Choices Are Not Independent: Stackelberg Security Games with Nested Quantal Response Models

Tien Mai, Arunesh Sinha

School of Computing and Information Systems, Singapore Management University
atmai@smu.edu.sg, aruneshs@smu.edu.sg

Abstract

The quantal response (QR) model is widely used in Stackelberg security games (SSG) to model a bounded rational adversary. The QR model is a model of human response from among a large variety of prominent models known as discrete choice models. QR is the simplest type of discrete choice models and does not capture commonly observed phenomenon such as correlation among choices. We introduce the nested QR adversary model (based on nested logit model in discrete choice theory) in SSG which addresses shortcoming of the QR model. We present tractable approximation of the resulting equilibrium problem with nested QR adversary. We do so by deriving an interesting property of the equilibrium problem, namely a loosely coupled split into nested problems that mirrors the nested decision making by the adversary in the nested QR model. We show that each separate nested problem can be approximated efficiently and that the loosely coupled overall problem can be solved approximately by formulating it as a discretized version of a continuous dynamic program. Finally, we conduct experiments that show the scalability and parallelizability of our approach, as well as advantages of the nested QR model.

Introduction

Discrete choice models (Train 2009) are a prominent class of models for modelling human response when faced with choices over alternatives. Indeed, many prominent behavioral game models such as quantal responding players (McKelvey and Palfrey 1995) have origins in discrete choice theory; the model adopted widely in the game theoretic works is the quantal response model (called multinomial logit (MNL) model in the discrete choice literature). MNL (McFadden 1973) is the simplest class of discrete choice models and suffers from deficiencies such as the inability to model correlations among alternatives and failure to account for observed human behavior in the presence of large number of choices. In this work, with the aim of addressing the aforementioned problems with MNL model, we present a first work on employing the nested logit (NL) discrete choice model (Ben-Akiva 1973) in Stackelberg security games (Tambe 2011) and solving these efficiently.

As far as we know, there is no work on use of NL models in Stackelberg security games (SSG). We argue that the NL model is natural in SSG when a human attacker faces numerous options with correlations, as the NL model posits that the player partitions the choice set into nests, first chooses a nest, and then an alternative within the chosen nest. The model allows taking into consideration correlations between alternatives that belong to the same nest. For example, in deployed applications of SSG for wildlife security (Fang et al. 2016) the adversary chooses where to attack in a large forest with the number of choices ranging in thousands; here, intuitively, it is more natural that the adversary chooses a sub-region (and choices within a sub-region are correlated) and then chooses a specific place in that sub-region to attack. The NL model is also mathematically well-motivated and derived using concepts from generalized extreme value distributions (Train 2009). The resulting Stackelberg equilibrium problem with the NL model is quite challenging.

Our *main result* in this work is a tractable guaranteed approximation of the equilibrium computation in a SSG where the follower employs a NL response model; we call this the nested QR model in the context of SSG. We achieve this overall result via a series of contributions that reveal interesting properties of the NL model in a game. Our *first contribution* is a result that shows that the main equilibrium computation optimization splits into a two level structure, with the root optimization dependent on multiple sub-optimizations at the lower level (a generalization of bi-level optimization). The tree structure of the optimizations' dependency reflects the tree structure of the nests in the NL model. The overall defender resource budget is divided among the multiple lower level optimizations, and the overall bound on budget loosely couples the lower level optimizations.

Next, as our *second contribution*, we show that the lower level optimizations can be transformed to an *unconstrained* optimization in a single scalar variable but where the objective function can be evaluated only by solving yet another multi-variable optimization (which is convex). We solve each lower level optimization by performing a grid search on the single variable, where the grid points are determined adaptively by bounding the gradient at the prior explored grid point. This provides a large computational saving over a naive set of grid points constructed at regular intervals. We also provide approximation bounds for this approach.

Our *third contribution* is a dynamic programming based algorithm to solve the root optimization via a recursive for-

mulation of the problem in terms of resource budget allotted (these can be real numbers) to the lower level optimizations. As solving the continuous dynamic programming problem is intractable in practice, we solve an discretized version and show approximation bounds for the solution.

Finally, we test our algorithms over many scenarios. Our results show the practical scalability of our algorithm and the ability to exploit parallel computation, and compares it with other baselines as well as other possible design choices in our algorithm. We also show in simulation that the nested QR model better models human response when there are a large number of correlated choices, thereby, providing higher utility to the defender in a SSG. We believe this work provides a pathway to explore richer and more realistic behavioral game models for real world applications.

Related Work

Our work most closely relates to SSG with QR adversaries and discrete choice modeling, especially in product pricing.

SSG: SSGs (Tambe 2011) are a prominent class of security models that have many real world applications as well (Pita et al. 2008; Fang et al. 2017; An et al. 2012; Sinha et al. 2018). A number of behavioral models have been explored in SSG, the most studied is the QR model (Yang et al. 2011; Yang, Ordonez, and Tambe 2012; Haghtalab et al. 2016). Variations of QR have also been explored including the subjective utility QR model (Nguyen et al. 2013) and prospect theory inspired models (Kar et al. 2015). In contrast, we handle the failure of QR (and variations) to model correlation among alternatives; this shortcoming is known as the independence from irrelevant alternatives (IIA) property in discrete choice literature; IIA means that the ratio between the probabilities of choosing two alternatives does not change even if the value of the other alternatives change.

Among work on gradient based solving of saddle point problems and games (Mertikopoulos and Sandholm 2016; Mertikopoulos et al. 2018; Li et al. 2020), a closely related work (Ling, Fang, and Kolter 2019) focuses on learning and solving a zero-sum game with NL responses by players, following up on a similar work for QR model (Ling, Fang, and Kolter 2018). Our model differs as our game is general-sum and only one of the players is NL responding, while the other is perfectly rational (typical SSG setting). Another work (Perrault et al. 2020) applies similar differentiating through optimization technique for SSG with QR adversary and like all gradient based methods this guarantees only a local optimum. For the highly non-convex NL adversary problem, our focus is on solving the SSG with guaranteed approximation by exploiting the structure of the NL model.

Besides the above work, some works (Jiang et al. 2013; Cerny et al. 2020; Milec et al. 2021) have focussed on understanding other variations of the QR model in the Stackelberg setting and analyzing computational and learnability aspects of the same. These variations still do not take into account correlation among alternatives.

Discrete Choice Modeling: Discrete choice models have a long history and have been very popular in many modeling problems that involve human behavior. Among them, the nested logit model (Ben-Akiva 1973) seems to be the first attempt to overcome the IIA issue from the classical MNL model. This model is widely used in various applications in, for instance, transportation modeling (Ben-Akiva and Lerman 1985), healthcare (de Bekker-Grob, Ryan, and Gerard 2012), or revenue management (Li and Huh 2011). In fact, in the context of descriptive representation, the NL model is found empirically more widely applicable than MNL and the advantages of MNL over standard QR have been experimentally validated (Goldberg 1995; Bhat 1995; McFadden 1977). However, in prescriptive optimization (i.e., decision-making), the use of the NL is limited due to its complicated nonlinear structure.

An area where NL has been used in decision-making is product pricing problems where a firm needs to make a pricing decision for a set of products, assuming that customers select a product according to a discrete choice model. The literature of product pricing has a number of works making use of the NL (Li and Huh 2011; Gallego and Wang 2014; Rayfield, Rusmevichientong, and Topaloglu 2015). To the best of our knowledge, all the related studies in this area only solve unconstrained optimization problems (i.e. no constraints on the prices) or problems with price bounds (i.e., box constraints where the price of each product can vary freely between a lower and upper bounds). In contrast, we consider more general constraints (i.e., budget constraints) accounting for dependency between security decisions. Such constraints make our model more challenging to handle.

Model

Background and Notation

Notation: Boldface characters represent matrices (or vectors), and a_i denotes the i-th element of vector $\mathbf{a}$. We use $[m]$, for any $m \in \mathbb{N}$, to denote the set $\{1, \ldots, m\}$.

NL Model: The nested logit framework models a situation when a player is faced with options that can be grouped into N non-overlapping nests. Let the choices in each nest $n \in [N]$ be given by $\mathcal{K}_n$ and the utility of choosing $j \in \mathcal{K}_n$ be u_j. Then, the NL model posits that the player chooses option j with probability

$$q_j = \sum_{n \in [N]} \frac{W_n^{\sigma_n}}{\sum_{n' \in [N]} W_{n'}^{\sigma_{n'}}} \frac{e^{\lambda u_j}}{\sum_{j' \in \mathcal{K}_n} e^{\lambda u_{j'}}}$$

where $W_n = \sum_{j' \in \mathcal{K}_n} e^{u_{j'}}\ \forall n$, and $\sigma_n \in [0,1]\ \forall n$, $\lambda \geq 0$ are parameters. This above formulation is derived using the random utility maximization (extreme value) framework (Train 2009) and is a well-accepted discrete choice model that handles correlations within a nest n. Note that the MNL (QR) model is a special case of the above where $\sigma_n = 1$ for all $n \in [N]$. Moreover, since λ is a fixed scaling factor for utility u_j of any choice j, for sake of simplicity of notation we absorb λ into u_j and just write u_j instead of λu_j in the rest of the paper.

SSG with NL adversary

A SSG model is one where a defender aims to protect a set of targets and adversary aims to attack one of these targets.

Following the NL description above, we specify the number of targets as $\sum_{n\in[N]}|\mathcal{K}_n|$. The defender's pure strategy is to allocate M security resources to the targets. The mixed strategy is represented by the marginal probabilities of defending target j given by x_j, which should satisfy $\sum_{n\in[N]}\sum_{j\in\mathcal{K}_n} x_j \le M$ and $x_j \in [0,1]$. The vector $\mathbf{x}$ represents marginal probabilities for all targets. Following standard terminology, for every target j, if the adversary attacks j and the target is protected then the defender obtains reward r_j^d and the adversary obtains l_j^a. Conversely, if the defender is not protecting target j, then the defender obtains l_j^d ($r_j^d > l_j^d$) and the adversary gets r_j^a ($r_j^a > l_j^a$). We assume all utility parameters are upper and lower bounded by fixed constants. Given x_j, the expected utility of the defender and attacker for an attack on an operational facility j is formulated as follows: $u_j^d(x_j) = x_j r_j^d + (1-x_j)l_j^d$ and $u_j^a(x_j) = x_j l_j^a + (1-x_j)r_j^a$. Succinctly,

$$u_j^d(x_j) = \delta_j^d x_j + l_j^d \quad \text{where } \delta_j^d = r_j^d - l_j^d \ge 0$$
$$u_j^a(x_j) = -\delta_j^a x_j + r_j^a \quad \text{where } \delta_j^a = r_j^a - l_j^a \ge 0.$$

Under a nested logit adversary response model, the defender's expected utility can be formulated as

$$f(\mathbf{x}) = \sum_{n\in[N]} \frac{W_n^{\sigma_n}}{\sum_{n'\in[N]} W_{n'}^{\sigma_{n'}}} \sum_{j\in\mathcal{K}_n} \frac{e^{u_j^a(x_j)} u_j^d(x_j)}{\sum_{j'\in\mathcal{K}_n} e^{u_{j'}^a(x_{j'})}} \quad (1)$$

where $W_n = \sum_{j'\in\mathcal{K}_n} e^{u_{j'}^a(x_{j'})}$. The Stackelberg equilibrium can be computed by solving the following problem

$$\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}), \qquad \text{(OPT-0)}$$

where $\mathcal{X}$ is the feasible set of marginal probabilities $\{\mathbf{x} \in [0,1]^M \mid \sum_{n\in[N]}\sum_{j\in\mathcal{K}_n} x_j \le M\}$.

Equilibrium Computation

We can re-formulate the optimization (OPT-0) as

$$\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) = \max\{\delta \mid \exists \mathbf{x}\in\mathcal{X}, f(\mathbf{x}) \ge \delta\}$$

Next, note that $f(\mathbf{x}) \ge \delta$ can be equivalently written as

$$\sum_{n\in[N]} \frac{W_n^{\sigma_n}}{\sum_{n'\in N} W_{n'}^{\sigma_{n'}}} \sum_{j\in\mathcal{K}_n} \frac{e^{u_j^a(x_j)} u_j^d(x_j)}{W_n} \ge \delta \text{ or equivalently}$$

$$\sum_{n\in[N]}\sum_{j\in\mathcal{K}_n} W_n^{\sigma_n - 1}\phi(x_j) \ge \delta \sum_{n'\in[N]} (W_{n'})^{\sigma_{n'}} \text{ or equivalently}$$

$$\sum_{n\in[N]} \Big(W_n^{\sigma_n - 1}\big(\sum_{j\in\mathcal{K}_n}\phi(x_j)\big) - \delta W_n^{\sigma_n}\Big) \ge 0 \quad (2)$$

where $\phi(x_j) = e^{u_j^a(x_j)} u_j^d(x_j)$ for notational convenience. Thus, the optimization is now

$$\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) = \max\{\delta \mid \exists \mathbf{x}\in\mathcal{X},\ \text{Eq. 2 holds}\}$$

Next, we perform binary search over δ above where for a given $\delta \in \mathbb{R}$, we need to check if

$$\max_{\mathbf{x}\in\mathcal{X}} \left\{ \sum_{n\in[N]} \Big(W_n^{\sigma_n - 1}\big(\sum_{j\in\mathcal{K}_n}\phi(x_j)\big) - \delta W_n^{\sigma_n}\Big)\right\} \ge 0 \quad (3)$$

The above problem is challenging as: (1) it is non-convex and (2) it is not separable in x_j's, thus, prior known methods for QR (Yang, Ordonez, and Tambe 2012) cannot be used. We address this problem in the next sub-sections.

Nested Budget Constraints

We first split the budget M into a *real-valued* budgets $M_1, \ldots, M_N$ per nest with $\sum_n M_n = M$. Then, consider the following restricted feasible set for fixed $M_1, \ldots, M_N$

$$\mathcal{X}_r = \left\{ \mathbf{x}\in[0,1]^m \Big| \sum_{j\in\mathcal{K}_n} x_j \le M_n, \forall n \in [N] \right\}$$

$\mathcal{X}_r$ has N independent sub-sets, each corresponding to the targets $\mathcal{K}_n$ in nest n. In this section, we show how to solve the maximization problem in (3) for $\mathbf{x} \in \mathcal{X}_r$ for a given $\mathcal{X}_r$ and then in the next sub-section we use this solver as a sub-routine to solve the over all problem with $\mathbf{x}\in\mathcal{X}$. With decomposable $\mathcal{X}_r$, the maximization problem in Eq. (3) decomposes into N problems, one for each nest. We show this problem for $n \in [N]$ below:

$$\max_{x_j, j\in\mathcal{K}_n} \quad W_n^{\sigma_n - 1}\Big(\sum_{j\in\mathcal{K}_n}\phi(x_j)\Big) - \delta W_n^{\sigma_n} \quad \text{(subOPT-1)}$$
$$\text{subject to} \quad \sum_{j\in\mathcal{K}_n} x_j \le M_n\ ,$$
$$x_j \in [0,1],\ \forall j \in \mathcal{K}_n$$

In the above formulation we introduce the variable W_n, then (subOPT-1) can be further formulated equivalently as

$$\max_{W_n\in[\underline{W}_n, \overline{W}_n]} W_n^{\sigma_n - 1} g(W_n) - \delta W_n^{\sigma_n} \qquad \text{(subOPT-2)}$$

where $\overline{W}_n > \underline{W}_n > 0$ are upper and lower bounds of W_n, and $g(W_n)$ is defined as

$$g(W_n) = \max_{x_j, j\in\mathcal{K}_n} \quad \sum_{j\in\mathcal{K}_n}\phi(x_j) \qquad \text{(subOPT-3)}$$
$$\text{subject to} \quad \sum_{j'\in\mathcal{K}_n} e^{u_{j'}^a(x_{j'})} = W_n\ ,$$
$$\sum_{j\in\mathcal{K}_n} x_j \le M_n\ ,$$
$$x_j \in [0,1],\ \forall j \in \mathcal{K}_n$$

First, we show that $g(W_n)$ can be computed efficiently.

Proposition 1. *Given $W_n > 0$, (subOPT-3) can be reformulated as a strict convex optimization.*

Proof. The variable transformation $y_j = e^{-\delta_j^a x_j}$, or $x_j = \frac{-\ln y_j}{\delta_j^a}$, transforms (subOPT-3) into the equivalent problem:

$$g(W_n) = \max_{y_j\in[e^{-w_j^a},1], j\in\mathcal{K}_n} \quad \sum_{j\in\mathcal{K}_n} y_j e^{l_j^a}\Big(\frac{-\delta_j^d \ln y_j}{\delta_j^a} + l_j^d\Big)$$
$$\text{subject to} \quad \sum_{j\in\mathcal{K}_n} y_j e^{l_j^a} = W_n\ ,$$
$$\sum_{j\in\mathcal{K}_n} \frac{-\ln y_j}{\delta_j^a} \le M_n\ .$$

Algorithm 1: *Subproblem*$_n(M_n)$

1 Form the grid $\{W_n^1, W_n^2, \ldots, W_n^T\}$ as stated in Theorem 1 and for all t evaluate $\Phi(W_n^t) = (W_n^t)^{\sigma_n - 1} g(W_n^t) - \delta (W_n^t)^{\sigma_n}$ using the solver for $g(\cdot)$

2 Using the above, solve (subOPT-4) to get $\widetilde{W}_n$

3 Using obtained $\widetilde{W}_n$ and given M_n solve (subOPT-3) to get x_j^* for all $j \in \mathcal{K}_n$

4 return x_j^* for all $j \in \mathcal{K}_n$

Then, it can be readily verified that the objective above is strictly concave, and the constraints are convex. □

The bounds $\underline{W}_n, \overline{W}_n$ can be readily shown to be

$$\overline{W}_n = \max \Big\{ \sum_{j' \in \mathcal{K}_n} e^{-\delta^a_{j'} x_{j'} + l^a_{j'}} \Big| \sum_{j' \in \mathcal{K}_n} x_{j'} \leq M_n \Big\}$$

$$\underline{W}_n = \min \Big\{ \sum_{j' \in \mathcal{K}_n} e^{-\delta^a_{j'} x_{j'} + l^a_{j'}} \Big| \sum_{j' \in \mathcal{K}_n} x_{j'} \leq M_n \Big\}$$

with a note that the above maximization and minimization problems can be converted into convex optimization problems using the variable transformation $y_j = e^{-\delta^a_j x_j}$.

The convex optimization (subOPT-3) always has a unique optimal solution for any $W_n \in [\overline{W}_n, \underline{W}_n]$ that can be computed to arbitrary precision in polynomial time. In practice, for better running time, we use the dual descent method for solving the convex optimization above due to the separability of the problem into the y_j's as well as a closed form solution for the Lagrangian in the primal variables. We defer description of this standard method to the appendix and also show in the appendix that evaluating g has time complexity $O(|\mathcal{K}_n|)$ assuming fixed (constant) precision of computer.

Since (subOPT-2) is single scalar variable problem, we can use grid search to find a good approximation of the optimal solution. To build an efficient grid search method, we first bound $|g(W_n + \epsilon) - g(W_n)|$, for $\epsilon > 0$.

Proposition 2. *For any W_n and $0 < \epsilon < 1$, we have*

$$|g(W_n + \epsilon) - g(W_n)| \leq \epsilon \tau_n \text{ where}$$
$$\tau_n = \max_{j \in \mathcal{K}_n} \left\{ \delta^d_j + |l^d_j| + \frac{\delta^d_j (e^{l^a_j} + 1)}{\delta^a_j e^{l^a_j - \delta^a_j}} \right\}$$

Next, let $\Phi(W_n) = W_n^{\sigma_n - 1} g(W_n) - \delta W_n^{\sigma_n}$. We bound $|\Phi(W_n + \epsilon) - \Phi(W_n)|$ using the result from Prop. 2:

Proposition 3. *For any W_n and $0 < \epsilon \leq \rho < 1$, we have*

$$|\Phi(W_n + \epsilon) - \Phi(W_n)| \leq \rho \mathcal{T}_n \text{ where}$$
$$\mathcal{T}_n = \max \Big\{ (1 - \sigma_n)(W_n)^{\sigma_n - 2} (g(W_n) + \tau_n) + \delta \sigma_n (W_n)^{\sigma_n - 1}; (W_n)^{\sigma_n - 1} \tau_n \Big\}$$

From the above result about the change of Φ with a change in its argument, we can construct a one-dimensional grid $\{W_n^1 = \underline{W}_n, W_n^2, \ldots, W_n^T \geq \overline{W}_n\}$ where the grid points are irregularly spaced at $\mathcal{G}(W_n^t, \rho)$ apart, i.e., $W_n^{t+1} = W_n^t + \mathcal{G}(W_n^t, \rho)$. The form of $\mathcal{G}(W_n^t, \rho)$ and the number of grid points T is specified in the theorem and its corollary below. Given this grid, we find an approximate solution of (subOPT-2) by searching over the grid

$$\widetilde{W}_n^* = \text{argmax}_{t=0,\ldots,T} \left\{ \Phi(W_n^t) \right\} \qquad \text{(subOPT-4)}$$

Theorem 1. *Given $\rho > 0$, if we choose $\mathcal{G}(W_n^t, \rho) = \rho / \mathcal{T}_n$ then* (subOPT-4) *returns an approximate solution $\widetilde{W}_n^*$ with*

$$|\Phi(\widetilde{W}_n^*) - \Phi(W_n^*)| \leq \rho,$$

where W_n^* is an optimal solution to (subOPT-2).

Corollary 1. *The number of points in the grid $\{W_n^1, \ldots, W_n^T\}$ is at most*

$$T \leq \frac{1}{\rho} (\overline{W}_n - \underline{W}_n) \max \left\{ (1 - \sigma_n)(\underline{W}_n)^{\sigma_n - 2} (\widetilde{g}_n + \tau_n) + \delta \sigma_n (\underline{W}_n)^{\sigma_n - 1}; (\underline{W}_n)^{\sigma_n - 1} \tau_n \right\},$$

where $\widetilde{g}_n = \max_{x_j \in [0,1], j \in \mathcal{K}_n} \Big\{ \sum_{j \in \mathcal{K}_n} \phi(x_j) \Big| \sum_{j \in \mathcal{K}_n} x_j \leq M_n \Big\}$

Approximation Algorithm Putting all the claims till now together, we state the algorithm to solve (subOPT-1) (recall, this is equivalent to (subOPT-2)) in Algorithm 1. The algorithm first solves for $\Phi(\cdot)$ (line 1, 2) using the grid approach stated in Theorem 1. Note that we need to evaluate $g(\cdot)$ in line 1, which is done using the dual descent solver stated earlier. From Theorem 1 the algorithm provides additive ρ approximate solution and from Corollary 1 it has $O(|\mathcal{K}_n|/\rho)$ time complexity. For time complexity, we used T from Corollary 1 with the earlier stated assumption that all utility parameters are bounded.

Full Budget Constraint

We now return to the original aim of solving the problem stated in Equation 3 with $x \in \mathcal{X}$. The dependency between the security variables of different subset $\mathcal{K}_n$ makes the problem more challenging. We show an efficient solution of the problem by formulating it as a dynamic programming problem where the sub-problems are the problem solved in the previous sub-section.

Dynamic Programming Formulation Recall the problem stated in Equation 3 under the full budget constraint is

$$\max_{\substack{x_j, j \in \mathcal{K}_n \\ n \in [N]}} \quad \sum_{n \in [N]} W_n^{\sigma_n - 1} \Big(\sum_{j \in \mathcal{K}_n} \phi(x_j) \Big) - \delta (W_n)^{\sigma_n}$$
$$\text{subject to} \quad \sum_{n \in [N]} \sum_{j \in \mathcal{K}_n} x_j \leq M \ ,$$
$$x_j \in [0, 1], \ \forall j \in \mathcal{K}_n, \forall n \in [N]$$

We define the following value function $V^n(\mathcal{M})$, for *given* $n \in [N]$ and $\mathcal{M} \in [0, M]$

$$V^n(\mathcal{M}) = \max_{\substack{x_j, j \in \mathcal{K}_k \\ k=n,\ldots,N}} \sum_{k=n}^{N} W_k^{\sigma_k - 1}\Big(\sum_{j \in \mathcal{K}_k} \phi(x_j)\Big) - \delta(W_k)^{\sigma_k}$$
$$\text{subject to} \quad \sum_{k=n}^{N} \sum_{j \in \mathcal{K}_k} x_j \le \mathcal{M}\ ,$$
$$x_j \in [0,1],\ \forall j \in \mathcal{K}_k, k \in \{n, \ldots, N\}$$

Let us define a notation for problem (subOPT-1)

$$\mathcal{H}^n(M_n) = \max_{x_j, j \in \mathcal{K}_n} W_n^{\sigma_n - 1}\Big(\sum_{j \in \mathcal{K}_n} \phi(x_j)\Big) - \delta(W_n)^{\sigma_n}$$
$$\text{subject to} \quad \sum_{j \in \mathcal{K}_n} x_j \le M_n\ ,$$
$$x_j \in [0,1],\ \forall j \in \mathcal{K}_n.$$

Then, it can be readily checked that $V^n(\mathcal{M})$ satisfies the Bellman equation below. Further, from standard dynamic programming set-up (Bertsekas 2011), $V^1(M)$ is the optimal value of the problem in Equation 3.

$$V^n(\mathcal{M}) = \begin{cases} \max\limits_{M_n \in [0, \mathcal{M}]} \left\{\mathcal{H}^n(M_n) + V^{n+1}(\mathcal{M} - M_n)\right\} \\ \qquad\qquad \text{for } n = 1, \ldots, N \\ 0 \qquad\qquad \text{for } n = N+1 \end{cases} \tag{DYN}$$

Approximation for the Continuous Dynamic Program The above dynamic program is based on a continuous state space, which is not tractable to handle. We build a tractable approximation algorithm by discretizing the interval $[0, M]$. By choosing a step size $\psi > 0$ we discretize $[0, M]$ into $\lceil M/\psi \rceil$ points as $\{0, \psi, 2\psi, \ldots, \lfloor M/\psi \rfloor \psi\}$. We define the approximate value function $\widehat{V}^n$ as a surrogate of V^n

$$\widehat{V}^n(k\psi) = \begin{cases} \max\limits_{k' \in \{0,1,\ldots,k\}} \left\{\widehat{\mathcal{H}}^n(k'\psi) + \widehat{V}^{n+1}((k-k')\psi)\right\} \\ \qquad\qquad \text{for } n = 1, \ldots, N \\ 0 \qquad\qquad \text{for } n = N+1 \end{cases} \tag{DYN-APPROX}$$

Note that in (DYN-APPROX) we use $\widehat{\mathcal{H}}^n(\cdot)$ instead of $\mathcal{H}^n(\cdot)$ because we solve (subOPT-1) approximately. Given the above discrete recursive formulation, a standard dynamic programming approach can solve the problem. The next result follows from standard properties of dynamic programs (Bertsekas 2011):

Proposition 4. *The dynamic programming system* (DYN-APPROX) *can be solved in* $\mathcal{O}(\lceil M/\psi \rceil N)$ *steps.*

Next we analyze the approximation errors yielded by the approximate dynamic program (DYN-APPROX). A first step is the following observation:

$$\left|\widehat{V}^1\Big(\Big\lfloor \frac{M}{\psi} \Big\rfloor \psi\Big) - V^1(M)\right| \le \left|\widehat{V}^1\Big(\Big\lfloor \frac{M}{\psi} \Big\rfloor \psi\Big) - V^1\Big(\Big\lfloor \frac{M}{\psi} \Big\rfloor \psi\Big)\right| + \left|V^1\Big(\Big\lfloor \frac{M}{\psi} \Big\rfloor \psi\Big) - V^1(M)\right|$$

Towards obtaining the bound above we bound the gaps

$$\left|\widehat{V}^n(k\psi) - V^n(k\psi)\right|,\ \forall k = 0, \ldots, \left\lfloor \frac{M}{\psi} \right\rfloor$$
$$|V^n(\mathcal{M}) - V^n(\mathcal{M} + \epsilon)|,\ \forall n = 1, \ldots, N, \epsilon \in [0, \psi], \text{ and for all } \mathcal{M} \in [0, M - \epsilon]. \tag{4}$$

We do so via a sequence of results. Recall that, from Theorem 1, we can solve (subOPT-1) with a worst-case approximation error of ρ, i.e., $|\widehat{\mathcal{H}}^n(\mathcal{M}) - \mathcal{H}^n(\mathcal{M})| \le \rho$ for all $\mathcal{M} \in [0, M]$. We first define auxiliary quantities that show up in our result: for $n = 1, \ldots, N$ define

$$J_n^{\mathcal{H}}(\psi) = \sup_{\substack{\epsilon \in [0,\psi] \\ \mathcal{M} \in [0, m-\epsilon]}} \{|\mathcal{H}^n(\mathcal{M} + \epsilon) - \mathcal{H}^n(\mathcal{M})|\}$$
$$J_n^{V}(\psi) = J_n^{\mathcal{H}}(\psi) + J_{n+1}^{V}(\psi)$$
$$J_n^{\widehat{V}}(\psi) = \rho + J_n^{\mathcal{H}}(\psi) + J_{n+1}^{\widehat{V}}(\psi) + J_{n+1}^{V}(\psi)$$

and $J_{N+1}^{V}(\psi) = 0$, $J_{N+1}^{\widehat{V}}(\psi) = 0$. The following readily follows from the above definition

Proposition 5. *$J_n^V(\psi)$ and $J_n^{\widehat{V}}(\psi)$ can be computed as*

$$J_n^V(\psi) = \sum_{n'=n}^{N} J_{n'}^{\mathcal{H}}(\psi)$$
$$J_n^{\widehat{V}}(\psi) = (N - n + 1)\rho + \sum_{n'=n}^{N} (n' + 1 - n) J_{n'}^{\mathcal{H}}(\psi)$$

Next, define the scalar

$$\mathcal{U}_n^* = (1 - \sigma_n)(\mathcal{U}_n^1)^{\sigma_n - 2}\mathcal{U}_n^2 \max_{j \in \mathcal{K}_n}\{\delta_j^a e^{l_j^a}\} + (\mathcal{U}_n^1)^{\sigma_n - 1} \max_{j \in \mathcal{K}_n}\left\{\delta_j^a e^{l_j^a}(\delta_j^d + |l_j^d|) + \delta_j^d e^{l_j^a}\right\} + \delta\sigma_n(\mathcal{U}_n^1)^{\sigma_n - 2} \max_{j \in \mathcal{K}_n}\{\delta_j^a e^{l_j^a}\}$$
$$\mathcal{U}_n^1 = \sum_{j \in \mathcal{K}_n} e^{l_j^d - \delta_j^d} \text{ and } \mathcal{U}_n^2 = \sum_{j \in \mathcal{K}_n} e^{l_j^a}(\delta_j^d + |l_j^d|).$$

We prove the following:

Lemma 1. *$J_n^{\mathcal{H}}(\psi) \le \psi \mathcal{U}_n^*$, for all n.*

We bound the required gaps (as stated above in Equation 4) using the auxiliary quantities as follows:

Lemma 2. *We have*

$$J_n^V(\psi) \ge \sup_{\substack{\epsilon \in [0,\psi] \\ \mathcal{M} \in [0, M-\epsilon]}} |V^n(\mathcal{M} + \epsilon) - V^n(\mathcal{M})|$$
$$J_n^{\widehat{V}}(\psi) \ge \max_{k=0,\ldots,\lfloor \frac{M}{\psi} \rfloor} \left|\widehat{V}^n(k\psi) - V^n(k\psi)\right| \tag{5}$$

Using the above results, we bound the approximation error of (DYN-APPROX).

Theorem 2. *$\widehat{V}^1\left(\left\lfloor \frac{M}{\psi} \right\rfloor \psi\right)$ is the optimal value obtained by solving* (DYN-APPROX)*, the approximation error can be bounded as*

$$\left|\widehat{V}^1\Big(\Big\lfloor \frac{M}{\psi} \Big\rfloor \psi\Big) - V^1(M)\right| \le N\rho + \Big(\sum_{n \in [N]} (n+1)\mathcal{U}_n^*\Big)\psi$$

Algorithm 2: *BinSearchDynProg*(M)

while $D - d > \epsilon$ **do**
  $\delta = (D + d)/2$
  Solve (DYN-APPROX) with δ, using the sub-routine *Subproblem*$_n(M_n)$ to solve $\widehat{\mathcal{H}}^n(\cdot)$
  if $\widehat{V}^1\left(\left\lfloor \frac{M}{\psi} \right\rfloor \psi\right) > 0$ **then**
    $d = \delta$
  else
    $D = \delta$
For δ, backtrack using (DYN-APPROX) recursive formulation to find M_n for all $n \in [N]$
Using the sub-routine *Subproblem*$_n(M_n)$ obtain x_j^* for all $j \in \mathcal{K}_n, \forall n \in [N]$
return x_j^* for all $j \in \mathcal{K}_n, \forall n \in [N]$

Proof. We bound the gap using Eq 4 and Lemma 2 (for the first inequality below) as below

$$\left|\widehat{V}^1\left(\left\lfloor \frac{M}{\psi} \right\rfloor \psi\right) - V^1(M)\right| \leq J_1^{\widehat{V}}(\psi) + J_1^V(\psi)$$

$$= N\rho + \sum_{n\in[N]} (n+1) J_n^{\mathcal{U}}(\psi) \overset{(a)}{\leq} N\rho + \Big(\sum_{n\in[N]} (n+1)\mathcal{U}_n^*\Big)\psi,$$

where the equality is due to Proposition 5 and (a) is due to Lemma 1. □

Corollary 2. *Given any $\epsilon > 0$, an additive ϵ-approximation solution of* (DYN) *can be obtained by solving* (DYN-APPROX) *with $\rho = \frac{\epsilon}{2N}$ and $\psi = \frac{\epsilon}{\sum_{n\in[N]} 2(n+1)\mathcal{U}_n^*}$.*

The corollary also implies that if the number of nests increases, to maintain the precision level ϵ, one needs to decrease ρ, thus increase the number of grid points.

Algorithm and Complexity Analyses We collect all our results and present the overall algorithm in Algorithm 2. As stated earlier, the overall approach is a binary search on δ which forms the while loop in the algorithm. Inside the loop, the (DYN-APPROX) is solved. After the binary search, standard backtracking (in dynamic programming) is used to obtain the optimal choices of M_n for all $n \in [N]$, which are further used to provide the final solution. Recall that (DYN-APPROX) optimizes the surrogate $\widehat{V}^1(\cdot)$ objective. Thus, the binary search provides a direct guarantee about $\widehat{V}^1(\cdot)$ and using Corollary 2 we can obtain the following guarantee for Algorithm 2.

Lemma 3. *Given choice of ρ, ψ from Corollary 2, Algorithm 2 runs in $O((1/\epsilon)\log(1/\epsilon)\lceil M/\psi\rceil N^2 \max_n |\mathcal{K}_n|)$ time and solves problem* (OPT-0) *with $O(\epsilon)$-additive approximation.*

Experiments

We denote our method (i.e., Algorithm 2) as DYN. We generate 8 random instances for each measurement that we report. For each game instance, each target is assigned randomly to a nest $\mathcal{K}_n$, $n \in [N]$, and the nest parameters σ_n are generated randomly in $[0.5, 1]$. We also choose the resource parameter as $M = \sum_{n\in[N]} |\mathcal{K}_n|/5$. All the experiments were conducted using Matlab on a PC with Intel(R) Core(TM) i7-9700 CPUs running at 3.00GHz. We use a single CPU core for all the experiments except the computational scalability part where 8 CPU cores are used to demonstrate how our algorithm scales up with parallel computing.

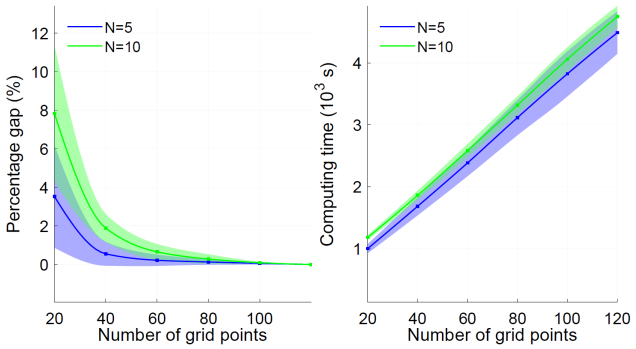

Figure 1: Performance as the number of grid points in (DYN-APPROX) increases

Analyzing the Number of Grid Points

In this experiment we determine the number of grid points needed in (DYN-APPROX) to have a good approximation. We first note that the bound for the inner optimization (subOPT-2) stated in Theorem 1 is conservative. For (subOPT-2), in order to have good approximations, we simply select ρ such that the number of grid points is 10. We then pick the interval between two consecutive points that achieves the best values and generate another set of 5 points to further improve the solution within this interval. Overall, this only requires to compute $g(W_n)$ 15 times, but provides very good approximation to the true solution.[1]

For the discretized dynamic program (DYN-APPROX), we note that the bound in Corollary 2 is too conservative and in practice we will need a much smaller number. To assess the performance of DYN w.r.t the number of grid points, we vary the number of grid points (denoted as T) from 20 to 120 and compare the percentage gaps between the resultant objective values and the objective values given by $T = 200$. Figure 1 reports those gaps as well as the computing times as T increases, with $N = 5$ and $N = 10$. We observe that with $T = 60$, the algorithm is able to reduce the gaps to under 0.5%, and with $T = 100$ the gaps will go under 0.1%. Based on this observation, we use $T = 100$ for rest of the experiments. The left plot in Figure 1 also shows that one would need more grids points to achieve a similar precision level if the number of nests N increases. This is consistent with the remark below Corollary 2. Moreover, computing time is proportional to the number of grid points, which is an expected observation.

[1] We do not know exactly the "true solution" but can obtain a near-optimal solution using a large number of grid points.

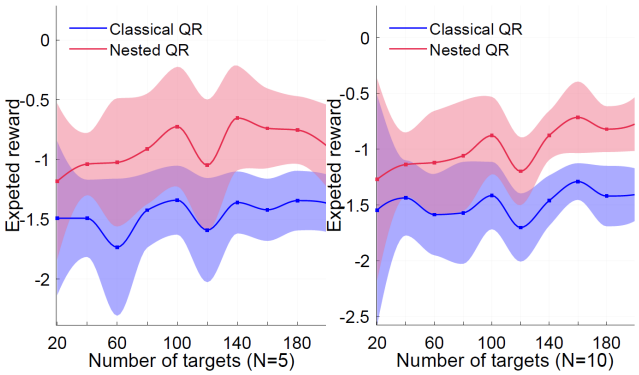

Figure 2: Performance comparison between classical QR and nested QR adversaries.

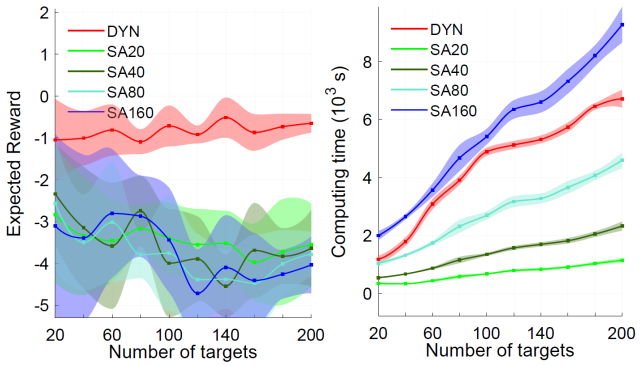

Figure 3: DYN and SA ($N = 5$)

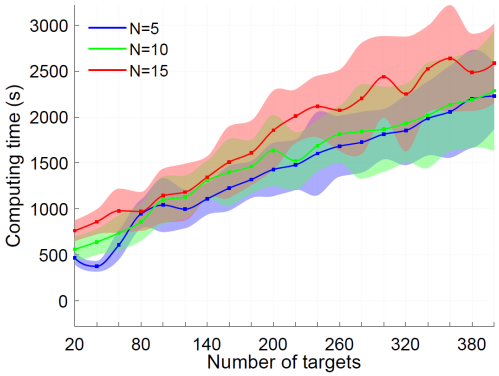

Figure 4: Scalability as the number of targets increases.

Classical QR versus Nested QR

We evaluate the performance hit if the correlation between alternatives is ignored by modelling the problem using QR. To this end, for each nested game instance, we set all the nest parameters to 1, i.e., $\sigma_n = 1$ for all $n \in [N]$ to convert the nested model into classical QR (or MNL). Such an instance can be solved efficiently by binary search and convex optimization (Yang, Ordonez, and Tambe 2012). Recall that for nested model σ_n's are chosen at random from $[0.5, 1]$. We then compare the objective values given by solutions obtained from solving classic QR instances and from nested instances in Figure 2. We see that the gaps are significant, indicating significant losses when the correlation is ignored.

DYN versus a Sampling-based Approach

Our dynamic programming approach requires generating several grid points and solving several instances of the inner problem (subOPT-2). An alternative approach that would secure a performance guarantee is to sample the nested budgets $\{M_n,\ n \in [N]\}$ such that $\sum_{n\in[N]} = M$ and solve subOPT-2 for each sample. It is then expected that this sampling approach would give good solutions if the number of samples is sufficient. In this experiment, we compare DYN with a sampling-based approach (denoted as SA) as a baseline. For the SA, we simply sample $\{M_n,\ n \in [N]\}$ and solve the inner problems for each sample and pick the best solutions. Figure 3 reports comparisons of the two approaches, in terms of objective value and computing time, where the SA approach is denoted by "SA" followed by the number of samples (denoted by $\mathcal{L}$). Here we plot the comparisons with $N = 5$ and provide a similar figure with $N = 10$ in the appendix. The objective values given by the SA are remarkably lower than those given by DYN, indicating that the numbers of samples used (up 160) are still far below a sufficient number. On the other hand, we see that SA becomes more costly with $\mathcal{L} \geq 160$. In general, the DYN clearly outperforms the SA baseline approach.

Computational Scalability

We look at the scalability of out approach as the number of targets increases. As mentioned, we use parallel computing with 8 CPU cores to compute $\widehat{\mathcal{H}}^n(k\psi)$ with different n and k in (DYN − APPROX) simultaneously. We vary the number of targets up to 400 and report the computing times in Figure 4 with $N \in \{5, 10, 15\}$. We see that the computing time scales linearly as the number of targets increases. With 400 targets, the DYN approach only needs about 40 minutes to finish. As compared to the nonparallel setting, we observe that the parallelized code can expedite our algorithm up to 4 times. We also provide a figure showing how DYN scales up as the number of nests increases in the appendix. Moreover, to further assess the scalability in much larger problem settings, we increase the number of targets to 2500 and the number of nests up to 500, and we observe that DYN needs about 2.7 and 6.1 hours to terminate with $N = 5$ and $N = 500$, respectively. This is tractable in practice and more parallel CPU cores can make the computation even faster.

Conclusion

We presented a first use of the nested logit (or nested QR) model, a well-established discrete choice model, in SSGs and an efficient approximate solution with guarantees of the same. Our scalable approximation can also tackle previously unaddressed constrained pricing problems in the product pricing literature. We hope that our work can lead to improved security in real world deployed applications, especially when adversarial choices exhibit correlations.

Acknowledgement

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2020-017), and the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant (Grant No: 20-C220-SMU-010).

References

An, B.; Shieh, E.; Tambe, M.; Yang, R.; Baldwin, C.; DiRenzo, J.; Maule, B.; and Meyer, G. 2012. PROTECT–A Deployed Game Theoretic System for Strategic Security Allocation for the United States Coast Guard. *Ai Magazine*, 33(4): 96–96.

Ben-Akiva, M. 1973. *The structure of travel demand models*. Ph.D. thesis, MIT.

Ben-Akiva, M.; and Lerman, S. R. 1985. *Discrete Choice Analysis: Theory and Application to Travel Demand*. MIT Press, Cambridge, Massachusetts.

Bertsekas, D. P. 2011. Dynamic programming and optimal control 3rd edition, volume II. *Belmont, MA: Athena Scientific*.

Bhat, C. R. 1995. A heteroscedastic extreme value model of intercity travel mode choice. *Transportation Research Part B: Methodological*, 29(6): 471–483.

Cerny, J.; Lisý, V.; Bošanský, B.; and An, B. 2020. Dinkelbach-Type Algorithm for Computing Quantal Stackelberg Equilibrium. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 246–253.

de Bekker-Grob, E. W.; Ryan, M.; and Gerard, K. 2012. Discrete choice experiments in health economics: a review of the literature. *Health economics*, 21(2): 145–172.

Fang, F.; Nguyen, T. H.; Pickles, R.; Lam, W. Y.; Clements, G. R.; An, B.; Singh, A.; Schwedock, B. C.; Tambe, M.; and Lemieux, A. 2017. PAWS—A deployed game-theoretic application to combat poaching. *AI Magazine*, 38(1).

Fang, F.; Nguyen, T. H.; Pickles, R.; Lam, W. Y.; Clements, G. R.; An, B.; Singh, A.; Tambe, M.; and Lemieux, A. 2016. Deploying PAWS: Field optimization of the protection assistant for wildlife security. In *Twenty-eighth IAAI conference*.

Gallego, G.; and Wang, R. 2014. Multiproduct price optimization and competition under the nested logit model with product-differentiated price sensitivities. *Operations Research*, 62(2): 450–461.

Goldberg, P. K. 1995. Product differentiation and oligopoly in international markets: The case of the US automobile industry. *Econometrica: Journal of the Econometric Society*, 891–951.

Haghtalab, N.; Fang, F.; Nguyen, T. H.; Sinha, A.; Procaccia, A. D.; and Tambe, M. 2016. Three Strategies to Success: Learning Adversary Models in Security Games. In *25th International Joint Conference on Artificial Intelligence (IJCAI)*.

Jiang, A. X.; Nguyen, T. H.; Tambe, M.; and Procaccia, A. D. 2013. Monotonic maximin: A robust stackelberg solution against boundedly rational followers. In *International Conference on Decision and Game Theory for Security*, 119–139. Springer.

Kar, D.; Fang, F.; Delle Fave, F.; Sintov, N.; and Tambe, M. 2015. " A Game of Thrones" When Human Behavior Models Compete in Repeated Stackelberg Security Games. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 1381–1390.

Li, H.; and Huh, W. T. 2011. Pricing multiple products with the multinomial logit and nested logit models: Concavity and implications. *Manufacturing & Service Operations Management*, 13(4): 549–563.

Li, J.; Yu, J.; Nie, Y.; and Wang, Z. 2020. End-to-end learning and intervention in games. *Advances in Neural Information Processing Systems*, 33.

Ling, C. K.; Fang, F.; and Kolter, J. Z. 2018. What Game Are We Playing? End-to-end Learning in Normal and Extensive Form Games. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 396–402.

Ling, C. K.; Fang, F.; and Kolter, J. Z. 2019. Large scale learning of agent rationality in two-player zero-sum games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6104–6111.

McFadden, D. 1973. Conditional logit analysis of qualitative choice behavior. *Frontiers in Econometrics*, 105–142.

McFadden, D. 1977. Modelling the Choice of Residential Location. *Transportation Research Record*.

McKelvey, R. D.; and Palfrey, T. R. 1995. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1): 6–38.

Mertikopoulos, P.; Lecouat, B.; Zenati, H.; Foo, C.-S.; Chandrasekhar, V.; and Piliouras, G. 2018. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *International Conference on Learning Representations*.

Mertikopoulos, P.; and Sandholm, W. H. 2016. Learning in games via reinforcement and regularization. *Mathematics of Operations Research*, 41(4): 1297–1324.

Milec, D.; Černỳ, J.; Lisỳ, V.; and An, B. 2021. Complexity and Algorithms for Exploiting Quantal Opponents in Large Two-Player Games. In *AAAI Conference on Artifical Intelligence*.

Nguyen, T.; Yang, R.; Azaria, A.; Kraus, S.; and Tambe, M. 2013. Analyzing the effectiveness of adversary modeling in security games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27.

Perrault, A.; Wilder, B.; Ewing, E.; Mate, A.; Dilkina, B.; and Tambe, M. 2020. End-to-end game-focused learning of adversary behavior in security games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 1378–1386.

Pita, J.; Jain, M.; Ordónez, F.; Portway, C.; Tambe, M.; Western, C.; Paruchuri, P.; and Kraus, S. 2008. ARMOR Security for Los Angeles International Airport. In *AAAI*, 1884–1885.

Rayfield, W. Z.; Rusmevichientong, P.; and Topaloglu, H. 2015. Approximation methods for pricing problems under the nested logit model with price bounds. *INFORMS Journal on Computing*, 27(2): 335–357.

Sinha, A.; Fang, F.; An, B.; Kiekintveld, C.; and Tambe, M. 2018. Stackelberg security games: Looking beyond a decade of success. In *27th International Joint Conference on Artificial Intelligence (IJCAI)*.

Tambe, M. 2011. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge university press.

Train, K. E. 2009. *Discrete choice methods with simulation*. Cambridge university press. https://eml.berkeley.edu/books/choice2.html.

Yang, R.; Kiekintveld, C.; Ordonez, F.; Tambe, M.; and John, R. 2011. Improving resource allocation strategy against human adversaries in security games. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 458.

Yang, R.; Ordonez, F.; and Tambe, M. 2012. Computing optimal strategy against quantal response in security games. In *11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.