a single leader and one follower, where the defender plays the role of the leader and the adversary plays the role of the follower. The defender has to protect a set of targets from being attacked by the adversary. The defender has a limited number of resources, e.g., she may need to protect 8 targets with 3 guards. Each player has a set of pure strategies. In SSGs, a pure strategy of an adversary is defined as attacking a single target; and a pure strategy of a defender is defined as an assignment of all the security resources to the set of targets (e.g. assigning the three resources to targets 1, 3 and 6). An assignment of a security resource to a target is also referred to as covering a target. A mixed-strategy is defined as a probability distribution over the set of all possible pure strategies.

We use the following notation to describe a SSG, also listed in Table 1: the defender has a total of M resources to protect a set of targets $\mathcal{T} = \{t_i\}$. The outcomes of the SSG depend only on whether or not the attack is successful. 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. Correspondingly, the attacker receives penalty P_i^a in the former case; and reward R_i^a in the latter case. A key property of SSG is that while the games may be non-zero-sum, $R_i^d > P_i^d$ and $R_i^a > P_i^a$, $\forall i$ [28]. In other words, adding resources to cover a target helps the defender and hurts the attacker.

We represent the defender's mixed-strategy by x which describes the probability that each target will be protected by a resource and denote these individual probabilities by x_i. So we have $x = \langle x_i \rangle$ as the marginal distribution on each target. In the example where the defender has to protect 8 targets with 3 resources (guards), the defender's mixed-strategy can be written as $x = \langle x_1, \ldots, x_8 \rangle$. We focus on generating marginal distributions rather than distributions over the original defender pure strategies (e.g., the original $\binom{8}{3}$ pure strategies) for improved algorithmic efficiency [19,29]. In this paper, we consider the case without any constraints on assigning the resources, which models important domains such as ARMOR deployed at LAX [5]. Korzhyk et al. show in [29] that the marginal probability distribution of covering each target is equivalent to a mixed-strategy over the original combinational defender pure strategies in such domains. Moreover, given the marginal coverage on each target, we could use a technique called 'comb sampling' [30] to implement the corresponding mixed-strategy over the set of the actual assignments of the resources.

[2] https://www.mturk.com.

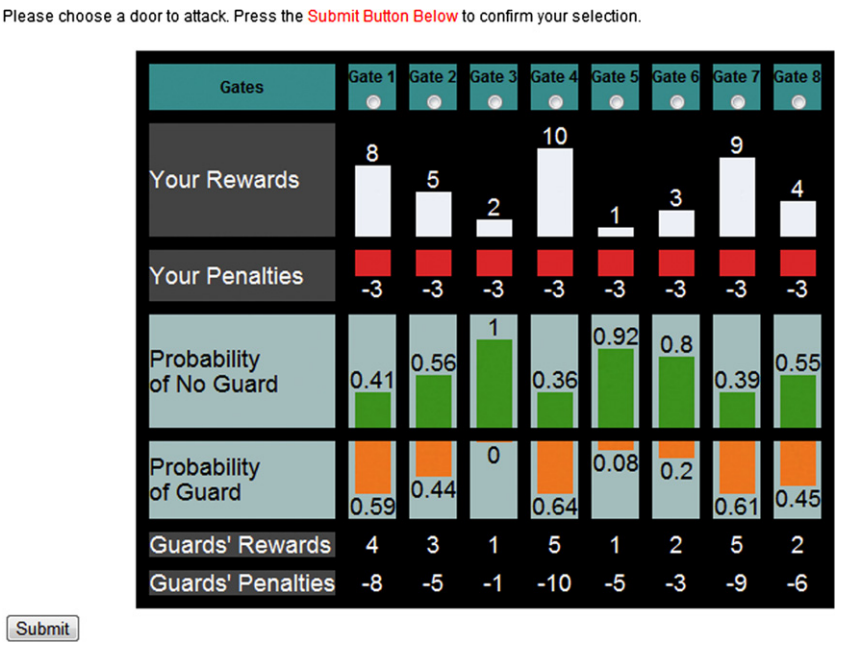


Fig. 1. Game interface for our simulated online SSG.

In SSGs, 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. We denote the attacker's choice using a vector of variables $q = \langle q_i \rangle$ for $t_i \in T$, where $q_i \in [0, 1]$ represents the probability that target t_i will be attacked. Furthermore, we could compute the expected utility for the adversary assuming the target t_i is attacked by the adversary as

$$U_i^a(x_i) = x_i P_i^a + (1 - x_i) R_i^a \tag{1}$$

and the expected utility for the defender in this case is

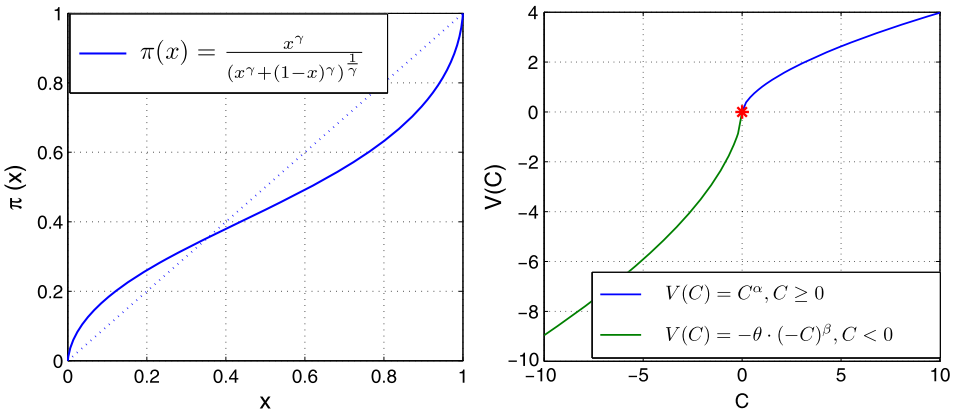
$$U_i^d(x_i) = x_i R_i^d + (1 - x_i) P_i^d \tag{2}$$

2.2. A simulated online SSG

We develop a game, called "The Guards and The Treasure", to simulate the security model at the LAX airport, which has eight terminals that can be targeted in an attack [5]. Fig. 1 shows the interface of the game. Players are introduced to the game through a series of explanatory screens describing how the game is played. In each game instance a subject is asked to choose one of the eight gates to open (attack). They are told that guards are protecting three of the eight gates, but not which ones. The defender's mixed strategy, represented as the marginal probability of covering each target, $\langle x_i \rangle$, is given to the subjects. At the same time, the subjects are also told the reward on successfully attacking each target as well as the penalty of getting caught at each target. The three gates protected by the guards are drawn randomly from the probability shown on the game interface. If subjects select a gate protected by the guards, they receive a penalty; otherwise, they receive a reward. Subjects are rewarded based on the reward/penalty shown for each gate. For example, in the game shown in Fig. 1, the probability that gate 1 (target 1) will be protected by a guard is 0.59. Assuming the subjects choose gate 1, he/she gets reward of 8 if gate 1 is not protected by the guard; or get a penalty of -3 if gate 1 is protected by a guard.

3. New models for predicting attacker behaviors

Existing models of adversary behavior in SSGs have poor performance in predicting the behavior of human adversaries [20]. In order to design better defender strategy, better models of adversary decision-making need to be developed. In this section, we present three models of adversary's behavior in SSGs, based on using Prospect Theory and Quantal Response Equilibrium. All of the models have key parameters. We describe in the next section our methodology for setting these parameters in each case.



(a) weighting function (b) value function

Fig. 2. Prospect Theory empirical function forms.

3.1. Prospect Theory

Prospect Theory provides a descriptive model of how humans make decision among alternatives with risk, which is a process of maximizing the 'prospect', which will be defined soon, rather than the expected utility. More formally, the prospect of a certain alternative is defined as

$$\sum_l \pi(x_l) V(C_l) \tag{3}$$

In Eq. (3), x_l denotes the probability of receiving C_l as the outcome. The weighting function $\pi(\cdot)$ describes how probability x_l is perceived by individuals. An empirical function form of $\pi(\cdot)$ (Eq. (4)) is shown in Fig. 2(a) [31].

$$\pi(x) = \frac{x^\gamma}{(x^\gamma + (1-x)^\gamma)^{\frac{1}{\gamma}}} \tag{4}$$

The key concepts of a weighting function are that individuals overestimate low probability and underestimate high probability [16,31]. Also, $\pi()$ is not consistent with the definition of probability, i.e. $\pi(x) + \pi(1-x) \leqslant 1$ in general.

The value function $V(C_l)$ in Eq. (3) reflects the value of the outcome C_l. PT predicts that individuals are risk averse regarding gain but risk seeking regarding loss, implying an S-shaped value function [16,31]. A key component of Prospect Theory is the reference point. Outcomes lower than the reference point are considered as loss and higher as gain.

$$V(C) = \begin{cases} C^\alpha, & C \geqslant 0 \\ -\theta(-C)^\beta, & C < 0 \end{cases} \tag{5}$$

Eq. (5) is a general form for the value function where C is the relative outcome to the reference. In Eq. (5), we assume the reference point to be at 0. α and β determine the extent of non-linearity in the curves. If the parameters $\alpha = 1.0$ and $\beta = 1.0$, the function would be linear; typical values for both α and β are 0.88 [31]. θ captures the idea that the loss curve is usually steeper than the gains curve, a typical value of θ is 2.25 [31], which reflects a finding that losses are a little more than twice as painful as gains are pleasurable. The function is also displayed in Fig. 2(b) [31]. Given these parameters, we will henceforth denote this value function with $V_{\alpha,\beta,\theta}$.

In a SSG, the prospect of attacking target t_i for the adversary is computed as

$$\texttt{prospect}(t_i) = \pi(x_i) V_{\alpha,\beta,\theta}\left(P_i^a\right) + \pi(1-x_i) V_{\alpha,\beta,\theta}\left(R_i^a\right) \tag{6}$$

According to Prospect Theory, subjects will choose the target with the highest prospect. Thus,

$$q_i = \begin{cases} 1, & \text{if } \texttt{prospect}(t_i) \geqslant \texttt{prospect}(t_{i'}),\ \forall t_{i'} \in T \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

3.2. Quantal Response

Quantal Response Equilibrium (QRE) is an important solution concept in behavioral game theory [17]. It is based on a long history of work in single-agent problems and brings that work into a game-theoretic setting [32,33]. It assumes 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. Given the strategy profile of all the other players, the response of a player is modeled as a quantal response (QR model): he/she selects action i with a probability given by

$$q_i(x) = \frac{e^{\lambda U_i^a(x)}}{\sum_{t_k \in T} e^{\lambda U_k^a(x)}} \tag{8}$$

where, $U_i^a(x)$ is the expected utility for the attacker for selecting pure strategy i. Here, $\lambda \in [0, \infty]$ is the parameter that captures the rational level of player p: one extreme case is $\lambda = 0$, when player p plays uniformly random; the other extreme case is $\lambda \to \infty$, when the quantal response is identical to the best response. Combining Eqs. (8) and (1),

$$q_i(x) = \frac{e^{\lambda R_i^a} e^{-\lambda(R_i^a - P_i^a)x_i}}{\sum_{t_k \in T} e^{\lambda R_k^a} e^{-\lambda(R_k^a - P_k^a)x_k}} \tag{9}$$

In applying the QR model to the security game domain, we only consider noise in the response of the adversary. The defender uses a computer decision support system to choose her strategy hence is able to compute optimal strategy. On the other hand, since the attacker observes the defender's strategy first to decides his response, it can only hurt the defender to add noise in her response. Recent work [33] shows Quantal Level-k [32] to be best suited for predicting human behavior in simultaneous move games. The key idea of level-k is that humans can perform only a bounded number of iterations of strategic reasoning: a level-0 player plays randomly, a level-k ($k > 1$) player best response to the level-$(k-1)$ player. We applied QR instead of Quantal Level-k to model the attacker's response because in Stackelberg security games the attacker observes the defender's strategy, so level-k reasoning is not applicable.

3.3. Quantal Response with Rank-related Expected Utility

We modify the Quantal Response Model by taking into consideration the fact that individuals are attracted to extreme events, such as the less uncertain and highest payoff. This idea is inspired by the rank-dependent Expected Utility Model [27], in which the utilities of choosing different alternatives are based on the their ranks. We adapt this idea to security games, but we only consider such effect on the target covered with minimum resources. That is the adversary would prefer the target covered with minimum resources since he is most likely to be successful attacking that target. This could significantly reduce the defender's reward in the case when this target with fewest resources also gives a large penalty to the defender.

We modify the QR model by adding extra weight to the target covered with minimum resources. We refer this modified model as Quantal Response with Rank-related expected Utility (QRRU) model, where the probability that the attacker attacks target t_i is computed as

$$q_i(x) = \frac{e^{\lambda_u U_i^a(x_i)} e^{\lambda_s S_i(x)}}{\sum_{t_k \in T} e^{\lambda_u U_k^a(x_k)} e^{\lambda_s S_k(x)}} \tag{10}$$

where $S_i(x) \in \{0, 1\}$ indicating whether t_i is covered with least resource.

$$S_i(x) = \begin{cases} 1, & \text{if } x_i \leqslant x_i', \ \forall t_{i'} \in T \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

The denominator in Eq. (10) is only for normalizing the probability distribution so all the q_i sum up to 1. In the numerator, we have two terms deciding the probability that target t_i will be chosen by the adversary. The first term $e^{\lambda_u U_i^a(x_i)}$ relates to the expected utility for the adversary to choose target t_i. $U_i^a(x_i)$ is computed as in Eq. (1). The parameter $\lambda_u \geqslant 0$ represents the level of error in adversary's computation of the expected utility, which is equivalent to λ in Eq. (8). The second term $e^{\lambda_s S_i(x)}$ relates to the adversary's preference for the least covered target. Note that if t_i is not covered with the minimum resource, this term equals to 1 so there is no extra weight added to the non-minimum covered targets; if t_i is covered with minimum resource, this term will be $\geqslant 1$, adding extra weight to the probability that adversary will choose t_i. The parameter $\lambda_s \geqslant 0$ represents the level of the adversary's preference to the minimum covered target. $\lambda_s = 0$ indicates no preference to the minimum covered target. As λ_s increase, this preference becomes stronger.

4. Computing optimal defender strategy

Given the new models of adversary behavior in SSG, new algorithms need to be developed to compute the optimal defender strategy since the existing algorithms are based on the assumption of a perfectly rational adversary. We now describe efficient computation of the optimal defender mixed strategy assuming a human adversary whose response follows one of the three models we proposed: Prospect Theory (PT-Adversary), Quantal Response (QR-Adversary) or Quantal Response with Rank-related Utility (QRRU-Adversary).

4.1. Computing against a PT-adversary

Assuming that the adversary's response follows Prospect Theory (PT-adversary), we developed two methods to compute the optimal defender strategy.

4.1.1. BRPT

Best Response to Prospect Theory (BRPT) is a mixed integer programming formulation for computing the optimal leader strategy against players whose responses follow a PT model. We first present an abstract version of our formulation of BRPT in Eqs. (12)–(16), and then present a more detailed operational version in Eqs. (17)–(29) that uses piecewise linear approximation to provide the BRPT MILP (Mixed Integer Linear Program).

$$\max_{x,q,a,d,z} \; d \tag{12}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_i \leqslant M \tag{13}$$

$$\sum_{i=1}^{n} q_i = 1, \quad q_i \in \{0,1\} \tag{14}$$

$$0 \leqslant a - \left(\pi(x_i) V\left(P_i^a\right) + \pi(1-x_i) V\left(R_i^a\right)\right) \leqslant K(1-q_i), \quad \forall i \tag{15}$$

$$K(1-q_i) + \left(x_i R_i^d + (1-x_i) P_i^d\right) \geqslant d, \quad \forall i \tag{16}$$

The objective is to maximize d, the defender's expected utility. Eq. (13) enforces that the constraint on the total amount of resources is met. In Eq. (14), the integer variables q_i represent the attacker's pure strategy. In BRPT, q_i is constrained to be binary variable, since, as justified and explained in [18], we assume the adversary has a pure strategy best response: $q_i = 1$ if t_i is attacked and 0 otherwise. Eq. (15) is the key to decide the attacker's strategy, given a defender's mixed strategy $x = \langle x_i \rangle$. The variable a represents the attacker's 'benefit' of choosing a pure strategy $\langle q_i \rangle$. Since we are modeling attacker's decision making using Prospect Theory, the benefit perceived by the adversary for attacking target t_i is the attacker's 'prospect', which is calculated as $(\pi(x_i)V(P_i^a) + \pi(1-x_i)V(R_i^a))$ following Eq. (3). The attacker tries to maximize a by choosing the target with the highest 'prospect', as enforced by Eq. (15). In particular, the inequality on the left side of Eq. (15) enforces that a is greater or equal to the 'prospect' of attacking any target. On the right hand of Eq. (15), we have a constant parameter K with a very large positive value. For targets with $q_i = 0$, the upper bound of the difference between a and the 'prospect' is K, therefore, the bounds is not operational. For target with $q_i = 1$ (i.e. the target chosen by the attacker), the value of a is forced to be equal to the actual 'prospect' of attacking that target. In Eq. (16), the constant parameter K enforces that d is only constrained by the target that is attacked by the adversary (i.e. $q_i = 1$).

We now present the BRPT MILP based on our piecewise linear approximation of the weighting function as discussed earlier. We use the empirical functions introduced in Section 3.1 for the weighting function $\pi(\cdot)$ and value function $V(\cdot)$. 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 as input to the optimization formula in Eqs. (13)–(16). The key challenge to solve that optimization problem is that the $\pi(\cdot)$ function is non-linear and non-convex. If we apply the function directly, we have to solve a non-linear and non-convex mixed-integer optimization problem, which is difficult. Therefore, we approximately solve the problem by representing the non-linear $\pi(\cdot)$ function as a piecewise linear function. This transforms the problem into a MILP, which is shown in Eqs. (17)–(29).

$$\max_{x,q,a,d,z} \; d \tag{17}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \sum_{k=1}^{5} x_{ik} \leqslant M \tag{18}$$

$$\sum_{k=1}^{5} (x_{ik} + \bar{x}_{ik}) = 1, \quad \forall i \tag{19}$$

$$0 \leqslant x_{ik}, \bar{x}_{ik} \leqslant c_k - c_{k-1}, \quad \forall i,\ k = 1 \ldots 5 \tag{20}$$

$$z_{ik} \cdot (c_k - c_{k-1}) \leqslant x_{ik}, \quad \forall i,\ k = 1 \ldots 4 \tag{21}$$

$$\bar{z}_{ik} \cdot (c_k - c_{k-1}) \leqslant \bar{x}_{ik}, \quad \forall i,\ k = 1 \ldots 4 \tag{22}$$

$$x_{i(k+1)} \leqslant z_{ik}, \quad \forall i,\ k = 1 \ldots 4 \tag{23}$$

$$\bar{x}_{i(k+1)} \leqslant \bar{z}_{ik}, \quad \forall i,\ k = 1 \ldots 4 \tag{24}$$

$$z_{ik}, \bar{z}_{ik} \in \{0,1\}, \quad \forall i,\ k = 1 \ldots 4 \tag{25}$$

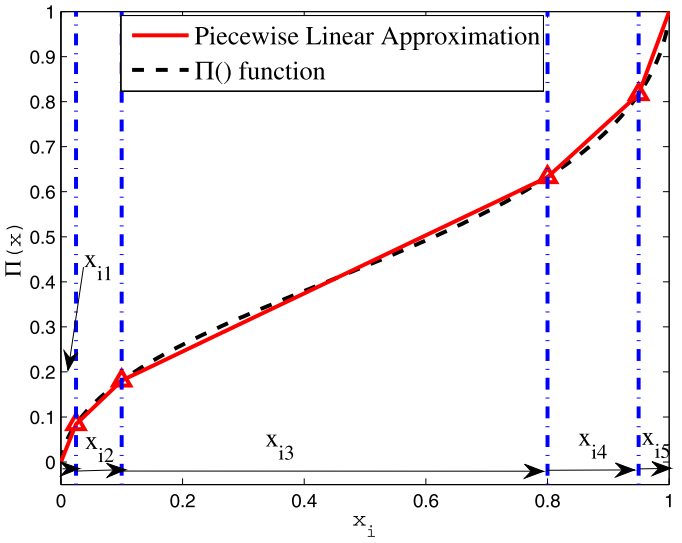


Fig. 3. Piecewise approximation of the weighting function.

$$x'_i = \sum_{k=1}^{5} b_k x_{ik}, \qquad \bar{x}'_i = \sum_{k=1}^{5} b_k \bar{x}_{ik}, \quad \forall i \tag{26}$$

$$\sum_{i=1}^{n} q_i = 1, \quad q_i \in \{0,1\} \tag{27}$$

$$0 \leqslant a - \left(x'_i \left(P_i^a\right)' + \bar{x}'_i \left(R_i^a\right)'\right) \leqslant M(1-q_i), \quad \forall i \tag{28}$$

$$M(1-q_i) + \sum_{k=1}^{5} \left(x_{ik} R_i^d + \bar{x}_{ik} P_i^d\right) \geqslant d, \quad \forall i \tag{29}$$

Let $\tilde{\pi}(\cdot)$ denote the use of a piecewise linear approximation of the weighting function $\pi(\cdot)$, as shown in Fig. 3. We empirically set 5 segments[3] for $\tilde{\pi}(\cdot)$. This function is defined by $\{c_k | c_0 = 0,\ c_5 = 1,\ c_k < c_{k+1},\ k = 0,\ldots,5\}$ that represent the endpoints of the linear segments and $\{b_k \mid k = 1,\ldots,5\}$ that represent the slope of each linear segment. In order to represent the piecewise linear approximation, i.e. $\tilde{\pi}(x_i)$ (and simultaneously $\tilde{\pi}(1-x_i)$), we partition x_i (and $1-x_i$) into five segments, denoted by variables x_{ik} (and $\bar{x}_{ik}$). Therefore, x'_i which equals $\tilde{\pi}(x_i)$ can be calculated as the sum of the linear function in each segment

$$x'_i = \tilde{\pi}(x_i) = \sum_{k=1}^{5} b_k \cdot x_{ik}$$

which is shown in Eq. (26). At the same time, we can enforce the correctness of partitioning x_i (and $1-x_i$) by ensuring that segment x_{ik} (and $\bar{x}_{ik}$) is positive only if the previous segment is used completely. This is enforced in Eqs. (19)–(25) by using the auxiliary integer variable z_{ik} (and $\bar{z}_{ik}$). $z_{ik} = 0$ indicates that the kth segment of x_i (i.e. x_{ik}) has not been completely used, therefore, the following segments can only be set to 0, and vice versa. Eq. (26) defines $x'_i = \tilde{\pi}(x_i)$ as the value of the piecewise linear approximation of x_i, and $\bar{x}'_i = \tilde{\pi}(1-x_i)$ as the value of the piecewise linear approximation of $1-x_i$.

4.1.2. RPT

Robust-PT (RPT) modifies the base BRPT method to account for the possible uncertainty in adversary's choice caused (for example) by imprecise computations [34]. 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 the prospect for the attacker within ϵ of the optimal prospect.

$$\max_{x,h,q,a,d,z} d \tag{30}$$

s.t. Constraints (18)–(28)

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

$$\sum_{i=1}^{n} h_i \geqslant 1 \tag{31}$$

$$h_i \in \{0,1\}, \qquad q_i \leqslant h_i, \quad \forall i \tag{32}$$

$$\epsilon(1-h_i) \leqslant a - \left(x_i'\left(P_i^a\right)' + \bar{x}_i'\left(R_i^a\right)'\right) \leqslant M(1-h_i) + \epsilon, \quad \forall i \tag{33}$$

$$M(1-h_i) + \sum_{k=1}^{5}\left(x_{ik}R_i^d + \bar{x}_{ik}P_i^d\right) \geqslant d, \quad \forall i \tag{34}$$

We modify the BRPT optimization problem as follows: the first 11 constraints are equivalent to those in BRPT (Eq. (18)–(28)); in Eq. (31), the binary variable h_i indicates the ϵ-optimal strategy for the adversary; the ϵ-optimal assumption is embedded in Eq. (33), which forces $h_i = 1$ for any target t_i that leads to a prospect within ϵ of the optimal prospect, i.e. a; Eq. (34) enforces d to be the minimum expected utility for defender on the targets that lead to ϵ-optimal prospect for the attacker. RPT attempts to maximize the minimum for the defender over the ϵ-optimal targets for the attacker, thus providing robustness against attacker (human) deviations within that ϵ-optimal set of targets.

4.2. Computing an optimal strategy against a Quantal Response adversary

Assuming the adversary follows a quantal response (QR-adversary), we now present the algorithm to compute the defender's optimal strategy against a QR-adversary. Given the quantal response of the adversary, which is described in Eq. (9), the best response of defender is to maximize her expected utility:

$$\max_x U^d(x) = \sum_{i=1}^{n} q_i(x)U_i^d(x)$$

Combined with Eqs. (9) and (2), the problem of finding the optimal mixed strategy for the defender can be formulated as

$$\max_x \frac{\sum_{t_i\in T} e^{\lambda R_i^a} e^{-\lambda(R_i^a - P_i^a)x_i}((R_i^d - P_i^d)x_i + P_i^d)}{\sum_{t_k\in T} e^{\lambda R_k^a} e^{-\lambda(R_k^a - P_k^a)x_k}} \tag{35}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_i \leqslant M \tag{36}$$

$$0 \leqslant x_i \leqslant 1, \quad \forall i, j \tag{37}$$

Algorithm 1 BRQR.

```
1: opt_g ← −∞;
2: for it ← 1, ..., IterN do
3:     x^(0) ← randomly generate a feasible starting point
4:     (opt_l, x*) ← Find-Local-Minimum(x^(0))
5:     if opt_g > opt_l then
6:         opt_g ← opt_l, x^opt ← x*
7:     end if
8: end for
9: return opt_g, x^opt
```

Unfortunately, since the objective function in Eq. (35) is non-linear and non-convex, finding the global optimum is extremely difficult. Therefore, we focus on methods to find local optima. To compute an approximately optimal strategy against a QR-adversary efficiently, we develop the Best Response to Quantal Response (BRQR) heuristic described in Algorithm 1. We first take the negative of Eq. (35), converting the maximization problem to a minimization problem. In each iteration, we find the local minimum using the *fmincon*() function in Matlab with the Interior Point Algorithm with a 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. We empirically set *IterN* to 300 in our experiments.

4.3. Computing against a QRRU-adversary

We now present the algorithm to compute defender optimal strategy assuming the adversary's behavior follows the QRRU model. The adversary's response given this model is computed as in Eq. (10). The optimal defender strategy against a QRRU-adversary is computed by solving the following optimization problem:

$$\max_{x,s,x_{\min}} \frac{\sum_{t_i \in T} e^{\lambda_u R_i^a} e^{-\lambda_u (R_i^a - P_i^a)x_i} e^{\lambda_s s_i} ((R_i^d - P_i^d)x_i + P_i^d)}{\sum_{t_k \in T} e^{\lambda_u R_k^a} e^{-\lambda_u (R_k^a - P_k^a)x_k} e^{\lambda_s s_k}} \tag{38}$$

s.t. Constraint (36), (37)

$$x_i - (1 - s_i)K \leqslant x_{\min} \leqslant x_i, \quad \forall t_i \in T \tag{39}$$

$$\sum_{t_i \in T} s_i = 1 \tag{40}$$

$$s_i \in \{0, 1\}, \quad \forall t_i \in T \tag{41}$$

where the integer variables s_i are introduced to represent the function $S_i(x)$ as shown in Eq. (11). In constraint (39), K is a constant with a very large value. Constraints (39) and (40) enforces $x_{\min}$ to be the minimum value among all the x_i. Simultaneously, s_i is set to 1 if target t_i has the minimum coverage probability assigned; and is set to 0 otherwise. The above optimization problem is a non-linear and non-convex mixed integer programming problem, which is difficult to solve directly. Therefore, we developed Best Response to a QRRU-Adversary (BRQRRU), an algorithm that iteratively computes the defender's optimal strategy. The iterative approach breaks down the mixed-integer non-linear programming problem into sub-problems without integer variables. For each sub-problem, one of the target is assumed to be the least covered target. Then, under this constraint, the maximum defender expected utility and the associated defender mixed strategy are computed by solving a non-linear programming problem (similar to BRQR). Finally, the sub-problem generating the highest maximum defender expected utility is found as the 'actual' optimal solution, and the associated defender mixed-strategy is the optimal defender strategy assuming a QRRU-adversary.

Algorithm 2 BRQRRU.

1: $opt_g \leftarrow -\infty$;
2: **for** $t_{i'} \in T$ **do**
3: $(opt_l, x^*) \leftarrow$ `Find-Optimal-Defender-Strategy`$(s_{i'} = 1)$
4: **if** $opt_g > opt_l$ **then**
5: $opt_g \leftarrow opt_l$, $x^{opt} \leftarrow x^*$
6: **end if**
7: **end for**
8: return opt_g, x^{opt}

Algorithm 2 shows the pseudo code of the algorithm. Algorithm 2 describes BRQRRU. In each iteration, one target $t_{i'}$ is conditioned to be covered with minimum resource, therefore $s_{i^*} = 1$. This reduces the optimization problem to the following

$$\max_{x} \frac{\sum_{t_i \in T} e^{\lambda_u R_i^a} e^{-\lambda_u (R_i^a - P_i^a)x_i} e^{\lambda_s s_i} ((R_i^d - P_i^d)x_i + P_i^d)}{\sum_{t_k \in T} e^{\lambda_u R_k^a} e^{-\lambda_u (R_k^a - P_k^a)x_k} e^{\lambda_s s_k}} \tag{42}$$

s.t. Constraint (36), (37)

$$x_{i^*} \leqslant x_i, \quad \forall t_i \in T \tag{43}$$

where there are no integer variables involved since s_i, $\forall t_i \in T$ are all pre-defined parameters of the optimization problem. Therefore, we could solve it using the same method of local search with random restart as that in BRQR. `Find-Optimal-Defender-Strategy`$(s_{i'} = 1)$ on Line (3) in Algorithm 2 calls Algorithm 1 to solve the optimization problem in Eqs. (42)–(43).

5. Parameter estimation

In this section, we describe our methodology for setting the values of the parameters for the different models of human behavior introduced in the previous section. We set the parameters for our later experiments using data collected in a preliminary set of experiments with human subjects playing the online game we introduced in Section 2.2. We posted the game on Amazon Mechanical Turk as a Human Intelligent Task (HIT) and asked subjects to play the game. Subjects played the role of the adversary and were able to observe the defender's mixed strategy (i.e., randomized allocation of security resources). In order to avoid non-compliant participants, we only allowed workers whose HIT approval rates were greater than 95% and who had more than 100 approved HITs to participate in the experiment.

Let G denote a game instance, which is a combination of a payoff structure $\{(R_i^a, P_i^a, R_i^d, P_i^d),\ t_i \in T\}$, and a defender's strategy x. Given a game instance G, we denote the choice of the jth subject as $\tau_j^G \in T$. We include seven payoff structures in the experiments: four of which are selected based on using a classification method we explain in detail in Section 5.1; the other three are taken directly from Pita et al. [20]. For each payoff structure we tested five different defender strategies. This results in $7 * 5 = 35$ different game instances. Each of the subjects played all 35 games. In total, 80 subjects participated in the preliminary experiment.

Table 2
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($\lvert\frac{R_i^d}{P_i^a}\rvert$)

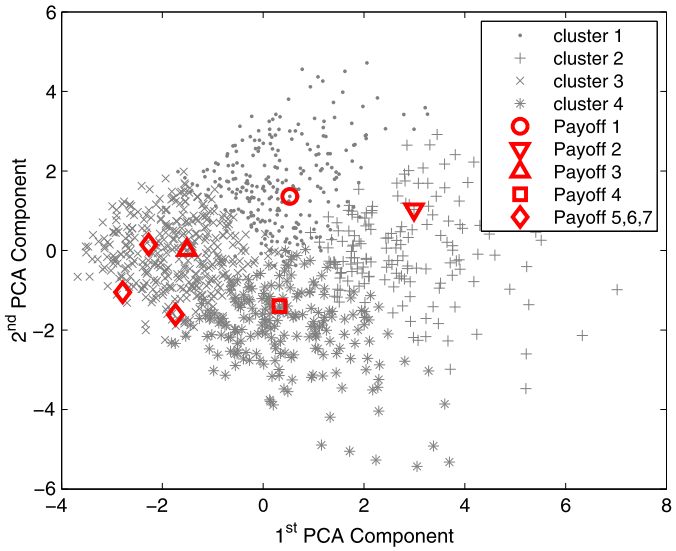


Fig. 4. Payoff structure clusters (color in the web version).

5.1. Selecting payoff structures

Even for a restricted class of games such as security games, there are an infinite number of possible game instances depending on the specific values of the payoffs for each of the targets. Since we cannot conduct experiments on every possible game instance we need a method to select a set of payoffs structures to use in our experiments. Our main criteria for selecting payoffs structures are (1) to select a diverse set of payoff structures that cover different regions in the space of possible security games and (2) to select payoff structures that will differentiate between the different behavioral models (in other words, the models should make different predictions in different test conditions). In the first round our goal was to select game instance that would distinguish between the three key families of prediction methods (BRPT, RPT, BRQR). In the second round of selection we need to further differentiate within the families. Since there is not yet a well-understood method to select such game instances in the literature, we introduce a procedure for making such selections below.

We first sample randomly 1000 different 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 [20]. We then use k-means clustering to group the 1000 payoff structures into four clusters based on eight features, which are defined in Table 2. 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. 4, all 1000 payoff structures are projected onto the first two Principal Component Analysis (PCA) dimensions for visualization. The three payoff structures (5–7) that were first used in Pita et al. [20] are marked in Fig. 4. All three of these payoff structures belong to cluster 3, indicating that the game instances used in the previous experiments we all similar in terms of the features we used for classification.[4]

To select specific payoff structures from these clusters we first generated five defender strategies based on the following families of algorithms: DOBSS, COBRA, BRPT, RPT and BRQR. Here we select only one algorithm from each family (e.g., only one version of BRQR). At this point we did not have preliminary data to set the parameters of the algorithms, since we are deciding which payoff structures to test on. Instead, we set the parameters as follows: DOBSS has no parameters; for COBRA

[4] In [20], there were four payoff structures used, but we only use three of those here. The fourth payoff structure is a zero-sum game, and the deployed Stackelberg security games have not been zero sum [5,6]. Furthermore, in zero-sum games, defender's strategies computed from DOBSS, COBRA and MAXIMIN collapse into one — they turn out to be identical.

we use parameters drawn from [20]; BRPT and RPT use the empirical parameter settings for Prospect Theory [31]; BRQR uses a value of $\lambda = 0.76$ which we set using the data reported in [20] (using the method to be described in Section 5.3).

We use the following the criteria to select payoff structures that differentiate among the different families of 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, we have 10 such measurements.
- We remove payoff structures that have a mean or minimum of these 10 quantities below a given threshold. This results in a subset of about 250 payoff structures in total for all four clusters. We then select one payoff structure closest to the cluster center from each of these subsets.

The four payoff structures (1–4) we selected from different clusters and are marked in Fig. 4.

5.2. Parameter estimation for Prospect Theory

An empirical setting of parameter values is suggested in the literature [31] based on various experiments conducted with human subjects. We also include this setting of parameter values in our experiments to evaluate the benchmark performance of the prospect theory. At the same time, we provide a method to estimate the parameter values for the PT model using a set of empirical response data collected for the SSG domain. In this section, we describe our method of estimating the parameter values based on using grid search.

The empirical functions we used in the PT model for the adversary have four parameters that must be specified: $\alpha, \beta, \theta, \gamma$, as shown in Eqs. (4) and (5). Varying the values for these four parameters will change the responses predicted by the PT-model. We denote the weighting and value function as $\pi_\gamma(\cdot)$ and $V_{\alpha,\beta,\theta}(\cdot)$, for a given a set of parameter values. We then define the fit of a parameter setting to a given data set of subjects' choices as the percentage of subjects who choose the target predicted by the model. The fit can be computed as

$$\texttt{Fit}(\alpha, \beta, \theta, \gamma \mid G) = \frac{1}{N} \sum_{j=1..N} q_{\tau_j^G}(\alpha, \beta, \theta, \gamma \mid G) = \sum_{t_i \in T} \frac{N_i}{N} q_i(\alpha, \beta, \theta, \gamma \mid G)$$

where $q_i(\cdot) \in \{0, 1\}$ indicates whether the PT model predicts target t_i to be chosen by the subjects and is computed using Eq. (7), N_i is the number of subjects who choose target t_i, and $N = \sum_{t_i \in T} N_i$ is the total number of subjects.

We estimate the parameter setting with the best fit for PT model by maximizing the fit function over all 35 game instances

$$\max_{\alpha,\beta,\theta,\gamma} \sum_{G} \texttt{Fit}(\alpha, \beta, \theta, \gamma \mid G) \tag{44}$$

$$\text{s.t.} \quad 0 < \alpha, \ \beta < 1, \ \theta \geqslant 1, \ 0 < \gamma < 1 \tag{45}$$

The constraints in (45) restrict the feasible range of all the four parameters, as defined in the prospect theory model. The objective function in Eq. (44) cannot be expressed as a closed-form expression of α, β, θ and γ. Without a closed form it is difficult to apply gradient descent or any other analytical search algorithm to find the optimal solution. Therefore, we use grid search [35,36] to solve the problem as follows:

(1) We first uniformly sample a set of values for each parameter across the feasible ranges, with the following grid intervals: $\Delta_\alpha = 0.05$, $\Delta_\beta = 0.05$, $\Delta_\gamma = 0.05$, and $\Delta_\theta = 0.1$. This gives a set of different values for each of the four parameters. For simplicity, we represents the four sets of sampled values as the following: $\{\alpha_{k_1} = \alpha_l + k_1 \cdot \Delta_\alpha\}$, where α_l is the lower bound of the region; similarly $\{\beta_{k_2} = \beta_l + k_2 \cdot \Delta_\beta\}$; $\{\theta_{k_3} = \theta_l + k_3 \cdot \Delta_\theta\}$; and $\{\gamma_{k_4} = \gamma_l + k_4 \cdot \Delta_\gamma\}$. The feasible region of θ does not have upper bound, so we set it to 5 which is twice as the suggested empirical value [31].

(2) In total, we have $20 \cdot 20 \cdot 20 \cdot 40 = 320k$ different combinations of the four parameter values. We then evaluate the objective function on each of the combinations $(\alpha_{k_1}, \beta_{k_2}, \theta_{k_3}, \gamma_{k_4})$ and take the parameter combination with the best aggregate fit as the solution:

$$\left(\alpha^*, \beta^*, \theta^*, \gamma^*\right) = \arg \max_{k_1,k_2,k_3,k_4} \sum_{G} \texttt{Fit}(\alpha_{k_1}, \beta_{k_2}, \theta_{k_3}, \gamma_{k_4} \mid G)$$

The parameter settings estimated using the method described above are:

$$\left(\alpha^*, \beta^*, \theta^*, \gamma^*\right) = (1.0, 0.6, 2.2, 0.6)$$

5.3. Parameter estimation for the QR model

We now explain how we estimate the parameter for the Quantal Response Model (QR Model). The parameter λ in the QR model represents the level of noise in the adversary's response function. We employ Maximum Likelihood Estimation (MLE) to fit λ using data we collected. Given a game instance G and N samples of the subjects' choices $\{\tau_j(G),\ j = 1 \dots N\}$, the likelihood of λ is

$$L(\lambda \mid G) = \prod_{j=1..N} q_{\tau_j^G}(\lambda \mid G)$$

where, $\tau_j^G \in T$ denotes the target attacked by the jth player and $q_{\tau_j^G}(\lambda \mid G)$ can be computed by Eq. (9). For example, if player j attacks target t_3 in game G, we would have $q_{\tau_j^G}(\lambda \mid G) = q_3(\lambda \mid G)$. Furthermore, the log-likelihood of λ is

$$\log L(\lambda \mid G) = \sum_{j=1}^{N} \log q_{\tau_j(G)}(\lambda \mid G) = \sum_{t_i \in T} N_i \log q_i(\lambda)$$

Combining with Eq. (8),

$$\log L(\lambda \mid G) = \lambda \sum_{t_i \in T} N_i U_i^a(x_i) - N \log\left(\sum_{t_i \in T} e^{\lambda U_i^a(x)} \right)$$

We learn the optimal parameter setting for λ by maximizing the total log-likelihood over all 35 game instances:

$$\max_{\lambda} \sum_{G} \log L(\lambda \mid G) \tag{46}$$

$$\text{s.t.} \quad \lambda \geqslant 0 \tag{47}$$

The objective function in Eq. (46) is concave, since for each G, a $\log L(\lambda \mid x)$ is a concave function. This can be demonstrated by showing that the second order derivative of $\log L(\lambda \mid G)$ is non-positive $\forall G$:

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

Therefore, $\log L(\lambda \mid x)$ only has one local maximum. We use gradient descent solve the above optimization problem. The MLE of λ is

$$\lambda^* = 0.55$$

5.4. Parameter estimation for the QRRU model

For the QRRU Model, we need to estimate two parameters: λ_u and λ_s as defined in Eq. (10). We again apply Maximum Likelihood Estimation, similar to the method for the QR model. Given a game instance G, and the responses of N subjects $\{\tau_j(G),\ j = 1 \dots N\}$, the log-likelihood of a parameter setting (λ_u, λ_s) is

$$\log L(\lambda_u, \lambda_s \mid G) = \sum_{j=1}^{N} \log q_{\tau_j(G)}(\lambda_u, \lambda_s \mid G) = \sum_{t_i \in T} N_i \log q_i(\lambda_u, \lambda_s)$$

Combining with Eq. (10),

$$\log L(\lambda_u, \lambda_s \mid G) = \lambda_u \sum_{t_i \in T} N_i U_i^a(x_i) + \lambda_s \sum_{t_i \in T} N_i S_i(x) - N \log\left(\sum_{t_i \in T} e^{\lambda_u U_i^a(x_i) + \lambda_s S_i(x)} \right)$$

We learn the optimal parameter settings for the QRRU Model by maximizing the total log-likelihood over all 35 game instances:

$$\max_{\lambda_u, \lambda_s} \sum_{G} \log L(\lambda_u, \lambda_s \mid G) \tag{48}$$

$$\text{s.t.} \quad \lambda_u \geqslant 0, \lambda_s \geqslant 0 \tag{49}$$

The objective function in Eq. (48) is a concave function, since $\forall G$ the Hessian matrix of $\log L(\lambda_u, \lambda_s \mid G)$ is negative semi-definite. We include the details of proof in the appendix and only show here that $\forall \langle \lambda_u, \lambda_s \rangle$

$$\langle \lambda_u, \lambda_s \rangle \cdot H(\lambda_u, \lambda_s \mid G) \cdot \langle \lambda_u, \lambda_s \rangle^T \leqslant 0$$

where $H(\lambda_u, \lambda_s \mid G)$ is the Hessian matrix of $\log L(\lambda_u, \lambda_s \mid G)$ computed as the following

$$H(\lambda_u, \lambda_s \mid G) = -N \begin{pmatrix} \frac{\sum_{i<j}(U_i^a - U_j^a)^2 e^{A_i + A_j}}{(\sum_{t_i \in T} e^{A_i})^2} & \frac{\sum_{i<j}(U_i^a - U_j^a)(S_i - S_j) e^{A_i + A_j}}{(\sum_{t_i \in T} e^{A_i})^2} \\ \frac{\sum_{i<j}(U_i^a - U_j^a)(S_i - S_j) e^{A_i + A_j}}{(\sum_{t_i \in T} e^{A_i})^2} & \frac{\sum_{i<j}(S_i - S_j)^2 e^{A_i + A_j}}{(\sum_{t_i \in T} e^{A_i})^2} \end{pmatrix}$$

where, $A_i = \lambda_u U_i^a(x_i) + \lambda_s S_i(x)$. Therefore, we can use gradient descent to solve the optimization problem in Eqs. (48) and (49). The MLE parameters based on our data set are:

$$\left(\lambda_u^*, \lambda_s^*\right) = (0.6, 0.77)$$

6. Experimental results and discussion

We evaluated the performances of defender strategies as well as the accuracy of different adversary models with human subjects using the online game "The Guard and The Treasure" introduced in Section 2.2. We conducted two set of evaluations: the first set includes the same 7 payoff structures used in the experiments in the previous section; the second set focuses on comparison between the QR model and the QRRU model.

6.1. Experimental settings

The design of the simulated game was already provided in Section 2.2. We now present a detailed description of the experimental settings. In total, we included 70 game instances (comprising 7 payoff structures and 10 strategies for each payoff structure) in the first set and 12 game instances (comprising 4 new payoff structures and 3 strategies for each payoff structure) in the second set. To avoid confusion between these two sets of payoff structures, we will number the first seven payoff structures as 1.1–1.7, and the next four as 2.1–2.4.

Each game instance is played by at least 80 different participants (the actual number of subjects for each game instance ranges between 80 to 91). Each subject is asked to play 40 out of the 70 games. For the purpose of a within-subject comparison, we want a subject to play the 10 different strategies for the same payoff structure. Therefore, the 40 games is composed of 4 payoff structures and 10 defender strategies for each. Furthermore, in order to mitigate the ordering effect on subject responses, we randomize the order of the game instances played by each subject. We generated 40 different orderings of the games using Latin square design. The order played by each subject was drawn uniformly randomly from the 40 possible orderings. To further mitigate ordering effect, no feedback on success or failure is given to the subjects until the end of the experiment. As motivation to the subjects, they earn or lose money based on whether or not they succeed in attacking a gate; if the subject opens a gate not protected by the guards, they win; otherwise, they lose.

The participants were recruited on Amazon Mechanical Turk. Note that these participants differ from those who played the game to provide data for estimating the parameter, as discussed in the previous section. In order to avoid non-compliant participants, we only allowed workers whose HIT approval rates were greater than 95% and who had more than 100 approved HITs to participate in the experiment. They were first given a detailed instruction of the game explaining to them how the game is played. Then two practical rounds of games were provided to help them get familiar with the game. After all the learning and practising, they were given enough time to finish all the games.

Each participant first received 50 cents for participating in the game. Then they gain bonus based on the outcomes of the games they played, with each point worth 1 cent. On average, the subjects who participated in the first set of experiment (i.e. payoff 1.1–1.7) received $1.45 as bonus based on their total scores across 40 game instances they played; the subjects who participated in the second set of experiment (i.e. payoff 2.1–2.4) received $0.44 as bonus based on their total scores across 12 game instances they played. Participants were given 5 hours in total to finish the experiment which was shown to be sufficiently long given that the average time they spent was 28 minutes for the first set of 40 games and 8 minutes for the second set of 12 games.

In the following part of this section, we first describe the parameter settings for the different leader strategies. We then provide our experimental results, and follow that up with analysis. We compare both the quality of different defender strategies against the human participants and the accuracy of different adversary models in the sense that how well the human participants follow the assumption of these models.

6.2. Algorithm parameters

For the seven payoff structures (1.1–1.7) introduced in Section 5, we tested ten different mixed strategies generated from seven different algorithms: Maximin, Dobss [18], Cobra [20], Brpt, Rpt, Brqr, Brqrru. We include Maximin as a benchmark algorithm. Maximin assumes that adversary always selects the target that is worst to the defender. Table 3 lists the parameter settings of these ten strategies for each of the seven payoff structures.

Table 3
Parameter settings for different algorithms.

Payoff	1.1	1.2	1.3	1.4	1.5	1.6	1.7
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
BRPT-E	$(\alpha, \beta, \theta, \gamma) = (0.88, 0.88, 2.25, 0.64)$						
RPT-E	$(\alpha, \beta, \theta, \gamma) = (0.88, 0.88, 2.25, 0.64)$, $\epsilon = 2.5$						
BRPT-L	$(\alpha, \beta, \theta, \gamma) = (1, 0.6, 2.2, 0.6)$						
RPT-L	$(\alpha, \beta, \theta, \gamma) = (1, 0.6, 2.2, 0.6)$, $\epsilon = 2.5$						
BRQR-76	$\lambda = 0.76$						
BRQR-55	$\lambda = 0.55$						
BRQRRU	$(\lambda_u, \lambda_s) = (0.6, 0.77)$						

- DOBSS and MAXIMIN have no parameters.
- For COBRA, we set the parameters following the methodology presented in [20] as closely as possible for payoff structures 1.1∼1.4. In particular, the values we set for α meet the entropy heuristic discussed in that work. For payoff structures 1.5∼1.7 that are identical to payoff structures first used by Pita et al., we use the same parameter settings as in their work.
- For both BRPT-E and RPT-E, the parameters for Prospect Theory are empirical values suggested by literatures [31]. For RPT-E, we empirically set ϵ to 25% of the maximum potential reward for the adversary, which is 10 in our experimental settings.
- We tried another set of parameters for Prospect Theory, which are learned from our first set of experiment as described in Section 5.2. We denote these two algorithms as BRPT-L and RPT-L.
- For BRQR, we tried two different values for the parameter λ, $\lambda = 0.76$ is the values learned from the data reported by Pita et al. [20]; $\lambda = 0.55$ is the value learned from data collected in our first set of experiments with participants from Amazon Mechanical Turk. We will refer to the strategies resulting from these two parameter settings of the BRQR algorithm as BRQR-76 and BRQR-55 respectively.
- For BRQRRU, the parameters are learned from the data collected our first set of experiments.

6.3. Quality comparison

We evaluated the performance of different defender strategies using the defender's expected utility and the statistical significance of our results using the bootstrap-t method [37].

6.3.1. Average performance

We first evaluated the average defender expected utility, $U^d_{avg}(x)$, of different defender strategies based on the subjects' choices:

$$U^d_{avg}(x) = \frac{1}{N}\sum_{j=1}^{N} U^d_{\tau_j}(x) = \frac{1}{N}\sum_{t_i \in T} N_i U^d_i(x_i)$$

where τ_j is the target selected by the jth subject, N_i is the number of subjects that chose target t_i and N is the total number of subjects. Fig. 5 displays $U^d_{avg}(x)$ for the different strategies in each payoff structure. We also displayed the normalized defender average expected utility of different strategies within each payoff structure in Fig. 6. After normalization, $U^d_{avg}(x)$ for each defender strategy varies between 0 and 1, with the highest $U^d_{avg}(x)$ in each payoff structure scaled to 1 and the lowest $U^d_{avg}(x)$ scaled to 0.

Overall, BRQR-76, BRQR-55 and BRQRRU performed better than other algorithms. We compare the performance of three algorithms with each of the other seven algorithms and report the level of statistical significance in Tables 4, 5 and 6. We summarize the results below:

- MAXIMIN is outperformed by all three algorithms with statistical significance in all seven payoff structures. DOBSS is also outperformed by all three algorithms with statistical significance except for payoff structure 1.6.
- In five of the seven payoff structures, COBRA is outperformed by all three algorithms with statistical significance. In payoff structure 1.3, the performance of COBRA is very close to the three algorithms, but there is no statistical significance either way. In payoff structure 1.5, COBRA is outperformed by all three algorithms but no statistical significance is achieved.
- The three algorithms outperform BRPT-E with statistical significance in all seven payoff structures. Furthermore, BRPT-L is outperformed by the three algorithms in all seven payoff structures with statistical significance in six cases except for in payoff structure 1.6.
- In four of the seven payoff structures, RPT-E is outperformed by the three algorithms with statistical significance. In payoff 1.3, RPT-E is outperformed by all three algorithms but the result is not statistical significant. In payoff structure

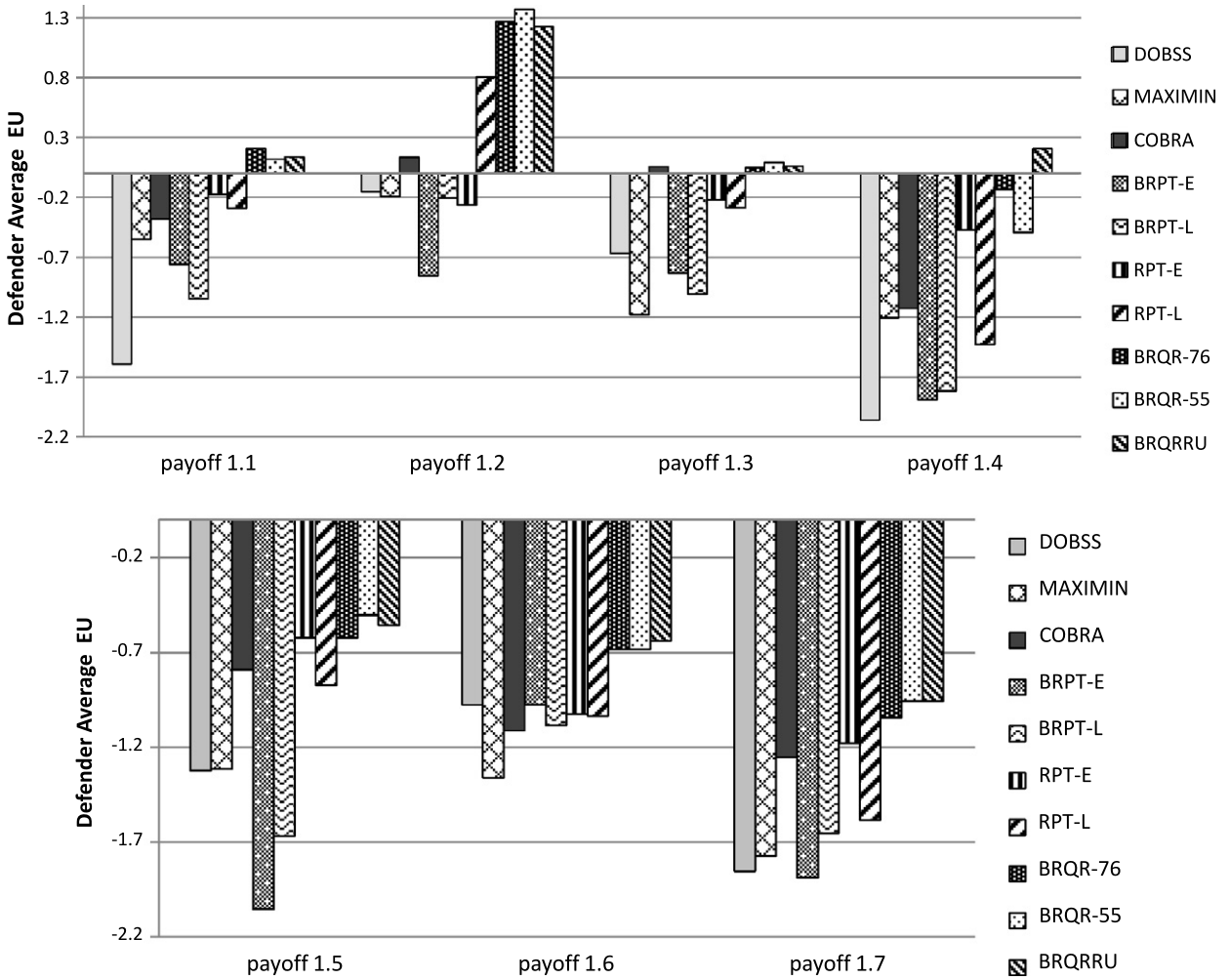


Fig. 5. Defender average expected utility achieved by different strategies.

1.4, RPT-E achieves very similar performance to BRQR-55 and is outperformed by BRQR-76 and BRQRRU. In payoff 1.5, RPT-E achieves very similar performance as BRQR-76 and is outperformed by BRQR-55 and BRQRRU. Furthermore, RPT-L is outperformed by all three algorithms with statistical significance in almost all seven payoff structures, except for in payoff structure 1.2 where the result of comparing BRQR-76 and BRQRRU with RPT-L doesn't have statistical significance.

Overall, any of the three quantal response (BRQR-76, BRQR-55 and BRQRRU) strategies would be preferred over the other strategies. However, the performance of the three strategies are close to each other in this set of experiments. In order to further differentiate the three strategies as well as prove the effectiveness of QRRU model, we conducted a separate set of experiments. We first select four new payoff structures from the 1000 random samples using the following rules:

- We first measure the distance between the BRQRRU strategy and each of the other two BRQR strategies using Kullback–Leibler (KL) 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, we measure this KL distance for the pair (BRQRRU, BRQR-76) and the pair (BRQRRU, BRQR-55). So we have two such measurements for each payoff structure.
- We sort the payoff structures in a descending order of the mean of these two distance.
- In the top 10 payoff structures, we select two payoff structures where the targets assigned with minimum coverage probability by BRQR-76 or BRQR-55 have large penalty for the defender; and two payoff structures where the penalty for the defender on such target is small.

The details of these four payoff structures and the defender strategies are included in the appendix. We conducted a new set of experiments with human subjects using these four payoff structures and the three QR model based strategies for each payoff structure. In total, we have $4 * 3 = 12$ game instances included in these experiments. Each subject is asked to play against all these 12 game instances. 80 subjects are involved in these experiments.

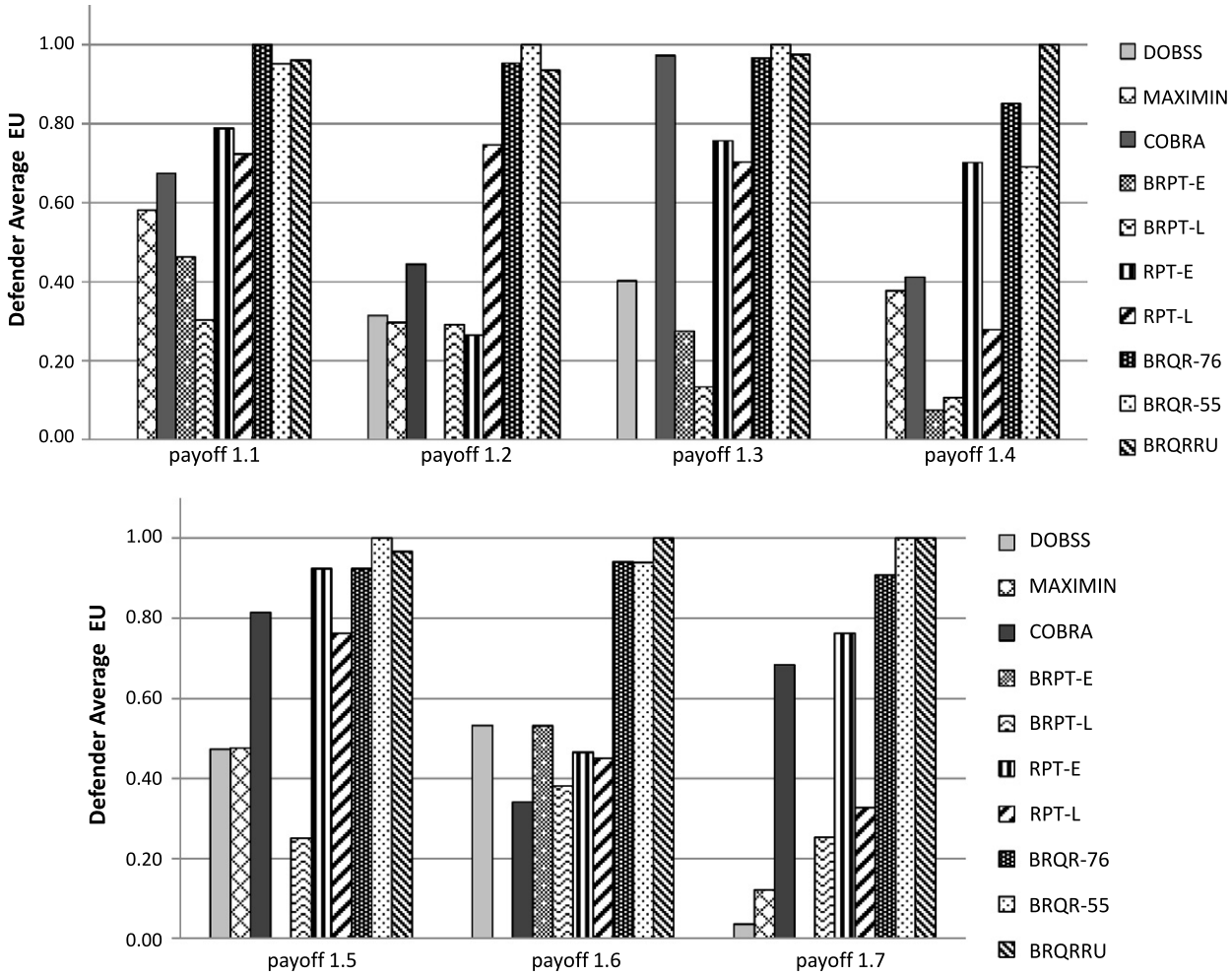


Fig. 6. Defender average expected utility (normalized between 0 and 1) achieved by different strategies.

Table 4
Level of statistical significance of comparing BRQR-76 to other algorithms: ***($p \leqslant 0.01$), **($p \leqslant 0.05$), *($p \leqslant 0.1$).

v.s.	DOBSS	MAXIMIN	COBRA	BRPT-E	RPT-E	BRPT-L	RPT-L
payoff 1.1	***	***	***	***	**	***	***
payoff 1.2	***	***	***	***	***	***	0.15
payoff 1.3	***	***	0.96	***	0.21	***	**
payoff 1.4	***	***	*	***	0.25	***	***
payoff 1.5	***	***	0.26	***	0.99	***	***
payoff 1.6	0.20	***	***	*	***	0.13	***
payoff 1.7	***	***	**	***	**	***	***

Table 5
Level of statistical significance of comparing BRQR-55 to other algorithms: ***($p \leqslant 0.01$), **($p \leqslant 0.05$), *($p \leqslant 0.1$).

v.s.	DOBSS	MAXIMIN	COBRA	BRPT-E	RPT-E	BRPT-L	RPT-L
payoff 1.1	***	***	**	***	*	***	***
payoff 1.2	***	***	**	***	***	***	*
payoff 1.3	***	***	0.86	***	0.16	***	**
payoff 1.4	***	***	**	***	0.95	***	**
payoff 1.5	***	***	0.37	***	0.12	***	**
payoff 1.6	0.16	***	***	**	***	0.11	***
payoff 1.7	***	***	***	***	***	***	***

Table 6
Level of statistical significance of comparing BRQRRU to other algorithms: ***($p \leqslant 0.01$), **($p \leqslant 0.05$), *($p \leqslant 0.1$).

v.s.	DOBSS	MAXIMIN	COBRA	BRPT-E	RPT-E	BRPT-L	RPT-L
payoff 1.1	***	***	**	***	*	***	***
payoff 1.2	***	***	**	***	***	***	0.27
payoff 1.3	***	***	0.99	***	0.27	***	**
payoff 1.4	***	***	**	***	0.18	***	***
payoff 1.5	***	***	0.40	***	0.33	***	*
payoff 1.6	0.15	***	***	**	***	0.11	***
payoff 1.7	***	***	***	***	***	***	***

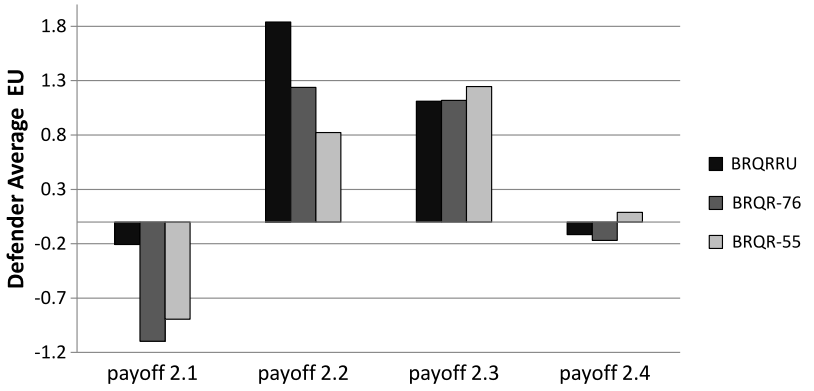


Fig. 7. Defender average expected utility achieved by QR model based strategies.

Table 7
Statistical significance (**: $p \leqslant 0.05$; ***: $p \leqslant 0.01$).

payoff 2.1	BRQRRU v.s. BRQR-76	***
	BRQRRU v.s. BRQR-55	**
payoff 2.2	BRQRRU v.s. BRQR-76	**
	BRQRRU v.s. BRQR-55	**
payoff 2.3	BRQR-76 v.s. BRQRRU	0.87
	BRQR-55 v.s. BRQRRU	0.40
payoff 2.4	BRQRRU v.s. BRQR-76	0.97
	BRQR-55 v.s. BRQRRU	0.35

Fig. 7 displays the defender average expected utility achieved by the three strategies. We report the statistical significance results in Table 7. In payoff structures 2.1 and 2.2, BRQRRU outperforms both BRQR-76 and BRQR-55 with statistical significance. In payoff structures 2.3 and 2.4, the three strategies have very close performance. No statistical significance is found in the results, as reported in Table 7.

As noted earlier, a very important feature of payoff structures 2.1 and 2.2, compared to payoff structures 2.3 and 2.4, is that the target covered with minimum resource by BRQR-76 and BRQR-55 (target 3 in payoff structure 2.1 and target 3 in payoff structure 2.2) has a large penalty ($\leqslant -6$) for the defender. In the experiments with payoff structures 2.1 and 2.2, more than 10% of subjects selected these targets (target 3 in payoff structure 2.1 and 2.2) while playing against BRQR-76 or BRQR-55, while no subjects chose this target while playing against BRQRRU — BRQRRU covers these targets with more resources. This is the main reason why BRQRRU significantly outperforms BRQR in payoff 2.1 and payoff 2.2. In payoff 2.3 and 2.4, similar observation is obtained in subjects' choice: the targets covered with minimum resources by BRQR-76 and BRQR-55 are selected more frequently compared to the case when BRQRRU is played. However, these targets (i.e. target 1 in payoff 2.3 and target 2 in payoff 2.4) have very small penalty for the defender (-1). Therefore we do not see significant differences in performance among the different BRQR strategies.

Based on the result in both sets of experiments, we conclude that the stochastic model based strategies are superior to their competitors, and BRQRRU is the preferred strategy within the stochastic model based strategies. In particular, BRQRRU achieves significantly better performance than BRQR when the target covered with minimum resource by BRQR has potentially a large penalty for the defender; and has a performance similar to the other stochastic model based strategies otherwise.

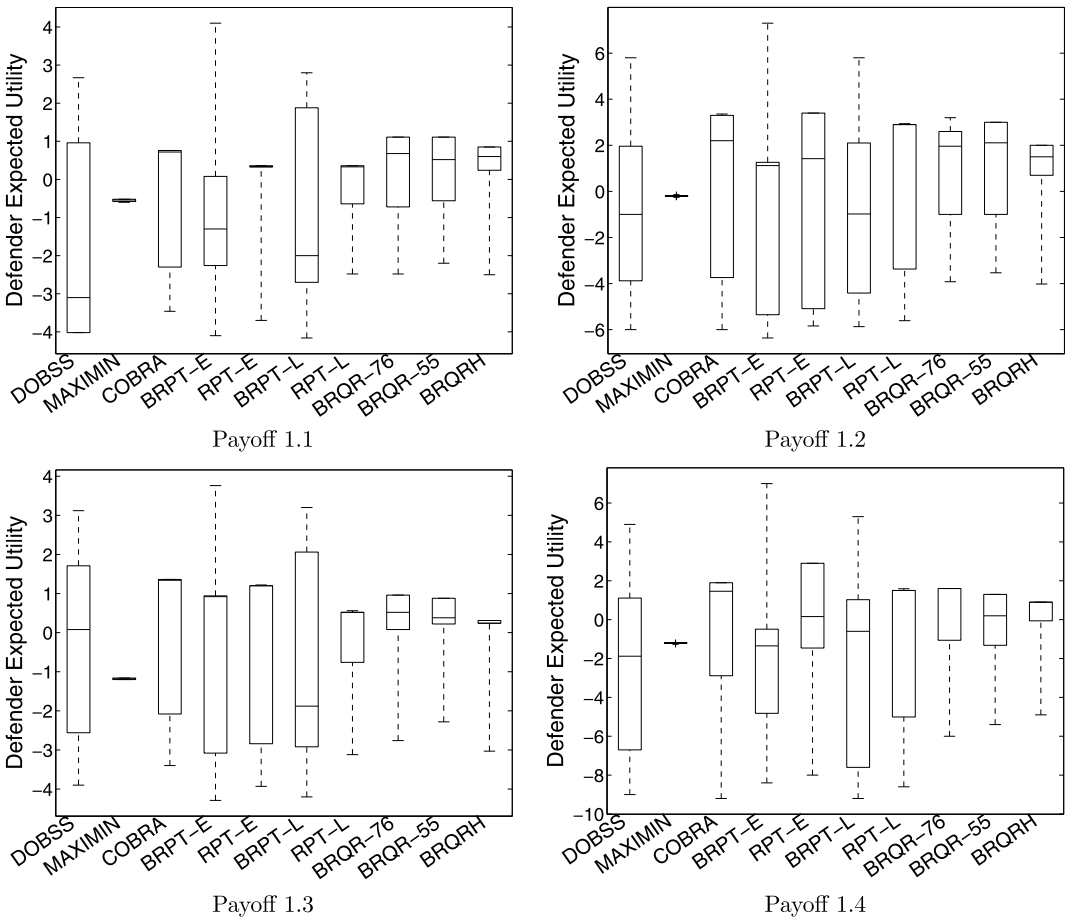


Fig. 8. Distribution of defender's expected utility against each individual subject.

6.3.2. Performance distribution

We now analyze the distribution of the performance of each defender strategy while playing against different adversaries (subjects). Given a game instance G, the defender expected utility achieved by playing strategy x against a subject j is denoted as $U^d_{\tau^G_j}(x)$. Figs. 8 and 9 display the distribution of $U^d_{\tau^G_j}(x)$ for different defender strategies against individual subjects in each payoff structure. The y-axis shows the range of the defender's expected utility against all different subjects. Each box with the extended dash line in the figure shows the distribution of this defender expected utility for each of the ten defender strategies: the dashed line specifies the range of $U^d_{\tau^G_j}(x)$ with the bottom band showing the minimum value and the top band showing the maximum value; the box specified the 25th to 75th percentiles of $U^d_{\tau^G_j}(x)$ with the bottom showing the 25th percentile value and the top showing the 75th value; the band inside the box specifies the median (50th percentile) of $U^d_{\tau^G_j}(x)$. We compare the distributions of different defender strategies from two perspectives:

Range: As presented in Fig. 8 and Fig. 9, in general, the defender expected utility has the smallest range when Maximin strategy is played (except that in payoff structure 1.7, the range of the defender expected utility when RPT-L is played is slightly smaller than that when Maximin is played). Cobra, Rpt, Brqr and Brqrru lead to larger range of defender expected utility than Maximin. Defender expected utility has the largest range when Dobss or Brpt is played.

Worst case: The lower band of the dashed line indicates the worst-case defender expected utility when different strategies are played. Maximin has the highest worst-case defender expected utility in general (except that in payoff 1.5, the worst-case defender expected utility by playing Brqr-76 is better than that by playing Maximin). Dobss and Brpt lead to lowest worst-case defender expected utility. The worst-case defender expected utility from playing Cobra, Rpt, Brqr and Brqrru are in between the two extreme cases. Furthermore, Brqr and Brqrru lead to higher worst-case defender expected utility than Cobra and Rpt.

In general, by playing Maximin, the defender expected utility against each individual adversary achieves the smallest variance, hence it is most robust to the uncertainty in adversary's choice. However, it does so by assuming that the adversary could select any target hence making the expected utility on each target equal. Maximin does not exploit the different preferences adversary may have among different targets. Brpt and Dobss assume the subjects select the target that maxi-

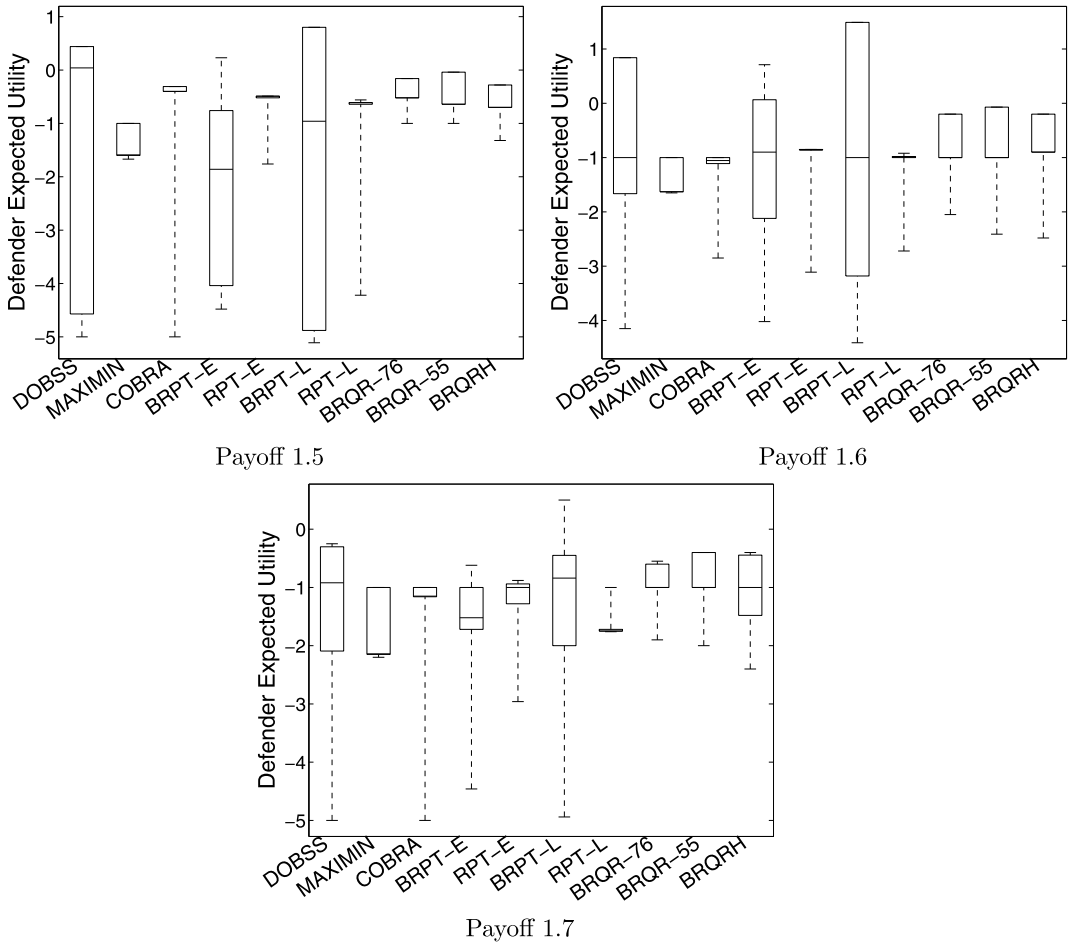


Fig. 9. Distribution of defender's expected utility against each individual subject.

mizes their expected utility and do not consider the possibility of deviations from the optimal choice by the adversary. This leads to arbitrarily lower defender expected utility when the adversary deviates from the predicted choice.

COBRA, RPT, BRQR and BRQRRU all try to be robust against such deviations. BRQR and BRQRRU consider some (possibly very small) probability of adversary attacking any target using a soft-max function. 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 the non-ϵ-optimal targets. When the non-ϵ-optimal targets have high defender penalties, COBRA and RPT become vulnerable to adversary's deviation. For example, target 6 in payoff structure 1.2 has a small reward ($=1$) and a large penalty ($=-10$) for the attacker. Both COBRA and RPT consider this target to be in the non-ϵ-optimal set and assign very small probability to cover this target ($\leqslant 0.05$). However, approximately 10% of the subjects have chosen this target. Since this target has a high defender penalty (-6), COBRA and RPT lose reward on this target. Similar examples include target 5 in payoff structure 1.4 and target 8 in payoff structure 1.1.

6.4. Model prediction accuracy

In this section, we evaluate how well each model predicts the actual responses of human participants using three different metrics [38]: mean square deviation (*MSD*), a proportion of inaccuracy (*POI*), and Euclidean distance (*ED*).

We first extend the definition of *MSD* from that in [38] which is designed for a 2-action game, in order to suit our domain where the player has 8 actions to take. Given the choices of the N subjects, the *MSD* of a model is computed as

$$MSD = \left\{ \frac{1}{N} \sum_{n=1}^{N} (p_{\tau(n)} - 1)^2 \right\}^{1/2} \tag{50}$$

where, $\tau(n)$ represents the index of the target chosen by subject n, p_i is the predicted probability by a model that target i will be chosen.

Table 8
Ability of behavioral models to predict attacker decision.

Model	Out of sample			In sample		
	MSD	*POI*	*ED*	*MSD*	*POI*	*ED*
DOBSS	0.81	0.67	0.76	0.85	0.73	0.80
PT-E	0.84	0.71	0.81	0.87	0.75	0.84
PT-L	0.84	0.71	0.81	0.86	0.74	0.83
QR-76	0.79	0.67	0.23	0.83	0.73	0.22
QR-55	0.81	0.67	0.22	0.84	0.73	0.21
QRRU	0.80	0.65	0.21	0.83	0.70	0.18
COBRA	0.91	0.83 (0.35)	0.94	0.91	0.83 (0.42)	0.93
RPT-E	0.93	0.87 (0.52)	0.99	0.94	0.88 (0.56)	0.99
RPT-L	0.93	0.86 (0.49)	0.98	0.93	0.86 (0.54)	0.96

The *POI* score is meant to put models with deterministic prediction on the same footing as those with stochastic prediction. It treats the target with the highest predicted probability as the predicted target, and computes the proportion of the subjects who didn't choose the predicted target. The *POI* score is computed as

$$POI = \frac{1}{N}\sum_{n=1}^{N}(1 - \tilde{p}_{\tau(n)}) \tag{51}$$

where, $\tau(n)$ is the index of the target chosen by subject n. $\tilde{p}_{\tau(n)} = 1$ if $\tau(n)$ is the predicted target; and $\tilde{p}_{\tau(n)} = 0$ otherwise. Note that for models with deterministic prediction, the *POI* score is exactly equal to the square of *MSD* value.

The Euclidean distance measures the difference between the actual distribution of the subjects' choices and the prediction of the model. It is computed as

$$ED = \sqrt{\sum_{i \in T}\left(p_i - p_i^{act}\right)^2} \tag{52}$$

where p_i is the probability predicted by the model that target i will be chosen, and p_i^{act} is the actually percentage of subjects who have chosen target i.

Table 8 presents the ability of different models to predict the attacker decision measured with the three different criteria.[5] The measurements for both the out-of-sample data (70 rounds of games) and in-sample data (35 rounds of games) are displayed in the table. Better predictive power is indicated by lower *MSD* value and *POI* score and lower *ED* value. The top four models all have deterministic prediction and the three quantal response related models have stochastic prediction. The last three models (COBRA, RPT-E and RPT-L) don't have a strict definition of the prediction of the attacker's behavior. They are modifications of the base models for robustness. For example, COBRA modifies DOBSS by assuming that attacker will deviate from choosing the target with the highest expected utility to any other targets whose expected utilities are within ϵ of the highest value. However, within this subset of possibly chosen targets, the model doesn't explicitly predict the behavior of the attacker but rather plays a maximin strategy (i.e. maximizing the lowest expected utility). RPT-E and RPT-L modify PT-E and PT-L in similar ways. Given the above property of these three models, we compute the *POI* score in two different ways by using two different definitions of the model prediction.

- The first definition predicts a single target with the lowest expected utility for the defender within the subset of possible deviations. Therefore the *POI* score counts the proportion of subjects who have chosen any other targets.
- The second definition predicts all the targets within the subset of the possible deviations. Therefore, the *POI* score only counts for the targets outside this subset.

The *POI* score computed with the first definition should be equal to or higher than the value computed with the second definition. Note that the second definition doesn't satisfy the property of prediction since the sum of the predictions on all targets might be larger than 1. We use this definition to mainly show the importance of accounting for deviation of attackers' decision. The *POI* values computed with the second definition are shown in parentheses in Table 8. The observations from the table are summarized blow,

1. For the out-of-sample data, less than 30% of the subjects have selected the target predicted by PT-E or PT-L; in the other words, more than 70% of the subjects have deviated from the prediction. For DOBSS, on average 67% of the subjects deviated from the predicted response. Similar patterns can be observed for the in-sample data.

2. Both RPT and COBRA take into consideration the deviation of the subjects' responses from their optimal action. The percentage of subjects deviate from the model prediction decreased significantly: for the out-of-sample data, the *POI* score

[5] MAXIMIN doesn't have a prediction of adversary behavior, so we exclude it from the analysis.

of COBRA is 0.35 compared to 0.67 of DOBSS; the *POI* score of RPT-E decreased by 0.19 compared to PT-E; the *POI* score of RPT-L decreased by 0.22 compared to PT-L. Similar patterns are observed for the in-sample data.

3. The *POI* score of QR-76 and QR-55 is the same as DOBSS. This is expected since the target predicted by the QR model to be chosen with the highest probability is the target with the highest expected utility for the attacker, which is the also prediction of DOBSS. In other words, QR-76 and QR-55 have the same predicted target as DOBSS. At the same time, QRRU has the lowest *POI* score among all the models in both the out-of-sample data and in-sample data. The *MSD* scores of the three QR-related models are better (lower) than other models (except that in the out-of-sample data QR-55 has the same score as DOBSS).

4. The advantage of the three QR related models is most significant under the *ED* score, which represents the error of the model in predicting the distribution of subjects' choices. As shown in Table 8, the three QR-related models have significantly lower *ED* scores than the other models. This is essentially the reason why the three models achieved significantly better defender expected utility than the other models.

7. Related work

In the first few sections of the paper, we discussed recent developments of game-theoretical approaches to solve Stackelberg security games. We discuss additional related work in this section.

Motivated by real-world security problems, there have been many algorithms developed to compute optimal defender strategies in Stackelberg games [19,18,30]. The first such algorithm to be used in a real application is DOBSS (Decomposed Optimal Bayesian Stackelberg Solver) [18], which is central to the ARMOR system [5] at LAX airport and the GUARDS system [7] built for the Transportation Security Administration. Other works related to Stackelberg security games include those of Agmon et al. [3,39] and those of Gatti et al. [40,4] on multi-robot patrolling. However, an important limitation of all of this work is the assumption of a perfectly rational adversary, which may not hold in many real world domains.

Recent work [20] developed a new algorithm COBRA, which provided a solution for designing better defender strategies against human adversaries by modeling an adversary's behavior taking into consideration (i) human deviation from the utility maximizing strategy and (ii) human anchoring bias when given limited observation of defender mixed strategy. COBRA significantly outperforms DOBSS in the experimental results against human subjects. We have provided an extensive comparison of our new approaches with COBRA in this paper.

Another line of related work in Stackelberg security games have been trying to design more robust strategies to deal with different kinds of uncertainties [41–43]. Yin et al. [42] proposed a unified efficient algorithm that addresses both execution uncertainties of the defender and observation uncertainties of adversaries in SSGs. Kiekintveld et al. [43] address payoff uncertainty by introducing a general model of infinite Bayesian Stackelberg security games which allows payoffs to be represented using continuous payoff distributions. Although the simulation based experiment showed promising result of these studies, the performances of these models against real human subjects are left unaddressed. Our work differs from these efforts in that they do not handle bounded rationality of human adversary while we propose different models to explicitly predict human decision making accounting for their bounded rationality.

Many models have been proposed to capture human bounded rationality in their decision making in psychology and cognitive science [44,34,45]. A key challenge of applying these models to game-theoretical framework to help design better strategy is the transition from a (sometimes descriptive) model to a computational model. On the other hand, there has been a growing interest in the game theory literature to develop more realistic computational models of human decision making in games [9,11,32]. Most of these models find empirical support from the data of human playing games. However, few research efforts have attempted to bridge the gap between computational game theory and behavioral game theory, and fewer still in the important area of SSGs, which is the topic of this article. In this paper, we explore such method by developing computation models of human adversary decision making based on two descriptive theories: Prospect Theory and rank-dependent Expected Utility. The outperforming result of these new models provide a clear direction for future work in further improve the models of human bounded rationality in security games.

Outside the area of Stackelberg security games, there have been several recent investigations of human subjects interacting with agents. For example, Melo et al. [46] investigate the impact of expression of an automated agent's anger or happiness in how a human participant may play the game. In repeated prisoner's dilemma games, agents' expressions are shown to significantly affect human subjects' cooperation or defection. Similarly, Azaria et al. [47] focus on road selection games, and advice an automated system may provide to human subjects; Peled et al. [48] focus on bilateral bargaining games, designing agents that negotiate proficiently with people. Aside from the obvious difference that our focus is on SSGs, another key is our focus on efficiently computing optimal mixed strategies for the defender.

8. Summary and future work

There is a significant interest in game-theoretic techniques to solve security problems. Several real-world application based on using these techniques have been deployed across the nation, including ARMOR [5], IRIS [6], GUARDS [7] and PROTECT [8]. These systems have adopted the traditional game-theoretical assumption of perfectly rational adversaries. While this was appropriate in the first generation of Stackelberg security game applications, it is now critical to develop new

methods to compute defender strategies addressing human bounded rationality — particularly as the range of SSG applications where these deployed systems will face human adversaries continue to grow. New methods need to be developed to compute defender strategy against bounded rationality of real human adversaries.

In this paper, we address this problem by applying two fundamental theories in human decision-making, Prospect Theory (PT) and Quantal Response Equilibrium (QRE), to model the adversary's behavior in security games. The contributions of this article include:

(i) proposing two mathematical models of adversary's decision-making based on using PT;
(ii) providing a method to adapt the parameters of PT for the two PT-based methods;
(iii) proposing one mathematical model of adversary's decision-making based on quantal response (QR) and a second one that uses modified QR (based on rank dependent utility);
(iv) developing efficient algorithms to compute optimal strategy for the defender under each of these four adversary models;
(v) most extensive-to-date experiment to verify the effectiveness of the proposed approaches.

We compare our new approaches to three benchmark algorithms, DOBSS, MAXIMIN and COBRA in seven different payoff structures, where COBRA is the leading contender for addressing human bounded rationality presented in previous work. Experiment results show that our new methods based on using the QR model achieved statistically significant better performance than all three benchmark algorithms as well as the PT-based new methods. Furthermore, we identify that in cases where the targets covered with minimum resource have large penalty to the defender, the modified QR model achieved significantly better performance than the basic QR model. By providing new models that better predicts the behavior of human adversary and new algorithms that computes strategies outperforming our leading competitor, this paper has advanced the state-of-the-art.

While the research reported in this article takes an important step forward in addressing bounded rationality of human adversaries in the context of security games, there are still many open topics for future research. One key area is to translate the results obtained here in controlled experiments on AMT into specific, real-world security applications. Most of the issues related to making this transition are not unique to our work, but apply more generally to studies in agent/human interactions. For example, the specific conditions tested in the lab and the way in which decisions are presented is not likely to be exactly reflected in real interactions, and neither is the population of adversaries identical to the population of adversaries in a real-world security setting. However, our methods are based on fundamental features of human decision-making that are robustly supported in a large number of behavioral studies and these methods would thus translate into real-world applications. In addition, the parameters offer some ability to tune the models over time to specific settings or populations of interest, and our methodology provides techniques for tuning these parameters. The parameter settings in our work can serve as initial settings in a real deployment to be adapted over time. Alternatively, the parameters can initially be set conservatively (e.g., somewhat close to settings that result in a standard equilibrium), and adapted over time from this starting point. Another interesting possibility that could be explored in future work is to develop ways to incorporate different sources of information (such as prior knowledge of the biases of specific adversaries) into the models in a general way.

Furthermore, the current game model is an abstraction of real-world security scenario, in particular, the one at the Los Angeles international airport. The model can be further refined to reflect more details of the scenario. For example, the current game assumes covering each target with a single unit of resources and a binary effect of protecting the targets with the resources (i.e. protected/not protected). An interesting direction for future work is to explore the effect of having multiple units of resources to protect a target. At the same time, another interesting direction for future work is to extend the current game model to deal with domains with continual and online interaction between the defender and the attacker.

Acknowledgements

This research was supported by Army Research Office under the grant # 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.

Appendix A. Payoff structure information

The four payoff structures selected from the four clustering groups are displayed in Table A.9. The three payoffs that are identical to that first used by Pita et al. are shown in Table A.10. The four payoff structures selected for the second evaluation set of experiment for comparing the three QR model based strategies are listed in Table A.11.

Table A.9
Payoff structures.

Target	1	2	3	4	5	6	7	8
				(a) Payoff structure 1.1				
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 penalty	−7	−4	−6	−8	−4	−2	−9	−3
				(b) Payoff structure 1.2				
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 penalty	−10	−1	−10	−8	−4	−10	−5	−3
				(c) Payoff structure 1.3				
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 penalty	−6	−9	−3	−7	−7	−2	−5	−2
				(d) Payoff structure 1.4				
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 penalty	−4	−8	−5	−8	−9	−4	−1	−6

Table A.10
Payoff structures.

Target	1	2	3	4	5	6	7	8
				(a) Payoff structure 1.5				
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 penalty	−2	−4	−3	−3	−3	−2	−4	−3
				(b) Payoff structure 1.6				
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 penalty	−3	−2	−3	−2	−3	−3	−2	−3
				(c) Payoff structure 1.7				
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 penalty	−3	−3	−3	−3	−3	−3	−3	−3

Table A.11
Payoff structures.

Target	1	2	3	4	5	6	7	8
				(c) Payoff structure 2.1				
defender reward	10	1	1	8	8	6	2	4
defender penalty	−4	−6	−9	−10	−7	−4	−5	−8
subject reward	7	6	1	6	7	7	6	6
subject penalty	−4	−3	−4	−8	−10	−5	−4	−5
				(b) Payoff structure 2.2				
defender reward	2	7	1	10	1	10	3	2
defender penalty	−6	−4	−5	−1	−7	−4	−4	−7
subject reward	7	6	1	10	6	3	2	8
subject penalty	−6	−6	−6	−2	−2	−9	−10	−3
				(c) Payoff structure 2.3				
defender reward	1	1	10	7	4	9	6	9
defender penalty	−1	−7	−8	−6	−7	−1	−8	−7
subject reward	2	1	1	4	1	5	6	7
subject penalty	−10	−1	−4	−10	−6	−2	−1	−8

(*continued on next page*)

Table A.11 (*continued*)

Target	1	2	3	4	5	6	7	8
				(d) Payoff structure 2.4				
defender reward	7	3	6	1	10	1	8	9
defender penalty	−2	−1	−5	−4	−5	−8	−8	−10
subject reward	7	1	6	1	3	1	7	2
subject penalty	−9	−1	−10	−7	−3	−1	−5	−1

Appendix B. Defender mixed-strategy

The defender's mixed-strategy from each algorithm in each payoff structures are displayed in Tables B.12–B.19.

Table B.12
Defender's mixed-strategy for payoff structure 1.1.

Target	1	2	3	4	5	6	7	8
DOBSS	0.49	0.53	0.15	0.36	0.44	0.59	0.37	0.07
MAXIMIN	0.74	0.59	0.24	0.06	0.52	0.34	0.18	0.32
COBRA	0.57	0.62	0.18	0.22	0.51	0.44	0.34	0.11
BRPT-E	0.39	0.51	0.17	0.26	0.43	0.70	0.26	0.28
RPT-E	0.44	0.58	0.24	0.16	0.51	0.41	0.28	0.38
BRPT-L	0.53	0.54	0.10	0.36	0.42	0.60	0.39	0.06
RPT-L	0.60	0.63	0.20	0.17	0.52	0.41	0.29	0.18
BRQR-76	0.57	0.58	0.18	0.21	0.51	0.47	0.30	0.18
BRQR-55	0.58	0.59	0.18	0.19	0.52	0.47	0.28	0.20
BRQRRU	0.55	0.56	0.25	0.20	0.48	0.45	0.28	0.25

Table B.13
Defender's mixed-strategy for payoff structure 1.2.

Target	1	2	3	4	5	6	7	8
DOBSS	0.42	0.78	0.08	0.47	0.64	0	0.60	0
MAXIMIN	0.75	0.18	0.34	0.08	0.49	0.45	0.30	0.40
COBRA	0.48	0.53	0.09	0.43	0.74	0	0.70	0.02
BRPT-E	0.28	0.93	0.07	0.34	0.59	0.05	0.52	0.23
RPT-E	0.31	0.61	0.09	0.38	0.64	0.07	0.56	0.34
BRPT-L	0.43	0.78	0.04	0.48	0.65	0.01	0.61	0.01
RPT-L	0.51	0.49	0.09	0.39	0.71	0.03	0.69	0.09
BRQR-76	0.54	0.52	0.21	0.36	0.64	0.16	0.58	0
BRQR-55	0.56	0.50	0.23	0.34	0.63	0.19	0.55	0
BRQRRU	0.46	0.40	0.30	0.25	0.54	0.30	0.45	0.30

Table B.14
Defender's mixed-strategy for payoff structure 1.3.

Target	1	2	3	4	5	6	7	8
DOBSS	0.53	0.37	0.12	0.25	0.06	0.72	0.31	0.64
MAXIMIN	0.12	0.48	0.24	0.54	0.31	0.63	0.58	0.10
COBRA	0.48	0.42	0.16	0.29	0.07	0.81	0.36	0.42
BRPT-E	0.42	0.24	0.29	0.19	0.09	0.78	0.28	0.72
RPT-E	0.41	0.29	0.41	0.25	0.14	0.78	0.36	0.36
BRPT-L	0.58	0.39	0.09	0.21	0.05	0.76	0.28	0.65
RPT-L	0.36	0.47	0.27	0.32	0.11	0.75	0.40	0.32
BRQR-76	0.36	0.43	0.20	0.36	0.13	0.72	0.43	0.37
BRQR-55	0.34	0.43	0.22	0.37	0.12	0.73	0.44	0.36
BRQRRU	0.33	0.38	0.33	0.33	0.33	0.67	0.35	0.28

Table B.15
Defender's mixed-strategy for payoff structure 1.4.

Target	1	2	3	4	5	6	7	8
DOBSS	0.22	0.37	0.19	0.44	0.05	0.58	0.69	0.47
MAXIMIN	0.59	0.21	0.41	0.36	0.44	0.63	0.080	0.29

Table B.15 (*continued*)

Target	1	2	3	4	5	6	7	8
COBRA	0.24	0.42	0.21	0.50	0.04	0.66	0.39	0.53
BRPT-E	0.28	0.27	0.22	0.33	0.08	0.54	0.90	0.38
RPT-E	0.37	0.31	0.29	0.37	0.10	0.60	0.53	0.43
BRPT-L	0.16	0.37	0.14	0.48	0.04	0.60	0.73	0.48
RPT-L	0.25	0.43	0.21	0.53	0.07	0.66	0.35	0.50
BRQR-76	0.35	0.33	0.30	0.44	0.20	0.62	0.36	0.42
BRQR-55	0.37	0.32	0.32	0.41	0.23	0.62	0.33	0.40
BRQRRU	0.34	0.34	0.34	0.39	0.34	0.58	0.29	0.38

Table B.16
Defender's mixed-strategy for payoff structure 1.5.

Target	1	2	3	4	5	6	7	8
DOBSS	0	0.59	0.45	0.51	0.56	0	0.62	0.27
MAXIMIN	0.56	0.53	0	0.49	0.37	0	0.45	0.60
COBRA	0	0.64	0.23	0.63	0.52	0	0.55	0.40
BRPT-E	0.16	0.49	0.41	0.46	0.51	0.16	0.52	0.28
RPT-E	0.28	0.53	0.12	0.52	0.48	0.18	0.53	0.36
BRPT-L	0.02	0.61	0.43	0.50	0.57	0.02	0.65	0.21
RPT-L	0.13	0.62	0.13	0.60	0.49	0.13	0.53	0.37
BRQR-76	0.12	0.61	0.16	0.55	0.52	0	0.57	0.46
BRQR-55	0.13	0.62	0.12	0.56	0.52	0	0.58	0.48
BRQRRU	0.15	0.60	0.10	0.52	0.49	0.15	0.56	0.41

Table B.17
Defender's mixed-strategy for payoff structure 1.6.

Target	1	2	3	4	5	6	7	8
DOBSS	0.56	0.45	0.19	0.68	0	0.19	0.65	0.30
MAXIMIN	0.53	0.64	0	0.49	0	0.27	0.59	0.48
COBRA	0.58	0.55	0	0.53	0	0.31	0.62	0.41
BRPT-E	0.49	0.46	0.21	0.67	0.05	0.21	0.64	0.29
RPT-E	0.56	0.56	0.01	0.54	0.01	0.31	0.63	0.38
BRPT-L	0.58	0.43	0.15	0.73	0	0.15	0.69	0.26
RPT-L	0.58	0.56	0	0.54	0	0.28	0.63	0.40
BRQR-76	0.58	0.59	0	0.60	0	0.19	0.66	0.38
BRQR-55	0.59	0.60	0	0.61	0	0.16	0.67	0.37
BRQRRU	0.58	0.59	0.05	0.60	0	0.16	0.66	0.36

Table B.18
Defender's mixed-strategy for payoff structure 1.7.

Target	1	2	3	4	5	6	7	8
DOBSS	0.59	0.44	0.10	0.65	0	0.25	0.62	0.36
MAXIMIN	0.49	0.36	0	0.52	0.48	0.17	0.49	0.48
COBRA	0.57	0.48	0	0.59	0	0.33	0.56	0.47
BRPT-E	0.54	0.41	0.19	0.60	0.09	0.27	0.57	0.35
RPT-E	0.54	0.44	0	0.57	0.18	0.30	0.54	0.43
BRPT-L	0.61	0.42	0.08	0.70	0.01	0.20	0.66	0.32
RPT-L	0.52	0.41	0	0.55	0.21	0.25	0.52	0.53
BRQR-76	0.59	0.44	0	0.63	0.08	0.22	0.60	0.45
BRQR-55	0.59	0.44	0	0.64	0.08	0.20	0.61	0.45
BRQRRU	0.59	0.44	0	0.64	0.08	0.20	0.61	0.45

Table B.19
Defender's mixed-strategy.

Target	1	2	3	4	5	6	7	8
(a) Payoff structure 2.1								
BRQRRU	0.30	0.45	0.35	0.38	0.35	0.35	0.39	0.42
BRQR-55	0.31	0.50	0.17	0.42	0.36	0.34	0.43	0.47
BRQR-76	0.31	0.52	0.13	0.41	0.36	0.36	0.45	0.47

(*continued on next page*)

Table B.19 (*continued*)

Target	1	2	3	4	5	6	7	8
				(b) Payoff structure 2.2				
BRQRRU	0.38	0.35	0.35	0.30	0.43	0.35	0.35	0.51
BRQR-55	0.51	0.39	0.07	0.36	0.63	0.22	0.16	0.66
BRQR-76	0.51	0.40	0.05	0.41	0.63	0.20	0.14	0.65
				(c) Payoff structure 2.3				
BRQRRU	0.32	0.32	0.32	0.32	0.32	0.27	0.66	0.44
BRQR-55	0.10	0.38	0.36	0.36	0.30	0.32	0.71	0.47
BRQR-76	0.15	0.38	0.32	0.35	0.27	0.33	0.71	0.47
				(d) Payoff structure 2.4				
BRQRRU	0.26	0.31	0.37	0.31	0.36	0.31	0.52	0.56
BRQR-55	0.29	0	0.40	0.19	0.42	0.47	0.55	0.69
BRQR-76	0.29	0	0.40	0.20	0.41	0.50	0.54	0.65

Appendix C. Distribution of subjects' choices

The distributions of subjects' choices while playing against each payoff/strategy combination are displayed in Tables C.20–C.27.

Table C.20
Distribution of subjects' choices (%) in payoff structure 1.1.

Target	1	2	3	4	5	6	7	8
DOBSS	17.44	5.81	6.98	1.16	4.65	23.26	4.65	36.05
MAXIMIN	5.81	3.49	2.33	50.00	0.00	27.91	10.47	0.00
COBRA	8.14	2.33	1.16	26.74	0.00	31.40	10.47	19.77
BRPT-E	23.26	8.14	22.09	4.65	3.49	16.28	19.77	2.33
RPT-E	11.63	1.16	2.33	43.02	1.16	38.37	2.33	0.00
BRPT-L	8.14	4.65	24.42	5.81	8.14	22.09	5.81	20.93
RPT-L	4.65	5.81	2.33	45.35	5.81	18.60	2.33	15.12
BRQR-76	3.49	9.30	8.14	30.23	1.16	32.56	9.30	5.81
BRQR-55	5.81	1.16	9.30	36.05	2.33	29.07	9.30	6.98
BRQRRU	8.14	2.33	1.16	36.05	2.33	33.72	8.14	8.14

Table C.21
Distribution of subjects' choices (%) in payoff structure 1.2.

Target	1	2	3	4	5	6	7	8
DOBSS	13.33	11.11	4.44	3.33	15.56	7.78	17.78	26.67
MAXIMIN	2.22	30.00	1.11	31.11	12.22	0.00	22.22	1.11
COBRA	8.89	22.22	11.11	14.44	10.00	14.44	12.22	6.67
BRPT-E	23.33	10.00	8.89	14.44	13.33	8.89	20.00	1.11
RPT-E	20.00	25.56	7.78	10.00	12.22	8.89	15.56	0.00
BRPT-L	13.33	13.33	14.44	4.44	14.44	5.56	16.67	17.78
RPT-L	8.89	40.00	8.89	4.44	13.33	7.78	11.11	5.56
BRQR-76	1.11	21.11	4.44	17.78	15.56	2.22	16.67	21.11
BRQR-55	0.00	35.56	0.00	14.44	6.67	5.56	17.78	20.00
BRQRRU	2.22	32.22	1.11	35.56	8.89	1.11	17.78	1.11

Table C.22
Distribution of subjects' choices (%) in payoff structure 1.3.

Target	1	2	3	4	5	6	7	8
DOBSS	23.08	10.99	18.68	2.20	12.09	19.78	5.49	7.69
MAXIMIN	41.76	1.10	0.00	0.00	0.00	20.88	0.00	36.26
COBRA	23.08	8.79	10.99	3.30	8.79	17.58	3.30	24.18
BRPT-E	31.87	17.58	4.40	6.59	10.99	19.78	6.59	2.20
RPT-E	24.18	20.88	0.00	3.30	8.79	16.48	2.20	24.18
BRPT-L	19.78	10.99	23.08	5.49	10.99	15.38	7.69	6.59
RPT-L	46.15	1.10	13.19	3.30	12.09	12.09	4.40	7.69

Table C.22 (*continued*)

Target	1	2	3	4	5	6	7	8
BRQR-76	31.87	1.10	4.40	2.20	14.29	10.99	0.00	35.16
BRQR-55	38.46	2.20	3.30	0.00	10.99	18.68	0.00	26.37
BRQRRU	32.97	0.00	0.00	2.20	0.00	15.38	0.00	49.45

Table C.23
Distribution of subjects' choices (%) in payoff structure 1.4.

Target	1	2	3	4	5	6	7	8
DOBSS	5.43	6.52	8.70	10.87	21.74	20.65	22.83	3.26
MAXIMIN	0.00	7.61	0.00	7.61	0.00	8.70	70.65	5.43
COBRA	4.35	5.43	5.43	11.96	15.22	10.87	41.30	5.43
BRPT-E	3.26	14.13	4.35	17.39	19.57	18.48	13.04	9.78
RPT-E	0.00	10.87	2.17	22.83	14.13	9.78	38.04	2.17
BRPT-L	10.87	14.13	10.87	13.04	15.22	10.87	22.83	2.17
RPT-L	3.26	2.17	8.70	9.78	17.39	7.61	48.91	2.17
BRQR-76	0.00	11.96	0.00	9.78	13.04	11.96	53.26	0.00
BRQR-55	0.00	8.70	1.09	13.04	16.30	8.70	48.91	3.26
BRQRRU	1.09	2.17	0.00	10.87	1.09	10.87	67.39	6.52

Table C.24
Distribution of subjects' choices (%) in payoff structure 1.5.

Target	1	2	3	4	5	6	7	8
DOBSS	14.77	5.68	13.64	1.14	7.95	17.05	28.41	11.36
MAXIMIN	1.14	1.14	47.73	2.27	10.23	1.14	36.36	0.00
COBRA	7.95	1.14	40.91	0.00	7.95	11.36	30.68	0.00
BRPT-E	12.50	19.32	1.14	4.55	1.14	6.82	38.64	15.91
RPT-E	0.00	6.82	54.55	3.41	3.41	1.14	30.68	0.00
BRPT-L	11.36	6.82	13.64	5.68	3.41	12.50	25.00	21.59
RPT-L	6.82	4.55	54.55	2.27	1.14	7.95	22.73	0.00
BRQR-76	3.41	2.27	38.64	0.00	4.55	19.32	31.82	0.00
BRQR-55	0.00	1.14	53.41	0.00	5.68	12.50	27.27	0.00
BRQRRU	0.00	1.14	51.14	1.14	9.09	3.41	34.09	0.00

Table C.25
Distribution of subjects' choices (%) in payoff structure 1.6.

Target	1	2	3	4	5	6	7	8
DOBSS	8.33	11.67	13.33	28.33	18.33	8.33	6.67	5.00
MAXIMIN	3.33	0.00	41.67	50.00	1.67	3.33	0.00	0.00
COBRA	0.00	1.67	46.67	43.33	3.33	3.33	0.00	1.67
BRPT-E	26.67	1.67	10.00	25.00	20.00	10.00	5.00	1.67
RPT-E	8.33	1.67	40.00	35.00	3.33	1.67	6.67	3.33
BRPT-L	5.00	13.33	15.00	26.67	13.33	3.33	6.67	16.67
RPT-L	0.00	1.67	13.33	63.33	15.00	3.33	3.33	0.00
BRQR-76	3.33	0.00	45.00	43.33	3.33	3.33	1.67	0.00
BRQR-55	3.33	0.00	45.00	41.67	1.67	5.00	1.67	1.67
BRQRRU	3.33	0.00	30.00	45.00	15.00	1.67	3.33	1.67

Table C.26
Distribution of subjects' choices (%) in payoff structure 1.7.

Target	1	2	3	4	5	6	7	8
DOBSS	6.10	10.98	19.51	23.17	19.51	12.20	1.22	7.32
MAXIMIN	7.32	9.76	32.93	19.51	1.22	9.76	19.51	0.00
COBRA	2.44	8.54	45.12	24.39	3.66	6.10	8.54	1.22
BRPT-E	6.10	9.76	12.20	32.93	18.29	9.76	4.88	6.10
RPT-E	6.10	12.20	39.02	26.83	0.00	1.22	8.54	6.10
BRPT-L	12.20	10.98	14.63	18.29	14.63	13.41	4.88	10.98
RPT-L	17.07	1.22	21.95	23.17	0.00	21.95	14.63	0.00
BRQR-76	7.32	13.41	41.46	17.07	1.22	7.32	12.20	0.00
BRQR-55	6.10	13.41	37.80	28.05	1.22	6.10	7.32	0.00
BRQRRU	4.88	12.20	29.27	23.17	4.88	18.29	3.66	3.66

Table C.27
Distribution of subjects' choices (%).

Target	1	2	3	4	5	6	7	8
				(a) Payoff structure 2.1				
BRQRRU	72.94	10.59	0.00	2.36	3.53	9.41	1.18	0.00
BRQR-55	54.12	16.47	10.59	1.18	7.06	7.06	2.35	1.18
BRQR-76	58.82	9.41	11.76	3.53	5.88	8.24	2.35	0.00
				(b) Payoff structure 2.2				
BRQRRU	2.35	3.53	0.00	84.71	1.18	0.00	3.53	4.71
BRQR-55	2.35	5.88	10.59	68.24	1.18	2.35	1.18	8.24
BRQR-76	2.35	5.88	16.47	57.65	3.53	7.06	2.35	4.71
				(c) Payoff structure 2.3				
BRQRRU	0.00	4.71	1.18	2.35	0.00	67.06	15.29	9.41
BRQR-55	14.12	2.35	2.35	2.35	0.00	43.53	22.35	12.94
BRQR-76	14.12	2.35	0.00	1.18	1.18	48.24	16.47	16.47
				(d) Payoff structure 2.4				
BRQRRU	49.41	9.41	8.24	2.35	4.71	5.88	15.29	4.71
BRQR-55	35.29	35.29	2.35	1.18	5.88	3.53	16.47	0.00
BRQR-76	41.18	29.41	4.71	0.00	7.06	1.18	15.29	1.18

References

[1] R. Yang, C. Kiekintveld, F. Ordonez, M. Tambe, R. John, Improving resource allocation strategy against human adversaries in security games, in: IJCAI, 2011, pp. 458–464.
[2] N. Gatti, Game theoretical insights in strategic patrolling: Model and algorithm in normal-form, in: ECAI-08, 2008, pp. 403–407.
[3] N. Agmon, S. Kraus, G.A. Kaminka, Multi-robot perimeter patrol in adversarial settings, in: ICAT, 2008, pp. 2339–2345.
[4] N. Basiloco, N. Gatti, F. Amigoni, Leader-follower strategies for robotic patrolling in environments with arbitrary topologies, in: AAMAS, 2009, pp. 57–64.
[5] J. Pita, M. Jain, F. Ordonez, C. Portway, M. Tambe, C. Western, P. Paruchuri, S. Kraus, Deployed armor protection: The application of a game theoretic model for security at the Los Angeles international airport, in: AAMAS, 2008, pp. 125–132.
[6] J. Tsai, S. Rathi, C. Kiekintveld, F. Ordonez, M. Tambe, IRIS – a tool for strategic security allocation in transportation networks, in: AAMAS, 2009, pp. 37–44.
[7] J. Pita, M. Tambe, C. Kiekintveld, S. Cullen, E. Steigerwald, GUARDS – game theoretic security allocation on a national scale, in: AAMAS, 2011, pp. 37–44.
[8] B. An, J. Pita, E. Shieh, M. Tambe, C. Kiekintveld, J. Marecki, GUARDS and PROTECT: Next generation applications of security games, ACM SIGecom Exchanges 10 (2011) 31–34.
[9] C.F. Camerer, T. Ho, J. Chongn, A cognitive hierarchy model of games, QJE 119 (2004) 861–898.
[10] M. Costa-Gomes, V.P. Crawford, B. Broseta, Cognition and behavior in normal-form games: An experimental study, Econometrica 69 (2001) 1193–1235.
[11] S. Ficici, A. Pfeffer, Simultaneously modeling humans' preferences and their beliefs about others' preferences, in: AAMAS, 2008, pp. 323–330.
[12] Y. Gal, A. Pfeffer, Modeling reciprocal behavior in human bilateral negotiation, in: AAAI, 2007
[13] M. Tambe, Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned, Cambridge University Press, New York, NY, 2011.
[14] D. Korzhyk, V. Conitzer, R. Parr, Security games with multiple attacker resources, in: IJCAI, 2011, pp. 273–279.
[15] J. Letchford, Y. Vorobeychik, Computing randomized security strategies in networked domains, in: AARM Workshop in AAAI, 2011.
[16] D. Kahneman, A. Tvesky, Prospect theory: An analysis of decision under risk, Econometrica 47 (1979) 263–292.
[17] R.D. McKelvey, T.R. Palfrey, Quantal response equilibria for normal form games, Games and Economic Behavior 2 (1995) 6–38.
[18] P. Paruchuri, J.P. Pearce, J. Marecki, M. Tambe, F. Ordonez, S. Kraus, Playing games for security: An efficient exact algorithm for solving bayesian Stackelberg games, in: AAMAS, 2008, pp. 895–902.
[19] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordonez, M. Tambe, Computing optimal randomized resource allocations for massive security games, in: AAMAS, 2009, pp. 689–696.
[20] J. Pita, M. Jain, F. Ordonez, M. Tambe, S. Kraus, Solving Stackelberg games in the real-world: Addressing bounded rationality and limited observations in human preference models, Artificial Intelligence Journal 174 (2010) 1142–1171.
[21] C. Fox, R. Clemen, Subjective probability assessment in decision analysis: Partition dependence and bias toward the ignorance prior, Management Science 51 (2005) 1417–1432.
[22] K.E. See, C.R. Fox, Y.S. Rottenstreich, Between ignorance and truth: Partition dependence and learning in judgment under uncertainty, Journal of Experimental Psychology: Learning, Memory, and Cognition 32 (2006) 1385–1402.
[23] D.L. McFadden, Econometric analysis of qualitative response models, in: Z. Griliches, M.D. Intriligator (Eds.), Handbook of Econometrics, vol. 2, Elsevier, 1984, pp. 1395–1457.
[24] K. Train, Discrete Choice Methods with Simulation, Cambridge University Press, Cambridge, UK, 2003.
[25] D.L. McFadden, Quantal choice analysis: A survey, Annals of Economic and Social Measurement 5 (1976) 369–390.
[26] D.L. McFadden, A method of simulated moments for estimation of discrete choice models without numerical integration, Econometrica 57 (1989) 995–1026.
[27] E. Diecidue, P.P. Wakker, On the intuition of rank-dependent utility, The Journal of Risk and Uncertainty 23 (2001) 281–289.
[28] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, M. Tambe, Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness, in: JAIR, 2011, pp. 297–327.
[29] D. Korzhyk, V. Conitzer, R. Parr, Complexity of computing optimal Stackelberg strategies in security resource allocation games, in: AAAI, 2010.
[30] J. Tsai, Z. Yin, J. Kwak, D. Kempe, C. Kiekintveld, M. Tambe, Urban security: Game-theoretic resource allocation in networked physical domains, in: AAAI, 2010.
[31] D. Kahneman, A. Tvesky, Advances in prospect theory: Cumulative representation of uncertainty, Journal of Risk and Uncertainty 5 (1992) 297–322.
[32] D.O. Stahl, P.W. Wilson, Experimental evidence on players' models of other players, JEBO 25 (1994) 309–327.
[33] J.R. Wright, K. Leyton-Brown, Beyond equilibrium: Predicting human behavior in normal-form games, in: AAAI, 2010.
[34] H. Simon, Rational choice and the structure of the environment, Psychological Review 63 (1956) 129–138.
[35] M.K. Sen, P.L. Stoffa, Global Optimization Methods in Geophysical Inversion, Elsevier, New York, 1995.
[36] J.C. Becsey, L. Berke, J.R. Callan, Nonlinear least squares methods: A direct grid search approach, Journal of Chemical Education 45 (1968) 728.

[37] R.R. Wilcox, Applying Contemporary Statistical Techniques, Academic Press, 2003.
[38] N. Feltovich, Reinforment-based vs. belief-based learning models in experimental asymmetric-information games, Econometrica 68 (2000) 605–641.
[39] N. Agmon, S. Kraus, G.A. Kaminka, V. Sadow, Adversarial uncertainty in multi-robot patrol, in: IJCAI, 2009, pp. 1811–1817.
[40] N. Gatti, Game theoretical insights in strategic patrolling model and algorithm in normal-form, in: ECAI, 2008, pp. 403–407.
[41] M. Aghassi, D. Bertsimas, Robust game theory, Mathematical Programming 107 (2006) 231–273.
[42] Z. Yin, M. Jain, M. Tambe, F. Ordonez, Risk-averse strategies for security games with execution and observational uncertainty, in: AAAI, 2011.
[43] C. Kiekintveld, J. Marecki, M. Tambe, Approximation methods for infinite bayesian Stackelberg games: Modeling distributional payoff uncertainty, in: AAMAS, 2011, pp. 1005–1012.
[44] R. Hastie, R.M. Dawes, Rational Choice in an Uncertain World: the Psychology of Judgement and Decision Making, Sage Publications, Thousand Oaks, 2001.
[45] C. Starmer, Developments in non-expected utility theory: The hunt for a descriptive theory of choice under risk, Journal of Economic Literature XXXVIII (2000) 332–382.
[46] C. de Melo, P. Carnevale, J. Gratch, The effect of expression of anger and happiness in computer agents on negotiations with humans, in: AAMAS, 2011, pp. 937–944.
[47] A. Azaria, Z. Rabinovich, S. Kraus, C.V. Goldman, Strategic information disclosure to people with multiple alternatives, in: AAAI, 2011, pp. 594–600.
[48] N. Peled, Y. Gal, S. Kraus, A study of computational and human strategies in revelation games, in: AAMAS, 2011, pp. 345–352.