Computing Optimal Strategy against Quantal Response in Security Games

Rong Yang[+], Fernando Ordonez[+,*], Milind Tambe[+]
[+] University of Southern California, Los Angeles, CA, US
[*] University of Chile, Santiago, Chile
[+]{yangrong,fordon,tambe}@usc.edu
[*]fordon@dii.uchile.cl

ABSTRACT

To step beyond the first-generation deployments of attacker-defender security games – for LAX Police, US FAMS and others – it is critical that we relax the assumption of perfect rationality of the human adversary. Indeed, this assumption is a well-accepted limitation of classical game theory and modeling human adversaries' bounded rationality is critical. To this end, quantal response (QR) has provided very promising results to model human bounded rationality. However, in computing optimal defender strategies in real-world security games against a QR model of attackers, we face difficulties including (1) solving a nonlinear non-convex optimization problem efficiently for massive real-world security games; and (2) addressing constraints on assigning security resources, which adds to the complexity of computing the optimal defender strategy.

This paper presents two new algorithms to address these difficulties: GOSAQ can compute the globally optimal defender strategy against a QR model of attackers when there are no resource constraints and gives an efficient heuristic otherwise; PASAQ in turn provides an efficient approximation of the optimal defender strategy with or without resource constraints. These two novel algorithms are based on three key ideas: (i) use of a binary search method to solve the fractional optimization problem efficiently, (ii) construction of a convex optimization problem through a non-linear transformation, (iii) building a piecewise linear approximation of the non-linear terms in the problem. Additional contributions of this paper include proofs of approximation bounds, detailed experimental results showing the advantages of GOSAQ and PASAQ in solution quality over the benchmark algorithm (BRQR) and the efficiency of PASAQ. Given these results, PASAQ is at the heart of the PROTECT system, which is deployed for the US Coast Guard in the port of Boston, and is now headed to other ports.

Categories and Subject Descriptors

H.4 [**Computing Methodology**]: Game Theory

General Terms

Algorithm, Security

Keywords

Game Theory, Human Behavior, Optimization, Quantal Response

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

1. INTRODUCTION

The recent real-world applications of attacker-defender Stackelberg security games, ARMOR, IRIS [7] and GUARDS [12], provide software assistants that help security agencies optimize allocations of their limited security resources. These applications require efficient algorithms that derive mixed (randomized) strategies for the defender (security agencies), taking into account an attacker's surveillance and best response. The algorithms underlying these applications [7] or most others in the literature [1, 10] have assumed perfect rationality of the human attacker, who strictly maximizes his expected utility. While this is a standard game-theoretic assumption and appropriate as an approximation in first generation applications, it is a well-accepted limitation of classical game theory [4]. Indeed, algorithmic solutions based on this assumption may not be robust to the boundedly rational decision making of a human adversary (leading to reduced expected defender reward), and may also be limited in exploiting human biases.

To address this limitation, several models have been proposed to capture human bounded rationality in game-theoretic settings [14, 5, 11]. Among these, the quantal response (QR) model [11] is an important solution concept. QR assumes errors in human decision making and suggests that instead of strictly maximizing utility, individuals respond stochastically in games: the chance of selecting a non-optimal strategy increases as the associated cost decreases. The QR model has received widespread support in the literature in terms of its superior ability to model human behavior in games [6, 14], including in recent multi-agent systems literature [17]. An even more relevant study in the context of security games showed that defender security allocations assuming a quantal response model of adversary behavior outperformed several competing models in experiments with human subjects [18]. QR is among the best-performing current models (with significant support in the literature) and one that allows tuning of the 'adversary rationality level' as explained later. Hence this model is one that can be practically used by security agencies desiring to not be locked into adversary models of perfect rationality.

Unfortunately, in computing optimal defender strategies in security games assuming an adversary with quantal response (QR-adversary), we face two major difficulties: (1) solving a nonlinear non-convex optimization problem efficiently for massive real-world security games; and (2) addressing resource assignment constraints in security games, which adds to the complexity of computing the optimal defender strategy. Yet, scaling-up to massive security problems and handling constraints on resource assignments are essential to address real-world problems such as computing strategies for Federal Air Marshals Service (FAMS) [7] and the US Coast Guard (USCG) [13].

Yang et al. [18] introduced the algorithm BRQR to solve a Stackelberg security game with a QR-adversary. BRQR however was not guaranteed to converge to the optimal solution, as it used a nonlinear solver with multi-starts to obtain an efficient solution to a non-convex optimization problem. Furthermore, that work did not consider resource assignment constraints that are included in this paper. Nevertheless we compare the performance of the proposed algorithms against BRQR, since it is the benchmark algorithm. Another existing algorithm that efficiently computes the Quantal Response Equilibrium [15] only applies to cases where all the players have the same level of errors in their quantal response, a condition not satisfied in security games. In particular, in security games, the defender's strategy is based on a computer-aided decision-making tool, and therefore it is a best response. Adversaries, on the other hand, are human beings who may have biases and preferences in their decision making, so they are modeled with a quantal response. Therefore, new algorithms need to be developed to compute the optimal defender strategy when facing a QR-adversary in real-world security problems.

In this paper, we provide the following five contributions. First, we provide an algorithm called GOSAQ to compute the defender optimal strategy against a QR-adversary. GOSAQ uses a binary search method to iteratively estimate the global optimal solution rather than searching for it directly, which would require solving a nonlinear and non-convex fractional problem. It also uses a nonlinear variable transformation to convert the problem into a convex problem. GOSAQ leads to a ε-optimal solution, where ε can be arbitrarily small. Second, we provide another algorithm called PASAQ to approximate the optimal defender strategy. PASAQ is also based on binary search. It then converts the problem into a Mixed-Integer Linear Programming problem by using a piecewise linear approximation. PASAQ leads to an efficient approximation of the global optimal defender strategy and provides an arbitrarily near-optimal solution with a sufficiently accurate linear approximation. Third, we show that both GOSAQ and PASAQ can not only solve problems without resource assignment constraints, such as for the LAX police[7], but also problems with resource assignment constraints, such as problems for FAMS [7] and USCG [13]. Fourth, we provide the correctness/approximation-bound proof of GOSAQ and PASAQ. Fifth, we provide detailed experimental analysis on the solution quality and computational efficiency of GOSAQ and PASAQ, illustrating that both GOSAQ and PASAQ achieve better solution quality and runtime scalability than the previous benchmark algorithm BRQR [18]. Indeed, PASAQ can potentially be applied to most of the real-world deployments of the Stackelberg Security Game, including ARMOR and IRIS [7] that are based on a perfect rationality model of the adversary. This should improve the performances of such systems when dealing with human adversaries. In fact, PASAQ is at the heart of the PROTECT system [13] deployed by the US Coast Guard at the port of Boston and that is now headed to other ports in the US.

2. PROBLEM STATEMENT

We consider a Stackelberg Security Game [7, 18, 9] (SSG) with a single leader and at least one follower, where the defender plays the role of the leader and the adversary plays the role of the follower. The defender and attacker may represent organizations and need not be single individuals. 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 $\mathcal{T} = \{1, \ldots, |\mathcal{T}|\}$. The outcomes of the SSG depend only on whether or not the attack is successful. So given a target 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. Note that a key property of SSG is that while the games may be non-zero-sum, $R_i^d > P_i^d$ and $R_i^a > P_i^a$, $\forall i$ [9]. In other words, adding resources to cover a target helps the defender and hurts the attacker.

Table 1: Notations used in this paper

$\mathcal{T}$	Set of targets; $i \in \mathcal{T}$ denotes target i
x_i	Probability that target i is covered by a resource
R_i^d	Defender reward for covering i if it's attacked
P_i^d	Defender penalty on not covering i if it's attack
R_i^a	Attacker reward for attacking i if it's not covered
P_i^a	Attacker penalty on attacking i if it's covered
$\mathcal{A}$	Set of defender strategies; $A_j \in \mathcal{A}$ denotes j^{th} strategy
a_j	Probability for defender to choose strategy A_j
M	Total number of resources

We denote the j^{th} individual defender strategy as A_j, which is an assignment of all the security resources. Generally, we could represent A_j as a column vector $A_j = \langle A_{ij} \rangle^T$, where A_{ij} indicates whether or not target i is covered by assignment j. Let $\mathcal{A} = \{A_j\}$ be the set of feasible assignments of resources and let a_j be the probability of selecting strategy j. Given this probability of selecting defender strategies we can compute the likelihood of protecting any specific target i as the *marginal* $x_i = \sum_{A_j \in \mathcal{A}} a_j A_{ij}$. The marginals x_i clearly sum to M, the total number of resources [8, 18]. Previous work [7] has shown that defender strategies in SSGs can be represented in terms of these marginals, leading to more concise equivalent representations. In particular, the defender's expected utility if the adversary attacks target i can be written as:

$$U_i^d(x_i) = x_i R_i^d + (1 - x_i) P_i^d$$

and the adversary's expected utility on attacking target i is

$$U_i^a(x_i) = x_i P_i^a + (1 - x_i) R_i^a$$

These marginal coverage vectors can be converted to a mixed strategy over actual defender strategies when there are no resource constraints [8], such as in ARMOR [7].

In the presence of constraints on assignments of resources, we may end up with marginals that cannot be converted to probabilities over individual strategies [8]. However, as Section 2.2 shows, we can address this difficulty if we have a complete description of defender strategies set $\mathcal{A}$. In this case we can add constraints enforcing that the marginals are obtained from a convex combination of these feasible defender strategies.

In SSGs, our goal is to compute a mixed strategy for the leader to commit to based on her knowledge of the adversary's response. More specifically, given that the defender has limited resources (e.g., she may need to protect 8 targets with 3 guards), she must design her strategy to optimize against the adversary's response to maximize effectiveness.

2.1 Optimal Strategy against Quantal Response

In this work, we assume a QR-adversary, i.e. with a quantal response $\langle q_i, i \in \mathcal{T} \rangle$ [11] to the defender's mixed strategy $\boldsymbol{x} = \langle x_i, i \in \mathcal{T} \rangle$. The value q_i is the probability that adversary attacks target i, computed as

$$q_i(\boldsymbol{x}) = \frac{e^{\lambda U_i^a(x_i)}}{\sum_{k \in \mathcal{T}} e^{\lambda U_k^a(x_k)}} \tag{1}$$

where $\lambda \geq 0$ is the parameter of the quantal response model [11], which represents the error level in adversary's quantal response. Simultaneously, the defender maximizes her utility (given her computer-aided decision making tool):

$$U^d(\boldsymbol{x}) = \sum_{i \in \mathcal{T}} q_i(\boldsymbol{x}) U_i^d(x_i)$$

Therefore, in domains without constraints on assigning the resources, the problem of computing the optimal defender strategy against a QR-adversary can be written in terms of marginals as:

$$\texttt{P1:} \begin{cases} \max\limits_{\boldsymbol{x}} \dfrac{\sum_{i \in \mathcal{T}} e^{\lambda R_i^a} e^{-\lambda(R_i^a - P_i^a)x_i}((R_i^d - P_i^d)x_i + P_i^d)}{\sum_{i \in \mathcal{T}} e^{\lambda R_i^a} e^{-\lambda(R_i^a - P_i^a)x_i}} \\ \text{s.t. } \sum\limits_{i \in \mathcal{T}} x_i \leq M \\ 0 \leq x_i \leq 1, \quad \forall i \in \mathcal{T} \end{cases}$$

Problem P1 has a polyhedral feasible region and is a non-convex fractional objective function.

2.2 Resource Assignment Constraint

In many real world security problems, there are constraints on assigning the resources. For example, in the FAMS problem [7], an air marshal is scheduled to protect 2 flights (targets) out of M total flights. The total number of possible schedule is $\binom{M}{2}$. However, not all of the schedules are feasible, since the flights scheduled for an air marshal have to be connected, e.g. an air marshal cannot be on a flight from A to B and then on a flight C to D. A resource assignment constraint implies that the feasible assignment set $\mathcal{A}$ is restricted; not all combinatorial assignment of resources to targets are allowed. Hence, the marginals on targets, $\boldsymbol{x}$, are also restricted.

Definition 1. *We consider a marginal coverage* $\mathbf{x}$ *to be feasible if and only if there exists* $a_j \geq 0, A_j \in \mathcal{A}$ *such that* $\sum_{A_j \in \mathcal{A}} a_j = 1$ *and for all* $i \in \mathcal{T}$, $x_i = \sum_{A_j \in \mathcal{A}} a_j A_{ij}$.

In fact, $\langle a_j \rangle$ is the mixed strategy over all the feasible assignments of the resources. In order to compute the defender's optimal strategies against a QR-adversary in the presence of resource-assignment constraints, we need to solve P2. The constraints in P1 are modified to enforce feasibility of the marginal coverage.

$$\texttt{P2:} \begin{cases} \max\limits_{\boldsymbol{x},a} \dfrac{\sum_{i \in \mathcal{T}} e^{\lambda R_i^a} e^{-\lambda(R_i^a - P_i^a)x_i}((R_i^d - P_i^d)x_i + P_i^d)}{\sum_{i \in \mathcal{T}} e^{\lambda R_i^a} e^{-\lambda(R_i^a - P_i^a)x_i}} \\ \text{s.t. } \sum\limits_{i \in \mathcal{T}} x_i \leq M \\ x_i = \sum\limits_{A_j \in \mathcal{A}} a_j A_{ij}, \quad \forall i \in \mathcal{T} \\ \sum\limits_{A_j \in \mathcal{A}} a_j = 1 \\ 0 \leq a_j \leq 1, \quad \forall A_j \in \mathcal{A} \end{cases}$$

3. BINARY SEARCH METHOD

We need to solve P1 and P2 to compute the optimal defender strategy, which requires optimally solving a non-convex problem which is in general an NP-hard problem [16]. In this section, we describe the basic structure of using a binary search method to solve the two problems. However, further efforts are required to convert this skeleton into actual efficiently runnable algorithms. We will fill in the additional details in the next two sections.

For notational simplicity, we first define the symbols $\forall\ i \in \mathcal{T}$ in Table 2. We then denote the numerator and denominator of the objective function in P1 and P2 by $N(\boldsymbol{x})$ and $D(\boldsymbol{x})$:

Table 2: Symbols for Targets in SSG

$\theta_i := e^{\lambda R_i^a} > 0$	$\beta_i := \lambda(R_i^a - P_i^a) > 0$	$\alpha_i := R_i^d - P_i^d > 0$

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

The key idea of the binary search method is to iteratively estimate the global optimal value (p^*) of the fractional objective function of P1, instead of searching for it directly. Let $\mathscr{X}_f$ be the feasible region of P1 (or P2). Given a real value r, we can know whether or not $r \leq p^*$ by checking

$$\exists \boldsymbol{x} \in \mathscr{X}_f, \text{ s.t. } rD(\boldsymbol{x}) - N(\boldsymbol{x}) \leq 0 \qquad (2)$$

We now justify the correctness of the binary search method to solve any generic fractional programming problem $\max_{\boldsymbol{x} \in \mathcal{X}_f} N(\boldsymbol{x})/D(\boldsymbol{x})$ for any functions $N(\boldsymbol{x})$ and $D(\boldsymbol{x}) > 0$.

Lemma 1. *For any real value* $r \in \mathscr{R}$, *one of the following two conditions holds.*

(a) $r \leq p^* \Longleftrightarrow \exists \mathbf{x} \in \mathscr{X}_f$, *s.t.*, $rD(\mathbf{x}) - N(\mathbf{x}) \leq 0$

(b) $r > p^* \Longleftrightarrow \forall \mathbf{x} \in \mathscr{X}_f$, $rD(\mathbf{x}) - N(\mathbf{x}) > 0$

PROOF. We only prove (a) as (b) is proven similarly. **'⇐':** since $\exists x$ such that $rD(\boldsymbol{x}) \leq N(\boldsymbol{x})$, this means that $r \leq \frac{N(\boldsymbol{x})}{D(\boldsymbol{x})} \leq p^*$;

'⇒': Since P1 optimizes a continuous objective over a closed convex set, then there exists an optimal solution $\boldsymbol{x}^*$ such that $p^* = \frac{N(\boldsymbol{x}^*)}{D(\boldsymbol{x}^*)} \geq r$ which rearranging gives the result. □

Algorithm 1 describes the basic structure of the binary search method. Given the payoff matrix (P_M) and the total number of security resources ($numRes$), Algorithm 1 first initializes the upper bound (U_0) and lower bound (L_0) of the defender expected utility on Line 2. Then, in each iteration, r is set to be the mean of U and L. Line 6 checks whether the current r satisfies Equation (2). If so, $p^* \geq r$, the lower-bound of the binary search needs to be increased; in this case, it also returns a valid strategy x^r. Otherwise, $p^* < r$, the upper-bound of the binary search should be decreased. The search continues until the upper-bound and lower-bound are sufficiently close, i.e. $U - L < \epsilon$. The number of iterations in Algorithm 1 is bounded by $O(\log(\frac{U_0 - L_0}{\epsilon}))$. Specifically for SSGs we can estimate the upper and lower bounds as follows:

Algorithm 1: Binary Search

```
1  Input: $\epsilon$, $P_M$ and $numRes$;
2  $(U_0, L_0) \leftarrow$ EstimateBounds($P_M, numRes$);
3  $(U, L) \leftarrow (U_0, L_0)$;
4  while $U - L \geq \epsilon$ do
5      $r \leftarrow \frac{U+L}{2}$;
6      ($feasible$, $\boldsymbol{x}^r$) $\leftarrow$ CheckFeasibility($r$);
7      if $feasible$ then
8          $L \leftarrow r$
9      else
10         $U \leftarrow r$
11 return $L, \boldsymbol{x}^L$;
```

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

Upper bound: Since $P_i^d \leq U_i^d \leq R_i^d$ we have $U_i^d \leq \max_{i\in\mathcal{T}} R_i^d$. The defender's utility is computed as $\sum_{i\in\mathcal{T}} q_i U_i^d$, where U_i^d is the defender utility on target i and q_i is the probability that the adversary attacks target i. Thus, the maximum R_i^d serves as an upper bound of U_i^d.

We now turn to feasibility checking, which is performed in Step 6 in Algorithm 1. Given a real number $r \in \mathscr{R}$, in order to check whether Equation (2) is satisfied, we introduce CF-OPT.

$$\texttt{CF-OPT:} \qquad \min_{\boldsymbol{x}\in\mathscr{X}_f} rD(\boldsymbol{x}) - N(\boldsymbol{x})$$

Let δ^* be the optimal objective function of the above optimization problem. If $\delta^* \leq 0$, Equation (2) must be true. Therefore, by solving the new optimization problem and checking if $\delta^* \leq 0$, we can answer if a given r is larger or smaller than the global maximum. However, the objective function in CF-OPT is still non-convex, therefore, solving it directly is still a hard problem. We introduce two methods to address this in the next two sections.

4. GOSAQ: ALGORITHM 1 + VARIABLE SUBSTITUTION

We now present Global Optimal Strategy Against Quantal response (GOSAQ), which adapts Algorithm 1 to efficiently solve problems P1 and P2. It does so through the following nonlinear invertible change of variables:

$$y_i = e^{-\beta_i x_i}, \forall i \in \mathcal{T} \tag{3}$$

4.1 GOSAQ with No Assignment Constraint

We first focus on applying GOSAQ to solve P1 for problems with no resource assignment constraints. Here, GOSAQ uses Algorithm 1, but with a *rewritten* CF-OPT as follows given the above variable substitution:

$$\begin{aligned}
\min_{\boldsymbol{y}} \quad & r\sum_{i\in\mathcal{T}} \theta_i y_i - \sum_{i\in\mathcal{T}} \theta_i P_i^d y_i + \sum_{i\in\mathcal{T}} \frac{\alpha_i\theta_i}{\beta_i} y_i \ln(y_i) \\
\text{s.t.} \quad & \sum_{i\in\mathcal{T}} \frac{-1}{\beta_i}\ln(y_i) \leq M \qquad (4) \\
& e^{-\beta_i} \leq y_i \leq 1, \qquad \forall i \qquad (5)
\end{aligned}$$

Let's refer to the above optimization problem as GOSAQ-CP.

Lemma 2. *Let* $\mathtt{Obj}_{CF}(\mathbf{x})$ *and* $\mathtt{Obj}_{GC}(\mathbf{y})$ *be the objective function of CF-OPT and GOSAQ-CP respectively;* $\mathscr{X}_f$ *and* $\mathscr{Y}_f$ *denote the feasible domain of CF-OPT and GOSAQ-CP respectively:*

$$\min_{\mathbf{x}\in\mathscr{X}_f} \mathtt{Obj}_{CF}(\mathbf{x}) = \min_{y\in\mathscr{Y}_f} \mathtt{Obj}_{GC}(\mathbf{y}) \tag{6}$$

The proof, omitted for brevity, follows from the variable substitution in equation 6. Lemma 2 indicates that solving GOSAQ-CP is equivalent to solving CF-OPT. We now show that GOSAQ-CP is actually a convex optimization problem.

Lemma 3. *GOSAQ-CP is a convex optimization problem with a unique optimal solution.*

PROOF. We can show that both the objective function and the nonlinear constraint function (4) in GOSAQ-CP are strictly convex by taking second derivatives and showing that the Hessian matrices are positive definite. The fact that the objective is strictly convex implies that it can have only one optimal solution. □

In theory, convex optimization problems like the one above, can be solved in polynomial time through the ellipsoid method or interior point method with the volumetric barrier function [2] (in practice there are a number of nonlinear solvers capable of finding the only KKT point efficiently). Hence, GOSAQ entails running Algorithm 1, performing Step 6 with $O(\log(\frac{U_0-L_0}{\epsilon}))$ times, and each time solving GOSAQ-CP which is polynomial solvable. Therefore, GOSAQ is a polynomial time algorithm.

We now show the bound of GOSAQ's solution quality.

Lemma 4. *Let L^* and U^* be the lower and upper bounds of* GOSAQ *when the algorithm stops, and* $\mathbf{x}^*$ *is the defender strategy returned by* GOSAQ*. Then,*

$$L^* \leq Obj_{P1}(\mathbf{x}^*) \leq U^*$$

where $Obj_{P1}(\mathbf{x})$ *denotes the objective function of* P1*.*

PROOF. Given r, Let $\delta^*(r)$ be the minimum value of the objective function in GOSAQ-CP. When GOSAQ stops, we have $\delta^*(L^*) \leq 0$, because from Lines 6-8 of Algorithm 1, updating the lower bound requires it. Hence, from Lemma 2, $L^*D(\boldsymbol{x}^*) - N(\boldsymbol{x}^*) \leq 0 \Rightarrow L^* \leq \frac{N(\boldsymbol{x}^*)}{D(\boldsymbol{x}^*)}$. Similarly, $\delta^*(U^*) \geq 0 \Rightarrow U^* > \frac{N(\boldsymbol{x}^*)}{D(\boldsymbol{x}^*)}$ □

Theorem 1. *Let* $\mathbf{x}^*$ *be the defender strategy computed by* GOSAQ*,*

$$0 \leq p^* - Obj_{P1}(\mathbf{x}^*) \leq \epsilon \tag{7}$$

PROOF. p^* is the global maximum of P1, so $p^* \geq Obj_{P1}(\boldsymbol{x}^*)$. Let L^* and U^* be the lower and upper bound when GOSAQ stops. Based on Lemma 4, $L^* \leq Obj_{P1}(\boldsymbol{x}^*) \leq U^*$. Simultaneously, Algorithm 1 indicates that $L^* \leq p^* \leq U^*$.

Therefore, $0 \leq p^* - Obj_{P1}(\boldsymbol{x}^*) \leq U^* - L^* \leq \epsilon$ □

Theorem 1 indicates that the solution obtained by GOSAQ is an ϵ-optimal solution.

4.2 GOSAQ with Assignment Constraints

In order to address the assignment constraints, we need to solve P2. Note that the objective function of P2 is the same as that of P1. The difference lies in the extra constraints which enforce the marginal coverage to be feasible. Therefore we once again use Algorithm 1 with variable substitution given in Equation 3, but modify GOSAQ-CP as follows (which is referred as GOSAQ-CP-C) to incorporate the extra constraints:

$$\begin{aligned}
\min_{\boldsymbol{y},a} \quad & r\sum_{i\in\mathcal{T}} \theta_i y_i - \sum_{i\in\mathcal{T}} \theta_i P_i^d y_i + \sum_{i\in\mathcal{T}} \frac{\alpha_i\theta_i}{\beta_i} y_i \ln(y_i) \\
\text{s.t.} \quad & \texttt{Constraint } (4), (5) \\
& \frac{-1}{\beta_i}\ln(y_i) = \sum_{A_j\in\mathcal{A}} a_j A_{ij}, \quad \forall i \in \mathcal{T} \qquad (8) \\
& \sum_{A_j\in\mathcal{A}} a_j = 1 \qquad (9) \\
& 0 \leq a_j \leq 1, \quad A_j \in \mathcal{A} \qquad (10)
\end{aligned}$$

Equation (8) is a nonlinear equality constraint that makes this optimization problem non-convex. There are no known polynomial time algorithms for generic non-convex optimization problems, which can have multiple local minima. We can attempt to solve such non-convex problems using one of the efficient nonlinear solvers but we would obtain a KKT point which can be only locally optimal. There are a few research grade global solvers for non-convex programs, however they are limited to solving specific problems or small instances. Therefore, in the presence of assignment constraints, GOSAQ is no longer guaranteed to return the optimal solution as we might be left with locally optimal solutions when solving the subproblems GOSAQ-CP-C.

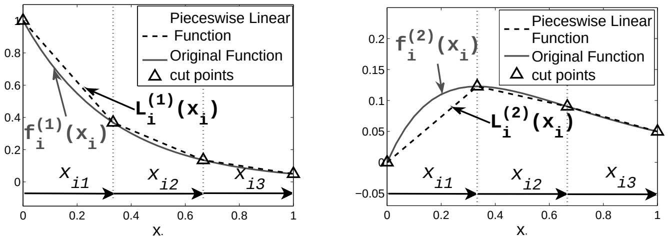
5. PASAQ: ALGORITHM 1 + LINEAR APPROXIMATION

Since GOSAQ may be unable to provide a quality bound in the presence of assignment constraints (and as shown later, may turn out to be inefficient in such cases), we propose the Piecewise linear Approximation of optimal Strategy Against Quantal response (PASAQ). PASAQ is an algorithm to compute the approximate optimal defender strategy. PASAQ has the same structure as Algorithm 1. The key idea in PASAQ is to use a piecewise linear function to approximate the nonlinear objective function in `CF-OPT`, and thus convert it into a Mixed-Integer Linear Programming (MILP) problem. Such a problem can easily include assignment constraints giving an approximate solution for a SSG against a QR-adversary with assignment constraints.

In order to demonstrate the piecewise approximation in PASAQ, we first rewrite the nonlinear objective function of `CF-OPT` as:

$$\sum_{i\in\mathcal{T}}\theta_i(r-P_i^d)e^{-\beta_i x_i}+\sum_{i\in\mathcal{T}}\theta_i\alpha_i x_i e^{-\beta_i x_i}$$

The goal is to approximate the two nonlinear function $f_i^{(1)}(x_i)=e^{-\beta_i x_i}$ and $f_i^{(2)}(x_i)=x_i e^{-\beta_i x_i}$ as two piecewise linear functions in the range $x_i\in[0,1]$, for each $i=1..|\mathcal{T}|$. We first uniformly divide the range $[0,1]$ into K pieces (segments). Simultaneously, we introduce a set of new variables $\{x_{ik},k=1..K\}$ to represent the portion of x_i in each of the K pieces, $\{[\frac{k-1}{K},\frac{k}{K}],k=1..K\}$. Therefore, $x_{ik}\in[0,\frac{1}{K}],\forall k=1..K$ and $x_i=\sum_{k=1}^{K}x_{ik}$. In order to ensure that $\{x_{ik}\}$ is a valid partition of x_i, all x_{ik} must satisfy: $x_{ik}>0$ only if $x_{ik'}=\frac{1}{K},\forall k'<k$. In other words, x_{ik} can be non-zero only when all the previous pieces are completely filled. Figures 1(a) and 1(b) display two examples of such a partition.



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

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

Figure 1: Piecewise Linear Approximation

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

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

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

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

where, $\{\mu_{ik},k=1..K\}$ is the slope of each linear segment.

Table 3: Notations for Error Bound Proof

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

5.1 PASAQ with No Assignment Constraint

In domains without assignment constraints, PASAQ consists of Algorithm 1, but with `CF-OPT` rewritten as follows:

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

$$\text{s.t.}\ \sum_{i\in\mathcal{T}}\sum_{k=1}^{K}x_{ik}\leq M \tag{11}$$

$$0\leq x_{ik}\leq\frac{1}{K},\quad\forall i,\quad k=1\dots K \tag{12}$$

$$z_{ik}\frac{1}{K}\leq x_{ik},\quad\forall i,\quad k=1\dots K-1 \tag{13}$$

$$x_{i(k+1)}\leq z_{ik},\quad\forall i,\quad k=1\dots K-1 \tag{14}$$

$$z_{ik}\in\{0,1\},\quad\forall i,\quad k=1\dots K-1 \tag{15}$$

Let's refer to the above MILP formulation as `PASAQ-MILP`.

Lemma 5. *The feasible region for* $\mathbf{x}=\langle x_i=\sum_{k=1}^{K}x_{ik},i\in\mathcal{T}\rangle$ *of* `PASAQ-MILP` *is equivalent to that of* `P1`

JUSTIFICATION. The auxiliary integer variable z_{ik} indicates whether or not $x_{ik}=\frac{1}{K}$. Equation (13) enforces that $z_{ik}=0$ only when $x_{ik}<\frac{1}{K}$. Simultaneously, Equation (14) enforces that $x_{i(k+1)}$ is positive only if $z_{ik}=1$. Hence,$\{x_{ik},k=1..K\}$ is a valid partition of x_i and $x_i=\sum_{k=1}^{K}x_{ik}$ and that $x_i\in[0,1]$. Thus, the feasible region of `PASAQ-MILP` is equivalent to `P1`

Lemma 5 shows that the solution provided by PASAQ is in the feasible region of `P1`. However, PASAQ approximates the minimum value of `CF-OPT` by using `PASAQ-MILP`, and furthermore solves `P1` approximately using binary search. Hence, we need to show an error bound on the solution quality of PASAQ.

We first show Lemma 6, 7 and 8 on the way to build the proof for the error bound. Due to space constraints, many proofs are abbreviated; full proofs are available in an on-line appendix[1]. Further, we define two constants which are decided by the game payoffs: $C_1=(\overline{\theta}/\underline{\theta})e^{\overline{\beta}}\{(\overline{R^d}+\overline{P^d})\overline{\beta}+\overline{\alpha}\}$ and $C_2=1+(\overline{\theta}/\underline{\theta})e^{\overline{\beta}}$. The notation used is defined in Table 3. In the following, we are interested in obtaining a bound on the difference between p^* (the global optimal obtained from `P1`) and $Obj_{P1}(\tilde{\mathbf{x}}^*)$, where $\tilde{\mathbf{x}}^*$ is the strategy obtained from PASAQ. However, along the way, we have to obtain a bound for the difference between $Obj_{P1}(\tilde{\mathbf{x}}^*)$ and its corresponding piecewise linear approximation $\tilde{Obj}_{P1}(\tilde{\mathbf{x}}^*)$.

Lemma 6. *Let* $\tilde{N}(\mathbf{x})=\sum_{i\in\mathcal{T}}\theta_i\alpha_i L_i^{(2)}(x_i)+\sum_{i\in\mathcal{T}}\theta_i P_i^d L_i^{(1)}(x_i)$ *and* $\tilde{D}(\mathbf{x})=\sum_{i\in\mathcal{T}}\theta_i L_i^{(1)}(x_i)>0$ *be the piecewise linear approximation of* $N(\mathbf{x})$ *and* $D(\mathbf{x})$ *respectively. Then,* $\forall\mathbf{x}\in\mathscr{X}_f$

$$|N(\mathbf{x})-\tilde{N}(\mathbf{x})|\leq(\overline{\theta}\overline{\alpha}+\overline{P^d}\overline{\theta}\overline{\beta})\frac{|\mathcal{T}|}{K}$$

$$|D(\mathbf{x})-\tilde{D}(\mathbf{x})|\leq\overline{\theta}\overline{\beta}\frac{|\mathcal{T}|}{K}$$

[1] http://anon-aamas2012.webs.com/FullProof.pdf

Lemma 7. *The difference between the objective funciton of* P1, $Obj_{P1}(\mathbf{x})$*, and its corresponding piecewise linear approximation,* $\tilde{Obj}_{P1}(\mathbf{x})$*, is less than* $C_1\frac{1}{K}$

PROOF.

$$|Obj_{P1}(\boldsymbol{x}) - \tilde{Obj}_{P1}(\boldsymbol{x})| = |\frac{N(\boldsymbol{x})}{D(\boldsymbol{x})} - \frac{\tilde{N}(\boldsymbol{x})}{\tilde{D}(\boldsymbol{x})}|$$

$$= |\frac{N(\boldsymbol{x})}{D(\boldsymbol{x})} - \frac{N(\boldsymbol{x})}{\tilde{D}(\boldsymbol{x})} + \frac{N(\boldsymbol{x})}{\tilde{D}(\boldsymbol{x})} - \frac{\tilde{N}(\boldsymbol{x})}{\tilde{D}(\boldsymbol{x})}|$$

$$\leq \frac{1}{\tilde{D}(\boldsymbol{x})}(|Obj_{P1}(\boldsymbol{x})||D(\boldsymbol{x}) - \tilde{D}(\boldsymbol{x})| + |N(\boldsymbol{x}) - \tilde{N}(\boldsymbol{x})|)$$

Based on Lemma 6, $|Obj_{P1}(\boldsymbol{x})| \leq \overline{R^d}$, and $\tilde{D}(\boldsymbol{x}) \geq |\mathcal{T}|\underline{\theta}e^{-\overline{\beta}}$.

$$|Obj_{P1}(\boldsymbol{x}) - \tilde{Obj}_{P1}(\boldsymbol{x})| \leq C_1\frac{1}{K} \qquad \square$$

Lemma 8. *Let* $\tilde{L}^*$ *and* L^* *be final lower bound of* PASAQ *and* GOSAQ*,*

$$L^* - \tilde{L}^* \leq C_1\frac{1}{K} + C_2\epsilon$$

Lemma 9. *Let* $\tilde{L}^*$ *and* $\tilde{U}^*$ *be the final lower and upper bounds of* PASAQ*, and* $\tilde{\mathbf{x}}^*$ *is the defender strategy returned by* PASAQ*. Then,*

$$\tilde{L}^* \leq \tilde{Obj}_{P1}(\tilde{\mathbf{x}}^*) \leq \tilde{U}^*$$

Theorem 2. *Let* $\tilde{x}^*$ *be the defender strategy computed by* PASAQ*,* p^* *is the global optimal defender expected utility,*

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

PROOF. The first inequality is implied since $\tilde{x}^*$ is a feasible solution. Furthermore,

$$p^* - Obj_{P1}(\tilde{x}^*) = (p^* - L^*) + (L^* - \tilde{L}^*) + (\tilde{L}^* - \tilde{Obj}_{P1}(\tilde{x}^*)) + (\tilde{Obj}_{P1}(\tilde{x}^*) - Obj_{P1}(\tilde{x}^*))$$

Algorithm 1 indicates that $L^* \leq p^* \leq U^*$, hence $p^* - L^* \leq \epsilon$. Additionally, Lemma 7, 8 and 9 provide an upper bound on $\tilde{Obj}_{P1}(\tilde{x}^*) - Obj_{P1}(\tilde{x}^*)$, $L^* - \tilde{L}^*$ and $\tilde{L}^* - \tilde{Obj}_{P1}(\tilde{x}^*)$, therefore

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

Theorem 2 suggests that, given a game instance, the solution quality of PASAQ is bounded linearly by the binary search threshold ϵ and the piecewise linear accuracy $\frac{1}{K}$. Therefore the PASAQ solution can be made arbitrarily close to the optimal solution with sufficiently small ϵ and sufficiently large K.

5.2 PASAQ With Assignment Constraints

In order to extend PASAQ to handle the assignment constraints, we need to modify PASAQ-MILP as the follows, referred to as PASAQ-MILP-C,

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

s.t. Constraint $(11) - (15)$

$$\sum_{k=1}^{K} x_{ik} = \sum_{A_j\in\mathcal{A}} a_j A_{ij}, \quad \forall i \in \mathcal{T} \tag{16}$$

$$\sum_{A_j\in\mathcal{A}} a_j = 1 \tag{17}$$

$$0 \leq a_j \leq 1, \quad A_j \in \mathcal{A} \tag{18}$$

PASAQ-MILP-C is an MILP so it can be solved optimally with any MILP solver (e.g. CPLEX). We can prove, similarly as we did for Lemma 5, that the above MILP formulation has the same feasible region as P2. Hence, it leads to a feasible solution of P2. Furthermore, the error bound of PASAQ relies on the approximation accuracy of the objective function by the piecewise linear function and the fact that the subproblem PASAQ-MILP-C can be solved optimally. Both conditions have not changed from the cases without assignment constraints to the cases with assignment constraints. Hence, the error bound is the same as that shown in Theorem 2.

6. EXPERIMENTS

We separate our experiments into two sets: the first set focuses on the cases where there is no constraint on assigning the resources; the second set focuses on cases with assignment constraints. In both sets, we compare the solution quality and runtime of the two new algorithms, GOSAQ and PASAQ, with the previous benchmark algorithm BRQR. The results were obtained using CPLEX to solve the MILP for PASAQ. For both BRQR and GOSAQ, we use the MATLAB toolbox function fmincon to solve nonlinear optimization problems[2]. All experiments were conducted on a standard 2.00GHz machine with 4GB main memory. For each setting of the experiment parameters (i.e. number of targets, amount of resources and number of assignment constraints), we tried 50 different game instances. In each game instance, payoffs R_i^d and R_i^a are chosen uniformly randomly from 1 to 10, while P_i^d and P_i^a are chosen uniformly randomly from -10 to -1; feasible assignments A_j are generated by randomly setting each element A_{ij} to 0 or 1. For the parameter λ of the quantal response in Equation (1), we used the same value ($\lambda = 0.76$) as reported in [18].

6.1 No Assignment Constraints

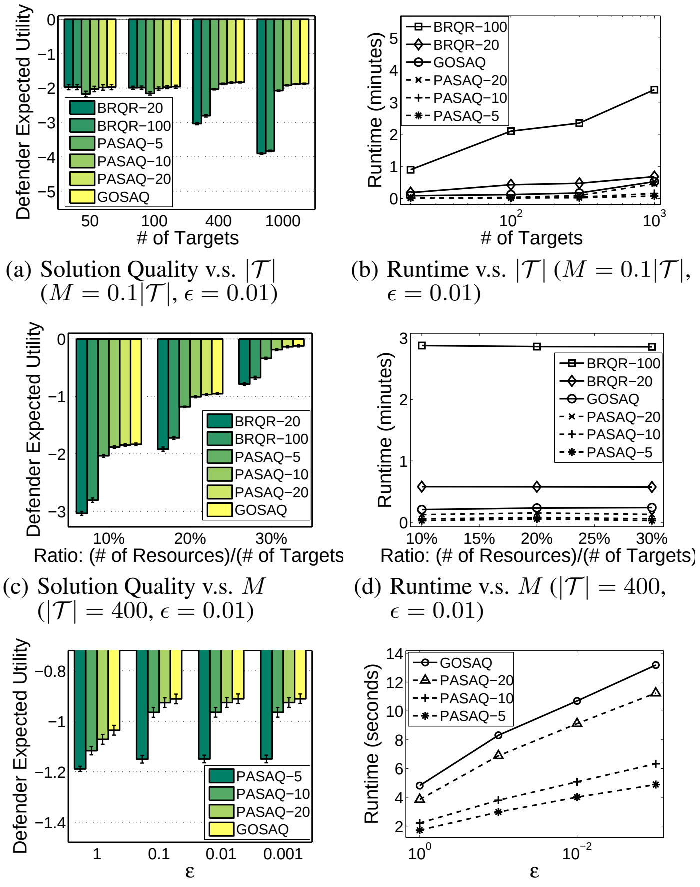
We first present experimental results comparing the solution quality and runtime of the three algorithms (GOSAQ,PASAQ and BRQR) in cases without assignment constraints.

Solution Quality: For each game instance, GOSAQ provides the ϵ-optimal defender expected utility, BRQR presents the best local optimal solution among all the local optimum it finds, and PASAQ leads to an approximated global optimal solution. We measure the solution quality of different algorithms using average defender's expected utility over all the 50 game instances.

Figures 2(a), 2(c) and 2(e) show the solution quality results of different algorithms under different conditions. In all three figures, the average defender expected utility is displayed on the y-axis. On the x-axis, Figure 2(a) changes the numbers of targets ($|\mathcal{T}|$) keeping the ratio of resources (M) to targets and ϵ fixed as shown in the caption; Figure 2(c) changes the ratio of resources to targets fixing targets and ϵ as shown; and Figure 2(e) changes the value of the binary search threshold ϵ. Given a setting of the parameters ($|\mathcal{T}|$, M and ϵ), the solution qualities of different algorithms are displayed in a group of bars. For example, in Figure 2(a), $|\mathcal{T}|$ is set to 50 for the leftmost group of bars, M is 5 and $\epsilon = 0.01$. From left to right, the bars show the solution quality of BRQR (with 20 and 100 iterations), PASAQ (with 5,10 and 20 pieces) and GOSAQ.

Key observations from Figures 2(a), 2(c) and 2(e) include: (i) The solution quality of BRQR drops quickly as the number of targets increases; increasing the number of iterations in BRQR improves the solution quality, but the improvement is very small. (ii) The solution quality of PASAQ improves as the number of pieces increases;

[2] We also tried the KNITRO [3] solver. While it gave the same solution quality as fmincon, it was three-times slower than fmincon; as a result we report results with fmincon



(a) Solution Quality v.s. $|\mathcal{T}|$ ($M = 0.1|\mathcal{T}|$, $\epsilon = 0.01$)

(b) Runtime v.s. $|\mathcal{T}|$ ($M = 0.1|\mathcal{T}|$, $\epsilon = 0.01$)

(c) Solution Quality v.s. M ($|\mathcal{T}| = 400$, $\epsilon = 0.01$)

(d) Runtime v.s. M ($|\mathcal{T}| = 400$, $\epsilon = 0.01$)

(e) Solution Quality v.s. ϵ ($|\mathcal{T}| = 400$, $M = 80$)

(f) Runtime v.s. ϵ ($|\mathcal{T}| = 400$, $M = 80$)

Figure 2: Solution Quality and Runtime Comparison, without assignment constraints (better in color)

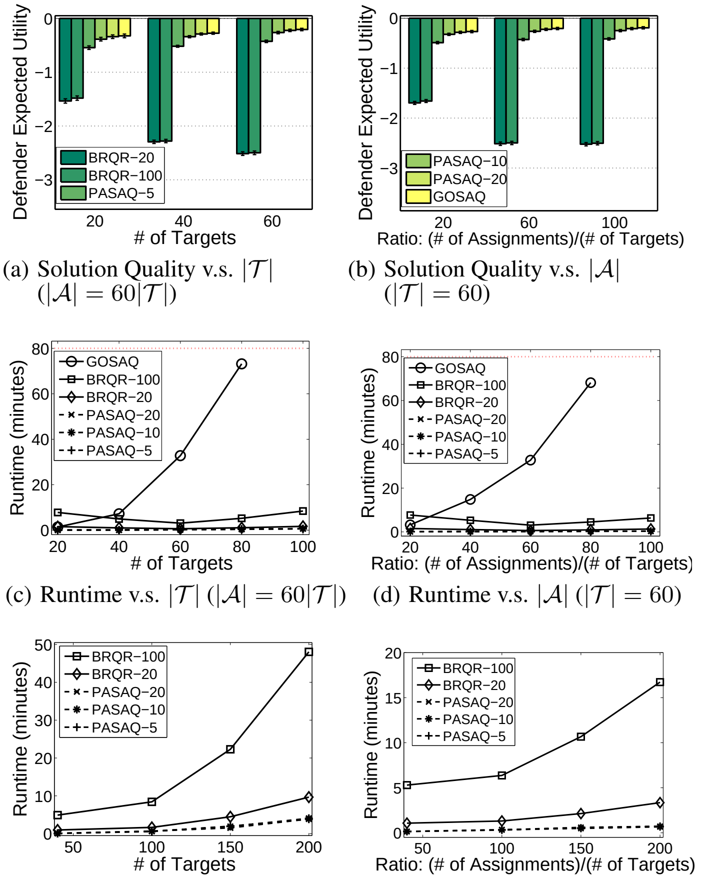
and it converges to the GOSAQ solution as the number of pieces becomes larger than 10. (iii) As the number of resources increases, the defender expected utility also increases; and the resource count does not impact the relationship of solution quality between different algorithms. (iv) As ϵ becomes smaller, the solution quality of both GOSAQ and PASAQ improves. However, after epsilon becomes sufficiently small (≤ 0.1), no substantial improvement is achieved by further decreasing the value of ϵ. In other words, the solution quality of both GOSAQ and PASAQ converges.

In general, BRQR has the worst solution quality; GOSAQ has the best solution quality. PASAQ achieves almost the same solution quality as GOSAQ when it uses more than 10 pieces.

Runtime: We present the runtime results in Figures 2(b), 2(d) and 2(f). In all three figures, the y-axis display the runtime, the x-axis displays the variables which we vary to measure their impact on the runtime of the algorithms. For BRQR run time is the sum of the run-time across all its iterations.

Figure 2(b) shows the change in runtime as the number of targets increases. The number of resources and the value of ϵ are shown in the caption. BRQR with 100 iterations is seen to run significantly slower than GOSAQ and PASAQ. Figure 2(d) shows the impact of the ratio of resource to targets on the runtime. The figure indicates that the runtime of the three algorithms is independent of the change in the number of resources. Figure 2(f) shows how runtime of GOSAQ and PASAQ is affected by the value of ϵ. On the x-axis, the value for ϵ decreases from left to right. The runtime increases linearly as ϵ decreases exponentially. In both Figures 2(d) and 2(f), the number of targets and resources are displayed in the caption.

Overall, the results suggest that GOSAQ is the algorithm of choice when the domain has no assignment constraints. Clearly, BRQR has the worst solution quality, and it is the slowest of the set of algorithms. PASAQ has a solution quality that approaches that of GOSAQ when the number of pieces is sufficiently large (≥ 10), and GOSAQ and PASAQ also achieve comparable runtime efficiency. Thus, in cases with no assignment constraints, PASAQ offers no advantages over GOSAQ.



(a) Solution Quality v.s. $|\mathcal{T}|$ ($|\mathcal{A}| = 60|\mathcal{T}|$)

(b) Solution Quality v.s. $|\mathcal{A}|$ ($|\mathcal{T}| = 60$)

(c) Runtime v.s. $|\mathcal{T}|$ ($|\mathcal{A}| = 60|\mathcal{T}|$)

(d) Runtime v.s. $|\mathcal{A}|$ ($|\mathcal{T}| = 60$)

(e) Runtime v.s. $|\mathcal{T}|$ ($|\mathcal{A}| = 60|\mathcal{T}|$)

(f) Runtime v.s. $|\mathcal{A}|$ ($|\mathcal{T}| = 60$)

Figure 3: Solution Quality and Runtime Comparison, with assignment constraint (better in color)

6.2 With Assignment Constraint

In the second set, we introduce assignment constraints into the problem. The feasible assignments are randomly generated. We present experimental results on both solution quality and runtime.

Solution Quality: Figures 3(a) and 3(b) display the solution quality of the three algorithms with varying number of targets ($|\mathcal{T}|$) and varying number of feasible assignments ($|\mathcal{A}|$). In both figures, the average defender expected utility is displayed on the y-axis. In Figure 3(a) the number of targets is displayed on the x-axis, and the ratio of $|\mathcal{A}|$ to $|\mathcal{T}|$ is set to 60. BRQR is seen to have very poor performance. Furthermore, there is very little gain in solution quality from increasing its number of iterations. While GOSAQ provides the best solution quality, PASAQ achieves almost identical solution quality when the number of pieces is sufficiently large (> 10). Figure 3(b) shows how solution quality is impacted by the number of feasible assignments, which is displayed on the x-axis. Specifically, the x-axis shows numbers of assignment constraints $\mathcal{A}$ to be 20 times, 60 times and 100 times the number of targets. The number of targets is set to 60. Once again, BRQR has significantly lower solution quality, and it drops as the number of assignments increases; and PASAQ again achieves almost the same solution quality as GOSAQ, as the number the number of pieces is larger than 10.

Runtime: We present the runtime results in Figures 3(c), 3(e), 3(d) and 3(f). In all experiments, we set 80 minutes as the cutoff. Figure 3(c) displays the runtime on the y-axis and the number of targets on the x-axis. It is clear that GOSAQ runs significantly slower than both PASAQ and BRQR, and slows down exponentially as the number of targets increases. Figure 3(e) shows extended runtime result of BRQR and PASAQ as the number of targets increases. PASAQ runs in less than 4 minutes with 200 targets and 12000 feasible assignments. BRQR runs significantly slower with higher number of iterations.

Overall, the results suggest that PASAQ is the algorithm of choice when the domain has assignment constraints. Clearly, BRQR has significantly lower solution quality than PASAQ. PASAQ not only has a solution quality that approaches that of GOSAQ when the number of pieces is sufficiently large (≥ 10), PASAQ is significantly faster than GOSAQ (which suffers exponential slowdown with scale-up in the domain).

7. CONCLUSION

This paper marks an advance over the state-of-the-art in security games. It goes beyond the assumption of perfect rationality of human adversaries embedded in deployed applications [7] and most of the current algorithms [1, 10] for Stackelberg security games; instead, it models the human adversaries' bounded rationality using the quantal response (QR) model. This work overcomes the difficulties in developing efficient methods to solve the massive security games in real applications, including solving a nonlinear and non-convex optimization problem and handling constraints on assigning security resources in designing defender strategies. In addressing these difficulties, key contributions in this paper include: (i) a new algorithm, GOSAQ, which guarantees the global optimal solution in computing the defender strategy against an adversary's quantal response; (ii) an efficient approximation algorithm, PASAQ, which provides more efficient computation of the defender strategy with nearly-optimal solution quality; (iii) algorithms solving problems with resource assignment constraint; (iv) proof of correctness/approximation-error of the algorithms; (v) detailed experimental results which show that both GOSAQ and PASAQ achieve much better solution quality than the benchmark algorithm (BRQR), and that PASAQ achieves much better computational efficiency than both GOSAQ and BRQR. Given these results, PASAQ is at the heart of the PROTECT system which is currently being used for the US Coast Guard in the port of Boston, and is currently being deployed in the port of New York.

8. ACKNOWLEDGMENTS

This research was supported by Army Research Office under the grant # W911NF-10-1-0185. We also thank Rajiv Maheswaran for giving valuable comments. Fernando Ordonez would also like to acknowledge the support of grant ICM/FIC P10-024-F.

9. REFERENCES

[1] N. Basiloco, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. *In AAMAS*, 2009.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, 2004.

[3] R. H. Byrd, J. Nocedal, and R. A. Waltz. Knitro: An integrated package for nonlinear optimization. In *Large-Scale Nonlinear Optimization*, pages 35–59. G. di Pillo and M. Roma, eds, Springer-Verlag, 2006.

[4] C. F. Camerer. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, Princeton, New Jersey, 2003.

[5] C. F. Camerer, T. Ho, and J. Chong. A cognitive hierarchy model of games. *QJE*, 119(3):861–898, 2004.

[6] P. A. Haile, A. Hortacsu, and G. Kosenok. On the empirical content of quantal response equilibrium. *The American Economic Review*, 98(1):180–200, March 2008.

[7] M. Jain, J. Pita, J. Tsai, C. Kiekintveld, S. Rathi, F. Ordonez, and M. Tambe. Software assistants for patrol planning at lax and federal air marshals service. *Interfaces*, 40(4):267–290, 2010.

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

[9] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *In JAIR*, 2011.

[10] J. Letchford and Y. Vorobeychik. Computing randomized security strategies in networked domains. *AARM Workshop In AAAI*, 2011.

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

[12] J. Pita, M. Tambe, C. Kiekintveld, S. Cullen, and E. Steigerwald. Guards - game theoretic security allocation on a national scale. *In AAMAS*, 2011.

[13] E. Shieh, B. An, R. Yang, M. Tambe, C. Baldwin, J. DiRenzo, B. Maule, and G. Meyer. Protect: A deployed game theoretic system to protect the ports of the united states. *In AAMAS*, 2012.

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

[15] T. Turocy. A dynamic homotopy interpretation of the logistic quantal response equilibrium correspondence. *Games and Economic Behavior*, 51(2):243–263, 2006.

[16] S. A. Vavasis. Complexity issues in global optimization: a survey. In *Handbook of Global Optimization*, pages 27–41. In R. Horst and P.M. Pardalos, editors, Kluwer, 1995.

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

[18] R. Yang, C. Kiekintveld, F. Ordonez, M. Tambe, and R. John. Improving resource allocation strategy against human adversaries in security games. *In IJCAI*, 2011.